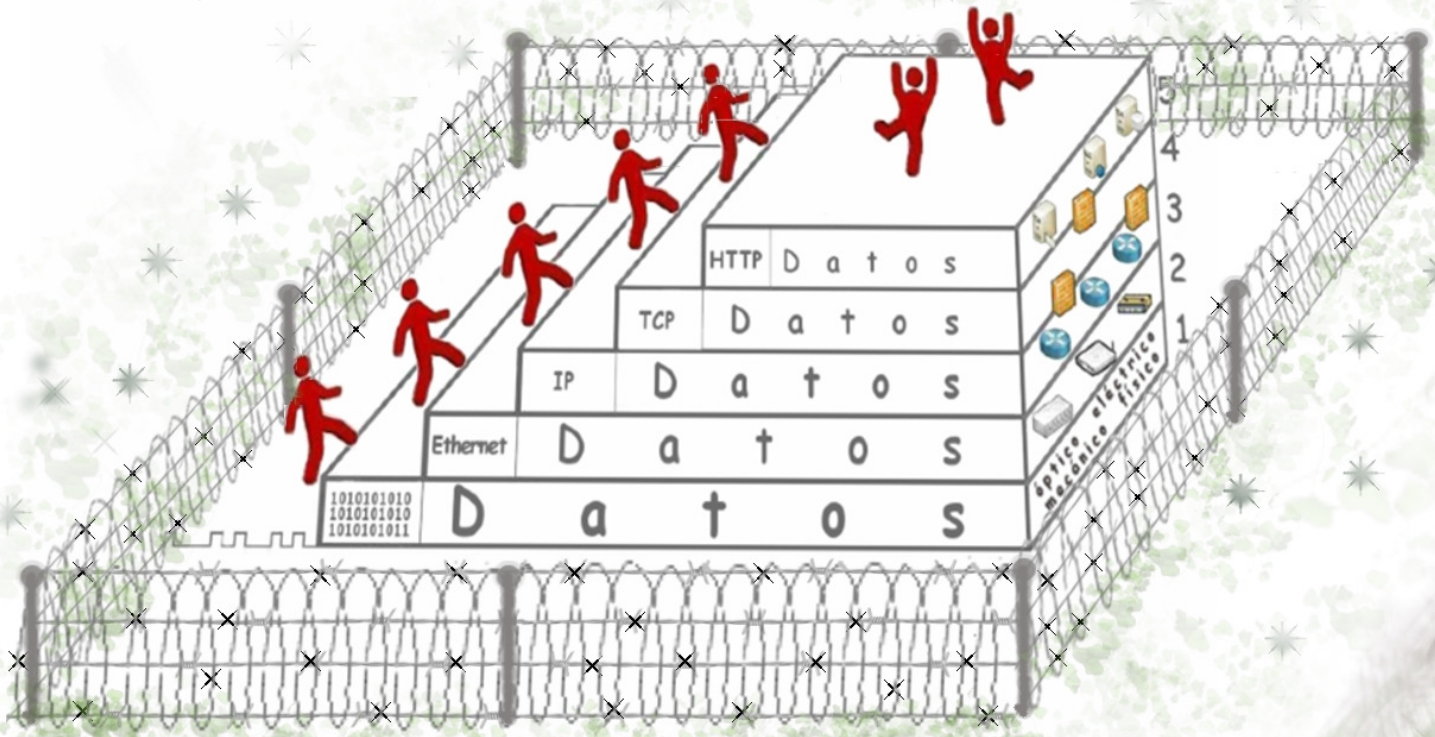
DarFE
Learning Consulting, S.L.

Seguridad Por Niveles



Alejandro Corletti Estrada
(acorletti@DarFE.es - acorletti@hotmail.com)

www.darFE.es

RPI (Madrid): 03/119554.9/11

Seguridad Por Niveles

Autor: Alejandro Corletti Estrada

(acorletti@DarFE.es - acorletti@hotmail.com)

www.darFE.es

Madrid, septiembre de 2011.

INDICE

PRÓLOGO (Por Arturo Ribagorda Garnacho)

Cuando en el lejano año 1992 la editorial de una recién aparecida revista dedicada a la seguridad de la información, por cierto la primera en su género en lengua española, me propuso encargarme de la sección bibliográfica, el problema que trimestralmente me asaltaba era elegir un libro que reseñar. Y no porque fueran muchos los publicados, sino antes bien por lo contrario.

Sin embargo, al abandonar hace tiempo esa periódica tarea, la dificultad era justo la contraria, la producción editorial era tan abundante que resultaba difícil seleccionar un puñado de los libros aparentemente más notables de donde entresacar el que a priori (o sea antes de leer) parecía merecer una recensión.

Y es que la década de los noventa del pasado siglo alumbró una tecnología, Internet (o si queremos ser rigurosos la Web) y un derecho, la protección de la privacidad (que aún estando contemplado en la Declaración Universal de los Derechos Humanos de 1948, no comienza a materializarse en leyes nacionales hasta los ochenta y sobre todo los citados noventa), que conllevaron un súbito interés por la seguridad de la información, y una paralela generalización de los estudios –universitarios o no–, de los congresos –científicos o comerciales– y, como se ha mencionado, una rica oferta de libros, manuales, enciclopedias y revistas acerca de esta materia.

Además, hoy en día, muchas de estas publicaciones lo son en formato electrónico, lo que comporta una fácil actualización que es tan importante en una materia muchas de cuyas facetas se hallan en la frontera del conocimiento, y son por tanto susceptibles de constantes avances.

Todo lo anterior, es motivo de una gran satisfacción para todos los que llevamos varias décadas estudiando e investigado en la seguridad de la información, y que ni por asomo hubiésemos imaginado este auge allá por los años ochenta del pasado siglo.

Por todo ello, es para mí un placer presentar un nuevo libro dedicado a la seguridad, escrito por un profesional de larga trayectoria, que ha ido plasmando a lo largo del tiempo en artículos e informes sus estudios y experiencias, y que finalmente ha dado el paso generoso de agruparlos y estructurarlos para conformar una publicación electrónica, que pone a disposición de los numerosos interesados sea profesionalmente o a título de meros usuarios.

Espero que su lectura completa, o consulta puntual, sirva para elevar el nivel de conocimiento en la seguridad, imprescindible para el desarrollo de esta sociedad de la información en la que nos encontramos inmersos.

Madrid, agosto de 2011

Arturo Ribagorda Garnacho
Catedrático de Universidad
Universidad Carlos III de Madrid

Presentación del libro (Por Jorge Ramió Aguirre)

No resulta fácil hacer una presentación de un libro cuando éste abarca un tema tan amplio como es el caso de "**Seguridad por niveles**", escrito por mi amigo y colega Alejandro Corletti, y que además cuenta con una extensión en páginas tan importante. No obstante, ha sido un placer leer este documento para tener una visión globalizada de su orientación y de su temática e intentar resumirlo en unas pocas palabras en esta presentación.

Lo primero que llama la atención es el título, que me parece un acierto, estableciendo que la seguridad de la información es un proceso que puede estudiarse e incluso implementarse por niveles, tomando como punto de partida los 7 niveles del modelo OSI: nivel 1 Físico, nivel 2 Enlace, nivel 3 Red, nivel 4 Transporte, nivel 5 Sesión, nivel 6 Presentación y nivel 7 Aplicación, aunque luego se plasme en el libro de acuerdo a los niveles del modelo TCP/IP.

Tal vez a más de alguno les pueda sorprender que un autor abarque el tema de la seguridad de esta manera, pero creo que es una interesante forma de marcar y delimitar uno de los terrenos principales en los que se mueve esta especialidad de la seguridad, el de las redes y sus aplicaciones.

Es así como tras una introducción a esta temática en los capítulos primero y segundo, Alejandro nos presenta desde el capítulo tercero hasta el séptimo los aspectos de seguridad que se dan en cada uno de esos niveles, incluyendo además en cada capítulo un conjunto de ejercicios prácticos muy interesantes, así como el alto número de herramientas usadas en él.

Intentar desmenuzar y resumir cada uno de esos capítulos sería demasiado extenso y tedioso para una presentación. Además de infructuoso, porque ya en el libro se observa un buen uso de la concreción y síntesis sobre lo más interesante de cada apartado. Indicar, eso sí, que la lectura y el posterior desarrollo y comprensión de esos ejemplos prácticos le permitirán al lector tener un excelente nivel de preparación en esta temática, siendo por tanto muy recomendable el uso de este documento como material de consulta en cursos de seguridad.

En seguridad siempre existirá quien ataque y quien se defienda, una especie de inevitable ying y yang; un entorno muy similar al militar que bien conoce Alejandro, quien ejerció como Jefe de Redes del Ejército Argentino durante 3 años y cuya filosofía aplica perfectamente en su libro.

A modo de anexo al grueso del libro, "Seguridad por niveles" termina con un octavo capítulo dedicado a otros conceptos de la seguridad de la información, haciendo una breve introducción a la criptografía, infraestructuras de clave pública, ataques a sistemas, IPSec, auditorías de seguridad, la familia ISO 27000 y planes de continuidad.

En resumen, más de setecientas páginas que constituyen un importante aporte a la difusión de la seguridad de la información desde la perspectiva de los niveles del modelo OSI. Y más aún por su condición de freeware y documento online que, sin lugar a dudas, verá nuevas y actualizadas ediciones en el futuro para el beneplácito de todos aquellos -profesionales o no- que creemos que la seguridad de la información debe tener un lugar muy destacado en la docencia, la investigación, la innovación tecnológica y el desarrollo empresarial e industrial de un país.

Desde Criptored damos la bienvenida a este libro y agradecemos a profesionales de la talla de Alejandro Corletti, asiduo colaborador en esta red temática con 19 documentos aportados y a punto de leer su Tesis Doctoral, que dediquen cientos de horas a publicar un libro gratuito para el provecho de miles de personas, como seguro así será el número de sus descargas desde Internet.

Jorge Ramió
Coordinador de Criptored

PRÓLOGO DEL AUTOR

La idea de escribir este libro fue la de volcar en un solo texto la cantidad de apuntes y artículos que tenía dando vueltas por Internet.

Manteniendo mi filosofía "**Open Source**" me propuse difundirlo gratuitamente para que pueda aprovecharlo todo aquel que le sea de utilidad en sus estudios.

Como todo desarrollo tecnológico de este siglo, estimo que a medida que pase el tiempo contendrá conceptos o herramientas que van quedando fuera de vigor, de ser así os ruego encarecidamente que me lo hagáis saber a través de mi correo electrónico para poder subsanarlos, también si halláis errores de forma o fondo, los cuales seguramente estarán omnipresentes como en todo escrito.

Intentaré ir actualizando esta obra, y difundiéndola como nuevas versiones: "*Seguridad_por_Niveles-v02, Seguridad_por_Niveles-v03..... Seguridad_por_Niveles-vnn*", las cuales siempre estarán disponibles en la Web: www.darFE.es, seguramente en otros sitios más y las hallarás fácilmente con cualquier buscador de Internet.

Ruego también a todo lector que pueda sumar conocimientos, conceptos, desarrollos, herramientas, etc. que también me los haga llegar por mail; todos ellos los iré incorporando, por supuesto citando su autoría y haciendo mención a su autor como "**Colaborador**" de esta obra. Siempre será bienvenido lo que sume al ámbito de la seguridad.

Por último os pido que sepáis aceptar que todo esto lo hago con la mejor buena voluntad y dentro de mis limitaciones, así que "no seáis duros con esta obra", es sencillamente una sana y humilde intención de intentar aportar algo en la red, y nada más.

He impreso un cierto número de ejemplares para cualquiera que los desee adquirir en formato papel, para ello nuevamente, invito a que se ponga en contacto conmigo, o a través de la empresa con la cuenta: info@darFE.es.

Afectuosamente:

Alejandro Corletti Estrada

(acorletti@DarFE.es - acorletti@hotmail.com)

0. PRESENTACIÓN.

Estoy absolutamente convencido que la mejor forma de avanzar con bases sólidas hacia temas de seguridad informática, es con un detallado conocimiento de la arquitectura de capas, es decir empezando la casa por los cimientos, y poco a poco ir levantando paredes, pisos hasta llegar al techo.

Por alguna razón, los precursores de las telecomunicaciones, fueron dividiendo este problema grande, en problemas menores que deberían funcionar de manera autónoma, recibiendo y entregando resultados. No es más ni menos que el viejo lema "divide y vencerás" aplicado a las nuevas tecnologías. Desde hace muchas décadas que este concepto se perfecciona y hasta podríamos decir que se "actualiza hacia la necesidad del mercado", y el viejo modelo OSI, muy pesado y lento en todos sus pasos y decisiones, está prácticamente desplazado en la "World Wide Web" para dar paso a un modelo flexible, dinámico, eficiente y ajustado al siglo XXI regulado por una serie de "Request For Commentaries (RFC)" que siguen manteniendo esta idea de capas pero mucho más efectivo.

Creo que la mejor manera de llegar a comprender los secretos de la seguridad, es analizando nivel a nivel; de esta forma, cuando comprendemos los por qué de cada encabezado, recién allí podemos decir que han fraguado las estructuras de ese piso y podemos seguir construyendo el siguiente. Es más, la experiencia me dice que suele ser muy positivo a la hora de plantearse una estrategia de seguridad, organizarse por niveles, y considerar primero qué haré a nivel físico, luego a nivel enlace, red, transporte y así recién llegar a la seguridad al nivel de aplicación, cada uno de estos niveles tiene sus medidas, conceptos y herramientas de defensa particulares. Si bien hay que destacar que hoy en día la "ambición" de la mayoría de los fabricantes, les lleva a ofrecer productos que solapan más de una capa, siempre es bueno a la hora de hacer uso de ellos, tener muy claro qué es lo que estoy haciendo, casi podríamos decir que bit a bit..... os aseguro que cuando lleguéis a pensar de esta forma, la seguridad no pasará sólo por descargar aplicaciones de Internet y hacer "clic", "clic", "clic", "clic", "clic"..... hasta causar un daño o creer que estamos seguros, sino que llegaréis al fondo de la cuestión, que es como se debe operar para llegar al éxito evitando sorpresas o "imponderables".

Este libro está presentado en dos partes:

- **PARTE I**: Conceptos y protocolos por niveles.
- **PARTE II**.: Seguridad por niveles.

En la primera parte intentaré ir avanzando con un profundo análisis de los diferentes protocolos de comunicaciones que aplican en cada capa, dejando siempre ejercicios prácticos de cada tema y en particular presentando herramientas, las cuales trataré que sean casi todas de código abierto ("Open Source") y demostraciones de cada una de ellos.

En la segunda parte, se desarrollará el conjunto de medidas que en ese nivel es conveniente tener en cuenta.

No es mi deseo en este libro hacer propaganda de un Sistema Operativo u otro, haré un marcado esfuerzo por ser imparcial. Si bien os iréis dando cuenta que la gran mayoría de las herramientas que figurarán serán de aplicación en ambiente Linux, creo también que para poder asegurar

eficientemente una infraestructura, es necesario conocer el mundo libre y el comercial, pues la realidad del mercado así lo impone. Es más, hasta he llegado a reconocer que mucha de la gente que menosprecia a Microsoft, lo hacer por no haber llegado a profundizar en las medidas de seguridad que éste ofrece, por lo tanto dejan o encuentran sistemas inseguros pero por falta de conocimiento de sus administradores, en la gran mayoría de los casos no es por otra razón y también es muy cierto que la masa de las intrusiones y/o ataques se centran en estos sistemas por ser lo más presentes en el mundo de la empresa. En resumen, a pesar de ser partidario de Linux, no voy a menospreciar ninguna plataforma, lo que sí voy a evitar es hacer propaganda de cualquier herramienta de pago, pues creo que se puede llegar al máximo nivel de seguridad con el sólo empleo de herramientas “libres”.

PARTE I

Conceptos y protocolos por niveles

CAPÍTULO 1: Introducción

1.1. Presentación de modelo de capas.

Son varios los protocolos que cooperan para gestionar las comunicaciones, cada uno de ellos cubre una o varias capas del modelo OSI (Open System interconnection); la realidad, es que para establecer la comunicación entre dos equipos Terminales de Datos (ETD) se emplea más de un protocolo, es por esta razón que se suele hablar no de protocolos aislados, sino que al hacer mención de alguno de ellos, se sobreentiende que se está hablando de una ***PILA* de protocolos**, la cual abarca más de un nivel OSI, son ejemplo de ello X.25, TCP/IP, IPX/SPX, ISDN, etc.

Una forma de agruparlos es como se encuentran cotidianamente los siete niveles del modelo OSI en tres grupos que tienen cierta semejanza en sus funciones y/o servicios:

OSI	Generalizado
Aplicación	APLICACION
Presentación	
Sesión	
Transporte	TRANSPORTE
Red	RED
Enlace	
Físico	

La ISO (International Standard Organization), estableció hace 15 años este modelo OSI que hoy lleva la denominación ISO 7498 o más conocida como X.200 de ITU.

1.2. Modelo OSI y DARPA (TCP/IP).

El modelo OSI es, sin lugar a dudas, el estándar mundial por excelencia, pero como todo esquema tan amplio presenta una gran desventaja, el enorme aparato burocrático que lo sustenta. Toda determinación, protocolo, definición o referencia que éste proponga debe pasar por una serie de pasos, en algunos casos reuniendo personal de muchos países, que demoran excesivo tiempo para la alta exigencia que hoy impone Internet. Hoy al aparecer un nuevo dispositivo, protocolo, servicio, facilidad, etc. en Internet, el mercado si es útil, automáticamente lo demanda, como ejemplo de esto hay miles de casos (chat, IRC, SMS, WAP, etc...). Si para estandarizar cualquiera de estos se tardara más de lo necesario, los fabricantes se verían en la obligación de ofrecer sus productos al mercado, arriesgando que luego los estándares se ajusten a ello, o en caso contrario, los clientes finales sufrirían el haber adquirido productos que luego son incompatibles con otros. Hoy

no se puede dar el lujo de demorar en una red cuyas exigencias son cada vez más aceleradas e imprevisibles.

Para dar respuesta a esta nueva REVOLUCION TECNOLOGICA (Internet), aparecen una serie de recomendaciones ágiles, con diferentes estados de madurez, que inicialmente no son un estándar, pero rápidamente ofrecen una guía o recomendación de cómo se cree que es la forma más conveniente (según un pequeño grupo de especialistas) de llevar a cabo cualquier novedad de la red.

Se trata aquí de las RFC (Request For Commentaries), que proponen una mecánica veloz para que el usuario final no sufra de los inconvenientes anteriormente planteados, dando respuesta a las necesidades del mercado eficientemente.

Se produce aquí un punto de inflexión importante entre el estándar mundial y lo que se va proponiendo poco a poco a través de estas RFC, las cuales en muchos casos hacen referencia al modelo OSI y en muchos otros no, apareciendo un nuevo modelo de referencia que no ajusta exactamente con lo propuesto por OSI. Este modelo se conoce como Pila, stack o familia TCP/IP o también como modelo DARPA, por la Agencia de Investigación de proyectos avanzados del DoD (Departamento de Defensa) de EEUU, que es quien inicialmente promueve este proyecto.

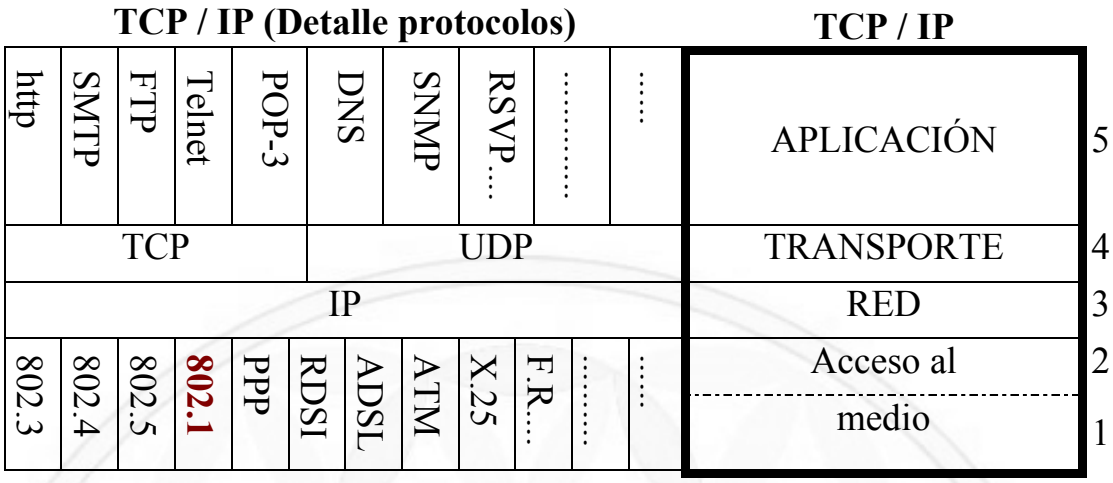
Este modelo que trata de simplificar el trabajo de las capas, y por no ser un estándar, se ve reflejado en la interpretación de los distintos autores como un modelo de cuatro o cinco capas, es más, existen filosóficos debates acerca de cómo debe ser interpretado.

En este texto, se va a tratar el mismo como un modelo de cinco capas, solamente por una cuestión práctica de cómo ajustan las mismas a los cuatro primeros niveles del modelo OSI, tratando de no entrar en la discusión Bizantina del mismo, y dejando en libertad al lector de formar su libre opinión sobre el mejor planteamiento que encuentre.

Si se representan ambos modelos, sin entrar en detalles de si las distintas capas coinciden exactamente o no (pues éste es otro gran tema de discusión, que no será tratado en este texto), se pueden graficar más o menos como se presenta a continuación:

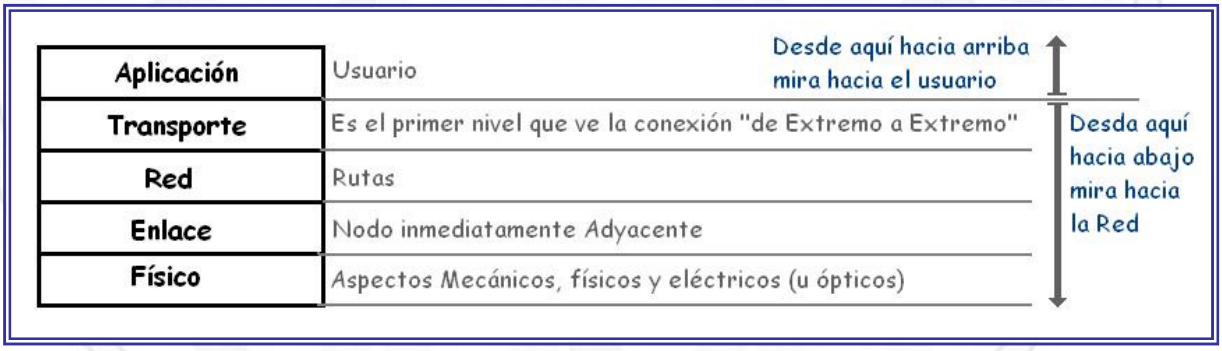
<table>
<tr><th>OSI</th><th>DARPA o TCP/IP</th></tr>
<tr><td>Aplicación</td><td rowspan="3">Aplication</td></tr>
<tr><td>Presentación</td></tr>
<tr><td>Sesión</td></tr>
<tr><td>Transporte</td><td>Transport</td></tr>
<tr><td>Red</td><td>Internetwork</td></tr>
<tr><td>Enlace</td><td>Medium Access</td></tr>
<tr><td>Físico</td><td>Phisical</td></tr>
</table>

Otra forma de presentar el modelo TCP/IP podría ser a través de una relación de los protocolos que trabajan en cada uno de los niveles (si bien algunos de ellos a veces pueden realizar funciones que superan su nivel específico), sobre esta base lo veríamos como se presenta a continuación.



En la imagen anterior, hemos presentado también el nivel de "Acceso al medio" bajo la hipótesis que pueda ser considerado como un solo nivel o como dos, pues encontraréis bibliografía que lo presenta de una u otra forma.

Antes de continuar avanzando sobre el concepto de capas, vamos a presentar una idea que sería fundamental no olvidarla y mantener siempre presente. Cada capa regula, o es encargada de una serie de funciones que deberían ser "autónomas" (cosa que a veces no se cumple), es decir no tendría por qué depender de lo que se haga en otro nivel. Dentro de este conjunto de tareas, es necesario destacar la razón de ser de cada una de ellas, su objetivo principal, el cual lo podríamos resumir en el cuadro siguiente:



Sobre el cuadro anterior insistiremos durante todo el texto, pues será la base del entendimiento de cada uno de los protocolos que abordemos, por ahora tenlo presente (y recuerda su número de página) pues ¡volveremos!

Para hacernos una idea más clara sobre el porqué de los niveles, a continuación presentamos lo que el model OSI propone para cada uno de ellos. Este esquema presenta, como acabamos de ver, la división de los servicios y funciones en **siete niveles**. Esto no necesariamente se cumple, pues es sólo una propuesta de estandarización para poder acotar el diseño de los componentes tanto de Hardware como de Software. Una "Suite', Familia o Pila de protocolos que justamente se separa en algunos aspectos de este modelo es **TCP/IP**, la cual por ser un estándar DE FACTO, es hoy tenida en cuenta por la masa de las industrias de telecomunicaciones. Los niveles son:

-1.2.1. Nivel 1 (Físico):

- Recibe las tramas de nivel 2, las convierte en señales eléctricas u ópticas y las envía por el canal de comunicaciones.
- Define aspectos mecánicos, eléctricos u ópticos y procedimentales.
- Algunas de las especificaciones más comunes son: RS 232, V.24/V.28, X.21, X.25, SONET, etc.
- Funciones y servicios:
  - Activar/desactivar la conexión física.
  - Transmitir las unidades de datos.
  - Gestión de la capa física.
  - Identificación de puntos extremos (Punto a punto y multipunto).
  - Secuenciamiento de bit (Entregar los bits en el mismo orden que los recibe).
  - Control de fallos físicos del canal.

1.2.2. Nivel 2 (Enlace):

- Establece la **conexión con el nodo inmediatamente adyacente**.
- Proporciona los medios para asegurar la confiabilidad a la ristra de bits que recibió.
- Básicamente efectúa el control de flujo de la información.
- Funciones o servicios:
  - División de la conexión del enlace de datos (Divide un enlace de datos en varias conexiones físicas).
  - Control de flujo (Regula la velocidad a la cual la capa de enlace trabaja dinámicamente).
  - Proporciona parámetros de Calidad de Servicio (QoS), por ejemplo: Tiempo medio entre fallas, BER (Bit Error Rate), disponibilidad de servicio, retarde en el tránsito, etc.
  - Detección de errores (CRC {Control de Redundancia Cíclica} – Checksum).
  - Corrección de errores (ARQ {Allowed to ReQuest}, FEC {Forward Error Control}), sin eximir a capas superiores de hacerlo.
  - La IEEE lo subdivide en dos capas MAC (Medium Access Control) y LLC (Logical Link Control), si bien esto no es contemplado por OSI.

Algunas de las especificaciones más comunes son: LAP-B {X.25}, LAP-D {ISDN}, ISO 4335 del HDLC, I 122 del Frame Relay, también se puede tener en cuenta protocolos propietarios como ODI (Open Data Interface) y NDIS (Network Drivers Interface Standard).

1.2.3. Nivel 3 (Red):

La tarea fundamental de este nivel es la de **enrutado y conmutación de paquetes**. Es por esta razón que su trabajo acorde al tipo de conexión es muy variable. En una red de conmutación de paquetes puede ser implementado en detalle, en cambio al conmutar circuitos prácticamente no tiene sentido.

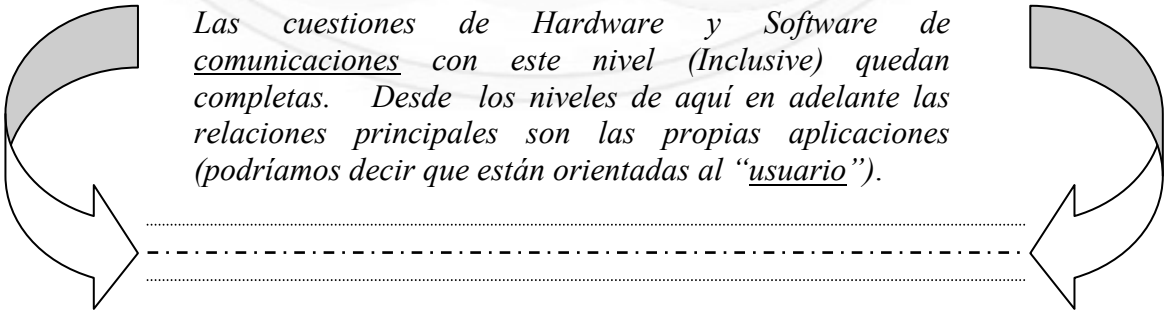
Sus funciones y servicios son:

- Encaminamiento y retransmisión (Define las rutas a seguir).
- Conmutación de paquetes.
- Multiplexación de conexiones de red.
- Establecimiento de circuitos virtuales.
- Direccionamiento de red.

1.2.4. Nivel 4 (Transporte):

- Su tarea fundamental es la **conexión de extremo a extremo** (end to end).
- Permite al usuario elegir entre distintas calidades de servicio.
- Optimiza la relación costo beneficio.
- Se definen cinco clases que van desde la cero (sin recuperación y eliminando paquetes dañados) hasta la cuatro (Detección y corrección de errores extendida).
- Funciones y servicios:
  - Correspondencia entre direcciones de transporte y de red.
  - Supervisión de red.
  - Facturación de extremo a extremo.

Algunos ejemplos de este nivel son: SPX, TCP, X. 224.



Las cuestiones de Hardware y Software de comunicaciones con este nivel (Inclusive) quedan completas. Desde los niveles de aquí en adelante las relaciones principales son las propias aplicaciones (podríamos decir que están orientadas al "usuario").

1.2.5. Nivel 5 (Sesión):

Permite el diálogo entre usuarios, entre dos ETD, se establece, usa, cierra una conexión llamada sesión.

Funciones y servicios:

- Establecimiento del diálogo Half Dúplex o Full Dúplex.
- Reseteado de sesión a un punto preestablecido.
- Establecimiento de puntos de control en el flujo de datos para comprobaciones intermedias y recuperación durante la transferencia de archivos .
- Abortos y rearranques.

Son algunos ejemplos de este nivel: Net BIOS Net BEUI, ISO 8327.

1.2.6. Nivel 6 (Presentación):

Asigna una sintaxis a los datos (Cómo se unen las palabras).

Funciones y servicios:

- Aceptación de datos de nivel siete (Enteros, caracteres, etc), negociando la sintaxis elegida (Ej: ASCII, EBCDIC,etc.).
- Transformación de datos para fines especiales (Ej: Compresión).
- Codificación de caracteres gráficos y funciones de control gráfico.
- Selección del tipo de terminal.
- Formatos de presentación.
- Cifrado.

1.2.7. Nivel 7 (Aplicación):

Sirve de ventana a los procesos de aplicación. Tiene en cuenta la semántica (significado) de los datos.

Funciones y servicios:

- Servicios de directorio (Transferencia de archivos).
- Manejo de correo electrónico.
- Terminal virtual.

- Procesamiento de transacciones.

Son algunos ejemplos de este nivel: X.400, X.500, SMTP, Telnet, FTP.

1.3. Conceptos de: Primitivas, servicios y funciones, SAP, UDP y UDS.

1.3.1. Ente: Elemento activo que ejerce funciones o proporciona servicios a sus niveles adyacentes.. El ente puede ser software (Ej: Compresión de datos) o hardware (Ej: Microprocesador para armado de paquetes).

1.3.2. SAP (Service Access Point): Punto situado en la interfaz entre dos capas. En dicho punto estarán disponibles los Servicios requeridos y las Respuestas. Indica explícitamente hacia que protocolo se debe dirigir por medio de esa interfaz. A través del SAP se puede multiplexar procesos, pues es el que indica hacia qué proceso se refiere un determinado encabezado (Header).

1.3.3. Primitivas: Los mensajes entre entes se llevan a cabo a través de cuatro primitivas:

- Solicitud.
- Respuesta.
- Confirmación.
- Indicación.

1.3.4. SDU (Service Data Unit): Datos que se mantienen inalterados entre capas pares y se van transmitiendo en forma transparente a través de la red.

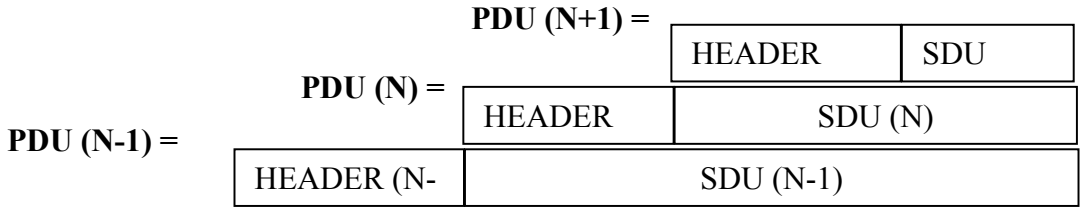
1.3.5. PDU (Protocol Data Unit): UDS más la información de control (encabezado) de ese nivel.

1.3.6. IDU (Interface Data Unit): Unidad de información que se transmite a través de cada SAP.

1.3.7. ICI (Information Control Interface): Información que el ente N+1 transfiere al ente N para coordinar su funcionamiento y queda en ese nivel (No pasa al siguiente).

Gráficos Resumen:

UDP = UDS + HEADER (encabezado).



En cada capa se **"Encapsula"** el PDU recibido de la capa superior y se agrega un Header (En la capa 2 también se agrega una cola).

1.4. Funciones y/o servicios.

Sin entrar en detalles específicos de diferencias entre servicios y/o funciones, en este punto se tratará de desarrollar cuáles son las tareas que se pretende realice un esquema de comunicaciones para poder transmitir información en forma transparente a un usuario. Una vez analizadas estas tareas, se dividirán en un enfoque de niveles que es el que propone OSI, entrando en el detalle de cuál de ellos desempeña cada una de las funciones y/o servicios.

1.4.1. Segmentación y reensamble:

Esta tarea trata de ajustar el tamaño de los datos a transferir al óptimo a colocar en el canal de comunicaciones. Este tamaño dependerá de varias causas:

- Determinados protocolos sólo aceptan un tamaño máximo o exacto de información (Ej ATM = 53 Bytes, Ethernet < 1518 Bytes, etc).
- Control de errores más eficiente.
- Equilibrado uso del canal (Evitar monopolios).
- Empleo de buffer más pequeños.
- DESVENTAJAS: Mayor información de control. Genera más interrupciones.

1.4.2. Encapsulamiento:

Se entiende por encapsulamiento al agregado de información de control a las unidades de datos y al tratamiento de ese bloque como un todo llamado UDP (Unidad de Datos del Protocolo), el cual es entregado al nivel inferior como una "Caja Negra" pues es totalmente transparente para el nivel inferior todo lo que existe allí adentro, tomándolo completo como una unidad de datos

para ese nivel. Esta información de control que se adiciona puede incluir alguno de los siguientes items:

- Dirección.
- Códigos de detección de errores.
- Información de control del protocolo.

1.4.3. Control de la conexión:

Esta tarea comprende todos los pasos necesarios para el establecimiento de la conexión, la transferencia de datos y el cierre de la conexión en los casos en que esta secuencia sea necesaria.

1.4.4. Entrega ordenada:

A medida que la información va descendiendo de nivel en el modelo, como así también cuando es colocada en el canal de comunicaciones y transferida a través del mismo, va sufriendo transformaciones y/o viaja por caminos diferentes. Acorde al nivel responsable de estas transformaciones, existirán tareas que se encargarán por distintas técnicas, de entregar al nivel superior las unidades de datos en la misma secuencia con que fue recibido en su nivel par en el ETD origen.

1.4.5. Control de flujo:

Esta actividad consiste en la posibilidad de regular la corriente de información a través del canal de comunicaciones. El control de flujo va desde la técnica más simple: "Parada y espera", hasta la de "Ventana deslizante", que permite tener hasta "n" tramas en el canal pendientes de ser entregadas o recibidas.

1.4.6. Control de errores:

El control de errores es la actividad que permite asegurar la confiabilidad de los datos en cada uno de los niveles pares de cada ETD. Como se tratará más adelante, el control de errores de un nivel, *no exime de ejecutar esta tarea a cualquier otro*, pues cada uno abarcará determinados

tramos dentro de la red, pudiendo ocurrir que el error no se produzca en el de su responsabilidad, ante lo cual no sería detectado, excepto que otra capa también lo esté haciendo. Para esta actividad se pueden emplear dos técnicas, FEC (Forward Error Control) y BEC (Backward Error Control).

1.4.7. Direccionamiento:

El concepto de direccionamiento es muy amplio, abarcando de alguna u otra forma, más de un nivel del modelo.

Si se hace una analogía con un envío postal, para un usuario final, la única dirección que le interesa, es la del domicilio postal al que desea enviar su carta. Detrás del mismo, existe todo un sistema diseñado y puesto en funcionamiento que permite que la carta que es depositada en un buzón, sea colocada en una determinada bolsa (y no otra) cuyo código de identificación sólo conocen los empleados de las sucursales de correo: esta bolsa se dirigirá hacia un avión, ferrocarril, camión etc... cuya identificación de vuelo, andén, etc... sólo conocerá el nivel de empleados del ferrocarril, aeropuerto o transporte automotor. Este proceso se puede desglosar hasta el mínimo detalle formando parte de un conjunto de direccionamiento absolutamente desconocido para un usuario final. No puede caber duda que quien diseñó el sistema de distribución de correo, conoce este detalle, y lo fue fraccionando por niveles de distribución para justamente lograr este efecto de transparencia.

Al referirse a un sistema de transferencia de datos ahora, es difícil luego de este ejemplo pensar que con una sola dirección el mismo funcionaría. Este planteamiento es el necesario para detallar todos los tipos de direccionamiento existentes, los cuales se pueden clasificar en cuatro categorías:

- Direccionamiento de nivel:

  Cada una de los distintos tipos de direcciones que se emplean en cada nivel, acorde al protocolo que se está empleando en ese nivel (Ej : X.25, Frame Relay, Ethernet, etc...).

- Espacio de direcciones:

  Se puede tratar como: Local (“Mi Red”) o Global (Todos los ETD a los que se puede tener acceso fuera de la Red Local).

- Identificador de conexión:

  A qué tipo de protocolo se está accediendo.

- Modo de direccionamiento:

  Se trata del tipo de destinatario del mensaje, este puede ser: Unicast – Multicast – Broadcast.

1.4.8. Multiplexado:

El concepto de multiplexado físico, a través de las distintas técnicas (TDM, PDM, FDM, etc) permite compartir un mismo canal físico por varios canales lógicos. Bajo este mismo concepto varias aplicaciones pueden estar ejecutándose durante una misma sesión (Ej: En una conexión a Internet, se puede estar consultando una página Web {HTTP}, enviando un correo {SMTP}, transfiriendo un archivo {FTP}, etc). Estos son ejemplos donde un mismo nivel permite operar con más de un nivel superior, entendiéndose como multiplexión lógica.

1.4.9. Servicios de transmisión:

Los distintos tipos de servicios de transmisión ofrecen las opciones de optimizar la relación costo/beneficio en el esquema de comunicaciones; por medio de éste se puede establecer las siguientes opciones:

- Prioridades (Se basa en que ciertos mensajes necesitan ser transmitidos con menor demora que otros, como pueden ser los de control o servicios de red).
- Grado de Servicio (Distintas opciones de calidad de Servicio {QoS}).
- Seguridad (Permite implementar estrategias de seguridad, en cuanto a la confiabilidad de datos, descarte de tramas, recuperación, fallas, etc...).

1.5. Presentación de la familia (pila) de protocolos TCP/IP.

En 1973, los investigadores Vinton Cerf de la Universidad UCLA y Robert Kahn del MIT, elaboran la primera especificación del protocolo de comunicaciones TCP. Y es en 1983 cuando se abandona el protocolo de comunicaciones anterior NPC y se sustituye por el actual protocolo TCP/IP.

En 1987 la red dedicada a las news USENET, se integra en Internet. Usenet fue creada por tres estudiantes de Duke y Carolina del Norte en 1979, Tom Truscott, Jim Ellis y Steve Bellovin. En cuanto al WWW (World Wide Web), todo empezó en 1980 en el CERN (Consejo Europeo para la investigación Nuclear), Suiza. El investigador Tim Berners-Lee implementó una aplicación que establecía enlaces entre una serie de nodos y permitía ir avanzando por ellos. Diez años más tarde formó un equipo junto con Robert Cailliau y realizaron el primer prototipo sobre una máquina NEXT. La conexión se realizaba haciendo TELNET a una máquina, ejecutando en esta última el navegador.

En 1993 Berners-Lee crea junto a Eric Bina en el NCSA el primer navegador gráfico Mosaic, y un año más tarde funda la compañía Netscape Communications.

Esta breve introducción histórica, es la que va dando origen a los primeros protocolos de esta familia, a los cuales se van sumando muchos más que permiten al día de hoy implementar todos los servicios que ofrece esta arquitectura.

1.6. Fuentes de información (RFC).

Como se mencionó anteriormente, la velocidad de avance de Internet, no soporta un burocrático sistema de estandarización como se venía haciendo con otras familias de protocolos, nace así la idea de las RFC (Request For Commentaries). Estas recomendaciones, no buscan estandarizar rigurosamente esta familia, sino que a medida que aparece una nueva funcionalidad, servicio, implementación, protocolo, etc... inmediatamente se puede describir la mejor forma de llevarla a cabo mediante una RFC, la cual tiene diferentes "Estados de madurez" y rápidamente sienta un precedente. En la actualidad superan holgadamente las tres mil.

El organismo responsable de las RFC es **IETF** (Internet Engineering Task Force); su página Web es: http://ietf.org, en ella encontrarás toda la información al respecto.

El listado de las RFCs puede verse en una "Wiki" dentro de esta Web (http://wiki.tools.ietf.org/rfc/index).

La RFC número 1, fue publicada en abril de 1969 y su título es "Host Software". Existen muchas anécdotas en su historia, a nosotros la que más nos gusta está dedicada a las personas que piensan que los informáticos (o la informática) es aburrida y se refiere al "Pez de abril". Esta fecha es análoga a lo que en España y muchos otros Países es el día de los Inocentes (28 de diciembre, por los santos inocentes). En este caso, la versión que parece ser más cierta es que en Francia, durante el reinado de Carlos IX, se seguía festejando el Año Nuevo el día 25 de marzo, pero en 1564 este Rey adoptó el calendario Gregoriano y comenzó a celebrarse esta fiesta el 01 de enero. Ante este hecho, parece ser que algunos franceses opuestos al cambio o tal vez algo despistados, continuaron enviando regalos y festejando la fecha antigua, cuya duración era de una semana, por lo tanto terminaba el primero de abril, esto derivó en broma por medio del envío de regalos ridículos o invitando a fiestas inexistentes y así nació la tradición del "1 de abril" y el nombre de Pez, viene por la constelación de "piscis", pues ese día el Sol abandonaba la misma.

Volviendo a nuestras RFC, el 1 de abril de 1973 la IETF publica la **RFC-527** "ARPAWOCKY" que es una especie de poesía totalmente en broma, a partir de ésta ya en 1989 IETF todos los años (con alguna excepción) publica una o dos de estas RFCs, así que ¡cuidado con las del 01 de abril!

1.7. Breve descripción de protocolos que sustentan a TCP/IP (PPP, ISDN, ADSL, Ethernet, X.25, Frame Relay y ATM).

Como se verá más adelante, el protocolo IP puede ser implementado sobre una gran cantidad de protocolos que le permitan transportar (llevar) la totalidad de la información que éste encapsula; es por esta razón que se trata aquí de dar una muy breve descripción de los más importantes de ellos.

1.7.1. PPP:

El Point to Point Protocol, es la implementación más simple del nivel de enlace para acceder a redes, por medio de líneas de telefonía conmutada, estableciendo como su nombre lo indica un enlace punto a punto con un nodo de acceso a la red, a través de un módem y por medio de los protocolos de la familia HDLC (High Level Data Link Connection), más específicamente LAP-M (Link Access Procedure - Modem).

1.7.2. ISDN (o RDSI):

ISDN es una tecnología de acceso en el rango de las telecomunicaciones y particularmente a los servicios de circuito virtual por conmutación de paquetes. ISDN pretende crear un sistema completo que permita abastecer cualquier servicio actual y futuro al usuario

Existen dos tipos de servicios ISDN: Básico o BRI (Basic Rate Interfaz) y PRI (Primary Rate Interfaz).

El BRI ofrece como servicio dos canales de 64 Kbps (Canal B: Bearer) y uno de 16 Kbps (Canal D: Delta) por eso es comúnmente llamado 2B + D, sumando un ancho de banda utilizable de 144 Kbps, si bien se debe tomar en cuenta que existen 48 Kbps empleados para separación de bandas y balanceo, que imponen un ancho de banda total de 192 Kbps siendo estos últimos transparentes y no utilizables para el usuario.

El cliente se encuentra representado por el CPE (Equipamiento del lado del Usuario), accediendo a una central telefónica llamada CO (Central Office), la cual es la encargada de la conmutación para lo cual emplea el sistema de señalización Nro 7 (SS 7) dentro de la red ISDN y el sistema de señalización DSS1 (Digital subscriber signalling) con el usuario por medio del canal D.

El PRI ofrece dos posibilidades, según la norma Europea se constituye con 30 B + D, posibilitando un ancho de banda disponible de 2,048 Mbps y según la norma de EEUU 23 B + D haciendo posible 1,544 Mbps. La unión de varios PRI puede hacerse bajo el esquema de ATM que de hecho constituye la base de B - ISDN (Broadband) o ISDN de banda ancha.

El ISDN mantiene las características de discado, es decir, se paga por su uso y en relación a las líneas dedicadas suele ser más económico hasta un máximo de 2 o 3 horas diarias de uso (que se corresponderían a unos 100 Mb diarios).

1.7.3. XDSL:

La DSL usa modernas técnicas de **procesamiento digital** para aprovechar la infraestructura de cobre instalada y crear lazos digitales remotos de alta velocidad en distancias de hasta 5.400 metros sin hacer conversiones de digital a analógico.

En un edificio grande o en un campus universitario, una DSLAM (DSL Access Multiplexer) se conecta a los cables telefónicos de cobre existentes que corren por las subidas del edificio hasta las computadoras de los usuarios.

Los PC del usuario se conectan a un **módem DSL** vía conexiones Ethernet estándar y el DSLAM, usado en lugar de conmutadores telefónicos de voz, transmite por sistema multiplex el tráfico de datos desde las líneas DSL a una interfaz ATM.

Esta transmisión de datos **punto a punto** en forma digital a elevado ancho de banda -hasta 7 Mbps o 8 Mbps- le da a la DSL una significativa ventaja sobre los sistemas ISDN y los módem de 56 Kbps. La transmisión analógica usa sólo una **pequeña porción** de la capacidad del alambre de cobre para transmitir información y por esta razón la velocidad máxima es de 56 Kbps. Aunque el ISDN es un buen sistema para transmisión a 64 Kbps - 2,048 Mbps- su tecnología no puede manejar las demandas de aplicaciones que requieren gran ancho de banda.

DSL crea conexiones más rápidas que ambos con **grandes canales de datos y mayores anchos de banda**. Estos grandes anchos de banda le permiten a la DSL manejar las demandas de aplicaciones que consumen mucho ancho de banda: videoconferencias en tiempo real, telemedicina y educación a distancia, por ejemplo.

Además del mayor ancho de banda, la DSL es en muchos aspectos una tecnología **más barata que la ISDN**.

Variantes de DSL

Las diferentes implementaciones de DSL sirven como canales de alta velocidad para conexiones remotas, pero tienen diferentes velocidades y características de operación.

- **ADSL (Asymmetric Digital Subscriber Line):** siendo esta variante la más flexible, dado que proporciona numerosas velocidades ascendentes y descendentes, se ha convertido hoy en día en la más popular para las empresas y los usuarios en el hogar.

  Velocidad máxima ascendente: 8 Mbit/segundo.

  Velocidad máxima descendente: 64 Mbit/segundo.

  Distancia máxima: 5.400 m.

- **HDSL (High bit-rate DSL):** Esta es la más vieja de las variantes de las tecnologías DSL. Se usa para transmisión digital de banda ancha dentro de instalaciones de empresas y compañías telefónicas que requieren dos cables entrelazados y que usan líneas T1.

  Velocidad ascendente máxima: velocidad de T1.

  Velocidad descendente máxima: velocidad de T1.

Distancia máxima: 3.600 m.

- **(ISDL) ISDN DSL:** Esta variante está más próxima a las velocidades de transferencia de datos de ISDN y puede ser activada en cualquier línea ISDN.

  Velocidad máxima ascendente: 128 kbits/s.

  Velocidad máxima descendente: 128 kbits/s.

  Distancia máxima: 5.400 m.

- **RADSL (Rate-Adaptive DSL):** Esta variante soporta software que automática y dinámicamente ajusta la velocidad a la cual pueden transmitirse las señales en la línea telefónica de determinado cliente.

  Velocidad máxima ascendente: 1 Mbit/s.

  Velocidad máxima descendente: 12 Mbit/s.

  Distancia máxima: 5.400 m.

- **SDSL (Single-Line DSL):** Esta variante es una modificación de HDSL.

  Velocidad máxima ascendente: 768 kbit/s

  Velocidad máxima descendente: 768 kbit/s.

  Distancia máxima: 3.000 m.

- **VDSL:** Es lo último en DSL y es una tecnología en desarrollo.

  Velocidad máxima ascendente: 2,3 Mbit/s.

  Velocidad máxima descendente: 52 Mbit/s.

  Distancia máxima: 1.350 m.

1.7.4. Ethernet: Se tratará en detalle más adelante.

1.7.5. X.25:

En 1974, el CCITT emitió el primer borrador de X.25. Este original sería actualizado cada cuatro años para dar lugar en 1985 al "Libro Rojo" ampliando e incorporando nuevas opciones y servicios, que posteriormente siguieron siendo ajustadas.

El concepto fundamental de X.25 es el de **Red de Conmutación de paquetes,** siendo la precursora de este tipo de tecnologías. Por nacer muy temprano, su desarrollo fue pensado sobre redes de telecomunicaciones de la década del 70, teniendo en cuenta nodos de conmutación electromecánicos o híbridos, líneas exclusivamente de cobre, equipos terminales de datos de muy poca "inteligencia" y baja velocidad de procesamiento. Basado en estos parámetros es que hace especial hincapié en la detección y control de errores, que como se puede esperar, se logra mediante una enorme redundancia en la transmisión. Para lograr este objetivo es que implementa un esquema de tres niveles asociados directamente a los equivalentes del modelo OSI.

En X.25 se definen los procedimientos que realizan el intercambio de datos entre los dispositivos de usuario y el nodo de ingreso a la red X.25 (no define lo que sucede dentro de la misma).

1.7.6. Frame Relay:

Es una de las técnicas de fast packet switching, llamada habitualmente conmutación de tramas. Es empleado fundamentalmente para el reemplazo de líneas punto a punto. Esta técnica opera sobre líneas de alta calidad, pues reduce sensiblemente la importancia que X.25 le da a la detección de errores, dejando esta tarea únicamente a los extremos. Por lo tanto, si las líneas son de baja calidad se deberá transmitir las tramas de extremo a extremo, bajando el rendimiento, incluso hasta ser peores que X.25 si el canal es muy malo.

Las estaciones terminales son responsables de:

- Cobertura de errores.
- Control de secuencia.
- Control de flujo.

Sus características son:

- Alta velocidad y baja latencia.
- Basado en circuitos virtuales de nivel 2.
- Se reemplaza el término canal lógico por DLCI (Data Link Connection Identifier).
- Este DLCI identifica al circuito virtual en cualquier punto de la red (Igual que el canal lógico en X.25).
- Cada DLCI tiene significado local.
- La conmutación se produce a nivel trama.
- Orientado al tráfico por ráfagas.
- Comparte puertos.
- Permite el uso dinámico de ancho de banda.

1.7.7. ATM:

Primera solución capaz de eliminar la barrera entre LAN y WAN. Aplica el concepto de conmutación rápida de paquetes (llamados celdas).

Es un concepto, no un servicio que emplea nuevas técnicas de conmutación con muy bajo retardo. Se emplea un mínimo de retardo en cada nodo, dejando el control de errores y de flujo en los extremos. Asume que los enlaces son digitales, por lo tanto posee un muy bajo índice de errores. Integra voz, datos y en algunos casos vídeo.

Emplea la transmisión en banda ancha (Broadband).

¿Por qué B – ISDN?

La conmutación de circuitos se adapta perfectamente a los servicios Isocrónicos (Sincrónico pero continuo en su retardo, Ej: Voz).

La conmutación de paquetes se adapta perfectamente a la transferencia de datos.

ATM es una solución de compromiso: Una conmutación de paquetes que puede asegurar una entrega rápida y continua de voz e imágenes.

El ATM Forum se crea porque las normas ITU salen con demoras de cantidad de años, y la dinámica de los avances tecnológicos no puede soportar tanto tiempo. El ATM Forum fue fundado en octubre de 1991, es un consorcio internacional formado para acelerar el uso de los productos y servicios ATM a través de una rápida convergencia y demostración de las especificaciones. No es un instituto de estandarización sino que trabaja en colaboración con instituciones como ITU y ANSI.

- Nace en los laboratorios Bell a fines de los 80'.
- La Unidad de transferencia de información es llamada celda la cual es de tamaño fijo (53 Byte) y son "relevadas" (Relay) entre cada nodo ATM por eso su concepto de Cell Relay.

  EEUU propone 64 Byte + 5 Byte (Necesita cancelar eco en TE).

  Europa propone 32 Byte + 4 Byte (Era baja la eficiencia de datos por celda).

  Se adopta : 64 + 32 = 96; 96 / 2 = **48 + 5** (Máx valor sin cancelar eco).
- Premisas de ATM: Red altamente confiable y de alta velocidad, nodos inteligentes para tratar errores.
- Reúne conceptos de conmutación de paquetes y de circuitos.
- Se le denomina asíncrono por la discontinuidad entre celdas a nivel de usuario, pero a nivel físico es estrictamente sincrónico pues lo soporta SDH (Jerarquía Digital Sincrónica).
- Es Orientado a la Conexión, técnica que logra mediante el empleo de circuitos virtuales (VPI y VCI).
- Tecnología capaz de conmutar millones de unidades por segundo a través de los nodos introduciendo retardos muy bajos, para lograrlo:

- Reduce las funciones en los nodos: Se le quita a éstos toda la carga de procesamiento que no sea estrictamente imprescindible para el encaminamiento exitoso de la llamada.
- Delega funciones en los extremos: Confiando en la inteligencia que se posee hoy en los equipos terminales de datos, confía en éstos muchas de las tareas.

1.8. Presentación de protocolos TCP, UDP e IP.

Dentro de esta pila de protocolos, como el modelo de referencia lo trata de mostrar, existen dos niveles bien marcados. Hasta el nivel cuatro (transporte) inclusive. "miran" hacia la red, por lo tanto todas las actividades que aquí se desarrollan tienen relación con el canal de comunicaciones y los nodos por los que pasa la información. Dentro de esta división, se encuentra el corazón de esta familia, se trata del protocolo IP de nivel 3 (red) y de los dos protocolos de nivel 4 (transporte) UDP y TCP. Sobre estos tres cae toda la responsabilidad de hacer llegar la información a través de la red, es por esta razón que se les trata de remarcar con esta presentación, y sub-clasificarlos de alguna forma respecto al resto.

1.9. Presentación de protocolos: FTP, Telnet, ARP y R_ARP, SMTP, POP3, IMAP, MIME, SNMP, HTTP, ICMP, IGMP, DNS, NetBIOS, SSL y TLS.

El resto de esta familia "miran" hacia el usuario. Se debe contemplar aquí las dos excepciones que son "ARP, R_ARP e ICMP-IGMP", que en realidad participan también en las tareas de red, pero como un complemento de la misma. Con estas excepciones salvadas, todo lo demás que se verá tiene como función principal cierta interacción con el usuario final para ofrecer servicios y/o funciones.

1.10. Protocolo IPv6.

Ante varios problemas que fueron apareciendo durante la larga vida del protocolo IP versión 4, desde hace varios años se está estudiando, y en la actualidad ya implementando en laboratorio, universidad, algunas empresas y troncales de Internet, una nueva versión del mismo llamada IP versión 6 (Ipv6) o IP Next Generation (IPNG), el cual ya ha llegado a un importante nivel de madurez, pero aún no se ha lanzado al mercado definitivamente.

EJERCICIOS DEL CAPÍTULO 1:

1. ¿Para qué sirve dividir en capas la comunicación entre dos ETD?
2. ¿Qué modelos de capas hemos presentado? ¿Por qué sus diferencias?
3. ¿Recuerdas cuáles son las capas del modelo TCP/IP?
4. ¿Cuál es el objetivo fundamental de cada un de las cuatro primeras capas?
5. De forma práctica, ¿a qué crees que se refiere el concepto Service Access Point?
6. ¿Cómo está formada la Unidad de Datos de Protocolo, por ejemplo del nivel 3?
7. ¿En qué consiste la segmentación y reensamble?
8. Si una capa superior hace control de errores, ¿puede que se haga también en alguna otra?
9. ¿Se te ocurre algún ejemplo de multiplexado físico que uses en la vida cotidiana?
10. Si tuvieras que profundizar en un tema respectivo a la pila TCP/IP, ¿cuál sería la fuente más exacta para obtener información?
11. Cita algunos ejemplos de protocolos que sustentan a TCP/IP.
12. La línea ADSL que casi todos tenemos en casa, ¿a qué tecnología responde? ¿Es la única forma de implementar esta tecnología?
13. ¿Hemos visto algún protocolo que aproxime velocidades WAN y LAN?
14. Si alguien se refiere a protocolos de nivel 2 y 3, ¿su tarea fundamental está orientada a servicios de usuario o de red?
15. ¿Qué versiones conoces del protocolo IP?

CAPÍTULO 2: Principios del análisis de la Información.

En principio, independientemente del nivel que estemos analizando o evaluando, toda secuencia de unos y ceros se interpreta como "información"; la misma puede estar relacionada a datos, encabezados, control, etc… pero siempre la consideraremos "información".

Este flujo de información que se está intercambiando entre dos ETD se denomina "Tráfico" y es lo que se presenta a continuación.

2.1. Tráfico: Broadcast, multicast y dirigido:

El tráfico que se produce en una red se puede clasificar desde dos puntos de vista: por su sentido y por su forma de direccionamiento.

2.1.1. Por su sentido:

La dirección en que las señales fluyen en la línea de transmisión es un factor clave que afecta a todos los sistemas de comunicaciones de datos. Existen tres tipos de flujo de la información:

- Simplex:

  La transmisión entre dos equipos se realiza en un único sentido (por ejemplo la TV).
- Half-Dúplex:

  La transmisión se realiza en los dos sentidos, aunque no simultáneamente (por ejemplo los walkie talkies).
- Dúplex:

  Transmisión simultánea e independiente en ambos sentidos (por ejemplo el teléfono).

2.1.2. Por forma de direccionamiento:

- Unicast:

  Se trata de una transmisión de un ETD a sólo un ETD.
- Multicast:

  Se trata de una transmisión de un ETD hacia un determinado grupo.
- Broadcast:

Es el tipo de transmisión de un ETD hacia absolutamente todos los ETD que escuchen la misma.

2.2. ¿Cómo se analiza el tráfico?

El análisis de tráfico consiste en "desarmar" cada trama, paquete, segmento, bloque de información (más adelante veremos que cada uno de los conceptos anteriores tiene un significado diferente) y analizarlos "bit" a "bit". Puede parecernos horroroso y/o inhumano, pero aquí se esconde la razón de ser de la seguridad. Gracias a Dios en la actualidad contamos con muy buenas herramientas que hacen más amigable esta dura tarea.

Cuando un dispositivo de red comienza a recibir información (más adelante iremos de lleno a este tema) cada uno de los niveles de esta pila TCP/IP comienza su tarea identificando "bit" a "bit" a qué módulo le corresponde trabajar. Un módulo no es más que un código, script o programa que tiene todas las órdenes que debe realizar en ese nivel y con esa secuencia de bits. Una vez que reúne suficiente información para identificar unívocamente a qué módulo llamar, automáticamente le pasa el control a éste y a partir de allí comienza su tarea. Concretamente cada módulo es lo que puede o no puede hacer un determinado "protocolo" de comunicaciones, es decir el conjunto de reglas que impone ese módulo o "protocolo" para ese nivel del modelo de capas. Sería de imaginar que ya te suene al menos ideas como "Ethernet", "IP", "http", etc... pues cada uno de esos conceptos son justamente "protocolos" de comunicaciones que operan a un nivel específico del modelo de capas y sus funciones y/o servicios son gobernados por este programita que estamos llamando "módulo", y bajo esta idea es que encontraremos en cualquier ordenador conectado a una red "TCP/IP" muchos módulos y cada uno de ellos se encargará de procesar esta secuencia de información que se le entrega cuando es llamado.

El primer módulo que podríamos considerar en este texto es la librería "libpcap", que como su nombre parece dejar entrever, es la librería encargada de las "capturas de tráfico". Esta librería tiene una característica muy importante si nuestra conexión a la red lo permite, que es la de poder escuchar en modo "promiscuo". Como iremos viendo a lo largo del libro, cuando una información circula por la red e ingresa a un ETD, a medida que cada nivel la va evaluando, decide si se dirige hacia él o no (en cada nivel), cuando no es para él entonces debe descartar esa información y/o en algunos casos reenviarla. Cuando se logra operar en modo "promiscuo" esto implica que no descarte información, sino que procese todo, sea para este ETD o para cualquier otro. Por esta razón la idea de analizar tráfico de una red, está particularmente dirigida a poder escuchar TODO el tráfico que circula por ella.

Basado en esta librería "libpcap" se encuentran dos comandos que desde muy remoto permiten justamente "escuchar" o "esnifar" esta totalidad del tráfico; ellos son "**tcpdump**" y más adelante "**tethereal**". Estos fueron los verdaderos pioneros del análisis de tráfico y más adelante nos cansaremos de recurrir sobre todo a "tcpdump".

Al principio, la captura de tráfico debo reconocer que era lisa a llanamente INHUMANA, la información se debía interpretar en casi la totalidad de los casos en forma hexadecimal y a lo sumo

en formato ASCII, hasta que aparecieron los analizadores de protocolos con sus interfaces gráficas y su humanización del análisis binario.

En sus inicios se les llamó "**Sniffers**", y existía una notable diferencia entre estos y los analizadores de protocolos, pues los primeros sólo se dedicaban a capturar el tráfico de la red y representarlo en su forma hexadecimal, sin desempeñar ninguna de las tareas que hoy realiza un analizador de protocolos, un ejemplo aún vigente sigue siendo el mencionado comando "tcpdump" de Linux. En la actualidad, muchos de estos sniffers fueron incorporando más y más funcionalidades, pues una vez que está capturada la información, es muy simple realizar estadísticas, comparaciones, presentaciones gráficas de la misma, etc. Por lo tanto hoy, es muy confusa la denominación que se emplea para estos productos, pero siendo estrictos conceptualmente, un **sniffer** sólo captura tráfico y lo presenta de manera más o menos amigable (y nada más). **Un analizador de protocolos**, realiza esta tarea y a su vez procesa esta información para obtener todas las posibles necesidades de usuario con la misma.

2.3. ¿Qué es un analizador de protocolos?

Un analizador de protocolos captura conversaciones entre dos o más sistemas o dispositivos. No solamente captura el tráfico, sino que también lo analiza, decodifica e interpreta, brindando una representación de su escucha en lenguaje entendible por medio de la cual, se obtiene la información necesaria para el análisis de una red y las estadísticas que el analizador nos proporciona.

Esencialmente, un analizador de protocolos es una herramienta que provee información acerca del flujo de datos sobre una LAN, mostrando exactamente qué es lo que está sucediendo en ella, detectando anomalías, problemas o simplemente tráfico innecesario. Una vez que un problema es aislado, se pueden analizar las causas que lo producen y tomar las medidas para evitarlo.

Un analizador de protocolos debería proporcionar tres tipos de información sobre una LAN:

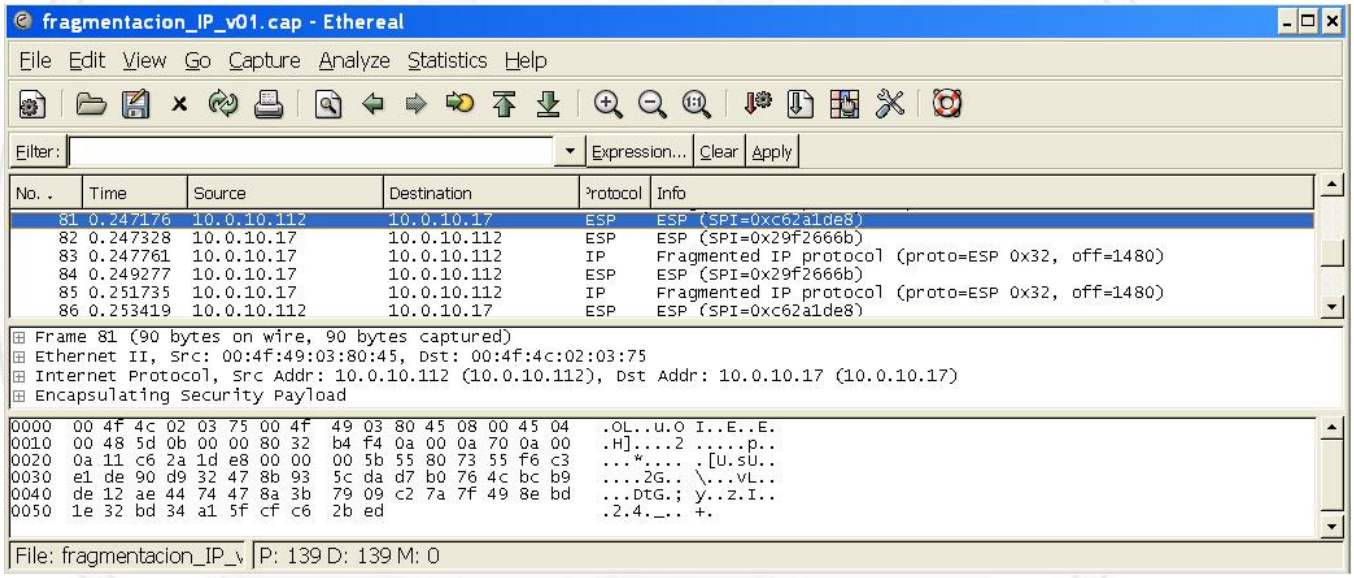
- **Estadísticas** sobre tráfico de datos, estado de los dispositivos y líneas de errores en la LAN. Esta información ayuda a identificar tramas y condiciones generales que pueden señalar un problema inesperado o causar un bajo rendimiento en la red. Permite también determinar nuevas necesidades de Hardware para segmentar o crear subredes dentro de la LAN como podría ser el empleo de Switch o router y la ubicación y configuración correcta de los mismos.
- C**aptura de paquetes y decodificación** de los mismos en los distintos protocolos de cada nivel. Debería permitir también el filtrado correspondiente, que posibilite especificar en el mayor grado de detalle lo que se desea estudiar, dejando de lado la información innecesaria. Se suele filtrar por Dirección MAC, IP, Nombre NetBIOS, puertos, tipo de protocolo, secuencias de bit, etc.
- **Representación de información histórica** en lapsos diarios, semanales, mensuales o en períodos establecidos por el usuario. Esta información provee una perspectiva histórica para cualquier nuevo problema o indica un problema potencial antes que este suceda.

Las estadísticas de estaciones de trabajo o servidores permiten identificar el tráfico generado por cada uno de ellos y el porcentaje de ancho de banda que consumen. Con esta información se puede

determinar cuál es la que hace mayor uso del canal físico y cuáles son los recursos más usados. Por ejemplo si una estación genera un alto porcentaje de tráfico, esto puede estar indicando un fallo en su tarjeta de red, permitiendo tomar las medidas correspondientes, basadas en observaciones reales de la red, y no por prueba y error.

Un concepto importante es que un analizador de protocolos no emplea el protocolo SNMP (Single Network Monitor Protocol, que veremos más adelante); esta herramienta cuenta con la información específica que le permite identificar las diferentes secuencias de bit, y por medio de los encabezados establecidos para cada protocolo, los que responden a estos patrones los asocia a uno de ellos y lo entrega a su módulo, el cual "desarma" las cadenas binarias en la información contenida en ellas. Por ejemplo SNMP no podría brindar información, sobre sesiones Telnet, TCP/IP, uso de ancho de banda, qué tipos de paquetes se emplean, etc. Un SNMP se debe considerar como un muy buen COMPLEMENTO de un analizador de protocolos.

Durante todo este libro haremos uso del analizador de protocolos "Wireshark" o "Ethereal" (a decir verdad aún no sé por qué le han cambiado este nombre histórico), que es de libre distribución y considero el más completo del mercado.



2.4. Detección de sniffers.

Las técnicas de detección de sniffers que se emplean son varias y todas se basan en poder determinar si la interfaz de red se encuentra en modo promiscuo, lo cual es un claro síntoma de que desea recibir todo el tráfico que pasa por ella, actividad no necesaria para ningún host que preste servicios en una red:

a. La más simple de estas es enviar un mensaje ARP a una dirección MAC falsa, si responde, es que se encuentra en modo promiscuo. La masa de los sniffers o analizadores de protocolos ya prevén esta técnica y simplemente anulan todo tipo de respuesta a nivel MAC.

b. Test específico del Sistema Operativo: Es muy similar al anterior, pero se envían mensajes ICMP de eco (Tipo 8), con la dirección MAC no existente en la red, se emplean también con mensajes que pueden ser Unicast, Multicast o Broadcast, y basado en el tipo de respuesta se determinará también qué sistema operativo posee el host (Linux: responde ante unicast [este es un bug que la mayoría de los sistemas Linux hoy tienen resuelto, pero existe un error en la pila TCP/IP de este S.O. con el cual responderán siempre a una dirección IP real, aunque la MAC sea falsa], BSD: responde ante multicast, NT: Lo hace analizando sólo el primer octeto MAC contra la dirección IP cuyo primer octeto sea Broadcast, independientemente del resto de la dirección MAC.

c. Test DNS: Esta técnica envía información acerca de una dirección IP y escucha por cualquier solicitud de resolución DNS desde un host hacia el servidor correspondiente.

d. Test de latencia del sistema: Este es el más complejo pero también el más eficiente. Se trata de enviar paquetes ICMP y medir los tiempos de respuesta. Si se incrementa el tráfico en la red, una interfaz en modo promiscuo irá tardando cada vez más tiempo en responder que el resto de las interfaces, pues ésta deberá procesar la totalidad de las tramas, mientras que el resto sólo lo hará con las tramas dirigidas a estas.

2.5. Introducción al Ethereal (o Wireshark) (Como herramienta de análisis y captura).

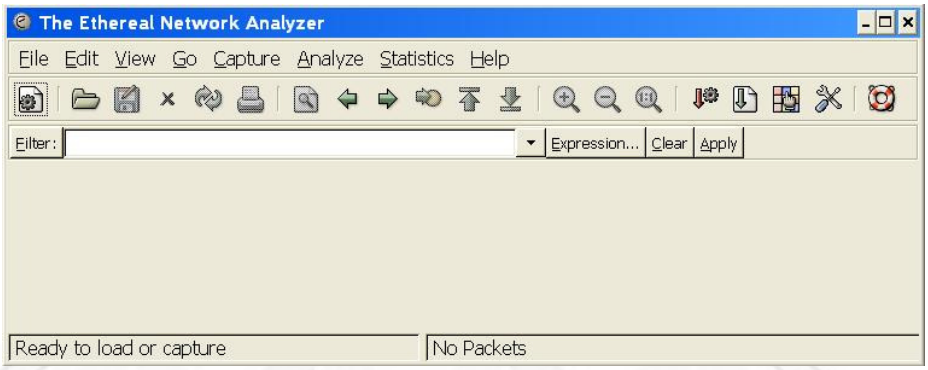
En virtud de ser una herramienta fundamental para todo el trabajo que desarrollaremos en el libro, en este apartado se hace una presentación inicial con los conceptos básicos que es necesario tener para poder comenzar a emplearla.

De aquí en adelante lo llamaremos directamente "Ethereal" pero se da por entendido que nos estamos refiriendo a ambos. Como ya se mencionó con anterioridad, un analizador de protocolos es una herramienta que permite capturar, filtrar y analizar el tráfico de una red. Los datos capturados pueden ser guardados para un análisis posterior o analizados inmediatamente después de la captura. Esta herramienta puede ser una combinación de hardware y software (como por ejemplo el Internet Advisor de HP), o simplemente software como es el caso de Ethereal, Iris, Network Monitor de Microsoft. Ethereal permite lo siguiente:

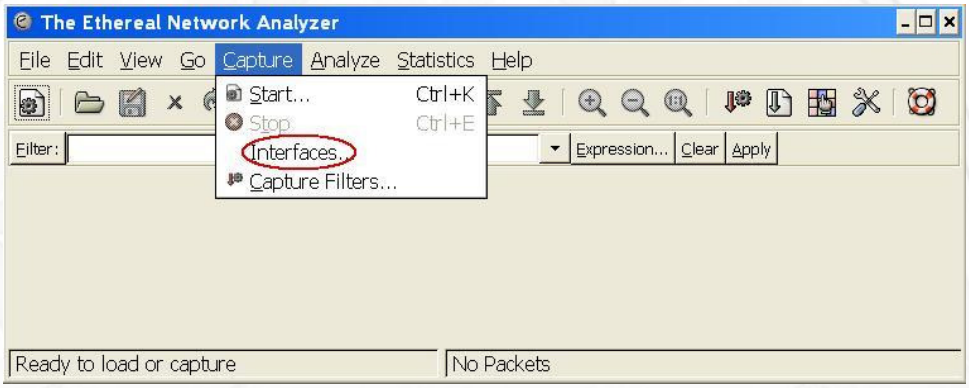
- Capturar tramas directamente desde la red.
- Mostrar y filtrar las tramas capturadas.
- Editar las tramas y transmitirlas por la red.
- Capturar tramas desde un ordenador remoto.
- Realizar análisis y estadísticas.

Es muy importante el concepto de MODO PROMISCUO, cuyo significado es que el adaptador de red permite el ingreso ("escucha") de absolutamente todas las tramas que pasan por el cable. Se debe tener en cuenta que un adaptador que trabaja en modo promiscuo significa que delega todo el trabajo en la CPU por lo tanto representa una sobrecarga de tareas al ETD que se le instala, no siendo así en el que opera NO en modo promiscuo, que posee mecanismos de filtrado que liberan de las actividades de nivel 2 al ETD.

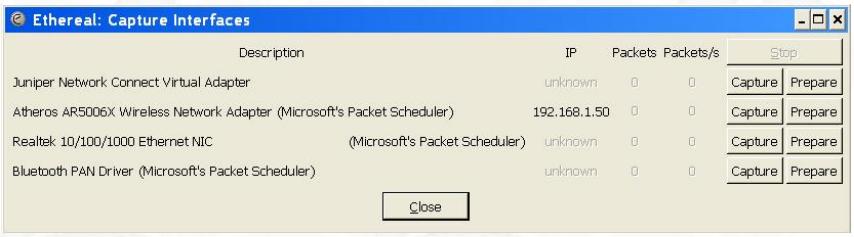
Al abrir Ethereal nos presenta una imagen como la que vemos a continuación.



La primera actividad que debemos hacer es seleccionar y configurar la interfaz que emplearemos.



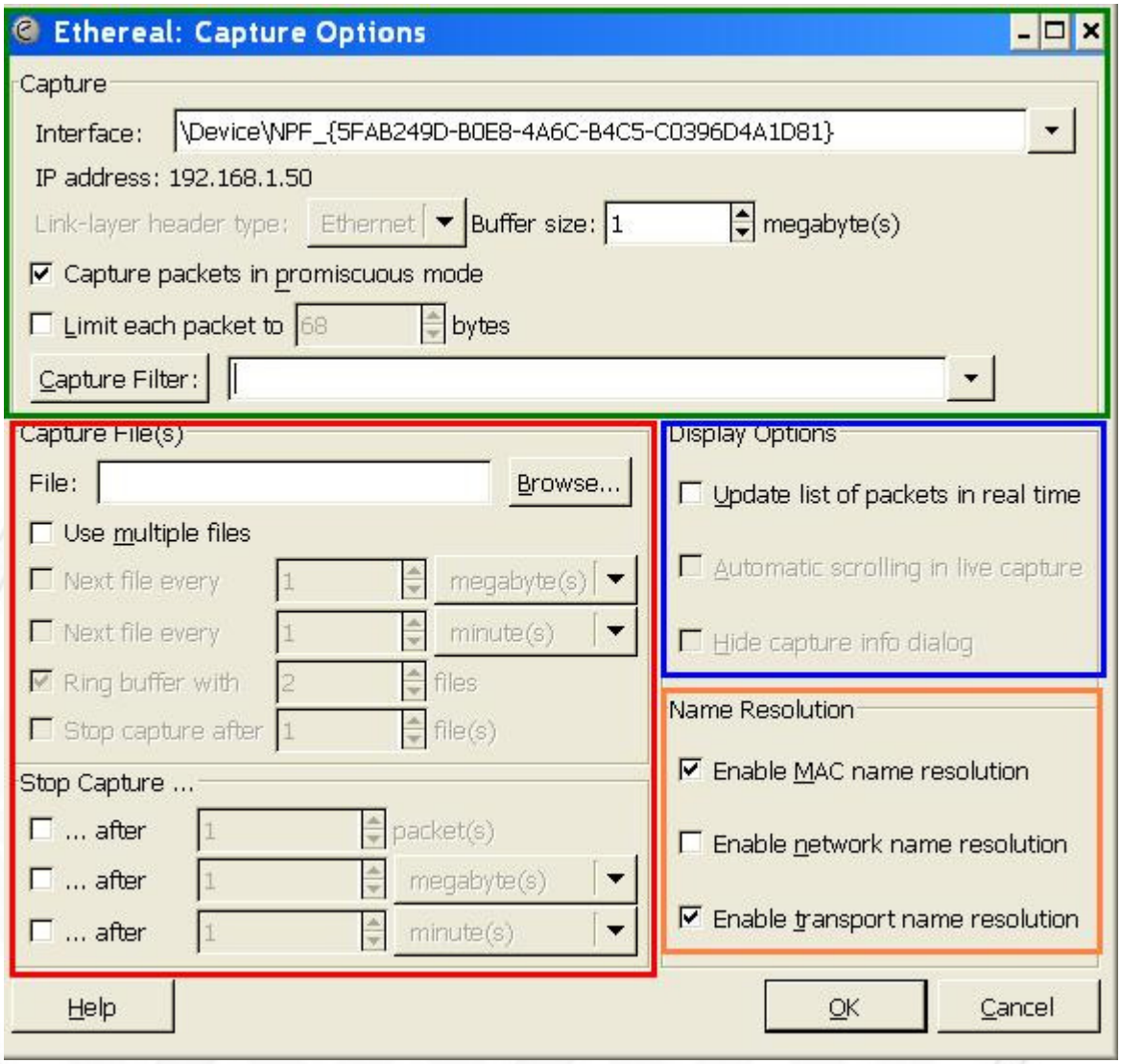
Al seleccionarla nos presenta la siguiente ventana.



Como se puede apreciar, en mi ordenador existen cuatro potenciales interfaces a emplear, y en estos momentos sólo tiene dirección IP la "Realtek 10/100/1000 Ethernet". No es necesario que una interfaz tenga dirección IP para que escuche, pero suele ser necesario para capturar tráfico que deseo generar a la red y verificar sus respuestas, por esta razón, seleccionaremos la Interfaz Ethernet.

NOTA: Un detalle que me ha sucedido es que para las redes WiFi, el sistema Operativo Windows, no me permite colocar la interfaz de red (por ejemplo: en la imagen anterior la que figura como "Atheros AR5006X Wireless") en modo promiscuo, pero sí lo hace con Linux (en mi caso con Debian), no sé si es algo propio de mi portátil y este interfaz de WiFi o es un tema de Microsoft, pero os dejo la inquietud.

Nuestro siguiente paso debería ser seleccionar la interfaz que emplearemos (haciendo "Clic" en "Prepare" y/o "Interface") y se nos presentará la ventana siguiente:



Como podéis apreciar se nos presentan cuatro ventanas (que hemos marcado con los colores: verde, rojo, azul y naranja).

Ventana verde: Arriba de todo nos figura la interfaz que hemos seleccionado, su dirección IP y más abajo "Buffer Size"; este parámetro es muy importante, pues es el límite de información que podremos capturar, una vez llegado allí se detendrá la captura. También podemos ver que por defecto nos aparece seleccionado el modo "promiscuo" ya mencionado, esta opción puede ser desactivada cuando sencillamente queramos analizar tráfico desde y hacia nuestro ordenador. También nos ofrece la alternativa de limitar el tamaño de los paquetes capturados, esta opción puede ser muy útil si ya se sabe la información que se desea capturar, por ejemplo un patrón de tráfico que aparece dentro de una URL (Ej: http://www.midominio.es /search?source=root) en este caso sabemos que esa cadena no aparecerá jamás por arriba de los 80 bytes (más adelante lo veremos en detalle) y por lo tanto no sería necesario capturar el resto de la información pues esto nos llenaría la memoria de esa captura con mayor rapidez e innecesariamente. Por último nos ofrece la posibilidad de "Filtrar la captura", este campo es fundamental. Ethereal presenta dos tipos de filtro:

- Filtro de Captura: Selecciona qué tramas captura y cuáles no. El formato de este filtro es el mismo que el Comando "tcpdump", y lo desarrollaremos más adelante.
- Filtro de visualización: Una vez que se ha capturado tráfico y en otra ventana, nos presentará toda la información capturada la cual puede contener grandes volúmenes de datos. Con este filtro, se permite seleccionar qué deseo ver, ocultando el resto. El formato de este filtro difiere bastante del anterior, es propio de Ethereal y contiene una interfaz gráfica muy amigable para su empleo.

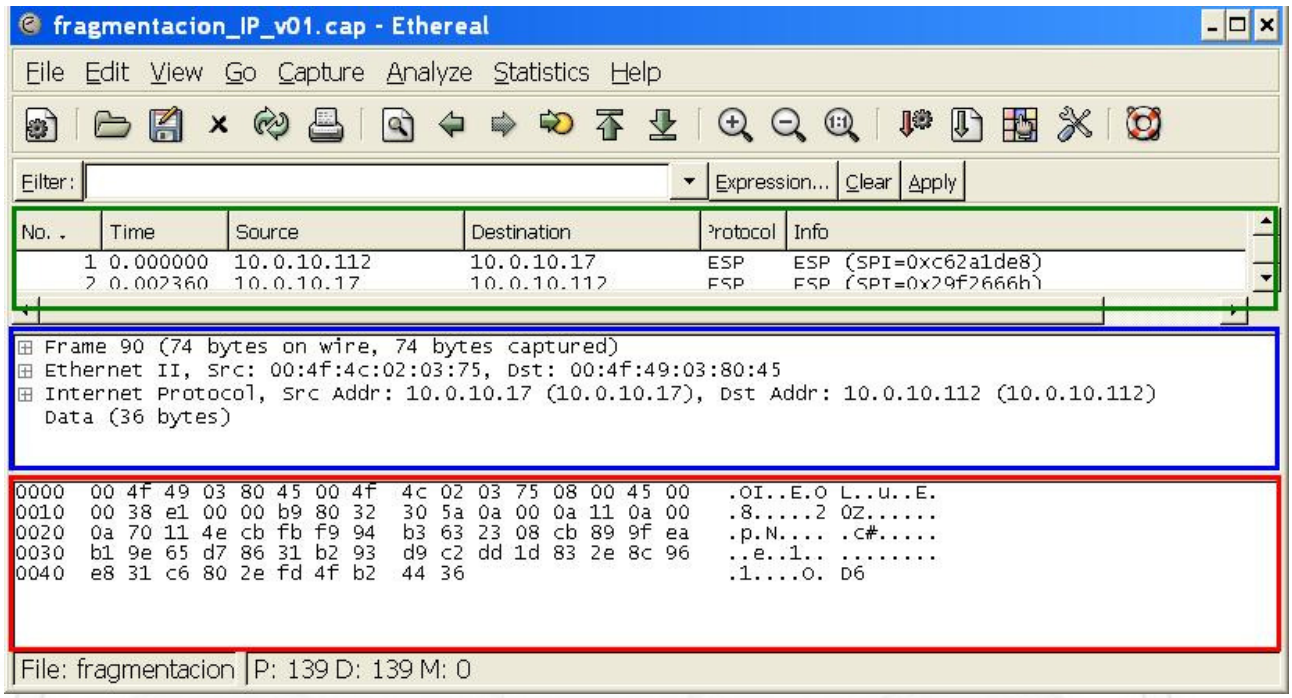
Ventana Roja: Por ahora no es necesario que la desarrollemos.

Ventana azul: Como su título lo indica, nos ofrece diferentes opciones de visualización. Si seleccionamos la primera, las tramas se nos presentarán a medida que se está capturando. Es muy útil para capturas breves y en local, pero si estoy capturando tráfico desde un servidor remoto y todo este volumen de datos nos está llegando por la red, es muy peligroso pues satura el ancho de banda, así que tened cuidado con su empleo. Las dos opciones siguientes son de menor importancia, y nos permiten que la ventana haga "scrolling" (no sé como traducirlo) mientras captura, es decir nos aparezca la barra de desplazamiento vertical a la derecha y encole tramas, y la última opción, es sencillamente ver o no una ventana que por medio de barras y números nos llevará en tiempo real los resúmenes de cada "protocolo" que captura (probadla).

Ventana Naranja: Nos permite la resolución de esos "esquemas de direccionamiento", es decir si seleccionamos cualquiera de ellos, en vez de su dirección numérica, Ethereal tiene las tablas con todos los códigos para resolver los nombres de los fabricantes de las Tarjetas de nivel Enlace (MAC), puede resolver los nombres de red y también los DNS, (todo esto será tratado cuando abordemos esos niveles).

Por de pronto esta ventana no nos será de gran ayuda, pero creímos conveniente al menos hacer unas breves referencias de su uso. Para seguir adelante, sólo verificad que esté seleccionada la interfaz de red sobre la que queríamos capturar y presionad "OK".

Se nos presentará por fin la ventana de captura (y ya debería estar capturando si estamos conectados a una red), con algo similar a lo que se presenta a continuación:



En ella también hemos remarcado tres ventanas con colores:

Ventana Verde (Superior): nos presenta toda la secuencia de tramas capturadas en el orden de captura. En la primera columna se ve el número correspondiente a esa trama, en la segunda y tercera la dirección origen y destino (puede ser IP o MAC), en la cuarta el protocolo de máximo nivel que alcanza esa trama y la última columna un breve resumen de esa trama. Haciendo "Clic" sobre el encabezado de cualquiera de ellas, se ordenará por ese campo.

Ventana Azul (Central): Nos presenta en formato "Humano" cada uno de los niveles del modelo de capas (pero al revés) y nos permite desplegar ("+" o "-" y en Wireshark con flechas) hasta el máximo detalle todos sus campos.

Ventana Roja (Inferior): En esta se ve en filas de 16 pares de números hexadecimales toda la trama y a la derecha la misma información pero con el carácter ASCII que se corresponde a cada par hexadecimal. Esa ventana es de suma importancia cuando analizamos el contenido de una trama si el mismo no está cifrado.

Por ahora, para nuestra presentación inicial de la herramienta, creemos que es suficiente. Si deseas puedes dedicarle un tiempo para ir familiarizándote con ella pues será de uso principal a lo largo de todo el texto. Para detener y/o volver a capturar, sencillamente desde el menú superior seleccionas "Captura" y allí tienes "Start" y "Stop", también puedes hacerlo desde unos iconos que figuran en la barra de menú gráfico (debajo).

NOTA: Somos conscientes que hemos iniciado de forma "brusca" presentando esta herramienta, la cual no tienes por qué comprender ahora, ni tampoco valorar toda su potencialidad. También hemos hecho mención a protocolos, direcciones y términos que pueden aún resultarte extraños, pero sinceramente estamos convencidos que es necesario que empieces a tenerla en cuenta desde YA. No te asustes, tennos un poco de paciencia y verás que te ¡enamorarás de Wireshark!

Captura, filtrado y análisis de tramas.

Cerrando un poco más este tema, el análisis de datos comienza con la vista de los datos capturados, esta pantalla muestra la totalidad de las tramas que ingresaron a nuestra tarjeta, las

cuales pueden ser filtradas con anterioridad a la captura o luego de ella (filtros de captura y de visualización que comentamos) para seleccionar las que se desee analizar.

La presentación de captura entonces se divide en tres partes:

a. Panel resumen: Muestra la totalidad de las tramas presentando en columnas la siguiente información:

   1) Trama: Número de trama capturada, en el orden que fue capturada.

   2) Tiempo: Permite identificar el tiempo en el que inició la captura de esta trama o puede ser configurado para identificar la hora del día en que fue capturada.

   3) Dirección MAC o IP origen: Muestra la dirección de hardware o software del ETD que emitió la trama.

   4) Dirección MAC o IP destino: Muestra la dirección de hardware o software del ETD que recibió la trama.

   5) Protocolo: El protocolo usado para transmitir la trama.

   6) Descripción: Resumen del contenido de la trama.

b. Panel de detalle: Muestra todo el grado de detalle de la trama seleccionada en el panel anterior, desplegando la totalidad de los protocolos incluidos en esa trama.

c. Panel hexadecimal: Muestra en formato hexadecimal la totalidad de los bytes que fueron capturados en la trama seleccionada en el panel resumen.

2.6. Captura, filtrado y análisis de tramas.

Como ya se mencionó, la gran diferencia entre un sniffer y un analizador de protocolos, pasa por los servicios que este último ofrece, los cuales en general van orientados hacia una mejor interpretación de la información capturada.

Para poder mejorar la visualización de la información, es muy importante "Pulir el bosque", es decir, **poder filtrar** lo que no se desea para clarificar la información que se está buscando. Todos los analizadores de protocolos poseen los dos filtros mencionados:

- **Filtro de captura:** Permite seleccionar qué es lo que se desea ingresar a la memoria de la herramienta y qué no. Esta funcionalidad es de suma importancia, pues en redes de alto tráfico, es muy fácil que se desborde la memoria del PC donde se ejecuta el analizador de protocolos, y en el caso de desear capturar únicamente una determinada dirección, protocolo, puerto, etc… ¿de qué sirve almacenar el resto del tráfico? Este filtro permite registrar (Capturar) sólo lo que se desea, descartando el resto de la información que viaja por el cable. Recordad que los comandos de este filtro son los mismos que "tcpdump".
- **Filtro de visualización:** En este caso, se trata de presentar una "mejor vista" de lo que ya ha sido capturado. Este filtro se emplea, una vez que se detuvo la captura, para poder elegir qué se desea visualizar dentro de toda la información que ya se encuentra en memoria.

2.7. Presentación hexadecimal, binaria y decimal.

No nos odiéis!!! Os garantizamos que esto es algo así como esas materias que no nos gustan pero debemos superar, casi como en "desafío extremo" lo de ¡prueba superada! Pero es vital, si vamos a trabajar de forma eficiente y seria, Sí o Sí debemos familiarizarnos con poder interpretar la información cuando se nos presenta en forma binaria, decimal, hexadecimal, y como sabemos que el libro recién empieza aún debes estar descansado; por esa razón nos atrevimos a desarrollar este tema aquí.

2.7.1. Bit: estado lógico equivalente a 1 o 0.

2.7.2. Byte u Octeto: Agrupación de 8 bit.

Esta definición es la que realmente se universalizó para el tratamiento de la información y la palabra octeto es hoy una de las bases de la transmisión de información, la razón de ser de esta convención radica en:

- La capacidad suficiente de codificación que posee un octeto, es decir 256 posibilidades diferentes.

  Si se plantea el conjunto de posibilidades este irá desde: 0000 0000, 0000 0001 , 0000 0010 , 0000 0011 , 0000 01000.....1111 1111.

  Ante lo cual permite hasta 256 códigos diferentes.

  Es fácil el pasaje entre el sistema decimal, hexadecimal y binario.

Suma Decimal	128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 256								
Peso Decimal	128	64	32	16	8	4	2	1	
Binario	B	b	B	b	b	B	b	b	
Peso hexadecimal	8	4	2	1	8	4	2	1	
Suma hexadecimal	8 + 4 + 2 + 1 = F				8 + 4 + 2 + 1 = F				FF
EJEMPLO									
Suma Decimal	128 + 32 + 2 + 1 = 163								
Peso Decimal	128	0	32	0	0	0	2	1	
Binario	1	0	1	0	0	0	1	1	
Peso hexadecimal	8	0	2	0	0	0	2	1	
Suma hexadecimal	8 + 2 = A				2 + 1 = 3				A3

Hexadecimal: Conjunto de16 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

2.7.3. Caracter: Es la unidad de información a nivel alfabeto humano, representa cualquier símbolo del alfabeto usado como alfabeto normal. Se los clasificará en:

a. **Alfabéticos**: Letras en mayúsculas y minúsculas.
b. **Numéricos**: Dígitos de 0 a 9.
c. **Especiales**: Puntuación, paréntesis, operaciones aritméticas y lógicas, comerciales, etc.
d. **De operación y control**: Destinados al control de la transmisión de información (Retorno de carro, nulo, SYN, ACK, DLE, EOT, SOH, etc)

2.7.4. Bloque, Mensaje, Paquete, Trama: Son distintas formas de agrupamiento de Bytes, y se definen acorde a las distintas técnicas de transmisión de información o Protocolos de Comunicaciones.

EJERCICIOS DEL CAPÍTULO 2

1. ¿Desde qué puntos de vista clasificarías los tipos de tráfico?
2. ¿Qué ejemplos prácticos conoces del tráfico: Simplex, semidúplex y dúplex?
3. Un mensaje que va dirigido sólo a un grupo de usuarios, ¿qué forma de direccionamiento emplea?
4. ¿Qué implica hacer análisis de tráfico?
5. ¿Qué es un protocolo de comunicaciones?
6. ¿Qué es un módulo?
7. ¿Existe alguna diferencia entre un sniffer y un analizador de protocolos?
8. ¿Qué tipos de información debería proporcionar un analizador de protocolos?
9. ¿Es posible detectar si alguien está usando un sniffer en nuestra red LAN?, ¿Cómo lo haría?
10. ¿Qué permite hacer un analizador de protocolos como Ethereal?
11. ¿Qué es el modo promiscuo? ¿Has encontrado alguna relación entre lo que te presenta Ethereal y el modelo de capas que estamos presentando?
12. ¿Cómo me explicarías que un octeto son 256 posibilidades?
13. Pasar a binario los siguientes números decimales: 32, 127, 41, 79 y 376. ¿Cuántos bits necesito para representar cada uno de ellos?
14. Pasar a decimal los siguientes números binarios: 111, 1011, 100100, 10101011.
15. Pasar a binario y a decimal los siguientes números hexadecimales: FF, A1, 02, 17.
16. ¿Qué tipos de caracteres conoces?.....(no vale poner: mal humorado, alegre, cabr.... etc..)

Herramienta ETHEREAL (o Wireshark)

1. Prueba hacer capturas con diferentes interfaces de red (Ej: Ethernet, WiFi, Bluetooth).
2. Ajusta el buffer de captura lo más pequeño que se pueda, ¿qué valor resultó?
3. Selecciona la captura en modo promiscuo y luego deselecciónala y realiza más capturas, ¿qué diferencias encuentras?
4. Ajusta al máximo el límite de Bytes de captura de paquetes, ¿hasta qué valor has llegado?
5. Ejercita diferentes capturas con las opciones de visualización.
6. Verifica las diferencias que presenta cuando seleccionas la resolución de nombres a nivel de MAC, de red y de transporte.
7. Ordena una captura por dirección de origen.
8. Ordena por protocolo.
9. ¿Cómo harías para visualizar únicamente las tramas que se corresponden con tu dirección IP?

DESAFÍOS:

1. ¿Te animas a modificar la forma de presentación del campo "Time" de una captura para que se pueda ver en formato dd/mm/ss?
2. ¿Encontraste cómo puedes aumentar y/o disminuir cada una de las 3 ventanas de visualización?
3. Busca en Internet los comandos "tcpdump" y prueba de filtrar la captura con los siguientes objetivos de captura:
    a. Únicamente las tramas dirigidas a tu ETD.
    b. Únicamente las tramas que salen de tu ETD.
    c. Selecciona una dirección (o nombre) destino, y captura únicamente el tráfico entre ambos.
4. ¿Encuentras alguna forma de "Colorear" para hacer más evidente algunas tramas que te sean de especial interés?

CAPÍTULO 3: El nivel FÍSICO.

Como ya hemos mencionado el nivel físico:

- Recibe las tramas de nivel 2, las convierte en señales eléctricas u ópticas y las envía por el canal de comunicaciones.
- Define aspectos mecánicos, eléctricos u ópticos y procedimentales.
- Algunas de las especificaciones más comunes son: RS 232, V.24/V.28, X.21, X.25, SONET, etc.
- Funciones y servicios:
  - Activar/desactivar la conexión física.
  - Transmitir las unidades de datos.
  - Gestión de la capa física.
  - Identificación de puntos extremos (punto a punto y multipunto).
  - Secuenciamiento de bit (entregar los bit en el mismo orden que los recibe).
  - Control de fallos físicos del canal.

Desde el punto de vista de la seguridad física, nos interesa tener en cuenta, los aspectos relacionados a:

- Edificios, instalaciones, locales.
- Autenticación y control de acceso físico.
- Medios empleados para la transmisión de la información.
- Conductos y gabinetes de comunicaciones.
- Medios físicos empleados para el almacenamiento (incluido backup) y procesamiento de la información.
- Documentación, listados, plantillas, planos, etc.

3.1. Edificios, instalaciones, locales.

En principio es necesario contar con planos de cada uno de ellos, hoy en día es obligatorio para toda edificación nueva la aprobación de los planos de telecomunicaciones, los cuales estandarizan con máximo detalles el "cableado físico" de nuestras instalaciones. La experiencia demuestra que este tipo de documentación es de vital importancia en virtud de lo flexibles que deben ser las redes

modernas. Contando con esta documentación, es mucho más fácil poder expandir nuestras redes con la máxima certeza y seguridad.

Si bien es necesario un nivel de seguridad mínima en todos los locales, esto no debe llevarnos a gastos o medidas de seguridad excesivas, es mucho más importante identificar los "sectores clave" de cada edificio y centrar allí nuestra atención. Es decir, cada inversión debe ser proporcional al recurso que estemos protegiendo, no tiene sentido invertir grandes sumas de dinero en medidas de seguridad física, sobre locales que no contienen información clave. Es mucho más importante realizar un trabajo metódico de identificación (que finalizará siendo un **Análisis de Riesgo** más adelante) de nuestros recursos, valorarlos y determinar qué impacto tienen los mismos en la Organización, para luego sí invertir adecuadamente en medidas de seguridad.

Una vez identificados, insistimos en volcar toda la información lo más detalladamente posible en los planos respectivos, y que los mismos pasen a formar parte de un "**Sistema de Gestión de la Seguridad de la Información**" (SGSI). Por ahora este concepto de SGSI (que aún no abordaremos), tiene que dejarnos la idea de que una documentación que se confecciona, si simplemente se guarda, tarde o temprano se pierde, se desactualiza o pierde valor. Por lo tanto, desde ya debemos ir concienciándonos en al menos crear una sencilla infraestructura para integrar todos los documentos, acciones, medidas y decisiones que se adopten de forma tal que todo ello esté siempre "VIVO", con ello queremos decir, que se controle, actualice, se firme, se pueda acceder siempre a la última versión, se sepa cómo y quién puede leer, modificar o eliminar y por último se integre al conjunto.

Las medidas de seguridad a considerar en las instalaciones se detallarán en la PART II de este libro, pero para darnos una idea inicial abordan el conjunto de barreras físicas, medidas contra incendio, humedad, climatización, contaminación ambiental e instalaciones eléctricas.

3.2. Autenticación y control de acceso físico.

- Autenticación: Identificar que quien dice ser, realmente lo sea.
- Control de Acceso físico: Permitir que cada rol identificado pueda acceder exclusivamente a donde esté autorizado.

Estos dos conceptos son el punto de partida de este apartado. No hay que abundar mucho en explicaciones, es exactamente lo que nos sucede cada vez que intentamos ingresar a un edificio donde existe un control de seguridad. Nos piden nuestra documentación, que puede ser algo interno de esa Organización y con ello podré acceder al área donde estuviere mi área de trabajo, o soy una persona externa, ante lo cual se suele solicitar una identificación válida en ese entorno (por ejemplo el DNI), e ingresaré exclusivamente al sector habilitado para las visitas hasta que me venga a buscar alguien de la empresa y me acompañe, etc.

Lo primero que deseamos remarcar antes de seguir avanzando, es que en España (y en muchos países más), cada vez que un Organismo Público o Privado "toma nota" (en papel o en forma digital) de los datos personales para almacenarlos (sea por el tiempo que fuere), entra en juego la Ley Orgánica de Protección de datos (LOPD) y el Real decreto 1720 (dic-2007), los cuales nos

obligan a realizar un conjunto de medidas que no pueden ser dejadas de lado. En este texto, no entraremos en detalle sobre las mismas, pero no os vayáis a olvidar hacerlo si vais a trabajar con este tipo de datos.

La autenticación y control de acceso físico hoy ofrecen un sinnúmero de posibilidades ajustables a la necesidad concreta que tengamos. La implementación y fundamento de ellas las desarrollaremos con más detalle cuando abordemos temas relacionados a criptografía, pues en estos momentos nos quedarían muy en el aire si lo hiciéramos. En cuanto a las herramientas, el mercado las actualiza día a día, pero como idea tenemos: tarjetas con banda magnética, con chip, huella digitalizada, sistemas de claves (por teclado, voz, rostro, iris), cámaras y personal de vigilancia, sistemas electrónicos de pasarela, etc.

3.3. Medios empleados para la transmisión de la información.

Sobre este tema nos explayaremos bastante pues consideramos que se debe tener claro desde dónde empezamos a considerar la seguridad de nuestros sistemas.

Toda señal de comunicaciones para propagarse necesita de un medio físico, sin éste sería imposible establecer una comunicación; en la actualidad los medios físicos que contamos son los siguientes:

3.3.1. Cable de pares trenzados:

El cable de pares se compone de conjuntos de pares conductores (enlazados) torsionados entre sí, con pasos de torsión distintos en cada par para evitar cruces por diafonía.

El diámetro de los hilos está entre 0.32 y 0.91mm. Ahora se utiliza para transmisión de alta frecuencia en MDF y MDT para distancias medias y cortas.

3.3.2. Cable de cuadretes:

Es un caso particular del caso anterior que aún sigue vigente en los millones de tendidos telefónicos. En vez de enlazar 2 hilos, se enlazan 4. Hay 2 tipos: el cuadrete en estrella y DM. Los cables de pares, cable de cuadretes en estrella y DM tienen un margen de utilización de frecuencia muy bajo, su frecuencia de utilización más alta es 300khz analógica, y si es digital se puede llegar a 4Mhz.

3.3.3. Cables trenzados de 4 pares:

Un par de cables trenzados es un par de alambres que se cruzan o trenzan entre sí para minimizar la interferencia electromagnética entre los pares de cables.

Cada par de cables conforma un enlace para transmisión de señales de datos completo. El flujo entre ambos cables es igual, pero de sentido contrario. Este flujo de corrientes produce campos electromagnéticos que pueden introducir ruidos a los pares vecinos. De todos modos, los campos correspondientes a cada par de cables tienen polaridades opuestas. Trenzando los cables entre sí, los campos magnéticos de cada uno se cancelan mutuamente, lo cual minimiza el ruido y/o la interferencia generada por cada par de cables.

Mediante 2 boletines técnicos (TSB 36: Especificaciones de cables y TSB 40: Equipos de interconexión, jacks, patcheras, etc), dividen al tipo de cable UTP [Unshield Twisted Pair] en varias categorías diferentes, según su ancho de banda:

- Cat **3**: Hasta 16 Mhz
- Cat **4**: Hasta 20 Mhz
- Cat **5:** (Cable sólido de pares trenzados), 22 o 24 AWG (0,643mm o 0,511mm), 100 Mhz
- Cat **5e:** (Categoría 5 mejorada), 26 AWG (0,409mm), 100 Mhz, UTP
- Cat **6:** (Cable sólido de pares trenzados), 24 AWG (0,511mm), 300 Mhz, FTP
- Cat 7**:** (Cable sólido de pares trenzados apantallados por par),23 AWG (~0,600mm), 600 Mhz, STP

El UTP Cat 5e es el que aún domina en el mercado. Es un cable diseñado específicamente para la transmisión de datos y se basa en pares de alambres de cobre retorcidos mediante una hélice en sentido antihorario y una vuelta de 5 a 15 cm. (A mayor cantidad de vueltas por cm es de mayor calidad, pero también más difícil de manipular).

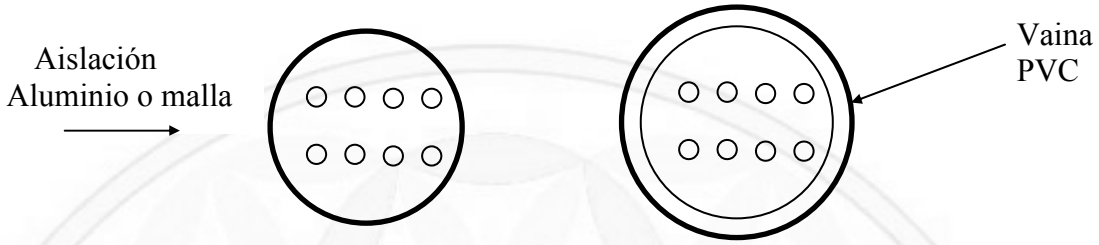
Este giro sobre sí mismo le permite eliminar tanto las componentes internas como externas de inducción y modulación cruzada, agrupando en el mismo cuatro pares diferentes. En un cable dado, cada par tiene un paso diferente del resto de los pares, y esto hace que un cable sea una unidad fabricada bajo estrictas especificaciones y no un simple conjunto de pares.

Esto mismo hace que su instalación deba ser más cuidadosa y considerar que no se puede tirar violentamente del mismo ya que variaría el paso de la hélice del roscado y por lo tanto la respuesta física del cable.

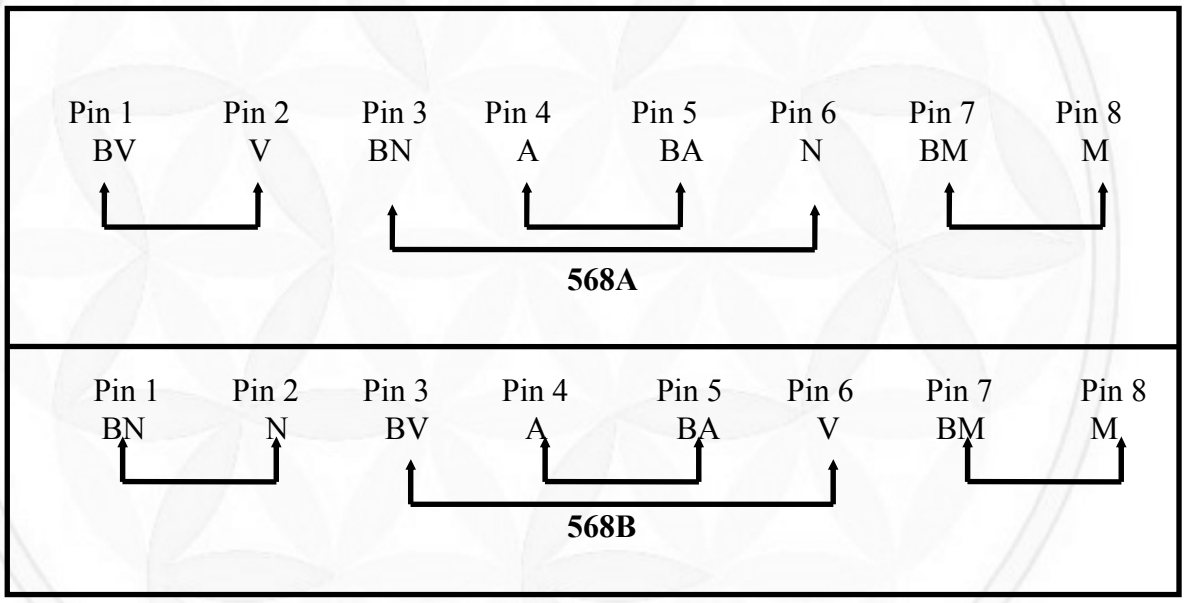
La impedancia característica del mismo es de 100 Ohms y la longitud máxima de cada segmento de 100 mts.

Para el caso de datos hasta categoría 5e, de los cuatro pares posibles se usan 2, uno para transmisión y otro para recepción, quedando dos libres. Este concepto ya no aplica a categoría 6 y 7.

Una variación de este cable es el que se conoce como STP (shield twisted pair), que es el mismo cable anterior con un blindaje externo, generalmente un papel de aluminio. Si bien puede disminuir aun más la interferencia obliga a tener un sistema de masas donde en ningún caso existan más de 3 ohms entre los conectores y la masa del sistema.



Hay dos estándares de conexión de los pares de cables trenzados, según se muestra en la figura:



En la figura anterior, las abreviaturas se corresponden a:

V:	verde
N:	Naranja
A:	Azul
M:	Marrón
BV:	Blanco y Verde, **BN**: Blanco y Naranja, **BA**: Blanco y Azul, **BM**: Blanco y Marrón

3.3.4. Cable coaxial:

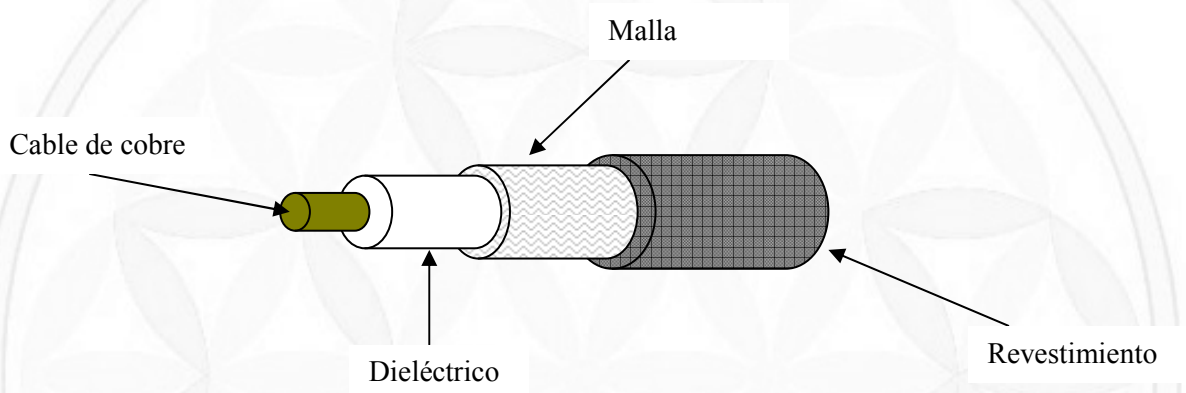
Este cable ya prácticamente se está dejando de emplear pues se reemplazó por fibra óptica, pero lo mencionamos aquí pues aún está instalado en varias redes. El cable coaxial consiste en

un conductor recubierto en primer lugar por material aislante, luego por una malla conductora y finalmente por una cubierta de material plástico aislante flexible.

En las aplicaciones LAN, la malla es eléctricamente neutra, y sirve como una malla de protección interna de aislamiento de los ruidos del conductor. La malla también contribuye a eliminar las pérdidas de señal confinando la señal transmitida al cable.

El cable coaxial puede trabajar en un mismo rango de frecuencias, a mayor distancia que el cable par trenzado, pero en contraposición, es más caro.

El cable coaxial de 50 Ohms está "reconocido" por la norma, pero **"no se recomienda"**, y la puesta a tierra se convierte en obligatoria de acuerdo a las prescripciones de la norma ANSI / TIA / EIA 607, como parte integral del cableado de telecomunicaciones.

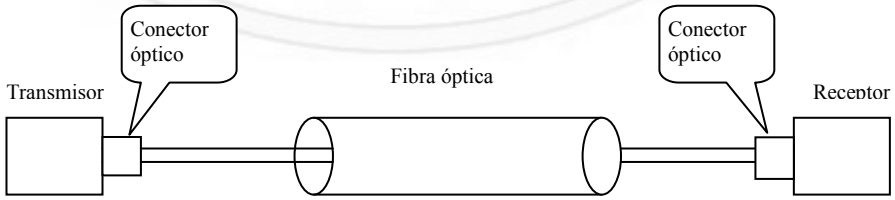


3.3.5. Fibra óptica:

3.3.5.1. Sistemas de transmisión de fibra óptica.

Para poder implementar la tecnología de transmisión por medio de luz, es necesario contar con todo un sistema diseñado para este uso, los componentes básicos del mismo son:

a. Fuente óptica: convierte la señal eléctrica en luz.
b. El cable de fibra óptica que transporta la señal.
c. El detector óptico que convierte la señal nuevamente a electrones.



Como fuentes ópticas se emplean comúnmente el diodo LED o LD de modulación directa, mientras que como detector óptico se emplean el ADP o el PIN – PD de alta sensibilidad y de respuesta veloz.

3.3.5.2. Características de la luz.

La luz se puede definir como el agente físico que ilumina objetos y los hace visibles, siendo emitida por cuerpos en combustión, ignición, incandescencia, etc. Desde el punto de vista físico, la luz es una radiación u onda electromagnética. El espectro electromagnético se extiende desde las ondas de radio hasta los rayos gamma. De todo este espectro, sólo una zona muy pequeña es detectable por el ojo humano, y es lo que se llama el espectro visible o luz visible.

Toda onda está caracterizada por dos parámetros fundamentales:

- La velocidad de propagación.
- La frecuencia.

La velocidad de propagación es la distancia recorrida por una señal en una unidad de tiempo. Toda onda electromagnética se desplaza en el vacío a 300.000 km/s. La frecuencia es el número de veces que la onda repite su período en un segundo; en el caso de la luz es del orden de varios cientos de billones de ciclos por segundo.

Otro parámetro a considerar es el de longitud de onda, que se refiere a la distancia que la señal viaja durante un período, es por esta razón que se mide en metros.

$$\lambda = C / f$$

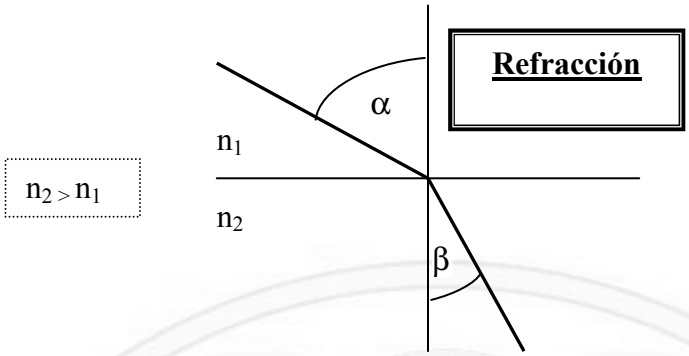
- λ: Longitud de onda.
- C: Velocidad de propagación.
- f: Frecuencia.

La idea de longitud de onda o de frecuencia dentro del espectro visible, se asocia a la idea de un determinado color de una determinada luz. Una luz de un color puro se llama monocromática. Si está compuesta por todos los colores, se llama luz blanca.

3.3.5.3. Propagación de la luz:

La luz se propaga en el vacío en forma rectilínea de acuerdo con lo que se denomina rayo o haz lumínico; en cualquier medio transparente cumple esta propiedad, siempre que la composición de ese material sea la misma en todo su recorrido. Todo medio físico opondrá resistencia al paso de una señal electromagnética, produciendo el efecto de disminuir su velocidad respecto al vacío. La relación entre la velocidad de la luz en el vacío y en un medio real se denomina índice de refracción.

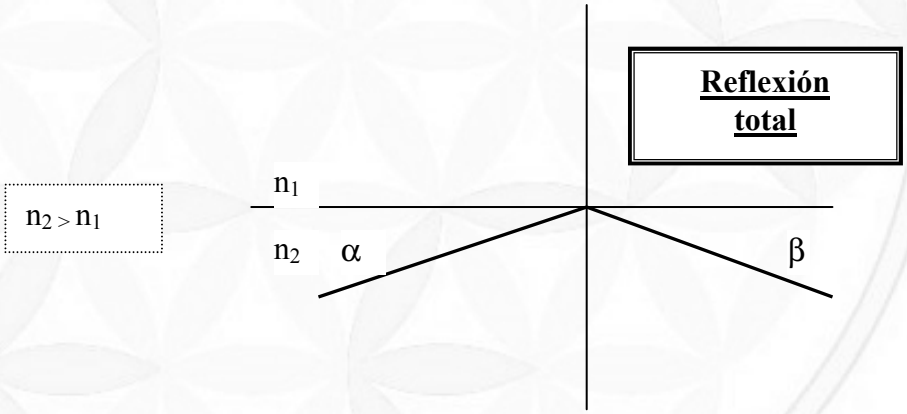
3.3.5.4. Reflexión y refracción de la luz.



Al incidir una onda luminosa sobre una superficie plana divisoria de dos medios de índice de refracción diferente, su trayectoria se desviará acorde a la siguiente relación:

$$\text{Sen } \alpha / \text{sen } \beta = n_2 / n_1$$

Como se puede apreciar, si se va incrementando el ángulo de incidencia desde n_2 a n_1, llegará un momento en el cual, el ángulo α llegará a ser de 90 grados, siendo siempre β menor a este valor (si: $n_2 > n_1$). Superado este umbral, el haz de luz deja de pasar a la superficie n_1, para producir el fenómeno denominado Reflexión total, en el cual la luz se propaga dentro del medio n_2, con un ángulo igual al de incidencia.



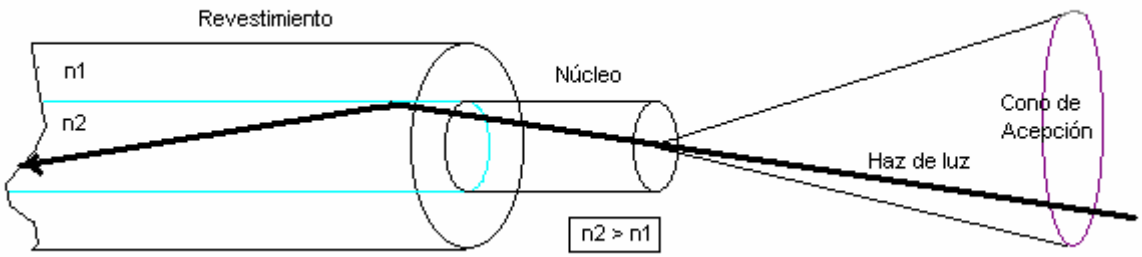
Si el ángulo de incidencia del haz de luz se mantiene inferior al valor descrito, la luz se reflejará dentro de la superficie n_2. Al ángulo dentro del cual se produce la reflexión total se lo denomina Angulo de acepción o aceptación.

3.3.5.5. Fibra óptica (F.O.) (Descripción general).

La F.O. es un dispositivo de material dieléctrico (no conductor de c.e.) que es capaz de confinar y guiar la luz.

Las F.O. usadas en telecomunicaciones están formadas por dos cilindros concéntricos llamados núcleo y revestimiento con diferentes índices de refracción (n1 en el revestimiento y n2 en el núcleo), por medio de los cuales, si se ingresa un haz de luz dentro del cono de acepción, se producirá una y otra vez el fenómeno de reflexión total, transportando de esta forma la señal.

Los diámetros que se suelen emplear son 125 µm para el revestimiento y desde los 9 a 62,5 µm para el núcleo acorde al modo (a tratar más adelante).



3.3.5.6. Propiedades de la F.O:

Basados en los diferentes parámetros del material, es que se puede clasificar las F.O. en cuatro grupos acorde a diferentes propiedades:

a.	**Propiedades Ópticas**	a.1.	Perfil de refracción	a.1.1.	Monomodo	
				a.1.2.	Multimodo	- Índice gradual
						- Índice escalón
		a.2.	Apertura Numérica			
b.	**Propiedades de transmisión**	b.1.	Atenuación	b.1.1.	Intrínseca	
				b.1.2.	Extrínseca	
		b.2.	Ancho de banda			
		b.3.	Diámetro del Campo modal			
		b.4.	Longitud de onda de corte			
c.	**Propiedades geométricas**	c.1.	Diámetro del revestimiento			
		c.2.	Diámetro del núcleo.			
		c.3.	Concentricidad			
		c.4.	No circularidad			
d.	**Propiedades físicas**	d.1.	Módulo de Young			
		d.2.	Carga de rotura			
		d.3.	Alargamiento en el punto de rotura			
		d.4.	Coeficiente de dilatación lineal			

3.3.6. Radiocomunicaciones:

Técnicas que permiten el intercambio de información entre dos puntos geográficamente distantes mediante la transmisión y recepción de ondas electromagnéticas.

Espectro de radiofrecuencias

Banda	Designación	Longitud de onda	Uso en comunicaciones
300m KHz – 3 MHz	MF	1 km – 100 m	Radiodifusión AM
3 MHz – 30 MHz	HF	100 m – 10 m	Onda corta (Radioaficionados)
30 MHz – 300 MHz	VHF	10 m – 1 m	TV – Radio FM - Radiollamadas
300 MHz – 3 GHz	UHF	1 m – 10 cm	Microondas – TV
3 GHz – 30 GHz	SHF	10 cm – 1 cm	Microondas - Satélite

3.3.6.1. Naturaleza de las ondas de radio:

Cuando se aplica una potencia de radiofrecuencia a una antena, los electrones contenidos en el metal comienzan a oscilar. Estos electrones en movimiento constituyen una corriente eléctrica que produce la aparición de un campo magnético concéntrico al conductor y un campo electrostático cuyas líneas de fuerza son perpendiculares a las líneas de fuerza del campo magnético. Estos campos siguen paso a paso las variaciones de la corriente eléctrica que les da origen.

La velocidad de las ondas de radio que viajan en el espacio libre es igual a la velocidad de la luz, es decir 300.000 km/s, y la relación entre longitud de onda y frecuencia está dada por la ecuación:

$$C = \lambda * f$$

C: Velocidad de la luz.

λ: Longitud de onda.

f: Frecuencia.

3.3.6.2. Propagación por onda terrestre:

En este caso las ondas se mantienen en contacto permanente con la superficie terrestre. Como consecuencia de ello, el contacto con el terreno provoca la aparición de corrientes eléctricas que debilitan la señal original a medida que se aleja de la antena emisora. Este tipo de señal es poco empleada en transmisión de datos.

3.3.6.3. Propagación por onda espacial o ionosférica:

Con excepción de las comunicaciones locales que pueden realizarse con onda terrestre, la mayoría de las comunicaciones comprendidas en la banda de 3 a 30 MHz (o HF) se efectúan por onda espacial.

Esta transmisión se basa en la capa de la atmósfera denominada ionósfera, la cual en virtud de los rayos ultravioletas que chocan con los átomos de esta capa produciendo que algunos electrones salten hacia niveles mas externos, se encuentra con una gran presencia de iones positivos y electrones libres dependiendo su cantidad de la mayor o menor incidencia de los rayos solares.

Al llegar a esta capa, las ondas de radio provenientes de la Tierra, se produce un fenómeno de refracción que devuelve los mismos a la corteza terrestre. El comportamiento de la Ionosfera es informado permanentemente por laboratorios especializados a través de los mapas ionosféricos.

Esta técnica no es muy conveniente en la transmisión de información por la gran distancia que recorre y los ruidos que va sumando a lo largo de ella.

3.3.6.4. Propagación en línea recta o de alcance visual (o directo):

Esta transmisión como su nombre lo indica, implica el alcance visual de las antenas, lo cual, en virtud de la curvatura terrestre, está bastante limitado. Se emplea fundamentalmente en VHF y UHF. Un ejemplo de esta son las transmisiones de TV y FM.

3.3.7. Microondas:

Sistemas de telecomunicaciones que trabajan en UHF y aún más altas y utilizan el haz radioeléctrico como si fuera un rayo de luz para establecer un enlace punto a punto entre dos estaciones.

Se pueden clasificar en:

- Analógicos:

  Estas fueron las primeras en emplearse y su empleo fue para telefonía y televisión. Ya no se fabrican más pero aún quedan muchas instaladas.

- Digitales:

  Son los que en la actualidad acapararon el mercado de las microondas, emplean modulaciones multinivel y en cuadratura (QAM), y poseen un amplio ancho de banda, superando varios centenares de Mbps.

  Dentro de esta clasificación entrarían las actuales redes **Wifi,** pues se encuentran en el rango de los 2,4-2,485 GHz para el estándar **802.11b**. aunque también debemos considerar el estándar **802.11ª:** que opera en 5,1-5,2 Ghz, 5,2-5,3 Ghz, 5,7-5,8 GHz). Cuando tratemos el nivel de enlace ampliaremos más estos conceptos.

3.3.8. Comunicaciones satelitales:

Sistema de comunicaciones que emplea uno o más satélites para reflejar las ondas electromagnéticas generadas por una estación transmisora con el objeto de hacerla llegar a otra estación receptora. Generalmente ambas están situadas en puntos geográficamente distantes, sin alcance visual.

Los satélites empleados en telecomunicaciones son los llamados geoestacionarios es decir que se encuentran situados en un punto fijo respeto a la Tierra, pero en la actualidad se están desarrollando otros tipos de órbitas para telecomunicaciones, en especial las de baja altura para evitar las enormes distancias que actualmente recorren las señales.

Los satélites se clasifican en:

- **LEO (Low Earth Orbit):** Poseen órbitas elípticas que oscilan entre los 400 y 2.500 km de altura.
- **MEO (Medium Earth Orbit):** Poseen órbitas elípticas que oscilan entre los 4.000 y los 15.000 km de altura
- **GEO (Geostationary Earth Orbit):** Poseen órbitas circulares que giran en un punto fijo respecto a la Tierra, se encuentran a 36.000 km de altura.

Un satélite posee dos antenas, una receptora (Uplink) que recibe la información de la Tierra y una transmisora (Downlink) que refleja la señal cambiada de frecuencia par no interferirse mutuamente.

Según su uso pueden ser de cobertura global, hemisférica o direccional (spot).

Los transponder son los sistemas encargados de recibir la señal, cambiar la frecuencia, amplificarla y retransmitirla (también suelen incluir funciones de multiplexado/demultiplexado); cada transponder abarca un número fijo de canales. Los transponder manejan varios anchos de banda, siendo los más usuales 36, 70 y 140 MHz. El número de transponder varía según el tipo de satélite.

3.3.9. Guía de onda:

Medio apto para la transmisión de señales de longitud de onda micrométricas. Estas señales se emplean en los sistemas de comunicaciones que trabajan a frecuencias elevadas y en distancias cortas, principalmente para conexiones entre antena y equipo transceptor.

En general las guías de onda están fabricadas con tubos huecos de una longitud entre 5 y 15 metros y con una sección acorde a la longitud de onda que se desea transmitir, que oscila ente 0,7 y 16 centímetros.

El material que se suele emplear es cobre para las longitudes de onda entre 3 a 9 mm y aluminio anodizado para longitudes de onda entre 10 a 25 cm.

3.3.10. Láser:

Equipos de telecomunicaciones que transmiten por medio de emisores que generan un haz de luz coherente (que podrá o no ser visible al ojo humano). Este haz convenientemente modulado permite transmitir señales de información entre dos puntos geográficamente distantes.

Su alcance estará limitado por la potencia de su haz de luz y por supuesto el alcance visual (máximo aproximado 10 km); se debe tener especialmente en cuenta que son sumamente vulnerables a todo aquello que afecte el haz de luz (niebla, lluvia, polvo, etc.).

Respecto a las microondas y las transmisiones de radio, posee mucho mayor ancho de banda en distancias cortas, lo que lo hacen especialmente apto para interconexiones de redes LAN en ciudades.

3.3.11. Infrarrojos:

Este tipo de transmisión se inició para interconexión de hardware a muy corta distancia (calculadoras, Palm, etc…) y en la actualidad han evolucionado como un medio muy eficaz de implementación de redes LAN inalámbricas.

3.4. Conductos y gabinetes de comunicaciones.

3.4.1. Los conductos:

Los conductos (o vías, o ductos) son los espacios que se reservan para el paso de los medios de comunicaciones (habitualmente cables y/o fibras ópticas).

Las instalaciones modernas suelen venir preparadas para el tendido de estos medios e identificados en sus planos los mismos. Estos diseños se hacen para albergar tanto telefonía como datos.

En la actualidad es obligatorio su diseño y la construcción con los planos correspondientes en toda obra nueva, firmados por personal autorizado, y estrictamente separados de los conductos eléctricos.

Los conductos de comunicaciones se los suele relacionar con el concepto de:

- Cableado Vertical: O troncal, es el que establece la comunicación entre gabinetes de comunicaciones.
- Cableado Horizontal: El que llega a cada puesto de trabajo.

Existen diferentes conductos de comunicaciones:

- Falsos suelos o techos: Suelen encontrarse en oficinas, y se trata de "losas" (de yeso, plástico o resinas) que se pueden mover manualmente y dejan espacios (entre 30 y 70 cm) en los cuales habitualmente están instaladas bandejas metálicas o plásticas sobre las que se van instalando los medios de comunicaciones (es una muy buena práctica precintarlos).
- Tubos o caños: Pueden ser metálicos o de PVC, varían su diámetro en base al diseño y capacidad de la edificación, y de ello dependerá la cantidad de medios que puedan ser pasados dentro de ellos.
- Montantes: Se suele denominar así a los espacios verticales que unen diferentes pisos de un edificio y que permiten el acceso al personal, generalmente mediante escaleras verticales adosadas al muro interno, y con puerta de acceso y gabinete empotrado en cada planta.
- Embellecedores, canaletas o cable canal: Existe un sinnúmero de este tipo, de toda forma y color. Son muy prácticos a la hora de instalaciones que no cuentan con los anteriores y no se pretende romper mampostería, pues sencillamente van adosados o pegados a la pared quedando a la vista.

3.4.2. Gabinetes de Comunicaciones (o Rack de comunicaciones):

Los gabinetes de comunicaciones, son los "armarios" donde debe instalarse el Hardware de red. Lo que deberíamos hacer, es colocar uno de ellos en cada zona desde la cual vayamos a iniciar las "acometidas" a los puestos de usuario, y luego cada uno de ellos conectado (de forma troncal) hasta el o los que coloque en el CPD (Centro de Procesamiento de Datos). Por supuesto encontraremos de las más variadas arquitecturas e inimaginables estrategias al respecto, pero las normas y buenas prácticas intentan aconsejarnos respetar lo dicho al principio de este párrafo.

Un gabinete de comunicaciones tiene un ancho estándar de 19 pulgadas = 49 cm, y su altura se suele medir en "unidades". Cada unidad son 4,5 cm y es el ancho mínimo que tiene un módulo "rackeable", llamados así todos los dispositivos de red que están diseñados para ser instalados en un "Rack" (o gabinete..... la verdad es que me suena espantoso si lo llamara "Gabineteable" ¿no?). Es decir, todo dispositivo de red que vaya a instalar dentro de un gabinete (si es "Rackeable") tiene 49 cm de ancho y múltiplos de 4,5 de alto, pues hay módulos que ocupan, 2, 3, 4 o "n" unidades de altura. También existen unas bandejas que se instalan dentro de un gabinete y sirven para apoyar en ellas elementos que no son "rackeables".

A la hora de adquirir un gabinete, entonces es muy importante que analice bien su capacidad, pues de ello dependerá la cantidad de "unidades" que pueda poner en él. Como buena práctica SIEMPRE es conveniente tomarse un margen de 2 o más unidades de más para futuros crecimientos y también porque dentro del mismo gabinete debemos tener en cuenta siempre que deberá estar la tensión de alimentación de cada componente, lo cual nos suele sorprender por el espacio que terminan ocupando varios enchufes de corriente eléctrica.

Como imaginaréis los gabinetes se pueden adquirir del tamaño que se nos ocurra, desde 3 unidades hasta algunos más altos que una persona e inclusive de varias puertas y múltiplos de su

ancho (49 cm, 98 cm, 196 cm, etc...). con puertas de acceso frontales, laterales, traseras, superiores, con seguridad (llave), sin seguridad, con puertas de cristal, metal, plástico, etc... se trata de toda una industria.

3.5. Medios físicos empleados para el almacenamiento (incluido backup) y procesamiento de la información.

Los medios físicos que se mencionan aquí se presentan desde el punto de vista de la seguridad, es decir, aquí estamos hablando de servidores, torres de discos, cintas, CDs, DVDs, cajas fuertes, armarios.

Una información que se encuentre en tránsito, puede estar en un circuito de telecomunicaciones o fuera de él (por ejemplo en papel). En este apartado trataremos el caso contrario, es decir cuando no está en tránsito.

Toda información que no se encuentre en tránsito, puede estar inicialmente en dos estados:

- En Línea: La información la consideramos en línea, cuando los datos ingresados pasan de un lugar a otro en forma directa dentro del circuito informático.
- Fuera de Línea: En este caso, la información es retirada del circuito electrónico a través de una acción manual. El ejemplo más claro de este modo es la transferencia de información a través de un CD, DVD, Zip, memoria USB, etc.

Si la Información está en Línea, puede estar en dos estados:

- Almacenamiento digital: En memoria no volátil.
- Procesamiento digital: En memoria volátil.

En cualquiera de los anteriores, la información almacenada puede tratarse de:

- Original (o actual, o en servicio): Información sobre la cual concurre todo servicio o función para el normal desenvolvimiento del sistema informático.
- Copias de seguridad (backup): Información que es almacenada, para cualquier posible necesidad de recuperación futura.

Esta distinción es muy importante de tener en cuenta, pues las medidas de seguridad difieren bastante. Una información que se encuentra fuera de línea hasta podríamos decir que es la más peligrosa de todas, pues debemos mantener al detalle todo su "Ciclo de vida": cuándo salió, cuántas copias sacó, quién la sacó, de dónde la sacó, dónde está guardada, actualización, mantenimiento, borrado, destrucción, etc... Y si en algún momento cualquiera de estos datos quedan fuera de nuestro control, el impacto puede ser muy serio (si la información es valiosa), pues le hemos "perdido el rastro".

Una información que se encuentra en memoria volátil, desde el punto de la seguridad, lo más importante podríamos presentarlo como que es "no perderla" en cambio en memoria no volátil

debemos plantearnos su: Integridad, Confidencialidad y Disponibilidad (temas que trataremos más adelante).

Para cada una de estas situaciones, se implantan medidas de seguridad diferente, las cuales se tratan en la PARTE II de este libro.

3.6. Documentación, listados, plantillas, planos, etc.

Todo lo mencionado en este capítulo no tendría sentido si no se deja una constancia y documentación clara, concisa y entendible del conjunto de medidas de este nivel. Como ya se mencionó, toda esta documentación deberá estar integrada a un SGSI para que sea eficiente.

A continuación hacemos referencia a algunos de estos documentos y medidas:

- Planos de la red.
- Identificación de los medios de comunicaciones, numeración, extremos, puestos de trabajo conectados y bocas vacantes, tramos críticos.
- Planos y documentación de gabinetes de comunicaciones.
- Planos de los locales y conductos.
- Dispositivos de Hardware de red, dispositivos existentes, ubicación, claves de acceso, configuración de los mismos.
- Documentación de configuraciones, permisos de accesos, habilitación o deshabilitación de locales e instalaciones.
- Certificaciones de los medios de comunicaciones.
- Mecanismos de control de cambios.
- Plan de inspecciones periódicas.
- Plan de inspección física (recorridas, controles, verificación remota de configuraciones, control).
- Inventarios de equipamiento.
- Medidas de resguardo de información y control de actas de destrucción.
- Seguridad física en la guarda de documentación y archivos.
- Seguridad física de los locales y sistemas de monitorización de acceso.
- Coordinaciones con el personal de seguridad.
- Planes y medidas contra incendio, evacuación y contingencias.
- Medidas de control de Hardware de red.
- Medidas de control de otros componentes de acceso: módem, DTU/CSU, PAD en X.25, Placas ISDN, ATM, etc.

EJERCICIOS DEL CAPÍTULO 3

1. Si tienes acceso a alguna red de sistemas de información, o conoces alguien que te permita trabajar con una, lo primero que te invitamos a hacer es que verifiques si existen o no planos de la misma, los analices, compruebes la realidad con lo volcado en los mismos. De no existir, intenta hacer un relevamiento y comienza a darle forma a los mismos. (NOTA: Sería muy importante que puedas tener acceso a algún tipo de redes para seguir adelante con los ejercicios de este capítulo, y es la base de nuestra pirámide de seguridad).
2. Sobre esos planos, verifica sobre "el terreno" si están claramente etiquetados, si su situación es "caótica" (como suelen ser los gabinetes y ductos de cableado), intenta identificar si las características técnicas se corresponden con lo que desarrollamos en la teoría (sellos y marcas de fábrica, categoría de los cables, conectores, rosetas, regletas, etc…).
3. A primera vista, ¿qué encuentras positivo y negativo en cuanto a medidas de seguridad física?
4. ¿Has podido identificar zonas o sectores clave dentro de esa red?, ¿están bien demarcados?, ¿están bien seguros a nivel físico?
5. ¿Existe más documentación de los aspectos físicos de este sistema?, ¿la documentación está integrada con algún sistema de gestión (SGSI)?, ¿se actualiza la misma?, ¿hay registros de actualización, mantenimiento, cambios y mejoras?
6. ¿Encuentras a la vista medidas de Autenticación y control de acceso físico a estas dependencias?, ¿te parecen adecuadas?
7. ¿Qué medios de comunicaciones físicos has encontrado en esta red? Descríbelos, investiga sobre sus características, compáralos con la teoría desarrollada en el capítulo.
8. La totalidad de la "cadena física" de toda esta infraestructura, ¿Mantiene una misma característica? Es decir, ¿desde cualquier extremo se puede transmitir información a la velocidad máxima de esa red o encuentras "cuellos de botella" o dispositivos que no permiten esa velocidad y por lo tanto baja el rendimiento total?
9. Si tienes cables UTP y/o STP, trata de identificar su conectorizado (orden de colores, norma EIA/TIA 568ª o B), tanto en conectores como en rosetas, y match panel.
10. Si tienes fibras ópticas, investiga e identifica sus características (monomodo, multimodo, cantidad de "hilos" por cable, tipos de conector, latiguillos, etc). (Puedes ampliar información en el **ANEXO 1**).
11. Si tienes fibra óptica, haz un pequeño recorrido de su tendido para verificar si existen microcurvaturas (ángulos muy cerrados), o en sus extremos se encuentra debidamente "enroscada" en las cajas de acometida.
12. Verifica la existencia de cables o fibras redundantes en toda la red (para suplantar cualquier anomalía).
13. Si tienes algún tipo de radioenlace, identifica claramente sus extremos, y verifica su alcance visual o no, y hasta puedes hacer la prueba de generar interferencias con otros dispositivos o hasta haciendo "sombras" (por supuesto si no afectas el normal funcionamiento de la red).

14. Recorre los diferentes tipos de conductos de comunicaciones, y repasa los conceptos de cableado horizontal y vertical, ¿se ajustan a la teoría?
15. ¿Encuentras tramos o zonas donde los cables o fibras estén a la vista o puedan sufrir desperfectos por encontrarse mal colocados?, ¿cómo lo solucionarías?
16. Identifica los gabinetes de comunicaciones, ¿son estándares?, ¿responden a lo que desarrollamos en el capítulo?, ¿poseen alguna medida de seguridad física?, ¿su interior está identificado y claramente empaquetados sus medios?
17. Si tienes acceso al interior de algún gabinete, intenta seguir alguno de sus cables o fibras, analiza como es esta conexión, qué está comunicando, dónde están las tomas de tensión, ¿tienen unidades libres aún?, ¿cuántas?, ¿responde a alguna lógica la conexión?, ¿está bien ventilado?
18. Respecto a medidas físicas para el resguardo de la información, ¿Existen?, ¿Hay armarios o gabinetes seguros donde se almacene esta información?
19. ¿Se aprecian medidas físicas de identificación de la información de resguardo?, ¿se identifican y enumeran las copias de seguridad, los originales?
20. ¿Existe algún medio físico para la destrucción de la información obsoleta, o derogada?, ¿se deja alguna constancia de ello?
21. El Hardware de red, ¿está etiquetado e identificado?
22. ¿Existe alguna certificación de esta red?
23. ¿Hay algún plan de inspecciones periódicas?, ¿se cumple?, ¿se deja constancia?
24. ¿Hay personal de seguridad física?, ¿se coordina su actividad con la de la seguridad de la red y los sistemas?, ¿se deja constancia de ello?

DESAFÍOS:

Si tuvieras acceso a herramientas de cableado, te invitamos a que le dediques algún tiempo a:

- Armado de latiguillos (con pinza crimpeadora).
- Conectorizado de rosetas y patch pannel (con percutor).
- Identificación de cableado (con generador de tonos).
- Medición de pares (con analizador de pares).
- Certificación de enlaces (con herramientas de certificación).
- Medición de fibra óptica (con herramienta de medición).
- Conectorizado de fibra óptica (con instrumental de conectorizado).
- Análisis de potencia y alineación (para radioenlaces: desde los mismos dispositivos, en general cuentan con software para esta actividad).

NOTA: Para profundizar en estos aspectos puedes consultar el **ANEXO 1.**

CAPÍTULO 4: El nivel de ENLACE

Como ya hemos mencionado, existen varios protocolos de comunicaciones que operan a nivel de enlace. En este texto, nos centraremos exclusivamente en los dos que para un Administrador de Sistemas tienen mayor vigencia en la actualidad, los cuales están regulados por IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) como 802.3 (CSMA/CD – Ethernet) y 802.11 (Redes inalámbricas WLAN).

La familia 802.X, tiene este nombre justamente porque se crea un comité de IEEE en el año 80 durante el mes de febrero (2), cuando el concepto de redes LAN comienza a imponerse como algo digno de ser analizado. Dentro de este comité se conforman diferentes grupos de trabajo, los cuales en la actualidad son denominados de la siguiente forma:

- IEEE 802.1 – Normalización de interfaz.
- IEEE 802.2 – Control de enlace lógico.
- IEEE 802.3 – CSMA / CD (ETHERNET)
- IEEE 802.4 – Token bus.
- IEEE 802.5 – Token ring.
- IEEE 802.6 – MAN (ciudad) (fibra óptica)
- IEEE 802.7 – Grupo Asesor en Banda ancha.
- IEEE 802.8 – Grupo Asesor en Fibras Ópticas.
- IEEE 802.9 – Voz y datos en LAN.
- IEEE 802.10 – Seguridad.
- IEEE 802.11 – Redes inalámbricas WLAN.
- IEEE 802.12 – Prioridad por demanda
- IEEE 802.13 – Se ha evitado su uso por superstición
- IEEE 802.14 – Modems de cable.
- IEEE 802.15 – WPAN (Bluetooth)
- IEEE 802.16 - Redes de acceso metropolitanas sin hilos de banda ancha (WIMAX)
- IEEE 802.17 – Anillo de paquete elástico.
- IEEE 802.18 – Grupo de Asesoría Técnica sobre Normativas de Radio.
- IEEE 802.19 – Grupo de Asesoría Técnica sobre Coexistencia.
- IEEE 802.20 – Mobile Broadband Wireless Access.
- IEEE 802.21 – Media Independent Handoff.
- IEEE 802.22 – Wireless Regional Area Network.

4.1. Análisis de tramas Ethernet (IEEE 802.3):

El funcionamiento de una red LAN a nivel dos (enlace) que opere por medio de CSMA/CD (Carrier Sence Multiple Access/Colition Detect) se implementa por medio del protocolo Ethernet u 802.3 (la mínima diferencia entre ellas se verá en breve). Su funcionamiento es básicamente simple, si el canal está libre entonces se puede transmitir, caso contrario no. Como existe la posibilidad que un ETD escuche el canal, al estar éste libre comience la transmisión, y antes de llegar esta señal a cualquiera de los otros ETD de la LAN alguno de estos haga lo mismo, es que se analizan las colisiones. Una colisión se produce cuando dos ETD por tener el canal libre inician su transmisión, la cual no es otra cosa que un estado de tensión que oscila entre + 0,85Volt y - 0,85 Volt (o ausencia de ella) que se propaga por canal físico, al encontrarse dos señales dentro del mismo medio físico se produce una alteración en los niveles de tensión, la cual al llegar a cualquier ETD de la red se determina como una colisión. Los ETD que transmitieron pasan a un algoritmo de espera aleatorio (llamado disminución exponencial binaria) e intentan transmitir nuevamente al cumplirse el plazo determinado por el algoritmo (son múltiplos de un valor muy especial que se llama tiempo de ranura), si durante 51,2 microsegundos (tiempo de ranura) no se detecta ninguna colisión, éste se ha APROPIADO del canal y se asegura que ningún otro ETD pueda transmitir, por lo cual continuará con el resto de su trama (tamaño máximo 1518 Byte) y luego entrará nuevamente en compulsa por el medio físico.

4.1.1. Formato de las direcciones MAC (Medium Access Control).

Las Direcciones MAC son reguladas por IEEE y están formadas por 6 octetos, representados como pares de números hexadecimales (hh-hh-hh-hh-hh-hh).

Los primeros tres octetos identifican al fabricante de la tarjeta. Estos tres octetos son asignados por un grupo de IEEE llamado **RAC** (Registration Authority Commitee) y pueden ser consultados en http://www.ieee.org/index.html. Existe una metodología para solicitarlos y por ser 24 bit, se pueden asignar en el orden de 16.000.000 de valores. Estos tres primeros octetos se les denomina "**OUI"** (Organizationally Unique Identifier) o "**company_id**", de estos 3 Byte, los dos primeros bit tienen un significado especial:

- **bit 0**: Individual (valor = 0), establece que este valor pertenece a una sola dirección MAC. Grupal (Valor = 1), forma parte de un conjunto de direcciones MAC.
- **bit 1**: Universal (valor = 0), define que esta dirección es única en el mundo. Local (valor = 1) tiene significado solamente en el ámbito local.

Estos primeros 3 octetos, una vez asignados a una determinada empresa, se deja a criterio de la misma cómo asignará los valores de los 3 octetos siguientes denominados "**Extension identifier**", para que no puedan repetirse, pero IEEE-RAC no se responsabiliza ni establece ninguna pauta sobre los mismos. Es lógico pensar que un gran fabricante de tarjetas, complete la totalidad de los posibles números a emitir; IEEE-RAC establece que recién al haber completado el 90 % de las asignaciones, podrá solicitar otro OUI para continuar fabricando (en la actualidad ya existen varias empresas en esta situación).

La concatenación de **"OUI" + Extension identifier = "EUI" (Extended Unique Identifier,** conocido como "**EUI-48**", que es la verdadera denominación teórica de una dirección MAC.

Ejemplo de Representación gráfica de una EUI-48					
OUI (company_id)			**Extension Identifier**		
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
AC	DE	48	23	45	67
10101100	11011110	01001000	00100011	01000101	01100111
▲ bit más significativo					▲ bit menos significativo

4.1.2. Ethernet y 802.3:

El funcionamiento de este protocolo tiene sus orígenes en otro conocido como ALOHA (saludo de los hawaianos, que es donde nació), al principio se creyó muy poco probable que esta lógica de compulsa por un medio de comunicaciones fuera eficiente, pero en el muy corto plazo se descubrió que sí lo era. Digital, Intel y Xerox, se unen para ponerlo en funcionamiento sobre cable coaxial a 10 Mbps, y como inicialmente se lo empleó en enlaces satelitales que transmitían al "Ether", se le llamó Ethernet (y se le conocía como Ethernet DIX). En el año 1980 (80) y en el mes de febrero (2), IEEE toma cartas en el tema y crea el subcomité que estudiaría el tema de LAN y MAN, y por la fecha en que entra en funcionamiento se le llamó 802.x (x=distintas áreas), y será el responsable hasta la actualidad de regular el funcionamiento de estas redes.

Este grupo define todos los aspectos hoy conocidos como familia 802.x, de los cuales como mencionamos solamente en este texto se desea dejar claro algún aspecto de 802.3 y 802.11.

Lo más relevante aquí es que, si se recuerda el aspecto del nivel de enlace (nivel 2) del modelo OSI, éste "establece la comunicación con el nodo inmediatamente adyacente". En una topología LAN ¿cuál es el nodo inmediatamente adyacente? Ante esta cuestión IEEE, propone subdividir el nivel de enlace del modelo OSI en dos subniveles:

- **MAC** (Medium Acces Control): Responsable de todo lo referente al Hardware de red.
- **LLC** (Logical Link Control), 802.2 : Responsable de la comunicación con los protocolos superiores.

Modelo OSI (Ethernet)	**IEE (802.x)**
Enlace (nivel 2)	LLC
	MAC

La propuesta es muy coherente, pues facilita esta compleja actividad característica de las LAN. Pero desde ya, que esta propuesta no es reconocida por OSI, marcando una diferencia entre estos dos protocolos. Aparecen aquí estos dos estándares de mercado, que se recalca "NO SON IGUALES", si bien son muy parecidos. En el caso de CSMA/CD, que es el que interesa en este

texto, **todo hardware y software de red soporta ambos protocolos y acorde a la trama que se trate aplica uno u otro**.

La diferencia más importante se encuentra en dos octetos del encabezado (que se tratarán a continuación). Cuando se trata de tramas IEEE 802.3, el encabezado MAC tendrá siempre encima de él el subnivel LLC, por esta razón no necesita definir a quién le debe entregar los datos, pues solo existe una opción (LLC); en esta situación los dos octetos referidos establecen la longitud del campo de datos y se llaman "**Length**". Cuando la trama es Ethernet (el nivel de enlace de OSI, no se encuentra subdividido) se debe aclarar a qué protocolo entregará los datos en el nivel 3 (Red), por ejemplo IPX, IP, etc. en este caso estos dos octetos se denominan "**Ethertype**", y se emplean justamente para definir qué tipo de protocolo se encuentra arriba de Ethernet.

La forma de distinguir de qué trama se trata es mediante el valor en hexadecimal de estos dos octetos: todo valor inferior a 0550h se corresponde a una trama IEEE-802.3; por encima de este se trata de una trama Ethernet.

4.1.3. Algoritmo de disminución exponencial binaria:

Como se mencionó en la introducción, al producirse una colisión, los ETD responsables de la misma dejan de transmitir (en realidad se envía una señal de atasco para avisar a todos los ETD de la red de este hecho). Automáticamente estos equipos generan un número aleatorio entre 0 y 1. Este número es motivado por el algoritmo de disminución exponencial binaria que propone generar un número aleatorio acorde a la siguiente fórmula:

$$\textbf{Nro Rand} = 2^{n} - 1$$

n = cantidad de colisiones detectadas en esta lucha.

Al tratarse de la primera colisión: Nº Rand=2^1-1 = 1 => (Nro Random entre 0 y 1).

Este valor (0 ó 1) establece la cantidad de tiempos de ranura que esperará el ETD para volver a transmitir la trama que ocasionó la colisión, siendo el tiempo de ranura **51,2 µs**.

Si los dos ETD generan el mismo valor, colisionarán nuevamente, pero si obtienen valores diferentes, uno de los dos emitirá primero, y cuando pasen los 51,2 µs del segundo ETD y este desee transmitir, encontrará el canal ocupado y no podrá hacerlo (es decir que el primero ganó la compulsa).

Si hubiesen generado el mismo valor, es decir: los 2 ETD =1 ó los 2 ETD = 0, se producirá la segunda colisión, por lo tanto:

- Nro Rand = 2^2 -1 = 3 => (Nro Random entre 0, 1, 2 ó 3)

Si ambos equipos obtuvieran el mismo valor, colisionarían nuevamente y entonces sería:

- Nro Rand = 2^3 -1 = 8 => (Nro Random entre 0, 1, 2, 3 ,4, 5, 6, 7 u 8).

Si siguieran generando iguales números, esta situación se mantendría hasta:

- Nro Rand = 2^{10} -1 = 1023 => (Nro Random entre 0 y 1023).

Si aún así esto continuara, se ejecutaría el mismo algoritmo con exponente = 10, durante seis veces más, y luego se comienza nuevamente.

Esto que parece muy poco probable, si bien es poco frecuente, no es tan así, pues se debe tener en cuenta que en una red donde existen varios ETD conectados a un mismo **dominio de colisión**, en cualquier momento de esta secuencia, puede entrar en juego otro ETD, caso en el cual, este último comenzaría a tratar el algoritmo como su primera colisión, y los anteriores seguirían con su rutina de disminución de probabilidades, y así también puede ingresar un cuarto, quinto, etc.

El último concepto que aún queda por tratar es el de **tiempo de ranura** (51,2 µs). Este valor nace de la definición misma de Ethernet, y aparece en los inicios de este protocolo, cuando estas redes se implementaban con topología Bus sobre cable coaxial grueso, con el cual se podían unir hasta cinco segmentos de 500m a través de cuatro repetidores regenerativos (y sólo 3 de ellos cargados; se la conocía como norma 5-4-3). La distancia máxima que alcanzaba esta red era de 2.500m. Teniendo en cuenta el tiempo de latencia de los repetidores, una señal eléctrica tardaba en recorrer esta distancia ida y vuelta, aproximadamente este tiempo: 51,2 µs. El tema de fondo aquí radica en que si se tiene en cuenta dos ETD separados a esta distancia (el peor de los casos), suponiendo que uno de ellos comienza la transmisión, y un instante antes de llegar al segundo, éste escucha el canal y por estar desocupado, comienza a transmitir; entonces se producirá una colisión muy próxima al segundo ETD. El que inició la transmisión tomará consciencia de la colisión, cuando este estado anormal de tensión regrese a él, es decir, cuando haya recorrido los 2500m de ida y los 2500m de vuelta, que coincide con estos 51,2 µs. Si se supone que el segundo ETD no inició ninguna transmisión, al cabo de estos 51,2 µs ningún ETD de esta red podría transmitir, pues al escuchar el medio, lo encontraría ocupado. Esto se llama **Apropiarse del canal.**

Basado en estos conceptos es que se define que el tamaño mínimo de una trama Ethernet no puede ser menor de 64 Byte, pues **64 Byte = 512 bit** y **512 bit transmitidos a 10.000.000 de bit por segundo (10 Mbps) = 51,2 µs**. También se define que no podrá tener más de 1518 Byte, para evitar que el apropiado del canal sea eterno, evitando así monopolios del medio.

4.1.4. Armado de tramas:

Las tramas Ethernet son armadas en el subnivel MAC y responden a 14 octetos de encabezado y a 4 octetos de cola que es donde se realiza el CRC y entre estos campos van los datos.

Se debe tener en cuenta que para que todos los ETD de la red se sincronicen y sepan que se está por recibir una trama, antes de la misma se envían 7 octetos de preámbulo (10101010) y luego un octeto de inicio (10101011). Algunos autores lo consideran parte del encabezado Ethernet y otros no, en este texto no se considerarán parte del mismo.

El formato de una trama Ethernet es el que se detalla a continuación:

8	7	6	5	4	3	2	1
Dirección destino (6 Byte)							
Dirección Origen (6Byte)							
Tipo o longitud (2 Byte)							
Datos (Variable entre 46 y 1500 Byte)							
CRC (4 Byte)							

- Dirección destino: Especifica la dirección del host a alcanzar a nivel MAC.
- Dirección origen: Especifica la propia dirección a nivel MAC.
- Tipo o longitud: Si se trata del protocolo Ethernet el tipo de protocolo de nivel superior (Ethertype). Si es protocolo 802.3 especifica la longitud del campo de datos
- CRC: Control de redundancia cíclica, emplea el concepto de polinomio generador como divisor de la totalidad de la trama, el resto de esta operación se enmascara con una secuencia d-determinada de bit y se envía en este campo. Se trata entonces de una división binaria, en la cual se emplea como polinomio generador justamente el CRC-32, que figura abajo, por lo tanto el resto de esta división SIEMPRE será una secuencia de bit de longitud inferior a 32 bits, que será lo que se incluye en este campo. Los formatos estandarizados de estos CRCs son los que se presentan a continuación:
  - **CRC-12**: $X^{12} + X^{11} + X^3 + X^2 + X + 1$
  - **CRC-16**: $X^{16} + X^{15} + X^2 + 1$
  - **CRC CCITT V41**: $X^{16} + X^{12} + X^5 + 1$ (este código se utiliza en el procedimiento *HDLC*)
  - **CRC-32 (Ethernet)**: = $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
  - **CRC ARPA**: $X^{24} + X^{23} + X^{17} + X^{16} + X^{15} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^5 + X^3 + 1$
- Preámbulo: No está representado en la gráfica anterior, pues no es considerado como parte de la trama pero se trata de 7 byte que indican que comienza una trama y permite sincronizar relojes y 1 Byte de inicio.

4.1.5. Relación de Ethernet con Hub y Switch:

Las redes Ethernet fueron diseñadas con topología "**Bus físico**", y posteriormente se incorpora la idea de interconectar varios ETD a un solo dispositivo, que en definitiva cumplía con las mismas funciones de un conector "T" del tipo BNC, pero en vez de bifurcar la señal por dos caminos (como lo hace una "T"), lo hace por "n" caminos. Esto da origen a los Hub, los cuales literalmente "Explotan" la señal recibida por cualquiera de sus puertos por todos los restantes (menos el que ingresó). La nueva forma que adoptan estas redes es más parecida a una **estrella**, pero como la lógica sigue siendo la misma, es decir, que la señal que emite un ETD sea escuchada por todos los conectados, es que también se le suele llamar "**Bus lógico**". Más adelante estos Hubs incorporan más funciones, de las cuales la más importante es la de regenerara la señal, actuando como un repetidor regenerativo de múltiples puertos.

Teniendo en cuenta lo tratado anteriormente sobre las colisiones, es muy lógico pensar que a medida que aumenta el número de ETD, aumentan las colisiones. Esta es una realidad en estas redes, si bien no depende directamente de la cantidad de host conectados, sino más bien del tipo de tráfico que estos generen. No se trata de una ley matemática ni de algo fácilmente calculable, estos datos se obtienen más bien mediante mediciones de tráfico de red (análisis de tráfico).

Lo que se debe tener en cuenta es que a medida que se necesitan más puestos de trabajo, se pueden agregar más Hubs e interconectarlos entre ellos, esta es una de las grandes ventajas que posee esta red: su flexibilidad. Cada nuevo puesto incorporado generará más y más colisiones, produciendo una baja paulatina en el rendimiento general de la red. Recordar aquí que la única función de un Hub es explotar la señal. Por lo tanto si se poseen n Hubs, y un ETD emite una trama, esta será explotada por todas las bocas de los n Hubs.

Al detectar esta baja performance (mediante análisis de tráfico), la primera medida es segmentar la red en dominios de colisión. Esta actividad la lleva a cabo un dispositivo que trabaja a nivel 2 del modelo OSI, denominado Switch, el cual, no se describirá en este texto, pero básicamente posee tablas dinámicas por cada uno de sus puertos, donde va almacenando las direcciones MAC fuente de cada trama que pasa por él, y a medida que va "aprendiendo" las mismas, comienza a poder conmutar una trama acorde a la puerta en la que la tiene almacenada. Con este principio, si dos ETD se encuentran en la misma puerta, y un Switch recibe una trama Ethernet con MAC origen y destino en ese segmento de red, no lo reenviará por ningún otro puerto; si recibiera una trama con destino en otro segmento de red, únicamente lo conmutaría por el puerto en el que tiene almacenado ese segmento. Como se puede apreciar, si dialogan dos ETD de un mismo segmento, esto no inhabilita a hacerlo a otros dos de otro segmento, cosa que no se podría lograr con Hubs pues estos explotarían la señal por todas sus bocas y el postulado rector de esta metodología es que si se escucha ruido en el canal no se puede transmitir.

El objetivo entonces de un Switch es armar diferentes dominios de colisión, posibilitando que más de un ETD pueda transmitir a la vez.

La gran salvedad que se debe analizar aquí es que si el Switch no posee elementos de juicio para poder determinar que hacer con una trama, **opera exactamente igual que un Hub**, es decir, explota la señal por todas sus bocas. Esto es de vital importancia si se tiene en cuenta que una red mal diseñada (y en la gran mayoría de los casos, por falta de optimización de tráfico) genera una enorme cantidad de Broadcast a nivel MAC. Si se estudia este detalle, es fácil deducir ¿qué hará el Switch al recibir una trama con dirección destino Broadcast?..... Lo explotará por

todas sus bocas igual que un Hub. Por lo tanto en el análisis de tráfico es trascendente prestar atención a la generación de Broadcast que se produce en una red. Más adelante se seguirá analizando este tipo de tráfico (y también el Multicast), pues se produce también en otros niveles con igual impacto.

4.1.6. Actualizaciones de Ethernet:

Año a año van apareciendo nuevas técnicas para aumentar la velocidad y capacidades de este increíble protocolo que ronda ya en los cuarenta años de vida y parece ser el mayor superviviente de la historia de las telecomunicaciones. A continuación se presentan una serie de conceptos que han surgido en estos últimos tiempos:

- Fast y Giga Ethernet:

Las redes día a día van exigiendo un mayor ancho de banda. En la actualidad las necesidades de voz e imágenes hacen que los 10Mbps de Ethernet sean insuficientes. Para dar solución a este problema se comienzan a estudiar nuevas opciones dando origen a Fast Ethernet.

Se plantearon inicialmente dos propuestas:

- Mantener el protocolo CSMA/CD en todos sus aspectos, pero aumentar en un factor 10 la velocidad de la red. Al mantener el tamaño de trama mínimo (64 bytes) se reduce en diez veces el tamaño máximo de la red, lo cual da un diámetro máximo de unos 400 metros. El uso de CSMA/CD supone la ya conocida pérdida de eficiencia debida a las colisiones.
- Aprovechar la revisión para crear un nuevo protocolo MAC sin colisiones más eficiente y con más funcionalidades (más parecido en cierto modo a Token Ring), pero manteniendo la misma estructura de trama de Ethernet.

La primera propuesta tenía la ventaja de acelerar el proceso de estandarización y el desarrollo de productos, mientras que la segunda era técnicamente superior. El subcomité 802.3 decidió finalmente adoptar la primera propuesta, que siguió su camino hasta convertirse en lo que hoy conocemos como **Fast Ethernet**, aprobado en junio de 1995 como el suplemento 802.3u a la norma ya existente.

Los objetivos fundamentales son:

- Mantener el CSMA/CD.
- Soportar los esquemas populares de cableado. (Ej. 10BaseT).
- Asegurar que la tecnología Fast Ethernet no requerirá cambios en los protocolos de las capas superiores, ni en el software que corre en las estaciones de trabajo LAN.

Para acelerar el proceso se utilizó para el nivel físico buena parte de las especificaciones ya desarrolladas por ANSI para FDDI. Los medios físicos soportados por Fast Ethernet son fibra óptica multimodo o monomodo, cable UTP categoría 3 y categoría 5 y cable STP (Shielded Twisted Pair).

Los partidarios de la segunda propuesta, considerando que sus ideas podían tener cierto interés, decidieron crear otro subcomité del IEEE, el **802.12**, que desarrolló la red conocida

como **100VG-AnyLAN**. Durante cierto tiempo hubo competencia entre ambas redes por conseguir cota de mercado; hoy en día la balanza se decanta ampliamente hacia Fast Ethernet.

Giga Ethernet se aprueba por IEEE en el año 1998 como estándar **802.3Z** (zeta, por ser la última letra del alfabeto y pensar que será la última de esta familia....). También se lo conoce hoy como 1000 Base-X.

Para su implementación sobre pares de cobre, se creó la norma 802.3ab, que define el funcionamiento de este protocolo sobre cables UTP (Unshielded Twisted Pair) categorías 5, 5e o 6 y por supuesto para fibra óptica, de esta forma pasó a llamarse 1000 base-T.

En el 2002, IEEE ratificó una nueva evolución de este estándar para operar a 10 Gbps como **802.3ae**, funcionando sobre fibra óptica, pero ya existe la propuesta para cables de cobre. Mantiene aún la filosofía CSMA/CD (Carrier Sense Múltiple Access / Colision detection).

- La identificación según IEEE:

Para Fast Ethernet:

100: indica la velocidad de transmisión, 100 Mbps

BASE: tipo de señalización, baseband, sobre el medio sólo hay señales Ethernet

El tercer campo: indica el tipo de segmento

T: ("T" de twisted pair)

TX: usa dos pares de cable par trenzado para datos (ANSI X3T9.5)

T4: usa cuatro pares de cable par trenzado para teléfonía (El estándar T4 fue desarrollado para que cableados de menor calidad pudiesen utilizar Fast Ethernet)

Para Giga Ethernet:

1000: indica la velocidad de transmisión, 1000 Mbps

BASE: tipo de señalización, baseband, sobre el medio sólo hay señales Ethernet

El tercer campo: indica el tipo de segmento

LX: ("L" de long wavelength)

SX: ("S" de short wavelength)

T: ("T" de twisted pair) (Usa los 8 hilos del cable UTP.)

- Máximas frecuencias de los diferentes cables según categorías:

5: (Cable sólido de pares trenzados), 22 o 24 AWG (0,643mm o 0,511mm), 100 Mhz.

5e: (Categoría 5 mejorada), 26 AWG (0,409mm), 100 Mhz, UTP

6: (Cable sólido de pares trenzados), 24 AWG (0,511mm), 300 Mhz, FTP

7: (Cable sólido de pares trenzados apantallados por par),23 AWG (~0,600mm), 600 Mhz, STP

- El ancho de banda del cable en Ethernet y la ley de Shannon:

La principal dificultad con 100 Mbps es que los datos a alta frecuencia no se propagan sobre par trenzado o fibra (requeriría una forma de onda de 200 MHz si codificara con Manchester). UTP categoría 5 está hecho para soportar una frecuencia de 100 MHz.

En 1927, Nyquist determinó que el número de pulsos independientes que podían pasar a través de un canal de telégrafo, por unidad de tiempo, estaba limitado a dos veces el ancho de banda del canal: fp ≤ **2* Δf** (**fp**:frecuencia de pulsos, **Δf**: Ancho de banda). Es decir que para obtener 100 Mbps, al menos debería tener una frecuencia de 200 MHz, con transmisión de dos niveles. Por ahora nos quedaremos con esta definición, dejando a continuación la Ley de Shanon para quien desee profundizar en el tema.

Ley de Shannon:

Permite calcular la velocidad teórica máxima en la cual dígitos libres de error pueden ser transmitidos sobre un canal con ancho banda de limitado en presencia de ruido: $C= \Delta f * \log_2(1+S/N)$. (C: Capacidad del canal en bits por segundo, Δf: Ancho de banda en Hertz y S/N: relación señal-ruido).

Lo importante a destacar es que se hizo necesario recurrir a otras técnicas para poder llegar a tasas de transmisión de 100Mbps en cables categoría 5, como se verá más adelante.

- NRZI, MLT-3 y la codificación 4B5B

Nuevas formas de codificación de la forma de onda han sido implementadas para Fast Ethernet. Para reducir aun más los requerimientos de frecuencia sobre UTP, **100BaseTX** agrega una variación a **NRZI** (Non-Return-to-Zero, Invert-on-one) llamada **MLT-3** (Multiple Level Transition - 3 Niveles) o **NRZI-3**.

Cuando la información es una secuencia de ceros, en NRZI y MLT-3 se puede perder la codificación de la señal del reloj. Para resolver este problema se utiliza la codificación de bloque (block encoding) **4B5B** (la misma utilizada por FDDI).

Un código de bloque toma un bloque o grupo de bits y los "traduce" a un conjunto de código bits más grande. 4B5B toma cuatro bits y los traduce a cinco bits

Los códigos de bloque se diseñan para mejorar la señalización de línea al balancear los ceros y los unos transmitidos

- 100BaseTX: Uso del medio

100 Base-TX, UTP categoría 5

Hilo 1: T+; Hilo 2: T-, Hilo 3: R+ e Hilo 6: R-

Máximo 100 metros, conector RJ-45

Un cable cruzado se construye igual que en 10Mbps (1 con 3 y 2 con 6)

TX: usa dos pares de cable par trenzado para datos (ANSI X3T9.5)

FX: usa fibra óptica (ANSI X3T9.5) y usa dos hilos de fibra

TX y FX se conocen también como 100Base-X

Cada uno de los diferentes medios utiliza un tipo de codificación (códigos de bloque) y señalización de línea diferente:

100BaseFX utiliza codificación 4B/5B y señalización NRZI

100BaseTX utiliza codificación 4B/5B y señalización MLT-3 (ó NRZI-3)

- Giga Ethernet

Codificación de la señal para representar los datos

En gigabit se utilizan las mismas técnicas de señalización utilizadas en el canal de fibra y se han adaptado y extendido las utilizadas en Fast Ethernet.

1000Base-T utiliza un esquema de codificación de bloque llamado **4D-PAM5** que transmite utilizando los 8 hilos del cable UTP. Este esquema "traduce" 8 bits de datos a cuatro símbolos (4D) que serán transmitidos simultáneamente, uno sobre cada par.

Estos símbolos son enviados sobre el medio utilizando señales moduladas por amplitud de pulso de 5 niveles (**PAM5**).

Estos 5 símbolos son conocidos como -2, -1, 0, +1, +2 (+/- 2 realmente son +/-1V, y +/-1 es to +/- 0.5V)

- Estándares:

10Mbps

10 BASE-T

Funciona sobre cuatro alambres (dos pares trenzados) en un cable de Categoría 3 o de Categoría 5. Un Hub o un switch activo están en el medio y tiene un puerto para cada nodo. Ésta es también la configuración usada para el Ethernet 100BASE-T y gigabit. con señalización de codificación Manchester, cableado de par trenzado de cobre, topología de estrella - evolución directa del 1BASE-5.

100 mbps (Fast Ethernet)

100 BASE-T

Un término para cualquiera de los tres estándares de Ethernet de 100 Mbit/s sobre cable de par trenzado. Incluye 100BASE-TX, 100BASE-T4 y 100BASE-T2. A fecha de 2009, 100BASE-TX ha dominado totalmente el mercado, y con frecuencia es considerado ser sinónimo con 100BASE-T en el uso informal. Todos utilizan una topología de estrella.

100BASE-TX

Señalización codificada 4B5B MLT-3, cableado de cobre Categoría 5 con dos pares trenzados.

1000 Mbps (Giga Ethernet)

802.3z (1998), 1000BASE-X, Ethernet de 1 Gbit/s sobre fibra óptica.

802.3ab (1999), 1000BASE-T, Ethernet de 1 Gbit/s sobre par trenzado no blindado.

802.3an (2006), 10GBASE-T, Ethernet a 10 Gbit/s sobre par trenzado no blindado (UTP).

802.3bg (Borrador) 40Gb/s Ethernet Single-mode Fibre PMD Task Force.

802.3 (Borrador) 100Gb/s Ethernet Electrical Backplane and Twinaxial Copper Cable Assemblies Study Group.

Aunque no se trata de Giga Ethernet merece la pena mencionar también a:

802.3af (2003) Alimentación sobre Ethernet (PoE).

Toda la información actualizada sobre 802.3 está en: http://www.ieee802.org/3/

Muchos adaptadores de Ethernet y puertos de switches soportan múltiples velocidades, usando autonegociación para ajustar la velocidad y la modalidad dúplex para los mejores valores soportados por ambos dispositivos conectados. Si la auto-negociación falla, un dispositivo de múltiple velocidad detectará la velocidad usada por su socio, pero asumirá semidúplex. Un puerto Ethernet 10/100 soporta10BASE-T y 100BASE-TX. Un puerto Ethernet 10/100/1000 soporta 10BASE-T, 100BASE-TX, y 1000BASE-T.

- Otros datos de interés:

Los datos que no viajan en el vacío, circulan más despacio que la luz en el vacío.

C (velocidad de la luz en el vacio): 300.000 Km/s.

Coaxial grueso: 77% C (231.000 Km/s).

Coaxial delgado: 65% C (195.000 Km/s).

Par trenzado: 59% C (177.000 Km/s).

Fibra óptica: 66% C (198.000 Km/s).

¿Qué tan largo es un bit en 10 Mbps?

Coaxial grueso: 231.000 Km/s dividido en 10 millones de bits por segundo = 23.1 metros.

Coaxial delgado: 195.000 Km/s dividido en 10 millones de bits por segundo = 19.5 metros.

Par trenzado: 177.000 Km/s dividido en 10 millones de bits por segundo = 17.7 metros.

Fibra óptica: 198.000 Km/s dividido en 10 millones de bits por segundo = 19.8 metros.

Niveles de degradación

A lo largo de todo el desarrollo del tema Ethernet, se ha presentado con todo el detalle posible el proceso de la lógica CSMA/CD, pues se considera fundamental el estricto cumplimiento de lo que establecen los estándares, cuando no se respetan estos aspectos y la "lógica" de CSMA/CD falla (*por ejemplo por exceso de HUBs o Switchs, o por no respetar distancias máximas*), se comienzan a producir "colisiones tardías", las cuales el nivel 2 no sabe tratar, y deberán ser detectadas y corregidas por los niveles superiores (si es que están en capacidad de hacerlo), en esos casos el rendimiento de toda la red se ve afectado en órdenes de magnitud catastróficos. Para hacernos una idea de cuánto puede sufrir estos fallos, a continuación se presentan algunos valores:

- La **retransmisión** a nivel Ethernet ocurre, normalmente, dentro de tiempos del orden de cientos de microsegundos.
- Las restransmisiones en la subcapa LLC puede ocurrir en milisegundos.
- En la capa de transporte (capa 4) las restramisiones puesen tomar segundos.
- Las aplicaciones pueden esperar minutos.

4.1.7. Spoof de direcciones MAC:

Cuando se trabaja en entornos LAN, el nivel de enlace toma como identificador de un ETD su dirección MAC, la cual por encontrarse impresa en la tarjeta de red (y en teoría ser única en el mundo), inicialmente debería ser difícil de falsificar. La realidad hace que no sea tan difícil, e incluso el propio SO Linux permite modificarla a través del comando ifconfig. Cuando la información es recibida en una red LAN, el primer identificador de direcciones que aparece es esta dirección, y en base a esta, el ETD decide si la entrega al nivel de red o no. Por ser la puerta de entrada a un host destino, se ha trabajado mucho por distintas opciones de engaño, cualquiera de ellas son lo que se denominó MAC spoofing, lo cual implica falsificar una dirección MAC. En los ejercicios de este capítulo se presentan varias opciones de trabajo con este tema.

4.2. Presentación (Los estándares 802.11):

4.2.1. **WiFi** (Wireless Fidelity)

En realidad, WiFi es un nombre comercial desarrollado por un grupo de comercio industrial llamado WiFi Alliance (Inicalmente: 3Com – Aironet [hoy parte de CISCO] – Harris – Lucent – Nokia y Symbol technologies, hoy más de 150 miembros), el nombre "oficial" de esta alianza es **WECA** (Wireless Ethernet Compatibility Alliance) y son los primeros responsables de 802.11b.

WiFi describe los productos de WLAN basados en los estándares 802.11 y está pensado en forma más "amigable" que la presentación eminentemente técnica que ofrece IEEE. Se podría llegar a discutir si cubre o no todo lo que ofrece 802.11 o no, pues alguno de ellos podría ser puesto en duda, pero a los efectos de este texto, se hará más referencia a lo que establece 802.11, sin detenerse en estas diferencias.

La web de esta alianza es: www.wi-fi.org

En esas web se puede también consultar el estado "on line" de los productos que se encuentran certificados, el path completo de esta consulta es:

http://www.wi-fi.org/certified_products.php

El estándar **802.11** de IEEE se publica en junio 1997, luego de seis años de proceso de creación. Propone velocidades de 1 y 2Mbps y un rudimentario sistema de cifrado (el **WEP**:Wired Equivalent Privacy), opera en 2,4 GHz con RF e IR. Aunque WEP aún se sigue empleando, ha sido totalmente desacreditado como protocolo seguro.

En septiembre de 1999 salen a la luz el estándar **802.11b** que ofrece 11Mbps y el **802.11ª** que ofrece 54 Mbps, si bien los productos de la primera aparecieron en el mercado mucho antes.

En enero de 2004 aparece publicado el estándar 802.11n, ofreciendo una alternativa para alcanzar hasta 600Mbps como velocidad real de transmisión de datos. Se viene implantando desde el año 2008. A diferencia de 802.11b y 802.11g, éste permite operar en las dos bandas de frecuencias (2,4 GHz y 5Ghz) y gracias a esta novedad es que se hace compatible con equipamiento antiguo que no soporte uno u otro. La novedad también es que al abrir compatibilidad con la frecuencia de 5GHz, ofrece libertad de acción sobre la saturada banda de 2,4GHz que desde la acelerada explosión de WiFi está contaminada de dispositivos.

Existen muchas más variantes de esta familia, pero en este texto nos centraremos en las más empleadas.

Modo turbo: Algunos fabricantes ofrece velocidades superiores a lo que el estándar define. Estos procesos lo logran mediante la "vinculación de canales", es decir, dos canales son multiplexados juntos empleando el total de velocidad de la suma de ambos. Esto si bien es favorable aparentemente, tiene las desventajas de no respetar el estándar y de sacrificar la mitad de los canales de 802.11ª.

La familia 802.11, hoy se encuentra compuesta por los siguientes estándares:

- **802.11ª:** (5,1-5,2 Ghz, 5,2-5,3 Ghz, 5,7-5,8 GHz), 54 Mbps. OFDM: Multiplexación por división de frecuencias ortogonal
- **802.11b:** (2,4-2,485 GHz), 11 Mbps.
- 802.11c: Define características de AP como Bridges.

- 802.11d: Múltiples dominios reguladores (restricciones de países al uso de determinadas frecuencias).
- 802.11e: Calidad de servicio (QoS).
- 802.11f: Protocolo de conexión entre puntos de acceso (AP), protocolo IAPP: Inter Access Point Protocol.
- **802.11g**: (2,4-2,485 GHz), 36 o 54 Mbps. OFDM: Multiplexación por división de frecuencias ortogonal. Aprobado en 2003 para dar mayor velocidad con cierto grado de compatibilidad a equipamiento 802.11b.
- 802.11h: DFS: Dynamic Frequency Selection, habilita una cierta coexistencia con HiperLAN y regula también la potencia de difusión.
- **802.11i:** Seguridad (aprobada en Julio de 2004).
- 802.11j: Permitiría armonización entre IEEE (802.11), ETSI (HiperLAN2) y ARIB (HISWANa).
- 802.11k: Mejora la gestión de las redes WLAN.
- 802.11m: Mantenimiento redes wireless.
- 802.11n: Velocidad de 600Mbps pudiendo emplear 2,4 o 5 GHz.
- 802.11p: Frecuencia de 5,9GHz indicado para automóviles.
- 802.11r: Pensado para conmutación rápida y segura entre puntos de acceso.
- 802.11s: Interoperatibilidad entre fabricantes.
- 802.11v: Configuración remota de dispositivos cliente.
- 802.11w: Mejora en la capa de control de acceso al medio en cuanto su autenticación y codificación.
- 802.11y: Permite operar (con ciertas restricciones de Países y zonas) en la banda de 3,65 a 3,7 GHz.

Quizás el tema más importante a destacar es la posibilidad de expansión de 802.11. El incremento constante de mayores velocidades, hace que los 11 Mbps de 802.11b ya queden pequeños. La migración natural es hacia 802.11g, pues sigue manteniendo la frecuencia de 2,4GHz, por lo tanto durante cualquier trancisión en la que deban convivir, ambos estándares lo permiten. En cambio si se comienzan a instalar dispositivos 802.11a, los mismos no permiten ningún tipo de compatibilidad con 802.11b, pues operan en la banda de 5 GHz, si bien ahora tenemos esperanza con 802.11n.

Para acotar únicamente el tema de seguridad, se tratarán sólo 802.11ª, b, g, n y 802.11i.

Hoy en día se puede decir que existen cuatro estándares de WLAN:

- **HomeRF:** Es una inciativa lanzada por Promix, principalmente en EEUU y orientada exclusivamente al mercado residencial. Tiene sus bases en los estándares de teléfono digital inalámbrico mejorado (DECT)
- **BlueTooth:** Lo inició IBM, orientado al mercado comercial/ventas, y a la interconectividad de elementos de hardware. En realidad no compite con 802.11,

pues tiene la intención de ser una estándar con alcance nominal de 1 a 3 metros y a su vez no supera los 1,5 Mbps

- **802.11:** Cubre todo el espectro empresarial.
- **802.**16: WiMAX (Worldwide Interoperability for Microwave Access - Interoperabilidad mundial para acceso por microondas). Pensado para solucionar el problema de la "última milla" en zonas de difícil acceso.

Una iniciativa que se debe mencionar también es **HiperLAN** en sus versiones 1 y 2. Se trata de una verdadera analogía inalámbrica para ATM. Fue un competidor de 802.11 que opera en la frecuencia de 5 GHz y gozó del apoyo de compañías como Ericsson, Motorola, Nokia; Panasonic y Sony llegaron a crear regulaciones por parte de ETSI al respecto, pero no se logró imponer y hoy en día está prácticamente en desuso. En lo particular me hace acordar mucho a la batalla que hubo entre ATM y Ethernet (Fast Ethernet, Giga Ethernet....).

Definiciones a tener en cuenta en este apartado:

- **Access control:** Es la prevención del uso no autorizado de recursos.
- **Access Point** (**AP**): Cualquier entidad que tiene funcionalidad de estación y provee acceso a servicios de distribución vía **wireless medium** (**WM**) para estaciones asociadas.
- **Ad Hoc network**: red wireless compuesta únicamente por estaciones con iguales derechos.
- **Portal**: punto lógico desde el cual se conecta una red wireless con una no wireless.
- **Station** (**STA**):cualquier dispositivo que cumple con un nivel MAC conforme a 802.11 y un nivel físico que posee una interfaz wireless.
- **Portable station**: estación que puede ser movida de ubicación, pero que sólo puede Tx o Rx en estado fijo.
- **Movile station**: Estación que permite Tx o Rx en movimiento.

El tema de seguridad no puede ser tratado con seriedad si previamente no se tienen en cuenta varios conceptos de BASE que hacen a la arquitectura WiFi. Una vez comprendido el funcionamiento básico de esta infraestructura, recién entonces se puede hablar de seguridad. En virtud de este concepto, es que en este texto se trata inicialmente una serie de conceptos básicos, para luego profundizar en los aspectos de seguridad WiFi.

4.2.2. Modelo de capas de 802.11.

4.2.2.1. La capa física de 802.11: (El detalle de la misma se puede ver al final de este punto)

La capa física la componen dos subcapas:

- PLCP (Physical Layer Convergence Protocol): Se encarga de codificación y modulación.
  - Preámbulo (144 bits = 128 sincronismo + 16 inicio trama).
  - HEC (Header Error Control): CRC 32.
  - Modulación (propagación) DSSS o FHSS o IR.
- PMD (Physical Medium Dependence): Es la que crea la interfaz y controla la comunicación hacia la capa MAC (a través del SAP: Service Access Point)

Este nivel lo conforman dos elementos principales:

- **Radio**: Recibe y genera la señal.
- **Antena:** Existe una gran variedad y no será tratado en este texto.

El estándar 802.11 define en el punto 12 del mismo todas las especificaciones de servicio para este nivel, las cuales no serán tratadas en este texto. Hay algunos aspectos físicos que vale la pena profundizar para la comprensión de WiFi, de los cuales se recomienda especialmente:

- FHSS (Frequency Hopping Spread Spectrum) para la banda de 2,4 GHz (ISM: Industrial, Scientific and Medical band) en el punto 14 de la recomendación.
- DSSS (Direct Sequence Spread Spectrum) para 2,4 GHz, en el punto 15.
- IR (InfraRed), en el punto 16.

NOTA: Aunque esto no forma parte de los conceptos de WiFi, cuando se habla de transmisión, se deben diferenciar tres palabras:

- Modulación: Es el método de emplear una señal portadora y una moduladora (que da forma a la anterior). Cada una de ellas puede ser analógica o digital, con lo cual se obtienen cuatro posibles combinaciones de portadora y moduladora (AA, AD, DA y DD), con las cuales se conforman todas las técnicas de modulación. WiFi en la mayoría de los casos emplea la técnica QAM (Modulación en cuadratura de Fases con más de un nivel de amplitud).
- Propagación: Es la forma en la cual "van saliendo" las señales al aire. Aquí es donde verdaderamente se aplican las técnicas de DHSS y FHSS. SS (Spread Spectrum) es la técnica de emplear muchas subportadoras de muy baja potencia con lo cual se "expande" el espectro útil. En cuanto a DH y FH, el ejemplo típico que se emplea para estas técnicas es la analogía con una terminal de trenes, en la cual existen varios andenes. Para DH, los trenes estarían saliendo, primero el andén 1, luego el 2, a continuación el 3, 4, 5... y así sucesivamente, respetando siempre este orden. Para FH, la salida de los trenes no respeta el orden y puede ser aleatoria o acorde a un patrón

determinado (WiFi hace un muy buen uso de esto, pues en las subportadoras en las cuáles recibe mucha interferencia deja de usarlas, o emplea menos cantidad de bits en las mismas).

- Codificación: Es la asociación de bit a cada "muestra" que se obtiene. WiFi en la mayoría de los casos emplea el código Barker.

4.2.2.2. Descripción más detallada del nivel Físico y Subnivel MAC de 802.11.

Se presenta a continuación un breve resumen de los aspectos más importantes del nivel físico y subnivel MAC del estándar. Se ha respetado la puntuación del mismo para quien desee profundizar en esos puntos.

5.3. Servicios Lógicos: 802.11 propone dos categorías de servicios empleados en el subnivel MAC:

- Station Service (SS): Son los servicios específicos de las STAs.
- Distribution System Service: Estos servicios se emplean para pasar en cualquier sentido entre DS y BSS.

Los servicios determinarán distintos tipos de mensajes que fluirán por la red, independientemente de su categoría. La totalidad de los servicios (y/o mensajes) son:

a. Authentication: A diferencia de una red cableada, en 802.11 no existe una seguridad a nivel físico para prevenir el acceso no autorizado, por lo tanto este estándar ofrece la capacidad de autenticación por medio de este servicio. Si entre dos estaciones no se establece un adecuado nivel de autenticación, la asociación no podrá ser establecida. 802.11 soporta dos metodologías de autenticación:
   - Open System Authentication (OSA): Cualquier STA puede ser autenticada.
   - Shared Key Authentication: Este mecanismo requiere la implementación de Wireless Equivalent Privacy (WEP) y será tratado más adelante.

b. Deauthentication: Este servicio es invocado si una autenticación debe ser finalizada. Se trata de una notificación, no una solicitud, por lo tanto no puede ser rechazada, y puede ser invocado tanto por una STA (No AP), como por un AP.

c. Association: Antes que una STA pueda enviar mensajes vía un AP, la misma deberá encontrarse asociada a este último. Este servicio permite al DS conectar distintas STA dentro de una LAN Wireless, ubicando a cada una de ellas. En cualquier instante de tiempo, una STA sólo podrá estar asociada a un único AP. Este servicio es siempre iniciado por una STA no AP, nunca por un AP.

d. Deassociation: Este servicio es invocado si una asociación debe ser finalizada. Se trata de una notificación, no una solicitud, por lo tanto no puede ser rechazada, y puede ser invocado tanto por una STA (No AP), como por un AP.

e. Reassociation: Permite cambiar una asociación de un AP a otro, o también cambiar los parámetros de asociación de una STA con el mismo AP.

f. Distribution: Este tipo de mensajes se producen al ingresar información a un DS proveniente de un BSS. El encargado de generar estos mensajes será un AP y su objetivo es alcanzar el destino buscado.

g. Integration: Los mensajes que van o vienen dirigidos hacia/desde un portal, harán uso de este servicio.

h. Privacy: 802.11 al igual que sucede con autenticación (y por las mismas causas) provee la posibilidad de cifrar el contenido de los mensajes a través de este servicio. Este servicio que es opcional, también se lleva a cabo por WEP. Es muy discutible la solidez del mismo, pero la decisión fue tomada como una medida que permite tener un nivel de seguridad "al menos tan seguro como un cable".

i. MSDU delivery: Responsable de entregar la información al nivel físico

Existe una relación entre asociación y autenticación que provoca los tres "Estados" en los que se puede encontrar una STA en cualquier intervalo de tiempo:

- Estado 1: No autenticado – No asociado.
- Estado 2: Autenticado – No asociado.
- Estado 3: Autenticado – Asociado.

Estos servicios generan distintos tipos de mensajes, los cuales están clasificados en:

a. Data.
b. Control.
c. Management.

7. Formatos de trama :

7.1. tramas MAC:

Estas tramas poseen tres componentes:

- MAC Header.
- Body.
- Frame Check Sequence (FCS).

Octetos:	2	2	6	6	6	2	6	0-2312	4
	Frame Control	Duration / ID	Address 1	Address 2	Address 3	Sequence control	Address 4	Body	FCS
	MAC Header							***Body***	***FCS***

A continuación se desarrollan en detalle cada uno de los campos:

Si se analiza en detalle los dos octetos del **campo control**, los mismos están compuestos por los siguientes subcampos (Que se corresponden a los 2 octetos [16 bits] del campo "Frame Control"):

Bits:	2	2	4	1	1	1	1	1	1	1	1
	Protocol Version	Type	Subtype	To DS	From DS	More Frag	Retry	Pwr Mgt	More data	WEP	Order

a. Protocol Version (2 bit): El estándar actual es Versión 0.

b. Type (2 bit): 00= Management, 01=Control, 10=Data, 11=Reserved.

c. Subtype (4 bit): Definen el detalle del servicio y/o Primitiva (Ej: Association request y response, reassociation request y response, Beacon, Power Save, RTS, CTS, ACK, CF, etc...).

d. To DS (1 bit): En tramas de datos dirigidas hacia un DS=1, cualquier otro caso=0.

e. From DS (1 bit): En tramas que salen de un DS=1, cualquier otro caso=0.

f. More Frag (1 bit): En tramas de data o Management que poseen más fragmentos de MSDU=1, cualquier otro caso=0.

g. Retry (1 bit): Si es retransmisión de una trama anterior=1, cualquier otro caso=0.

h. Pwr Mgt (1 bit): Modo Power-Save=1, Modo Active=0

i. More Data (1 bit): Si una STA se encuentra en modo Power-Save y el AP posee más MSDU para ella=1, cualquier otro caso=0.

j. WEP (1 bit): Si el Body posee información que ha sido procesada con WEP=1, cualquier otro caso=0.

k. Order (1 bit): Si se emplea el servicio de Strictly Ordered=1, cualquier otro caso=0. (Este servicio permite reordenar la emisión de Broadcast y Multicast).

El segundo campo de la trama MAC es Duration ID, el cual consta también de 2 octetos. En la mayoría de los casos indica la duración de la trama, cuyo valor oscila entre 0 y 32767 (en decimal). La única excepción es cuando se trata de una trama de type=control con subtype= Power-Save, en cuyo caso los bit 14 y 15 son=1 y los restantes 14 bit indican la Association Identity (AID) de la estación que generó la trama, su valor oscila entre 1 y 2007.

Los otros campos que incluye la trama son los de direcciones, los cuales son empleados para indicar el BSSID. El formato de los mismos es el estándar de 48 bits (definido en IEEE 802-1990), respetando las mismas estructuras de Unicast, Multicast y Broadcast. Si bien en algunas tramas pueden no aparecer los cuatro, lo más normal es que sean los siguientes:

- Destination Address (DA): Identifica el recipiente final de la MSDU.
- Source Address (SA): Identifica el host que inició la transferencia de la MSDU.

- Receiver Address (RA): Identifica el recipiente inmediato al que será entregada la trama sobre el WM.
- Transmitter Address (TA): Identifica la STA que ha transmitido sobre el WM la MPDU

El campo que queda es el de Sequence Control Field que tiene 16 bits y consiste en 2 subcampos:

Bits:	4	12
	Fragment Number	Sequence Number

- Sequence Number: (12 bits) es un valor que se asigna a cada MSDU generada y oscila entre 0 y 4096, incrementándose en 1 por cada trama.
- Fragment Number: (4 bits) Si se emplea fragmentación (operación admitida por 802.11), este campo indica cada uno de los fragmentos, caso contrario es cero.

La cola de una trama 802.11 es el FCS (Frame Control Sequence) que es el CRC de grado 32, que corresponde al estándar IEEE CRC-32.

$$G_{(x)} = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

7.2. Formato de tipos de trama:

Volviendo al campo de control de la trama, como se detalló en el punto 7.1. El subcampo Type, identifica 3 tipos de trama: 00= Management, 01=Control, 10=Data. Por lo tanto, acorde a estos valores, se definen cada una de estas tramas, y su formato en cada caso difiere del formato genérico de la trama (definido en el punto 7.1.). A continuación se detallan cada uno de estos tipos.

7.2.1. Tramas de control:

Son las que en el subcampo Type tienen valor=01, y se emplean para control de red. Los distintos subtipos de tramas de control se detallan a continuación:

7.2.1.1. Formato de trama RTS (Request to send):

Octetos:	2	2	6	6	4
	Frame Control	Duration	RA	TA	FCS

- Frame Control, RA, TA y FCS, son los mismos anteriormente descritos.

- Duration: Este valor es el tiempo en microsegundos requeridos para transmitir los datos pendientes (o también una trama de administración) + una trama CTS + una trama ACK + 3 intervalos SIFS.

7.2.1.2. Formato de la trama Clear to Send (CTS):

Octetos:	2	2	6	4
	Frame Control	Duration	RA	FCS

- RA es copiado del valor e TA de la trama RTS inmediatamente previa.
- Duration: es el mismo que el de la trama RTS inmediatamente previa – el tiempo requerido para transmitir esta trama (CTS) y el SIFS que espera la misma.

7.2.1.3. Formato de la trama Acknowledgment (ACK):

Este formato es exactamente igual al de CTS.

7.2.1.4. Formato de la trama Power-Save (PS-Poll)

Octetos:	2	2	6	6	4
	Frame Control	AID	BSSID	TA	FCS

- BSSID: Es la dirección de la STA contenida en el AP.
- AID: Es el valor asignado a la STA transmitiendo la trama, en respuesta (o durante) una asociación.

7.2.1.5. Formato de la trama CF-End (Contention Free-End):

Octetos:	2	2	6	6	4
	Frame Control	Duration	RA	BSSID	FCS

- BSSID: es la dirección del AP.
- RA: es la dirección Broadcast de grupo.
- Duración: Debe ser siempre=0.

7.2.1.6. Formato de la trama CF-End + CF-ACK:

Es exactamente igual a la anterior.

7.2.2. Tramas de datos:

Estas tramas son independientes del subtipo, y los únicos detalles significativos son los campos Address que dependerán del valor de los bits To DS y From DS, presentando diferentes contenidos (o significados) en base a las cuatro combinaciones de estos dos bits. Las mismas no serán tratadas en este texto.

7.2.3. Tramas de administración:

El formato genérico de estas tramas es el que se detalla a continuación

Octetos:	2	2	6	6	6	2	0-2312	4
	Frame Control	Duration	DA	SA	BSSID	Sequence control	Body	FCS

Sobre este formato, basado en diferentes valores del Frame Control, se distinguen los subtipos de trama que presentan, las cuales pueden ser las siguientes:

7.2.3.1. Beacon frames:

El cuerpo (body) de una trama de administración "Subtipo" Beacon contiene la siguiente información:

- Timestamp.
- Beacon Interval.
- Capability Information.
- SSID.
- Supported rates.
- FH Parameter Set.
- DS Parameter Set.
- IBSS Parameters Set.
- TIM: (Sólo presente en tramas generadas por AP).

9. Descripción funcional de subnivel MAC:

La arquitectura de este subnivel incluye dos funciones principales: Función de coordinación distribuida (DCF) y Función de coordinación puntual (PCF), que se desarrollan a continuación:

9.1.1. DCF:

El método de acceso fundamental de 802.11 es conocido como CSMA/CA (Carrier Sense Multiple Access / Colision Avoidance) como lo hemos visto ya en Ethernet, y en este caso opera de la siguiente forma: una STA para transmitir, primero debe escuchar el canal para determinar si otra está transmitiendo. Si el medio está libre, entonces podrá transmitir acorde a los tiempos de espera correspondientes (que se detallarán más adelante), pues siempre debe dejar "ciertos intervalos de tiempo". Si el medio está ocupado, entonces la STA deberá esperar a que finalice la presente transmisión. En este último caso, como así también cuando acaba de transmitir la propia STA y desea enviar otra trama, la STA generará un valor random (y teniendo en cuenta también los tiempos que impone el estándar) y lo irá decrementando, hasta hacerse cero. Llegado este valor, podrá transmitir. Un aspecto que puede ser empleado también para acceso al medio, son las tramas RTS y CTS, que del mismo modo que la interfaz RS-232, solicita autorización y se habilita la Tx, por medio de estos cortos mensajes.

9.1.2. PCF:

Este método sólo puede ser empleado en modo infraestructura. Emplea un Point Coordinator (PC), quien coordinará dentro del BSS qué STA tiene permiso para transmitir. La operatoria es el conocido sondeo (Poll), realizado desde el PC. Esta PC posee un mecanismo prioritario de acceso al canal a través de las tramas de Management "Beacon", configurando lo que se denominará NAV (Network Allocation Vector). Este acceso prioritario, provisto por el PC puede ser utilizado para crear lo que se denomina Contention-Free (CF) Access Method (Método de acceso libre de colisiones).

9.1.3. Coexistencia de DCF y PCF:

Cuando un PC esté operando en un BSS, ambos métodos se irán alternando y pueden convivir.

9.2.3. Espacio entre tramas (IFS: Interframe Space):

Los intervalos de tiempo entre tramas son los IFS. Se definen cuatro tipos de IFS, para establecer prioridades de acceso al medio:

a. SIFS (Short IFS): Este intervalo se emplea en tramas ACK, CTS o en las sucesivas tramas de una operación de fragmentación. También en cualquier respuesta a un sondeo realizado por un PC. Es el intervalo más corto.

b. PIFS (PCF IFS): Se emplea en modo PCF (excepto en las respuestas a sondeos)

c. DIFS (DCF PFS): Se emplean en modo DCF par envío de tramas de administración y de datos.

d. EIFS (Extended IFS): Este intervalo se emplea cuando el nivel físico le informa al subnivel MAC que una trama que se ha transmitido, no tuvo una correcta recepción con incorrecto FCS. Se emplea este valor máximo de intervalo, para dar tiempo suficiente a la STA receptora, de informar este error de recepción.

9.2.4. Random Backoff time:

Cuando una STA desea transmitir, y la función "Carrier Sense" detecta ocupado el canal, deberá desistir de la Tx hasta que se desocupe el medio y durante un intervalo DIFS finalizada la Tx anterior si esta llega con éxito, si es motivo de errores, el intervalo de espera deberá ser EIFS. Una vez finalizado cualquiera de estos dos intervalos, generará un valor aleatorio denominado "Random Backoff Time", que deberá esperar antes de transmitir. El objetivo del mismo es minimizar colisiones. Este valor se compone de:

Backoff Time = Random () * Slot Time.

El valor Random está relacionado a un parámetro denominado "Contention Window" (Mínima y máxima) y sus límites oscilarán entre 0 y $\mathbf{2^n-1}$ Siendo "n" la cantidad de intentos de acceso (Muy similar a la técnica de tratamiento de colisiones de 802.3). Y el Slot time es un valor que depende de las características físicas del canal.

11. Entidad de administración de subnivel MAC:

Todas las STA que estén dentro de un BSS, estarán sincronizadas por un reloj común. La responsable de esta actividad es la función sincronización de tiempo (TSF).

En una red tipo infaestructura el AP será el "Timing Master" y llevará a cabo la TSF. Transmitirá periódicamente tramas "Beacon" que contienen copias del "TSF timer". Cualquier STA siempre aceptará estas tramas provenientes del que sirve su BSS.

11.2. Power Mode:

Una STA puede permanecer en dos estados:

- Despierta (Awake): Está en condicioones normales de operación.
- Dormida (Doze): No está en capacidad de Tx o Rx, y consume mucha menos potencia. Escucha periódicamente las tramas "Beacon", para ver si su AP necesita cambiarla de estado, para enviarle información.

La transición entre estos dos estados es controlada por cada STA (configurada). Lo importante a tener en cuenta aquí es que cuando una STA modifica su Power Mode, inmediatamente debe indicarlo a su AP a través de una trama de administración con los bit de PS configurados acorde al estado. El AP lleva una tabla de control de todas las STA, llamada "Traffic Indication Map" (TIM)

12. Especificaciones de servicio de nivel físico (PHY).

El estándar 802.11 posee diferentes especificaciones físicas, pero cada una de ellas siempre posee dos funciones:

- Función de convergencia: Adapta la trama MAC con el PMD.
- Sistema PMD (Physical Medium Depend): Define las características y métodos de Tx y Rx de datos a través de WM.

14. Especificaciones de FHSS (Frecuency Hopping Spread Spectrum), para la banda 2,4 Ghz ISM (Industrial, Scientific and Medical Band).

14.3.2. Formato de la trama a nivel físico:

Hasta ahora se ha tratado el formato de la trama MAC. A partir de ahora se analiza el formato de la trama en el nivel inferior del modelo, es decir, este nivel físico recibe la PDU de nivel 2 (es decir la trama MAC completa) a través del SAP correspondiente, arma su Header, lo suma a la PDU recibida y este conjunto es lo que va a Tx por el WM.

El conjunto total de bits que se inyectarán en el canal de comunicaciones, a través de este nivel se puede clasificar en tres grandes partes:

- Preámbulo: Se emplea para sincronizar la transmisión con todos los nodos que vayan a escucharla. Contiene dos campos:
  - Sincronización (SYNC) de 80 bits alternando ceros y unos.
  - Delimitador de inicio de trama (SFD) de 16 bits (0000 1100 1011 1101).
- Header: Contiene tres campos:
  - PLW (Physical Length Word): 12 bits que indican la longitud del campo de datos.
  - PSF (Physical Signaling Rate): 4 bits, de los cuales, el primero debe ser cero (Reservado), y las 8 combinaciones de los 3 bits siguientes indican a qué velocidad de transferencia de datos operará esta trama, desde 1 Mbps (000) hasta 4,5 Mbps (111), incrementándose de 0,5 Mbps.

- HEC (Header Error Check): 16 bits que emplean la técnica de CRC con el polinomio Generador $G_{(x)} = X^{16} + X^{12} + X^5 + 1$

- Datos: PDU de nivel 2. Cabe destacar aquí que en este nivel, los datos emplean la técnica de "Scrambler frame Synchronous", organizando bloques de 127 bits, que se irán mezclando en filas y columnas para minimizar los efectos de ráfagas de errores que puedan sufrir en el WM.

(fin de transcripción de la numeración del estándar, se vuelve a la numeración del texto)

4.2.3. La capa de enlace de 802.11:

Respetando el modelo OSI, en este texto se agrupará en el nivel de enlace, los dos subniveles que lo conforman (MAC: Medium Access Control y LLC: Logical Link Control). Desde el punto de vista de 802.11, sólo interesa hacer referencia al subnivel MAC (En los Ejercicios de este capítulo, se pueden apreciar varios tipos de tramas).

- **Capa MAC**: Controla el flujo de paquetes entre 2 o más puntos de una red . Emplea CSMA/CA: Carrier Sense Multiple Access / Collision Avoidance. Sus funciones principales son:

  - **Exploración:** Envío de Beacons que incluyen los SSID: Service Set identifiers o también llamados ESSID (Extended SSID), máximo 32 carateres.

  - **Autenticación:** Proceso previo a la asociación. Existen dos tipos:

    → Autenticación de sistema abierto: Obligatoria en 802.11, se realiza cuando el cliente envía una solicitud de autenticación con su SSID a un AP, el cual autorizará o no. Este método aunque es totalmente inseguro, no puede ser dejado de lado, pues uno de los puntos más fuertes de WiFi es la posibilidad de conectarse desde sitios públicos anónimamente (Terminales, hoteles, aeropuertos, etc.).

    → Autenticación de clave compartida: Es el fundamento del protocolo WEP (hoy totalmente desacreditado); se trata de un envío de interrogatorio (desafío) por parte del AP al cliente.

    - **Asociación:** Este proceso es el que le dará acceso a la red y sólo puede ser llevado a cabo una vez autenticado

    - **Seguridad:** Mediante WEP, con este protocolo se cifran los datos pero no los encabezados.

    - **RTS/CTS:** Funciona igual que en el puerto serie (RS-232), el aspecto más importante es cuando existen "nodos ocultos", pues a diferencia de Ethernet, en esta topología SÍ pueden existir nodos que no se escuchen

entre sí y que solo lleguen hasta el AP, (Ej: su potencia está limitada, posee un obstáculo entre ellos, etc), en estos casos se puede configurar el empleo de RTS/CTS. Otro empleo importante es para designar el tamaño máximo de trama (en 802.11: mínimo=256 y máximo=2312 Bytes).

- **Modo ahorro de energía:** Cuando está activado este modo, el cliente envió previamente al AP una trama indicando "que se irá a dormir". El AP coloca en su buffer estos datos. Se debe tener en cuenta que por defecto este modo suele estar inactivo (lo que se denomina Constant Awake Mode: CAM).
- **Fragmentación:** Es la capacidad que tiene un AP de dividir la información en tramas más pequeñas.

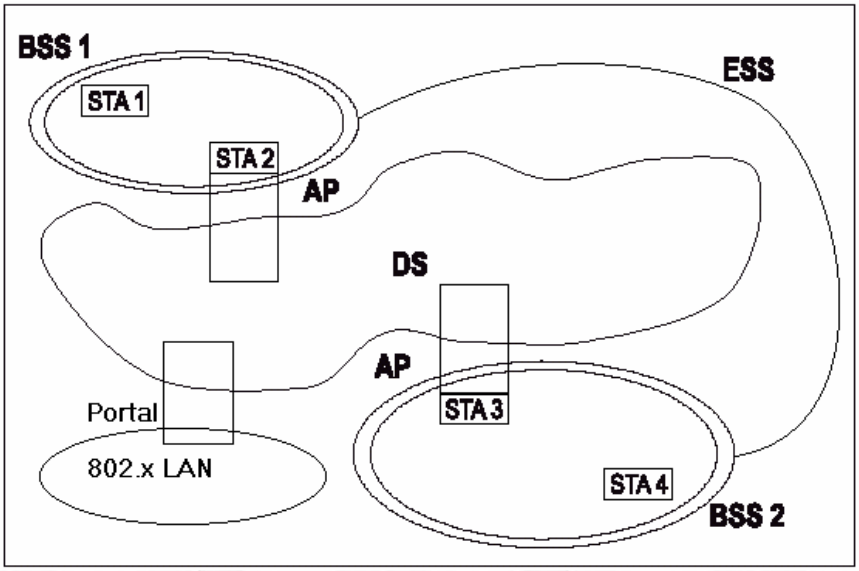
4.2.4. Topología WiFi.

802.11 presenta dos topologías:

- **Ad Hoc (o peer to peer)**: Dos o más clientes que son iguales entre ellos.
- **Infraestructura**: Red centralizada a través de uno o más Access Point (AP).

Descripción general de componentes de las mismas:

- **BSS** (Basic Service Set): Es el bloque básico de construcción de una LAN 802.11. En el caso de tratarse de únicamente 2 estaciones se denomina IBSS (Independent BSS), es lo que a menudo se denomina "Ad Hoc Netwok".
- **DS** (Distribution System): Es la arquitectura que se propone para interconectar distintos BSS. El **AP** es el encargado de proveer acceso al DS, todos los datos que se mueven entre BSS y DS se hacen a través de estos AP, como los mismos son también STA, son por lo tanto entidades direccionables.
- **ESS** (Extended Service Set): Tanto BSS como DS permiten crear wireless network de tamaño arbitrario, este tipo de redes se denominan redes ESS.
- La integración entre una red 802.11 y una No 802.11 se realiza mediante un **Portal**. Es posible que un mismo dispositivo cumpla las funciones de AP y Portal.



(Componentes de la arquitectura)

4.3. ARP (Address Resolution Protocol) (RFC 826, 1293, 1390):

Antes de segur avanzando hacia el nivel de Red, debemos tratar este protocolo, que si fuéramos estrictos no se lo puede asociar con exactitud a un nivel específico pues reúne información de nivel 2 y 3. Para nuestro análisis, lo consideraremos justamente como uno de estos protocolos que no responde exactamente a la clasificación de niveles y será un protocolo que opera "entre" los niveles 2 y 3.

4.3.1. Funcionamiento

Para que se pueda establecer la transferencia de datos entre dos ETD en la familia TCP/IP, estos deberán conocer obligatoriamente las direcciones IP y las de Hardware (MAC), del emisor y receptor; hasta que estas cuatro no se encuentren perfectamente identificadas, no se podrá iniciar ninguna transferencia de información. Bajo este esquema, es fácil pensar que si un ETD A desea envía información, conozca su Dirección MAC e IP, también es razonable que pueda conocer la dirección IP destino; el responsable de descubrir la dirección MAC faltante es el protocolo ARP. El mecanismo que emplea es el de mantener una tabla dinámica en memoria en cada ETD llamada caché ARP (la cual se puede analizar por medio del archivo ARP.exe), en la misma se van guardando todas las asociaciones de MAC-IP que escucha el ETD en la red. Al intentar transmitir información, analizará primero en su caché ARP si esta asociación existe, de no encontrarla generará un mensaje ARP.

4.3.2. Tipos de mensajes:

ARP trabaja por medio de una solicitud y una respuesta. La **Solicitud** es un broadcast de nivel 2 (FF.FF.FF.FF.FF.FF), el cual será escuchado por todos los ETD de la red con el formato que se graficará a continuación. Por ser Broadcast, todos los niveles 2 de todos los ETD de la red lo reconocerán como propio, entregando la UDP correspondiente al nivel 3 en todos los ETD. El único nivel 3 que lo tomará como suyo será el que identifique su propia dirección IP en esta cabecera quien responderá (**respuesta** ARP) colocando en la dirección MAC faltante la propia, pero ya no por medio de broadcast sino dirigida al ETD que generó la solicitud ARP, pues poseerá todos los datos necesarios. Al llegar a destino se completa toda la información necesaria para iniciar la transferencia de información, y se incluirá esta nueva asociación MAC-IP en la caché ARP.

Un caso obvio es cuando el ETD no pertenece a la propia red, por lo cual jamás será alcanzado por un broadcast de nivel 2 (pues un router lo filtraría). En esta situación, el router que sí escucha el broadcast (o puntualmente la solicitud ARP dirigida a él) reconoce que no se trata de una dirección de la red propia, y opera en forma similar a un ETD pero a través de tablas llamadas caché PROXY ARP, en las cuales mantiene asociaciones MAC-IP por cada puerto que posea, si en esta no se encuentra la dirección MAC buscada, generará un nuevo formato ARP por la puerta hacia la cual identifique la red IP correspondiente (este último paso puede variar acorde al tipo de protocolo que emplee el router), esto se repetirá a lo largo de toda una cadena de router hasta identificar la red IP correspondiente a la dirección buscada, donde se generará un broadcast que sí encontrará a la dirección IP deseada, la cual responderá la solicitud ARP, y por el camino inverso, y en forma dirigida llegarán la dirección MAC solicitada a destino, completando de esta forma las cuatro direcciones necesarias.

La última de las posibilidades existentes ocurre cuando un ETD no conoce su propia dirección IP, circunstancia que puede presentarse cuando bootea un ETD y solicita una asignación dinámica de dirección IP o también al inicializar un ETD que no poseen disco rígido. Ante este tipo de sucesos existe el Protocolo R_ARP (Reverse) (RFC 903), el cual genera un mensaje con formato semejante al ARP pero sin contener tampoco su propia dirección IP, la condición imprescindible para este protocolo es la existencia de un servidor R_ARP el cual recibirá este mensaje, resolviendo el direccionamiento IP del ETD que lo requiera. Los pasos de este protocolo son análogos a los del ARP. En el encabezado Ethernet, el campo identificador de protocolo de capa superior (SAP) llevará el valor 8035h que identifica R_ARP.

4.3.3. Formato del encabezado ARP.

Tipo de Hardware
Tipo de protocolo
Longitud de direcciones de Hardware
Longitud de direcciones de Protocolo

Código de operación
Dirección de Hardware del transmisor
Dirección IP del transmisor
Dirección de Hardware de receptor
Dirección IP de receptor

- Tipo de hardware: (16), interfaz de hardware empleado (Valor 1 para Ethernet).
- Tipo de protocolo: (16), identificador del protocolo que se emplea (Valor 0800 para IP).
- Longitud de dirección de Hardware y Protocolo: (8), limitan los campos posteriores a la cantidad de octetos que emplee cada uno de ellos.
- Código de operación: (16), (opcode), sólo se encuentran definidos 2 tipos, 1 = Solicitud, 2 = Respuesta (En realidad también están definidos 3 y 4 pero son para solicitud y respuesta de RARP) .
- Direcciones: (48) (32), especifican el tipo necesario para ser resuelto por ARP.

4.3.4. Ataque ARP.

Este ataque tiene sentido únicamente en redes LAN (no debe olvidarse que el 80% de los ataques suceden en este entorno), se trata de una actividad verdaderamente peligrosa, pues redirecciona el tráfico hacia el equipo deseado. La lógica de su implementación es la siguiente:

- Se debe escuchar el tráfico ARP.
- Al detectar una solicitud ARP, se espera la respuesta correspondiente.

- Se capturan ambas.
- Se modifica el campo dirección MAC de la respuesta, colocando la dirección MAC de la máquina que desea recibir el tráfico IP, falsificando la verdadera MAC de la respuesta.
- Se emite la respuesta ARP falsa y ya está.

¿Qué se logra con esto?

Si se supone que la solicitud ARP la emitió el host A y la respuesta ARP la emitió el host B, el resultado de estos mensajes es que el host A, al recibir la respuesta de B, almacena en su memoria caché ARP la dupla IP(B)-MAC(B). Si a continuación de este diálogo, el host A recibe otra respuesta ARP que le asocia la IP(B) con una nueva MAC, supóngase MAC (X), el host A, automáticamente sobreescribirá su memoria caché ARP con la nueva información recibida: IP(B)-MAC(X). A partir de este momento cada vez que emita información hacia la dirección IP del host A, la dirección MAC que colocará será la MAC(X), ante lo cual, el nivel Ethernet del host A descartará esa información, la cual sí será procesada por el protocolo Ethernet del host X, el cual por ser el intruso sabrá cómo procesarlo.

La totalidad de este ataque es conocido com “man in the middle” - ataque del hombre del medio, pues en definitiva el objetivo de máxima es poder “reencaminar” todo este flujo de información a través del host que fraguó la MAC para poder operar sobre las tramas que pasan por él.

El comando “arp”:

Muestra y modifica las tablas de conversión de direcciones IP en direcciones físicas que utiliza el protocolo ARP.

4.4. Telefonía Móvil.

Para cerrar el nivel de enlace, quisimos incluir aquí un breve resumen de los aspectos básicos relacionados a esta posibilidad de acceso a redes IP a través de la telefonía móvil centrando nuestra atención en la visión que se tiene respecto a la seguridad en las mismas.

La posibilidad de acceso a redes de datos desde un teléfono móvil surge a partir del momento de la puesta en marcha de este servicio de telefonía. Al principio con las redes GSM (Global System for Mobile Communications, u originariamente: Groupe Special Mobile), el acceso era exactamente igual que el de cualquier teléfono fijo, es decir a través de un modem analógico con una limitada velocidad, por esta razón es que nos centraremos en los servicios de telefonía móvil que fueron pensados no para redes analógicas, sino digitales y con la oferta de conmutación de paquetes, de los cuales el primero fue GPRS (General Packet Radio System), y luego UMTS (Universal Mobile Telecommunications System). La lógica de esta conexión, se inicia cuando un

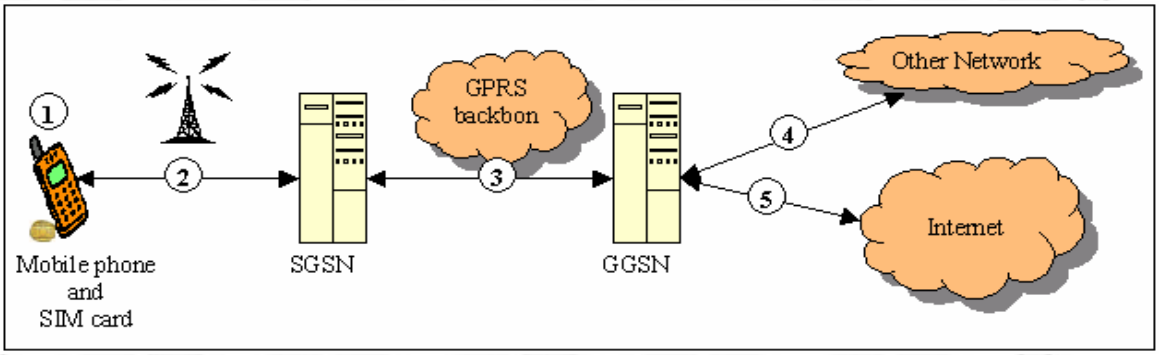
dispositivo desea realizar una comunicación de datos sobre la red GPRS (es decir se conecta a través de un modem de datos digital). Esta comunicación, a diferencia de la de voz, se establece con un primer nodo de la red telefónica que se denomina GGSN (Gateway GPRS Support Node), entre ambos se inicia el establecimiento de una relación denominada PDP (Packet Data Protocol), que de tener éxito finaliza con la creación de un "Contexto PDP" bajo el cual ya se establecieron todos los parámetros de seguridad y direccionamiento para que ese móvil pueda navegar por la red. No merece la pena entrar en más detalles al respecto, tampoco profundizar sobre el diálogo PPP o L2PP o el protocolo IP para Móviles (MIP), etc.

Hemos preferido en el presente texto centrar la atención en GPRS, por apreciarse como la tecnología que abarca la mayor parte de los conceptos a desarrollar, es decir:

- Si se analiza GSM, la visión es exactamente igual, pero sin la presencia de los SGSN (Serving GPRS Support Node) y GGSN.
- Si se analiza UMTS, la visión es exactamente igual, pero varía la interfaz radio.

Pero lo fundamental es que desde el enfoque de GPRS se pueden desarrollar conceptos que aglutinan el mayor porcentaje de aspectos a considerar.

4.4.1. Presentación.



En la figura atnterior, existen cinco áreas en las cuales la seguridad del sistema GPRS se encuentra expuesto:

a. Seguridad relacionada al MS (Mobile Station) y a la SIM Card (Subscriber Identity Module - módulo de identificación del suscriptor).
b. Mecanismos de seguridad entre MS y SGSN (Incluyendo también el radioenlace).
c. Seguridad en el Backbone PLMNs (Public Land Mobile Network), principalmente el tráfico entre SGSN y GGSN (también incluye el flujo entre abonado, HLR {Home Location Register} y SGSN).
d. Seguridad entre diferentes operadores.
e. Seguridad entre GGSN y las redes externas conectadas (Ej: Internet).

Desarrollo de las mismas:

a. Terminal y SIM Card:

La SIM Card contiene la identificación del abonado. Cuando es insertada en el ME (Mobile Equipment), ambos pasan a conformar una MS. La principal función de la SIM es trabajar en conjunto con la red GPRS para autenticar al abonado, antes que este obtenga acceso a la red. La SIM esta compuesta por:

- IMSI (International Mobile Suscriber Information): Es un idenmtificador único que consiste en tres dígitos Mobile Country Code (MCC), dos dígitos de Mobile Network Code (MNC) y 10 dígitos de Mobile Suscriber Identity Number (MSIN), es decir 15 dígitos en total.
- Ki: Es una clave de autenticación individual de abonado de 128 bits.
- Algoritmo generador de la clave de cifrado (A8): Este algoritmo emplea los 128 bits de Ki junto con un número Random, para generar una clave de 64 bits llamada GSRS-Kc.
- Algoritmo de autenticación (A3): Este algoritmo permite a una determinada identificación de abonado, autenticarse en la red GPRS. Este algoritmo debe responder a un desafío generado por la red, haciendo uso de la Ki.
- PIN (Personal Identification Number): Es una condición de acceso al MS.

El ME posee el algoritmo GPRS – A5 (instalado en el mismo) (Actualmente, este algoritmo puede residir en el ME o en el TE). Este algoritmo es empleado para cifrar datos y señalización durante la transferencia de datos. La seguridad del equipo confía en la integridad del IMEI (International Mobile Equipment Identity), que es un código de 14 dígitos decimales compuestos por 3 elementos:

- TAC (Type Aproval Code): 6 dígitos.
- FAC (Final Assembly Code): 2 dígitos.
- SNR (Serial Number): 6 dígitos.

El IMEI es almacenado en los terminales durante su fabricación, y el principal objetivo es poder tomar medidas contra equipos robados o en desuso.

Cada EIR posee tres listas con la totalidad de lo IMEI (Lista blanca, girs y negra).

b. Mecanismos de seguridad entre MS y SGSN (Incluyendo también el radioenlace):

Para proteger la confidencialidad de la Identidad del usuario, el IMSI nunca viajará como texto plano y normalmente ningún mensaje de señalización se envía sin cifrar (Se empleará el algoritmo A5-GPRS).

Para identificar un abonado móvil sobre el enlace de radio, se crea un TLLI (Temporary Logical Link Identity), este valor guarda relación con el IMSI y es guardado en el SGSN, y si

se hace presente un MS cuyo TLLI no se encuentre en esta base de datos, se solicitará a la MS que se identifique para poder acceder a la red.

c. Seguridad en el Backbone PLMNs, principalmente el tráfico entre SGSN y GGSN (también incluye el flujo entre abonado, HLR y SGSN).

   El operador es el responsable de la seguridad de su propio Intra PLMN backbone, el cual incluye todos los elementos de red y las conexiones físicas. Deberá prevenir el acceso no autorizado al mismo.

   El protocolo GTP (GPRS Tunneling Protocol) es el empleado entre nodos GSNs, a través del mismo puede ser tunelizado toda clase de protocolos (DNS, NTP, FTP, etc), este protocolo no cifra su payload.

d. Seguridad entre diferentes operadores.

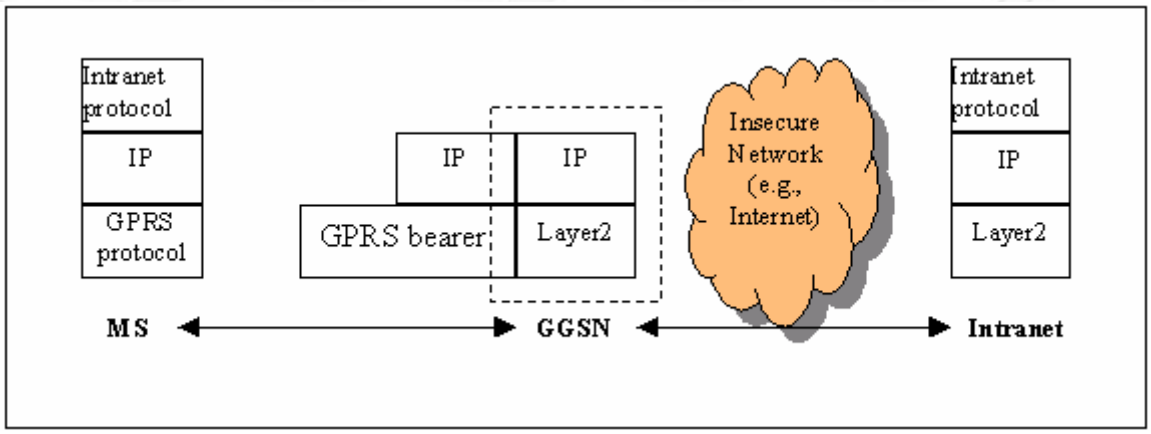
   La principal razón de intraoperabilidad entre operadores es el Roaming.

e. Seguridad entre GGSN y las redes externas conectadas (Ej: Internet y/o Red Corporativa).

   La interfase entre estos es Gi. Al conectarse a la red un MS, se le asigna una dirección IP (estática o dinámica). Una MS puede operar en modo transparente o no:

   - En modo transparente: el MS no necesita enviar ninguna autenticación y el GGSN no necesita participar en el proceso desautenticación ni control de acceso a esta MS.
   - En modo no transparente: se emplea algún servidor de autenticación, generalmente RADIUS.

Cualquier protocolo de seguridad que desee ser empleado aquí se hace sobre IP (Ej: IPSec) y no existe un protocolo específico de seguridad entre GGSN y la Intranet o Internet. Se puede apreciar en la siguiente gráfica como es el modelo de capas en cada interfase.



4.4.2. Distintos tipos de ataques que pueden llevarse a cabo en GPRS.

a. Terminal y SIM Card:
   - Integridad de datos: Al igual que cualquier PC conectada a Internet, estos elementos pueden sufrir los mismo tipos de ataques (Virus, troyanos, etc).
   - Robo de los mismos.
   - Escucha, spoof o manipulación de datos de estos dispositivos.
   - Pérdida de la confidencialidad de datos de usuario y autenticación del mismo.
   - Clonación de SIM Card (Este hecho ya fue detectado).

b. Interfase entre MS y SGSN:
   - Acceso no autorizado a datos: En el enlace aéreo, cualquier intruso puede escuchar el tráfico y pasivamente ir determinando tasa de transferencia, direcciones, longitudes, tiempos, etc, y luego activamente tratar de generar tráfico viendo qué sucede, tratando de iniciar conexiones.
   - La integridad de los datos puede ser manipulada en esta interfase.
   - Los ataques de negación de servicio son los más fáciles en esta interfase.
   - El acceso no autorizado a servicios puede ser llevado a cabo falsificando la conexión de un usuario.

c. Backbone GPRS:

   En esta zona se pueden generar los mismos ataques que entre MS y SGSN y se suma además la posibilidad de intrusiones desde otros PLMNs.

d. Interoperabilidad entre redes GPRS:

   La seguridad entre distintos operadores dependerá del grado de confiabilidad que tenga cada uno de ellos. El punto de especial interés en estos momentos está dado por la competencia que existe entre los operadores actuales por la "Captación de abonados", es decir que se compite por "Un mismo abonado", por lo tanto no es descabellado pensar que un determinado operador ataque a otro, para lograr que el abonado cambie su suscripción. Un claro ejemplo sería DoS, sobrefacturación, cortes de conexión, etc.

e. Interoperabilidad de GPRS PLMN y PDN: Este es el punto de especial interés, pues se trata de la interfase entre GPRS y las redes corporativas o Internet, desde las cuales puede provenir cualquier tipo de ataques.

4.4.3. Seguridad desde el punto de vista de interfaces.

a. **Gp:** Es la conexión lógica entre PLMNs que son usadas para soportar usuarios móviles de datos. Se emplea GTP para establecer una conexión entre un local SGSN y el usuario GGSN.

Generalmente el operador debe permitir el siguiente tráfico:

1) GTP: entre SGSN y el GGSN de los Roaming partners.
2) BGP: Información de ruteo entre GRX (GPRS Roaming Exchange) y/o Roaming partners.
3) DNS: Resolución para un abonado de un APN (Access Point Name).

ATAQUES TÍPICOS EN Gp:

Disponibilidad (DoS):

1) Border Gateway Bandwith Saturation (Saturación de Ancho de banda en Gateway de frontera): Un operador malicioso que esté conectado al mismo GRX (aunque no sea actualmente roaming partner), puede estar en capacidad de generar suficiente tráfico legítimo dirigido al Gateway de frontera, como para negar el acceso roaming al mismo.
2) DNS Flood: Los servidores DNS de la red pueden ser inundados con correctas o malformadas peticiones DNS u otro tráfico, negando a los abonados el acceso a su propio GGSN.
3) GTP Flood: SGSN y GGSN pueden ser inundados con tráfico GTP, ocasionando alto empleo de sus ciclos de CPU procesando datos inútiles.
4) Spoofed GTP PDP Context Delete: Un atacante que cuente con la información adecuada, puede armar mensajes con GTP PDP Context Delete, con los cuales borra los túneles entre SGSN y GGSN para un abonado determinado. Esta información puede ser obtenida a través de otros tipos de ataques a la red.
5) Bad BGP Routing Information: Un atacante que obtenga cierto control sobre los routers GRX o pueda inyectar información en las tablas de rutas, puede causar que el operador pierda rutas para roaming partners, negando el acceso a roaming.
6) DNS caché poisoning: Es posible engañar un DNS para que un usuario de un determinado APN (Access Point Name), resuelva un incorrecto GGSN o ninguno.

Autenticación y autorización:

1) Spoofed Create PDP Context Request: GTP no provee autenticación para los SGSN y GGSN en sí, esto significa que dada la información adecuada de un usuario, un atacante con acceso al GRX, otro operador o un usuario interno podría crear su propio SGSN y realizar un tunel GTP al GGSN del abonado,

falsificando la identidad del abonado, permitiendo con esto acceso ilegítimo a Internet o la red corporativa.

2) Spoofed Update PDP Context Request: Un atacante puede emplear supropio SGSN o comprometer alguno, para enviar una Update PDP Context Request a un SGSN que controle una sesión GTP, en esta situación, luego podría insertar su propio SGSN en la sesión GTP y obtener datos del abonado.

Integridad y confidencialidad:

Hay un informe (3GPP TS 09.60 V6.9.0) que expresa claramente que si un atacante tiene acceso a GGSN o SGSN (como un empleado de la empresa o un intruso que ha conseguido acceso a GRX) puede capturar todos los datos del abonado, pues la TPDU en GTP no va cifrada.

SOLUCIONES EN Gp:

El punto más importante en Gp es la debilidad de GTP, por lo tanto, empleando IPSec entre Roaming partners y limitando las tasas de tráfico, la mayoría de los riesgos se eliminarían.

Se debe tener en cuenta aquí lo mencionado en páginas anteriores sobre "interoperabilidad entre redes GPRS.

Las medidas específicas son:

1) Filtrado de paquetes entrantes y salientes para prevenir que el propio PLMN sea empleado para atacar otros Roaming partners.
2) Filtrado de paquetes GTP con control de estados, para permitir que sólo el tráfico requerido y desde las fuentes y destinos adecuados de los Roaming partners. Con esto se evita que desde otros PLMNs conectados al mismo GRX puedan iniciar algún tipo de ataque. El control de estados en GTP es crítico para proteger los GSNs
3) GTP traffic Shaping: Para prevenir que el ancho de banda sea empleado por terceros y que no se empleen recursos de procesador de los GSNs con malos fines, debería restringirse la tasa de tráfico GTP. Se podría limitar también la tasa de transferencia en los niveles 3 y 4 entre GTP, BGP y DNS.
4) Implementar túneles IPSec con Roaming partners.

b. **Gi:** Es la Interface entre GGSN y la red de paquetes (de la Empresa o Internet). Es la interface más expuesta.

Este tráfico, puede provenir de un MS hacia la red de paquetes o viceversa, por lo tanto puede tratarse de cualqueir tipo de tráfico generado o recibido por una aplicación cliente.

ATAQUES TÍPICOS EN Gi:

Disponibilidad:

1) Gi Bandwith saturation: Se trata aquí de inundar el enlace desde la PDN hasta el operador móvil.
2) Flooding a MS: Si un tráfico es dirigido a una IP puntual de un MS, como el ancho de banda de Internet suele ser muy superior al de GPRS, resulta muy fácil inundar el enlace de acceso a red.

Confidencialidad:

No existe protección de datos desde la MS al PDN o red corporativa.

Integridad:

Los datos pueden ser modificados a menos que se emplee seguridad en los niveles superiores.

Autenticación y Autorización:

A menos que túneles de nivel 2 y/o 3 sean empleados entre el GGSN y la red corporativa, puede ser posible a una MS acceder a la red corporativa de otro cliente. No se puede implementar autenticación únicamente a través de direcciones fuente, pues esta es fácilmente enmascarada (IP spoof).

Existen numerosos ataques que pueden ser posibles dependiendo de las aplicaciones empleadas por el abonado (virus, troyanos, etc).

Otros:

Ya existen artículos acerca de vulnerabilidades en GPRS que permiten el uso del canal sin pagar, generando una alta tasa de tráfico.

SOLUCIONES EN Gi:

1) Túneles lógicos desde GGSN a las redes corporativas. No debería ser posible enrutar tráfico desde Internet a la red corporativa, o entre redes corporativas entre sí. La implementación ideal aquí son los túneles de nivel 2 y 3. Si la conexión a la red corporativa es vía Internet, se debe usar IPSec.
2) Limitar tasas de tráfico. Sobre conexiones a Internet, priorizar el tráfico IPSec desde las redes corporativas. Esto evitará que cualquier ataque desde Internet deje fuera de servicio el tráfico corporativo. Se puede considerar también emplear interfaces físicas separadas.
3) Inspección de control de estados de paquetes: Considerar la posibilidad que únicamente se inicien las conexiones desde las MS hacia Internet y controlar las mismas. De no ser

posible, considerar dos tipos de servicios: uno en el cual se aplique lo mencionado y otro en el cual las conexiones puedan ser iniciadas desde Internet hacia las MS.

4) Filtrado de paquetes entrantes y salientes: prevenir la posibilidad de spoof IP con las direcciones IP asignadas a las MS.

c. **Gn**: Conexión lógica entre SGSN y GGSN, si es el mismo operador de PLMN (Public Land Mobile Network).

Las vulnerabilidades presentes en esta interface, pasan por usuarios de la misma operadora.

d. Interfaz entre MS y SGSN:

Se trataron en puntos anteriores.

Un punto de especial interés es que el MS está "Always On", lo que hace más probable su alcance para uso indebido. También hace más atractivo el servicio MobileMail.

e. Ataques presentados en artículos publicados en Internet:

- Ataques basura: Aprovechando un puerto externo usado por NAT para una determinada conexión, se genera tráfico hacia el mismo no necesariamente válido y/o ilegítimo, inundando el dispositivo que hacía uso real del mismo. Se probó esta te´cnica sobre UDP y TCP con iguales resultados.
- Ataques a NAT con puertos aleatorios: EL dispositivo NAT puede enviar o no respuestas que permitan identificar cuáles son los puertos que asigna a su red. En cualqueir caso, si el puerto está abierto, el tráfico pasa adentro de la red GPRS.
- Denial of Service Reset: Se verificó que es posible cortar conexiones de usuario por envíos de TCP Reset si se conocen los números de secuencia.
- Se plantea la posibilidad de instalar un falso servidor web en Internet y con una MS conectada a la red GPRS, pero haciendo IP Spoofing, realizar la conexión hacia el servidor. Una vez que la conexión se ha establecido, sería posible realizar ataques desde adentro como si el FW y el NAT no existieran.

4.4.4. Elementos vulnerables.

Dispositivos móviles programables desprotegidos.

Los dispositivos de red son (teóricamente) alcanzables por los clientes y potencialmente desde Internet.

Particular atención se debe prestar sobre la administración de claves, tanto en el almacenamiento como en el transporte de las mismas.

4.4.5. Autenticación GPRS:

a. Cuando la MS conecta, envía una solicitud de autenticación al SGSN, éste obtiene una autenticación de abonado desde HLR (Home Location Register) y AuC (Authentication Center), los cuales generan un "Tiplets" compuesto por: Random Number (RAND) – Signed Responsed (SRES) - Encryption Key GPRS-Kc (de 64 bits).
b. SGSN selecciona un número Random y lo envía a MS.
c. Tanto MS como SGSN cifran el número Random con el algoritmto A5 (IMSI nunca es enviado en texto plano), pero emplean los algoritmos A3 y A8 para autenticación.
d. SGSN valida el valor retornado y la sesión de datos es autenticada.
e. SGSN también consulta a EIR para ver si el dispositivo ha sido "hotlisted"

4.4.6. Criptografía en GPRS.

Interfaz radio:

a. MS y SGSN cifran empleando GEA (GPRS Encryption Algorithm): existen 7 de estos algoritmos, de los cuales ETSI ya definió 2, GAE1 y GEA2).
b. Se genera una nueva clave para cada sesión.
c. Permite periódicamente re-autenticación y nueva clave derivada.

Network: Crea VPN usando IPSec (Problemas con túneles empleando NAT).

4.4.7. Conclusiones GPRS

En el presente texto, se intentó simplemente dar una visión general de la telefonía móvil y las investigaciones realizadas sobre su seguridad. Sin entrar en mayores detalles, los aspectos sobre los cuales se trabaja sobre la seguridad en una red de telefonía móvil son:

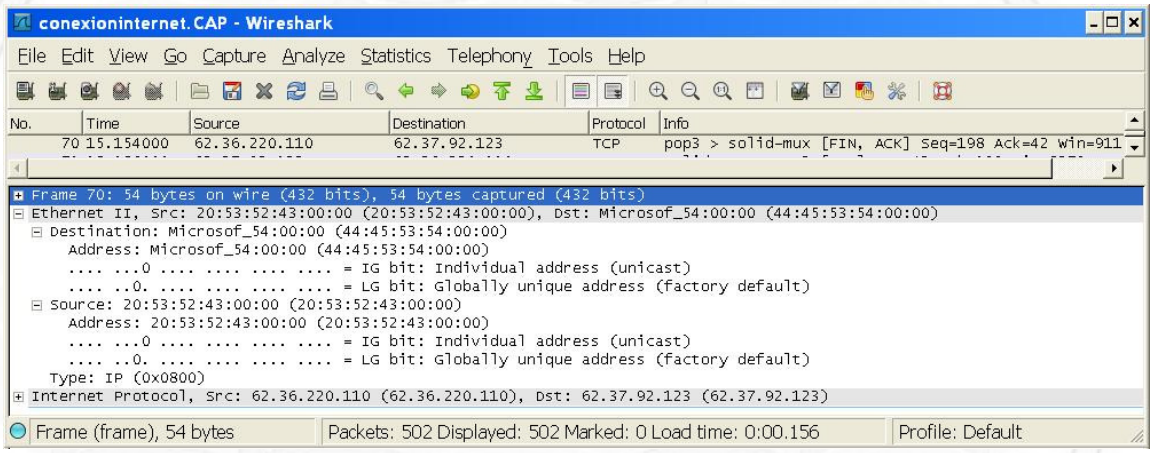
- Control del empleo de algoritmos de autenticación, generación de claves y criptografía.
- Control del empleo de las bases de datos en EIR y HLR.
- Verificaciones periódicas sobre el control de elementos perdidos, denunciados, rotos, o fuera de circulación.

- Análsis de tráfico en la interfaz radio.
- Bastionado de equipos.
- Segmentación estricta de redes (Autenticación, tránsito propio, facturación, roaming, DNSs, usuario particular y corporativo).
- Verificación de empleo de Proxies (en especial con GSM).
- Control de flujos de información.
- Empleo de ACLs.
- Empleo de túneles (GRE, IPSec, GTP, etc.).
- Control de reglas en Firewalls.
- Análisis de creación de contextos.
- Control de accesos.
- Empleo de tráfico no tunelizado.
- Control del tráfico de Roaming.
- Control del tráfico propio entre distintos emplazamientos a través de Gn.
- Auditorías de sistemas.
- Encapsulamientos en cambios de protocolos.
- Empleo de Sistemas de detección de intrusiones.
- Empleo de scaner de vulnerabilidades.
- Empleo de herramientas de cuantificación de bastionado de equipos.

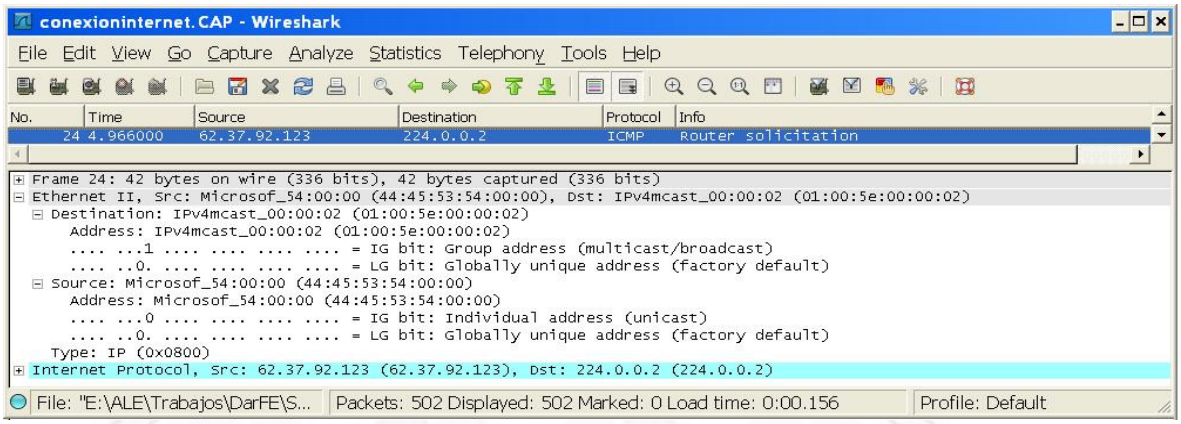
EJERCICIOS DEL CAPÍTULO 4 (Nivel de enlace)

A nivel enlace vamos a comenzar a trabajar con ejercicios de análisis de tráfico.

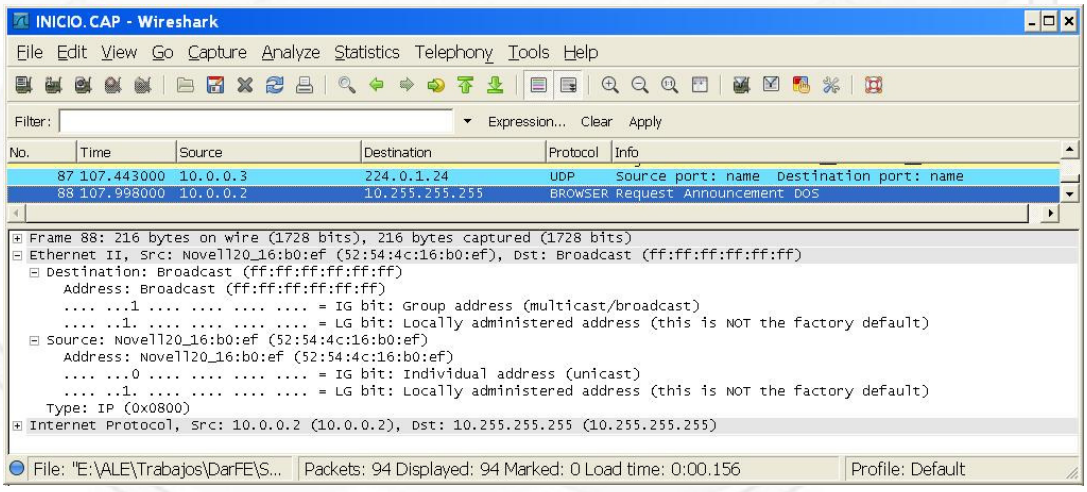
1. Para poder verificar la teoría con la práctica, nos interesa ser capaces de detectar tráfico Unicast, Multicast y Broadcast a nivel dos, es decir poder analizar el esquema de direccionamiento MAC y comprender bien sus 6 octetos presentados en forma hexadecimal, teniendo en cuenta los dos bits menos significativos del primer octeto, tal cual se explicó en la teoría. Para ello lanzaremos el analizador de protocolos (Ethereal o Wireshark) mientras navegamos por Internet o abrimos algún archivo compartido en otra máquina o enviamos un correo, etc. Y deberemos ser capaces de capturar los tres tipos de tramas que se presentan a continuación. El ejercicio consiste en evaluar lo que se explica en cada trama y compararlo con vuestras capturas.



Trama Unicast: Prestad atención a los bits menos significativos del primer octeto de la dirección destino, los mismos se encuentran con valor "0" por ser Unicast. Se puede ver claramente que los 6 octetos de la dirección origen y destino son pares de valores "hexadecimales" y se corresponden a lo mencionado en la teoría como el formato "EUI-48". Fijaros que, como el analizador de protocolos tiene dentro de sus librerías el listado de todos los fabricantes en esa fecha, inmediatamente puedo identificar los 3 primeros octetos de la dirección destino (44:45:53) y nos la presenta como "Microsoft", luego esta sería la tarjeta Nro: 54:00:00 que fabricó esta empresa, esto lo vimos como: "**OUI"** (Organizationally Unique Identifier) o "**company_id**".



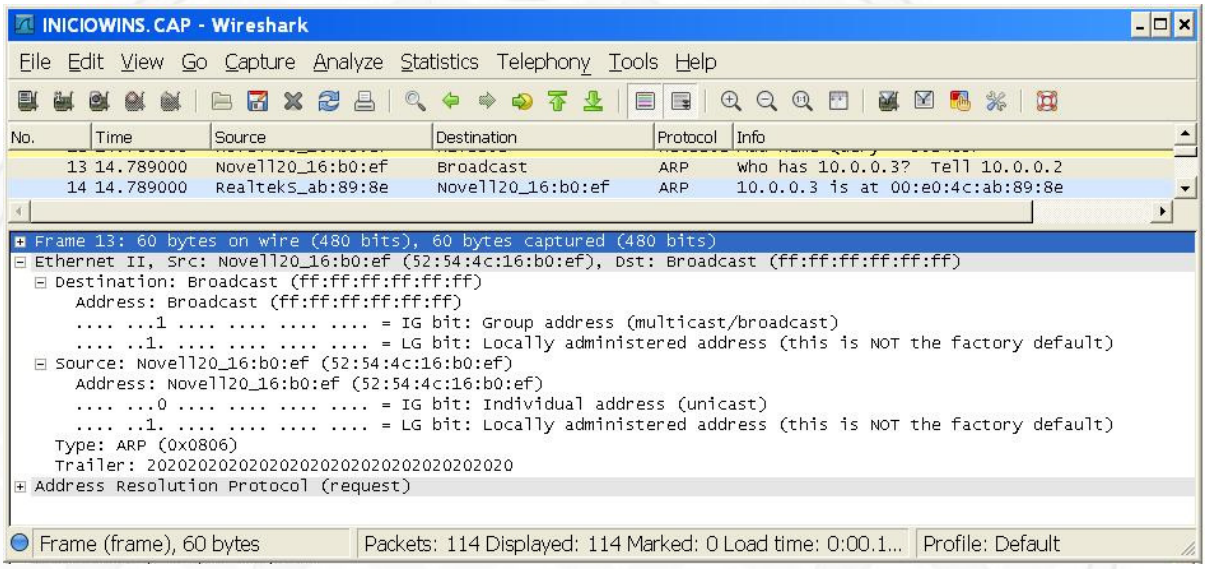
Trama Multicast: En esta trama, prestad atención al empleo del primer bit menos significativo del primer octeto con valor ".... ...1", esto lo vimos como: Grupal (Valor = 1), forma parte de un conjunto de direcciones MAC. En este caso, el conjunto de direcciones es un mensaje dirigido a un grupo de "Router" (lo veremos más adelante). Como detalle interesante para observar, es que la dirección IP destino (nivel de red) es 224.0.0.2 lo que identifica un Multicast también a nivel de red (nivel 3), y fijaros cómo este mismo valor se "copia" en los tres últimos octetos de la dirección MAC destino (00:00:02)..... ¿Qué raro no?.... ¿por qué será?..... es un tema que lo trataremos con detalle al llegar al nivel 3, pero nos pareció importante que prestéis atención a estos pequeños detalles desde el principio.



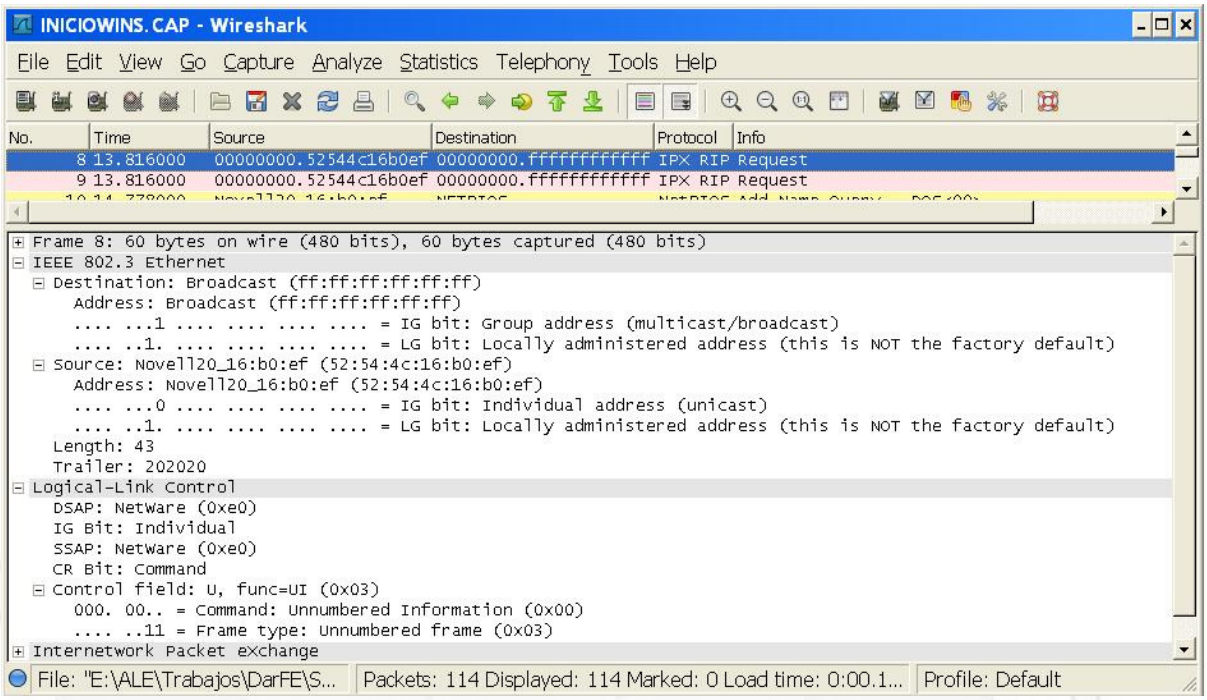
Trama Broadcast: Estas tramas como podéis apreciar, se identifican claramente por su dirección destino: ff:ff:ff:ff:ff:ff. El valor "ff" en hexadecimal, se corresponde con: "1111 1111" en binario y con: "255" en decimal, y será una constante en los esquemas de direccionamiento Broadcast. En esta trama lo que debéis notar es cómo ahora los dos bits menos significativos del primer octeto están puestos en "1", pues ahora, continuando con nuestra teoría: el primero de ellos identifica: Grupal (Valor = 1), forma parte de un conjunto de direcciones MAC, y el segundo: Local (valor = 1) tiene significado solamente en el ámbito local.

2. En este segundo ejercicio, continuando con el analizador de protocolos, os proponemos que lancéis capturas para poder identificar la diferencia entre tramas 802.3 y Ethernet. Para poder obtener tramas 802.3 uno de los mejores métodos es estar conectado a redes que soporten protocolos que no respeten estrictamente el modelo OSI y por lo tanto entrará en juego el mencionado grupo 802 de IEEE tal cual hemos dicho en la teoría, subdividiendo el nivel 2 en dos subniveles (MAC y LLC). Los dos protocolos más fáciles de hallar de esta pila son NetBios (de Microsoft) e IPX/SPX (de Novell), así que si tenemos acceso a alguna de estas redes pues sólo se trata de "sentarse a escuchar".

   El ejercicio consiste en evaluar lo que se explica en cada trama y compararlo con vuestras capturas.



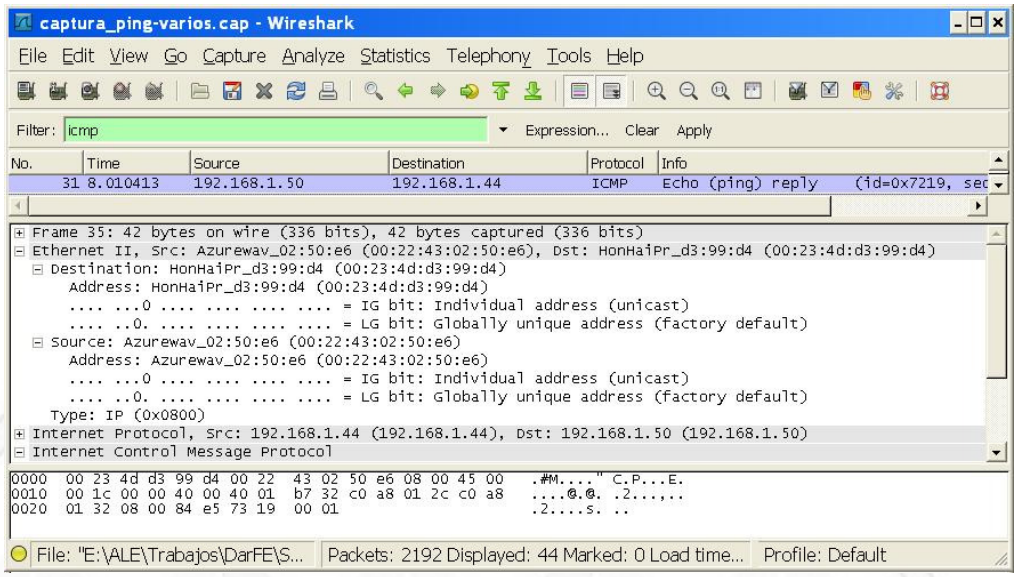
   Trama Ethernet: En este caso, se trata de una trama Broadcast, y podemos ver claramente que luego de los 12 octetos de direccionamiento, se ven los 2 octetos "Type" (o Ethertype como lo llamamos en la teoría), se trata del protocolo al que entregará su carga de datos de nivel superior, en este caso es "ARP" que ya lo hemos tratado y su valor en hexadecimal es "08 06" ocupando los dos octetos de ese campo.



Trama 802.3: Centrándonos ahora nuevamente a continuación de los 12 octetos de direccionamiento, podemos ver que el campo ahora es "Lenght" y su valor es "43". Como se trata de una trama 802.3, no necesita aclarar qué protocolo tiene en su capa superior, pues justamente IEEE subdividió este nivel e inexorablemente por arriba del subnivel MAC estará LLC, tal cual lo podemos ver en esta captura.

3. El conocido comando "**ping**" (que más adelante analizaremos con detalle), nos sirve inicialmente, para probar si se llega o no a una determinada dirección IP. Si bien este tema no se corresponde con este capítulo, por lo conocido que es este comando lo emplearemos de forma sencilla para analizar el comportamiento de Ethernet. Vamos a hacer diferentes pruebas desde Linux y desde Windows.

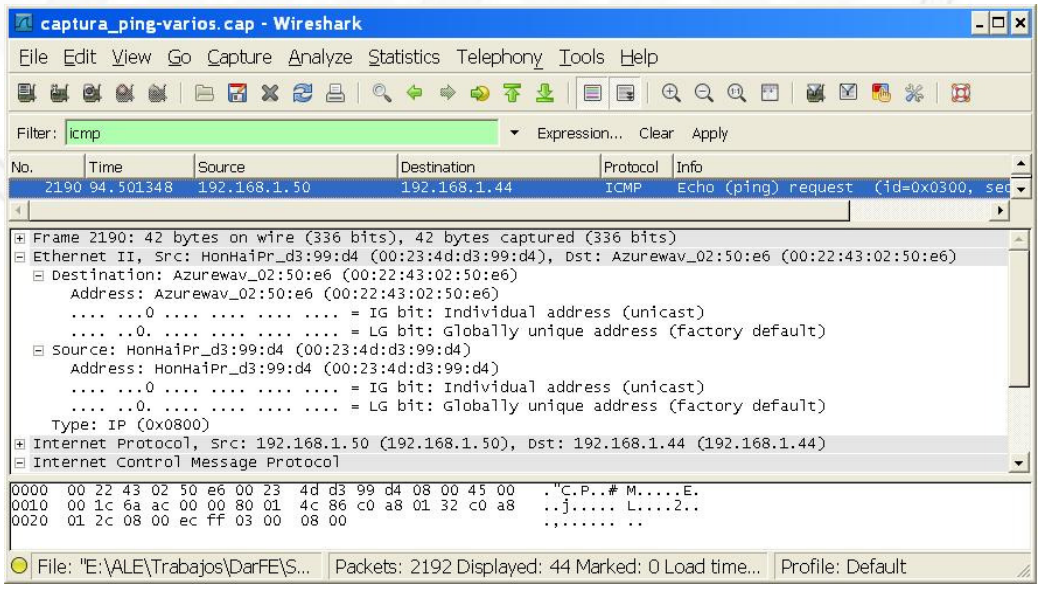
   3.1. Primero hagamos un aprueba desde Linux. Este Sistema Operativo nos ofrece la opción "-s" para definir un tamaño de datos a enviar. En el ejemplo que sigue se ejecutó el comando: "ping –s 0 192.168.1.50" (Desde la dirección IP:192.168.1.44), este ping se realiza desde una máquina Linux a una Windows XP, y en la imagen se muestra su respuesta (Reply).



¿¿¿¿¿¿Pero el tamaño mínimo de una trama Ethernet no era 64 Byte?????..... ¿Por qué entonces se ve con total claridad en la primera línea que esta trama tiene 42 bytes sobre el cable?.....

Fijaros como estamos comenzando a apreciar “anomalías” entre lo que la teoría nos dice y en la práctica sucede. Cada una de estas anomalías, omisiones, errores o “zonas grises” donde se desfasa la teoría y la realidad es por donde empiezan los problemas de seguridad, a través de estos pequeños sucesos se comienzan a abrir brechas que cuando se descubre su aplicación (para el lado malo) aparecen las dificultades. Aquí estamos viendo concretamente una de ellas.

3.2. Podemos ahora realizar la misma prueba desde Windows, la opción ahora será “-l” (letra ele). En el ejemplo que sigue se ejecutó el comando: “ping –l 0 192.168.1.44” (Desde la dirección IP:192.168.1.50), este ping se realiza desde una máquina Windows XP a una Linux, y en la imagen se muestra su Solicitud (Request).



Como podemos apreciar, el resultado es el mismo, una vez más se está violando la especificación de "tamaño mínimo de trama 64 Bytes".

3.3. Una prueba interesante ahora podría ser evaluar el rendimiento de una red Ethernet ante un alto volumen de ping. Lo que proponemos es que si dispones de dos máquinas en la misma LAN, pruebes de hacer diferentes secuencias de ping ininterrumpidos (Opción "-t" para Windows y por defecto en Linux). A continuación presentamos algunos ejemplos:

```
C:>ping -t -l 0 192.168.1.44

    Haciendo ping a 192.168.1.44 con 0 bytes de datos:
    Respuesta desde 192.168.1.44: bytes=0 tiempo=1ms TTL=64
    Respuesta desde 192.168.1.44: bytes=0 tiempo<1m TTL=64
    Respuesta desde 192.168.1.44: bytes=0 tiempo<1m TTL=64
```

Ej. Anterior: Ping con "cero" Byte de datos: tiempo de respuesta menor a 1 milisegundo.

```
C:>ping -t -l 1500 192.168.1.44

    Haciendo ping a 192.168.1.44 con 1500 bytes de datos:
    Respuesta desde 192.168.1.44: bytes=1500 tiempo=16ms TTL=64
    Respuesta desde 192.168.1.44: bytes=1500 tiempo=16ms TTL=64
    Respuesta desde 192.168.1.44: bytes=1500 tiempo=9ms TTL=64
```

Ej. Anterior: Ping con "1.500" Bytes de datos: tiempo de respuesta 16 ms.

```
C:>ping -t -l 65000 192.168.1.44

    Haciendo ping a 192.168.1.44 con 65000 bytes de datos:
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=186ms TTL=64
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=168ms TTL=64
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=198ms TTL=64
```

A partir de este momento (sin interrumpir este ping), otra máquina comenzó a generar también ping, fijaros el aumento en el tiempo de respuesta de las tramas siguientes.

```
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=771ms TTL=64
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=746ms TTL=64
    Respuesta desde 192.168.1.44: bytes=65000 tiempo=806ms TTL=64
```

Ej. Anterior: Ping con "65.000" Byte de datos: tiempo considerable.

El comando "ping" ofrece muchísimas más opciones que las veremos cuando analicemos más en detalle el mismo, por ahora sólo hemos querido presentarlo para evaluar aspectos de Ethernet.

4. los comandos para verificar y configurar la interfaz (o tarjeta) de red, son:

- "ipconfig" (para Windows): para ver todas sus opciones, se debe ejecutar "ipconfig /?"
- "ifconfig" (Para Linux): para ver todas sus opciones, se debe ejecutar "man ifconfig" Practica con ambos, prueba las diferentes opciones.

5. El comando "ifconfig" en Linux, posee una capacidad muy útil y a su vez peligrosa, la opción "hw". La misma permite modificar su dirección MAC. En Windows también puede hacerse desde "Conexiones de Red" → "Propiedades" (De la tarjeta que vaya a modificar) → "Configurar" → "Opciones Avanzadas", y allí colocar el valor que deseemos en la opción "Network address" (los seis pares de números hexadecimales sin separación).

   En este ejemplo lo haremos desde Linux. Para ello primero es necesario desactivar (comúnmente llamado "tirar abajo") la interfaz de red, luego ejecutar el comando y nuevamente "levantarla". Antes de hacer cualquiera de esta acciones te aconsejamos que consultes los 6 octetos de tu dirección MAC y los guardes en un archivo, papel o lo que fuera, pero haz lo posible por no perderlo, pues luego sino tendrás inconvenientes para restituirlos. La secuencia de comandos sería:

   - ifconfig eth2 down
   - ifconfig eth2 hw ether 00:1ª:13:b1:11:22 *(o cualquier secuencia)*
   - ifconfig eth2 up

   A partir de este momento tu MAC es la que acabas de configurar, haz hecho un "**MAC spoofing**".

6. Antes de avanzar un poco más ejercitemos un poco con nuestra tabla "arp", para visualizar las conexiones que ya ha reconocido, puedes hacerlo con el comando "arp –a" y verás algo como lo que mostramos a continuación:

```
C:\>arp -a
Interfaz: 192.168.1.50 --- 0x3
  Dirección IP          Dirección física      Tipo
  192.168.1.1           00-1d-20-0e-5d-df     dinámico
  192.168.1.21          00-1d-21-3ª-e1-c2     dinámico
  192.168.1.44          00-22-43-02-50-e6     dinámico
```

   Estás viendo los datos almacenados en una "Caché arp", en la cual presenta guardados la relación entre 3 direcciones IP y 3 direcciones MAC. Te proponemos que investigues las diferentes opciones que ofrece este comando, en particular que pruebes declarar direcciones estáticas, que verifiques cómo hacerlo para que continúen almacenadas al reiniciar el equipo, etc.

7. Ahora que hemos ejercitado algo de "**ifconfig**" y "**arp**", veamos cómo es la base de un " ataque ARP"

   Como se explicó en la teoría, este ataque debe lograr "infectar la tabla caché ARP de un host para que en la misma se pueda asociar una IP verdadera con una MAC falsa. En la parte

"Herramientas" de estos ejercicios lo verás con más detalle, aquí únicamente queremos que verifiques cómo es la lógica de este ataque y que hasta te convenzas que podrías hacerlo tu mismo con el sólo empleo de comando nativos de Linux y nada más.

- Tenemos dos ETD:

  ETD A: Windows XP - Dirección IP 192.168.1.50

  ETD B: Linux - Dirección IP 192.168.1.44

- Si ejecuto "ifconfig eth2" en el ETD A (Linux), me presenta la siguiente información (Se remarca en negrita los datos de especial interés):

  eth2

  Link encap:Ethernet direcciónHW **00:1a:13:b3:ec:b0**

  inet dirección:**192.168.1.44** Difusión:192.168.1.255 Máscara:255.255.255.0

  ARRIBA DIFUSIÓN MULTICAST MTU:1500 Métrica:1

  RX packets:0 errors:0 dropped:0 overruns:0 frame:0

  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

  colisiones:0 txqueuelen:1000

  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

  Interrupción:220 Dirección base: 0x4000

- Desde este ETD A (Linux) ejecutamos: "ping 192.168.1.50"

  Respuesta desde 192.168.1.50: bytes=0 tiempo=1ms TTL=64

  Respuesta desde 192.168.1.50: bytes=0 tiempo<1m TTL=64

  Detenemos el "ping" con "[Ctrl] + C"

- Ejecutamos "arp –a" en el ETD B (Windows)

Dirección IP	Dirección física	Tipo
192.168.1.1	00-1d-20-0e-5d-df	dinámico
192.168.1.44	00:1ª:13:b3:ec:b0	dinámico (*Podemos apreciar que la MAC es la original del ETD A (Linux)*

- Volvemos a nuestro ETD A (Linux) y ejecutamos la siguiente secuencia de comandos:

  ifconfig eth2 down

  ifconfig eth2 hw ether 00:1a:13:b3:33:44

  ifconfig eth2 up

- Ahora desde este mismo ETD A (Linux), repetimos el ping anterior: "ping 192.168.1.50"

  Respuesta desde 192.168.1.50: bytes=0 tiempo=1ms TTL=64

Respuesta desde 192.168.1.50: bytes=0 tiempo<1m TTL=64

- Detenemos el "ping" con "[Ctrl] + C"
- Si desde el ETD B (Windows)volvemos a ejecutar "arp –a"

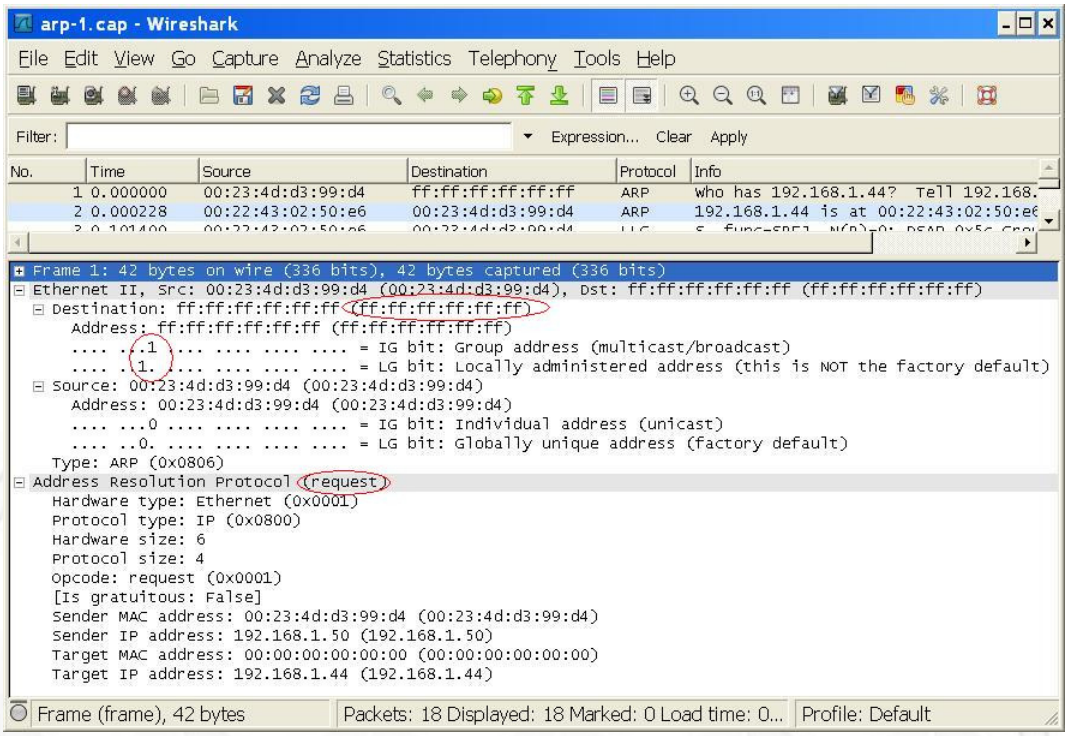
Dirección IP	Dirección física	Tipo
192.168.1.1	00-1d-20-0e-5d-df	dinámico
192.168.1.44	00:1ª:13:b3:33:44	dinámico *(Podemos apreciar que la MAC es la que acabamos de cambiar en el ETD A (Linux)*

CONCLUSIÓN: Acabamos de "infectar" la caché ARP del ETD B (Windows) con nuestra falsa MAC, a partir de ahora toda comunicación que el ETD B haga hacia la IP 192.168.1.44 lo hará con esta MAC falsa. ¿Pero de qué me sirve?.....

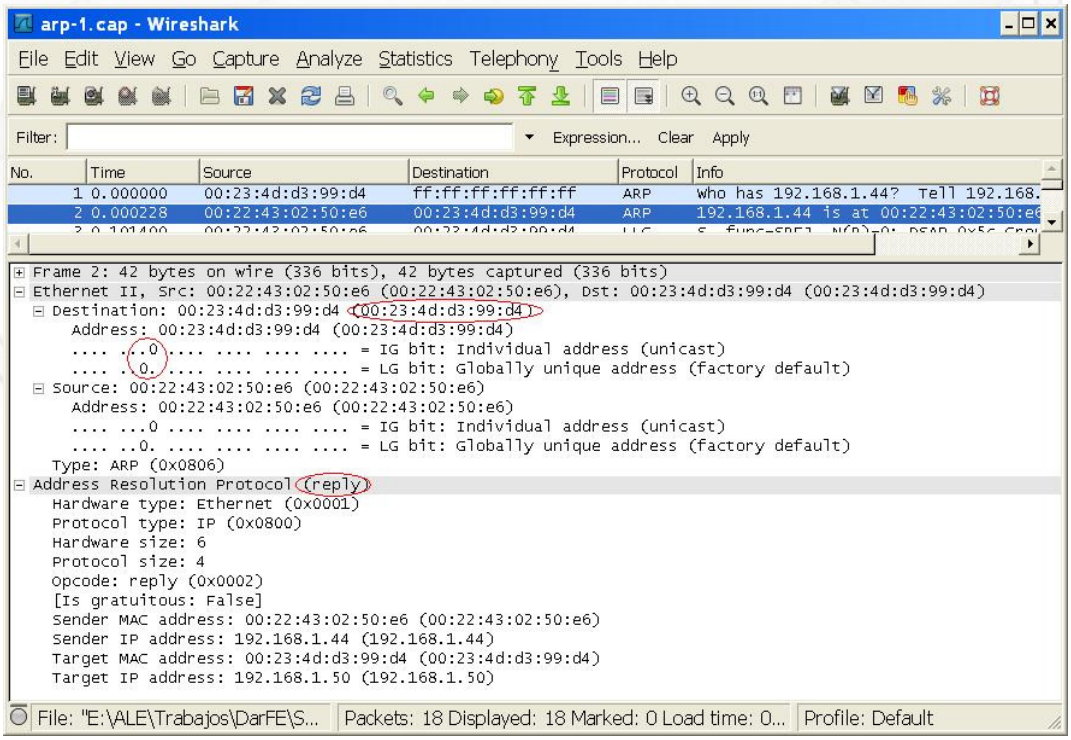
Esto es sólo el primer paso para que empieces a ejercitar estos temas, a medida que profundicemos en ellos irás viendo "la luz" sobre este ataque, por ahora, te invitamos a que releas la teoría de este ataque, investigues por Internet, reflexiones y ejercites acerca de ¿Qué sucedería si en vez de mi MAC pusiera la de otra IP real?, ¿Qué sucedería si en vez de un ping común y corriente generara algún tipo de tráfico con IPs falsas?, ¿Qué sucedería si me dedico a escuchar tráfico con Ethereal y a cada IP que escucho la "machaco" con una MAC falsa?, tenemos muchísimas posibilidades y opciones de ejercitar con estos dos comandos: ifconfig y arp. Te invitamos a que a este ejercicio le dediques su tiempo, y verás que encuentras muy buenas conclusiones.

8. Análisis de tráfico ARP.

8.1. En estos ejercicios con el protocolo ARP, inicialmente te invitamos a que una vez más lances tu analizador de protocolos y puedas verificar la solicitud y respuesta ARP, tal cual fue expuesta en la teoría. Si no has tenido ninguna conexión con otro ETD, sencillamente puedes hacer "ping" hacia él y podrás verificar que en el analizador de protocolos se ha generado un par de tramas con protocolo ARP, deberías capturar algo similar a las imágenes que te presentamos a continuación.



Presta atención a que, tal cual está resaltado en rojo, se trata de una Solicitud (Request) desde una dirección MAC origen, que se corresponde a la IP: 192.168.1.50, para que le respondan cuál es la MAC de la IP: 192.168.1.44 (que fue el destino de nuestro "ping").



En esta trama vemos la respuesta (Reply) al ARP anterior, y aquí queda claro que ya no se trata de un Broadcast a nivel MAC, sino de una trama dirigida concretamente a la MAC destino.

8.2. Hemos querido poner en evidencia un hecho que es de suma utilidad para detectar si por alguna razón existe algo en las asignaciones IP ←→ MAC que no cuadra del todo. Esto puede suceder por varias razones, una de ellas es justamente que alguien intencionadamente lo esté haciendo. Veamos cómo está la "caché ARP del ETD B (Windows), es decir la IP: 192.168.1.50:

- C:\>arp -a

  Interfaz: 192.168.1.50 --- 0x3

Dirección IP	Dirección física	Tipo
192.168.1.44	00-22-43-02-50-e6	dinámico

Como continuación del ejercicio anterior, ahora desde este mismo ETD B (Windows), hemos creado una entrada ARP estática en su tabla, ejecutando el siguiente comando:

- C:\>arp -s 192.168.1.44 00-22-43-02-aa-bb

Fijaros que le estamos "ordenando" que a la misma IP anterior (192.168.1.44) ahora le asigne otra MAC. A continuación podemos verificar su resultado consultando nuevamente su "caché ARP):

- C:\>arp -a

  Interfaz: 192.168.1.50 --- 0x3

Dirección IP	Dirección física	Tipo
192.168.1.44	**00-22-43-02-aa-bb**	estático

Hemos remarcado (en negrita) la nueva MAC que tiene asociada la vieja IP. No podría ser de otra forma, pues estas caché "pisan" cualquier asociación anterior al recibir una nueva. Si ahora intentamos hacer "ping" desde este ETD B (Windows) hacia el otro, por supuesto no nos responderá el segundo, pues ese "ping" está saliendo directamente (sin previas tramas ARP) hacia una MAC que el otro ETD al recibirla, directamente la descarta, si siquiera llegar a mirar su dirección de red (IP), pues a nivel MAC observa que no va dirigida a él. Abajo se presenta el "ping" con su resultado.

- C:\>ping 192.168.1.44

  Haciendo ping a 192.168.1.44 con 32 bytes de datos:

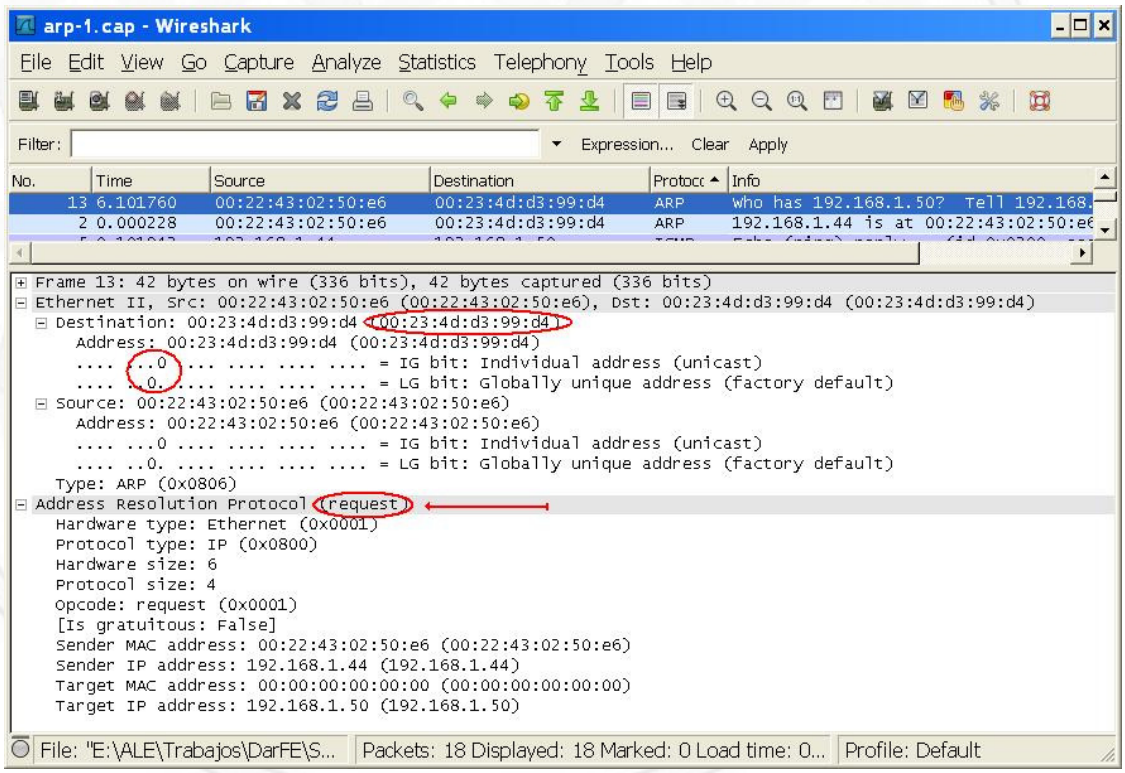
  Tiempo de espera agotado para esta solicitud.

  Tiempo de espera agotado para esta solicitud.

  .....

8.3. La parte más interesante de este ejercicio ARP es, si ahora lanzáramos un "ping" desde el segundo ETD A (Linux), hacia el primero ETD B (Windows). En esta situación la caché del ETD A (Linux) aún mantiene su asociación verdadera con la IP ←→ MAC del ETD B (Windows), entonces al ejecutar "ping" hacia este, directamente lo hará sin lanzar ninguna trama ARP, el ETD B (Windows) recibirá este "ping" y le responderá, pero hacia la MAC que encuentra en su caché ARP (la incorrecta) por lo tanto el ETD A (Linux) nuevamente descartará este trama..... pasado cierto tiempo el ETD A (Linux) notará que algo raro sucede, pues hasta hace unos segundos esta comunicación funcionaba y repentinamente dejó de hacerlo, y la lógica lo lleva a generar una solicitud ARP nuevamente, pero esta vez no lo hará con Broadcast sino de forma dirigida hacia la MAC que él tiene en su caché ARP.

Una solicitud ARP que no es Broadcast es un claro indicio de la existencia de una anomalía a nivel enlace, y puede considerarse seriamente para generar algún tipo de alarma. A continuación presentamos esta captura, que ha seguido todos los pasos que acabamos de describir.



Puede verse que se trata de una Solicitud (Request), y que va dirigida concretamente a la MAC 00:23:4d:d3:99:d4 que es la verdadera del ETD A (Windows).

Te invitamos a que hagas este ejercicio, y también a que te animes a relacionarlo con los anteriores, para verificar opciones de "MAC spoofing" y resultados que se observan con el protocolo ARP y sus tablas "caché".

EJERCICIOS CON HERRAMIENTAS

1. Para continuar midiendo el rendimiento de una red para el protocolo Ethernet, en este ejercicio os proponemos que empleéis la herramienta "**iperf**" (disponible gratuitamente para Linux).

NOTA: En este texto no deseamos describir la instalación ni las URLs desde donde instalar ni hacer funcionar las diferentes aplicaciones que usemos, pues las mismas son actualizadas con mucha frecuencia y su instalación varía tanto en versiones como en la distribución del SO, por lo tanto haremos mención a las aplicaciones y hoy en día cualquiera de ellas se puede encontrar con suma facilidad a través de cualquier buscador.

La herramienta "**iperf**" funciona a modo "cliente – servidor" por lo tanto para medir el ancho de banda, se debe ejecutar entre dos máquinas.

Una de ellas con el comando:

- iperf -s -p (puerto_en_el_que_deseo_escuchar)

Y la otra en modo cliente:

- iperf -c (Direccion_IP_del_Servidor) -p (puerto_en_el_que_escucha_el_Servidor)

  A partir de este momento podrás verificar los tiempos de respuesta en esa conexión. La mejor forma de realizar este ejercicio es contar con más de un cliente y verificar cómo se incrementa el tiempo de respuesta a medida que se suman más clientes y más tráfico.

A continuación presentamos un ejemplo de esta herramienta:

Desde el servidor:

iperf -s -u -f MB -i1 *(Esta opción "f B" nos muestra los resultados en Bytes, "-i1" indica cada cuántos segundos ser realizará la medición)*

```
------------------------------------------------------------
Server listening on UDP port 3001
Receiving 1460 byte datagrams
UDP buffer size: 0.01 MByte (default)
------------------------------------------------------------
[1634] local 192.168.1.50 port 3001 connected with 192.168.1.44 port 2960
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
```

Desde el cliente:

iperf -c 192.168.1.50 -f **B**

```
------------------------------------------------------------
Client connecting to 192.168.1.50, TCP port 3001
TCP window size: 64512 Byte (default)
```

```
------------------------------------------------------------
[632] local 192.168.1.44 port 2690 connected with 192.168.1.50 port 3001
[ ID] Interval Transfer Bandwidth
[632] 0.0-10.0 sec 124391197 Bytes 12860641 Bytes/sec
```

2. La próxima herramienta que os proponemos emplear es "**ettercap**". Se trata de una herramienta muy útil para entornos LAN, es de descarga gratuita, nació bajo línea de comandos y en la actualidad ya ofrece una interfaz gráfica aceptable. Puede funcionar como "sniffer" y también permite generar tráfico y ejecutar el ataque del hombre del medio (MITM) a través del protocolo ARP.

   Los ejercicios que te proponemos hacer, es que descargues e instales la herramienta, luego comienza con "sniffing" de la red. Una vez realizado esto, ya te aparecerán nuevas opciones para "ARP Poisonning" (envenenamiento de caché ARP) y también MITM. Con esto tienes para dedicarle un buen tiempo, encontrarás mucha información de su uso en Internet, por esa razón no creemos necesario extendernos más aquí.

   Seguramente te tentarán las opciones que tiene de robo de cuentas de usuario y contraseñas (aún cifradas con SSL), y terminarás sucumbiendo a la tentación de usarlas, pero sobre este tema aún no nos queremos dedicar, pues no forma p arte del nivel de enlace.

3. Vamos a trabajar un poco con la herramienta "**arpspoof**". El objetivo principal que queremos que ejercites con la misma es el poder "sniffar tráfico independientemente del lado del Switch que te encuentres, es decir, si un switch "conmuta" la comunicación entre dos de sus bocas, y mi ETD no forma parte de esos segmentos, aún nos queda una posibilidad de escuchar el tráfico entre ambos y eso es lo que haremos. Una vez logrado este objetivo, en realidad lo que acabas de hacer es un MITM entre ambos, así que de paso podrás verificar ambos casos.

   Supongamos que en nuestra LAN es la **191.168.2.0** (cuando veamos el nivel 3, entraremos en detalle sobre este esquema de direccionamiento), nuestra dirección IP es la **192.168.2.101**. Todos los ETD de esta LAN, están conectados a diferentes bocas de un único Switch, y desde el mismo se puede salir hacia Internet por medio del router ADSL que tiene la dirección IP: **192.168.2.1**, que también está conectado al mismo Switch.

   Nuestra intención es poder escuchar el tráfico que circula desde la IP (Víctima): **192.168.2.48** hacia el router y viceversa. En una situación normal, el Switch conmutaría los puertos de la víctima y el router y nosotros quedaríamos ajenos a ese diálogo.

   Lo primero que se debe hacer en nuestro ETD (Linux) es activar una opción de red que permite "encaminar" direcciones IP que justamente no sean las nuestra, esta opción se llama "ip forwading" y se activa a través de la siguiente línea de comandos:

   - **# echo 1 > /proc/sys/net/ipv4/ip_forward** *(por supuesto con usuario "root")*

   Con esto entonces, lograremos que cualquier paquete IP que nuestra máquina reciba y no sea para ella, lo reenviará hacia su destinatario.

Supongamos ahora que nuestra interfaz de red sea "**eth0**", entonces deberíamos ejecutar:

- **# arpspoof -i eth0 -t 192.168.2.1 192.168.2.48**

  La opción "**-i**" nos indica sobre qué interfaz de nuestro equipo se aplicará esta orden y la opción "**-t**", la dirección a la cual vamos a envenenar su "caché ARP".

Desde una nueva consola ahora ejecutamos:

- # **arpspoof -i eth0 -t 192.168.2.48 192.168.2.1**

Con este último envenenamos la dirección contraria. Es decir, ambos ETD tienen en su caché ARP nuestra dirección MAC asociada con la IP de su vieja comunicación.

A partir de este mismo momento, somos el MITM de este flujo de datos, por lo tanto si lanzáramos nuestro analizador de protocolos, capturaríamos todo el tráfico entre ambos, independientemente de que el Switch anteriormente nos dejara fuera, ahora este mismo Switch al recibir cualquier trama hacia la víctima o hacia el router, verá que la MAC destino es la nuestra, y por lo tanto la sacará por la boca a la cual físicamente estemos conectados nosotros.

4. Nuestra siguiente herramienta es "**arpwatch**", también de libre distribución para Linux. Este software lo podríamos catalogar como "muy necesario" en redes LAN, permite, por medio de la escucha de tráfico, mantener tablas "vivas" de esta asociación MAC ←→IP de cada uno de los hosts presentes en nuestra LAN, y al producirse alguna modificación de ellas tomar las acciones oportunas que nosotros hayamos pre configurado. Siempre se debe tener en cuenta lo mencionado en la teoría respecto a "Hubs y Switchs" para poder "escuchar" en los segmentos debidos, pues debes tener en cuenta que un Switch te dejará sin escuchar el resto de sus bocas. Existen varias formas de "escuchar" toda la red: con varios "sniffers" creando "port mirroring" en los switchs, configurando los switchs (y hasta los routers) para que dejen pasar tráfico ARP, colocando sondas en segmentos específicos, spplitters, etc. Todo esto debes analizarlo y estudiarlo muy en detalle antes de poner una herramienta como este en producción.

   Una vez instalado, su "corazón" está en el archivo "*arpwatch.conf*" que suele encontrarse en el directorio "/etc" pero puede variar en base a la distribución de Linux que se emplee. Encontrarás muchísima información al respecto en Internet.

   Sobre este software, también te invitamos a que lo pruebes, investigues y lo emplees seriamente en tus redes LAN.

5. A nivel Wifi:

   5.1. Instalando la herramienta "**aircrack-ng**" lo primero que debes comenzar a realizar es "escucha de tráfico" para familiarizarte con la misma. Dependiendo de la versión de aircrack, y de lo que vayas a realizar, deberás descargar también "**airdump**"

5.2. El próximo paso será el empleo de la funcionalidad de "Generar tráfico ARP". Este protocolo que tratamos en este capítulo, en el caso de este ejercicio, tiene por finalidad obligar al AP a generar nuevos vectores de inicialización (VI), y en la medida que logramos varios de ellos, se agilizará muchísimo la ruptura de la clave WEP. Para esta actividad la tarjeta WiFi debe permitir funcionar en modo "Monitor" (como el promiscuo de Ethernet), ten en cuenta que no todas lo soportan, así que si decides comprar una que lo haga (y hasta sería aconsejable una de mayor potencia) consulta primero la página de aircrack-ng (http://aircrack-ng.org/) donde mantienen un listado actualizado de las tarjetas compatibles.

5.3. Selecciona los AP de la lista que mayores capturas tengan y continúa la secuencia para el "Crack" de la clave de acceso.

5.4. Conéctate a es AP con la clave que acabas de descifrar.

5.5. Puedes intentar hacer los mismos pasos con la herramienta "airsnort" (http://airsnort.shmoo.com).

DESAFÍOS:

1. Investiga más herramientas para WiFi.
2. Investiga la tecnología "Bluetooth" que ya empieza a ser interesante como acceso a sistemas y redes.
3. Profundiza en el empleo de analizadores de protocolos (como Wireshark) pues son una herramienta básica y fundamental para la seguridad, y aún puedes sacarle muchísimo más provecho del que presentamos hasta ahora.

CAPÍTULO 5: El nivel de RED

En este capítulo desarrollaremos el nivel 3 del modelo de capas, de cual os recordamos que su función principal es el control y manejo de "rutas".

5.1. Análisis de datagramas (IP):

Se trata de un protocolo de nivel 3 no orientado a la conexión, permitiendo el intercambio de datos sin el establecimiento previo de la llamada. Una característica fundamental es que soporta las operaciones de fragmentación y defragmentación, por medio de las cuales un datagrama se subdivide y segmenta en paquetes más pequeños para ser introducidos a la red, y luego en destino se reconstruyen en su formato original para entregarlos al nivel superior (si bien en la actualidad, como se verá más adelante, esta tarea suele realizarla TCP). La otra operación que revista importancia es el ruteo, el cual implementa por medio de un esquema de direccionamiento que se trata a continuación. En la actualidad se continúa empleando la versión 4 de este protocolo, pero está en estudio, muy próxima a implementarse y en algunos ámbitos ya se empea, la Versión 6 o Next Generation. Por razones que se tratarán al final de esta sección se continúa postergando la entreda en vigor de esta versión 6, y por esta razón en estos párrafos se explicará lo referido a la versión 4.

5.1.1. Direcciones IP (rfc 791):

Internet por medio del empleo del campo de direcciones de un datagrama, sólo puede identificar en cada uno de los bit dos elementos:

- **HOST (H).**
- **NET (N).**

Este campo de direcciones, está constituido por cuatro octetos, los cuales se pueden presentar en binario (bbbbbbbb.bbbbbbbb.bbbbbbbb.bbbbbbb), en hexadecimal (hh.hh.hh.hh) o en decimal (d.d.d.d). Es importante habituarse a la correspondencia entre binario y decimal para un ágil manejo de estas direcciones que como ya puede apreciarse, oscilarán entre 0/255.0/255.0/255.0/255 en sistema decimal. Dentro de este espectro en los cuatro octetos, existen varias direcciones RESERVADAS, las dos más comunes (si bien prifundizaremos más en ellas) por ahora son:

- **00000000** (en binario) o 0 (en decimal): que especifica "La red donde me encuentro".
- **11111111** (en binario) o 255 (en decimal): que especifica un mensaje de "Broadcast".

Tal vez lo más importante para lograr entender plenamente este esquema de direccionamiento, es no olvidarse que a pesar que las direcciones IP se presentan en forma decimal, ningún dispositivo de red tiene diez dedos, por lo tanto todo (absolutamente TODO) dispositivo que tenga que trabajar sobre una dirección IP, la analizará como una secuencia de 32 bits (pues tiene sólo dos dedos) Lo arraigado que tenemos en nuestra mente el sistema decimal, nos lleva indefectiblemente a razonamientos no adecuados sobre estas direcciones, así que os invitamos a que hagáis un gran esfuerzo por tratar de no usar ocho dedos de vuestras manos y SIEMPRE tratar de razonar el esquema de direccionamiento IP, como secuencias binarias, hasta os permitiremos hacer un poco de trampa y "colar" alguna pasaje a decimal, pero insistimos, haced el esfuerzo de familiarizaros con el pasaje de decimal a binario en cada octeto, y veréis que es la mejor forma de ser un experto en redes IP.

Acorde al primer octeto, se pueden clasificar distintos tipo de redes**:**

0xxxxxxx Tipo A: Como el primer bit es 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 0 y 127.

10xxxxxx Tipo B: Como el primer bit es 1 (ya pesa 128) y el segundo obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 128 + (0 a 63) a 192.

110xxxxx Tipo C Como los dos primeros bit son 11 (ya pesa 192) y el tercero obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 192 + (0 a 31) a 223.

1110xxxx Tipo D Como los tres primeros bit son 111 (ya pesa 224) y el cuarto obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 224 + (0 a 15) a 239. **Este tipo de direcciones están reservadas para empleo de multicast.**

11110xxx Tipo E Como los cuatro primeros bit son 1111 (ya pesa 240) y el quinto obligatoriamente 0, este tipo de redes solo podrán abarcar el rango de direcciones entre 240 + (0 a 7) a 247. **Este tipo de direcciones están reservadas para uso experimental por parte de los organismos de Internet**.

Al diferenciar estos tipo de redes, a su vez por medio de un concepto denominado MASCARA DE RED que se tratará más adelante, en particular las tipo A, B y C determinan ciertos límites entre Host y Net que se detallan a continuación:

Tipo A: (0 a 127), el primer octeto identifica a Net y los otros tres a Host. Por lo tanto existirán 127 posibles redes A y cada una de ellas podrá contener tres octetos de Host lo que equivale a 2^{24} = 16.777.214 Host, **(N.H.H.H).**

Tipo B: (128 a 191) Los dos primeros octetos identifican a Net y los otros dos a Host. Por lo tanto existirán 2^{14} Net = 16.384 posibles redes B y cada una de ellas podrá contener dos octetos de Host lo que equivale a 2^{16} = 65.534 Host, **(N.N.H.H).**

Tipo C: (192 a 223) Los tres primeros octetos identifican a Net y el último a Host. Por lo tanto existirán 2^{21} Net = 2.097.152 posibles redes C y cada una de ellas podrá contener un octeto de Host lo que equivale a 2^{8} = 254 Host, **(N.N.N.H).**

Las cantidades mencionadas numéricamente son las reales si bien pueden no coincidir con algunas potencias pues dentro de los rangos establecidos, también existen determinadas direcciones reservadas.

5.1.2. Máscara de Red y Subred:

Dentro de una red IP, se pueden crear distintas subredes, empleando el concepto de "Máscara". Estas subredes "piden prestado" bit de los campos identidad de Host, y por medio del empleo de AND lógico se solapan con el bit correspondiente a esa posición de la Máscara de subred, Si este último es un uno, se corresponderá a un bit que identifica a una dirección de red (N), caso contrario será una dirección de Host.

Ej:

	decimal	Binario
Dirección IP	193.66.66.240	11000001.01000010. 01000010.11110000
Máscara	255.255.255.0	11111111.11111111. 11111111.00000000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNNN. HHHHHHHH

En este ejemplo se identifica el Host número 240 perteneciente a la red tipo C número 193.66.66. Si se deseara poder crear dos subredes dentro de esta misma red, para segmentar distintos grupos lógicos de nivel 3, se debería cambiar uno de los bit correspondientes al cuarto octeto de la máscara, de manera tal que permitiera solaparse con su correspondiente bit de la dirección IP. Cualquiera de los ocho bit de este último octeto podría ser elegido, y técnicamente se crearían dos subredes, pero el mejor planteo para el diseño "entendible humanamente" es el de **comenzar a enmascarar de izquierda a derecha**, lo que permitirá identificar por el valor decimal de ese octeto a que subred se refiere (Pues el ser humano piensa en decimal y no en binario). Para aclarar este planteo se propone el siguiente ejemplo en la creación de dos subredes dentro de la misma red anteriormente descripta:

Ej:

	decimal	Binario
Dirección IP	193.66.66.240	11000001.01000010. 01000010.11110000
Máscara caso 1	255.255.255.128	11111111.11111111. 11111111.10000000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNN. NHHHHHHH
Máscara caso 2	255.255.255.8	11111111.11111111. 11111111.00001000
Identificación de Host o Net		NNNNNNNN. NNNNNNNN. NNNNNNNN. HHHHNHHH

Para el **caso 1**, se crearon dos subredes (**Hasta ahora, pues aún falta un detalle más**), identificadas por el primer bit del cuarto octeto cuyo peso es de 128. Bajo este esquema lógico (humano), se identifican dos subredes cuyos límites están dados por el valor numérico (humano)

del cuarto octeto, es decir que se plantea la subred 193.66.66.1 a 127 y la subred 193.66.66.128 a 254, por lo tanto todo Host que en su cuarto octeto tenga un valor menor a 128, pertenecerá unívocamente a la primera subred y caso contrario a la segunda.

Ej:

Dirección IP 193.66.66.24 ---------- Subred 1
193.66.66.200 ---------- Subred 2
193.66.66.129 ---------- Subred 2
193.66.66.4 ---------- Subred 1
193.66.66.167 ---------- Subred 2
193.66.66.211 ---------- Subred 2

Para el **caso 2**, también se crearon dos subredes, identificadas por el quinto bit del cuarto octeto cuyo peso es 8, técnicamente podría funcionar (al igual que el caso 1), pero la identificación de esas dos subredes sólo será pensando en binario, lo cual para ninguna PC o router será una limitación, pero si lo es en gran medida para cualquier ser humano.

Ej:

Dirección IP 193.66.66.24 ---------- Subred ?
193.66.66.200 ---------- Subred ?
193.66.66.129 ---------- Subred ?
193.66.66.4 ---------- Subred ?
193.66.66.167 ---------- Subred ?
193.66.66.211 ---------- Subred ?

Por último falta aclarar un detalle más.

En el Caso 1 recientemente analizado (Dir IP : 193.66.66.240 – Máscara: 255.255.255.128), como se mencionó, se crearon dos subredes:

- subred 193.66.66.1 a 127 (*es decir su último octeto comienza con el primer bit = 0*)
- subred 193.66.66.128 a 254 (*es decir su último octeto comienza con el primer bit = 1*)

Recordando un concepto ya descripto, en un octeto la dirección de red no pueden ser todos unos (Broadcast), ni todos ceros (Mi red); en este caso el primer bit del último octeto será siempre "Todos unos" o "Todos ceros" pues es el único. Siguiendo este principio, NO SE DEBEN REALIZAR DOS SUBREDES EMPLEANDO UN SOLO BIT, por lo tanto, si se deseara implementar dos subredes en este ejemplo, la máscara será 255.255.255.192, no debiendo emplear el rango 00 ni 11 de los consecuentes primeros dos bit del octeto. La mayoría de los routers hoy en día ofrecen opciones para sí poder emplearlos, pero es una buena práctica "dimensionar" adecuadamente mis esquemas de direccionamiento para evitar el uso de los extremos.

Si se desea implementar tres subredes, la máscara será 255.255.255.224, la que también permitiría implementar hasta seis subredes (Pues no se podrían asignar los rangos 000 y 111 de los tres primeros bit del último octeto). Siguiendo este razonamiento cabe esperar que para siete subredes, la máscara debería ser 255.255.255.239 lo que también permitiría hasta catorce, y así sucesivamente.

5.1.3. Classless Interdomain Routing (CIDR) (RFC: 1518/1519):

Ante el inesperado crecimiento de Internet, se produce una saturación del rango de direcciones clase B, dejando libres algunas direcciones clase A y C, y presentando la particular característica que muy pocas empresas (o casi ninguna), puede cubrir una clase A completa, y muchas necesitan más de una clase C. Ante este hecho, se fueron tomando una serie de medidas por medio de las cuales se ajusta la distribución, se restringe la asignación de direcciones a empresas que lo justifiquen con mucho grado de detalle, se distribuyen direcciones en cinco zonas mundiales (RIPE, ARIN, APNIC, AfriNIC y LACNIC), pero esto comienza a provocar cada vez mayores tablas en los router con el consiguiente cuello de botella. Para presentar una solución a este problema (momentánea, pues la definitiva recién aparece con IPv6), nace CIDR o también llamado "Supernetting". Este concepto permite combinar subredes que comparten más de una clase C o subdividir redes clase A o B. El concepto de Supernetting se atribuye a la diferencia con Subnetting. En este último, para crear subredes, se emplea mayor cantidad de bit en la máscara de red. En el caso de Supernetting, es justamente lo contrario (y de ahí su nombre), pues se emplearán menos bit de máscara de la que correspondería a su clase.

Estas direcciones de red deben compartir los bit de más alto orden de la máscara de red, sin respetar el concepto clásico de "clase". A continuación se presenta un ejemplo:

NET 192.168.5 (1100 0000.1010 1000.0000 0101.0000 0000)
NET 192.168.6 (1100 0000.1010 1000.0000 0110.0000 0000)
NET 192.168.7 (1100 0000.1010 1000.0000 0111.0000 0000)
MASK 255.255.252.0 (1111 1111.1111 1111.1111 1100.0000 0000)

En este ejemplo se aprecian tres rangos de direcciones de red que clásicamente se definirían como clase C, el empleo de CIDR, se pone de manifiesto a través de la máscara de red, la cual reduce dos bit, colocando en su tercer octeto el valor 252 en vez del clásico que debería ser 255. Si se analiza la combinación de bit de dirección con los de host, se trataría aquí de la red 192.168.4, y el broadcast de red debería ser 192.168.127.255 **(11000000 . 10101000 . 00000111 . 11111111).** Las siguiente RFC hacen referencia a CIDR:

RFC 1467 - Difusión de CIDR en Internet

RFC 1517 - Condiciones de aplicabilidad de CIDR

RFC 1518 - Una arquitectura para la distribución de direcciones IP con CIDR

RFC 1519 - CIDR: asignación de direcciones y estrategia de agregación

RFC 1520 - Intercambiando información de encaminamiento a través de las fronteras de los proveedores en el entorno CIDR

Otro tema que surge con motivo de la escasez de direcciones IP, es la **RFC 1918** (address Allocation for Private Internets) que se publica en febrero de 1996, en la cual se excluye de los rangos de direeccionamiento público, es decir de carácter "Universal" un cierto rango de direcciones IP que pasan a denomintarse "Privadas". Resumidamente lo que se hace es verificar que rangos quedan aún disponibles y seleccionar los siguientes de cada clase:

- Clase A: 10.0.0.0/8
- Clase B: 172. 16 a 31.0.0/12
- Clase C: 192.168.0.0/24

Concretamente esta RFC los define en el punto 3. Private Address Space

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

10.0.0.0 - 10.255.255.255 (10/8 prefix)
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

Aclarando a su vez que:

Routers in networks not using private address space, especially those of Internet service providers, are expected to be configured to reject (filter out) routing information about private networks.

Es decir, todo Internet Service Provider (ISP) está obligado a rechazar y/o filtrar cualquier IP dentro de estos rangos.

5.1.4. Network Address Translation (NAT):

Sin lugar a dudas, lo que sustentó a la **RFC 1918** en cuanto a las direcciones IP privadas, fue el nacimiento de la técnica de **NAT**, y entre ambas prolongaron la vida de IP versión 4 hasta la actualidad (hecho que en su momento era inimaginable).

La lógica del funcionaiento de NAT, se basa en el mantenimiento de tablas en la memoria caché del host que lo imlementa, en las cuales asocia IP y puerto (orígen) con IP y puerto (destino), esta concatenación de IP-Puerto conforman lo que se denomina "**Socket**" y cuando tratemos el nivel de "transporte" se concretará esta idea. Un host que hace NAT, es una especie de pasarela que tiene una tarjeta conectada a una red (supongamos la privada, con sus direcciones IP privadas) y otra tarjeta conectada a otra red (supongamos a Internet) y habitualmente es el que se configura como **"Gateway por defecto"** en los hosts de la red

interna. Cuando este host recibe una petición desde una IP interna (esta a su vez abrió un puerto origen para esa conexión) hacia un puerto e IP externo, el dispositivo que hace NAT, automáticamente almacena en su tabla NAT este "socket", y saca por la interfaz de red externa un nuevo datagrama IP con un puerto origen que abre él y con su propia IP origen (conectada a la red externa) hacia el destino indicado, por lo tanto en realidad se establece una nueva conexión "extremo a extremo" hacia ambos lados. Al recibir la respuesta, mira en su tabla cuál fue el "Socket" interno que realizó esta petición, y le dirige a este un nuevo datagrama con las IP-Puerto "originales" a los cuales se había realizado la petición, es decir que para el usuario interno, este NAT es transparente, sólo puede percibirlo, porque la dirección MAC que envió esta respuesta es la del "Gateway por defecto" (en este caso el dispositivo que hizo NAT).

Se pueden diferenciar varios casos diferentes de NAT, de los cuales los más destacados son:

- **Estático**: Una única dirección privada se traduce a una única dirección pública. Este caso se suele emplear cuando se posee un servidor al que se puede llegar desde Internet pero que posee una dirección IP interna, y en este caso en NAT lo asocia con una dirección pública a la cual se puede acceder desde cualquier lugar del mundo.
- **Dinámico**: Se mapean direcciones desde uan red a otra en relaciones: 1 a "n", "n" a 1 o "n" a "n". El empleo más frecuente de este es cuando a través de una única o muy pocas direcciones públicas, navega por Internet toda la LAN de una empresa. Este tipo se denomina Network Address Port Translation (**NAPT**).

Ambos tipos de NAT son denominados "NAT Tradicional", y la RFC que trata este tema es la **RFC-3022** "Traditional IP Network Address Translation" (IP NAT tradicional) de enero de 2001. Esta RFC hace mención también al protocolo ICMP dentro del esquema NATP, este protocolo también puede ser enrutado, pero en vez del concepto de "socket" en este caso se concatena la dirección IP con el "identificador" del encabezado ICMP, y el resto opera de forma análoga. También describe varios casos, ejemplos y escenarios sobre los que se puede aplicar NAT.

Otra RFC que debe ser tenida en cuenta es la **RFC-2663** "IP Network Address Translator (NAT) Terminology and Considerations" que nos describe todos los términos y aspectos fundamentales y es de agosto de 1999.

5.1.5. Tablas de ruta:

Cada datagrama tiene sólo tres posibilidades:

- Ser pasado al nivel superior.
- Encaminarlo hacia alguna de las interfaces de red.
- Ser descartado.

Las tablas de rutas mantienen cuatro tipos de ellas:

- host (Se trata de una ruta a una simple y específica dirección IP).
- Subnet (Ruta hacia una subred).
- Network (Ruta hacia toda una red).
- Default (Cuando ninguna de las anteriores coincide).

Ejemplo:

```
C:\>route print
Network Address    Netmask          Gateway Address     Interface       Metric
0.0.0.0            0.0.0.0          192.168.40.1        192.168.40.123  1
127.0.0.0          255.0.0.0        127.0.0.1           127.0.0.1       1
192.168.40.0       255.255.255.0    192.168.40.123      192.168.40.123  1
192.168.40.123     255.255.255.255  127.0.0.1           127.0.0.1       1
192.168.40.255     255.255.255.255  192.168.40.123      192.168.40.123  1
224.0.0.0          224.0.0.0        192.168.40.123      192.168.40.123  1
255.255.255.255    255.255.255.255  192.168.40.123      192.168.40.123  1
```

En este ejemplo de tabla de ruteo (en Windows), se aprecia una dirección privada clase C con un host cuya dirección es 192.168.40.123 y contiene siete entradas:

- La primera entrada (0.0.0.0) contiene la ruta por defecto.
- La segunda (127.0.0.0) es la dirección de loopback.
- La tercera (192.168.40.0) es la dirección de esta red.
- La cuarta (192.168.40.123) es la ruta para el local host, se debe prestar atención, que esta hace referencia luego al loopback, es decir que todo datagrama hacia esta dirección deberá ser tratado internamente.
- La quinta (192.168.40.255) especifica el broadcast de subred.
- La sexta especifica la dirección multicast.
- La última indica la dirección de broadcast.

Esta tabla de rutas en la mayoría de los casos es mantenida automáticamente, y muchas de estas direcciones las obtendrá al iniciar el host, de datos locales y a través de información obtenida desde servidores de la red, también pueden ser adicionadas en forma manual, y especificar que sean permanentes o transitorias.

Detección de direcciones IP duplicadas:

Al iniciarse un host, envía una solicitud ARP particular, pues lo hace solicitando la dirección MAC que se corresponde a su propia dirección IP, si alguien contesta este mensaje es que esta dirección IP ya está en uso.

Multihomed:

Este término se refiere al caso que un host se encuentre configurado con más de una dirección IP, esta configuración se puede presentar de tres maneras:

- Múltiples IP por NIC.
- Múltiples NIC.
- Múltiples IP y NIC.

5.1.6. IP Multicasting:

El IP Multicasting es la transmisión de un datagrama IP a un cierto grupo de cero a "n" hosts identificados por una única dirección IP. Los miembros de este grupo son dinámicos, es decir que pueden unirse y dejar grupos en cualquier momento, sin existir ninguna restricción de ubicación o cantidad de miembros. Un host puede ser miembro de más de un grupo y no necesita ser miembro de un grupo para enviar datagramas a él.

Existen grupos definidos como permanentes, que son los que tienen asignada una dirección IP "Bien conocida", esta permanencia se refiere a su dirección, no a sus miembros, los cuales pueden incorporarse y dejarlo dinámicamente en cualquier momento.

El empleo de IP multicast está controlado por los router que pueden o no coexistir con los Gateway.

En el mapa de rutas de cada host, para soportar multicast, se agrega una línea adicional:

Network Address	Netmask	Gateway Address	Interface	Metric
224.0.0.0	224.0.0.0	192.35.129.1	192.35.129.1	1

En este ejemplo, la dirección 192.35.129.1 es la propia dirección del host (no la del gateway de la red), es decir que si este host desea enviar cualquier mensaje multicast, lo hará a través de su interfaz de red y no lo encaminará al gateway por default.

Un tema de especial interés es que para poder enviar un datagrama, siempre debe ser resuelta la dirección MAC, este tema lo trata la **RFC 1112** y se implementa de la siguiente manera: Ethernet identifica multicast colocando a uno el primer bit del primer octeto de su dirección MAC, es decir que este primer octeto adoptará el valor 01h, además IANA reserva (en acuerdo con IEEE que es quien asigna las direcciones MAC) el rango 00-00-5E-00-00-00 hasta 00-00-5E-7F-FF-FF, por lo tanto, los tres primeros octetos de la dirección MAC para multicast IP en redes Ethernet será siempre 01-00-5E. El grupo de direcciones IP es mapeado a una dirección multicast Ethernet, por reemplazo de los 23 bit de menor orden de la dirección IP a los 23 bit de menor orden de la dirección MAC, por ejemplo si un host con dirección MAC 01-22-A2-34-67-E1 deseara enviar un datagrama a la dirección multicast 224.0.0.3, la dirección MAC se formaría con los tres primeros octetos MAC multicast (01-00-5E), anexándole los últimos 23 bit de la dirección IP multicast (0.0.3 = 00-00-03 en hexadecimal), quedaría 01-00-5E-00-00-03. Un problema cierto que se plantea es que la dirección IP posee

cuatro octetos, es decir 32 bit, en el caso de multicast, los cuatro primeros bit están impuestos y siempre son 1110, por lo tanto quedan disponibles veintiocho bit, de los cuales solo se mapean 23, es decir que se puede presentar el caso que dos direcciones multicast IP sean mapeadas a la misma dirección MAC, en este caso, será descartado el datagrama a nivel IP, al descubrir en destino que no va dirigido al grupo correspondiente.

Por defecto los datagramas IP Multicast son generados con TTL=1, y por convención, los router multicast emplean los valores de TTL para determinar cuán lejano deben encaminar los datagramas, estos valores de TTL están definidos y son los siguientes:

TTL = 0 se refieren al mismo host.

TTL = 1 se refieren a la misma subred. Los routers multicast al recibir un datagrama multicast con TTL=1, lo decrementan a cero, pero a diferencia de un datagrama unicast, no envía ningún mensaje ICMP de TTL agotado, pues lo considera un evento normal.

TTL = 32 se refieren a la misma Site.

TTL = 64 se refieren a la misma región.

TTL = 128 se refieren al mismo continente.

TTL = 255 sin restricciones de ámbito.

La dirección multicast también ofrece información sobre las rutas, pues por ejemplo el rango comprendido entre 224.0.0.0 a 224.0.0.255 se emplea para un solo salto, es decir que ningún router multicast retransmitirá ningún datagrama con estas direcciones.

A continuación se presentan las direcciones multicast definidas por la RFC 1112:

224.0.0.0 Reserved
224.0.0.1 All Hosts on this Subnet
224.0.0.2 All Gateways on this Subnet (proposed)
224.0.0.3 Unassigned
224.0.0.4 DVMRP Routers(Distance Vector Multicast Routing Protocol, RFC 1075)
224.0.0.5 OSPFIGP OSPFIGP All Routers
224.0.0.6 OSPFIGP OSPFIGP Designated Routers
224.0.0.7-224.0.0.255 Unassigned
224.0.1.0 VMTP Managers Group (Versatile Message Transaction Protocol, RFC 1045).
224.0.1.1 NTP Network Time Protocol
224.0.1.2 SGI-Dogfight
224.0.1.3 Rwhod
224.0.1.4 VNP
244.0.1.5- 244.0.1.255 Unassigned
224.0.2.1 “rwho” Group (BSD) (unofficial)
232.x.x.x VMTP transient groups

5.1.7. Fragmentación IP:

Como se verá en el encabezado de IP, uno de los puntos fuertes de este protocolo es la fragmentación, si bien en la actualidad no es muy empleado porque TCP puede determinar los tamaños de la información a transmitir por el canal para que justamente no sea necesario tener que fragmentar y defragmentar información en nodos intermedios. Igualmente este aspecto se expresa particularmente en este párrafo para marcar la importancia de esta tarea pues es de las técnicas más empleadas para evadir sistemas de seguridad en las redes TCP/IP.

5.1.8. Formato del encabezado (datagrama) IP (Según RFC 791):

<table>
<tr><td colspan="3">Versión</td><td colspan="3">Longitud de cabecera</td></tr>
<tr><td colspan="2">precedencia</td><td>D</td><td>T</td><td>R</td><td>Reservado</td></tr>
<tr><td colspan="6">Longitud total</td></tr>
<tr><td colspan="6">Identificador</td></tr>
<tr><td colspan="2">Identificadores</td><td colspan="4">Desplazamiento de fragmentación</td></tr>
<tr><td colspan="6">Tiempo de vida (TTL)</td></tr>
<tr><td colspan="6">Protocolo</td></tr>
<tr><td colspan="6">Checksum de cabecera</td></tr>
<tr><td colspan="6">Dirección Fuente</td></tr>
<tr><td colspan="6">Dirección Destino</td></tr>
<tr><td colspan="6">Opciones y Relleno (Variable)</td></tr>
<tr><td colspan="6">Datos (Variable)</td></tr>
</table>

Versión: 4 bits

Siempre vale lo mismo (0100). Este campo describe el formato de la cabecera utilizada. En la tabla se describe la versión 4.

Tamaño Cabecera: 4 bits

Longitud de la cabecera, en palabras de 32 bits. Su valor mínimo es de 5 para una cabecera correcta, y el máximo de 15.

Tipo de Servicio: 8 bits (Precedencia – DTR – Reservado) (Se verá más adelante que actualmente están modificados)

Indica una serie de parámetros sobre la calidad de servicio deseada durante el tránsito por una red. Algunas redes ofrecen prioridades de servicios, considerando determinado tipo de paquetes "más importantes" que otros (en particular estas redes solo admiten los paquetes con prioridad alta en momentos de sobrecarga). Estos 8 bits se agrupan de la siguiente manera. Los 5 bits de menos peso son independientes e indican características del servicio:

Bit 0: sin uso, debe permanecer en 0.

Bit 1: 1 costo mínimo, 0 costo normal.

Bit 2: 1 máxima fiabilidad, 0 fiabilidad normal.

Bit 3: 1 maximo rendimiento, 0 rendimiento normal.

Bit 4: 1 mínimo retardo, 0 retardo normal.

Los 3 bits restantes están relacionados con la precedencia de los mensajes, un indicador ajunto que indica el nivel de urgencia basado en el sistema militar de precedencia (véase Message Precedence) de la CCEB, un organización de comunicaciones electrónicas militares formada por 5 naciones. La urgencia que estos estados representan aumenta a medida que el número formado por estos 3 bits lo hace, y responden a los siguientes nombres.

000: De rutina.

001: Prioritario.

010: Inmediato.

011: Relámpago.

100: Invalidación relámpago.

101: Procesando llamada crítica y de emergencia.

110: Control de trabajo de Internet.

111: Control de red.

Longitud Total: 16 bits

Es el tamaño total, en octetos, del datagrama, incluyendo el tamaño de la cabecera y el de los datos. El tamaño máximo de los datagramas usados normalmente es de 576 octetos (64 de cabeceras y 512 de datos). Una máquina no debería envíar datagramas mayores a no ser que tenga la certeza de que van a ser aceptados por la máquina destino.

En caso de fragmentación este campo contendrá el tamaño del fragmento, no el del datagrama original.

Identificador: 16 bits

Identificador único del datagrama. Se utilizará, en caso de que el datagrama deba ser fragmentado, para poder distinguir los fragmentos de un datagrama de los de otro. El originador del datagrama debe asegurar un valor único para la pareja origen-

destino y el tipo de protocolo durante el tiempo que el datagrama pueda estar activo en la red.

Indicadores: 3 bits

Actualmente utilizado sólo para especificar valores relativos a la fragmentación de paquetes:

bit 0: Reservado; debe ser 0

bit 1: 0 = Divisible, 1 = No Divisible

bit 2: 0 = Último Fragmento, 1 = Fragmento Intermedio (le siguen más fragmentos)

La indicación de que un paquete es indivisible debe ser tenida en cuenta bajo cualquier circunstancia. Si el paquete necesitara ser fragmentado, no se enviará.

Posición de Fragmento: 13 bits

En paquetes fragmentados indica la posición, en unidades de 64 bits, que ocupa el paquete actual dentro del datagrama original. El primer paquete de una serie de fragmentos contendrá en este campo el valor 0.

Tiempo de Vida (TTL): 8 bits

Indica el máximo número de direccionadores que un paquete puede atravesar. Cada vez que algún nodo procesa este paquete disminuye su valor en, como mínimo, un direccionador. Cuando llegue a ser 0, el paquete no será reenviado.

Protocolo: 8 bits

Indica el protocolo de siguiente nivel utilizado en la parte de datos del datagrama. Vea Números de protocolo IP para comprender como interpretar este campo.

Suma de Control de Cabecera: 16 bits

Se recalcula cada vez que algún nodo cambia alguno de sus campos (por ejemplo, el Tiempo de Vida). El método de cálculo (intencionadamente simple) consiste en sumar el complemento a 1 de cada palabra de 16 bits de la cabecera y hacer el complemento a 1 del valor resultante.

Dirección IP de origen: 32 bits

Dirección IP de destino: 32 bits

Opciones: Variable

Aunque no es obligatoria la utilización de este campo, cualquier nodo debe ser capaz de interpretarlo. Puede contener un número indeterminado de opciones, que tendrán dos posibles formatos:

Formato de opciones simple

Se determina con un sólo octeto indicando el Tipo de opción, el cual está dividido en 3 campos.

- Indicador de copia: 1 bit. En caso de fragmentación, la opción se copiará o no a cada nuevo fragmento según el valor de este campo:

0 = no se copia

1 = se copia.

- Clase de opción: 2 bits. Las posibles clases son:

0 = control

1 = reservada

2 = depuración y mediciones

3 = reservada.

- Número de opción: 5 bits. Identificador de la opción.

Formato de opciones compuesto

Un octeto para el Tipo de opción, otro para el Tamaño de opción, y uno o más octetos conformando los Datos de opción.

El Tamaño de opción incluye el octeto de Tipo de opción, el de Tamaño de opción y la suma de los octetos de datos.

La siguiente tabla muestra las opciones actualmente definidas:

Clase	Número	Tamaño	Descripción
0	0	-	Final de lista de opciones. Formato simple.
0	1	-	Ninguna operación (NOP). Formato simple.
0	2	11	Seguridad.
0	3	variable	Enrutado desde el Origen, abierto (Loose Source Routing).
0	9	variable	Enrutado desde el Origen, estricto (Strict Source Routing).
0	7	variable	Registro de Ruta (Record Route).
0	8	4	Identificador de flujo (Stream ID).
2	4	variable	Marca de tiempo (Internet Timestamping).

Final de Lista de Opciones:

Se usa al final de la lista de opciones, si ésta no coincide con el final de la cabecera IP.

Ninguna Operación (NOP):

Se puede usar para forzar la alineación de las opciones en palabras de 32 bits.

Seguridad:

Especifica niveles de seguridad que van desde “No Clasificado” hasta “Máximo Secreto”, definidos por la Agencia de Seguridad de la Defensa (de EE.UU.).

Enrutado desde el Origen (abierto) y Registro de Ruta (LSSR):

Esta opción provee el mecanismo para que el originador de un datagrama pueda indicar el itinerario que ha de seguir a través de la red y para registrar el camino seguido.

Los Datos de Opción consisten en un puntero (un octeto) y una lista de direcciones IP (4 octetos cada una) que se han de alcanzar ("procesar"):

El puntero indica la posición de la siguiente dirección de la ruta, dentro de la Opción; así, su valor mínimo es de 4.

Cuando un nodo de Internet procesa la dirección de la lista apuntada por el puntero (es decir, se alcanza esa dirección) incrementa el puntero en 4, y redirige el paquete a la siguiente dirección. Si el puntero llega a ser mayor que el Tamaño de Opción significa que la información de ruta se ha procesado y registrado completamente y se redirigirá el paquete a su dirección de destino.

Si se alcanza la dirección de destino antes de haber procesado la lista de direcciones completa (el puntero es menor que el Tamaño de Opción) la siguiente dirección de la lista reemplaza a la dirección de destino del paquete y es a su vez reeemplazada por la dirección del nodo que está procesando el datagrama ("Ruta Registrada"), incrementando, además, el puntero en 4.

Utilizando este método de sustituir la dirección especificada en origen por la Ruta Registrada se asegura que el tamaño de la Opción (y de la cabecera IP) no varía durante su recorrido por la red.

Se considera que la ruta especificada por el originador es "abierta" porque cualquier nodo que procesa el paquete es libre de dirigirlo a la siguiente dirección siguiendo cualquier otra ruta intermedia.

Sólo puede usarse una vez en un datagrama, y, en caso de fragmentación, la opción se copiará a los paquetes resultantes.

Enrutado desde el Origen (estricto) y Registro de Ruta (SSRR):

Exactamente igual que LSSR, excepto en el tratamiento que los nodos harán de este datagrama. Al ser la ruta especificada "estricta", un nodo debe reenviar el paquete directamente a la siguiente dirección, es decir, no podrá redireccionarlo por otra red.

Registro de Ruta:

Mediante el uso de esta Opción se puede registrar el itinerario de un datagrama. Los Datos de Opción consisten en un puntero (un octeto) y un espacio relleno de ceros que contendrá la Ruta Registrada para el paquete.

Cuando un nodo recibe un paquete en el que está presente esta opción, escribirá su dirección IP en la posición indicada por el puntero, siempre que ésta sea menor que el Tamaño de Opción, e incrementará el puntero en 4.

Es preciso que el espacio reservado para la Ruta Registrada tenga una longitud múltiplo de 4; si al intentar grabar su dirección un nodo detecta que existe espacio libre pero es menor de 4 octetos, el paquete no se reenvía (se pierde) y se notifica el error, mediante ICMP, al originador del datagrama.

Esta Opción no se copia en caso de fragmentación, y sólo puede aparecer una vez en un paquete.

Relleno: Variable

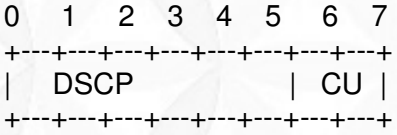
Utilizado para asegurar que el tamaño, en bits, de la cabecera es un múltiplo de 32. El valor usado es el 0.

5.1.9. Modificaciones DSCP {RFCs 2474 (DS: Servicios Diferenciados) y 3168 (DSCP: Sevicios diferenciados Codificación)}

RFC 2474 (Differentiated Services) Interpretation

Bits	Meaning
7-2	DSCP
1-0	ECN (Explicit Congestion Notification)

The DS field structure is presented below:



DSCP: differentiated services codepoint
CU: currently unused

Tabla de Conversión de DSCP a IP Precedencia:

DS Valor	Binario	Decimal	IP Precedencia (Viejo)
CS0	000 000	0	0
CS1	001 000	8	1
AF11	001 010	10	1
AF12	001 100	12	1
AF13	001 110	14	1
CS2	010 000	16	2
AF21	010 010	18	2
AF22	010 100	20	2
AF23	010 110	22	2
CS3	011 000	24	3
AF31	011 010	26	3
AF32	011 100	28	3
AF33	011 110	30	3
CS4	100 000	32	4
AF41	100 010	34	4
AF42	100 100	36	4

AF43	100 110	38	4
CS5	101 000	40	5
EF	101 110	46	5
CS6	110 000	48	6
CS7	111 000	56	7

CS: Class Selector (RFC 2474) Los 3 primeros bits. (111 preferente – 000 menos)

Afxy: Assured Forwarding (x=class, y=drop precedence) (RFC2597) los tres que siguen.

EF: Expedited Forwarding (RFC 3246)

La **RFC 2597** (Assured Forwarding (AF) Per Hop Behavior (PHB) Group) establece los soigueintes valores de AF:

Los valores recomendados del "AF codepoints" son los siguientes:

AF11 = '001010',
AF12 = '001100',
AF13 = '001110',
AF21 = '010010',
AF22 = '010100',
AF23 = '010110',
AF31 = '011010',
AF32 = '011100',
AF33 = '011110',
AF41 = '100010',
AF42 = '100100',
AF43 = '100110'.

	Class 1	Class 2	Class 3	Class 4
Low Drop Prec	001010	010010	011010	100010
Medium Drop Prec	001100	010100	011100	100100
High Drop Prec	001110	010110	011110	100110

5.1.10. IP Spoof:

Esta técnica se emplea para falsificar un verdadera dirección IP a través de la colocación de una falsa, la cual puede hacer uso de una existente en la red o nueva. Presenta especial peligrosidad cuando los dispositivos de seguridad emplean el campo dirección IP para la implementación de las medidas de sus medidas, como pueden ser: control de accesos, permisos, ámbito interno o externo, validación o generación de alarmas, establecimiento de sesiones, reglas, etc.

5.2. ICMP (Internet Control Messaging Protocol) (RFC: 792).

Este protocolo es quizás uno de los más importantes de esta pila pues es el que se encarga de la supervisión y control de la red. Un datagrama viaja entre router a través de la red hasta alcanzar su destino, si ocurre algún error o para controlar esta travesía es que se generan estos mensajes. Este sistema de reportes, es tratado por la red como cualquier otro datagrama (Nivel 3), pero el Software de la capa 3 los interpreta de manera distinta. ICMP no especifica las acciones a tomar, solamente sugiere la misma.

5.2.1. Tipos y códigos de los mensajes ICMP:

Todas las cabeceras de ICMP comienzan con tres campos:

- TIPO: (8), especifica el mensaje ICMP.
- CODIGO: (8), Brinda un poco más de información sobre el error.
- CHECKSUM (16), CRC 16.

Los campos que continúan a estos tres, varían acorde al tipo de error, pero en la mayoría de ellos se encuentra incluido el encabezado del datagrama que generó el mensaje ICMP y también los 64 primeros octetos de este para dejar unívocamente establecido la identificación del error.

El estudio del funcionamiento del protocolo ICMP se puede entender básicamente desarrollando el significado del campo TIPO, el cual representa los distintos tipos de mensajes.

5.2.2. Tipos y códigos de los mensajes ICMP:

- **0 y 8**: Eco de solicitud y de respuesta: No es ni más ni menos que el comando PING, que genera una solicitud y una respuesta (configurable), para determinar la continuidad del recorrido de un datagrama a lo largo de una red, su cantidad de saltos y el tiempo demorado.
- **3**: Destino no alcanzable: Se genera cuando un datagrama no encuentra la dirección IP destino. También ocurre cuando el bit de no fragmentar de la cabecera IP esta puesto en 1, y la red destino no soporta bloques del tamaño de ese datagrama, por lo cual no podrá ser entregado a esa red, causando la no llegada a destino. Dentro de este tipo es interesante tener en cuenta el campo Código, pues brinda información adicional sobre las causas por las cuales no se llega a destino, en particular si lo que se desea es obtener información sobre ese extremo (Extremadamente usado para ataques a redes). Los valores que toma son:

-0: Red inalcanzable.

-1: Host inalcanzable.

-2: Protocolo inalcanzable.

-3: Puerto inalcanzable.

-4: Fragmentación requerida y bit de no fragmentar puesto a 1 en el datagrama origen.

-5: Falla en la ruta.

-6: Red desconocida.

-7: Host desconocido.

-8: Host origen aislado.

-9: Acceso a la red administrativamente prohibido.

-10: Acceso al Host administrativamente prohibido.

-11: Red inalcanzable por tipo de servicio.

-12: Host inalcanzable por tipo de servicio.

- **4**: Fuente agotada: Sirve para regular el flujo de información. Implica un buffer lleno, causa por la cual sería conveniente que el Host transmisor dejara de hacerlo hasta que deje de recibir estos mensajes.
- **11**: Tiempo de vida excedido: El campo TTL llegó a 0.
- **5**: Se requiere redireccionamiento: Existe una ruta mejor.
- **12**: Problemas con el parámetro: Error semántico o sintáctico en el encabezamiento IP.
- **13 y 14**: Solicitud y respuesta de marcador de tiempo: Permite la sincronización de clock entre nodos, a través de la hora GMT (Greenwich Mean Time).
- **15 y 16**: Solicitud y repuesta de información: Permite obtener información de un nodo. Este fue originariamente pensado para los protocolos BOOTP y R_ARP.
- **17 y 18**: Solicitud y respuesta de máscara de dirección: Permite determinar las máscaras de las redes con que está conectada un nodo. Se emplea para el ruteo hacia esas redes.

Desde el punto de vista de la seguridad, este protocolo es fundamental, pues así como nos facilita el control y monitorización de la red, también ofrece información muy valiosa para quien la esté buscando. En el párrafo anterior, citamos únicamente los tipos y códigos más importantes, pero hay más. Sobre los anteriores ya podemos comenzar a plantearnos, por ejemplo:

- No es lo mismo saber que un "Destino es inalcanzable" (Tipo 3), que saber que ese "destino es inalcanzable porque 9: el "Acceso a la red o al host está administrativamente prohibido (Códigos 9 o 10), pues eso me informa sin lugar a dudas, que el datagrama está llegando, pero hay algún "administrador" que no nos quiere dejar llegar.....10: Acceso al Host administrativamente prohibido, o que estoy llegando, pero el "protocolo o el puerto" (códigos 2 o 3), eso sería una clara indicación que es a nivel 4 o 5 del modelo de capas donde se está impidiendo el paso. Tampoco

es lo mismo que nos indiquen que no llego a la red o al host (Códigos 0 o 1), y así podríamos seguir con más detalle.

- Un mensaje de fuente agotada (tipo 4), nos está indicando que estamos a un ápice de desbordar el buffer de ese dispositivo, o al menos que ya empezó a descartar datagramas (situación ideal para un ataque de negación de servicio)
- El tipo 5 (se requiere redireccionamiento) es la situación más buscada para el ataque del hombre del medio a nivel IP.....

Hemos querido presentar someramente algunos de los mensajes y empleos que ofrece este protocolo, para el lado del "bien" y para el lado del "mal", pues insistimos que es muy importante, pero no es motivo de esta capítulo profundizar sobre esto. Luego en la PARTE II se desarrollará con más detalle lo relacionado a seguridad de este protocolo.

5.3. IGMP (Internet Group Messaging Protocol) (RFC 1112).

Este protocolo es muy similar al anterior, pero está pensado para mensajes entre grupos de host empleando también los datagramas IP para transportar su información.

5.3.1. Multicast IP sobre Ethernet:

Los primeros cuatro bit de una dirección IP si se encuentran como 1110 identifican una dirección Tipo D o Dirección de multicast, los restantes 28 bit identificarán a que grupo se refiere, dentro de este esquema se debería comenzar con 224.0.0.0 la cual no se emplea por ser reservada. La siguiente es 224.0.0.1 que define a todos los grupos de host y router que participan en multicast.

Una dirección multicast, se debe aclarar que jamás puede definir una dirección origen, siempre se referirá a una destino.

Dentro de un esquema Ethernet, para que una dirección de multicast de nivel IP pueda ser entregado a los equipos deseados, es imprescindible que el nivel 2 sepa identificarlos para que de alguna manera no los descarte en ese nivel. Para que esto suceda, una solución podría ser un Broadcast de nivel 2, ante lo cual todas los Host de esa LAN lo reconocerían como propio, lo desencapsularían y entregarían el datagrama al nivel 3 (IP). Esta opción desde ya desperdicia bastante el esquema de direccionamiento de Ethernet, pues lo ideal sería que sólo sea tenido en cuenta por los Host que integran el grupo de multicast de nivel IP. Para implementar este procedimiento Ethernet ubica los últimos tres octetos de la dirección IP en los mismos últimos tres de la dirección NIC o MAC de este nivel en una dirección grupal específica que es 01-00-5E-00-00-00.

Ej: Si el multicast IP fuera 224.0.0.1 la dirección Ethernet es 01-00-5E-00-00-01

Los Host que participan de un esquema de multicast IP, pueden desempeñar tres niveles diferentes:

- Nivel 0: El Host no puede recibir ni enviar IP Multicast.
- Nivel 1: El Host no puede recibir pero puede enviar IP Multicast.
- Nivel 2: El Host puede recibir y enviar IP Multicast.

5.3.2. Fases de IGMP:

IGMP posee dos fases:

- Fase 1: Cuando un Host se incorpora a un grupo de multicast enviando un mensaje a todos los Host del Grupo, anunciándose como miembro. Los router locales reciben este mensaje y lo propagan al resto de los miembros.
- Fase 2: Como los grupos son dinámicos, los router locales periódicamente sondean (Poll) los host para mantener el estado de los grupos.

5.3.3. Formato del mensaje IGMP:

4	4	8	16
Versión	Tipo	Reservado	CRC
Direcciones de Grupo			

- Versión: La versión actual es la 1.
- Tipo: Posee dos valores, Consulta de un router (1), y respuesta enviada por un host (2).
- Reservado: Debe estar puesto a cero.
- CRC: Control de error de los 8 octetos.
- Grupo de direcciones: Reporte de los miembros de un grupo multicast. Si es una consulta debe ir puesto a cero.

Existe un protocolo para transmitir información de ruteo entre grupos multicast que emplea el algoritmo de vector distancia y se llama DVMRP (Distance Vector Multicast Routing Protocol).

5.4. DHCP Dynamic Host Configuration Protocol (RFC 1541, 1531, 1533 y 1534).

5.4.1. Evolución de los protocolos dinámicos (ARP, BOOTP):

Este protocolo es una evolución natural del BOOTP (BOOTstrap Protocol) que fue la primera implementación de protocolos de inicio para máquinas que no poseen disco rígido, las cuales al ser encendidas, deben primero hacerse presentes en la red y luego cargar el sistema operativo. Para automatizar este proceso IETF desarrollo este nuevo protocolo conocido como DHCP. Este último introduce dos grandes mejoras respecto al anterior:

- Primera: Permite a una máquina obtener toda la información necesaria en un solo mensaje.
- Segunda: Permite obtener una dirección IP rápida y dinámicamente.

5.4.2. Pasos de la asignación dinámica:

Todo cliente DHCP puede encontrarse en seis estados diferentes:

- Inicialización: Aún no posee ninguna dirección IP, para lo cual deberá contactar a todos los servidores DHCP en su red local. Para hacer esto generará un mensaje de descubrimiento DHCP.
- Selección: Espera una respuesta (Oferta DHCP) del primer servidor y lo elige.
- Solicitud: Ingresa a este estado cuando envía al servidor seleccionado un mensaje de solicitud DHCP.
- Enlace: Al recibir el ACK del servidor con la dirección solicitada, la cual ya queda asignada en forma definitiva por el lapso correspondiente.
- Renegociación: Al generar un mensaje de renegociación.
- Reenlace: Al generar un mensaje de reenlace.

Al ingresar una máquina al estado de enlace, inicia tres Timer de control de asignación, que en general son asignados por el servidor durante la asignación de direcciones:

- Renegociación: Por defecto es la mitad del intervalo de duración de la dirección asignada. Al expirar este tiempo el cliente debe renegociar su dirección IP.
- Reenlace: Por defecto expira al 87,5 % del tiempo de asignación. Al llegar a este tiempo, el cliente asume que el servidor se encuentra inactivo, y genera un mensaje Broadcast de reenlace
- Expiración: Expira si no recibe respuestas de ningún servidor y vence su tiempo de asignación.

El empleo de DHCP no genera un uso significativo del ancho de banda durante los períodos de uso. Para iniciarse un Host y obtener su dirección IP se generan cuatro tramas de 342 Byte cada una entre host-servidor:

a. DHCP descubrimiento (Búsqueda): Esta primera trama generada por el cliente es un broadcast para ubicar un servidor DHCP, pues el cliente no tiene conocimiento de dónde se encuentra este servicio.

b. DHCP oferta: Una vez que un servidor DHCP ha recibido una trama de descubrimiento y determinado que puede responder a esta, responde entregando una dirección IP dentro del rango que este servidor tiene asignada. Esta trama también es broadcast pues el host aún no tiene dirección IP.

c. DHCP solicitud: El cliente seleccionará la primer dirección IP que reciba (tener presente que si existe más de un servidor DHCP responderán todos), y solicitará al servidor el empleo de esa dirección. Esta también es Broadcast pues aún no está confirmada su dirección IP.

d. DHCP ACK: Una vez que el servidor recibe la respuesta, envía el Acknowledgement, el tiempo de vida de esa dirección y los parámetros necesarios de TCP/IP. Esta última será Broadcast también pues es el paso final de la negociación aún no cerrada.

Ej:	1	3.080	cliente1	*BROADCAST	DHCP	Discover
	2	3.155	Serv_DHCP	*BROADCAST	DHCP	Offer
	4	3.190	cliente1	*BROADCAST	DHCP	Request
	5	3.320	Serv_DHCP	*BROADCAST	DHCP	ACK

Un cliente DHCP renegocia su dirección IP antes de que esta expire. Esta renegociación solamente consiste en dos tramas también de 342 Byte cada una. Esta renegociación puede ocurrir de dos modos diferentes:

a. Al iniciar un cliente que aún posee una dirección IP cuyo tiempo de vida no expiró, envía una trama de solicitud al servidor DHCP. Esta trama es Broadcast, pues este host fue apagado, pudiendo no encontrarse ahora en la misma subred. Si el servidor encuentra satisfactoria la solicitud, responderá con una trama ACK de la misma forma que un inicio de host.

b. Al ocurrir la mitad del tiempo de vida de su propia dirección IP, el cliente inicia una renegociación. Esta es dirigida a la misma dirección IP del servidor que la otorgó.

En ambos casos, el cliente intentará renegociar dos veces, si no obtiene su trama ACK, al expirar el tiempo de vida se producirán las cuatro tramas habituales de obtención de dirección IP.

Ej:	1	5.200	Cliente1	Serv_DHCP	DHCP	Request
	2	5.302	Serv_DHCP	Cliente1	DHCP	ACK

Las negociaciones DHCP pueden ocurrir en los siguientes casos:

a. Inicio de clientes DHCP.

b. Renegociación automática de dirección IP, la cual se produce una vez durante la mitad del tiempo de vida de esa dirección DHCP, la cual por defecto suele ser 3 días.

c. Cuando un cliente es movido a una nueva subred.

d. Cuando un cliente reemplaza su tarjeta de red.

e. Cuando un cliente manualmente refresca o renegocia su dirección IP con el comando IPCONFIG.

f. Cuando un cliente reinicia su ETD.

Los protocolos DHCP y BOOTP son compatibles, de hecho un servidor DHCP puede ser programado para responder solicitudes BOOTP, sin embargo DHCP cambia el significado de dos campos en el Header del mensaje como se verá a continuación.

5.4.3. Formato del mensaje DHCP:

8	8	8	8
OP	HTYPE	HLEN	HOPS
TRANSACTION ID			
SECONDS		FLAGS	
CLIENT IP ADDRESS			
YOUR IP ADDRESS			
SERVER IP ADDRESS			
ROUTER IP ADDRESS			
CLIENT HARDWARE ADDRESS (16 octetos)			
SERVER HOST NAME (64 octetos)			
BOOT FILE NAME (128 octetos)			
OPTIONS (Variable)			

- OP: Toma valor (1) para solicitud y (2) para respuesta.
- HTYPE:
  - (1) DHCPDISCOVER.
  - (2) DHCPOFFER.
  - (3) DHCPREQUEST.
  - (4) DHCPDECLINE.
  - (5) DHCPACK.
  - (6) DHCPNACK.
  - (7) DHCPRELEASE.
- HLEN: Especifica el tipo y longitud de la dirección de Hardware (Ej: Ethernet tiene tipo 1 y longitud 6 octetos).

- HOPS: El cliente coloca (0), si es necesario pasar a través de distintos router , el servidor BOOTP o DHCP lo incrementará.
- TRANSACTION ID: Contiene un número entero que permite llevar el control entre las solicitudes y respuestas.
- SECONDS: Determina el tiempo transcurrido desde que se inició la operación..
- FLAGS: Identifica por medio del primer bit si es un Broadcast, los restantes quince deben estar puestos a cero.
- CLIENT IP ADDRESS:
- YOUR IP ADDRESS:
- SERVER IP ADDRESS:
- ROUTER IP ADDRESS:
- CLIENT HARDWARE ADDRESS:
- SERVER HOST NAME:
- BOOT FILE NAME: Puede contener el tipo de booteo (Ej: UNIX)
- OPTIONS: Define máscara de subred, hora,etc

Hasta aquí se aprecia el aspecto general de DHCP, pero si se desea analizar en detalle su funcionamiento es necesaria remontarse a sus orígenes y tener en cuenta como nacen los protocolos de booteo (como se mencionó en la introducción).

Al inicializarse una máquina sin disco rígido, esta carga directamente de la ROM, una imagen a su memoria que le permite comenzar una secuencia de actividades. Como las direcciones IP no pueden ser impuestas por el fabricante pues justamente se trata de un esquema lógico, e inclusive en redes privadas estas pueden estar repetidas en cualquier lugar del mundo; la dirección IP debe ser solicitada a otra máquina en la red, e inclusive también necesitará una configuración particular para cada red, que también dependerá de otra máquina que podrá ser o no la misma que la anterior.

Una primera aproximación es el Protocolo R_ARP (Mencionado con anterioridad) que permite descubrir servidores y direcciones IP fuente y destino para este problema. La primera limitación está en la poca información que en sus campos se transmite (solo la IP cliente), la segunda es que es de muy bajo nivel, lo que exige a cualquier programador mucho más esfuerzo pues se debe llegar hasta el Hardware.

Para mejorar a R_ARP se diseñó BOOTP (Precursor de DHCP). La primera cualidad de este protocolo es que **emplea IP y UDP, causa por la cual tiene acceso al nivel de aplicación**, sin ninguna tarea adicional. El segundo detalle significativo es la cantidad de información que con solo dos mensajes (Solicitud y respuesta) puede transferir. Para graficar este concepto es conveniente analizar su formato (Similar a DHCP):

8 bit	8 bit	8 bit	8 bit
Operación	Tipo de Hard	Long Hard	Hops
Identificador de transacción			
Segundos		No definido	
Dirección IP cliente			
Su dirección IP			
Dirección IP del Server			
Dirección IP del Router			
Dirección de Hardware cliente (16 octetos)			
Nombre del Server (64 octetos)			
Nombre del archivo de booteo (128 octetos)			
Area específica del vendedor (64 octetos)			

- Operación: Solicitud (1), Respuesta (2).
- Tipo y Longitud de Hardware: Especifica que tipo de hardware es empleado y su longitud (Ej: Ethernet : Tipo = 1 , Long = 6).
- Hops: El cliente coloca valor 0, si se permite el booteo a través de múltiples Router, el BOOTP Server, lo incrementa en 1.
- Identificador de transacción: Contiene un número entero generado pseudo-aleatoriamente, que permitirá reconocer respuestas con solicitudes.
- Segundos: Identifica el tiempo en el cual el cliente inició su operación de booteo.
- Direcciones y nombres: Estos campos le otorgan una gran flexibilidad a este protocolo, pues permiten ser llenados con la información que se posea, y en los campos que no se conozcan se colocarán ceros. Por lo tanto puede conocer o no su dirección, la del Server, el nombre del mismo, etc.
- Area específica del vendedor: Dentro de este campo se permite por medio de un formato establecido (un octeto de Tipo, un octeto de longitud y n octetos de valores):

Tipo	Longitud	Descripción
0	n	Relleno
1	4	Máscara de subred
2	4	Tiempo GMT
3	n	Direcciones IP de routers
4	n	Direcciones IP de Servidores de tiempo
6	n	Direcciones IP de DNS Server
7	n	Direcciones IP de de Log Server
10	n	Direcciones IP de Server de impresión
12	n	Nombres de cliente
13	2	Tamaño del archivo de booteo
128-254	n	Reservados para uso específico de esa Site
255	1	Fin de lista

Algunos de estos Tipos pueden obtenerse por medio de otros protocolos (Ej: ICMP, WINS,etc), pero los estándares recomiendan el empleo de estos campos en BOOTP para evitar tráfico de red.

Si se comparan con el formato DHCP se puede apreciar la similitud entre estos, pero en el detalle difieren.

Un detalle significativo de este protocolo es que no provee la información de booteo, le brinda toda la información de red necesaria, e inclusive en el campo *Nombre del archivo de booteo* puede especificar la dirección y el nombre de un servidor TFTP (por ejemplo), y con este dato completar el segundo paso que sería la obtención de los archivos de booteo. Este detalle aunque pase inadvertido es de suma importancia pues se puede desde optimizar el tráfico hasta poseer distintos sistemas operativos de booteo (Ej: UNIX, Windows 2000, etc) en distintos servidores, y acorde a la solicitud BOOTP cliente, asignarle cada servidor.

Si bien el protocolo BOOTP fue un avance significativo, representa una configuración bastante estática, el administrador de red, crea un serie de parámetros para cada Host en el cliente y el servidor, los cuales permanecerán en los mismos por períodos de tiempo relativamente largos.

A fines de los 90'se hacen reales dos hechos que envejecen prematuramente este protocolo:

- Integración masiva de las distintas redes privadas a Internet.
- Empleo cotidiano de computadoras portátiles.

La explosión de Internet hace que las asignaciones de direcciones IP sean escasas e imponen a muchas subredes rangos más pequeños que la cantidad de host que se poseen, obligando a reducir el tiempo de vida de las asignaciones IP dentro de una subred, para que de esta forma sean compartidas por más de un usuario. El empleo de Notebooks hace que las direcciones asignadas sean diferentes acorde a la subred en la cual se haga presente f'ísicamente. BOOTP no se adapta a estas situaciones pues realiza un mapeo estático. Bajo esta situación es que nace DHCP, tratado al principio.

5.4.4. Seguridad (¿Asignación dinámica o estática?):

Si bien el funcionamiento de la asignación dinámica, es un gran apoyo para los administradores de redes, y en realidad facilita notablemente esta tarea, eliminando a su vez las posibilidades de asignaciones repetidas de direcciones IP, lo cual era un problema bastante frecuente en redes grandes; nuevamente aparece esa peligrosa relación de facilidad Vs seguridad. Si se analiza en detalle la solicitud DHCP, se trata de un Broadcast, el cual será recibido por todos los servidores DHCP de la red (inclusive se puede programar cómo se desea operar con el mismo a través de los routers). Todos los servidores contestarán ofreciendo una dirección IP, y la que primero llegue al host solicitante, será la que este negocie en las dos tramas restantes.

Si se tiene en cuenta que en una red Ethernet, **no existe forma de asignar prioridades de acceso al canal**, nunca se podrá definir sectores de la red que sean clientes de ciertos servidores DHCP, como sí se puede hacer con servidores WINS o DNS (pues estos ya son

parte de la configuración IP, estática o dinámica), pues al host aún no posee ninguna configuración IP. Este detalle desde el punto de vista de la seguridad, presenta dos grandes problemas:

1) Si se posee más de un servidor DHCP, los rangos de cada uno de ellos, serán adjudicados **aleatoriamente en los distintos host de la red**. Este aspecto no permite planificar grupos de direcciones IP que rápidamente identifiquen la ubicación física de un host en la red.

Se desea hacer especial hincapié en este párrafo, pues en redes seguras (con IP estáticas bien asignadas), uno de las mayores ayudas que se tiene al realizar análisis de tráfico es que al capturar direcciones IP, inmediatamente se puede inferir de qué host se trata, si es parte de la red, dónde se encuentra el mismo, que funciones desempeña y por lo tanto qué derechos y obligaciones posee. Deténgase a pensar si esto es posible con asignaciones estáticas.

Dentro de esta línea de pensamiento es que se debe tener muy en cuenta este punto al comenzar a diseñar una red (pues luego se hace muy dificultoso). Es de vital importancia desde el vamos dedicarle todo el tiempo necesario para planificar la estrategia de direccionamiento de la red completa (y si no se hizo desde el principio es una tarea que SI o SI se debe hacer). Un muy buen punto de partida es armar planos de cada subred, dimensionar las subredes, asignarle rangos de IP privadas acorde a la magnitud de cada una de ellas, luego seguir avanzando de lo global a lo particular, es decir, dentro de cada subred continuar por regiones, edificios, pisos, grupos de trabajo, etc. hasta llegar a identificar al último host.

Ej: 10.65.130.67

Podría ser interpretado como:

- **10.65** (1ro y 2do octeto): SubRed perteneciente a la **zona A** (Desde 10.65 hasta 10.127, podría ser zona B desde 10.128 hasta 10.191)
- **130** (3er octeto): **Edificio A2 - Piso 2** (Edificio A1 = Subred 1 hasta 127, Edificio A2 = Subred 129 hasta 254, dentro del mismo: piso 1 = 129, Piso 2 = 130, Piso 3= 131,.....).
- **67** (4to octeto): **Grupo 1**: Podría asignarse desde 65 hasta 127 = grupo 1 y desde 129 a 191 grupo 2.

Este ejemplo está basado en REDES REALES de alta seguridad, donde se necesita tener inmediata visualización de toda dirección IP que se monitorice, y por más que parezca difícil la lógica de asignación, cualquier operador que trabaje con una consola en un muy corto período de aprendizaje, al ver pasar cualquiera de estas direcciones **inmediatamente ubica de dónde se encuentra** pues si se sigue una buena lógica, es extremadamente fácil familiarizarse con la red**.**

Esta planificación no implica la obligación de realizar todo con direcciones estáticas, pero si la de hacerlo en los segmentos en los cuales las aplicaciones son críticas, excluyendo estos rangos de direcciones de los servidores DHCP que se hayan instalado.

2) Cualquier host que se conecte físicamente a la red obtendrá una dirección IP y a partir de aquí navegará con total libertad por la red LAN.

5.5. IP Versión 6 (IP Next generation).

5.5.1. Conceptos:

Desde hace tiempo ya se hacen evidentes algunas falencias que hoy tiene la actual versión del Protocolo IP (Versión 4). Algunas de ellas son la mala distribución que utiliza de sus cantidades de Host en cada una de sus redes (A, B y C); son muy pocas o ninguna las empresas que poseen los millones de Host que permite una dirección tipo A, hoy tampoco existen 2.100.000 empresas, por lo tanto, tanto en A como en C se desperdicia la asignación de direcciones, si bien hoy se asignan porciones de las mismas, este no fue el sentido con que fueron creadas. Otra debilidad es la no posibilidad asignaciones geográficas , lo que representa una enorme carga de tablas en los router exteriores, casi ya imposible de controlar. También se suman detalles de seguridad, criptogrfiado, Prioridades, longitud variable de cabecera, etc.

5.5.2. Características:

Las características principales de IPv6 son:

- Direccionamiento: 128 bit.
- Encaminamiento: Direccionamiento jerárquico.
- Prestaciones: Cabecera simple de 40 Byte, alineada de a 64 bit, y cualquier otra información se agrega como cabecera en extensión (opcional).
- Versatilidad: Formato flexible de opciones.
- Multimedia: Id de flujos.
- Multicast: Obligatorio.
- Seguridad: Soporte de autenticación y cifrado.
- Autoconfiguración: Tres métodos PnP.
- Movilidad: Surce routing, seguridad, detección de móviles, hand-off.
- Fragmentación: Únicamente de extremo a extremo, es decir que sólo el origen puede fragmentar. Para implementar esto, hace uso de PMTU (Path MTU, RFC 1191), que es el mecanismo empleado para determinar la máxima unidad de datos que contendrá

un datagrama, una vez conocido este tamaño, armara todos los paquetes sin superar el mismo, por ende ningún router deberá fragmentarlo pues no será necesario.

- Tamaño de datagrama: Mantiene el mismo concepto que la versión 4 y propone un nuevo modelo de datagrama, llamado Jumbograma, el cual se define a través de una cabecera en extensión, y permite transmitir datagramas de hasta 4 Gbyte. La idea de esta nueva aplicación es permitir la transmisión de grandes volúmenes de datos entre servidores, los cuales no necesitan incrementar con tanta redundancia de cabecera, siendo el mejor representante de esto el empleo de cluster de servidores.

5.5.3. Encabezado de IPv6:

<table>
<tr><td>Versión</td><td>Clase de tráf.</td><td colspan="2">Rótulo de flujo</td></tr>
<tr><td colspan="2">Longitud de carga útil</td><td>Sig. Cabecera</td><td>Límite Saltos</td></tr>
<tr><td colspan="4">Dirección fuente</td></tr>
<tr><td colspan="4">Dirección destino</td></tr>
<tr><td colspan="4">Posibles cabeceras de extensión</td></tr>
</table>

- Versión: (4), se mantiene el mismo tamaño para permitir distinguirlo del versión 4 y que puedan convivir durante algún lapso de tiempo.
- Clase de tráfico (4): (Video, audio, datos, voz, etc).Cuanto más alto sea su valor más importante es, los valores que puede adoptar son:

  0 tráfico sin caracterizar

  1 tráfico “filler”

  2 transferencia de datos no atendida, como E-mail

  3 reservado

  4 transferencia de bloques de datos atendida, como FTP

  5 reservado

  6 tráfico interactivo, como TELNET

  7 tráfico de control de Internet, como protocolos de encaminamiento

- Rótulo de flujo: (24), Todos los datagramas del mismo flujo (Ej: todos los datagramas de una misma FTP).
- Longitud de carga útil: (16): Cantidad de octetos de datos.

- Siguiente cabecera: (8), se permiten varias, todas ellas van después del campo Dirección destino y aquí se identifican cuales van.
- Límite de saltos: (8), para evitar lazos infinitos.
- Dirección origen y destino (128 c/u), aparece aquí aparte de Net y Host un nuevo identificador llamado Dirección de Agrupación, que identifica regiones topológicas.
- Posibles cabeceras de extensión (Extension Headers):

  Irán colocadas antes del campo de datos, cada cabecera tendrá un primer campo (8 bit) que indica la próxima cabecera (Next Header) que indica si existe otra cabecera de extensión o si esta es la última:

  - Cabecera salto por salto (valor 0): Lleva información para analizar en cada router.
  - Cabecera extremo a extremo: Lleva información que solo se examinará en el destino.
  - Cabecera de enrutamiento (valor 43): Ruta fija.
  - Cabecera de fragmento (valor 44): Si existe fragmentación.
  - Cabecera de verificación de autenticidad(valor 51): Permite verificar autenticidad de origen.
  - Cabecera de confidencialidad: Los datos no deben ser leídos durante su paso por Internet.

5.5.4. Direccionamiento de IPv6:

- Direcciones de 128 bit (16 octetos) (más de 10^{38} direcciones posibles).
- A pesar de las restricciones de redes y reservadas aún quedan más de 1.500 direcciones por m^2 de la superficie de la tierra.
- Tres tipos de direcciones (unicast, anycast y multicast).
- No existen clases, similar al concepto de CIDR.

Notación general 3FFE:2213:AE56:54AD:34EF:9888:33EA:AA21

Los ceros contiguos se pueden eliminar, es decir los siguientes pares de octetos se podrían representar como están indicados a su derecha:

:002E: → :2E:

:000A: → :A:

:6700: → :6700:

:0004:0000:0000:0000:000A: → :4::A: (Sólo una vez se pueden resumir las secuencias seguidas de ceros, con dos puntos seguidos ::).

0004:0000:0000:0000:000A:0000:0000:1243:00AD: 4::A:0000:0000:1243:AD:

Las direcciones compatibles con Ipv4 se abrevian con un solo punto (en vez de doble), o cual indica que se trata de un solo octeto y no dos:

0:0:0:0:0:FFFF:201.200.32.129 → ::FFFF:201.200.32.129

5.5.5. Tipos de direcciones:

Se definen tres tipos de direcciones IPv6:

- Compatibles con IPv4

  Una dirección indicando un nodo IPv6 con una dirección que se puede mapear unívocamente al espacio IPv4. Tienen el prefijo IP 0:0:0:0:0:ffff. Por ejemplo, 0:0:0:0:0:FFFF:119.234.21.44

- Mapeadas a IPv4

  Una dirección IPv6 que indica un nodo sólo IPv4. Tienen el prefijo IP 0:0:0:0:0:0. Por ejemplo, 0:0:0:0:0:0:36.56.24.241.. Es importante darse cuenta de que las direcciones compatibles con IPv4 y las mapeadas a IPv4 utilizan el mismo espacio de direcciones. El prefijo sólo indica si el nodo soporta o no IPv6.

- Sólo IPv6

  Una dirección IPv6 que indica un nodo que soporta IPv6 donde los 32 bits inferiores no contienen necesariamente una dirección IPv4. Los 96 bits de orden superior son distintos de 0:0:0:0:0:FFFF o 0:0:0:0:0:0.

Direcciones especiales:

0:0:0:0:0:0:0:1 → Loopback.

0:0:0:0:0:0:0:0 → Dirección no especificada.

Concepto de prefijo: El prefijo es similar a la notación de cisco de máscara de red, es decir se anexa a continuación de la dirección IP, separado por una barra (/) en notación decimal el valor que identifica la cantidad de bit que están puestos a uno en la máscara de red (de izquierda a derecha):

0004:0000:0000:0000:000A:0000:0000:1243:00AD/48

RFC que hacen referencia a IPv6

1881 IPv6 Address Allocation Management.

1883 Internet Protocol, Version 6 (IPv6) Specification.

1884 IP Version 6 Addressing Architecture.
1887 An Architecture for IPv6 Unicast Address Allocation.
1897 IPv6 Testing Address Allocation.
1924 A Compact Representation of IPv6 Addresses.
1933 Transition Mechanisms for IPv6 Hosts and Routers.
1970 Neighbor Discovery for IP Version 6 (IPv6).
1971 IPv6 Stateless Address Autoconfiguration.
1972 A Method for the Transmission of IPv6 Packets over Ethernet Networks.
2073 An IPv6 Provider-Based Unicast Address Format.
2147 TCP and UDP over IPv6 Jumbograms.
2374 An IPv6 Aggregatable Global Unicast Address Format.
2375 IPv6 Multicast Address Assignments.
2460 Internet Protocol, Version 6 (IPv6) Specification.
2461 Neighbor Discovery for IP Version 6
2462 IPv6 Stateless Address Autoconfiguration.
2471 IPv6 Testing Address Allocation.
2473 Generic Packet Tunneling in IPv6 Specification.
2474 Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers.
2491 IPv6 over Non-Broadcast Multiple Access (NBMA) networks
2526 Reserved IPv6 Subnet Anycast Addresses.
2675 IPv6 Jumbograms.
2732 Format for Literal IPv6 Addresses in URL's.
2893 Transition Mechanisms for IPv6 Hosts and Routers.
2928 Initial IPv6 Sub-. TLA ID Assignments.

EJERCICIOS DEL CAPÍTULO 5 (Nivel de Red)

Los ejercicios de esta capítulo hemos decidido iniciarlos con una serie de prácticas de direccionamiento, pues insistimos, desde el punto de vista de la seguridad, es muy importante diseñar esquemas de direcciones IP que nos permitan hacer crecer nuestras redes de forma, lógica, intuitiva y flexible. Recordad siempre que es una muy buena práctica, el hecho de capturar un dirección IP e inmediatamente poder identificar a qué ubicación física se corresponde, a qué área de mi empresa, edificio, zona, si es troncal, LAN, WAN, etc. Si logramos hacerlo bien, esta tarea nos allanará mucho tiempo y esfuerzo a la hora de identificar, seguir o recuperarnos de cualquier tipo de incidencias

1. Identificación de Red y Host.

   Rodea con un círculo la parte del host de cada dirección IP de las que figuran a continuación, respetando las "Clases" y aclara que "Tipo" es:

   10.15.123.50

   119.18.45.0

   171.2.199.31

   209.240.80.78

   198.125.87.177

   199.155.77.56

   223.250.200.222

   192.15.155.2

   158.98.80.0

   123.102.45.254

   217.21.56.0

   148.17.9.155

   10.250.1.1

   100.25.1.1

   150.10.15.0

   195.0.21.98

   192.14.2.0

   25.250.135.46

   148.17.9.1

   171.102.77.77

   193.42.1.1

2. Máscaras de Red y Subred.

En los problemas que figuran a continuación, se plantea una dirección de red (respetando las clases), en cada uno de ellos deberás aclarar de qué clase se trata, cual es su máscara de red, y luego calcular qué máscara de subred debes (o puedes) emplear para dividirla en las subredes que se plantean como "necesarias".

Problema 1

Nº de subredes necesarias 2

Dirección de Red 192.168.40.0 ¿Tipo?..... ¿Máscara de Red?.....

¿Máscara de Subred?.....

¿Hasta cuántos host tendrá cada subred?.....

Problema 2

Nº de subredes necesarias 9

Dirección de Red 192.168.121.0 ¿Tipo?..... ¿Máscara de Red?.....

¿Máscara de Subred?.....

¿Hasta cuántos host tendrá cada subred?.....

Problema 3

Nº de subredes necesarias 20

Dirección de Red 192.168.1.0 ¿Tipo?..... ¿Máscara de Red?.....

¿Máscara de Subred?.....

¿Hasta cuántos host tendrá cada subred?.....

Problema 4

Nº de subredes necesarias 78

Dirección de Red 172.18.0.0 ¿Tipo?..... ¿Máscara de Red?.....

¿Máscara de Subred?.....

¿Hasta cuántos host tendrá cada subred?.....

Problema 5

Nº de subredes necesarias 146

Dirección de Red 10.0.0.0 ¿Tipo?..... ¿Máscara de Red?.....

¿Máscara de Subred?:.....

¿Hasta cuántos host tendrá cada subred?:.....

3. Direcciones IP válidas e inválidas para un host.

Identifica cuáles de las siguientes direcciones son correctas y utilizables.

(Si no se pueden usar explica la razón....)

192.10.10.1/24

245.150.190.10

172.17.33.45/14

192.168.256.4/24

135.70.191.255/24

135.70.191.255/16

127.0.0.0/8

93.0.128.1

0.230.190.192

238.3.44.56

200.10.10.128/25

200.10.10.175/22

10.127.255.255/8

10.127.255.255/9

225.0.0.32

4. Trabajo con tablas de ruta.

Investiga las diferentes opciones del comando "route" tanto en Windows como en Linux. Luego prueba diferentes opciones para añadir, modificar y borrar las mismas.

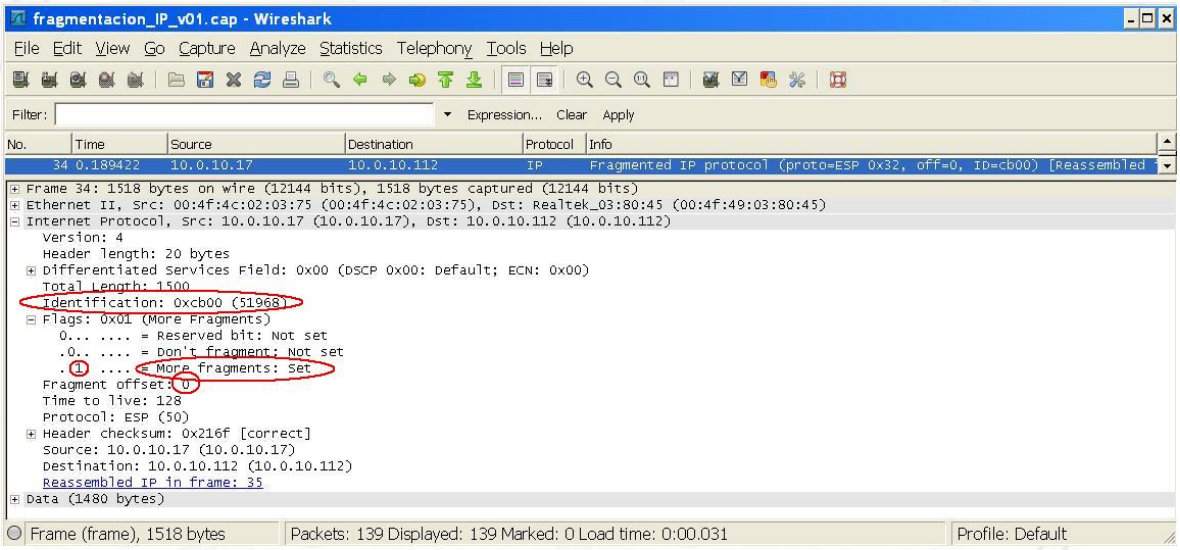
Si tuvieras acceso a algún dispositivo que cuente con más de una tarjeta, verifica cómo es su tabla de rutas.

Si tienes un ordenador portátil o de sobremesa con tarjeta Ethernet y WiFi, intenta realizar las diferentes posibilidades de conexión a tu router ADSL. ¿Qué sucede si estando conectado por WiFi, conectas a su vez el cable de tu tarjeta Ethernet?, ¿Sigue funcionando tu salida a Internet?, ¿Qué direcciones IP te indica tu tabla de rutas?

Lanza el analizador de protocolos y si continúas navegando por Internet investiga ¿con cuál IP, e Interfaz estás navegando?, ¿Qué sucede con la otra?.... ¿Te atreves a modificar tu configuración en las direcciones IP y en las tablas de ruta?

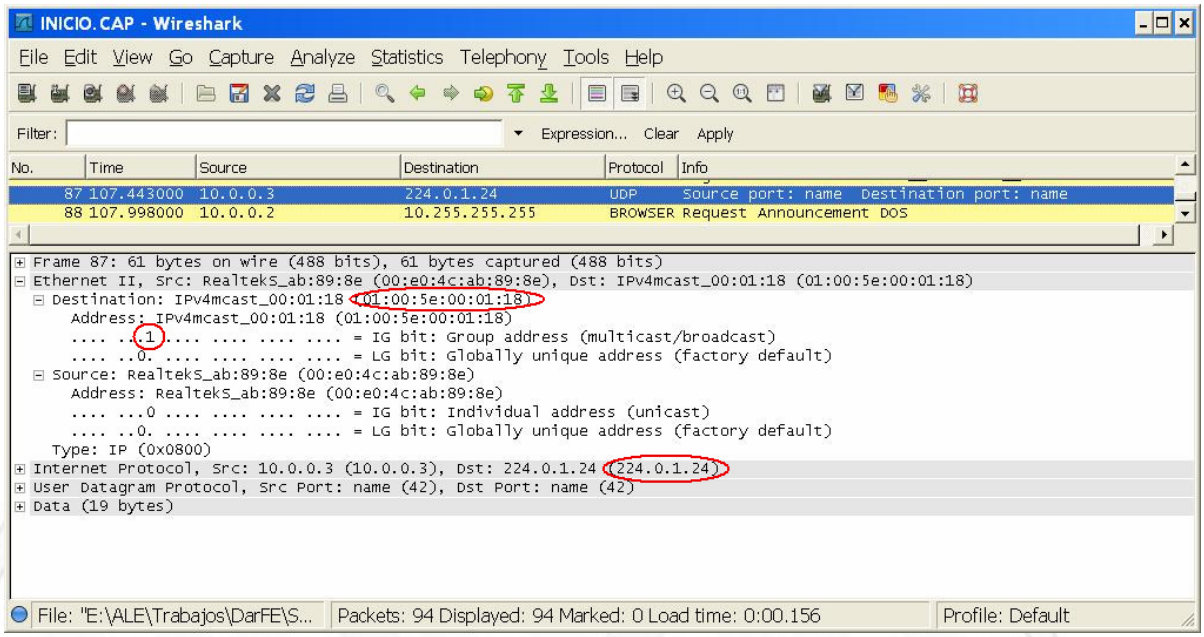
5. Ejercicios con capturas de tráfico con protocolo IP.

Fragmentación IP: Nuevamente empleando el comando "ping", generar tráfico con la opción de tamaño mayor a 1500 Bytes para que el protocolo IP se vea obligado a fragmentar, capturarlo con el analizador de protocolos y verificar el empleo de los campos de fragmentación, tal cual se presenta en la imagen siguiente:



Multicast IP:

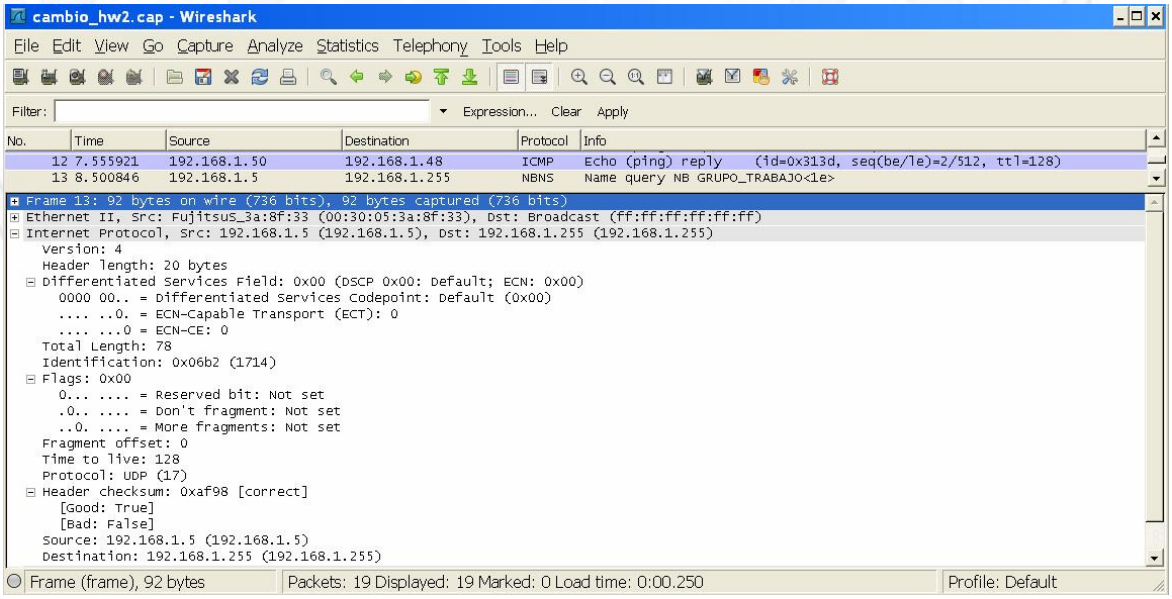
Ahora que hemos desarrollado el esquema completo de multicast a nivel enlace y red, te invitamos a que captures tráfico multicast IP y analices cómo se "solapan" los tres últimos octetos de la dirección IP, respecto a la dirección MAC. A continuación presentamos el ejemplo de captura remarcando este solapamiento:



¿Te atreves a describir cómo se relaciona este solapamiento, es decir por qué el último octeto IP es "24" y el MAC "18"? (¿Te acuerdas aún el pasaje de decimal a hexadecimal?)

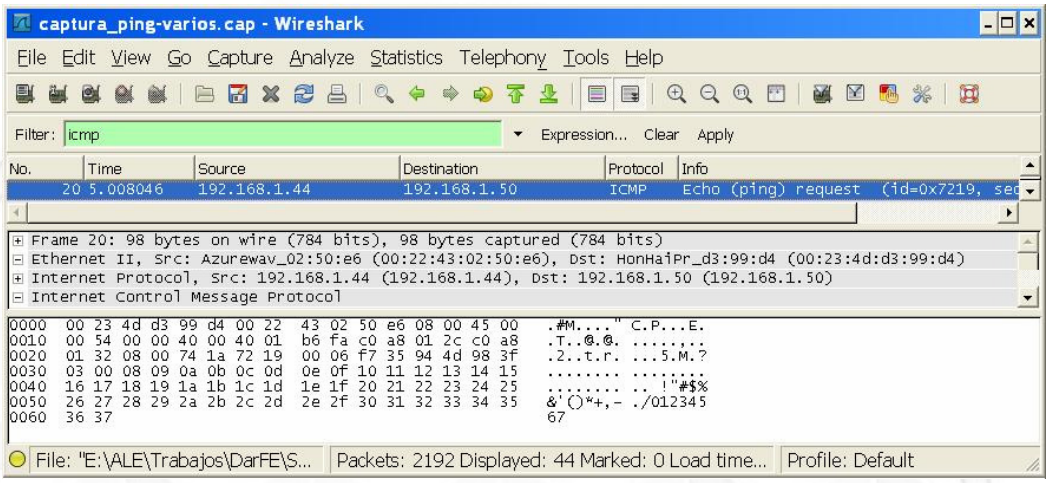
Encabezado IP:

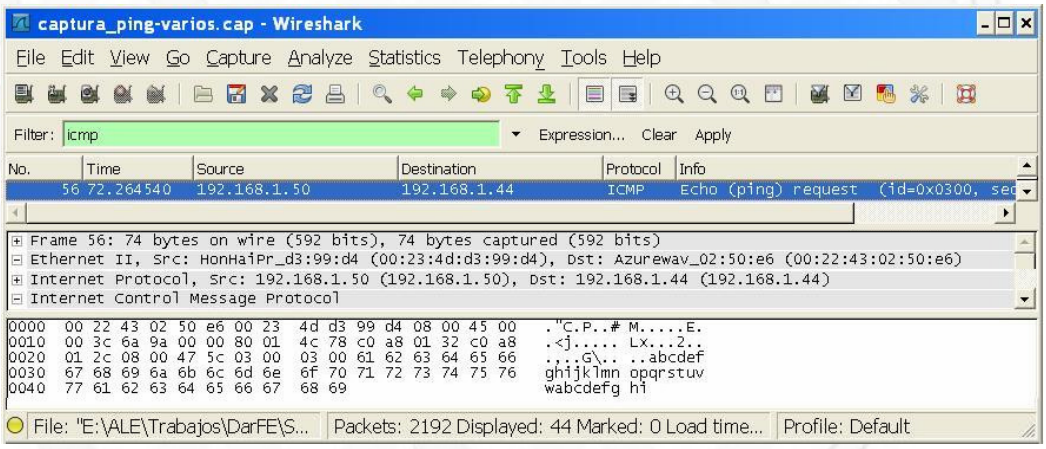
En este ejercicio, sencillamente captura un datagrama IP, despliega todos sus campos verifica y repasa cada uno de ellos en su encabezado. A continuación, te presentamos un ejemplo de Broadcast IP para que lo compares con tu captura:



5. Ejercicios de captura de tráfico con protocolo ICMP.

A continuación te presentamos dos capturas del comando "ping", una está ejecutada desde una máquina Windows y la otra desde Linux. Míralas con atención, ¿Qué diferencia puedes encontrar entre ambas?, ¿Te atreves a generar estos ping desde tu máquina, capturarlos y sobre esa base identificar cuál captura es la de Linux y cuál la de Windows? Investiga un poco este tema en Internet y corrobora lo que tú mismo has deducido.





REFLEXIÓN: Estamos comenzando a ver cómo es posible identificar diferentes SSOO sencillamente por pequeñas diferencias entre como cada uno de ellos implementa un protocolo, en este caso los tipos 0 y 8 de ICMP, pero seguiremos viendo este hecho con mucha frecuencia a lo largo del libro y es justamente una de las mayores fuentes de información para un intruso.

En la parte de herramientas de este Capítulo (a continuación), verás algunas que te permitirán generar todos los tipos y códigos ICMP que desees (hasta inventarte los tuyos), el trabajo importante que debes hacer con ellas, es justamente "Capturarlas", para poder analizarlas en detalle y comprender perfectamente y en la práctica su funcionamiento.

6. Investigación sobre ICMP: Te proponemos que a través de cualquier buscador, busques información, debilidades, funcionalidades, seguridad, etc. Sobre ICMP, pues sería muy importante que le dediques un tiempo y lo conozcas con el máximo detalle.

EJERCICIOS CON HERRAMIENTAS

1. Herramienta "**IPCalc**"

Emplearemos dos tipos de calculadora IP, la primera a través de línea de comandos y la segunda con entorno gráfico

No requiere mayores explicaciones, sencillamente te invitamos a que veas "man ipcalc" desde Linux.

La interfaz gráfica si empleas "**Gnome"**, se instala como "**gip**".

También puedes descargar "**ipcalc**" para Windows, es muy similar a la anterior.

2. Comando "**ping**"

- Probar diferentes opciones del comando "Ping" y describirlas.
- ¿Qué sucede con ping -b?, ¿Qué se dice de esto en Internet?
- Si puedes generarlo en entornos donde existen diferentes SSOO o versiones de ellos, un ejercicio excelente es verificar quiénes responden y cuáles no. Igualmente te invitamos a que lo investigues en Internet pues encontrarás mucho acerca de esta opción.
- Buscar en Internet direcciones IP que puedan tener alguno de estos SSOO y verificar si se puede obtener alguna respuesta.

3. Empleo de "**hping3**" (Lo emplearemos desde Linux)

- ¿Para qué sirve la opción "-E"?
- ¿Qué diferencia me ofrece el ejercicio anterior, si empleo la opción "-c" o si no la empleo? ¿En qué nivel de la pila de protocolos viaja la información?
- Ejecutar: "hping3 -0 -d 1 -E Nombre_Archivo Dirección_IP", capturar y describir qué está sucediendo.
- Ejecutar: "hping3 -x -y Direccion_IP". Verificar cuáles SSOO responden y cuáles no.
- Ejecutar: "hping3 -2 -c 2 -p 50 Direccion_IP". (Hacia un puerto cerrado de un host activo y hacia un host no activo). ¿Para qué nos sirve?

4. Comando "**Fragroute**" (lo emplearemos desde Linux).

- Crear un archivo básico para ejecutar Fragroute.
- Ejecutar "fragroute - f Archivo Direccion_IP.

- ¿Presenta algún mensaje de error?, ¿Por qué?
- ¿Existe alguna forma de evitar el mensaje de error anterior, sin tener que ejecutar ningún otro comando previo?
- Analizar el archivo "/etc/fragroute.conf", ¿Qué hace?
- Lanzar fragroute hacia una dirección IP, ¿Qué está haciendo?
- Intenta realizar combinaciones entre fragroute y hping3, por ejemplo "hping3 -c 1 -d 30 -E archivo_a_enviar Direccion_IP_Destino" ¿Qué sucede?
- ¿Se puede realizar algo similar a lo anterior desde dos máquinas diferentes?
- Prueba algunas opciones de **Fragtest**
- enviarse a uno mismo un correo con contenido extenso en texto plano (Capturar con Wireshark), (verificar la IP del receptor de ese correo) (Analizar el contenido de esa captura)
- Lanzar Fragroute hacia esa dirección.
- Reenviar el correo (y capturarlo) ¿Qué sucede?
- ¿Qué sucede si intercambio datos con otro host (otra_IP)?

5. Empleo de "**ICMPush**" (Lo emplearemos desde Linux).
   - ¿Qué acciones novedosas u originales se pueden desarrollar con tipo 0 y 8?
   - ¿Se puede hacer algo con ping en multicast y Broadcast?..... ¿Cómo se llamaría esta actividad y/o ataque?
   - ¿Qué tipos de ataque has investigado con tipo 0 y 8?
   - ¿Cómo podemos emplear ICMP para la detección de routers?
   - ¿Qué acciones se pueden realizar con los marcadores de tiempo?
   - ¿Las opciones de máscara (Mask) e información (info) qué nos permiten hacer?
   - ¿Qué ofrece la combinación de "-echo" con "-pat"?
   - ¿Qué estaríamos generando con: icmpush -du -c max -gbg 65000 Direccion_IP? ¿Hasta cuánto soporta la opción "-gbg"

6. Ejercicios con "**nmap**" (Ejecuta los comandos, describir y presentar conclusiones) (lo emplearemos en Linux, pero ya existe también para Windows).

   "**nmap**" es una herramienta fundamental en redes, como iremos viendo, se puede emplear para muchas actividades y diferentes niveles del modelo de capas, por ahora intentaremos acotarla a su uso para actividades correspondientes al nivel de red, pero más adelante continuaremos con ella, por ahora:

- Analiza las opciones: "PE/PP/PM", ¿Qué sucede en entornos LAN, y qué sucede en entornos WAN?
- ¿Qué hace la opción -PI?
- "-sP": (Sondeo Ping) ¿Cómo trabaja?, ¿Por qué lo llama Ping el man?
- "-sO": ("O", no "Cero") (IP protocol Scan).
- "-PO": (protocol list) (IP Protocol Ping).
- -PN (Desactiva PING)
- -PU (UDP Ping)
- La idea de los ejercicios anteriores es comprender que existen diferentes formas de "**ping**". A nivel ICMP, IP, TCP y UDP. En base a este planteo, Explica cómo (o con qué opciones) se puede ejecutar cada uno de ellos.
- Verificar qué sucede si hago "traceroute 80.58.61.250" (*DNS de Telefónica*) y "traceroute -U -p 53 80.58.61.250"
- ¿Qué sucede con la siguiente opción: nmap -sP—send-ip Dirección_IP?
- Supongamos que con alguna opción escaneamos un rango de red, ¿Qué implica que no reciba respuesta de ninguno, pero sí recibo una respuesta ICMP de uno de ellos con el tipo y código "Host inalcanzable"?, ¿Se puede probar y verificar este caso?..... (Se llama mapeo inverso [Inverse Mapping])

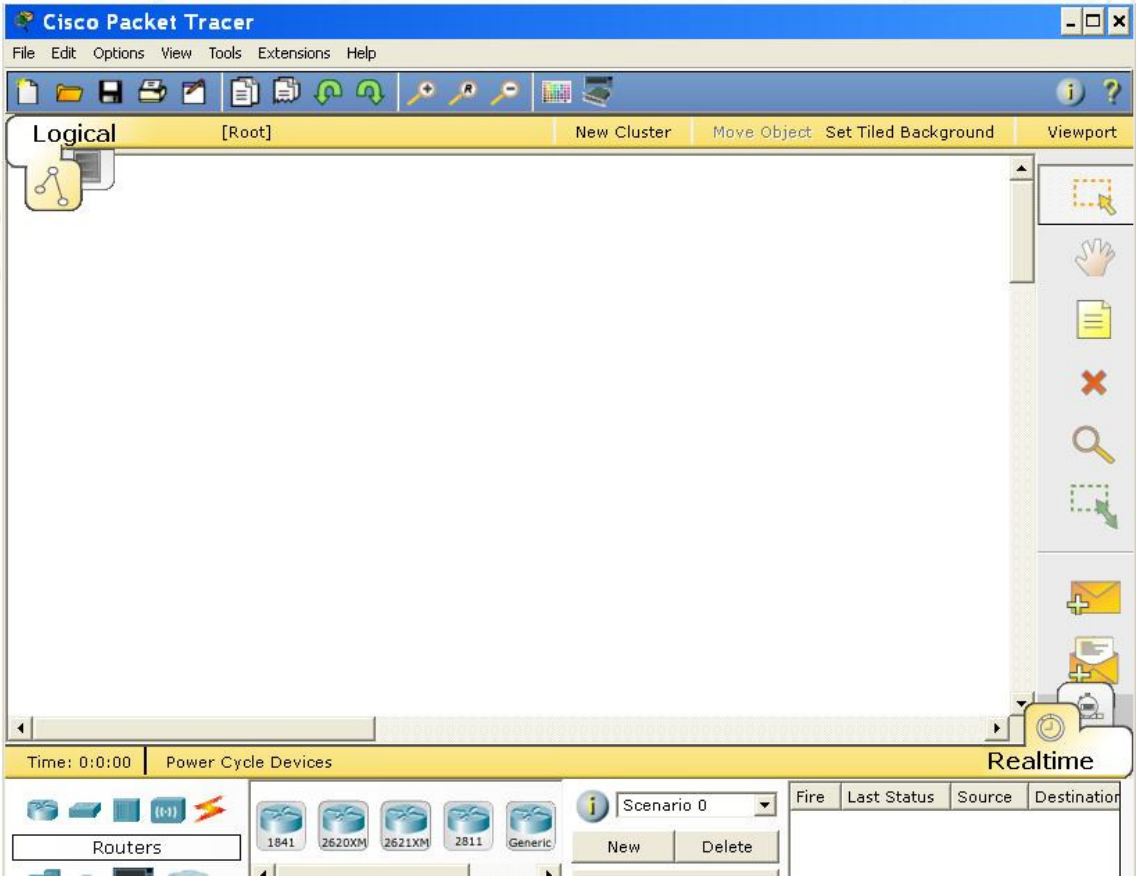
7. Ejercicios con **nmap** que amplían lo realizado con otras herramientas en el protocolo IP.
   - -f –mtu <valor> fragmenta paquetes con el MTU indicado.
   - --spoof-mac <dirección mac> Falsifica la dirección MAC.
   - --traceroute: Trace hop path to each host (Probar qué sucede con traceroute si ponemos la opción -sP y si no la ponemos).
   - --ip-options <options>: Send packets with specified ip options
   - --ttl <val>: Set IP time-to-live field

8. Ejercicios de red con "**Packet Tracer**"

Packet Tracer es una herramienta de las Academias CISCO para simulación de ejercicios empleando las características de sus dispositivos, para el común de los mortales que no cuentan con toda la gama de CISCO a su disposición, es una herramienta muy útil. Si nos centramos en el nivel de red, nos permite simular toda una arquitectura LAN y WAN, que sería casi imposible de tener a nuestra disposición.

En el caso de este capítulo quisimos incluir algunos ejercicios, para que practiques o veas cómo se puede diseñar una arquitectura de red, teniendo en cuenta todos los conceptos desarrollados para este nivel.

El empleo de la herramienta es sumamente intuitivo y puedes encontrar en Internet muchísima información al respecto.



A continuación se presentan los pasos a seguir en este ejercicio.

- Configurar una red WAN, compuesta de:
  - Vínculos troncales internos 10.0.0.0
  - Vínculo de salida a Internet 172.16.0.0
  - Redes LAN 192.168.0.0
- Configurar un router central al que estén conectados todos los demás.
- Configurar cada LAN con un Switch y al menos un host.
- Configurar todas las interfaces de red.
- Configurar las rutas estáticas para cada uno de ellos.
- Configurar las rutas y gateway por defecto.
- Verificar (con ping) el funcionamiento de toda la red.
- Implementar ACLs estándar, para evitar que desde las redes LAN se pueda acceder a las redes troncales.
- Configurar una zona desmilitarizad (DMZ) con servidor de correo, Web y DNS.
- Configurar un servidor Web para la Intranet
- Configurar un servidor de BBDD para la Intranet y la Web de Internet.
- Configurar todas las rutas hacia ellos.
- verificar que se pueda acceder a cada uno de ellos (por IP y por puerto).
- Definir Access Control Lists (ACL) estándar para filtrar adecuadamente el tráfico desde los usuarios de las LAN hacia las troncales y desde Internet hacia los rangos de red interna.
- Definir Access Control Lists (ACL) extendidas para filtrar adecuadamente el tráfico (por puertos).

Formato de una ACL:

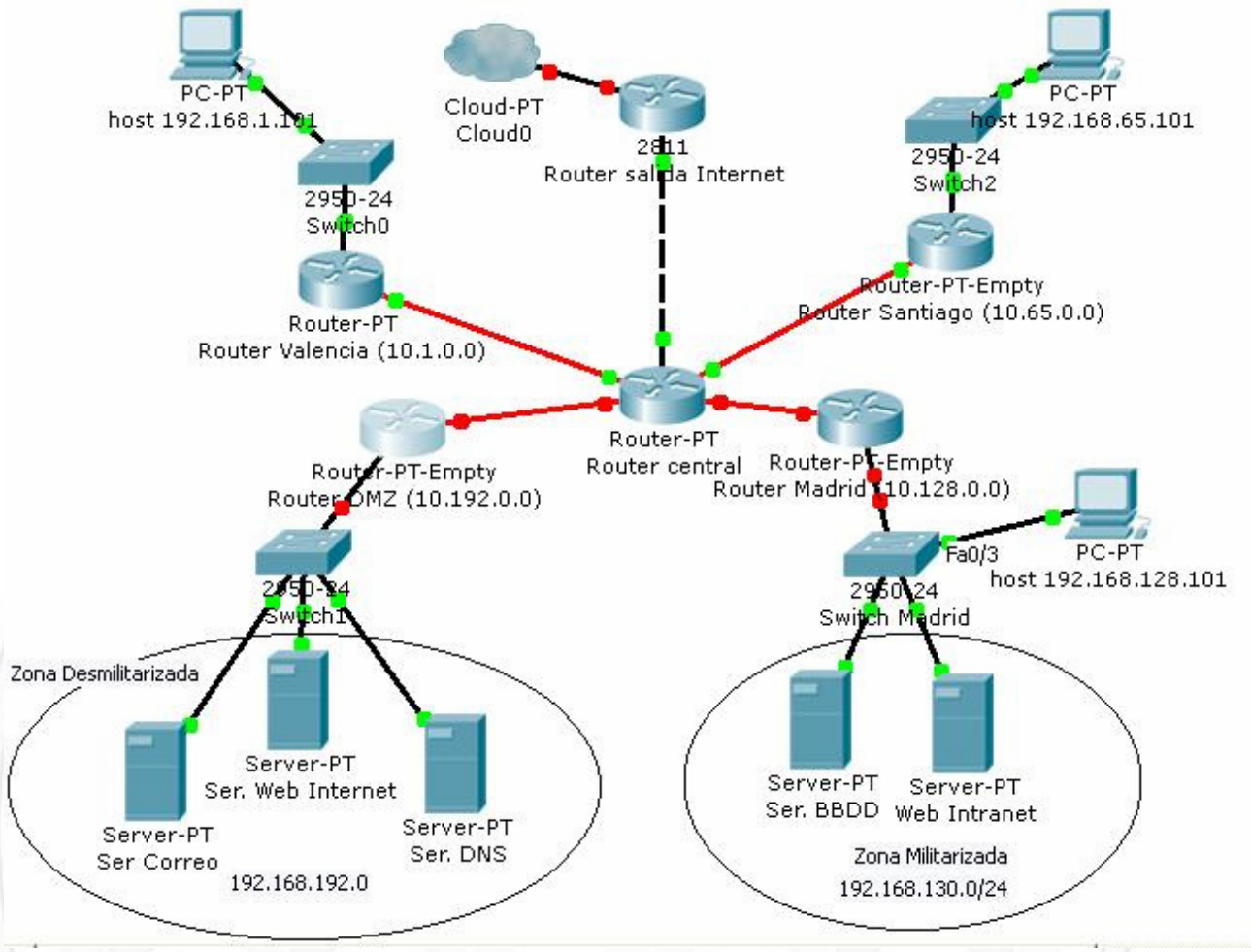
```
access-list[1-99][permit|deny][dirección de origen][máscara comodín]
```

Para negar el acceso de usuarios a las troncales:

(Ej: desde la red 192.168.1.0) en el router de Valencia:

```
router (config)# access-list 1 deny 10.0.0.0 0.255.255.255
router (config)# access-list 1 permit any
router (config-if)# ip access-group 1 out    (en la interfaz LAN)
```

Gráfico de ejemplo de este ejercicio (los rangos de red son sencillamente una opción cualquiera que puede tenerse en cuenta o no).



DESAFÍOS:

1. Ya hemos comentado el comando "**ifconfig**" (Linux) e "**ipconfig**" para Windows. Lanza tu analizador de protocolos, descubre alguna dirección IP que exista en tu red. Te invitamos a que investigues ¿Qué sucede si configuro mi interfaz de red con esa misma dirección IP?

   Te acuerdas aún los ejercicios de "**arpspoof**" y de ataque ARP, ¿Se te ocurre algo más si a su vez modificas tu dirección IP?

2. Profundiza sobre la configuración de un router (tarde o temprano lo deberás emplear).
3. ¿Te atreves a profundizar sobre los ataques del hombre del medio en el nivel de red?

CAPÍTULO 6: El nivel de TRANSPORTE

6.1. TCP (Transport Control Protocol) (RFC 793 , 812, 813, 879, 896 y 1122).

Se trata del protocolo responsable de establecer y gestionar sesiones (conexiones lógicas) entre usuarios locales o remotos. Es también quien se encarga de la fiabilidad, control de flujo, secuenciamiento, aperturas y cierres. Es un protocolo orientado a la conexión, por lo tanto es el responsable de la transmisión de extremo a extremo. Emplea ACK, temporizadores, N(s) y N® con segmentos de longitud variable.

Una característica de su empleo es que el control de flujo lo realiza el receptor, el cual envía un valor de ventana (al emisor). El Transmisor puede enviar un número máximo de ventanas no mayor a ese valor, y al llegar al mismo interrumpe la transmisión hasta recibir los ACK correspondientes, que liberen posiciones en su cola circular de envío (N(s)).

El TCP permite multiplexar varias sesiones en una misma computadora por medio del concepto de socket (explicado en terminología).

6.1.1. Establecimiento y cierre de conexiones:

Al establecerse una sesión TCP, se produce un triple “Handshake” (apretón de manos), el cual se establece por medio de los bit S y A (Característica de un protocolo orientado a la conexión), envío del primer segmento solicitando establecer una sesión y colocando su número de ventana en un cero relativo (Número generado pseudoaleatoriamente que establece el envío de la primer ventana) (bit S = 1), respuesta aceptando y enviando también su número de secuencia (bit S y A = 1) y por último establecimiento de la sesión TCP (bit A = 1); se inicia el cálculo del RTT (Round Trip Time), tiempo que le permite a TCP establecer el control de flujo calculando el promedio que tardan los segmentos enviados con sus correspondientes respuestas.

El cierre de una sesión TCP se produce al enviar el bit F = 1 ante lo cual se responderá con el bit A = 1 quedando finalizada la sesión

La ventana de recepción de TCP en redes Ethernet normalmente se configura a 8769 bytes, que equivale a seis segmentos de 1460 bytes, que sumados a los 20 byte del encabezado TCP, 20 de IP y 18 de Ethernet, hacen 1518 byte que es el tamaño máximo de la trama Ethernet. El tamaño de esta ventana se ajusta automáticamente acorde a una mecanismo de tiempos de recepción y está regulado por la **RFC 1323**.

6.1.2. Control de flujo:

La RFC 1122 define los mecanismos de control de flujo de TCP, basados en los acknowledgments (ACK) de recepción. Este mecanismo permite al receptor de segmentos, regular el tamaño de ventana del emisor, para impedirle el envío de volúmenes de información que no esté en capacidad de procesar.

6.1.3. PMTU (Path Maximun Unit Discovery)

Este mecanismo está descripto por la RFC 1191 y permite determinar el MSS (Maximun Segmenet Size) de una determinada conexión.

El concepto está basado en el empleo del protocolo ICMP, por medio del cual, cuando se establece una conexión a través de una red no local, el bit Don't Fragment (DF) del encabezado IP es configurado a uno, es decir que no permite la fragmentación de ese datagrama. Si a lo largo de su trayectoria, este datagrama se encuentra con una red que no soporta este tamaño (como sería el caso, por ejemplo, de un datagrama generado en una red Token ring, de 4000 Byte, que debe pasar por otra Ethernet), el router que lo recibe, al no poder fragmentar, descartará este datagrama, generando un mensaje ICMP de destino no alcanzable por fragmentación requerida y no permitida a la dirección origen del datagrama descartado, en el encabezado ICMP generalmente incluirá también el tamaño máximo permitido en esa red (dentro de un campo "no empleado" de 16 bit). El que originó el datagrama inicial, al recibir el mensaje ICMP, ajustará la MTU al tamaño indicado.

6.1.4. Retransmisión

TCP con cada segmento saliente inicia un timer de retransmisión (por defecto es 3 segundos al establecer la conexión, y se ajusta dinámicamente, RFC: 793), si ningún ACK es recibido al expirar este tiempo, entonces el segmento es reenviado, hasta llegar al *TcpMaxDataRetransmision*, que por defecto es cinco, luego de lo cual no vuelve a retransmitir ese segmento.

En el siguiente ejemplo se puede ver una secuencia de segmentos TCP, en los cuales luego del primero de ellos, se desconectó físicamente el host destino, por lo tanto no recibía el ACK de respuesta, en la primera columna el incremento fue duplicando su tiempo de envío entre cada segmento, y al llegar al último si no recibiera respuesta, abortaría la transmisión.

delta	source ip	dest ip	prot	flags	description
0.000	10.57.10.32	10.57.9.138	TCP	...A..,	len: 1460, seq: 8043781, ack: 8153124, win: 8760
0.521	10.57.10.32	10.57.9.138	TCP	...A..,	len: 1460, seq: 8043781, ack: 8153124, win: 8760
1.001	10.57.10.32	10.57.9.138	TCP	.. A..,	len: 1460, seq: 8043781, ack: 8153124, win: 8760
2.003	10.57.10.32	10.57.9.138	TCP	...A..,	len: 1460, seq: 8043781, ack: 8153124, win: 8760

4.007 10.57.10.32 10.57.9.138 TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760
..... 10.57.10.32 10.57.9.138 TCP ...A.., len: 1460, seq: 8043781, ack: 8153124, win: 8760

6.1.5. Velocidad de transferencia

TCP fue designado para optimizar su rendimiento sobre condiciones de enlace variables, dependiendo de varios factores:

- Velocidad del vínculo.
- Demora del vínculo.
- Tamaño de ventana.
- Congestión en los routers.

La capacidad de un vínculo está dada por lo que se conoce como el "producto ancho de banda/demora (Δf * RTT [Round Trip Time]). Si el enlace es de buena calidad, el tamaño de la ventana debería ser mayor o igual que la capacidad del mismo (65535 es el máximo), por el contrario, si el enlace posee mucho ruido o congestiones, el empleo de ventanas grandes no es conveniente, pues se aumentará la congestión o se reenviarán muchos paquetes.

6.1.6. Formato del segmento TCP:

<table>
<tr><td colspan="8">Puerto fuente</td></tr>
<tr><td colspan="8">Puerto destino</td></tr>
<tr><td colspan="8">Número de secuencia N(s)</td></tr>
<tr><td colspan="8">Número de aceptación N®</td></tr>
<tr><td colspan="4">Desplazamiento de datos</td><td colspan="4">Reservado</td></tr>
<tr><td colspan="2">Reservado</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td></tr>
<tr><td colspan="8">Ventana</td></tr>
<tr><td colspan="8">Checksum</td></tr>
<tr><td colspan="8">Puntero de urgente</td></tr>
</table>

Opciones y relleno (Variable)
Datos (Variable)

- Puerto fuente y destino: (16), especifican los procesos de nivel superior que utilizan la conexión TCP.
- Número de secuencia y de aceptación: (32), indican la secuencia o posición de los octetos de datos dentro del módulo completo de transmisión. Concatenado a esto, dentro de este campo también va el número de ventana deslizante.
- Desplazamiento de datos: (4), cantidad de palabras de 32 bit que contiene la cabecera.
- Reservado: (6), no se permite su empleo, quedan reservados para uso futuro.
- URG: (1), indica si es un segmento urgente.
- ACK: (1), acknowledgement.
- PSH: (1), Entregar los datos al recibir este segmento.
- RST: (1), reset.
- SYN: (1), sincronismo.
- FIN: (1), último segmento.
- Ventana: (16), cantidad máxima de segmentos que puede enviar el transmisor.
- Checksum: (16), CRC de cabecera y datos.
- Puntero de urgente: (16), si el bit URG está puesto a 1, identifica la posición del primer octeto dónde los datos son urgentes. TCP no dice que hay que hacer con los datos urgentes, sólo los marca.
- Opciones: Son de la forma (Tipo-longitud-valor). Hoy sólo existen definidas 3. O = Fin de lista de opciones. 1 = No operación. 2 = Tamaño máximo de segmento (MSS).
- Relleno: completa a múltiplo de 32 bit de la cabecera.
- Datos: UDP de nivel superior.

Un detalle interesante de TCP es la inclusión de un "Pseudo encabezado", el cual se emplea al realizar el checksum (CRC). La mecánica del mismo es tomar datos del nivel de red (IP), en particular la dirección origen y destino, formar con estos otro encabezado adicional para realizar el cálculo teniendo en cuenta estos datos dentro del checksum de TCP. Estos datos adicionales, no son tenidos en cuenta en la longitud de TCP, ni tampoco se transmiten, pues luego estarán incluidos en el encabezado IP. El objetivo principal de este Pesudo encabezado es proporcionar a TCP la garantía de entrega en el destino correcto de su

información, pues al llegar a destino el segmento TCP, nuevamente el host receptor tomará estos campos del nivel inferior y calculará su CRC, si el destino fuera incorrecto, este segmento sería descartado. Esta implementación rompe un poco el concepto de independencia de niveles, pero también sucede con otros protocolos y no es lo habitual.

Los protocolos más comunes que emplean TCP son los que se detallan a continuación:

Puerto	**Protocolo**
7	Eco
13	Fecha
20 y 21	FTP (File Transfer Protocol)
23	Telnet
25	SMTP (Single Mail Transfer Protocol)
80	HTTP (Hiper Text Transfer Protocol
137,138 y 139	NetBIOS

6.2. UDP (User datagram Protocol) (RFC 768).

Dentro de la pila de protocolos TCP/IP, existen dos opciones de protocolos de transporte. El TCP que se acaba de tratar y el UDP (que lamentablemente se abrevia igual que unidad de datos de protocolo, pero no se trata de esta sino de un protocolo de capa 4) el cual es un Protocolo NO ORIENTADO A LA CONEXIÓN, y se lo emplea con la masa de los protocolos de nivel superior cuando se opera sobre redes LAN. Está regulado por la **RFC 768**, y confía en la baja tasa de errores de una LAN y en los protocolos de nivel de aplicación, siendo por lo tanto un protocolo no confiable pero con la gran ventaja de la reducción de su cabecera a menos de 8 octetos.

6.2.1. Formato de la cabecera de UDP.

Puerto Origen
Puerto destino
Longitud
Checksum (Opcional)
Datos (Variable)

- Puerto origen y destino: (16), SAP de nivel de aplicación. El puerto origen es opcional, si no se emplea se colocan todos sus bit a cero.
- Longitud: (16), total de cabecera y datos.

- Checksum: (16), puede o no estar presente, si lo está es un CRC 16 de cabecera, datos y Pseudo cabecera cono TCP. Este campo también es opcional, y de no usarse también se rellena con ceros.
- Datos: (Variable), UDP (Unidad de datos de protocolo) de nivel superior.

Los protocolos más comunes que emplean UDP son los que se detallan a continuación:

Puerto	**Protocolo**
7	Eco
13	Fecha
53	DNS (Sistema de Nombres de Dominio)
67 y 68	BOOT cliente y Servidor
69	TFTP (Trivial File Transfer Protocol)
123	NTP (Network Time Protocol)
161 y 162	SNMP (Single Noetwork Monitor Protocol)

6.2.2. El peligro de los protocolos no orientados a la conexión:

Los protocolos no orientados a la conexión, como ya se vio, no establecen ni cierran la misma, sino que pasan directamente a la transferencia de datos, confiando que la información llegará al destino deseado.

Al generar este tipo de tráfico, **no se realiza ningún seguimiento del estado** de esa conexión, por eso también se los suele llamar sin estado. No existen números de secuencia, por lo tanto si se desea realizar un análisis de lo que está sucediendo se torna muy dificultoso. Por otro lado, **no se puede definir el "Sentido"** en el que se realiza la conexión, pues esta no existe, y sin ella tampoco se determinará quién desempeña el rol de cliente y quién el de servidor. Estos protocolos en particular son de especial interés en cuanto a las reglas de un firewall, justamente por las dos características que se acaban de mencionar.

6.3. Firewalls

Es probable que alguien se pregunte por qué razón tratamos el tema de Firewall (en adelante **FW**) en el nivel de transporte, pues deberíamos considerar también el nivel de aplicación. Tal vez pueda tener razón, pero apreciamos que al haber llegado hasta este punto del libro, ya se poseen los conocimientos necesarios para abordar este tema y a su vez nos sirve para "segmentar" conocimientos y que a través de esta teoría y ejercicios ya puedas comenzar a asegurar los niveles que "miran" hacia la red con una metodología sólida y robusta. Es probable que cuando luego tratemos el nivel de aplicación puedas ajustar aún más los conocimientos y pasos que has adquirido y realizado sobre los FW, pero por ahora ya tienes toda la base para que empecemos a organizar este tipo de barreras, así que ¡manos a la obra!

6.3.1. ¿Qué es un firewall?

Más allá de los conceptos teóricos y clasificaciones, un FW o cortafuegos, es un dispositivo que, como su palabra lo indica, hace las veces de "barrera", su misión es, de alguna forma bloquear el paso a cierto tipo de información, por lo tanto si seguimos esta lógica debería:

- Poder "escuchar" la totalidad del tráfico que deseemos analizar.

Esto ya lo conocemos, y es la técnica que emplea todo "sniffer".

- Estar en capacidad de "desarmar" los encabezados de cada protocolo.

Esto también lo conocemos y es nuestra conocida librería "libpcap".

- Tener patrones estandarizados para comprender cada protocolo (para verificar su uso correcto).

Esto es lo que vemos en "Wireshark" cuando nos despliega los campos de cada protocolo, o cuando deseamos implantar cualquier tipo de filtro.

- Contar con "elementos de juicio" para poder decidir qué es lo que dejará pasar y qué no.

Por ahora estos serían los elementos básicos que le podríamos pedir, y aquí justamente se establece una primera distinción, pues dependerá si lo que deseamos filtrar tiene su destino hacia un "host" específico o hacia una red, por lo tanto aquí tenemos una primera clasificación:

- FWs de hosts (a veces asociados a FWs personales y/o a servidores).
- FWs de red.

Este diferenciación es importante pues el primero de ellos filtra hacia el nivel de "Aplicación" de ese mismo equipo, sin embargo el segundo de ellos debe decidir si volverá a "enrutar o no" esa trama hacia la red destino.

En este texto centraremos la atención en los FWs de red por ser los que para un administrador de sistemas son tal vez más importantes. Si decimos que deben "enrutar o no", entonces no nos puede caber duda que debe poseer capacidad de decisición sobre "rutas" es decir su tabla de rutas nos tiene que ofrecer diferentes alternativas de enrutado. Este es otro tema que ya hemos tratado, y a nivel modelo de capas, implica la existencia de más de un dispositivo de red, los cuales pueden ser del mismo tipo o no, es decir, que aquí aplica cualquier dispositivo de nivel "físico y enlace" que nos permita tener una dirección IP a nivel "red". Queremos decir con ello, que puede tratarse de host que tenga, por ejemplo:

- Dos (o más) tarjetas Ethernet conectadas a redes o subredes diferentes.
- Una tarjeta ADSL (o más), y una (o más) ethernet conectadas a redes o subredes diferentes.
- Una tarjeta tarjetas WiFi (o más), y una (o más) ethernet conectadas a redes o subredes diferentes.
- Una tarjeta ADSL (o más), una tarjeta WiFi (o más) y una (o más) ethernet conectadas a redes o subredes diferentes.
- ....y así podríamos imaginar todas las combinaciónes que queráis a partir de dos dispositivos de nivel "físico y enlace".

Es importante que no olvidemos esto, pues justamente la decisición final que en definitiva tomará un FW de red, será transmitir o no cada trama por una de esas interfases, y por lo tanto deberá "reconstruirla" al completo en los niveles de red y enlace, pues saldrá hacia una nueva red con una diferente MAC origen.

Ahora entonces, una nueva cualidad que le deberíamos pedir a un FW de red será:

- Capacidad de enrutamiento: Es decir, operar como un "router".

La última característica que nos conviene analizar va relacionada a dos conceptos importantes que estuvimos tratando en este texto. El **primero** de ellos es el de establecimiento de "sesiones TCP", lo cual se trataba de ese "triple handshake" (S – SA –A) con el que queda establecida la sesión. Al desarrollar este tema, hicimos hincapié en la importancia que tiene el "sentido" de esta conexión, el cual queda marcado por el primer segmento con el flag SYN=1 y el ACK=0 (y es esa única combinación, la que lo indica), a partir de este momento se establece un "Indicador de sesión" (relacionado unívocamentea ese "socket"), el cual permite reconocer sin lugar a dudas todos los segmentos "TCP" entre ambos extremos. Esto guarda mucha relación con el **segundo** concepto que es la función de "Segmentación y reensamble" (o "Fragmentación y desfragmentación"), la cual como tratamos en la teoría, puede realizarla tanto el protocolo IP, como TCP. **Este tema desde el punto de vista de la seguridad es vital**, pues si recordáis aún los ejercicios que hicimos a nivel "red" con HPING3 y FRAGROUTE, una de las funcionalidades es poder justamente "fragmentar" un datagrama en varios más pequeños..... Esto se llama "**Ataque de fragmentación**". Por ejemplo, si sobre un servidor no se desea que alguien se pueda conectar en remoto con la cuenta "root", pues existen formas de "bloquear" y/o "filtrar" y/o "monitorizar" esta palabra, pero si con el empleo de alguna de estas herramientas, se fracciona la misma para que viaje por ejemplo en cuatro datagramas, el primero con la "**r**", el segundo con la "**o**", el tercero con la "**o**" y el último con la "**t**", entonces esa palabra pasaría por los niveles de red como cuatro datagramas diferentes y se "reconstruiría" recién al llegar al destino a nivel de transporte y/o aplicación, donde sería tratada tal cual fue escrita en el nivel análogo de origen. Este es sencillamente uno de los miles de ejemplos de este tipo de ataques, pues los hay mucho más elaborados y complejos. Lo que intentamos transmitir aquí es que la ÚNICA FORMA de analizar estos ataques es a través de lo que se llama "**control o seguimiento de sesión**", es decir, una condición más que deberíamos pedirle a un FW es:

- **Control de sesión (o control de estado)**: Que mantenga en memoria, las secuencias TCP que van pasando por él para "reconstruir" estas sesiones y analizarlas AL COMPLETO.

NOTA: Recordemos aquí que justamente mencionamos sobre el protocolo "UDP" que desde el punto de vista de la seguridad, era "peligroso", aquí tal vez radica su mayor peligro. Como acabamos de mencionar, el "control de sesión", en UDP es imposible, pues no posee ningún campo para ello, y aquí está la razón por la cual aconsejamos que intentemos evitar su empleo en fronteras de red (en salidas y entradas desde redes de diferente nivel de seguridad).

En resumen y más allá de las clasificaciones teóricas, lo que nos interesa pedirle a un FW de red ahora que conocemos bien el modelo de capas y sus protocolos es:

- Poder "escuchar" la totalidad del tráfico que deseemos analizar.
- Estar en capacidad de "desarmar" los encabezados de cada protocolo.

- Tener patrones estandarizados para cada protocolo (para verificar su uso correcto).
- Contar con "elementos de juicio" para poder decidir qué es lo que dejará pasar y qué no.
- Capacidad de enrutamiento.
- Control de sesión (o control de estado).

Cuando tratemos el nivel de aplicación, veremos que dentro de la familia que llamamos "FW de host" muchos textos incluyen una posibilidad de FWs que la llaman "FWs de aplicación", pues pueden también analizar determinados patrones de tráfico de ese protocolo concreto del nivel de aplicación, este tipo de metodologías habitualmente se implementan a través de un servicio (o aplicación) que se llama "**Proxy**" y lo veremos en el capítulo siguiente.

En mucha bibliografía veremos más calsificaciones, generaciones y nomenclaturas de FWs, pero por ahora preferimos centrarnos en lo que hemos mencionado y poder avanzar de forma práctica sobre esta tecnología, pues con lo que hemos hablado ya tenemos más que suficiente.

Antes de implantar un FW, es aconsejable darle una mirada a la **RFC 2979**: Behavior of and Requirements for Internet Firewalls (Comportamiento y requerimientos para los cortafuegos de Internet) de octubre del 2000, pues en ella se hacer referencia y se citan algunos ejemplos de protocolos y filtrados que "deberían" o "no deberían" ser configurados para el correcto funcionamiento de Internet y que no por ello debilitan nuestra seguridad

6.3.2. ¿Cómo funciona un firewall?

Una vez comprendido todos estos conceptos que le "pediríamos" a un FW, ahora veremos cómo es su funcionamiento básico.

Un cortafuegos funciona en base a reglas que va recorriendo secuencialmente trama a trama y una a una verificando si cumple o no con ellas para luego adoptar una resolución.

Cuando se instala un FW no trae ninguna regla configurada, aunque hoy en día suelen traer diferentes escenarios o políticas por defencto para facilitar un poco la tarea, pero a los fines de entender y estudiar su funcionamiento, partiremos desde cero. Es decir, lo primero que debemos hacer es comenzar a definir sus reglas o el conjunto de ellas que suele denominarse "Política" de ese FW, y para ello existen dos grandes filosofías:

- Política permisiva.
- Política restrictiva.

La mayoría de las veces no suele ser una cuestión de elección, sino más bien de imposición, pues la decisión sobre la política a aplicar estará basada en la situación existente. Al instalar un FW en una zona determinada, sobre la cual la organización esté operativa desde hace tiempo, en general, se corre el riesgo que al aplicar cada una de las reglas se pueda dejar fuera de servicio alguna actividad, aplicación o servicio que se esté prestando, por lo que se debe ser muy meticuloso a la hora de aplicar y monitorizar cada una de ellas. Hemos visto en muchas empresas que directamente el factor principal es la "disponibilidad", por lo tanto un minuto de

tirar abajo una aplicación impacta económicamente a la organización, y antes de aplicar una regla se debe analizar con toda rigurosidad que no existan posiblidades de fallo.

La **política permisva** consiste en aceptar todo el tráfico inicialmente, y poco a poco comenzar a "ajustar" las reglas hasta llegar a la situación deseada. Una **política restrictiva** parte del supuesto inverso: "Negar todo", y paulatinamente ir abriendo caminos por medio de las reglas que se determine como necesarias, siempre a través de un detallado análisis.

Es evidente que la política restrictiva es la más robusta, pues partimos de la base que no entrará ni saldrá tráfico que no haya sido evaluado previamente, en cambio una política permisiva nos puede sorprender el paso de algún tipo de información que expresamente no hayamos tenido en cuenta, pero reiteramos una vez más: esta decisición en la mayoría de los casos dependerá de la situación reinante, y nos veremos obligados a adoptar una u otra.

6.3.3. Las reglas de un FW.

Como mencionamos, el funcionamiento de un FW es a través de las reglas que en él se vayan configurando, el conjunto de estas constituirá la "política" de este FW.

Las reglas, independientemente de la tecnología y/o producto que empleemos, responden a un esquema básico que generalmente es del siguiente tipo:

Intefaz IP_Origen Puerto_Origen "SENTIDO" IP_destino Puerto_Destino ACCIÓN

- La **intefaz**, puede ser exclusivamente una de las que cuente ese dispositivo, más de una o todas.
- Las **IPs** y los **Puertos**, son suficientemente claros. Sólo cabe la salvedad que suele representarse con "**Any**" cualdo es cualquiera (o todos) ellos, y también que generalmente permiten el empleo de "mascaras" de la forma "/" y también la concatenación del tipo 135-137 (puertos: 135, 136 y 137), la separación por medio de comas o punto y comas: 22, 23, 53, 110 (puertos 22, 23 53 y 110 exlcusivamente), etc.
- El "**Sentido**" se suele representar como: "in" (entrante), "out" (saliente) o "both" (ambos).
- La "**Acción**" se la relaciona habitualmente a: "Accept" (Aceptar), "Deny" o "Drop" (Negar), "PASS" (Pasar) y "Log" (guardar en logs).

Como se mencionó antes, las reglas se irán siguiendo secuencialmente hasta llegar a la última en cada una de las tramas que sean capturadas, si al llegar a la última, ninguna de ellas "ha aplicado", entonces no se adoptará ninguna acción. Por esta razón, es que se suele encontrar casi siempre como última regla algo similar a:

Interfaz_(Any) Any Any ←→ Any Any Deny

Con ello estamos negando cualquier trama que haya llegado hasta aquí, y con ello aseguramos que no pase nada más que lo que las reglas permiten. Prestad atención a que hemos dicho "casi sempre", pues justamente puede suceder (como parte de una política permisiva) que querramos

"filtrar" exlusivamente aglunas tramas, pero todo lo demás dejarlo pasar y en este caso no encontraremos esta regla (aunque como veremos en los ejercicios, existen mejores formas, siempre "negando" al final).

6.3.4. Firewall en Linux

Linux, prácticamente desde su nacimiento posibilitó el manejo de filtrado de paquetes, tanto hacia él mismo (como mencionamos, FW de hosts) como hacia otros hosts (FWs de red). Las primeras opciones que ofrecía se ejecutaban a través del comando "**ipfwadm**" y estaban incorporadas en el kernel (núcleo) hasta su versión 2.0. Un poco más adelante apareció "**ipchains**" que inició la era de NAT (Network Address Translation, ya tratado en el capítulo de red) dentro de los filtros de este sistema operativo y esta aplicación sobrevivió hasta el kernel 2.2. a partir del cual se produjo un cambio en la filosofía de lso FWs para Linux con la creación del "**framework Netfilter**" que en el año 2000 se incorpora al kernel 2.3.

- Framework: Conjunto/grupo de conceptos, recomendaciones, buenas prácticas y criterios, en general ya estandarizaodos enfocados a un tema específico, que en este caso es el "filtrado" y sirven como referencia.
- Netfilter: Proyecto que reúne todas las herrmientas, servicios y programas para implantar un FW en Linux.

A partir de Netfilter se incorpora definitivamente la posibilidad del "Control de estados" o "seguimiento de sesiones" por medio del empleo de "tablas" y así nace la herramienta que actualmente se continúa empleando a nivel administrador o usuario: "iptables". A menudo se suele confundir "iptables" con "netfilter", ante lo cual debemos aclarar que el rpimero es sólo una parte del conjunto "netfitler", pero po ser el de uso más cotidiano, muchos suelen olvidarse del grupo y referirse únicamente a la herramienta.

Como todo FW, su funcionamiento está basado en reglas, pero estas se agrupan en cadenas que a su vez forman parte de tablas.

Netfilter, de forma nativa ofrece tres tablas (por defecto es filter):

- **filter**: Filtrado.
- **NAT**: Conversión (traslado) de direcciones.
- **mangle**: manipulación de paquetes.

De forma nativa ofrece tres cadenas:

- **INPUT**: es para filtrar paquetes que vienen hacia este host.
- **OUTPUT**: es para filtrar paquetes que salen de este host.
- **FORWARD**: reencaminar paquetes.

La sintaxis básica de iptables es de la forma:

```
iptables -t filter -A INPUT <opciones>
iptables -A INPUT <opciones>
```

En la primera de estas dos reglas se establece la tabla "filter", pero como la misma es la que viene por defecto, se puede escribir e implementar de la misma forma tal cual figura en la segunda regla. Es decir ambas reglas son equivalentes.

La estructura completa de una regla básicamente sería:

iptables → -t →tabla →tipo_operación →cadena →regla_con_parámetros →Acción

Las operaciones básicas sobre las cadenas (existen más) son:

- -t (tabla): indica qué tabla se empleará.
- -i (interfaz de entrada): mismo formato que ifconfig.
- -o (interfaz de salida): mismo formato que ifconfig.
- -p (protocolo): tcp, udp, etc....
- -s, -d: dirección IP fuento o destino (Se debe aclarara máscara, ej:/24).
- -sport, -dport: puertos fuente o destino.
- -A (add) : agrega una regla al final de la cadena.
- -I (insert) : agrega una regla al principio de la cadena.
- -R (replace) : reemplaza una regla por otra.
- -D (delete) : elimina una regla.
- -F (flush) : elimina todas las reglas de la cadena. Es equivalente a borrar las reglas una por una.
- -L (list) : muestra las reglas dentro de la cadena.
- -j: Acción (ver abajo).
- -Z: pone a cero todos los contadores.

Las acciones básicas:

- ACCEPT : aceptar el paquete/transacción.
- DROP : rechaza el paquete/transacción
- REJECT : rechaza el paquete/transacción. A diferencia de DROP, notifica al emisor que el paquete/transacción fue descartado.

En esta parte del texto hemos querido sencillamente realizar la presentación teórica de "netfilter", luego en la parte de "ejercicios", encontrarás un buen número de ellos, comenzando

desde el empleo de "iptables" por línea de comandos (que es la mejor forma de aprenderlo) y luego a través de algunas herramientas gráficas.

NOTA FINAL SOBRE FWs: En la doctrina militar, existe un viejo lema que dice que:

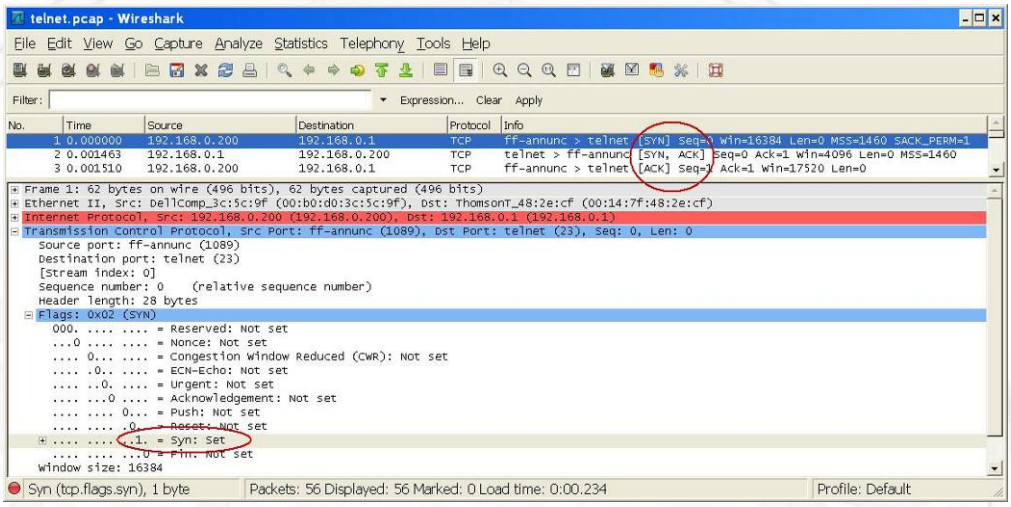
"TODO OBSTÁCULO DEBE SER CUBIERTO POR EL FUEGO"

Un alambrado, un campo minado, una fosa, un río, etc... Cualquier "barrera" cuya intención sea detener o demorar el avance del enemigo no tiene sentido, si a su vez no se la apoya con armas de fuego. Tal vez sea un detalle que no salte a simple vista o no se considere como para prestarle la atención suficiente, pero si nos detenemos un minuto en ello descubriremos que es fundamental, pues si el enemigo posee el tiempo y la libertad de acción suficiente, sin lugar a dudas sorteará el obstáculo tarde o temprano. Hasta un campo minado que a cualquiera puede parecerle letal, no lo es si se posee el tiempo y la tranquilidad necesaria para minimizarlo o neutralizarlo, es más se estudian y practican técnicas para ello hasta sin el empleo de tecnología, con un simple cuchillo y arrastrándose. Todo ello cambia substancialmente si en el momento de intentar sortear cualquier tipo de obstáculo, nos están abriendo fuego..... esto pasa a ser serio.

Esta nota viene a cuento de lo que hemos notado en un sinnúmero de empresas: colocan un FW y ¡¡¡queda desatendido!!!..... esto no tiene sentido. Un FW implica "sí o sí" una ardua tarea de mantenimiento, supervisión, actualización y monitorización (apoyo de fuego), sin esta actividad es como un alambrado abandonado. Por esta razón quisimos dejar esta última reflexión sobre este tema, pues a la hora de implementar esta tecnología debemos ser plenamente conscientes que acaba de empezar una actividad que debe ser integrada al "**ciclo de vida**" de la seguridad y JAMÁS DEJADA DE LADO.

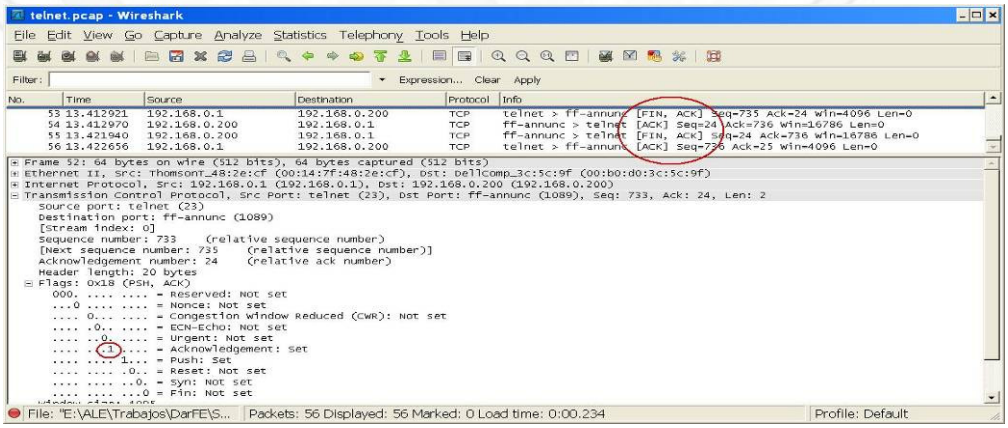
EJERCICIOS DEL CAPÍTULO 6 (Nivel de Transporte)

1. Una vez más recurriremos a Wireshark para analizar de forma práctica el funcionamiento de TCP. En este ejercicio, te invitamos a que captures tráfico TCP (por ejemplo haciendo una conexión a cualquier página Web, o enviando un correo electrónico) y le dediques unos momentos a identificar el establecimiento de una sesión TCP a través del triple Handshake (SYN – SYN ACK – ACK). Te presentamos a continuación una captura donde está remarcado dónde puedes comenzar tu análisis y te proponemos que lo hagas también tú con tu propia captura.

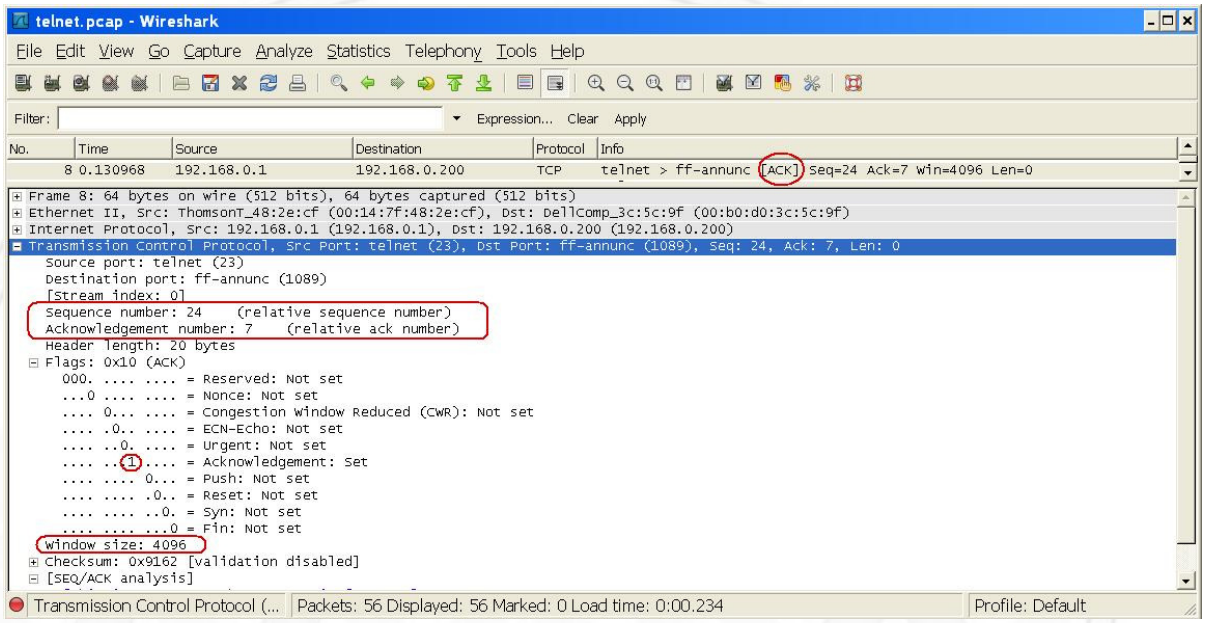


Como puedes apreciar, estas tres primeras tramas nos indican el establecimiento de una sesión TCP desde la dirección IP 192.168.0.200 con puerto fuente 1089 hacia la dirección IP 192.168.0.1 con puerto destino 23 (que como se verá más adelante es una conexión hacia el protocolo Telnet). EN rojo puedes ver arriba los tres pasos, y abajo el Flag "SYN=1" de la primer trama.

2. Este ejercicio es similar al anterior, pero en este caso el trabajo está en identificar el cierre normal de una sesión TCP. Nuevamente te presentamos abajo una captura para que tomes de referencia, y te proponemos que tú obtengas una similar.



3. En este ejercicio, te proponemos que investigues el funcionamiento de las secuencias TCP, este trabajo para comprenderlo en detalle te debería llevar un buen tiempo, pues no es tan fácil, por lo tanto, te invitamos a que comiences identificando en una captura, los campos que intervienen en el control de la secuencia y la técnica de ventana deslizante que se mencionó en la teoría. Para ello, una vez que tengas una captura, la mejor forma de avanzar, es comprender cómo se envían y reciben los números de secuencia y los "ACK", y luego relacionar todo ello con el control de tamaño de ventana. Cuando ya comprendas esta metodología, lo mejor que puedes hacer es profundizar sobre ella en Internet.



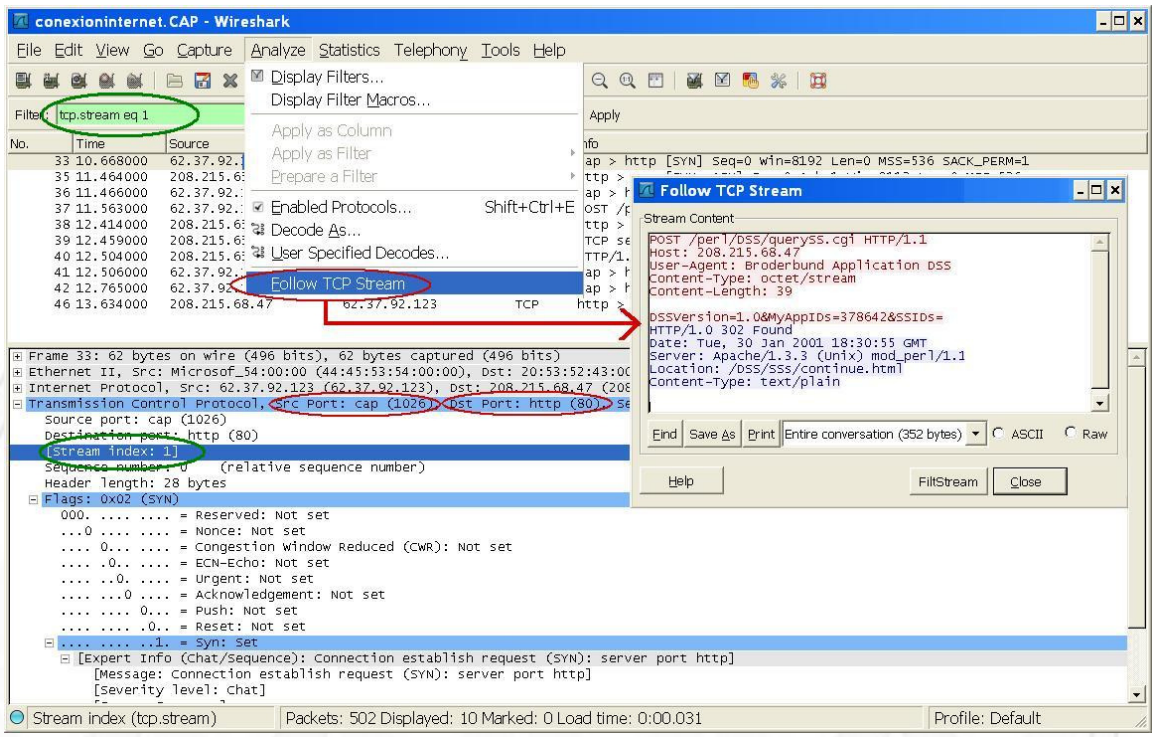
4. Un empleo muy potente de Wireshark es la posibilidad de "seguir flujos TCP", esta opción te permite filtrar todas las tramas correspondientes a una sesión TCP. En realidad lo que hace es verificar desde el primer segmento TCP con el flan "SYN=1" los puertos fuente y destino con sus correspondientes direcciones IP, y filtrar por estos campos.

   REFELXIÓN IMPORTANTÍSIMA: Recuerda siempre que las sesiones TCP (A diferencia de las UDP) tienen ¡¡¡SENTIDO!!!, es decir hay un host ORIGEN que establece la conexión (Con el primer paso del triple Handshake). Es decir, el SENTIDO con el que se establece una conexión TCP es desde ESE ORIGEN, hacia ESE DESTINO y esto nos permitirá, como veremos más adelante, "FILTRAR" las conexiones que entran o salen de nuestras redes siempre y cuando sean TCP, pues de ser UDP esto se nos complica (justamente por no poder establecer ese sentido).

   Lo que estamos haciendo en este ejercicio, **no es ni más ni menos que lo que hace un Firewall** (que se tratará más adelante) con control de estados. Es decir "filtra" y mantiene en memoria este establecimiento de cada sesión TCP y va "manteniendo el estado" de esa conexión hasta que la misma se cierra (flags: FIN – ACK) o por alguna razón se "resetea" (flan: RST) que es cuando las borra de su memoria caché. Aún tal vez no valores lo suficiente la trascendencia que tiene el comprender acabadamente este concepto, pero te aseguramos que es de suma importancia, por esa razón, es que no dejes pasar por alto este ejercicio, y tómatelo

con toda la seriedad que merece, filtrando, siguiendo e investigando todas las sesiones TCP que puedas, hasta que no te quede ninguna duda con ello.

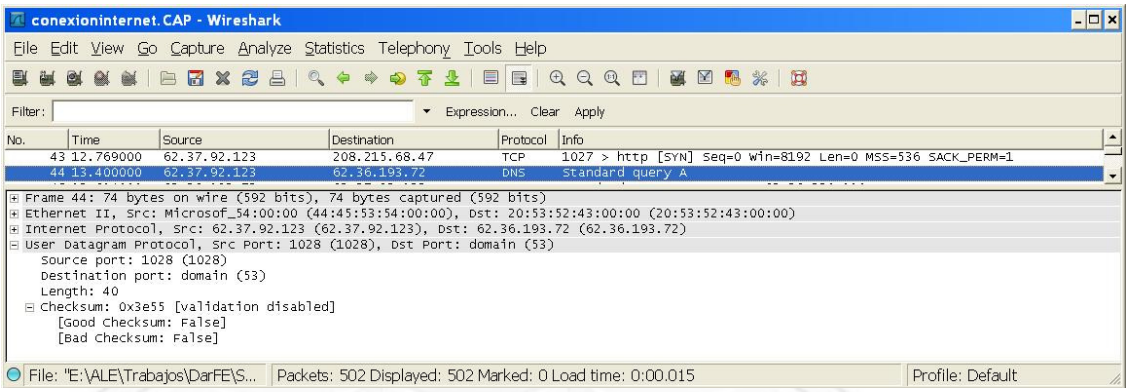
A continuación te presentamos una captura modelo, en la cual hemos destacado los pasos que debes seguir para analizar una secuencia TCP.



Como puedes apreciar, debes seleccionar (desde el menú superior) la opción “Analyze”, se te despegará una ventana donde encontrarás la opción “Follow TCP Stream”, esta opción filtrará toda la sesión TCP de la trama en la cual estés posicionado. Una vez “Clickeado” se te abrirá una nueva ventana similar a la que ves en primer plano (indicada con la flecha roja), con toda la información de ese flujo. Y a su vez como puedes apreciar (círculo verde) en la ventana principal, ya te ha generado un filtro de visualización (En nuestro ejemplo: “tcp.stream eq 1”) para ver únicamente los segmentos TCP de este flujo.

5. En este ejercicio pasamos a UDP, y te proponemos que hagas algunas capturas con este protocolo (por ejemplo al resolver nombres de dominio cuando abres un navegador y pones cualquier dirección URL, allí se genera un tráfico DNS que veremos más adelante, pero que por ahora te sirve para analizar el protocolo de nivel transporte UDP) y verifiques sus ocho octetos, tal cual lo tratamos en la parte teórica.

   A continuación te presentamos una captura para que tomes como referencia.



6. Ejercicios y prácticas sobre ataques TCP y UDP.

En esta práctica te describiremos brevemente qué tipo de ataques se pueden realizar en el nivel de transporte. Con las herramientas que hemos visto hasta ahora, ya estás en plena capacidad de realizarlos, si ves que con ellas no encuentras la solución, te proponemos que investigues por Internet hasta que la encuentres, pero insistimos en que ya estás perfectamente preparado para que los puedas hacer.

a. TCP Connect() Scanning --> SYN

Al enviar segmentos TCP con el "flag" SYN puesto a 1 (y el ACK=0), es decir solicitando el inicio del triple Handshake hacia un "socket" determinado con una dirección IP activa, pueden suceder muchas cosas: que el puerto esté cerrado, que esté administrativamente prohibido, que se esté filtrando, que exista un FW de red o de host, etc... Ante estos diferentes escenarios, los distintos hardware, SSOO y aplicaciones responden de manera diferente, y es uno de los modos más eficientes de iniciar el "fingerprinting" (huella digital) de un sistema.

b. Stealth port scanning (escaneo de puertos "sigiloso").

Hay veces en que el escaneo SYN no es lo suficientemente "prudente" como para pasar desapercibido. Algunos sistemas (Firewalls y filtros de paquetes) monitorizan la red en busca de paquetes SYN.

Para subsanar este inconveniente los paquetes FIN, en cambio, podrían ser capaces de pasar sin inadvertidos. Este tipo de Scaneo está basado en la idea de que los puertos cerrados tienden a responder a los paquetes FIN con el RST correspondiente. Los puertos abiertos, en cambio, suelen ignorar el paquete en cuestión.

c. "SYN Flood" (Inundación SYN).

Este ataque se basa en un incumplimiento de las reglas básicas del protocolo TCP por parte del cliente. Al iniciarse el triple "Handshake", como hemos visto, se envía el primer "SYN" El receptor de este, contesta con "SYN – ACK" y a partir de ese momento "inicia una conexión TCP" que queda en estado de "semi -abierta" hasta recibir el "ACK" final del triple handshake con lo cual queda abierta definitivamente,. Las conexiones "semi-abiertas"

implican parámetros almacenados en "memoria "caché" que caducarán al cabo de un cierto tiempo, liberando esos recursos. No obstante, si se envían muchas peticiones de conexión, de no estar adecuadamente configurado el "target", se desbordarán sus recursos.

d. Connection Flood

Es similar al anterior, pero esta vez sí se completa el "triple Handshake" reiterándolo una gran cantidad de veces intentando llegar al máximo de peticiones simultáneas que puede soportar el sistema objetivo.

e. Supernuke o Winnuke (OOB: Out Of Band)

Un ataque característico, y quizás el más común, de los equipos con Windows es el Nuke. Si bien ya hace que este problema quedó solucionado por este fabricante, no es raro aún encontrar alguno que aún lo sufre. Este ataque hace que los equipos que escuchan por el puerto NetBIOS sobre TCP/UDP 137 a 139, queden fuera de servicio, o disminuyan su rendimiento al enviarle paquetes UDP manipulados.

Generalmente se envían fragmentos de paquetes Out Of Band, que la máquina víctima detecta como inválidos pasando a un estado inestable. OOB es el término normal, pero realmente consiste en configurar el bit Urgente (URG) en los "flags" de TCP, al estar este bit puesto a uno, los bits del campo "Puntero de Urgente", se supone que deben indicar a partir de qué bit de los datos que transporta TCP comienzan los datos "urgentes" (repasar la teoría....). La idea es jugar con valores de datos inexistentes en este campo.

f. TCP Reset:

Este ataque intenta forzar el corte de una conexión entre dos máquinas, a través del envío del Flag "RST".

g. TCP Xmas.

Esta técnica es muy similar a las anteriores, y también se obtiene como resultado un paquete de reset si el puerto está cerrado. En este caso se envían segmentos TCP con los flags FIN, URG y PUSH puestos a "1", aunque también se lanza colocando a "1" los seis flags de TCP.

h. TCP Null scan.

En el caso de poner a cero todos los indicadores de la cabecera TCP, la exploración deberá recibir como resultado un paquete de reset en los puertos no activos.

i. Predicción de secuencia TCP:

Este ejercicio es complejo (debería estar en nuestra sección de desafíos). Las secuencias TCP, responden a un valor pseudoaleatorio con que se inician una vez completado el triple

"Handshake". Este valor nace de un concepto de "seed" (Semilla en inglés), y determinados SSOO y/o aplicaciones, no lo inician tan "aleatoriamente" como otros, facilitando que pueda predecirse y a su vez se comienzan a repetir superadas un poco más de cuatro horas. Si se escucha una suficiente cantidad de tráfico, permite "colarse" en una conexión TCP. Esta ataque fue y sigue siendo tan importante que llegó a tratarse a través de la **RFC-1918** "**Defending Against Sequence Number Attacks**" (defensa contra los ataques de número de secuencia), si quieres profundizar en él puedes verla en: http://tools.ietf.org/html/rfc1948.

j. SCAN UDP:

Se genera tráfico UDP a los diferentes puertos para detectar cuáles de ellos están abiertos y cuáles no.

k. UDP Flood (Inundación UDP).

Este ataque consiste en generar grandes cantidades de paquetes UDP contra la víctima elegida, la cual en general emitirá algún tipo de respuesta que también congestiona la red. Es usual dirigir este ataque contra máquinas que ejecutan el servicio "echo", de forma que se generan mensajes "echo" de un elevado tamaño.

EJERCICIOS CON HERRAMIENTAS

1. Empleo de "**nmap**". Nuevamente haremos uso de nmap, pero esta vez ya orientado al nivel de transporte. El trabajo que te proponemos aquí debe ser complementado con "Wireshark" para poder capturar las tramas de envío y recepción y poder comprender qué es lo que se está generando y la reacción del host destino, así que cada uno de las opciones de "nmap" que te presentamos deberás lanzarlas con el analizador de protocolos escuchando y capturando esas direcciones IP fuente y destino, para que de esta forma puedas seguir todo el rastro de lo ejecutado. Para ello ejercitaremos las siguiente opciones:

   - "**-sS**" (TCP SYN scan), con esta opción no se abrirá una conexión TCP completa, solo se envía un segmento con el "flag SYN" para saber si el puerto está abierto. ¿Cómo responde el destino si está abierto?, ¿Y si está cerrado?
   - "**-sT**" (TCP connect scan) este es el modo básico de escaneo que utiliza una función denominada "connect()" propia del sistema operativo. ¿Qué está enviando?, ¿Cuál es la lógica de este escaneo?, ¿Qué respuestas encuentras sobre puertos abierto y cerrados?
   - "**-sF -sX –sN**" (Stealth o sigiloso, conocido como Xmas (por Christmas) o árbol de navidad) es un modo de escaneo avanzado que en su momento fue muy conocido hoy tal vez no tan importante, pues los problemas que presentaba fueron casi todos

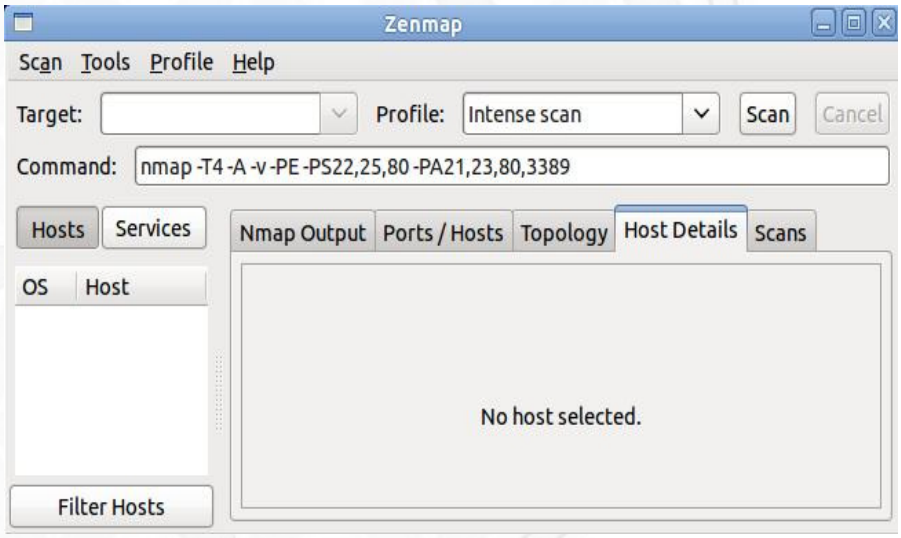
solucionados en los diferentes SSOO y aplicaciones. Prueba de lanzarlo sobre diferentes SSOO y aplicaciones, ¿Qué crees que está haciendo?, ¿Qué respuestas diferentes has obtenido?

- "**-sP**" (scan vía ping) Para saber que hosts están online. ¿Qué tipo de "ping" está generando?
- "**-sV**" (detectar versión) permite detectar diferentes versiones del servicios que se estén empleando. También lánzalo hacia diferentes plataformas y compara sus resultados
- "**-sU**" (scan UDP) escanea puertos UDP en vez de TCP.
- "**-sA**" (ataque ACK) Permite evaluar la existencia de reglas del tipo "iptables" (Que trataremos en el capítulo de Firewall).

2. Empleo de "**Zenmap**"

Zenmap es una herramienta gráfica muy amigable que se puede instalar tanto en Linux como en Windows muy fácilmente.

Por defecto ya te ofrece una serie de "opciones" ya preconfiguradas de scan, sobre las que únicamente debes seleccionar el "target" sobre el que lanzarás el escaneo. Como puedes ver en la imagen, el comando que se ejecutará será "nmap" puro y duro pero desde una interfaz amigable. Si ya has hecho los ejercicios con nmap, te resultará muy fácil de comprender.

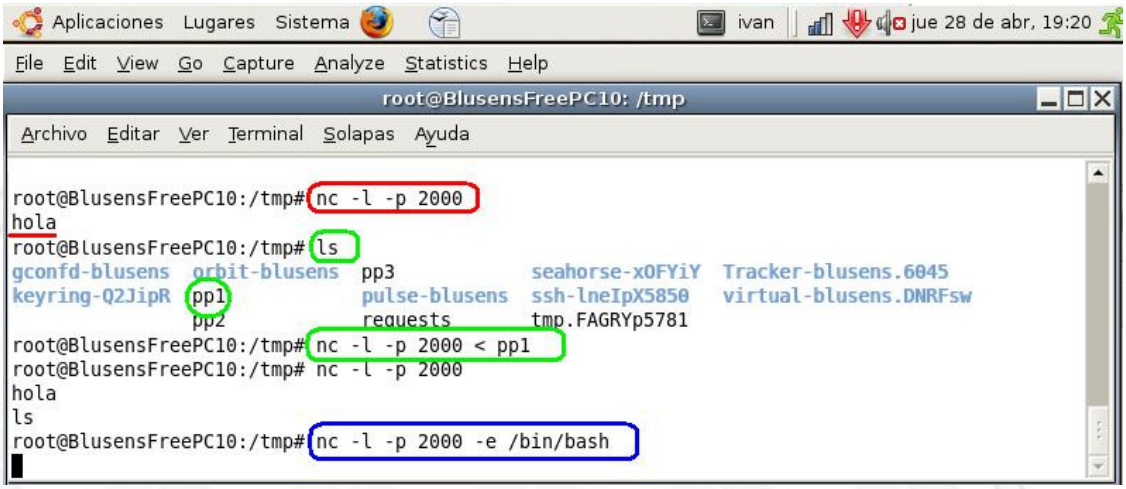


Te invitamos a que lo instales y ejecutes alguno de los scan que la herramienta te propone.

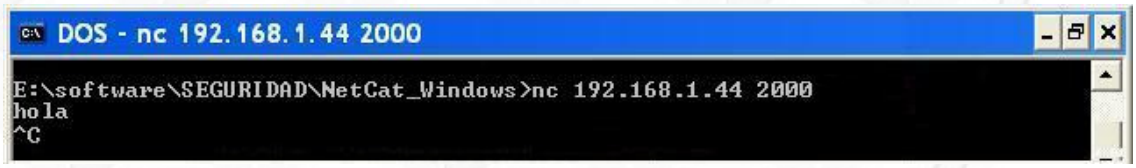
3. Comenzaremos a emplear otra de las herramientas importantes en seguridad "**netcat**". En estos primeros ejercicios, nos interesa que descubras su empleo básico que es justamente la creación de "sockets" entre máquinas, es decir lo que se trató en este capítulo.

- El primer ejercicio, será sencillamente establecer la conexión entre un cliente y un servidor, Para que aprecies con más detalle su funcionamiento, te presentaremos un ejemplo desde un host "Windows" y otro "Linux".

Lo primero que debes hacer es colocar el servidor en escucha sobre un puerto cualquiera (con la opción "**-l**", en nuestro caso emplearemos el puerto TCP 2000, y luego desde el cliente nos conectaremos al servidor:



En la imagen anterior puedes ver varios de los pasos que realizaremos en este ejercicio desde la interfaz de comandos de Linux (que opera como servidor). Al principio y remarcado en rojo puedes ver cómo se coloca en escucha el puerto 2000 (el cuál si no se especifica nada, por defecto es TCP), y en la siguiente línea (subrayado en rojo, se ve (como si fuera un Chat) el texto "hola" que se escribió desde la consola de Windows (como se muestra abajo).



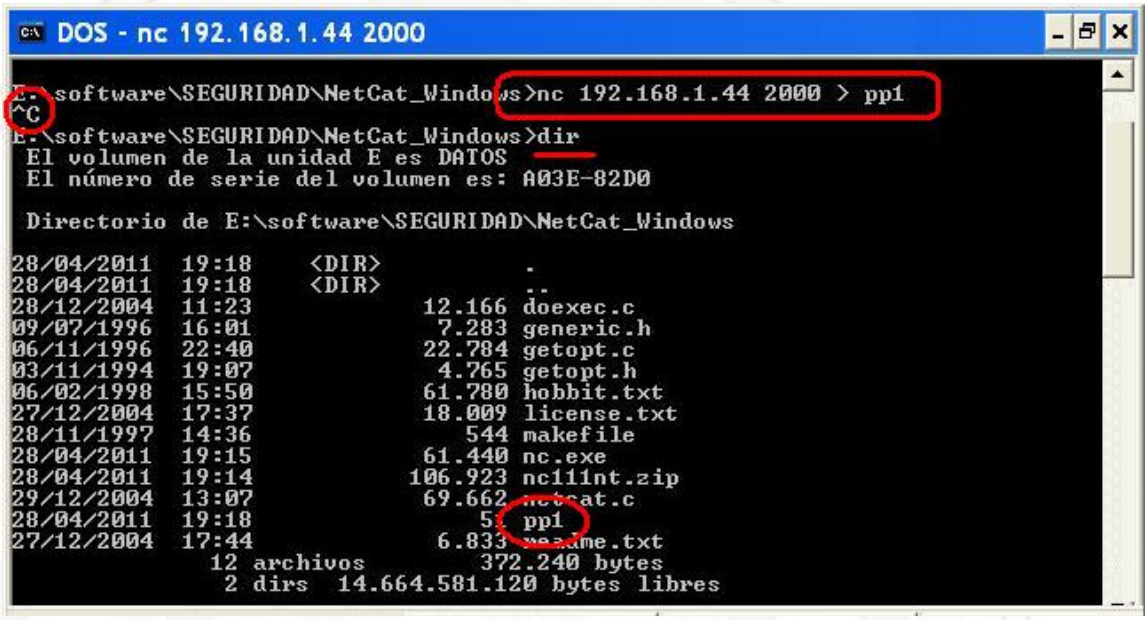
En la imagen anterior (color negra) puedes ver la interfaz de MSDOS de Windows, en la cual la primer línea se refiere al establecimiento de la conexión hacia el puerto 2000 del servidor (cuya dirección IP es 192.168.1.44), e inmediatamente después hemos pasado a escribir "hola" desde el teclado, que como has visto en la primer imagen, inmediatamente se reproduce en la consola del servidor.

- Nuestro próximo ejercicio, será la transferencia de un archivo desde el servidor hacia el cliente. En este caso debemos emplear el formato "redirector" de Linux (**"<"**) al establecer la conexión indicando en ambos extremos qué es lo que se desea enviar y transmitir. En nuestro caso hemos transferido un archivo que llamamos "**pp1**" que estaba en el servidor y como puedes verificar en las imágenes que siguen ha sido transferido al directorio MSDOS del cliente.

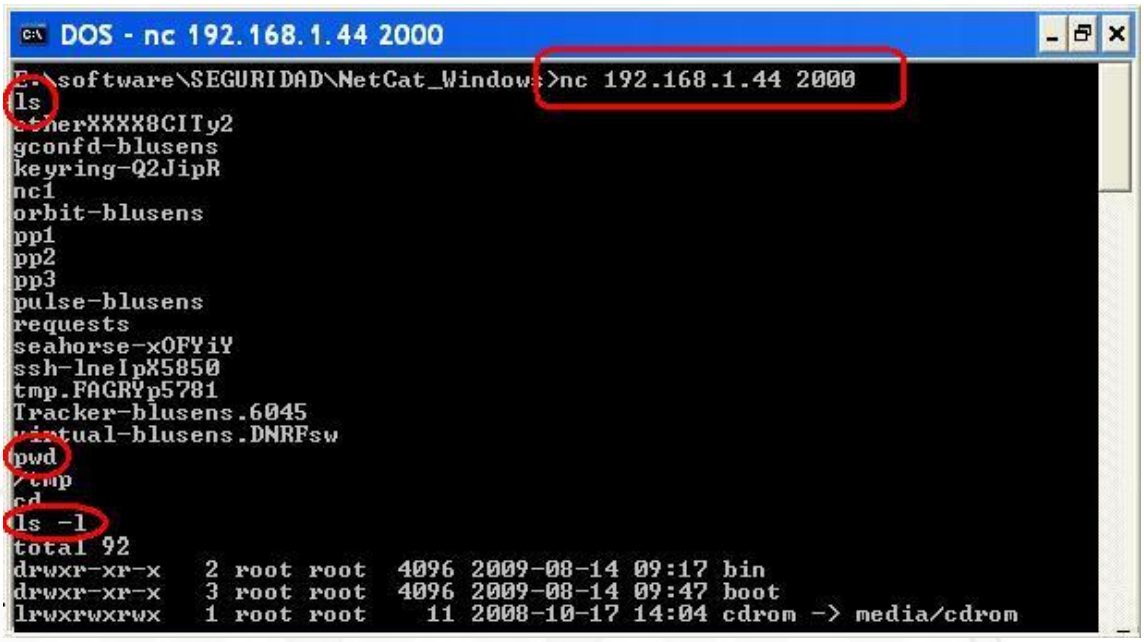
La conexión desde el lado del servidor, figura en la imagen de la consola (blanca) que está en los párrafos de arriba y responde al comando "**nc –l –p 2000 < pp1**", como puedes ver lo hemos recuadrado en verde y en las líneas anteriores hicimos un "ls" para que verifiques que el

archivo "pp1" está en el directorio Linux origen desde donde haremos la transferencia (todo ello remarcado en verde en la imagen anterior).

Aquí abajo te presentamos la conexión desde el lado cliente (Windows), remarcamos en rojo el comando que se ejecutó: "**nc 192.168.1.44 2000 > pp1**", una vez más puedes apreciar que nos conectamos al puerto 2000 y está vez empleamos el redirector para informarle que "recibiremos" el archivo "**pp1**". Luego puedes ver en la línea que sigue que hemos presionado las teclas "**[Ctrl + C]**" (Círculo rojo "**^C**") para salir de netcat, y al ejecutar "dir" (ya desde local) puedes ver que el archivo "pp1" ha sido transferido (También lo presentamos con un círculo rojo).



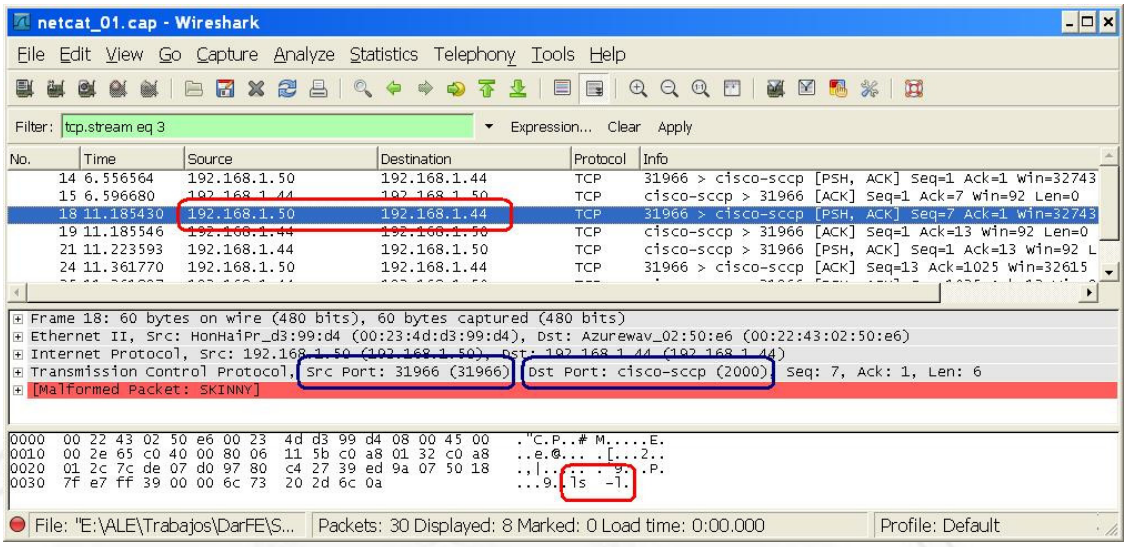
- Nuestro último ejercicio por ahora con "netcat" será abrir una "shell" (Consola Linux) remota desde nuestro cliente MSDOS.



Una vez más recurrimos a la imagen de la consola Linux (blanca) que pusimos en el primer párrafo y esta vez te pedimos que observes el último de los comandos que hemos remarcado (esta vez en azul), y pruebes de ejercitar algo similar al comando que ves allí, nosotros hemos ejecutado "**nc –l –p 2000 –e /etc/bash**". En este caso, nuevamente "escuchamos" (**-l**) el puerto (**-p**) 2000, e incluimos esta nueva opción de netcat "**-e**", que ordena ejecutar la apertura de una consola "bash" cuyo comando se encuentra en el directorio "/etc" (*justamente por este nombre es que en muchos textos encontrarás "programación en bash" cuando se programa directamente desde línea de comandos en Linux).*

Presta atención a este detalle, pues en la imagen anterior (negra) estás viendo una ventana MSDOS, pero ¿¿¿¿dentro de ella se están ejecutando comandos con formato Linux????? (con círculo rojo: "**ls**", "**pwd**", "**ls -l**"..... justamente lo que estás viendo es que en realidad esos comandos los estás ejecutando en la "shell" (o bash) del servidor remoto, no en tu consola MSDOS, y la orden "-e" te hace un eco en tu consola.

Abajo presentamos la captura con Wireshark, donde puedes apreciar la conexión hacia el puerto 2000 del servidor con la IP 192.168.1.44, desde la dirección IP 192.168.1.50 (cliente), y abajo hemos remarcado en rojo como en la captura puedes ver "viajar" en texto plano el comando "**ls -l**".



Te proponemos que practiques todas estas secuencias de empleo de la herramienta "netcat" que hemos presentado, para que comiences a familiarizarte con la apertura, mantenimiento y cierre de puertos y sockets, que es el tema tratado en este capítulo.

4. Empleo de la herramienta "**tcpdump**".

En muchos casos, puede sucedernos que no dispongamos de interfaz gráfica, que no podemos conectarnos en remoto a una de ellas, o que sencillamente no podamos instalar este tipo de software en un host. Para ello una solución que suele estar siempre al alcance de la mano y que viene instalado por defecto en muchas distribuciones Linux este comando "**tcpdump**", el cual es muy sencillo de utilizar, para ello te proponemos que lo ejercites con los ejemplos que te ponemos a continuación:

- Capturar tráfico cuya dirección IP de origen sea 192.168.1.44

Respuesta: tcpdump src host 192.168.1.44

- Capturar tráfico cuya dirección origen o destino sea 192.168.1.44

Respuesta: tcpdump host 192.168.1.44

- Capturar tráfico con destino a la dirección MAC 30:66:A2:6E:33:44

Respuesta: tcpdump ether dst 30:66:A2:6E:33:44

- Capturar tráfico con red origen 192.168.1.0/28

Respuesta: tcpdump src net 192.168.1.0 mask 255.255.255.240

(o también: tcpdump src net 192.168.1.0/28)

- Capturar tráfico con destino el puerto 22

Respuesta: tcpdump dst port 22

- Capturar tráfico con origen o destino el puerto 135

Respuesta: tcpdump port 135

- Capturar los paquetes de tipo ICMP

Respuesta: tcpdump ip proto \\icmp

- Capturar los paquetes de tipo UDP

Respuesta: tcpdump ip proto \\udp

(o también: tcpdump udp)

- Capturar solicitudes de DNS

Respuesta: tcpdump udp and dst port 53

- Capturar el tráfico al puerto telnet o SSH

Respuesta: tcpdump tcp and \(port 22 or port 23\)

- Capturar todo el tráfico excepto el web

Respuesta: tcpdump tcp and not port 80

(o también: tcpdump tcp and ! port 80)

5. Empleo de la herramienta "**iptables**".

Ya hemos desarrollado en la teoría el formato de "**iptables**", puedes consultarlo también con "man iptables" en cualquier distribución de Linux. La mejor forma de aprender a usarlo es justamente creando, aplicando reglas, y por supuesto verificando qué sucede desde otra máquina, así que para estos ejercicios, si bien puedes hacerlo con un equipo sólo, lo ideal es que cuentes con al menos dos hosts, para poder evaluar resultados.

Recordemos que la estructura completa de una regla básicamente sería:

iptables → -t →tabla →tipo_operación →cadena →regla_con_parámetros →Acción

Sobre el esquema anterior es que comenzaremos los ejercicios desde una consola Linux sobre el que ya esté instalado "iptables".

- Supongamos que deseamos rechazar los segmentos TCP que entran con destino a un servidor Web, el comando sería:

```
#iptables -t filter -A INPUT -p tcp -dport 80 -j DROP
```

Como la tabla por defecto es "filter", sería igual poner:

```
#iptables -p tcp -dport 80 -j DROP
```

¿Qué cambiarías a esta regla para que en vez de rechazar esto paquetes, los deje pasar?

Supongamos que en esta misma regla deseamos rechazar el puerto fuente TCP 80, y también el puerto destino TCP 80 ¿Cómo lo harías?

¿Si en vez de poner el número "80" pones "http", lo aceptará?

- Cada vez que desde una consola ejecutas: #**iptables -..... [enter]**, ingresas una nueva regla a "netfilter" y AUTOMÁTICAMENTE la misma entra en ejecución.

Es muy probable que en los intentos anteriores, te haya sucedido que aplicabas una regla, luego cambiabas algo y las cosas no respondían como querías. Esto sucede porque cada nueva regla que ingresaste, se "encola" en el sistema netfilter, por lo tanto tú creías que estabas ejecutando "esa" regla, cuando en verdad estabas ejecutando una "lista" de reglas, y en vez de cumplirse esa última se estaba cumpliendo alguna anterior.

Para poder ver todas las reglas que tienes ejecutándose debes escribir:

#iptables – L

Allí te listará todo lo que has escrito (siempre que su formato haya sido correcto) y se está ejecutando en tu host.

Tú que ya estás familiarizado con Linux, ¿Qué opción se te ocurre que puedes colocar inmediatamente después de "-L" para que te ofrezca más detalle este listado?

Ahora, ¿Te atreves a investigar cómo puedes borrar esas reglas?

Te invitamos a que profundices en esta tarea, pues puedes hacerlo una a una, de forma selectiva, todas ellas, etc... (Dedícale su tiempo).

- La opción "**-i**" permite asociar una regla a una determinada interfaz. Describe, ¿Qué acción realiza la siguiente regla?:

```
#iptables -I INPUT  -i lo -j ACCEPT
```

- ¿En qué consiste la opción "Masquerade"?
- ¿Cuál es el resultado de la siguiente regla?:

```
#iptables -A INPUT -s 172.18.1.100 -p TCP—dport 22 -j ACCEPT
```

- ¿Cómo debería ser la regla para permitir que todos los usuarios de la red: 192.168.10.0 con máscara: 255.255.255.0 puedan acceder al servidor Web de la empresa, que tiene instalado también "iptables"?
- Hasta ahora hemos tratado las reglas una a una, pero una potente opción que nos presenta la "programación bash" es redactar un sencillo archivo de texto plano, colocar todo el conjunto de reglas que queramos, asignarle los permisos correspondientes con "chmod" y luego ejecutarlo (./).

¿Te atreves a comenzar con uno de ellos que inicialmente contenga una regla sola y luego continuar sumándole más?

- Un trabajo con dos hosts:

Te proponemos a continuación una secuencia de tareas para que ejercites parte de lo visto hasta ahora desde dos hosts. Uno de ellos debería ser Linux con "iptables" y

el otro podría ser también Linux o cualquier otro sistema operativo. Te invitamos a que sigas esta secuencia paso a paso para que puedas analizar su evolución:

1) Realizar scan con **nmap** hacia los puertos tcp y udp de ambas máquinas.
2) Verificar con **nmap** localhost que los datos sean los correctos, hacerlo también con **Zenmap**.
3) Verificar desde una consola a través del comando "**netstat –etn",** qué puertos tengo establecidos con conexión (Established).
4) Si bien aún no hemos tratado este tema, pues lo veremos en el capítulo siguiente, vamos a instalar una herramienta muy sencilla que permite realizar conexiones remotas hacia ese host, que aún no haremos, pero nos permitirá verificar la apertura y cierre del puerto 23 (telnet). Para esta actividad, desde el host Linux debemos instalar "**telnetd**". Esta aplicación automáticamente al finalizar su instalación se ejecutará y dejará abierto el puerto TCP 23.
5) verificar nuevamente con **nmap** y **zenmap** ambos equipos.
6) Verificar cómo tengo las reglas de mi FW, ¿Qué comando empleo?
7) Borrar todas las reglas de mi FW ¿Qué comando empleo?
8) ¿Con qué regla de **iptables** puedo negar acceso al puerto 23? (en el host que se instaló telnetd).
9) ¿Con qué regla de **iptables** puedo negar acceso al puerto 23 a todo el mundo menos a la dirección IP del otro host?
10) ¿Con qué regla de **iptables** puedo negar la salida a navegar por Internet desde este mismo equipo?
11) ¿Qué sucede si apago este host y la vuelvo a encender?
12) ¿Es posible hacer un programa sencillo para que se niegue el acceso a este puerto Tcp 23, excepto al otro host y ejecutar directamente este programa, en vez de regla por regla?
13) Cómo se puede hacer para que este programa se inicie siempre que arranque este host?

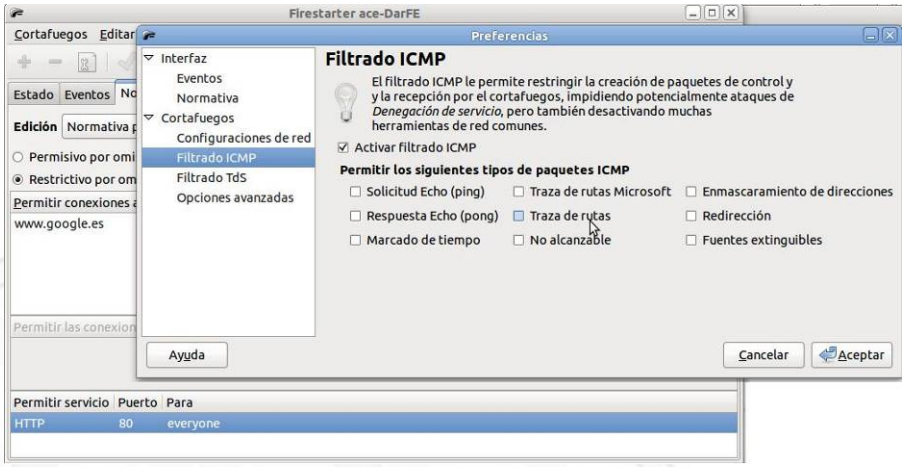
6. Empleo de la herramienta "**ufw**"

Una opción sencilla (tal vez la más sencilla) para comenzar nuestro camino hacia la "amigabilidad" de los FWs, es "**ufw**" (Uncomplicated FireWall) por defecto viene con Ubuntu (sino: apt-get install ufw).

Te invitamos a que lo pruebes y ejercites un poco antes de seguir nuestro avance sobre los FWs.

Como verás es muy fácil de usar, te proponemos que al menos realices las siguientes prácticas:

- Activar o desactivar (por defecto viene deshabilitado) .
  - Para activarlo: **#ufw enable**
  - Para desactivarlo: **#ufw disable**



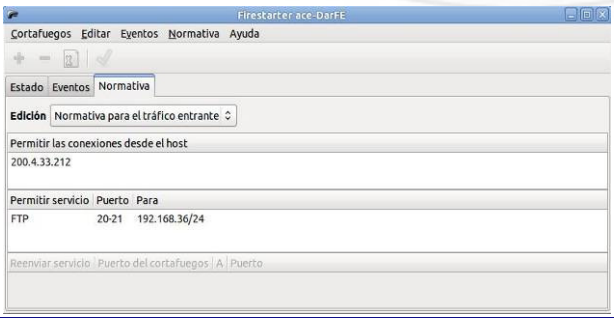
- Estando activado, es posible abrir o cerrar puertos (TCP o UDP).
  - Para abrir puertos TCP: **#ufw allow nnn/tcp**
  - Para abrir puertos UDP: **#ufw allow nnn/udp**
- Para cerrar puertos (TCP o UDP):
  - Para cerrar los puertos TCP: **#ufw deny nnn/tcp**
  - Para cerrar los puertos UDP: **#ufw deny nnn/udp**

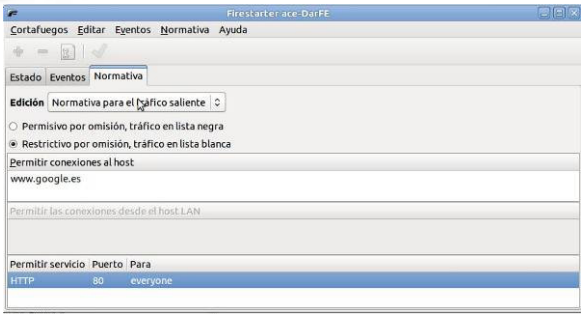
7. Empleo de la herramienta "**firestarter**".

Firestarter ya nos ofrece una interfaz gráfica un poco más amigable, desde la cual podemos configurar. En la imagen podemos apreciar en la ventana anterior, cómo se puede configurar reglas de filtrado, en este caso para el protocolo ICMP.

En la ventana posterior de esta misma imagen, se llega a ver también que nos ofrece la selección de políticas "restrictivas o permisivas", tal cual tratamos en la teoría.

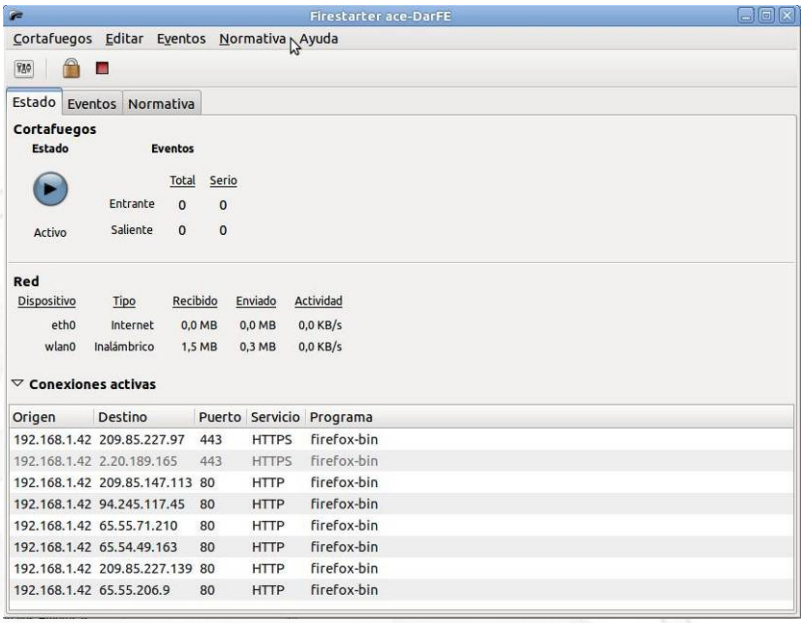
En las imágenes de abajo, vemos que se presenta la opción de configurar las reglas para el tráfico entrante o saliente tanto desde host y puertos, y cada uno de ellos posee los botones ("+", "-", etc) para implantar las mismas.





Por último, presentamos la imagen de "firestarter" funcionando (activo), en un ejemplo en el que se encuentra transmitiendo por la interfaz "**wlan0**" y nos presenta las conexiones activas.

Te proponemos que instales esta herramienta gráfica y la pruebes dedicándoles un tiempo para su análisis y evaluación, no encontrarás ninguna dificultad en ella.



8. Empleo de la herramienta "**Firewall Builder**".

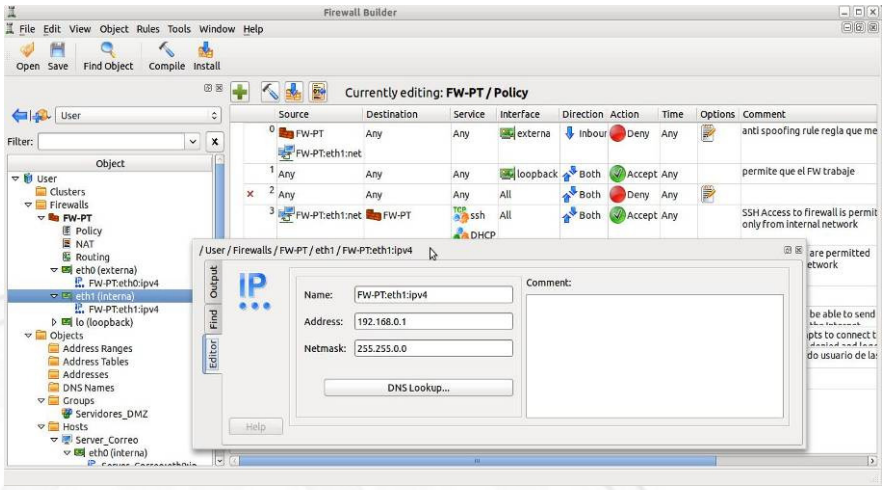
En nuestra opinión esta herramienta gráfica es una de las más potentes que podemos llegar a emplear en temas de seguridad con Firewalls. Existen ofertas comerciales de alto coste que no reúnen la potencia que ofrece Firewall Builder. En esta guía de ejercicios trabajaremos con la versión 4.2.

Para su instalación, puede ser que te encuentres con algún nivel de complejidad, dependiendo de la distribución de Linux que emplees, de su actualización y sobre todo de las librerías y paquetes que tengas instalado. Por nuestra parte, como hemos intentado hacer en todo el libro, no entraremos en temas de instalación pues suelen cambiar muy periódicamente, pero como siempre estamos seguros que es una muy buena práctica en temas de seguridad, que seas capaz de investigar a través de Internet hasta encontrar la solución a cada uno de estos problemas y desafíos. En el caso de "**firewall builder**" te aconsejamos que no escatimes esfuerzos hasta que lo tengas funcionando al 100% y no te quede ningún aspecto del mismo sin conocer pues será el día a día de todo administrador de sistemas que se quiera preciar de "experto en seguridad".

En pocas palabras, es una herramienta "excelente".

Todo este trabajo se puede ampliar, tomando como referencia la documentación (que es abundante) y se puede descargar gratuitamente de http://www.fwbuilder.org/. A riesgo de quedar desactualizados en poco tiempo, igualmente hemos decidido recomendar la guía: "Firewall Builder 4.0 User's Guide" de esta web, la cual citaremos en estos ejercicios.

Una vez instalado, desde consola, puedes ejecutar "**#fwbuilder**", se abrirá su interfaz gráfica y te presentará diferentes opciones de "escenarios" por defecto, te aconsejamos que elijas uno sencillo con no más de dos interfaces. Luego te preguntará sobre qué plataforma se trabajará, en nuestro caso seleccionamos "iptables" que es la que venimos tratando.



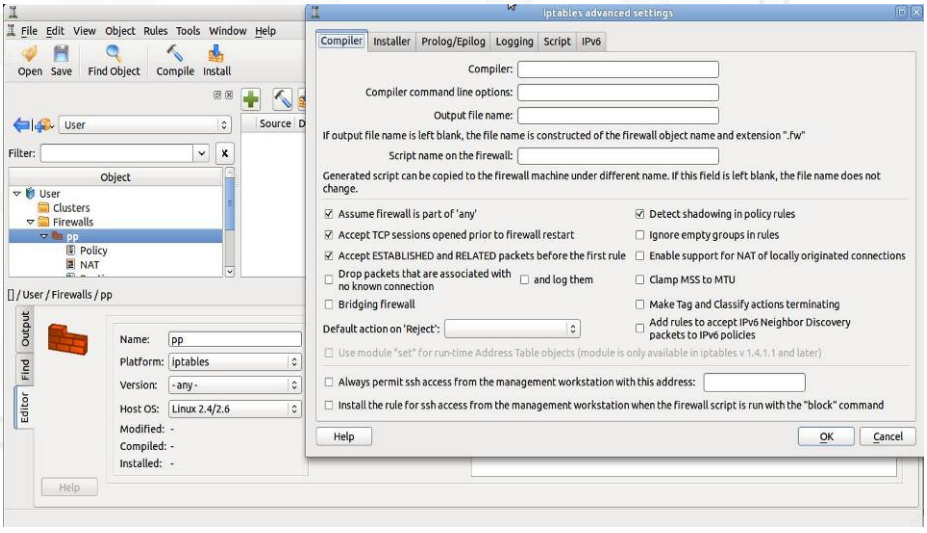
1) Una vez abierta la consola, el primer paso es crear un nuevo FW (nombre, interfaces, etc - Activar la opción "Plantilla por defecto"). (Puede consultarse en el capítulo 4 de la guía)

2) Para mantenerlo que aprendimos y ejercitamos con Packet tracer vamos a configurar las interfaces (eth0: Internet, 172.168.0.1/16 y eth1: LAN, 192.168.0.1/16).

Las interfaces, puedes editarlas y modificarlas las veces que lo desees, haciendo "doble clic" sobre ellas, se despliega la ventana de edición, como puedes ver en la imagen de aquí al lado.

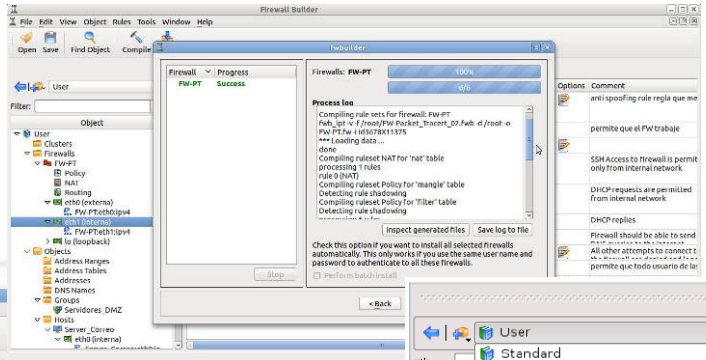
(La ventana de "edición" la puedes abrir sobre cualquier "objeto" haciendo "doble clic" sobre este).

3) Seleccionar el FW recientemente creado, editarlo (con doble "Click") y seleccionar el botón "Firewall Setting", se abrirá una nueva ventana "iptables advanced settings". En la parte inferir izquierda está el botón de "Help" que describe con todo detalle sus opciones. Analizar brevemente las mismas (como se ve en la primera de las imágenes).

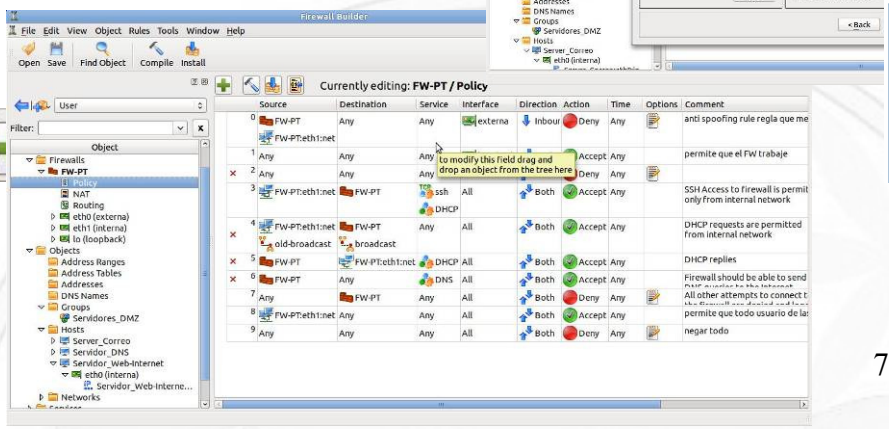


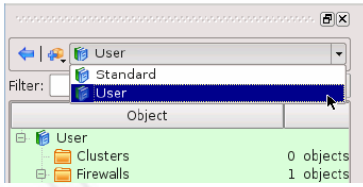
4) Al haber creado un FW, esta herramienta nos ofrece un conjunto de reglas por defecto que se corresponderán a la "política de este FW". Revisar y ajustar las reglas para la red interna 192.168.0.0/16 (Comentar TODO), el realizar comentarios sobre las reglas es fundamental pues nos servirá a futuro a nosotros o a cualquier otro que deba realizar cualquier tarea sobre el mismo. Como dato interesante, hemos visto FWs con cientos de reglas en los cuáles ya era imposible "adivinar" para qué estaban muchas de ellas.....

5) Cada regla puede Analizarse haciendo doble “click” en ella (se edita) y presenta tres “tabs”, en la indicada como “Output” se puede evaluar la misma en el formato “iptables” con que se ejecutará.



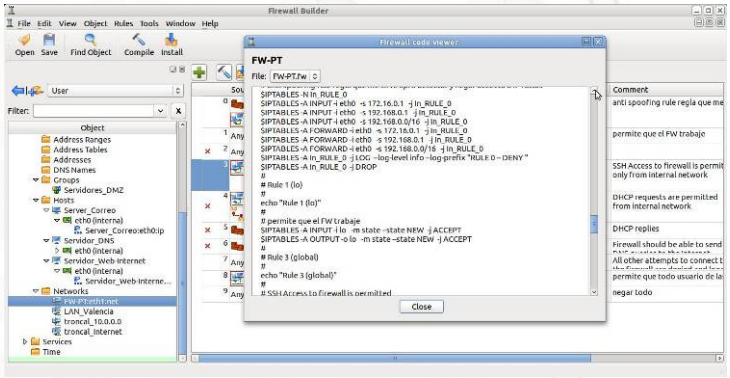
6) Cambiar de color las reglas que





rezcan más importantes (puede consultarse el capítulo 5 de la guía).

7) Analizar las Librerías “Standard” y “User”, ¿Qué servicios y objetos incorpora por defecto?



8) Para profundizar en el empleo de la GUI de FWBuilder, puede consultarse en la figura 5.30 de la guía)

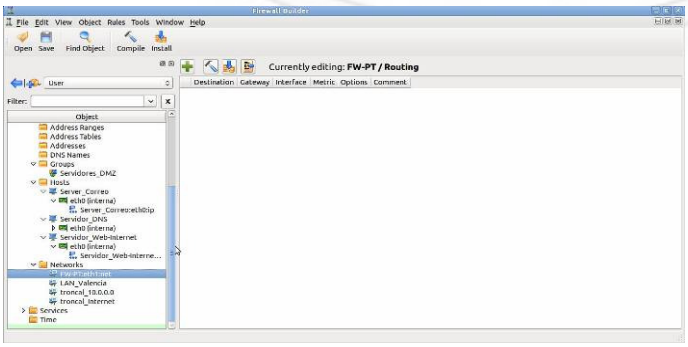
9) Crear nuevas redes (Valencia, Santiago, DMZ, etc) (puede consultarse el capítulo 5.19 de la guía). Practicar el “Drag and Drop” arrastrando los objetos que desees.

10) Crear troncales (10.0.0.0, 172.16.0.0).

11) Crear Hosts (Ser correo, Ser DMZ, etc).

12) Crear Grupos (Directorio, Administra-dores, etc...) (Asignarle a cada uno algunos hosts, que deberán ser creados).

13) Guardar.



14) Compilar reglas. Al compilar reglas, se puede apreciar que lo que se ha generado es un archivo que contiene cada una de las reglas en formato “iptables”, tal cual lo practicamos por línea de

comandos. Emplea la barra de herramientas:

15) Configurar diferentes opciones en la ventana "Filter" para visualizar únicamente ciertos rangos, nombres, interfaces, etc.

16) Analizar las opciones de "Logs".

17) Analizar "Inspeccionar Filas Generadas" (icono derecho de los 4 del menú). ¿Qué es esto? (Describir brevemente algunas reglas).

18) Si se desea profundizar sobre FWBuilder, una opción interesante y afín al trabajo que se viene realizando en el texto, es la del punto 4.2. de la guía "Configuring Cisco Router ACL", donde se describe la compatibilidad entre estos productos. (en la figura 4.5.3 se ve con claridad el empleo de las ACLs)

19) Volviendo al ejercicio anterior de Packet Tracer: Generar las siguientes reglas de filtrado:

   a. Nadie desde Internet puede ver las rutas troncales.
   b. Nadie desde Internet puede alcanzar las redes LAN.
   c. Desde Internet se puede llegar al puerto 25 del servidor de correo.
   d. Desde Internet se puede llegar al puerto 80 del servidor Web de Internet.
   e. Desde Internet se puede llegar al puerto 53 (tcp y udp) del Servidor DNS.
   f. Desde el Servidor DNS se puede salir hacia el puerto 53 (tcp y udp) de cualquier dirección.
   g. Desde el Servidor de correo se puede salir hacia el puerto 25 de cualquier dirección.
   h. Desde las LANs se puede salir únicamente al puerto 80 de cualquier dirección.

20) Supongamos que el FW estuviera filtrando también tráfico interno (cosa que no hace pues está en el segmento de salida hacia Internet), diseñar las siguientes reglas y grupos:

   a. Grupo Directivos.
   b. Grupo Gerencia Comercial.
   c. Grupo Gerencia Financiera.
   d. Grupo Administradores de sistemas.

21) Asignarles direcciones IP a algún host de cada LAN, e integrar esos hosts a los grupos anteriores.

22) Crear en "MZ" dos nuevos servidores de archivos: Datos-Financieros y Datos-Comerciales.

23) Crear las siguientes reglas:

   a. Únicamente los gerentes Comerciales pueden acceder al Servidor de Datos-Comerciales (al puerto 20 y 21).
   b. Únicamente los gerentes Financieros pueden acceder al Servidor de Datos-Financieros (al puerto 20 y 21).
   c. Los Directivos pueden acceder a todos (al puerto 20 y 21).

d. Cualquier administrador puede acceder a todos los hosts, servidores y troncales y a cualquier puerto pero por SSH (puerto 22).

24) El administrador general de toda la red, posee en su casa una línea ADSL con IP fija: 81.26.32.151. Desde allí puede ingresar a la red por SSH y conectarse a cualquier dispositivo, exactamente igual que si estuviera en la LAN.

FWBuilder de forma automática, ha generado la totalidad de las reglas y tablas que conforman la "política de seguridad de este FW", tened en cuenta que en nuestro ejercicio hemos creado un solo FW, pero podríamos haber creado todos los de nuestra organización en esta única interfaz gráfica, y a través de ella configurarlos y ajustarlos hasta el máximo nivel de detalle y FWBuilder se irá encargando de todo.

El detalle que no debemos pasar por alto ahora es el de "Cargar" los datos compilados en cada uno de los FWs y ejecutarlos, este es el último paso que debemos realizar, y también se realiza desde la interfaz gráfica, nuevamente desde la barra:

En nuestro ejercicio, como contamos con dos host, os aconsejamos que la primera vez lo ejecutéis en local, y luego de monitorizar su funcionamiento desde el otro host, si este a su vez también es Linux, probad de "crear" un nuevo "FW", configuradlo (de acuerdo a los pasos que acabamos de realizar), pero esta vez implantadlo en el host remoto.

Para finalizar todo este ejercicio con FWBuilder, a continuación os pegamos el formato que con sus más o sus menos debería quedar en un archivo de configuración compilado en formato "**iptables**" (hemos recortado con "....." muchas de sus partes, dejando únicamente las que consideramos te pueden servir de referencia en los pasos que acabamos de realizar):

```
#!/bin/sh
#
#  This is automatically generated file. DO NOT MODIFY !
#  Firewall Builder  fwb_ipt v4.2.0.3530
# Compiled for iptables (any version)
.....

FWBDEBUG=""
PATH="/sbin:/usr/sbin:/bin:/usr/bin:${PATH}"
export PATH
LSMOD="/sbin/lsmod"
IPTABLES="/sbin/iptables"
.....

LOGGER="/usr/bin/logger"
log() {
```

```
echo “$1”
command -v “$LOGGER” >/dev/null 2>&1 && $LOGGER -p info “$1”
      }
      .............

   reset_iptables_v4() {
$IPTABLES -P OUTPUT  DROP
$IPTABLES -P INPUT   DROP
$IPTABLES -P FORWARD DROP
      .............

   configure_interfaces() {
         :
# Configure interfaces
update_addresses_of_interface “eth0 172.16.0.1/16” “”
update_addresses_of_interface “eth1 192.168.0.1/16” “”
update_addresses_of_interface “lo 127.0.0.1/8” “”
      .............

# ================ Table ‘filter’, automatic rules
# accept established sessions
$IPTABLES -A INPUT   -m state—state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT  -m state—state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -m state—state ESTABLISHED,RELATED -j ACCEPT

   # ================ Table ‘nat’,  rule set NAT
         #
   # Rule 0 (NAT)
         #
   echo “Rule 0 (NAT)”
         #
   $IPTABLES -t nat -A POSTROUTING -o eth0  -s 192.168.0.0/16  -j SNAT—to-
source 172.16.0.1
   # ================ Table ‘filter’, rule set Policy
         #
```

```
# Rule 0 (eth0)
        #
echo “Rule 0 (eth0)”
        #
# anti spoofing rule (sirve para detectar y negar acceso a IP falsas)
$IPTABLES -N In_RULE_0
        $IPTABLES -A INPUT -i eth0   -s 172.16.0.1   -j In_RULE_0
        $IPTABLES -A INPUT -i eth0   -s 192.168.0.1   -j In_RULE_0
        $IPTABLES -A INPUT -i eth0   -s 192.168.0.0/16   -j In_RULE_0
        $IPTABLES -A FORWARD -i eth0   -s 172.16.0.1   -j In_RULE_0
        $IPTABLES -A FORWARD -i eth0   -s 192.168.0.1   -j In_RULE_0
        $IPTABLES -A FORWARD -i eth0   -s 192.168.0.0/16   -j In_RULE_0
    $IPTABLES -A In_RULE_0  -j LOG  --log-level info—log-prefix “RULE 0 -- DENY “
    $IPTABLES -A In_RULE_0  -j DROP
        .............

# All other attempts to connect to
# the firewall are denied and logged
$IPTABLES -N RULE_7
        $IPTABLES -A OUTPUT  -d 172.16.0.1   -j RULE_7
        $IPTABLES -A OUTPUT  -d 192.168.0.1   -j RULE_7
    $IPTABLES -A INPUT  -j RULE_7
    $IPTABLES -A RULE_7  -j LOG  --log-level info—log-prefix “RULE 7 -- DENY “
    $IPTABLES -A RULE_7  -j DROP
```

DESAFÍOS:

1. Investiga más opciones para el trabajo con puertos con “**nmap**”.
2. Realiza los trabajos que hemos hecho con “**netcat**” pero sobre puertos UDP.
3. Investiga la transferencia de archivos que realizamos con “**netcat**”, ¿Se puede realizar igualmente con todo tipo de archivos?
4. Investiga y practica la poderosa opción “**-c**” de “**netcat**”.
5. Investiga el empleo de “netcat” como escáner de puertos. Te proponemos que realices un escan de puertos aplicando “**nmap**” y que llegues a hacer lo mismo con “**netcat**”, ¿Te atreves?

6. Investiga cómo puedes hacer que tu archivo de reglas "iptables" se ejecute automáticamente cada vez que se inicia ese host (Una pista tienes relacionada al directorio /etc/init.d/).
7. Investiga el empleo de las tablas NAT en "FWBuilder".
8. Profundiza la administración remota de FWs con "FWBuilder".

CAPÍTULO 7: El nivel de APLICACIÓN

Como hemos comentado al principio, una vez superado el nivel cuatro (transporte), todas las funciones y/o servicios se orientan "de cara al usuario". Es decir, a partir de este nivel es poco probable que encontremos aspectos relacionados a la red, en cambio entraremos a lo que en el modelo TCP/IP engloba como "Aplicación", que os recordamos que aquí es donde existe la mayor diferencia con el modelo OSI que lo trata como tres capas diferentes (5: Sesión, 6: Presentación, 7: Aplicación).

En este nivel, trataremos los principales protocolos que lo componen. Tal vez sea el nivel donde mayor cantidad de protocolos existen, por esta razón es natural que alguien pueda pensar que este texto no está completo. Por nuestra parte, hemos decidido presentar los que figuran a continuación y en versiones posteriores del libro ir agregando los que podamos hasta lograr una visión lo más completa posible de este nivel.
Como hemos comentado al principio, una vez superado el nivel cuatro (transporte), todas las funciones y/o servicios se orientan "de cara al usuario". Es decir, a partir de este nivel es poco probable que encontremos aspectos relacionados a la red, en cambio entraremos a lo que en el modelo TCP/IP engloba como "Aplicación", que os recordamos que aquí es donde existe la mayor diferencia con el modelo OSI que lo trata como tres capas diferentes (5: Sesión, 6: Presentación, 7: Aplicación).

7.1. DNS (Domain Name System) (RFC 1706, 1591, 1034 y 1035):

Así como ya hemos visto que el protocolo ARP nos permitía asociar una dirección MAC, con una dirección IP, este protocolo ahora es quien permite la asociación de la dirección IP con el nombre que un usuario puede llegar a conocer o recordar. Veremos que los nombres que se emplean en Internet responde a un formato determinado, el cual queda establecido por la RFC 1591.

7.1.1. TLD (Top Level Domain) - (genéricos y geográficos).

Al nacer Internet, se implementó este servicio de nombres, a través de bases de datos de características planas, las cuales por el exponencial crecimiento de esta red, rápidamente obligó a estructurarse como un sistema jerárquico de archivos, residente en los servidores de nombres de dominio distribuidos en todo el mundo. El **NIC** (Network Information Center) lleva el registro mundial de todas las direcciones IP, y a que nombre se corresponde.

Dentro de esta jerarquía de nombres, el primer nivel se denomina Top Level Domain (**TLD**) y puede ser de dos formas, genéricos (gTLD: genericTLD) que se caracterizan por tres o cuatro caracteres, ellos son: com, mil, int, edu, net, gov, org y arpa o geográficos (ccTLD: country-codeTLD) que identifica al País de dos caracteres, por ejemplo ar, us, uk, br, etc, este

responde a los códigos internacionales de dos caracteres estandarizados por la norma **ISO 3166**.

En octubre de 2007, la Organización de Apoyo para Nombres Genéricos (**GNSO**:Generic Names Supporting Organization), uno de los grupos que coordina la política global de Internet en **ICANN** (Internet Corporation for Assigned Names and Numbers), completó su trabajo de desarrollo de políticas en los nuevos gTLD y aprobó una serie de recomendaciones. La Junta Directiva de la ICANN adoptó la política en junio de 2008.

Hay ocho gTLD que son anteriores a la creación oficial de la ICANN como organización. Estos son: .**com, .edu, .gov, .int, .mil, .net, .org** y .**arpa**. ICANN posteriormente hizo tres actualizaciones, una en el año 2000 y otra en 2003/4 donde se presentaron varias propuestas y fnalmente fueron aporbados: **.aero, .biz, .info, .coop, .name, .museum** y **.pro** en el 2000 y en el 2004: **.asia, .cat, .jobs, .mobi, .tel,** y **.travel**. La tercera fue en marzo de 2011 que acaba de ser aprobado también el gTLD: **xxx** para páginas relacionadas a pornografía.

La jerarquía se establece de derecha a izquierda, separada por puntos, avanzando en este orden de lo general a lo particular, llegando a identificar, por ejemplo, al usuario que pertenece al departamento de una empresa comercial de un determinado País (aperez.produccion.perezsa.com.ar), se estila el empleo del caracter @ si bien su uso no es obligatorio. Existen también niveles menores (regulados por la **RFC-1480**) que establecen subdominios como pueden ser localidades, colegios, librerías, agencias federales, etc.

En teoría, esta subdivisión puede tener hasta 127 niveles, y cada etiqueta contener hasta 63 caracteres, pero restringido a que la longitud total del nombre del dominio no exceda los 255 caracteres, aunque en la práctica los dominios son casi siempre mucho más cortos. Finalmente, la parte más a la izquierda del dominio suele expresar el nombre de la máquina (en inglés hostname).

Ver:

http://es.wikipedia.org/wiki/Dominio_de_Internet_gen%C3%A9rico

http://es.wikipedia.org/wiki/DNS

Los Internet Service Provider (**ISP**: Proveedores de Servicio de Internet) obtienen las asignaciones de direcciones IP de un registro local de Internet (**LIR**), del Registro Nacional de Internet (**NIR**), o del Registro Regional de Internet (**RIR**):

El papel de la **IANA** (Internet Assigned Numbers Authority) es asignar direcciones IP desde los rangos de las direcciones asignadas a los **RIR** de acuerdo a sus necesidades. Cuando un **RIR** requiere más direcciones IP para la asignación dentro de su región, **IANA** hace una asignación adicional al **RIR**. **IANA** no hace asignaciones directamente a los ISPs o usuarios finales salvo en circunstancias específicas, sino que lo hace a través de las cinco regiones en las que se ha dividido esta red mundial:

- RIPE: http://www.ripe.net

  (Réseaux IP Européens - en Francés: "IP para redes Europeas")

- AfriNIC: http://www.afrinic.net
  (Africa Numbers Internet Community)
- APNIC: http://www.apnic.net
  (Asia Pacific Network Information Centre)
- ARIN: https://www.arin.net/
  (American Registry for Internet Numbers)
- LACNIC: http://www.lacnic.net
  (Latin American and Caribean Internet Address Registry)

Dentro de los dominios genéricos, existen dos categorías: los **patrocinados** y los **no patrocinados**.

Los primeros reciben el apoyo de organismos privados, mientras que los segundos, por ser considerados de interés público, son mantenidos y regulados directamente por el ICANN y las entidades internacionales.

Los dominios genéricos no patrocinados, como .com, .name, .net, .org, .info, etc. siguen una estricta política y reglamentación fijada por el ICANN, que de cierta forma permite dar las máximas garantías de calidad al usuario final.

Los dominios genéricos patrocinados, como .aero, siguen una política y reglamentación fijada de forma compartida entre el ICANN y el organismo patrocinador.

No patrocinados

- .com, .net, .org, .biz, .info, .name, .edu, .gov, .int, .mil

Patrocinados

- .aero, .asia, .cat, .coop, .eu, .jobs, .mobi, .museum, .pro, .travel, .tel, xxx.

Infraestructuras **(Controladas por IANA directamente)**

- .arpa, .root

En fase de inicio

- .post, .mail.

En la página web:

http://www.icann.org/en/registrars/accredited-list.html

ICANN mantiene actualizado los agentes registradores reconocidos mundialmente especificando cada uno de los TLD sobre los que pueden operar.

Otra página Web que nos puede interesar es:

http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml

En ella encontraremos la distribución mundial del primer octeto de los rangos de direcciones IP y a qué zona geográfica pertenece.

7.1.2. Componentes principales de DNS.

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- Los Clientes DNS: Un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?);
- Los Servidores DNS: Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- Y las Zonas de autoridad, porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

Dentro de este sistema es que cuando se desea conectar con un determinado ETD, supongamos una Web, es probable que se conozca su nombre DNS pero casi seguro que no su dirección IP. En estos casos es que entra en juego el Sistema DNS, en el cual el primer Servidor DNS al cual el ETD se encuentra conectado, podrá conocer o no la dirección IP a la que se desea llegar, si la conoce, directamente la asocia por medio del sistema llamado "Solucionador de nombres", caso contrario elevará su solicitud en forma jerárquica al servidor de nombres de dominio inmediatamente superior a este, el cual procederá en forma análoga, hasta llegar al que posea esta información. El conjunto de nombres administrados por un servidor se llama ZONA, y por cada una de ellas existe un servidor primario y uno secundario (El cual resulta evidente al configurar un ETD para ingresar a Internet, pues obligatoriamente el Internet Service Provider al cual se llama deberá haberlos notificado para configurar el ETD correspondiente).

Acorde a la solicitud del cliente, un servidor DNS opera en forma **recursiva** o **iterativa (**definidas a través de un Flag en la solicitud cliente), La primera de ellas se produce cuando un servidor no conoce la dirección IP solicitada, entonces envía la misma a su Root, el cual operará de la misma forma hasta resolver la solicitud. Una solicitud Iterativa, es aquella en la cual ante la ausencia de respuesta, el servidor devuelve la misma al cliente pudiendo indicarle a que otro servidor recurrir o no, y es el cliente el responsable de ir iterando este pedido hasta encontrar solución.

El software más empleado para estos servidores es el **BIND** (Berkeley Internet Name Domain), que es un software de libre distribución empleado por los sistema Unix.

7.1.3. Tipos de registros DNS.

- **A** = Address – (Dirección) Este registro se usa para traducir nombres de hosts a direcciones IP.
- **CNAME** = Canonical Name – (Nombre Canónico) Se usa para crear nombres de hosts adicionales, o alias, para los hosts de un dominio. Es usado cuando se estan corriendo multiples servicios (como ftp y web server) en un servidor con una sola direccion ip. Cada servicio tiene su propia entrada de DNS (como ftp.ejemplo.com. y www.ejemplo.com.). esto también es usado cuando corres múltiples servidores http, con diferente nombres, sobre el mismo host.
- **NS** = Name Server – (Servidor de Nombres) Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- **MX (registro)** = Mail Exchange – (Registro de Intercambio de Correo) Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio.
- **PTR** = Pointer – (Indicador) También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IPs en nombres de dominio.
- **SOA** = Start of authority – (Autoridad de la zona) Proporciona información sobre la zona.
- **HINFO** = Host INFOrmation – (Información del sistema informático) Descripción del host, permite que la gente conozca el tipo de máquina y sistema operativo al que corresponde un dominio.
- **TXT** = TeXT - (Información textual) Permite a los dominios identificarse de modos arbitrarios.
- **LOC** = LOCalización - Permite indicar las coordenadas del dominio.
- **WKS** - Generalización del registro MX para indicar los servicios que ofrece el dominio. Obsoleto en favor de SRV.

- **SRV** = SeRVicios - Permite indicar los servicios que ofrece el dominio. RFC 2782
- **SPF** = Sender Policy Framework - Ayuda a combatir el Spam. En este record se especifica cual o cuales hosts están autorizados a enviar correo desde el dominio dado. El servidor que recibe consulta el SPF para comparar la IP desde la cual le llega, con los datos de este registro.

7.1.4. Zonas.

Cada servidor de nombres tiene autoridad para cero o más zonas. Hay tres tipos de servidor de nombres:

- Primario:

  Un servidor de nombres primario carga de disco la información de una zona, y tiene autoridad sobre ella.

- Secundario:

  Un servidor de nombres secundario tiene autoridad sobre una zona, pero obtiene la información de esa zona de un servidor primario utilizando un proceso llamado transferencia de zona. Para permanecer sincronizado, los servidores de nombres secundarios consultan a los primarios regularmente (típicamente cada tres horas) y re ejecutan la transferencia de zona si el primario ha sido actualizado. Un servidor de nombres puede operar como primario o secundario para múltiples dominios, o como primario para unos y secundario para otros. Un servidor primario o secundario realiza todas las funciones de un servidor caché.

- Caché:

  Un servidor de nombres que no tiene autoridad para ninguna zona se denomina servidor caché. Obtiene todos sus datos de servidores primarios o secundarios. Requiere al menos un registro NS para apuntar a un servidor del que pueda obtener la información inicialmente.

7.1.5. Los nodos raíz.

Existen 13 servidores raíz en todo Internet, cuyos nombres son de la forma letra.root-servers.org, aunque siete de ellos no son realmente servidores únicos, sino que representan múltiples servidores distribuidos a lo largo del globo terráqueo (ver tabla siguiente). Estos servidores reciben miles de consultas por segundo, y a pesar de esta carga la resolución de nombres trabaja con bastante eficiencia.

Los Servidor Raíz (root)

	Letra	Dirección IPv4	Nombre	Operador	Ubicación
1	**A**	198.41.0.4	ns.internic.net	VeriSign	*Distribuido empleando anycast*
2	**B**	192.228.79.201	ns1.isi.edu	USC-ISI	Marina Del Rey, California, U.S.
3	**C**	192.33.4.12	c.psi.net	Cogent Communications	*Distribuido empleando anycast*
4	**D**	128.8.10.90	terp.umd.edu	University of Maryland	College Park, Maryland, U.S.
5	**E**	192.203.230.10	ns.nasa.gov	NASA	Mountain View, California, U.S.
6	**F**	192.5.5.241	ns.isc.org	Internet Systems Consortium	*Distribuido empleando anycast*
7	**G**	192.112.36.4	ns.nic.ddn.mil	Defense Information Systems Agency	*Distribuido empleando anycast*
8	**H**	128.63.2.53	aos.arl.army.mil	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, U.S.
9	**I**	192.36.148.17	nic.nordu.net	Autonomica	*Distribuido empleando anycast*
10	**J**	192.58.128.30		VeriSign	*Distribuido empleando anycast*
11	**K**	193.0.14.129		RIPE NCC	*Distribuido empleando anycast*
12	**L**	199.7.83.42		ICANN	*Distribuido empleando anycast*
13	**M**	202.12.27.33		WIDE Project	*Distribuido empleando anycast*

Mientras que sólo trece nombres se utilizan para los servidores de nombres raíz, hay ás servidores físicos. Los servidores: A, C, F, I, J, K, L y M ya existen en varias ubicaciones en diferentes continentes, gracias al empleo de direccionamiento "**anycast**" para proporcionar

de servicios descentralizados. Como resultado hoy, la mayor parte de los servidores raíz se encuentran físicamente fuera de los Estados Unidos, lo que ha permitido obtener un alto rendimiento en todo el mundo. Para el que desee pensar mal, también significa que no es el único País que puede "ver pasar" absolutamente todo el tráfico mundial de Internet, pues en definitiva de eso se trata el control de todo este sistema DNS. Este será el responsable de decir a qué dirección IP se asigna cada nombre mundial, y por ser jerárquico, quien controla las raíces del árbol, controla todas las rutas.....

7.1.6. Formato de su encabezado.

El formato de un mensaje DNS es el siguiente:

16 bit	16 bit
Identificación	Parámetros
Número de Solicitud	Número de respuesta
Número de autoridad	Numero adicional
Sección Solicitud	
Sección respuesta	
Sección autoridad	
Sección de información adicional	

- Identificación: Identifica al mensaje, se emplea para llevar la correspondencia entre solicitudes y respuestas.
- Parámetros:
  * bit 1 (Q): Valor 0 = solicitud, valor 1 = respuesta.
  * bit 1 a 4 (OpCode) Tipo de consulta: Valor 0 = estándar, valor 1 = Inversa, valor 2 = solicitud de estado de servidor, (existen valor 3 y 4 en desuso).
  * bit 5 (A): Seteado si la solicitud es autoritativa (Flag de Autoritativo).
  * bit 6 (T): Seteado si el mensaje es truncado, es más largo de lo que permite el canal.
  * bit 7 (R): Seteado si se requiere recursión (Flag de recursividad).
  * bit 8 (V): Seteado si la recursión está disponible el servidor soporta recursión.
  * bit 9 a 11 (B): Reservados para uso futuro, su valor debe ser cero.
  * bit 12 a 15 (Rcode): (Tipo de respuesta) valor 0 = sin error, valor 1 = Error de formato en solicitud, valor 2 = falla en servidor, valor 3 = nombre inexistente, valor 4 = tipo de consulta no soportada, 5 = consulta rechazada.
- Numero de...: Lleva la cuenta del número de mensajes que se cursan en las secciones que le siguen en el formato.

- Sección solicitud: contiene las consultas deseadas, consta de tres sub-campos: Nombre de Dominio (longitud variable), Tipo de consulta (Host, mail, etc) y Clase de consulta (permite definir otros objetos no estándar en Internet).
- Sección respuesta, autoridad e información adicional: consisten en un conjunto de registros que describen nombres y sus mapeos correspondientes.

7.1.7. Conexiones UDP y TCP en DNS:

Los mensajes DNS se transmiten sobre UDP o sobre TCP, en ambos sobre el puerto 53, y la mecánica a seguir es la siguiente:

- Los mensajes transportados por UDP se restringen a 512 bytes. En el caso de TCP el mensaje va precedido de un campo de 2 bytes que indica la longitud total de la trama.
- Un "resolver" del DNS o un servidor que envía una consulta que no supone una transferencia de zona *debe* enviar una consulta UDP primero. Si la sección "answer" de la respuesta está truncada y el solicitante soporta TCP, debería intentarlo de nuevo usando TCP. Se prefiere UDP a TCP para las consultas porque UDP tiene un factor de carga mucho menor, y su uso es esencial para un servidor fuertemente cargado. El truncamiento de mensajes no suele ser un problema dados los contenidos actuales de la base de datos del DNS, ya que típicamente se pueden enviar en un datagrama 15 registros, pero esto podría cambiar a medida que se añaden nuevos tipos de registro al DNS.
- TCP debe usarse para actividades de transferencia de zonas debido a que el límite de 512 bytes de UDP siempre será inadecuado para una transferencia de zona.
- Los servidores de nombres deben soportar ambos tipos de transporte.

7.1.8. Inundación recursiva e iterativa:

La metodología de esta debilidad es la de aprovechar la potencia que tiene este protocolo para obtener información de algún nombre, para generar mayor cantidad de táfico en la red. Si se alcanzan volúmenes importantes, se habla de inundación, pues puede llegar a dejar fuera de servicio a una red o servidor, ocupándose únicamente de esta actividad. Todo aquel que haya desarrollado programas que implementen técnicas recursivas sabe bien la potencia que estas poseen (y seguramente los desbordamientos de memoria que sin lugar a dudas alguna vez le ocasionó por error). En el caso de la iteración, el problema es similar, pues se plantean casos en los cuales por falta de resolución se produce la "Iteración eterna", es decir que nunca dejará de soicitar un determinado nombre.

7.1.9. Herramientas empleadas con este protocolo.

- **Nslookup**: Es un cliente DNS que sirve para obtener direcciones IP a través del dominio y viceversa (Ej: nslookup www.google.es o nslookup 209.85.229.104)
- **Dig:** (Domain Information groper) otro comando flexibla para interrogar DNSs
- **host:** Sencilla a rápida resolución DNS

Página Web amigable para resoluciones DNS: http://www.nictools.net/

7.1.10. Vulnerabilidades de DNS.

Las vulnerabilidade sde este protocolo, pueden clasificarse en cuatro grandes familias:

- UDP: Entre los servidores se transfieren grandes volúmenes de información a través del puerto UDP 53, el cual por ser no orientado a la conexión lo hace especialmente difícil de controlar y filtrar, aprovechándose esta debilidad.
- Obtención de Información: Los servidores DNS almacenan información importante, la cual es muy buscada y fácilmente obtenible por un intruso.
- Texto plano: Toda la infromación viaja en texto plano.
- Falta de autenticación: El protocolo no ofrece ninguna técnica de autenticación.

7.1.11. DNS Spoof.

La técnica de spoof, ya la hemos presentado y se puede implementar en muchos protocolos y niveles, en este caso consiste en falsificar unarespuesta DNS, ofreciendo una dirección IP que no es la que verdaderamente está relacionada con ese nombre. Lo natural sería infectar o envenenar las tablas de un servidor DNS maesrto, y con ello se propagaría y cualquier consulta hacia el nombre infectado, lo repsondería con la dirección IP falsa. La realidad es que es relativamente difícil esta tarea, pues los DNS maestros de Internet suelen estar bastante asegurados y monitorizados como para que esta actividad, inclusive en el caso de lograrla, no sea detectada de forma bastante rápida y solucionado.

Pero como suele suceder, a veces existen atajos para estas acciones, en este caso, más que infectar al servidor, sería mucho más sencillo infectar al cliente, e inclusive no es necesario modificar las direcciones DNS primario y secundario que este tiene configurado, sino sencillamente modificando su base de datos local, pues al igual que la caché arp, DNS también maneja una tabla dinámica, de forma tal que si cerramos y abrimos nuevamente una página web, no se realice una nueva petición DNS. También posee una tabla estática que es

el archivo "**hosts**" (tanto en Windows como en Linix), en Windows se encuentra en: C:\WINDOWS\system32\drivers\etc\hosts, por defecto, viene configurado como lo mostramos a continuación:

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Éste es un ejemplo de archivo HOSTS usado por Microsoft TCP/IP para Windows.
#
# Este archivo contiene las asignaciones de las direcciones IP a los nombres de
# host. Cada entrada debe permanecer en una línea individual. La dirección IP
# debe ponerse en la primera columna, seguida del nombre de host correspondiente.
# La dirección IP y el nombre de host deben separarse con al menos un espacio.
#
#
# También pueden insertarse comentarios (como éste) en líneas individuales
# o a continuación del nombre de equipo indicándolos con el símbolo "#"
#
# Por ejemplo:
#
#      102.54.94.97     rhino.acme.com          # servidor origen
#       38.25.63.10     x.acme.com              # host cliente x

127.0.0.1       localhost
```

En el caso de Linux, podemos ver la misma estructura y se encuentra en "/etc/hosts".

Fijaros que si se logra insertar una línea en cualquiera de ellos, no se realizará la petición DNS, sino que sencillamente se dirigirá hacia la dirección IP que en este archivo figure. Lo practicaremos en la parte de ejercicios.

Las siguientes son algunas de lar RFC referidas a DNS:

RFC 1032 - Guía de administrador de DNS

RFC 1033 - Guía de las operaciones de administrador de DNS

RFC 1034 - Nombres de dominio - Conceptos y servicios

RFC 1035 - Nombres de dominio - Implementación y especificación

RFC 1101 - Codificación DNS de nombres de red y de otros tipos

RFC 1183 - Nuevas definiciones del DNS RR

RFC 1706 - Registros de recursos DNS NSAP

7.2. Telnet (Terminal remota)(RFC 854, 855 y 857):

7.2.1. Conceptos de telnet.

Este protocolo es el que hace posible el acceso a terminales remotas, operando las mismas como si fueran locales. Los comandos Telnet los usa el protocolo, no los usuarios debido a que el papel de Telnet es conectar al usuario, y que este se comunique en forma directa. Estos comandos se envían en un paquete llamado **command** que contiene 2 o 3 octetos, y son los que establecen la conexión. Una vez realizada la misma, habitualmente se solicitará un nombre de usuario y una contraseña, pues se está disponiendo el uso de los recursos de ese equipo. Era muy común su empleo para consultas BBS, a terminales en Internet y también para la administración remota de dispositivos de hardware como suelen ser Hub, Switch, Router, etc.

TELNET es un protocolo basado en tres ideas:

- El concepto de **NVT** (*Network Virtual Terminal:* Terminal virtual de red). Una NVT es un dispositivo imaginario que posee una estructura básica común a una amplia gama de terminales reales. Cada host mapea las características de su propia terminal sobre las de su correspondiente NVT, y asume todos los demás hosts harán lo mismo.
- Una perspectiva simétrica de las terminales y los procesos.
- Negociación de las opciones de la terminal. El protocolo TELNET usa el principio de opciones negociadas, ya que muchos host pueden desear suministrar servicios adicionales, más allá de los disponibles en la NVT. Se pueden negociar diversas opciones. El cliente y el servidor utilizan una serie de convenciones para establecer las características operacionales de su conexión TELNET a través de los mecanismos "DO, DON'T, WILL, WON'T"("hazlo, no lo hagas, lo harás, no lo harás").

Opciones negociadas de Telnet

Solicitud	Respuesta	Interpretación
DO	WILL	El remitente comienza utilizando la opción El remitente no debe
	WON'T	El remitente no debe utilizar la opción
WILL	DO	El remitente comienza utilizando la opción, después de enviar *DO*
	DON'T	El remitente no debe utilizar la opción
DON'T	WON'T	El remitente indica que ha desactivado la opción
WON'T	DON'T	El remitente indica que el remitente debe desactivar la opción

Existen 255 códigos de opción. De todas maneras, el protocolo Telnet proporciona un espacio de dirección que permite describir nuevas opciones.

7.2.2. La noción de terminal virtual

Cuando surgió Internet, la red (ARPANET) estaba compuesta de equipos cuyas configuraciones eran muy poco homogéneas (teclados, juegos de caracteres, resoluciones,

longitud de las líneas visualizadas). Además, las sesiones de los terminales también tenían su propia manera de controlar el flujo de datos entrante/saliente.
Por lo tanto, en lugar de crear adaptadores para cada tipo de terminal, para que pudiera haber interoperabilidad entre estos sistemas, se decidió desarrollar una interfaz estándar denominada como ya vimos NVT. Así, se proporcionó una base de comunicación estándar, compuesta de:

- caracteres ASCII de 7 bits, a los cuales se les agrega el código ASCII extendido;
- tres caracteres de control;
- cinco caracteres de control opcionales;
- un juego de señales de control básicas.

Por lo tanto, el protocolo Telnet consiste en crear una abstracción del terminal que permita a cualquier host (cliente o servidor) comunicarse con otro host sin conocer sus características.

7.2.3. La negociación.

Como se mencionó con anterioridad, este protocolo trabaja sobre TCP, es decir que primero se establece una sesión TCP y luego la conexión TELNET. Una vez realizada la misma, el cliente entra en una fase de negociación dinámica que determina las opciones de cada lado de la conexión, que justamente por ser dinámicas es que en cualquier momento pueden ser modificadas. Esta negociación se lleva a cabo por un conjunto de comandos TELNET, los cuales son precedidos por un caracter intérprete de comando (IAC) que es un octeto compuesto por todos unos (FF hex) y luego sigue el código de comando, y en el caso que este posea opción continuará un tercer octeto de opción negociada:

IAC	Command Code	Option Negotiated

7.2.4. Comandos y códigos.

A continuación se incluye la lista de códigos de comandos:

Comando	Dec	Hex	Descripción
End subNeg	240	FO	End of option subnegotiation command
No Operation	241	F1	No operation command
Data Mark	242	F2	End of urgent data stream.
Break	243	F3	Operator pressed the Break key or the Attention key.
Int process	244	F4	Interrupt current process
Abort output	245	F5	Cancel output from current process.
You there?	246	F6	Request acknowledgment

Erase char	247	F7	Request that operator erase the previous character.
Erase line	248	F8	Request that operator erase the previous line.
Go ahead!	249	F9	End of input for half-dúplex connections.
SubNegotiate	250	FA	Begin option subnegotiation
Will Use	251	FB	Agreement to use the specified option
Won't Use	252	FC	Reject the proposed option.
Start use	253	FD	Request to start using specified option.
Stop Use	254	FE	Demand to stop using specified option
IAC	255	FF	Interpret as command.

En el caso que los comandos posean opciones negociables, las mismas son identificadas por una Option ID, la cual sigue inmediatamente después del comando. Las Option ID son las que se detallan a continuación:

Dec	Hex	Código de opción	Descripción
0	0	Binary Xmit	Allows transmission of binary data.
1	1	Echo Data	Causes server to echo back all keystrokes.
2	2	Reconnect	Reconnects to another TELNET host.
3	3	Suppress GA	Disables Go Ahead! command.
4	4	Message Sz	Conveys approximate message size.
5	5	Opt Status	Lists status of options.
6	6	Timing Mark	Marks a data stream position for reference.
7	7	R/C XmtEcho	Allows remote control of terminal printers.
8	8	Line Width	Sets output line width.
9	9	Page Length	Sets page length in lines.
10	A	CR Use	Determines handling of carriage returns.
11	B	Horiz Tabs	Sets horizontal tabs.
12	C	Hor Tab Use	Determines handling of horizontal tabs.
13	D	FF Use	Determines handling of form feeds.
14	E	Vert Tabs	Sets vertical tabs.
15	F	Ver Tab Use	Determines handling of vertical tabs.
16	10	Lf Use	Determines handling of line feeds.
17	11	Ext ASCII	Defines extended ASCII characters.
1	12	Logout	Allows for forced log-off.
19	13	Byte Macro	Defines byte macros.
20	14	Data Term	Allows subcommands for Data Entry to be sent.
21	15	SUPDUP	Allows use of SUPDUP display protocol.
22	16	SUPDUP Outp	Allows sending of SUPDUP output.
23	17	Send Locate	Allows terminal location to be sent.
24	18	Term Type	Allows exchange of terminal type information.
25	19	End Record	Allows use of the End of record code (0xEF).
26	1A	TACACS ID	User ID exchange used to avoid more than 1 log-in.

27	1B	Output Mark	Allows banner markings to be sent on output.
2	1C	Term Loc#	A numeric ID used to identify terminals.
29	1D	3270 Regime	Allows emulation of 3270 family terminals.
30	1E	X.3 PAD	Allows use of X.3 protocol emulation.
31	1F	Window Size	Conveys window size for emulation screen.
32	20	Term Speed	Conveys baud rate information.
33	21	Remote Flow	Provides flow control (XON, XOFF).
34	22	Linemode	Provides linemode bulk character transactions.
255	FF	Extended options list	Extended options list.

7.2.5. Vulnerabilidades:

Si bien posee otras de menor magnitud, la que jamás se debe olvidar es que absolutamente todo bajo este protocolo va como Texto Plano, es decir, que se lee con total libertad.

RFC relacionadas con Telnet:

RFC 854 Especificaciones del protocolo Telnet.
RFC 855 Especificaciones de opciones de Telnet
RFC 856 Transmisión binaria en Telnet
RFC 857 Opción Eco de Telnet
RFC 858 Opción de suprimir continuación en Telnet
RFC 859 Opción Estado de Telnet
RFC 860 Opción Marca de tiempo de Telnet
RFC 861 Opción Lista extendida de opciones de Telnet

7.3. FTP (File Transfer Protocol) (RFC 959).

Este protocolo permite la transferencia remota de archivos sin establecer una sesión Telnet.

7.3.1. Establecimiento de la conexión y empleo de puerto de comando y puerto de datos.

Al igual que Telnet, es un protocolo que está pensado para velocidad y no para seguridad, por lo tanto toda su transmisión se realiza en texto plano, incluyendo usuario y contraseña. Para solucionar este problema son de gran utilidad aplicaciones como **scp** y **sftp**, incluidas en el

paquete **SSH** (Secure Shell) que veremos más adelante, que permiten transferir archivos pero cifrando todo el tráfico.

Servidor FTP:

Un servidor FTP es un programa especial que se ejecuta en un equipo servidor normalmente conectado a Internet (aunque puede estar conectado a otros tipos de redes, LAN, MAN, etc.). Su función es permitir el intercambio de datos entre diferentes servidores/ordenadores.

Por lo general, los programas servidores FTP no suelen encontrarse en los ordenadores personales, por lo que un usuario normalmente utilizará el FTP para conectarse remotamente a uno y así intercambiar información con él.

Cliente FTP:

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar archivos en un ordenador remoto, se necesitará utilizar un programa cliente FTP. Un cliente FTP es un programa que se instala en el ordenador del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.

Para utilizar un cliente FTP, se necesita conocer el nombre del archivo, el ordenador en que reside (servidor, en el caso de descarga de archivos), el ordenador al que se quiere transferir el archivo (en caso de querer subirlo nosotros al servidor), y la carpeta en la que se encuentra.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Windows, DOS, Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

7.3.2. Tipos de transferencia de archivos en FTP.

Es importante conocer cómo debemos transportar un archivo a lo largo de la red. Si no utilizamos las opciones adecuadas podemos destruir la información del archivo. Por eso, al ejecutar la aplicación FTP, debemos acordarnos de utilizar uno de estos comandos (o poner la correspondiente opción en un programa con interfaz gráfica):

- type ascii

  Adecuado para transferir archivos que sólo contengan caracteres imprimibles (archivos ASCII, no archivos resultantes de un procesador de texto), por ejemplo páginas HTML, pero no las imágenes que puedan contener.

- type binary

  Este tipo es usado cuando se trata de archivos comprimidos, ejecutables para PC, imágenes, archivos de audio...

Ejemplos de cómo transferir algunos tipos de archivo dependiendo de su extensión:

EXTENSION DEL ARCHIVO	TIPO DE TRANSFERENCIA
txt (texto)	ASCII
html (página WEB)	ASCII
Doc (documento)	Binario
ps (poscript)	ASCII
Hqx (comprimido)	ASCII
Z (comprimido)	Binario
ZIP (comprimido)	binario
ZOO (comprimido)	binario
Sit (comprimido)	binario
pit (comprimido)	binario
shar (comprimido)	binario
uu (comprimido)	binario
ARC (comprimido)	binario
tar (empaquetado)	binario

7.3.3. Funcionamiento.

Una característica particular de su funcionamiento es que emplea dos puertos (dos canales TCP), el puerto 20 por medio del cual transfiere datos, llamado Data Transfer Process (DTP), y el puerto 21 por medio del cual transmite las instrucciones de comando llamado Protocol Interpreter (PI). Al igual que telnet, los comandos los usa el protocolo y no el usuario; estos comandos son secuencias en ASCII de cuatro caracteres (QUIT, PASS, PORT, DELE, LIST, ABORT, etc). Las conexiones FTP se inician de manera similar a Telnet con el nombre o dirección del Host destino (Ej: ftp 205.29.24.11), luego se debe registrar como usuario válido (en algunos se suele emplear la cuenta anonymous o guest) y generalmente como cortesía se emplea como contraseña la cuenta de correo electrónico, para permitirle al administrador llevar un registro de accesos. Luego se define un directorio de inicio, un modo de transferencia de datos (ASCII o binario), se inicia la transferencia y por último se detiene.

Las tramas de control FTP, son intercambios TELNET y contienen los comandos y opciones de negociación mencionadas en el punto anterior, sin embargo la mayoría de los mismos son simples textos en ASCII y pueden y pueden ser clasificados en comandos y mensajes FTP, los cuales se detallan a continuación:

7.3.4. Comandos.

Comando	Descripción
ABOR	Abort data connection process.
ACCT <account>	Account for system privileges.

ALLO <bytes>	Allocate bytes for file storage on server.
APPE <filename>	Append file to file of same name on server.
CDUP <dir path>	Change to parent directory on server.
CWD <dir path>	Change working directory on server.
DELE <filename>	Delete specified file on server.
HELP <command>	Return information on specified command.
LIST <name>	List information if name is a file or list files if name is a directory.
MODE <mode>	Transfer mode (S=stream, B=block, C=compressed).
MKD <directory>	Create specified directory on server.
NLST <directory>	List contents of specified directory.
NOOP	Cause no action other than acknowledgement from server.
PASS <password>	Password for system log-in.
PASV	Request server wait for data connection.
PORT <address>	IP address and two-byte system port ID.
PWD	Display current working directory.
QUIT	Log off from the FTP server.
REIN	Reinitialize connection to log-in status.
REST <offset>	Restart file transfer from given offset.
RETR <filename>	Retrieve (copy) file from server.
RMD <directory>	Remove specified directory on server.
RNFR <old path>	Rename from old path.
RNTO <new path>	Rename to new path.
SITE <params>	Site specific parameters provided by server.
SMNT <pathname>	Mount the specified file structure.
STAT <directory>	Return information on current process or directory.
STOR <filename>	Store (copy) file to server.
STOU <filename>	Store file to server name.
STRU <type>	Data structure (F=file, R=record, P=page).
SYST	Return operating system used by server.
TYPE <data type>	Data type (A=ASCII, E=EBCDIC, I=binary).
USER <username>	User name for system log-in.

7.3.5. Mensajes (Son las respuestas a los comandos):

Código	Descripción
110	Restart marker at MARK yyyy=mmmm (new file pointers).
120	Service ready in nnn minutes.
125	Data connection open, transfer starting.
150	Open connection.
200	OK.
202	Command not implemented.

211	(System status reply).
212	(Directory status reply).
213	(File status reply).
214	(Help message reply).
215	(System type reply).
220	Service ready.
221	Log off network.
225	Data connection open.
226	Close data connection.
227	Enter passive mode (IP address, port ID).
230	Log on network.
250	File action completed.
257	Path name created.
331	Password required.
332	Account name required.
350	File action pending.
421	Service shutting down.
425	Cannot open data connection.
426	Connection closed.
450	File unavailable.
451	Local error encountered.
452	Insufficient disk space.
500	Invalid command.
501	Bad parameter.
502	Command not implemented.
503	Bad command sequence.
504	Parameter invalid for command.
530	Not logged onto network.
532	Need account for storing files.
550	File unavailable.
551	Page type unknown.
552	Storage allocation exceeded.
553	File name not allowed.

7.3.6. Modos de conexión.

FTP admite dos modos de conexión del cliente. Estos modos se denominan **Activo** (o Estándar, o PORT, debido a que el cliente envía comandos tipo PORT al servidor por el canal de control al establecer la conexión) y **Pasivo** (o PASV, porque en este caso envía comandos tipo PASV). Tanto en el modo Activo como en el modo Pasivo, el cliente establece una conexión con el servidor mediante el puerto 21, que establece el canal de control.

Modo Activo:

En modo Activo, el servidor siempre crea el canal de datos en su puerto 20, mientras que en el lado del cliente el canal de datos se asocia a un puerto aleatorio mayor que el 1024. Para ello, el cliente manda un comando PORT al servidor por el canal de control indicándole ese número de puerto, de manera que el servidor pueda abrirle una conexión de datos por donde se transferirán los archivos y los listados, en el puerto especificado.
Lo anterior tiene un grave problema de se.guridad, y es que la máquina cliente debe estar dispuesta a aceptar cualquier conexión de entrada en un puerto superior al 1024, con los problemas que ello implica si tenemos el equipo conectado a una red insegura como Internet. De hecho, los cortafuegos que se instalen en el equipo para evitar ataques seguramente rechazarán esas conexiones aleatorias. Para solucionar esto se desarrolló el modo Pasivo.

Modo Pasivo:

Cuando el cliente envía un comando PASV sobre el canal de control, el servidor FTP abre un puerto efímero (cualquiera entre el 1024 y el 5000) e informa de ello al cliente FTP para que, de esta manera, sea el cliente quien conecte con ese puerto del servidor y así no sea necesario aceptar conexiones aleatorias inseguras para realizar la transferencia de datos.
Antes de cada nueva transferencia, tanto en el modo Activo como en el Pasivo, el cliente debe enviar otra vez un comando de control (PORT o PASV, según el modo en el que haya conectado), y el servidor recibirá esa conexión de datos en un nuevo puerto aleatorio (si está en modo pasivo) o por el puerto 20 (si está en modo activo).

7.3.7. T_FTP (Trivial FTP).

Existe un protocolo de transferencia de archivos diseñado para operar en modo No Orientado a la Conexión que es el TFTP (Trivial) (RFC: 783, 1350), el cual difiere del FTP en que no se registra en la máquina remota y que opera sobre UDP en lugar de TCP. Se define en el puerto número 69, y es común su empleo en ETD que no poseen disco rígido para cargar aplicaciones o programas fuente. Posee un conjunto de comandos y parámetros que se detallan a continuación:

Comando	Descripción
Read Request	Request to read a file.
Write Request	Request to write to a file.
File Data	Transfer of file data.
Data Acknowledge	Acknowledgement of file data.
Error	Error indication

Parámetro	Descripción
Filename	The name of the file, expressed in quotes, where the protocol is to perform the read or

	write operation.
Mode Datamode	The format of the file data that the protocol is to transfer.

7.3.8.Vulnerabilidades.

FTP, es uno de los primeros protocolos de la familia y por esta razón, nace en una época en la cual la seguridad no era un problema. Este origen lo hace particularmente vulnerable.

Toda la comunicación, al igual que Telnet, viaja en texto plano, desde la cuenta de usuario, la contraseña, hasta los comandos y los datos.

La mejor y más práctica alternativa en la actualidad es su empleo por medio de SSL/TLS, como se verá más adelante.

Como medidas a tomar en el servidor, siempre se debe:

- Revisar permanentemente la configuración del servidor y de ser posible,
- emplear software de verificación de archivos (tipo **Tripware**).
- No colocar contraseñas encriptadas en el archivo etc/passw en el área ftp anónimo.
- Prestar especial atención a la configuración anónimo.
- Actualizar permanentemente el servidor.
- Nunca colocar archivos del sistema en el directorio ~ftp/etc.
- Que nunca cincidan los nombres de cuentas del directorio ~/ftp/etc/passwd con el /etc/passwd.
- No activar TFTP si no es estrictamente necesario.

7.4. SSH (Secure SHell)

7.4.1. Presentación e historia

SSH, o Secure Shell, podríamos traducirlo como “Intérprete de comandos seguro”, recordemos que el concepto de “**shell**” es el de la interfaz de comandos nativa de Linux. Es decir, SSH nos presenta una metodología a través de consola, en la cual podemos trabajar de forma segura, y veremos que segura desde la autenticación misma hasta el cierre de la sesión.

Ya hemos visto los dos protocolos básicos, que podríamos decir dieron origen a Internet: “FTP y Telnet”, al principio la red fue así, no existían interfases gráficas y todo lo necesario para administrar equipos eran estos dos protocolos, existían también los comandos “r” (remoto) de

Unix, que no los tratamos en este texto, pos su reconocida debilidad. Cuando el tema de la seguridad comenzó a cobrar importancia, se hace evidente que se debía hacer algo y así nace SSH. Este protocolo lo implementa por primera vez el Finlandés Tatu Ylöen como SSH versión 1 en el año 1995 como versión de software libre, pero a fin de ese mismo año crea la compañía SSH Communications Security, y la licencia cambia, manteniéndose gratuito para uso de investigación y particular, pero de pago para empresas. En el año 1997 se propuso el primer borrador por parte de IETF, y en 1999 empiezan a aparecer las primeras versiones libres que se mantienen hasta el día de hoy como OpenBSD, también llamada OpenSSH y es con la que trabajaremos en este texto.

7.4.2. OpenSSH

OpenSSH, no es simplemente un "comando", se trata de un conjunto de aplicaciones que permiten la administración completa de un equipo remoto de forma segura empleando el protocolo SSH, su código fuente se distribuye libremente con una licencia BSD (Berkeley Software Distribution), esta licencia, a diferencia de GPL (General Public License) permite el uso del código fuente en software no libre. Al tratarse de una aplicación de comunicación abierta entre ordenadores, su objetivo es que pueda ser transparente a la marca, fabricante o modelo, para ello existe un grupo llamado "OpenSSH Portability Team" cuya misión es mantener su código actualizado para que pueda soportar cualquier plataforma.

El conjunto de aplicaciones de OpenSSH comprende:

- **ssh**: reemplaza a telnet y rlogin
- **scp**: reemplaza a rcp
- **sftp**: reemplaza a ftp.
- **sshd**: servidor demonio SSH sshd (del lado servidor)
- **ssh-keyge:** herramienta para inspeccionar y generar claves RSA y DSA que son usadas para la autenticación del cliente o usuario.
- **ssh-agent** y **ssh-add**: herramientas para autenticarse de manera más sencilla, manteniendo las claves listas para no tener que volver a introducirlas en cada acceso.
- **ssh-keyscan**: escanea una lista de clientes y recolecta sus claves públicas.

El proyecto OpenSSH tiene su página en Español, si deseas consultarla es:

http://www.openssh.com/es/index.html

7.4.3. Cliente y servidor

La arquitectura de OpenSSH funciona de modo "Cliente – Servidor", es decir lo primero que se debe contar es con un servidor al que permita conectarnos, que en nuestro caso es sencillamente el demonio "**sshd**" con una adecuada configuración, tal cual veremos en la parte de ejercicios. Una vez configurado este servidor, se podrá conectar a él por medio de cualquier cliente SSH, los cuales pueden ser los comandos que mencionamos en el punto anterior a través de consola, o cualquier interfaz gráfica de las que abundan en Internet, de las cuales como todo estudiante de informática conoce, la más popular es "puTTY", pasos que también desarrollaremos en la parte de ejercicios.

7.4.4. Autenticación

Este tema lo considero uno de los más importantes hallazgos en el mundo de la seguridad informática. Se trata de la forma de compartir un secreto entre dos personas en un lugar donde están todos escuchando... Imaginemos que estamos en una habitación veinte personas y en voz alta deseo informarle al que está en el extremo opuesto "el secreto es: EPIBERTO"... pero ¡sin que se entere el resto!... ¿Os imagináis cómo podrías hacerlo?... difícil ¿no?, bueno la verdad es que ¡muy difícil! Así y todo hubo un par de genios que lo lograron y es lo que hoy se conoce como el algoritmo o protocolo Diffie-Hellman (debido a Whitfield Diffie y Martin Hellman) basado en una técnica de criptografía asimétrica (que veremos más adelante). Se trata de un problema matemático de "logaritmos discretos", se presenta como logaritmos discretos, pues se aplican únicamente números primos en las funciones exponenciales (que son la inversa de las logarítmicas), por lo tanto no hay un continuo entre un valor y otro, sino un salto desde un valor primo hasta el próximo primo, estos saltos hacen que se trate, no de una escala continua sino una escala "discreta" de valores.

Una vez que se entiende el planteo es algo muy sencillo, pero sencillo sólo "en un sentido". Es decir, el problema se plantea en que se puede obtener un valor (o resultado) muy sencillamente conociendo algunos parámetros, pero si intentara, a partir del resultado obtener los parámetros, debería realizar millones y millones de operaciones, lo que lo transforma en "computacionalmente imposible" o que tardaría un tiempo que no se justifica para resolverlo.

Este algoritmo, si deseas profundizar está tratado con máximo detalle más adelante en el capítulo de criptografía, por ahora lo que nos interesa es que comprendas el problema de "compartir un secreto" en una red pública, pues esto justamente es lo que logra hacer SSH en cada sesión.

A diferencia de FTP o Telnet (que os recordamos que usuario y password viajan en texto plano), SSH una vez que se establece el triple Handshake TCP desde el cliente al servidor, la primer trama que enviará será un conjunto de valores (correspondientes al algoritmo Diffie-Hellman), acompañado de una serie de "versiones y algoritmos" que este cliente puede soportar. Con este conjunto de valores el Servidor calculará su "**Clave de sesión**", y le responderá en una segunda trama, con otro conjunto de valores (producidos en su cálculo matemático) y también los algoritmos que este acepta. Al recibir esta segunda trama, con esos valores, el cliente ya puede resolver la "clave de sesión" que será la misma que resolvió el servidor, y esta clave (o secreto compartido), será con la que comenzará a cifrar TODO a partir de la siguiente trama, donde

recién allí empezará a enviar su nombre de usuario, contraseña y toda la información hasta cerrar la sesión SSH.

Por lo tanto, sobre lo que debemos prestar atención, es que únicamente viajaron dos tramas en texto plano, en las mismas no ha viajado ningún dato privado, y si fueran escuchadas por cualquier otro ordenador, no le podrían haber servido de nada. En la parte de ejercicios podrás verlo de forma práctica, y luego en el capítulo de criptografía se desarrollarán todos los conceptos teóricos de esta técnica.

7.4.5. Túneles SSH

Al establecerse una conexión SSH queda conformado un "túnel" entre ambos equipos, este concepto lo trataremos más adelante en detalle, pero por ahora nos basta con la sencilla analogía a un túnel de ferrocarril o de autovía, se trata de una camino que posee una única entrada y una única salida, y que desde afuera no se ve su interior, ni se puede entrar o salir por otro acceso que no sean los mencionados. Un túnel en terminología informática es lo mismo, pero se implementa mediante acciones criptográficas para que sólo entre al mismo quien tenga permiso, y para que todo aquel que pueda capturar su tráfico no tengo forma de interpretarlo, modificarlo, ni desviarlo de su origen, ni destino. Cuando a través de una red se crea un túnel queda conformada una especie de red particular entre ambos, esta idea es lo que en muchos textos verás como VPN (Virtual Private Network o Red Privada Virtual), pues justamente cumple esta idea, al final de la teoría se trata en detalle este tema.

7.4.6. sftp y scp.

El término "scp" en realidad puede interpretarse de dos formas diferentes, como el "programa scp" o como el "protocolo scp", el protocolo es una metodología para transferencia de múltiples archivos sobre TCP confiando su seguridad y autenticación en SSH (al igual que "sftp"), y el "programa scp" no es más que el script "cliente" que ejecuta los pasos de este protocolo, en realidad esta diferenciación no es relevante, pero queríamos mencionarla pues tal vez consultando otra bibliografía te pueda surgir esta duda. Lo que si merece la pena destacar es la diferencia entre scp y sftp, pues verás que algunas implementaciones cliente de SSH proveen un programa que llaman "**scp2**" que en realidad no es más que un enlace simbólico a "sftp". El programa "scp" sencillamente permite "copiar" archivos sin mayores detalles o potencialidades, en cambio "sftp" permite más operaciones que son típicas de una conexión por FTP como mover y borrar ficheros, gestionar permisos, listar directorios, etc...

Las RFC que actualmente regulan SSH son: **RFC-4250, RFC-4251, RFC-4252, RFC-4253, RFC-4254, RFC-4255, RFC-4256, RFC-4335, RFC-4344, RFC-4345, RFC-4419, RFC-4432, RFC-4462, RFC-4716** y **RFC-4819.**

7.5. SMTP (Simple Mail Transfer Protocol) (RFC: 821, 822, 1869).

7.5.1. Funcionamiento.

Es el método definido por Internet para transferencia de correo electrónico. Emplea el puerto TCP 25. Trabaja por medio del empleo de colas o spooler donde va almacenando los mensajes recibidos en los servidores hasta que un usuario se conecte y transfiera su correspondencia, si esto no sucede en un determinado tiempo (Programable), los mensajes son descartados o devueltos a su origen. Debe quedar perfectamente claro que su operatoria no es en tiempo real, sino que dependerá de la voluntad de sus corresponsales. Una característica particular es que todo el texto se transfiere en caracteres ASCII de 7 bit. Su conexión se produce por medio de tramas de comando y respuesta que incluyen instrucciones como mail, RCPT, OK, Texto, etc.

La **RFC-821** especifica el protocolo empleado para la transferencia de correo, y la **RFC-822** describe la sintaxis que deben seguir las cabeceras y su correspondiente interpretación, otra RFC que es conveniente tener en cuenta es la **974** que es la que define el estándar a seguir para el encaminamiento de correo a través de DNS.

7.5.2. Texto plano y extensiones:

Como se mencionó, el protocolo SMTP trabaja con texto plano (ASCII de 7 bit), lo cual en la actualidad no es suficiente para las aplicaciones que requieren imágenes, caracteres especiales, ficheros ejecutables, etc. Para este propósito es que se diseñaron las **RFC 1521** y **1522** que definen **MIME** (Multipropose Internet Mail Extension), el cual transforma cadenas de 8 bit en grupos de siete que son los que viajarán por el canal de comunicaciones, y realizará el proceso inverso del lado receptor.

7.5.3. Mensajes (cabecera y contenido).

Cada mensaje tiene:

- Una cabecera (o sobre) con estructura **RFC-822**. La cabecera termina con una línea nula (una línea con sólo la secuencia <CRLF>).
- Contenido: Todo lo que hay tras la línea nula es el cuerpo del mensaje.

La cabecera es una lista de líneas de la forma:

field-name: field-value

Algunos campos habituales son:

To: Receptores primarios del mensaje.

Cc: Receptores Secundario("carbon-copy") del mensaje.

From: Identidad del emisor.

reply-to: El buzón al que se han de enviar las repuestas. Este campo lo añade el emisor.

return-path: Dirección y ruta hasta el emisor. Lo añade el sistema de transporte final que entrega el correo.

Subject: Resumen del mensaje. Suele proporcionarlo el usuario.

7.5.4. Comandos y códigos.

Todos los comandos, réplicas o datos intercambiados son líneas de texto, delimitadas por un <CRLF>. Todas las réplicas tienen un código numérico el comienzo de la línea. La secuencia de envío y recepción de mensajes es la siguiente:

1) El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje *"220 Service ready"* o *"421 Service not available"* cuando el destinatario es temporalmente incapaz de responder.

2) Se envía un HELO (abreviatura de "hello"), con el que el receptor se identificará devolviendo su nombre de dominio. El SMTP emisor puede usarlo para verificar si contactó con el SMTP de destino correcto.

   Si el emisor SMTP soporta las extensiones de SMTP definidas inicialmente por la **RFC-1651** que luego quedó obsoleta y reemplazada por la **RFC-1869**, y ahora por la **RFC-2821**, puede sustituir el comando HELO por EHLO. Un receptor SMTP que no soporte las extensiones responderá con un mensaje *"500 Syntax error, command unrecognized"*. El emisor SMTP debería intentarlo de nuevo con HELO, o si no puede retransmitir el mensaje sin extensiones, enviar un mensaje QUIT.

   Si un receptor SMTP soporta las extensiones de servicio, responde con un mensaje multi-línea *250 OK* que incluye una lista de las extensiones de servicio que soporta.

3) El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Nótese que una ruta puede ser más que el par *buzón@nombre de dominio del host*. Además, puede contener una lista de los hosts de encaminamiento. Si se acepta, el receptor replica con un *"250 OK"*.

4) El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje(puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO:<*forward-path*>. Cada uno de ellos recibirá una respuesta *"250 OK"* si el servidor conoce el destino, o un *"550 No such user here"* si no.

5) Cuando se envían todos los comandos rcpt, el emisor envía un comando *DATA* para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con

"354 Start mail input, end with <CRLF>.<CRLF>". Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.

6) El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con *"250 OK"* o el mensaje de error apropiado si cualquier cosa fue mal.

7) Ahora hay varias acciones posibles:

- El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con *"221 Service closing transmission channel"*.
- El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes (si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papelees y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 de arriba.
- El emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 3 para enviar un nuevo MAIL.

Un aspecto que se comentó en el párrafo anterior es la opción de ESMTP, la cual "Extiende" las opciones iniciales de la RFC-821, lo que acabamos de ver estaba relacionado al comando EHLO (ESMTP) o (HELO) SMTP, es decir en este primer intercambio de datos, tanto el cliente como el servidor acuerdan qué protocolo soportan. Si ambos soportan ESMTP (que en la actualidad deberían ser todos según la RFC que se menciona al final del párrafo) entonces entra en juego la RFC-1869, la cual también ha quedado obsoleta por la RFC-2821, que es la acutal (se pueden citar algunas más pero están en estado experimental), esta cierra con varias mejoras:

- Ampliación del campo de datos a más de 512 caracteres.
- Diferentes o nuevos códigos de errores (ampliación de los mismos).
- Parámetros adicionales para los comandos MAIL y RCPT.
- Reemplazos (opcionales) para algunos comandos.

Una facilidad que ofrece SMTP es la unificación de mensajes con destino múltiple, los cuales son grabados como un sólo mensaje en los servidores, los que se encargan de distribuirlos a los "n" corresponsales. Se detallan a continuación los comandos y respuestas:

Comando	Descripción
DATA	Begins message composition.
EXPN <string>	Returns names on the specified mail list.
HELO <domain>	Returns identity of mail server.
HELP <command>	Returns information on the specified command.
MAIL FROM <host>	Initiates a mail session from host.
NOOP	Causes no action, except acknowledgement from server.
QUIT	Terminates the mail session.
RCPT TO <user>	Designates who receives mail.
RSET	Resets mail connection.

SAML FROM <host>	Sends mail to user terminal and mailbox.
SEND FROM <host>	Sends mail to user terminal.
SOML FROM <host>	Sends mail to user terminal or mailbox.
TURN	Switches role of receiver and sender.
VRFY <user>	Verifies the identity of a user.

Código de respuesta	Descripción de la respuesta
211	(Response to system status or help request).
214	(Response to help request).
220	Mail service ready.
221	Mail service closing connection.
250	Mail transfer completed.
251	User not local, forward to <path>.
354	Start mail message, end with <CRLF><CRLF>.
421	Mail service unavailable.
450	Mailbox unavailable.
451	Local error in processing command.
452	Insufficient system storage.
500	Unknown command.
501	Bad parameter.
502	Command not implemented.
503	Bad command sequence.
504	Parameter not implemented.
550	Mailbox not found.
551	User not local, try <path>.
552	Storage allocation exceeded.
553	Mailbox name not allowed.
554	Mail transaction failed.

7.5.5. Pasarelas SMTP.

Una pasarela SMTP es un host con dos conexiones a redes distintas. Las pasarelas SMTP se pueden implementar de forma que conecten distintos tipos de redes. Se puede prohibir el acceso a la pasarela a determinados nodos de la red, empleando la sentencia de configuración RESTRICT. Alternativamente, la seguridad se puede implementar con un fichero de autorización de accesos, que es una tabla en la que se especifican de quién y a quién se puede enviar correo por la pasarela.

7.5.6. Terminología.

Algo de terminología referida al correo electrónico:

- Agente de usuario (UA, user agent): programa que se usa como interfaz de usuario para el correo electrónico (leer, componer, enviar, gestionar, etc.)
- Agente de transferencia de mensajes (MTA, message transfer agent): se encarga del encaminamiento y almacenamiento de los mensajes de correo hasta su destino final.
- Protocolo de acceso al correo electrónico: lo usa un UA para acceder a un MTA, y recoger el correo para un usuario. Ejemplo: POP, IMAP.
- Protocolo de envío de correo electrónico: lo usa un MTA para enviar correo a otro MTA (también puede usarlo un UA para enviarlo a un MTA). Ejemplo: SMTP.

7.6. POP (Post Office Protocol) (RFC: 1082, 1725, 1734, 1939).

Este protocolo permite a un usuario conectarse a un sistema y entregar su correo usando su nombre de usuario y contraseña (muy usado en UNIX) a través del puerto TCP 110.

7.6.1. Características.

La metodología a seguir para descargar correo es la explicada en SMTP, lo cual implica que cuando un servidor recibe un mail, establece la sesión SMTP con este destino y entrega su mensaje, esto exigiría que el destino final se encuentre siempre encendido y disponga de los recursos necesarios para desempeñar la tarea de cliente y Servidor SMTP. Ninguna de las dos características son exigibles a un ordenador personal (Recursos y no apagado). Un método intermedio es descargar la función de servidor SMTP de la estación de trabajo del usuario final, pero no la función de cliente. Es decir, el usuario envía correo directamente desde la estación, pero tiene un buzón en un servidor. El usuario debe conectar con el servidor para recoger su correo.

El POP describe cómo un programa que se ejecuta en una estación de trabajo final puede recibir correo almacenado en sistema servidor de correo. POP usa el término "maildrop" para referirse a un buzón gestionado por un servidor POP.

7.6.2. Modos.

Existe una gran variedad de clientes de correo que emplean POP cuya última versión es la 3 (**RFC-1725**). Cuando estos clientes emplean POP3, tienen la opción de dejar sus mensajes a un servidor y verlos remotamente (modo *en línea*), o transferir sus mensajes a un sistema local y consultarlos (modo *fuera de línea*). Cada uno tiene sus ventajas y desventajas, en el modo *en línea*, independientemente de la ubicación física en la que se encuentre el usuario podrá consultar su mail pues este se encuentra almacenado en un servidor, se debe tener en cuenta que cada vez que se conecte al servidor le serán transferido la totalidad de los mensajes (si la conexión es lenta, puede demorar mucho); desde el punto de vista del Administrador, permite centralizar los backup de toda la información almacenada, pero tiene la desventaja que de no controlarse puede llenar fácilmente un disco rígido (inclusive este es un tipo de ataque de negación de servicio). El modo fuera de línea permite organizar en carpetas locales la información histórica acorde a la preferencia del cliente, y este al conectarse al servidor solamente bajará los mail nuevos, reduciendo el tiempo de conexión.

POP - 3 no soporta el uso de carpetas globales o compartidas, como tampoco el uso de listas globales de direcciones, por lo tanto no existe forma de ver la totalidad de las cuentas de una organización automáticamente (Lo debe organizar manualmente el Administrador).

7.6.3. MIME (Multimedoa Internet Mail Extension).

Una mejora que aparece a POP son las extensiones MIME, ya mencionadas (Multimedia Internet Mail Extension), las cuales están estandarizadas por las **RFC-2045** a **2049** y su tarea principal es extender el contenido de los mensajes de correo para poder adjuntar datos de tipo genéricos. Define 5 tipos de cabeceras:

- MIME-version: La actual es 1.0
- Content-Description: Una descripción en texto plano del objeto del cuerpo, suele ser de utilidad cuando el objeto es no legible.
- Content-Id: Un valor unívoco especificando el contenido de esta parte del mensaje.
- Content-Transfer-Encoding: Indica cómo codificar y decodificar los datos.
- Content-Type: Indica con qué aplicación se tratarán los datos

También propone 8 tipos: Text, Image, Audio, Video, Application, Message, Model y Multipart. y varios subtipos: Text: html, plain o richtext; Image: gif, jpeg; ...

7.7. IMAP 4 (Internet Message Access Protocol Versión 4) (RFC: 1203, 1730 a 1733, 3501).

7.7.1. Historia.

El protocolo IMAP tiene sus orígenes en el año 1986, pero la versión actual que es la que nos interesa está estandarizada por la **RFC-3501** (revisión 1) en marzo del 2003. Fue diseñado como una versión moderna de POP3, fundamentalmente basado en una mayor interacción con el servidor SMTP para optimizar recursos locales y de red.

7.7.2. Mejoras que ofrece.

Posee las mismas características que POP3 y agrega algunas más que permiten escalar el servicio de correo a entornos de grupo.

Aparte de los modos en línea y fuera de línea, IMAP - 4 introduce un tercer modo que se llama *Desconexión*. En POP - 3, el cliente al conectarse y autenticarse automáticamente comenzaba a recibir la totalidad de los mail que se encontraban en el Servidor; en IMAP - 4 cuando un cliente se conecta y autentica en un servidor, este consulta sus *"banderas de estado"* para todos los mensajes existentes. Estas banderas permiten identificar a cada mensaje como: *Leído, borrado o respondido*, por lo tanto, puede ser configurado para bajar únicamente los marcados como *no leídos*, reduciendo sensiblemente el tiempo de conexión. Este modo facilita también cualquier anomalía que puede surgir durante la conexión pues el servidor IMAP - 4 entrega al **cliente sólo una copia de sus correos**, y mantiene el original hasta la sincronización completa de su caché, por lo tanto si se pierde una conexión durante una transferencia, al producirse la próxima conexión como primer medida se verá donde se abortó la previa par no reenviar todo, y en segundo lugar no se perderá ningún mensaje hasta que se sincronice el cliente y el servidor

Introduce también el concepto de *Vista Previa*, con lo cual el cliente puede revisar los encabezados de todos los mensajes y sobre estos decidir cuáles leer o borrar antes de bajarlos pues por ejemplo en conexiones dial-up es sumamente desagradable perder gran cantidad de tiempo en la recepción de avisos publicitarios forzados a ser recibidos por "no se sabe quien"

Introduce también el concepto de carpetas compartidas las cuales pueden ser consultadas por grupos de usuarios, y también el de pizarrones de noticias (semejante al protocolo NNTP: Network News Transfer Protocol), en los cuales los usuarios pueden "pinchar" sus carteles de novedades.

7.7.3. Vulnerabilidades del correo electrónico.

Las dos grandes vulnerabilidades que sufre el correo electrónico son referidas a su privacidad y su seguridad, dentro de ellas existen debilidades concretas que se tratan a continuación:

La privacidad es fácilmente vulnerable pues el correo viaja como texto plano, es decir, que si no se emplea algún algoritmo criptográfico, cualquiera puede tener acceso al mismo. En este tema, la mejor analogía es la del correo postal, en el cual a nadie se le ocurre enviar una carta sin el sobre.

La seguridad es atacada con dos técnicas puntuales: las bombas de correo (varias copias de un mismo mail a un solo destino) y el Spam (Correo no autorizado).

Las herramientas con que se cuenta para defenderse de estas vulnerabilidades son:

- S/MIME: Desarrollado por RSA el cual es una especificación para obtener autenticación por medio de firmas digitales. Se lo considera uno de los más seguros
- PGP: Pretty Good Privacy, el cual es un producto completo desarrollado por Phillip Zimmerman que ofrece que dentro de sus muchas funciones ofrece también autenticación, no repudio y criptografía siendo soportado por la mayoría de los clientes de correo.
- PEM: Privacy Enhanced Mail, el cual es una norma para permitir la transferencia de correo seguro. Con cualidades similares a PGP, siendo el estándar más reciente de los tres.

7.8. SNMP (Single Network Monitor Protocol).

Este es el protocolo que habilita las funciones que permiten administrar redes no uniformes. Esta regulado por la **RFC 1155, 1156** y **1157**, y básicamente separa dos grupos: Administradores y Agentes. Los Administradores (**NMS**: Network Managment Station) son los responsables de la administración del dominio ejecutando un determinado Software de monitorización. Los agentes tienen a su vez un Software residente que responde a las solicitudes del administrador con la información guardada en sus bases de datos locales (MIB: Managment Information Base). Estas consultas en realidad pueden ejecutarse por dos métodos:

- **Poll** (Sondeo): La estación administradora sondea uno por uno a los agentes cada un determinado período de tiempo, y estos van informando si apareciera alguna novedad en su MIB desde el último sondeo.
- **Interrupción**: Los Agentes al aparecer alguna novedad en su MIB, envían un mensaje interrumpiendo los procesos del Administrador para notificar sus cambios.

Como puede deducirse cada uno de ellos tiene sus ventajas y desventajas; si una novedad apareciera inmediatamente después que un sondeo fue realizado a un agente, el Administrador tomaría conocimiento de este suceso recién en el próximo sondeo, lo cual por ejemplo en una red de Terapia Intensiva de un Hospital no sería muy saludable. Por el contrario, si se produjera alguna anomalía en el canal de comunicaciones en un sistema por interrupción, el Administrador nunca volvería a detectar novedades en un Agente que se encuentre sobre ese vínculo. Estos son algunos ejemplos, pero en virtud de la cantidad de posibilidades que existen es que se suelen implementar estrategias mixtas de monitoreo de red, que permitan superar estas contingencias.

Otro tema de especial interés en SNMP es la relación costo / beneficio de mantener la Administración absoluta de la red hasta los últimos recursos, lo cual genera un gran volumen de tráfico en la misma. Se suelen establecer límites sobre el nivel de importancia de los agentes a monitorear para reducir la carga que impone este protocolo.

7.8.1. Formato del encabezado:

Versión	Comunidad	PDU

- Versión: Indica el número de versión, los valores admitidos son 1, 2 y 3.
- Comunidad: Este nombre indica el grupo al cual pertenece el mensaje y es empleado para la autenticar al administrador antes que pueda ingresar al agente.
- PDU (Protocol Data Unit): Existen cinco tipos de PDU: GetRequest, GetNextRequest, GetResponse, SetRequest y Trap.

La PDU tiene a su vez un formato que es el siguiente:

PDU Type	Request ID	Error Status	Error Index	Objeto 1 Valor 1	Objeto 2 Valor 2	…. ….

- PDU type: Especifica el tipo de PDU, sus valores son:

  0 GetRequest.

  1 GetNextRequest.

  2 GetResponse.

  3 SetRequest.

- Request ID: Valor entero que controla la correspondencia entre agente y administrador.
- Error status: valor entero que indica operación normal o cinco tipos de error:

  0 noError.

  1 tooBig: El tamaño de la GetResponse PDU requerida, excede lo permitido.

  2 noSuchName: El nombre del objeto solicitado no tiene correspondencia con los nombres disponibles en la MIB.

  3 badValue: La SetRequest contiene un tipo inconsistente, longitud o valor para la variable.

  4 readOnly: No definido en la **RFC-1157**.

  5 genErr: Otros errores, los cuales no están explícitamente definidos.

- Error index: Identifica la entrada en la lista que ocasionó el error.
- Object/value: Define el objeto con su valor correspondiente.

Existe también otro formato de PDU, que es el de Trap PDU, el cual tiene los siguientes campos:

PDU Type	Enterprise	Agent Address	Gen Trap	Spec Trap	Time Stamp	Objeto 1 Valor 1	Objeto 2 Valor 2	…. ….

- PDU Type: Valor 4.
- Enterprise: Identifica al administrador de la "empresa" que definió la trap.
- Agent Address: Dirección IP del agente.
- Generic Trap Type: Campo que describe el evento que está siendo reportado, los siguientes siete valores están definifdos:

  0 coldStart: La entidad ha sido reinicializada, indicando que la configuración pudo ser alterada.

  1 warmStart: La entidad ha sido reinicializada, pero la configuración no fue alterada.

  2 linkDown: El enlace ha fallado.

  3 linkUp: El enlace ha conectado.

  4 authenticationFailure: El agente ha recibido una autenticación SNMP indebida desde el administrador.

  5 egpNeighborLoss: Un EGP vecino esta caído.

  6 enterpriseSpecific: Un trap no genérico ha ocurrido, el cual es identificado por los campos Specific Trap Type y Enterprise.
- Specific Trap Type: Empleado para identificar un Trap no genérico.
- Timestamp: Representa el tiempo transcurrido entre la última reinicialización y la generación del presente trap.
- Combinación de la variable con su valor.

7.8.2. SNMP Versión 3.

En el mes de enero del año 1998 IETF propone un conjunto de **RFC** desde la **2271** a la **2275,** las cuales definen un conjunto de medidas para implementar las tres grandes falencias que poseía el protocolo SNMP, estas son:

- Autenticación.
- Seguridad.

- Control de acceso.

Estos nuevos estándares propuestos son los que definen la nueva versión de este protocolo denominada versión 3. El propósito es definir una arquitectura modular que de flexibilidad hacia futuras expansiones.

Luego de un tiempo, en el mes de abril de 1999 aparecen ya como borrador estándar los mismos conceptos con algunas mejoras, dejando obsoletos los anteriores. Estas son las recomendaciones **2571** a la **2575**, las cuales sientan definitivamente el funcionamiento de SNMPv3. Estas son:

2571 An Architecture for Describing SNMP Management Frameworks. B. Wijnen, D. Harrington, R. Presuhn. April 1999. (Format: TXT=139260 bytes) (Obsoletes RFC2271) (Status: DRAFT STANDARD)

2572 Message Processing and Dispatching for the Simple Network Management Protocol (SNMP). J. Case, D. Harrington, R. Presuhn, B. Wijnen. April 1999. (Format: TXT=96035 bytes) (Obsoletes RFC2272) (Status: DRAFT STANDARD)

2573 SNMP Applications. D. Levi, P. Meyer, B. Stewart. April 1999. (Format: TXT=150427 bytes) (Obsoletes RFC2273) (Status: DRAFT STANDARD)

2574 User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen. April 1999. (Format: TXT=190755 bytes) (Obsoletes RFC2274) (Status: DRAFT STANDARD)

2575 View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). B. Wijnen, R. Presuhn, K. McCloghrie. April 1999. (Format: TXT=79642 bytes) (Obsoletes RFC2275) (Status: DRAFT STANDARD)

En estas nuevas RFC aparecen una serie de conceptos que son los que se definen a continuación.

Entidad:

El concepto de entidad es una conjunto de módulos que interactúan entre sí, cada entidad implementa una porción de SNMP y puede actuar como los tradicionales nodo AGENTE, nodo GESTOR, o combinación de ambos.

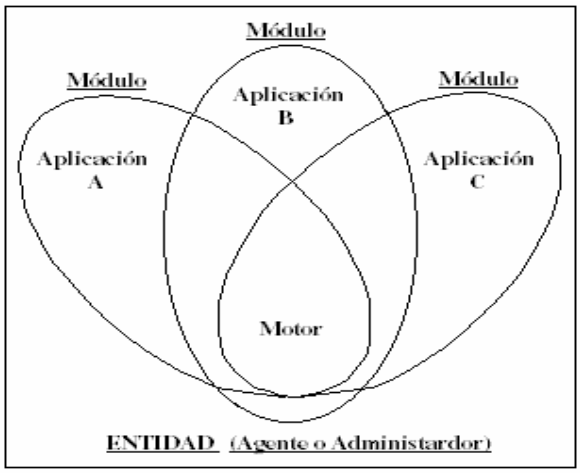
Cada entidad incluye un MOTOR SNMP, siendo éste el encargado de implementar las funciones de:

- Envío de mensajes.
- Recepción de mensajes.
- Autenticación.
- Cifrado y descifrado de mensajes.
- Control de acceso a los objetos administrados.

Estas funciones son provistas como servicios a una o más aplicaciones.

El conjunto de motor y aplicaciones son definidas como los módulos de esta entidad.

Gestor tradicional SNMP:



Un Gestor tradicional SNMP interactúa con los agentes SNMP a través del envío de comandos (get,get next y set) y recepción de respuestas. Este incluye 3 categorías de Aplicaciones:

- Aplicaciones Generadoras de Comandos: Monitorizan y controlan la administración de datos de un agente remoto.
- Aplicación Generadora de Notificaciones: Inicia mensajes asincrónicos.
- Aplicación Receptora de Notificaciones: Procesa mensajes entrantes asincrónicos.

Estas tres aplicaciones hacen uso de los servicios del motor SNMP.

Este motor debe contener:

1) Un Despachador: Encargado de administrar el tráfico. Para mensajes salientes, recibe las PDU (Unidad de datos de Protocolo) de las aplicaciones, determina el tipo de procesamiento requerido (Ej: SNMPv1, SNMPv2 o SNMPv3) y entrega estos datos al módulo de procesamiento de mensajes correspondiente. Para mensajes entrantes, acepta mensajes del nivel de transporte y lo deriva al módulo de procesamiento de mensajes correspondiente. Consecuentemente al recibir los mensajes procesados desde el módulo, los entregará hacia la aplicación apropiada o hacia el nivel de transporte según corresponda.
2) Un Subsistema de Procesamiento de Mensajes: Es el responsable del armado y desarmado de la PDU de este nivel. Recibe y entrega los mensajes del despachador. Si es necesario luego de armar la PDU (mensaje saliente) o antes de desarmarla (mensaje entrante), pasaría la misma al Subsistema de Seguridad
3) Un Subsistema de Seguridad: Es quien ejecuta las funciones de autenticación y cifrado. Recibe y entrega los mensajes al Subsistema de Procesamiento de Mensajes. Este

subsistema soporta uno o más modelos distintos de seguridad llamado **User-Based Security Model (USM)** y está definido por la **RFC-2574**.

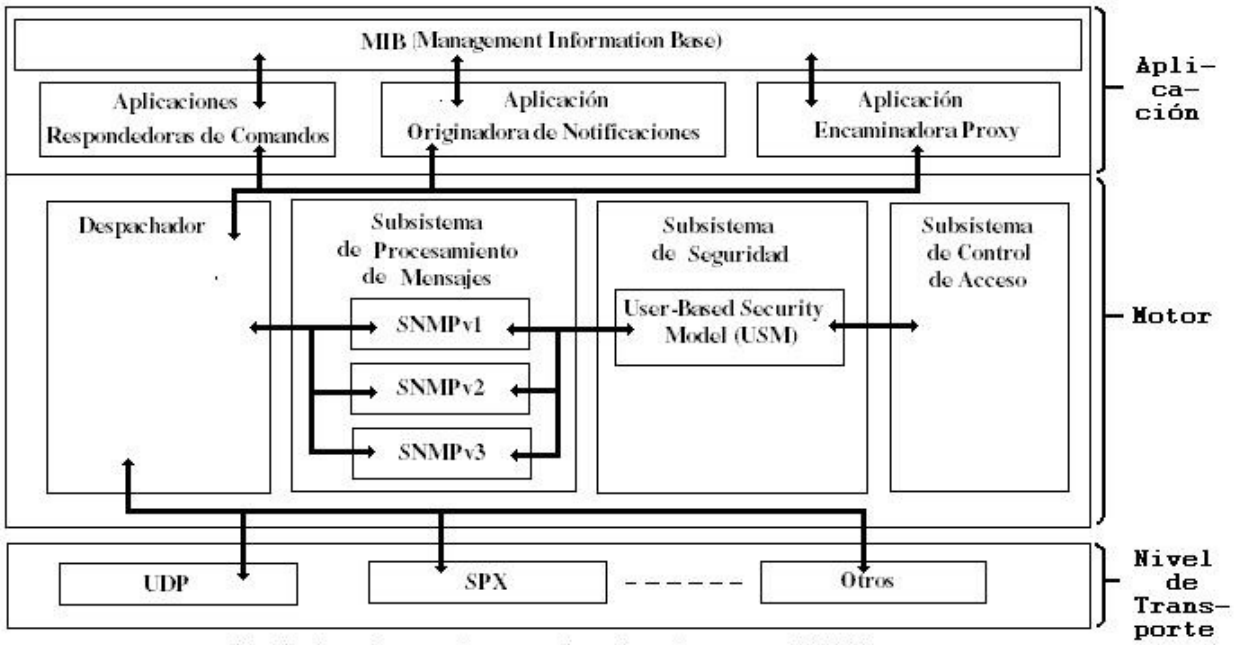
Agente Tradicional SNMP:

El agente contiene también 3 tipos de aplicaciones:

- Aplicaciones Respondedoras de Comandos: Provee acceso a los datos administrados.
- Aplicación Generadora de Notificaciones: Inicia mensajes asincrónicos.
- Aplicación Encaminadora Proxy: Encamina mensajes entre entidades.

El Agente tiene los mismos componentes que el Administrador e incluye uno más denominado:

- Subsistema de Control de Acceso: Es el encargado de proveer servicios de autorización para controlar el acceso a las MIBs.Un Agente soporta uno o más modelos de Control de Accesos diferentes llamado **View-Based Access Control Model (VACM)** y se encuentra definido por la **RFC-2575**.



Modelo de referencia de Agente SNMP

Subsistema de procesamiento de mensajes:

Este modelo recibe PDU desde el despachador, tanto salientes como entrantes, las encapsula o desencapsula y acorde a la existencia de mecanismos de seguridad la entrega o no al USM para el tratamiento de los parámetros de seguridad (agregado, cifrado o decifrado) y finalmente las devuelve al despachador para que este las entregue al nivel correspondiente.

Este modelo opera sobre los cinco primeros campos del encabezado SNMP, el cual se detalla a continuación:

		Msg Version	Modelo de

		Msg ID	
		Msg Max Size	
		Msg FLAGS	
		Msg Security Model	
		Msg Autoritative Engine ID	Modelo de seguridad de usuario (USM)
		Msg Autoritative Engine Boot	
		Msg Autoritative Engine Time	
		Msg User Name	
		Msg Autentication Parameters	
		Msg Privacy Parameters	
	Cifrado	Context Engine ID	Espacio de PDU
		Context Name	
		PDU (Aplication)	

- *Msg Version:* Corresponde el Nro 3.
- *Msg ID:* Identificador entre las dos entidades para coordinar solicitudes y respuestas
- *Msg Max Size:* Expresa el tamaño máximo en octetos que soporta el emisor.
- *Msg FLAGS:* Están definidos los tres primeros bit y significan lo siguiente:

  Bit 1 (bit de reporte): Si está en 1 entonces el receptor debe especificar bajo qué condiciones este puede causar un reporte. También es empleado en toda solicitud Get o Set.

  Bit 2 (Priv Flag): Indica que se emplea criptografía.

  Bit 3 (Aut Flag): Indica que se emplea autenticación.

- *Msg Security Model: Indica qué modelo de seguridad se emplea (1 = SNMPv1, 2 = SNMPv2, 3 = SNMPv3).*

Subsistema de seguridad:

El subsistema de seguridad ejecuta funciones de autenticación y cifrado, para las mismas define uno o más distintos *Modelos de seguridad de Usuario (USM)*. Específicamente la **RFC-2574** establece que este modelo protege contra lo siguiente:

- Modificación de Información.
- Falsificación de entidad.
- Modificación de mensaje.
- Difusión de información.

También aclara que no protege contra ataques de negación de servicio ni análisis de tráfico.

Este modelo se emplea para proveer autenticación y privacidad, para esto define dos claves, una clave privada (PrivKey) y otra clave de autenticación (AutKey). El valor de estas claves no es accesible vía SNMP y se emplean de la siguiente forma:

Autenticación:

Se definen dos alternativas para esta tarea, HMAC-MD5-96 y HMAC-SHA-96.

La mecánica de esta función es que a través de una cadena de bit de entrada de cualquier longitud finita, generará un único resumen de salida de longitud fija. Que en el caso de esta norma es de 20 Byte para SHA o 16 Byte para MD5.

Esta función es llamada "**One Way**" pues no es posible a través del resumen de salida obtener el texto de entrada, también resultará computacionalmente imposible obtener un valor de salida igual a través de otro valor de entrada, como así tampoco desde un valor de salida ya calculado, obtener otro valor de entrada diferente al verdadero.

La aplicación aquí propuesta toma los datos y la clave y produce un resumen:

- Resumen = H (clave, datos).

En cualquiera de los dos casos, se toman como válidos los primeros 96 bit, descartando el resto.

Esta especificación soporta el protocolo HMAC [**RFC-2104**] con la opción SHA1 (Hash Message Autentication-Secure Hash Standard Versión 1) [**RFC-2404**] y MD5 (Messge Digest Verión 5) [**RFC-2403**].

Criptografía:

Para esta actividad USM emplea el algoritmo DES (Data encryptation Standard) [**ANSI X3.106**] en el modo cifrado encadenado de bloques (CBC). La clave privada (PrivKey) antes mencionada de longitud 16 byte es empleada aquí dividiéndola en dos, los primeros 8 Byte, es decir 64 bit son empleados como clave para DES, el cual solo tendrá en cuenta 56, dejando 8 para control de paridad. Los últimos 8 Byte son empleados como Vector de Inicialización (IV) para comenzar con el cifrado en cadena.

Esta técnica CBC, se basa en tomar el primer bloque de texto plano, y realizar una operación XOR con un Vector de inicialización y luego de esta operación recién se pasará al cifrado de ese bloque. En el segundo bloque se realizará nuevamente la operación XOR, pero esta vez será el texto plano de este bloque con el bloque cifrado anteriormente, y luego se cifrará. Esta mecánica se irá realizando en los sucesivos bloques, es decir XOR con el bloque cifrado anterior y luego cifrado.

El descifrado se realiza en forma inversa.

- cifrado = E (clave, texto).
- D (clave, cifrado) = texto.

Campos del encabezado de USM:

Antes de tratar los campos de este modelo se debe tener en cuenta al concepto de autoritativo:

- Caso 1: Cuando un mensaje SNMP contiene datos que esperan una respuesta (Get, GetNext, Get Bulk, Set o Informes), entonces el receptor de ese mensaje es Autoritativo.
- Caso 2: Cuando un mensaje SNMP contiene datos que no imponen respuesta (Trap, Respuestas o Reportes), entonces el emisor de ese mensaje es Autoritativo.

Acorde a la gráfica anterior del encabezado SNMPv3, se puede apreciar que USM emplea los seis campos siguientes al Modelo de Procesamiento de Mensajes. Estos campos se detallan a continuación:

- *Msg Autoritative Engine ID:* Identificador de la entidad Autoritativa.
- *Msg Autoritative Engine Boot:* Este valor es un contador monótono creciente que identifica la cantidad de veces que la entidad autoritativa fue inicializada o reinicializada desde su configuración inicial.
- *Msg Autoritative Engine Time:* Este valor es un entero que describe el tiempo transcurrido en segundos desde el momento en que la entidad autoritativa incrementó el *Msg Autoritative Engine Boot* (es decir el tiempo desde la última vez que inició o reinició). Las entidades autoritativas llevan su tiempo exacto en segundos y las no autoritativas llevarán por cada entidad autoritativa con la que se comuniquen una apreciación del mismo, que servirá para compararlos en el momento oportuno (como se verá a continuación). Este valor son 32 bit, en el caso de no reinicializarse una entidad se irá acumulando y al llegar al valor máximo volverá a cero (En realidad como es un valor de 32 bit, 2^{32} segundos son en el orden de 68 años, por lo tanto el sistema debería ser extremadamente sólido para no detenerse nunca en este lapso)
- *Msg User Name:* Nombre del usuario de este mensaje.
- *Msg Authentication Parameters:* Aquí es donde va el código de autenticación es decir el valor obtenido por HMAC. En el caso de no emplear autenticación es nulo.
- *Msg Privacy Parameters:* El valor aquí expuesto es el que se empleará para obtener el Vector de Inicializaciín (VI) para el algoritmo DES. En el caso de no emplear criptografía es nulo.

La secuencia de pasos a seguir con estos campos para la transmisión de un mensaje en este modelo es:

1) Como primera actividad se criptografían los datos en caso de implementar esta función.
2) Si se realizó el paso a. entonces se coloca en el campo *Msg Privacy Parameters* el valor correspondiente para generar el IV.
3) Si se emplea autenticación, la totalidad del mensaje se ingresa para obtener el resumen HMAC y su resultado es ubicado en el campo *Msg Authentication Parameters.*

En el caso de la recepción sería:

1) Realiza el cálculo de HMAC.

2) Compara el valor calculado con el correspondiente al campo campo *Msg Authentication Parameters*.

3) Si ambos valores son iguales, entonces toma el mensaje como auténtico y no ha sido alterado.

4) Verifica si el mismo está en un tiempo de ventana válido. Esta actividad se realiza de la siguiente forma:

a) Toda entidad no autoritativa guarda tres parámetros en forma local de cada entidad autoritativa con la que se comunica, estos son:

- El valor más reciente de *Msg Autoritative Engine Boot* recibido en la última comunicación.
- El valor de tiempo estimado que debería tener la entidad autoritativa.
- El último valor de tiempo recibido de la entidad autoritativa en el campo *Msg Autoritative Engine Time.*

b) Al recibir un mensaje compara los campos del mensaje recibido con estos parámetros almacenados localmente.

c) Las condiciones para que un mensaje sea considerado no auténtico son:

- Diferencia de *Msg* Autoritative Engine Boot.
- Diferencia en ± 150 segundos entre el valor calculado de Msg Autoritative Engine Time y el recibido en el mensaje.

d) Si un mensaje es considerado no auténtico, una indicación de error es enviada al módulo respectivo.

5) Finalmente si está cifrado, descifra el mismo.

Localización de claves:

Una clave localizada es un secreto compartido entre un usuario y un motor SNMP autoritativo.

El problema del empleo de una sola clave por parte del usuario con todos los agentes es que si se descubriera la misma, sería vulnerable todo el sistema. Si el caso fuera lo contrario es decir que se deseara emplear una clave distinta para cada agente, entonces el usuario debería recordar todas las contraseñas lo cual en la práctica no es viable.

Para dar solución a estos problemas la **RFC-2574** propone este proceso por el cual una clave única de usuario (o pueden ser dos: una para privacidad y otra para autenticación) es convertida a múltiples claves únicas también, una para cada motor SNMP, este proceso es lo que se denomina **Localización de claves**. Las características fundamentales que propone este proceso son:

- Cada agente SNMP tiene su propia clave única para cada usuario autorizado a administrarlo, por lo tanto si la clave de uno de ellos es comprometida, no lo serán las del resto.

- La clave de un usuario es diferente en cada agente SNMP, por lo tanto si se compromete la clave de un agente, no comprometerá al resto ni a la clave del usuario.
- La administración de la red, puede realizarse en forma segura remotamente desde cualquier punto de la red.

Subsistema de Control de Accesos:

Este subsistema se ejecuta en los agentes tradicionales, permitiendo determinar quien está autorizado a acceder a la MIB de los mismos. El único modelo definido para esta actividad se denomina *Modelo de Control de Accesos basado en Vistas* (VACM: View-Based Access Control Model) y está definido en la **RFC-2575**.

VACM tiene dos características fundamentales:

- Determina si un acceso a la MIB local está permitido.
- Posee su propia MIB en la cual se definen las políticas de acceso y habilita la configuración remota.

Se definen cinco elementos que constituyen la VACM:

1) Grupos: Es un conjunto de cero o más duplas {Modelo de Seguridad, Nombre de seguridad} que definen los Objetos que pueden ser administrados por ese Nombre.
2) Nivel de seguridad: Define qué tareas serán permitidas (lectura, escritura o notificación) para cada grupo.
3) Contexto: Es un subconjunto de instancias de objetos en la MIB local. Permite agrupar objetos con distintas políticas de acceso.
4) Vistas de la MIB: Define conjuntos específicos de objetos administrados, los cuales se pueden agrupar en jerarquías de árboles y familias de manera tal que se pueda, por ejemplo, restringir su acceso a determinados grupos.
5) Política de acceso: VACM permite a un motor SNMP ser configurado para asegurar un conjunto de accesos correctos, los cuales dependen de los siguientes factores:
   - Los distintos usuarios pueden tener distintos privilegios de acceso.
   - El nivel de seguridad para una determinada solicitud.
   - El modelo de seguridad empleado para procesar las solicitudes de mensajes.
   - El contexto de la MIB.
   - La instancia al objeto para el cual fue solicitado el acceso.

Listado de RFCs relacionadas a SNMP.

SNMPv1

- RFC 1157 - Simple Network Management Protocol
- RFC 1155 - Structure of Management Information
- RFC 1212 - Concise MIB Definitions

SNMPv2

- RFC 1901 - Community-based SNMPv2
- RFC 3416 - Protocol Operations for SNMPv2
- RFC 3417 - Transport Mappings for SNMP
- RFC 1908 - SNMPv1 and SNMPv2 Coexistence

SNMPv2 Definición de datos

- RFC 2578 - Structure of Management Information
- RFC 2579 - Textual Conventions
- RFC 2580 - Conformance Statements

SNMPv3

- RFC 3411 - Architecture for SNMP Frameworks
- RFC 3412 - Message Processing and Dispatching
- RFC 3413 - SNMPv3 Applications
- RFC 3414 - User-based Security Model
- RFC 3415 - View-based Access Control Model

MIBs

- RFC 1213 - Management Information Base II
- RFC 1573 - Evolution of the Interfaces Group of MIB-II
- RFC 1757 - Remote Network Monitoring MIB
- RFC 2011 - Internet Protocol MIB
- RFC 2012 - Transmission Control Protocol MIB
- RFC 2013 - User Datagram Protocol MIB
- RFC 3418 - MIB for SNMP

Authentication/Privacy

- RFC 1321 - MD5 Message-Digest Algorithm
- RFC 2104 - HMAC: Keyed-Hashing for Message Authentication
- RFC 2786 - Diffie-Helman USM Key

7.9. HTTP (HiperText Transfer Protocol) (RFC 1945 - 2616):

Se trata del protocolo principal que regula todo el sistema de navegación a través de páginas Web.

7.9.1. Conceptos.

Este es el protocolo empleado entre clientes y servidores Web. La diferencia con los demás protocolos de nivel de aplicación es que este establece una sesión por cada información requerida (texto, sonido, gráficos, etc), esta finaliza al completarse la solicitud. Es normal la apertura de varias sesiones para bajar una sola página.

Desde la versión 1.0 en adelante incorpora MIME (Multimedia Internet Mail Extensions) para soportar la negociación de distintos tipos de datos. Hoy ya está vigente la versión 1.2 (**RFC-2774**).

El acceso a este protocolo es por medio del puerto TCP 80 por defecto, pero es común en redes privadas el empleo de otro para incrementar las medidas de seguridad.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. El desarrollo de aplicaciones web necesita frecuentemente mantener estado. Para esto se usan las **cookies** (ver a continuación). Esto le permite a las aplicaciones web instituir la noción de "sesión", y también permite rastrear usuarios ya que las cookies pueden guardarse en el cliente por tiempo indeterminado.

Una **cookie** (literalmente galleta) es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas. En ocasiones también se le llama "huella".

De esta forma, los usos más frecuentes de las cookies son:

- Llevar el control de usuarios: cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas para cada página del servidor. Sin embargo una cookie no identifica a una persona, sino a una combinación de ordenador y navegador.
- Conseguir información sobre los hábitos de navegación del usuario, e intentos de spyware, por parte de agencias de publicidad y otros. Esto puede causar problemas de privacidad y es una de las razones por la que las cookies tienen sus detractores.

Originalmente, sólo podían ser almacenadas por petición de un CGI desde el servidor, pero Netscape dio a su lenguaje Javascript la capacidad de introducirlas directamente desde el cliente, sin necesidad de CGIs. En un principio, debido a errores del navegador, esto dio algunos problemas de seguridad. Estas vulnerabilidades fueron descubiertas por Esteban Rossi. Pueden ser borradas, aceptadas o bloqueadas según desee, para esto sólo debe configurar convenientemente el navegador web.

En realidad, las cookies son sólo datos, no código, luego no pueden borrar ni leer información del ordenador de los usuarios. Sin embargo, las cookies permiten detectar las páginas visitadas por un usuario en un sitio determinado o conjunto de sitios. Esta información puede ser recopilada en un perfil de usuario. Estos perfiles son habitualmente anónimos, es decir, no contienen información personal del usuario (nombre, dirección, etc). De hecho, no pueden contenerla a menos que el propio usuario la haya comunicada a alguno de los sitios visitados. Pero aunque anónimos, estos perfiles han sido objeto de algunas preocupaciones relativas a la privacidad.

7.9.2. Solicitudes y respuestas.

Existen dos tipos de encabezado, el de solicitud y el de respuesta y son los siguientes:

Solicitud

Method	Request URI	HTTP version

- Method: Define el método a ser ejecutado sobre el recurso.
- Request URI (Uniform Resource Identifier): Recurso sobre el que se aplica la solicitud.
- HTTP Version: La versión a ser utilizada.

Respuesta

HTTP version	Status Code	Reason phrase

- HTTP Version: La versión a ser utilizada.
- Status code: Se trata de un entero de 3 dígitos que es el resultado de intentar entender y satisfacer la respuesta.
- Reason phrase: Descripción textual del status code.

Ejemplo de un diálogo HTTP:

Para obtener un recurso con, por ejemplo en este caso pondríamos:

http://www.ejemplo.com/index.html

1) Se abre una conexión al host www.ejemmplo.com, al puerto 80 que es el puerto por defecto para HTTP.

2) Se envía un mensaje en el estilo siguiente:

GET /index.html HTTP/1.1
Host: www.ejemplo.com
User-Agent: nombre-cliente
[Línea en blanco]

La respuesta del servidor está formada por encabezados seguidos del recurso solicitado, en el caso de una página web:

HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html>
<body>
<h1>Página principal</h1>
(Contenido)
.
.
.
</body>
</html>

7.9.3. URL y URI.

(De Wikipedia, la enciclopedia libre)

Una URL común está compuesta por cuatro partes:

- **Protocolo:** También llamado esquema **URL**, especifica que protocolo es utilizado para acceder al documento.
- Nombre del ordenador: Especifica su nombre (usualmente un nombre de dominio o una dirección IP) donde el contenido está alojado.
- **Directorios:** Secuencia de directorios separados por barras ("/") que define la ruta a seguir para llegar al documento.
- **Archivo:** El nombre del archivo donde el recurso se encuentra ubicado.

De esta forma, podemos analizar cualquier URL dada:

http://	www.htmlquick.com	/reference/	uris.html
Protocol	**Domain name**	**Directories**	**File**

URL significa Uniform Resource Locator, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

Desde 1994, en los estándares de Internet, el concepto de URL ha sido incorporado dentro del más general de **URI** (Uniform Resource Identifier - Identificador Uniforme de Recurso), pero el término URL aún se utiliza ampliamente.

El **URL** es la cadena de caracteres con la cual se asigna una dirección única a cada uno de los recursos de información disponibles en Internet. Existe un URL único para cada página de cada uno de los documentos de la World Wide Web.

El **URL** de un recurso de información es su dirección en Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada. Por ello el URL combina el nombre del ordenador que proporciona la información, el directorio donde se encuentra, el nombre del archivo y el protocolo a usar para recuperar los datos.

7.9.4. Esquema URL.

Un URL se clasifica por su esquema, que generalmente indica el protocolo de red que se usa para recuperar, a través de la red, la información del recurso identificado. Un URL comienza con el nombre de su esquema, seguida por dos puntos, seguido por una parte específica del esquema.

Algunos ejemplos de esquemas URL:

- **http** - recursos HTTP
- **https** - HTTP sobre SSL
- **ftp** - File Transfer Protocol
- **mailto** - direcciones E-mail
- **ldap** - búsquedas LDAP Lightweight Directory Access Protocol
- **file** - recursos disponibles en la computadora local, o en una red local
- **news** - grupos de noticias Usenet (newsgroup)
- **gopher** - el protocolo Gopher (ya en desuso)
- **telnet** - el protocolo telnet
- **data** - el esquema para insertar pequeños trozos de contenido en los documentos Data: URL

Algunos de los esquemas URL, como los populares "mailto", "http", "ftp" y "file", junto a los de sintaxis general URL, se detallaron por primera vez en 1994, en la **RFC-1630**, sustituido un año después por los más específicos **RFC-1738** y **RFC-1808**.

Algunos de los esquemas definidos en la primera RFC aun son válidos, mientras que otros son debatidos o han sido refinados por estándares posteriores. Mientras tanto, la definición de la sintaxis general de los URL se ha escindido en dos líneas separadas de especificación de URI: **RFC-2396** (1998) y **RFC-2732** (1999), ambos ya obsoletos pero todavía ampliamente referidos en las definiciones de esquemas URL.

7.9.5. Sintaxis Genérica URL

Todos los URL, independientemente del esquema, deben seguir una sintaxis general. Cada esquema puede determinar sus propios requisitos de sintaxis para su parte específica, pero el URL completo debe seguir la sintaxis general.

Usando un conjunto limitado de caracteres, compatible con el subconjunto imprimible de ASCII, la sintaxis genérica permite a los URL representar la dirección de un recurso, independientemente de la forma original de los componentes de la dirección.

Los esquemas que usan protocolos típicos basados en conexión usan una sintaxis común para "URI genéricos", definida a continuación:

esquema://autoridad/ruta?consulta#fragmento

La **autoridad** consiste usualmente en el nombre o Dirección IP de un servidor, seguido a veces de dos puntos (":") y un número de Puerto TCP. También puede incluir un nombre de usuario y una clave, para autenticarse ante el servidor.

La **ruta** es la especificación de una ubicación en alguna estructura jerárquica, usando una barra diagonal ("/") como delimitador entre componentes.

La **consulta** habitualmente indica parámetros de una consulta dinámica a alguna base de datos o proceso residente en el servidor.

El **fragmento** identifica a una porción de un recurso, habitualmente una ubicación en un documento.

7.9.6. Ejemplo: URL en http.

Los URL empleados por HTTP, el protocolo usado para transmitir páginas web, son el tipo más popular de URL y puede ser usado para mostrarse como ejemplo. La sintaxis de un URL HTTP es:

esquema://anfitrión:puerto/ruta?parámetro=valor#enlace

- **esquema**, en el caso de HTTP, en la mayoría de las veces equivale a http, pero también puede ser https cuando se trata de HTTP sobre una conexión TLS (para hacer más segura la conexión).
- Muchos navegadores, para autenticación en http, permiten el uso de esquema://usuario:contraseña@anfitrión:puerto/.... Este formato ha sido usado como una "hazaña" para hacer difícil el identificar correctamente al servidor involucrado. En consecuencia, el soporte para este formato ha sido dejado de lado por algunos navegadores. La sección 3.2.1 de RFC 3986 recomienda que los navegadores deben mostrar el usuario/contraseña de otra forma que no sea en la barra de direcciones, a causa de los problemas de seguridad mencionados y porque las contraseñas no deben mostrarse nunca como texto plano.
- **anfitrión**, la cual es probablemente la parte que más sobresale de un URL, es en casi todos los casos el nombre de dominio de un servidor, p.ej.: www.wikipedia.org, google.com, etc.
- La porción **:puerto** especifica un número de puerto TCP. Usualmente es omitido (en este caso, su valor por omisión es 80) y probablemente, para el usuario es lo que tiene menor relevancia en todo el URL.
- La porción **ruta** es usada por el servidor (especificado en anfitrión) de cualquier forma en la que su software lo establezca, pero en muchos casos se usa para especificar un nombre de archivo, posiblemente precedido por nombres de directorio. Por ejemplo, en la ruta /wiki/Vaca, wiki sería un (seudo-)directorio y Vaca sería un (seudo-)nombre de archivo.
- La parte mostrada arriba como **?parámetro=valor** se conoce como porción de consulta (o también, porción de búsqueda). Puede omitirse, puede haber una sola pareja parámetro-valor como en el ejemplo, o pueden haber muchas de ellas, lo cual se expresa como ?param=valor&otroParam=valor&.... Las parejas parámetro-valor únicamente son relevantes si el archivo especificado por la ruta no es una página Web simple y estática, sino algún tipo de página automáticamente generada. El software generador usa las parejas parámetro-valor de cualquier forma en que se establezca; en su mayoría transportan información específica a un usuario y un momento en el uso del sitio, como términos concretos de búsqueda, nombres de usuario, etc. (Observe, por ejemplo, de qué forma se comporta el URL en la barra de direcciones de su navegador durante una búsqueda Google: su término de búsqueda es pasado a algún programa sofisticado en google.com como un parámetro, y el programa de Google devuelve una página con los resultados de la búsqueda.)
- La parte **#enlace**, por último, es conocida como identificador de fragmento y se refiere a ciertos lugares significativos dentro de una página; por ejemplo, esta página tiene enlaces internos hacia cada cabecera de sección a la cual se puede dirigir usando el ID de fragmento. Esto es relevante cuando un URL de una página ya cargada en un navegador permite saltar a cierto punto en una página extensa. Un ejemplo sería este enlace, que conduce a esta misma página y al comienzo de esta

sección. (Observe cómo cambia el URL en la barra de dirección de su navegador cuando hace clic en el enlace.)

7.9.7. Referencias URI.

El término referencia URI se refiere a un caso particular de un URL, o una porción de éste, tal como es usada en un documento HTML, por ejemplo, para referirse a un recurso particular. Una referencia URI habitualmente se parece a un URL o a la parte final de un URL. **Las referencias URI introducen dos nuevos conceptos**:

- la distinción entre referencias absolutas y relativas,
- el concepto de un identificador de fragmento.

Un **URL absoluto** es una referencia URI que es parecida a los URL definidos arriba; empieza por un esquema seguido de dos puntos (":") y de una parte específica del esquema. Un **URL relativo** es una referencia URI que comprende sólo la parte específica del esquema de un URL, o de algún componente de seguimiento de aquella parte. El esquema y componentes principales se infieren del contexto en el cual aparece la referencia URL: el URI base (o URL base) del documento que contiene la referencia.

Una referencia URI también puede estar seguida de un carácter de numeral ("#") y un puntero dentro del recurso referenciado por el URI en conjunto. Esto no hace parte del URI como tal, sino que es pensado para que el "agente de usuario" (el navegador) lo interprete después que una representación del recurso ha sido recuperada. Por tanto, no se supone que sean enviadas al servidor en forma de solicitudes HTTP.

Ejemplos de URL absolutos:

- http://es.wikipedia.org/w/wiki.phtml?title=URL&action=history
- http://es.wikipedia.org/wiki/URL#Esquemas_en_URL

Ejemplos de URL relativos:

- //en.wikipedia.org/wiki/Uniform_Resource_Locator
- /wiki/URL
- URL#Referencias_URI

Diferenciación entre mayúsculas/minúsculas

De acuerdo al estándar actual, en los componentes esquema y anfitrión no se diferencian mayúsculas y minúsculas, y cuando se normalizan durante el procesamiento, deben estar en minúsculas. Se debe asumir que en otros componentes sí hay diferenciación. Sin embargo, en la práctica, en otros componentes aparte de los de protocolo y anfitrión, esta diferenciación es dependiente del servidor Web y del sistema operativo del sistema que albergue al servidor.

7.9.8. URL en el uso diario.

Un HTTP URL combina en una dirección simple los cuatro elementos básicos de información necesarios para recuperar un recurso desde cualquier parte en la Internet:

- El **protocolo** que se usa para comunicar,
- El **anfitrión** (servidor) con el que se comunica,
- El **puerto** de red en el servidor para conectarse,
- La **ruta al recurso** en el servidor (por ejemplo, su nombre de archivo).

Un URL típico puede lucir como:

http://es.wikipedia.org:80/wiki/Special:Search?search=tren&go=Go

Donde:

- http es el protocolo,
- es.wikipedia.org es el anfitrión,
- 80 es el número de puerto de red en el servidor (siendo 80 valor por omisión),
- /wiki/Special:Search es la ruta de recurso,
- ?search=tren&go=Go es la cadena de búsqueda; esta parte es opcional.

Muchos navegadores Web no requieren que el usuario ingrese "http://" para dirigirse a una página Web, puesto que HTTP es el protocolo más común que se usa en navegadores Web. Igualmente, dado que 80 es el puerto por omisión para HTTP, usualmente no se especifica. Normalmente uno sólo ingresa un URL parcial tal como www.wikipedia.org/wiki/Train. Para ir a una página principal se introduce únicamente el nombre de anfitrión, como www.wikipedia.org.

Dado que el protocolo HTTP permite que un servidor responda a una solicitud redireccionando el navegador Web a un URL diferente, muchos servidores adicionalmente

permiten a los usuarios omitir ciertas partes del URL, tales como la parte "www.", o el carácter numeral ("#") de rastreo si el recurso en cuestión es un directorio. Sin embargo, estas omisiones técnicamente constituyen un URL diferente, de modo que el navegador Web no puede hacer estos ajustes, y tiene que confiar en que el servidor responderá con una redirección. Es posible para un servidor Web (pero debido a una extraña tradición) ofrecer dos páginas diferentes para URL que difieren únicamente en un carácter "#".

7.9.9. Códigos de estado http.

(De Wikipedia, la enciclopedia libre)

La siguiente es una lista de códigos de respuesta del HTTP y frases estándar asociadas, destinadas a dar una descripción corta del estatus. Estos códigos de estatus están especificados por el **RFC-2616**, y algunos fragmentos en los estándares **RFC-2518**, **RFC-2817**, **RFC-2295, RFC-2774** y **RFC-4918**; otros no están estandarizados, pero son comúnmente utilizados.

El código de respuesta está formado por tres dígitos: el primero indica el estado y los dos siguientes explican la naturaleza exacta de la respuesta, dentro de las cinco clases de respuesta que están definidas y se presentan a continuación.

1xx: Respuestas informativas

Petición recibida, continuando proceso.

Esta clase de código de estatus indica una respuesta provisional, que consiste únicamente en la línea de estatus y en encabezados opcionales, y es terminada por una línea vacía. Desde que HTTP/1.0 no definía códigos de estatus 1xx, los servidores no deben enviar una respuesta 1xx a un cliente HTTP/1.0, excepto en condiciones experimentales.

100 Continúa

Esta respuesta significa que el servidor ha recibido los encabezados de la petición, y que el cliente debería proceder a enviar el cuerpo de la misma (en el caso de peticiones para las cuales el cuerpo necesita ser enviado; por ejemplo, una petición Hypertext Transfer Protocol). Si el cuerpo de la petición es largo, es ineficiente enviarlo a un servidor, cuando la petición ha sido ya rechazada, debido a encabezados inapropiados. Para hacer que un servidor cheque si la petición podría ser aceptada basada únicamente en los encabezados de la petición, el cliente debe enviar Expect: 100-continue como un encabezado en su petición inicial (vea Plantilla:Web-RFC: Expect header) y verificar si un código de estado 100 Continue es recibido en respuesta, antes de continuar (o recibir 417 Expectation Failed y no continuar).[1]

101 Conmutando protocolos

102 Procesando (WebDAV - RFC 2518)

2xx: Peticiones correctas

Esta clase de código de estado indica que la petición fue recibida correctamente, entendida y aceptada.

200 OK

Respuesta estándar para peticiones correctas.

201 Creado

La petición ha sido completada y ha resultado en la creación de un nuevo recurso.

202 Aceptada

La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente pudiere no ser satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar.

203 Información no autoritativa (desde HTTP/1.1)

204 Sin contenido

205 Recargar contenido

206 Contenido parcial

La petición servirá parcialmente el contenido solicitado. Esta característica es utilizada por herramientas de descarga como wget para continuar la transferencia de descargas anteriormente interrumpidas, o para dividir una descarga y procesar las partes simultáneamente.

207 Estado múltiple (Multi-Status, WebDAV)

El cuerpo del mensaje que sigue es un mensaje XML y puede contener algún número de códigos de respuesta separados, dependiendo de cuántas sub-peticiones sean hechas.

3xx: Redirecciones

El cliente tiene que tomar una acción adicional para completar la petición.

Esta clase de código de estado indica que una acción subsiguiente necesita efectuarse por el agente de usuario para completar la petición. La acción requerida puede ser llevada a cabo por el agente de usuario sin interacción con el usuario si y sólo si el método utilizado en la segunda petición es GET o HEAD. El agente de usuario no debe redirigir automáticamente una petición más de 5 veces, dado que tal funcionamiento indica usualmente un Bucle infinito.

300 Múltiples opciones

Indica opciones múltiples para el URI que el cliente podría seguir. Esto podría ser utilizado, por ejemplo, para presentar distintas opciones de formato para video, listar archivos con distintas extensiones o word sense disambiguation.

301 Movido permanentemente

Esta y todas las peticiones futuras deberían ser dirigidas a la URI dada.

302 Movido temporalmente

Este es el código de redirección más popular, pero también un ejemplo de las prácticas de la industria contradiciendo el estándar. La especificación HTTP/1.0 (RFC 1945) requería que el cliente realizara una redirección temporal (la frase descriptiva original fue "Moved Temporarily"), pero los navegadores populares lo implementaron como 303 See Other. Por tanto, HTTP/1.1 añadió códigos de estado 303 y 307 para eliminar la ambigüedad entre ambos comportamientos. Sin embargo, la mayoría de aplicaciones Web y librerías de desarrollo aún utilizan el código de respuesta 302 como si fuera el 303.

303 Vea otra (desde HTTP/1.1)

La respuesta a la petición puede ser encontrada bajo otra URI utilizando el método GET.

304 No modificado

Indica que la petición a la URL no ha sido modificada desde que fue requerida por última vez. Típicamente, el cliente HTTP provee un encabezado como If-Modified-Since para indicar una fecha y hora contra la cual el servidor pueda comparar. El uso de este encabezado ahorra ancho de banda y reprocesamiento tanto del servidor como del cliente.

305 Utilice un proxy (desde HTTP/1.1)

Muchos clientes HTTP (como Mozilla[2] e Internet Explorer) no se apegan al estándar al procesar respuestas con este código, principalmente por motivos de seguridad.

306 Cambie de proxy

Esta respuesta está descontinuada.

307 Redirección temporal (desde HTTP/1.1)

Se trata de una redirección que debería haber sido hecha con otra URI, sin embargo aún puede ser procesada con la URI proporcionada. En contraste con el código 303, el método de la petición no debería ser cambiado cuando el cliente repita la solicitud. Por ejemplo, una solicitud POST tiene que ser repetida utilizando otra petición POST.

4xx Errores del cliente

La solicitud contiene sintaxis incorrecta o no puede procesarse.

La intención de la clase de códigos de respuesta 4xx es para casos en los cuales el cliente parece haber errado la petición. Excepto cuando se responde a una petición HEAD, el servidor debe incluir una entidad que contenga una explicación a la situación de error, y si es una condición temporal o permanente. Estos códigos de estado son aplicables a cualquier método de solicitud (como GET o POST). Los agentes de usuario deben desplegar cualquier entidad al usuario. Estos son típicamente los códigos de respuesta de error más comúnmente encontrados.

400 Solicitud incorrecta

La solicitud contiene sintaxis errónea y no debería repetirse.

401 No autorizado

Similar al 403 Forbidden, pero específicamente para su uso cuando la autentificación es posible pero ha fallado o aún no ha sido provista. Vea autentificación HTTP básica y Digest access authentication.

402 Pago requerido

La intención original era que este código pudiese ser usado como parte de alguna forma o esquema de Dinero electrónico o micropagos, pero eso no sucedió, y este código nunca se utilizó.

403 Prohibido

La solicitud fue legal, pero el servidor se rehúsa a responderla. En contraste a una respuesta 401 No autorizado, la autentificación no haría la diferencia.

404 No encontrado

Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.

405 Método no permitido

Una petición fue hecha a una URI utilizando un método de solicitud no soportado por dicha URI; por ejemplo, cuando se utiliza GET en una forma que requiere que los datos sean presentados vía POST, o utilizando PUT en un recurso de sólo lectura.

406 No aceptable

407 Autenticación Proxy requerida

408 Tiempo de espera agotado

El cliente falló al continuar la petición - excepto durante la ejecución de videos Adobe Flash cuando solo significa que el usuario cerró la ventana de video o se movió a otro. ref

409 Conflicto

410 Ya no disponible

Indica que el recurso solicitado ya no está disponible y no lo estará de nuevo. Este código debería ser utilizado cuando un recurso haya sido quitado

intencionalmente; sin embargo, en la práctica, un código 404 No encontrado es expedido en su lugar.

411 Requiere longitud

412 Falló precondición

413 Solicitud demasiado larga

414 URI demasiado larga

415 Tipo de medio no soportado

416 Rango solicitado no disponible

El cliente ha preguntado por una parte de un archivo, pero el servidor no puede proporcionar esa parte, por ejemplo, si el cliente preguntó por una parte de un archivo que está más allá de los límites del fin del archivo.

417 Falló expectativa

421 Hay muchas conexiones desde esta dirección de Internet

422 Entidad no procesable (WebDAV - RFC 4918)

La solicitud está bien formada pero fue imposible seguirla debido a errores semánticos.

423 Bloqueado (WebDAV - RFC 4918)

El recurso al que se está teniendo acceso está bloqueado.

424 Falló dependencia (WebDAV) (RFC 4918)

La solicitud falló debido a una falla en la solicitud previa.

425 Colección sin ordenar

Definido en los drafts de WebDav Advanced Collections, pero no está presente en "Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol" (RFC 3648).

426 Actualización requerida (RFC 2817)

El cliente debería cambiarse a TLS/1.0.

449 Reintente con

Una extensión de Microsoft: La petición debería ser reintentada después de hacer la acción apropiada.

5xx Errores de servidor

El servidor falló al completar una solicitud aparentemente válida.

Los códigos de respuesta que comienzan con el dígito "5" indican casos en los cuales el servidor tiene registrado aún antes de servir la solicitud, que está errado o es incapaz de ejecutar la petición. Excepto cuando está respondiendo a un método HEAD, el servidor debe incluir una entidad que contenga una explicación de la

situación de error, y si es una condición temporal o permanente. Los agentes de usuario deben desplegar cualquier entidad incluida al usuario. Estos códigos de repuesta son aplicables a cualquier método de petición.

500 Error interno

Es un código comúnmente emitido por aplicaciones empotradas en servidores Web, mismas que generan contenido dinámicamente, por ejemplo aplicaciones montadas en IIS o Tomcat, cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor Web.

501 No implementado

502 Pasarela incorrecta

503 Servicio no disponible

504 Tiempo de espera de la pasarela agotado

505 Versión de HTTP no soportada

506 Variante también negocia (RFC 2295)

507 Almacenamiento insuficiente (WebDAV - RFC 4918)

509 Límite de ancho de banda excedido

Este código de estatus, mientras que es utilizado por muchos servidores, no es oficial.

510 No extendido (RFC 2774)

7.9.10. Comandos y encabezados HTML

Comandos

Comando	Descripción
GET	Solicita el recurso ubicado en la URL especificada
HEAD	Solicita el encabezado del recurso ubicado en la URL especificada
POST	Envía datos al programa ubicado en la URL especificada
PUT	Envía datos a la URL especificada
DELETE	Borra el recurso ubicado en la URL especificada

Encabezados de petición

Nombre del encabezado	Descripción
Accept	Tipo de contenido aceptado por el navegador (por ejemplo, *texto/html*).

Accept-Charset	Juego de caracteres que el navegador espera
Accept-Encoding	Codificación de datos que el navegador acepta
Accept-Language	Idioma que el navegador espera (de forma predeterminada, inglés)
Authorization	Identificación del navegador en el servidor
Content-Encoding	Tipo de codificación para el cuerpo de la solicitud
Content-Language	Tipo de idioma en el cuerpo de la solicitud
Content-Length	Extensión del cuerpo de la solicitud
Content-Type	Tipo de contenido del cuerpo de la solicitud (por ejemplo, *texto/html*).
Date	Fecha en que comienza la transferencia de datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
From	Permite especificar la dirección de correo electrónico del cliente
From	Permite especificar que debe enviarse el documento si ha sido modificado desde una fecha en particular
Link	Vínculo entre dos direcciones URL
Orig-URL	Dirección URL donde se originó la solicitud
Referer	Dirección URL desde la cual se realizó la solicitud
User-Agent	Cadena con información sobre el cliente, por ejemplo, el nombre y la versión del navegador y el sistema operativo

Encabezados de respuesta

Nombre del encabezado	Descripción
Content-Encoding	Tipo de codificación para el cuerpo de la respuesta
Content-Language	Tipo de idioma en el cuerpo de la respuesta
Content-Length	Extensión del cuerpo de la respuesta
Content-Type	Tipo de contenido del cuerpo de la respuesta (por ejemplo, *texto/html*).
Date	Fecha en que comienza la transferencia de datos
Expires	Fecha límite de uso de los datos
Forwarded	Utilizado por equipos intermediarios entre el navegador y el servidor
Location	Redireccionamiento a una nueva dirección URL asociada con el documento
Server	Características del servidor que envió la respuesta

7.9.11. CGI, ISAPI, NSAPI, Servlets y Cold Fusion:

Toda aplicación Web que posea cierto tipo de interacción con el cliente debe acceder a las funciones del servidor. En la actualidad existen dos interfaces que son las más difundidas en el mercado **CGI** (Common Gateway Interface) e **ISAPI** (Internet Server Application Programming Interface).

CGI: Es un método estándar de escribir programas para que funcionan en los servidores Web, a estos programas se los suele llamar "Scripts CGI", los cuales por lo general toman sus datos de entrada de formas HTML que les permiten luego ejecutar tareas particulares. Estos programas ofrecen una gran facilidad de desarrollo y como la interfaz con el usuario es HTML, se pueden acceder desde cualquier navegador. Cada llamada a ejecución de un scripts consume tiempo de CPU y recursos del servidor, por esta razón se debe prestar especial atención a la simultaneidad de las mismas.

ISAPI: En los casos donde prive la eficiencia, es una buena alternativa el empleo de esta interfaz, pues a diferencia de la anterior, las aplicaciones que emplean ISAPI, se compilan dentro de archivos DLL del servidor, siendo sensiblemente más eficientes. Estos archivos son el método nativo del ambiente Windows. La desventaja aquí es que un colapso de DLL puede provocar serios problemas en el servidor.

NSAPI: Es una versión de ISAPI desarrollada por Nestcape, la cual también trabaja con sistemas Unix que soportan objetos compartidos.

Servlets: Se trata de componentes del lado del servidor, que son independientes de las plataformas pues se ejecutan en una máquina virtual Java (JVM). Por ejecutarse dentro del servidor, no necesitan una interfaz gráfica de usuario, permitiendo una interacción completa entre los mismos (usuario y servidor). En el caso de los servlets de Java, estos ofrecen una solución para generar contenido dinámico, son objetos de programa que pueden cargarse en dinámicamente en los servidores Web, ampliando su funcionalidad, desempeñándose mejor que las CGI. Estos Servlets a su vez son muy seguros y pueden emplearse sobre protocolos de seguridad como SSL.

Cold Fusion: Se trata de un producto que integra navegador, servidor y base de datos en importantes aplicaciones Web. Posee una perfecta integración con HTML, lo que lo convierte en una excelente oferta.

7.9.12. Vulnerabilidades:

- Uno de los principales agujeros de **IIS** (Internet Information Server) es debido a la explotación de ISAPI, si los programas de ISAPI se ejecutan bajo la cuenta IUSR_MACHINENAME y se logra invertir la misma, se heredan los permisos de la misma, a partir de aquí se puede ejecutar cualquier tipo de programas, incluso las llamadas al sistema.
- Autenticación arbitraria de solicitudes remotas.
- Autenticación arbitraria de servidores Web.
- Falta de privacidad de solicitudes y respuestas.

- Abusos de características y recursos del servidor.
- Abuso sobre servidores que explotan sus errores y problemas de seguridad.
- Abuso sobre la información de registro (Robo de direcciones, nombres de dominio, archivos, etc.).

7.10. NetBIOS over TCP/IP (RFC 1001 y 1002).

NetBIOS es el protocolo nativo de Microsoft y sobre el cual se basan gran parte de las aplicaciones que operan sobre los niveles de red para las arquitecturas de este fabricante. Inicialmente funcinaba sin la necesidad del empleo de TCP/IP, pero justamente por prescindir de esta pila, se trataba de un protocolo que generaba una gran cantidad de Broadcast innecesario. Con la inevitable conexión a Internet de toda red de área local, se hizo obligatorio el empleo este modelo de capas, y aparece así esta nueva metodología de empleo de NetBIOS sobre TCP/IP, pero manteniendo su estructura particular de nombres, dominios y puertos, temas que se tratarán a continuación.

7.10.1. Puertos.

En el caso de las redes Microsoft Windows, es común el empleo del protocolo NetBIOS sobre TCP (NetBT), el cual emplea los siguientes puertos:

- UDP port 137 (name services)
- UDP port 138 (datagram services)
- TCP port 139 (session services)
- TCP port 445 (Windows 2k en adelante)

7.10.2. Ámbito.

Una vez que un host ha inicializado su dirección IP en forma estática o dinámica, el próximo paso es registrar su nombre NetBIOS. Como ya se analizó, la dirección IP es considerada como un nivel de red en el modelo de capas, se podría desarrollar perfectamente todo el tráfico de una red a través del direccionamiento MAC e IP, pero cada usuario debería conocer esta terminología de cada recurso ofrecido, lo cual quizás sea más eficiente pero muy poco práctico. Para resolver esta situación es que en redes Microsoft, se emplea el protocolo NetBIOS (Servicio Básico de Entradas y Salidas de red), su funcionamiento es similar al BIOS de una PC, el cual hace de interfaz entre el Hardware de la PC y su sistema operativo; en este caso hace la misma tarea pero entre el Hardware de red (Tarjeta de red) y el sistema operativo de

red. Proporciona varios servicios de los cuales el que interesa en este análisis es el Servicio de Nombres, el cual permite identificar recursos o usuarios por medio de quince Byte, y reserva el décimo sexto para reconocer qué tipo de nombre representan los quince anteriores. El protocolo NetBIOS trabaja en el nivel cinco del modelo OSI. Si recordamos que un nivel es autárquico, no tiene forma de relacionarse la dirección IP con el Nombre Net BIOS.

Este protocolo se encuentra estandarizado por las **RFC: 1001** y **1002.** El acceso a este protocolo lo controla la Interfaz **TDI** (Transport Driver Interface). Se define una interfaz de software y una convención de nombres, por lo tanto siendo rigurosos, en realidad no se trata de un protocolo en si mismo. El concepto de NetBT nace de las primeras versiones de Microsoft, que implementaban un protocolo llamado NetBEUI, el cual es muy ágil, pero no ruteable, ante lo cual se sustenta en base a Broadcast, este funcionamiento es inaceptable ante redes que por su tamaño comienzan a emplear switch, y por supuesto no lo soportan los routers. Para mantener su estrategia de nombres es que en los entornos Windows de Microsoft se emplea hoy NetBT. Al salir a Internet, este sistema de nombres pierde sentido, pues es reemplazado por DNS, pero en los entornos locales resuelve los servicios de NT workstation, Server Service, Browser, messenger y netlogon. Es por esta razón que su ámbito está acotado a las redes LAN y no tiene significado en Internet.

7.10.3. Esquema de nombres.

Este sistema de nombres es plano, es decir que todos los nombres de una red deben ser únicos. Su longitud máxima es de 16 caracteres, quedando el último de ellos reservado para identificar el servicio, este último carácter puede tomar los valores que se detallan a continuación:

<computername>[00h] Workstation Service

<computername>[03h] Messenger Service

<computername>[06h] RAS Server Service

<computername>[1Fh] NetDDE Service

<computername>[20h] Server Service

<computername>[21h] RAS Client Service

<computername>[BEh] Network Monitor Agent

<computername>[BFh] Network Monitor Application

<username>[03] Messenger Service

<domain_name>[1Dh] Master Browser

<domain_name>[1Bh] Domain Master Browser

Group Names

<domain_name>[10h] Domain Name

<domain_name>[1Ch] Domain Controllers

<domain_name>[1Eh] Browser Service Elections

7.10.4. Protocolo nodo.

El método de resolución de estos nombres respecto de su dirección IP, depende de cómo sea configurado. El sistema de resolución se realiza a través del llamado protocolo "**nodo**"y ofrece las siguientes opciones:

- nodo-B: emplea Broadcast para resolver el nombre.
- nodo-P: emplea una comunicación Punto a punto.
- nodo-M: emplea primero nodo-B, y si no recibe respuesta usa nodo-P.
- nodo-H: emplea primero nodo-P, y si no recibe respuesta usa nodo-B (caso típico WINS:Windows Internet Name Service).

Un servidor DNS puede ser configurado con un registro especial (Adicionado puntualmente como una zona DNS), que le instruye para que pase al servidor WINS todo nombre que no encuentre en su base de datos.

7.10.5. WINS (Windows Internet Name Services).

Para poder establecer una relación entre estos niveles, es que debe existir una tabla de entradas y salidas, que establezca una correspondencia entre ellos. Esta tabla, nuevamente puede ser estática o dinámica. Inicialmente se operaba estáticamente por medio de un archivo llamado LMHOST.exe, el cual se instalaba en cada equipo de la red, y debía ser actualizado permanentemente en cada cambio en todos los equipos. En la actualidad existen servicios que en forma dinámica van llevando esta relación, en el entorno Microsoft se llama Servicio WINS (Windows Internet Name Service). Este servicio se instala en uno o varios servidores, y automáticamente actualiza las tablas; al comunicarse un host con otro por medio de un nombre NetBIOS, este previamente solicita su dirección IP al servidor WINS, el cual se lo proporciona como se verá a continuación.

a. Trama de Solicitud de registro de nombre: Al iniciar cualquier ETD, necesita registrar su nombre en el servidor por medio de esta trama que ocupa 110 Byte y es dirigida a la dirección IP del servidor.
b. Trama de respuesta: El servidor WINS responderá con un mensaje satisfactorio o de error en forma dirigida. Este mensaje ocupará 104 Byte. Si el nombre es único, el mensaje será satisfactorio. Si se encontrara repetido, el servidor consultará al propietario para determinar si aún está vigente, caso afirmativo enviará un mensaje de error a la solicitud, caso contrario será satisfactorio.

Cuando un nombre fue aceptado por un servidor WINS, este responde con el mensaje satisfactorio correspondiente y le establecerá un tiempo de vida, este tiempo determinará cuándo el cliente deberá renegociar su registro de nombre. Esta renegociación habilita al cliente por un nuevo tiempo de vida. El tiempo de vida definirá el tráfico que se generará por renegociaciones de nombres en la red. Por defecto este lapso suele ser de 144 horas (seis días), lo que producirá una renegociación cada tres días. Esta tarea genera las siguientes tramas:

a. Trama de solicitud de renegociación: Se trata de una trama dirigida de 110 Byte.
b. Trama de respuesta: También dirigida de 104 Byte.

Como se mencionó con anterioridad si un host desea comunicarse con otro por medio del servicio de Nombres, este necesitará resolverlo por medio del servidor WINS, este proceso es justamente llamado servicio de resolución de nombres NetBIOS. Este tráfico como es imaginable ocurre muy frecuentemente, por ejemplo al inicializarse un host, al Explorar la red, al conectarse con un servidor, para un trabajo de impresión, etc. Esta tarea consta de las siguientes tramas:

a. Trama de solicitud de consulta: La envía un cliente WINS a su servidor, contiene el nombre que desea resolver. Esta trama ocurre cada vez que un host desea comunicarse con otro, a menos que este dato se encuentre en su caché de nombres NetBIOS donde residen por defecto durante diez minutos. Esta es una trama dirigida que consta de 92 Byte.
b. Trama de respuesta: Si el nombre solicitado está registrado en el servidor WINS, este contesta con una trama de respuesta dónde incluye la dirección IP solicitada. Esta trama también es dirigida y consta de 104 Byte.

Si el nombre no se encuentra registrado en el servidor WINS, este responderá informando que ese nombre no existe en su base de datos. El cliente generará un mensaje con el protocolo b-node que es un Broadcast para resolver la dirección IP asumiendo que el host destino no es un cliente WINS.

Cuando un host detiene un servicio o se apaga, este libera el nombre dejándolo disponible para otro host. Este proceso también implica un tráfico de dos tramas que se detallan a continuación:

a. Trama de solicitud de liberación: Esta trama es generada por el cliente WINS al servidor en forma dirigida e implica 110 Byte.
b. Trama de respuesta de liberación: La genera el servidor informando que se acepta la liberación asignando el tiempo de vida igual a cero. También se trata de una trama dirigida y consta de 104 Byte.

7.10.6. Los comandos “net”.

Gran parte de las instrucciones de una arquitectura de red Microsoft se aplican mediante los comandos “net”, su uso se realiza desde la interfaz de comandos (MSDOS) y como toda línea

de comandos de esta familia podemos desplegar todas las opciones con "**>net help**" o "**>net /?**"

Es muy frecuente cuando capturamos tráfico en redes Microsoft, ver pasar en el nivel de aplicación "**protocolo SMB**" (Server Message Block: Servidor de Bloques de Mensajes). A este servicio (o servidor) se puede acceder justamente mediante dos puertos "NetBIOS": 139 (Para versiones anteriores a Windows 2000) y 445 (Para versiones posteriores), mediante el comando "**net use**" es que se permite establecer una "sesión nula" por medio de la cual los servidores Microsoft ofrecen la opción de "compartir archivo e impresoras", la integración con Linux es la denominada SAMBA, este comando en particular si no está debidamente asegurado ese servidor es tal vez una de las mayores debilidades de Microsoft, opera sobre el recurso conocido como IPC$. Otra opción útil es "**net view**" que nos listará los dominios disponibles en la red (*aconsejamos probar la herramienta* "**netviewx**"...). Sobre los comandos "net" podríamos hablar muchas páginas más, pero preferimos dejarte esta inquietud para que investigues por tu cuenta, pues existe abundante información en Internet.

Una de las herramientas más populares para investigación de NetBIOS es **Legión**, que se verá en la parte de ejercicios.

7.10.7. Vulnerabilidades.

Como se mencionó, este protocolo no tiene sentido en Internet, pues la resolución de nombres en este ámbito se realiza por medio del protocolo DNS, los puertos de este protocolo (137, 138, 139 y 445) siempre van a estar abiertos en las redes Microsoft, por lo tanto si se logra acceder a una red LAN de estos productos, se sabe cómo poder acceder a cualquier host. Las cuentas de usuario y contraseña de cualquier cliente de una red, no suelen responder a un alto nivel de seguridad, por lo tanto suelen ser altamente violables. Si un intruso logra conectarse desde el exterior con cualquier host de la LAN, rápidamente podría obtener las listas de usuarios de la red, sus servicios, grupos y recursos compartidos, lo cual no es deseado por ningún administrador. También se debe tener presente que en el 16to caracter se pone de manifiesto casi toda la organización de la LAN, por lo tanto cualquiera que pueda capturar este tráfico lo verá.

Por esta razón, JAMAS SE DEBE DEJAR PASAR POR UN ROUTER NI FIREWALL, los puertos de este protocolo, **se deben bloquear siempre en la frontera de toda LAN con Internet**, pues no tiene sentido ninguna comunicación desde adentro hacia fuera o viceversa bajo este protocolo.

7.11. SSL y TLS

7.11.1. Historia

El Secure Socket Layer, (SSL) es un protocolo diseñado originalmente por Netscape Development Corporation para garantizar la seguridad de las transacciones entre sus servidores y clientes en su versión 2.0. A partir de la versión 3.0 se convirtió un estándar utilizado no sólo para el WWW, sino para muchas otras aplicaciones utilizadas en Internet.

Siendo un protocolo que **trabaja entre la capa de aplicación y la capa de transporte**, permite que las aplicaciones existentes sean fácilmente adaptadas para hacer uso de este protocolo, proporcionando privacidad, integridad y autenticidad en la información transmitida.

Basado en el uso de la criptografía de claves públicas, utiliza los certificados X-509 para el manejo de estas claves y la infraestructura desarrollada en torno de estos certificados.

Actualmente es el estándar de comunicación segura en los navegadores Web más importantes, como Nestcape Navigator e Internet Explorer, y se espera que pronto se saquen versiones para otras otros protocolos de la capa de Aplicación (correo, FTP, etc.).

La identidad del servidor Web seguro (y a veces también del usuario cliente) se consigue mediante el Certificado Digital correspondiente, del que se comprueba su validez antes de iniciar el intercambio de datos sensibles (Autenticación), mientras que de la Integridad de los datos intercambiados se encarga la Firma Digital mediante **funciones hash** y la comprobación de resúmenes de todos los datos enviados y recibidos.

Desde el punto de vista de su implementación en los modelos de referencia OSI y TCP/IP, SSL se introduce como una "especie de nivel o capa adicional", situada entre la capa de Aplicación y la capa de Transporte, sustituyendo los sockets del sistema operativo, lo que hace que sea independiente de la aplicación que lo utilice.

Este protocolo también puede aplicar algoritmos de compresión a los datos a enviar y fragmentar los bloques de tamaño mayor a 2^{14} bytes, volviéndolos a reensamblar en el receptor.

7.11.2. De SSL a TLS:

A partir de la versión 3.0 de SSL toma participación IETF-TLS (Internet Engineering Task Force) Group y la convierte en un estándar de Internet bajo la denominación de **TLS** (Transport Layer Security) protocol en el año 1996. Luego se publica como **RFC-2246** (The TLS Protocol Version 1.0) en enero de 1999. En esta RFC hace especial hincapié en que si bien TLS se basa en SSLv3 de Nestcape, sus diferencias no son dramáticas pero pueden llegar a ocurrir casos en que entre ambos ocurra no operatividad.

Un tema de especial interés es que TLS fue diseñado especialmente para evitar el ataque de hombre del medio, es por esta razón que presenta mucha dificultad para pasar a través de proxies, pues considera a estos justamente como un ataque.

7.11.3. Versiones:

En la actualidad existe la versión SSL 3.0 que es la que se estandariza como TLS 1.0. y con las salvedades mencionadas es la que sigue vigente hasta la fecha.

7.11.4. Handshake:

Un túnel TLS se inicia a través de una conexión normal, por ejemplo, en el caso de ser HTTP, primero se establece esta conexión en texto plano y a través del "Handshake" se crea la conexión segura por medio del intercambio de claves (con el método de Diffie - Hellman o Fortezza , certificados RSA o no, y resúmenes MD-5 o SHA) y la generación del secreto compartido, junto con el código de autenticación de mensaje (MAC), se verán más adelante en detalle en el capítulo de “criptografía”.

Hay dos formas de realizar el handshake para iniciar una conexión TLS:

Handshake Completo: Se lleva a cabo el handshake completo para iniciar una conexión, lo cual puede incluir no autenticar las partes, autenticar el server, o autenticar el server y el cliente. Se elige el ciphersuite a usar y se intercambian las claves y secretos.

Handshake Abreviado: Se lleva a cabo el handshake abreviado para reanudar una conexión previa mantenida en el caché del server. Solo se verifican los parámetros del ciphersuite a usar.

Handshake Completo

Esta tabla muestra los diferentes mensajes en la conexión completa:

Cliente	Server
1) Client Hello	
	2) Server Hello
	3) Certificate
	4) Server Exchange
	5) Certificate Request
	6) Server Hello Done
7) Client Certificate	
8) Client Key Exchange	
9) Certificate Verify	
10) Change Cipher Spec	
11) Finished	
	12) Change Cipher Spec
	13) Finished
14) HTTP Request	
	15) HTTP Response
	16) Close Notify Alert

17) Close Notify Alert	

1) Client Hello

Una vez que el cliente (web browser) ingresa una URL **https://.....**, el browser llama al *parser* para que decodifique la URL ingresada. Este se da cuenta que el protocolo elegido es https, e inmediatamente **crea un socket al host, al port 443** (el predefinido para HTTP/TLS cuando el protocolo de transporte es TCP).

Una vez creado el socket, el cliente manda un mensaje ClienteHello al server, en el cual van:

- La versión de SSL,TLS (generalmente 3,1).
- Un número Random que servirá luego para verificar integridad. Y crear el secreto compartido.
- Un session-id (inicialmente con valor 0).
- Los cifrados que soporta el cliente.
- Si empleará o no compresión.
- Genera un hash con todo esto para evitar modificaciones.

2) Server Hello

El server al recibir el mensaje anterior, genera este segundo mensaje con los siguientes datos:

- Versión de TLS que soporta el server.
- Número random generado por el server.
- La session-id que el server asigna a esta sesión.
- El algoritmo de cifrado que elige de los propuestos por el cliente.
- Y el algoritmo de compresión que empleará o no

3) Certificate

El servidor genera este mensaje con la lista de certificados que esté usando. Si depende de una CA, le enviará también el de ésta, para que el cliente pueda validarlo.

4) Server Key Exchange

Si no se emplean certificados, se genera este mensaje para transmitir su clave pública, caso contrario no se emite.

5) Certificate Request

Si el server debe autenticar al cliente (si el servicio que presta así lo requiere), genera este mensaje, en el cual se especifica la lista de CAs (Autoridades de Certificación) en los que confía este server, y los tipos de certificados requeridos, ordenados por preferencia del server.

6) Server Hello Done

Una vez generado el mensaje anterior, se arma este mensaje, que indica el fin del Hello del server. El server envía todos los mensajes juntos, desde el **SeverHello** hasta el **ServerHelloDone** en este momento y se queda esperando la respuesta del cliente.

Ahora, estos cuatro mensajes son mandados al cliente en un solo registro (o en varios si el Record Protocol tiene que fragmentarlo). No se genera el MAC, no se hace compresión, ni se encripta el registro, ya que ningún registro **change_cipher_spec** ha sido mandado aún. Los parámetros de seguridad (ciphersuite) elegidos pasan a ser el Estado Pendiente de TLS.

Todos los mensajes se pasan por la función HASH (incluido el **ClientHello** recibido) para poder verificar luego que no hayan sido falsificados.

7) Client Certificate

El cliente ahora procede a autenticar al server. Si el server puede ser autenticado el cliente prosigue con el handshake, de lo contrario se aborta el handshake. Si se recibe el mensaje **Certificate Request**, el cual le dice al cliente que debe ser autenticado, el cliente debe responder con una lista de certificados de cliente, para que el server lo pueda autenticar. Por lo tanto se genera un mensaje **Certificate**, similar al enviado por el server.

Los datos del mensaje son:

- Lista de certificados del cliente.

8) Client Key Exchange

Para comenzar a generar el secreto compartido, se genera este mensaje. Se trata aquí de 46 bytes de datos generados aleatoriamente (más 2 bytes de la versión de TLS que usa el cliente), que se envían cifrados con la clave pública del server, la cual se obtuvo del certificado del server.

Antes de mandar el mensaje client_key_exchange, el cliente calcula el secreto previo basado en la clave pública que recibió del server, y luego calcula el "key block", a partir del cual se derivan los MAC Secrets, las session keys y los IVs.

9) Certificate Verify

Para probar que los datos que se han recibido y enviado no han sido ni falsificados ni modificados, se genera este mensaje. Tanto el cliente como el server ingresan todos los mensajes del Handshake Protocol (los recibidos y los enviados desde el comienzo del handshake) en las funciones de hashing.

Para evitar que los valores generados por MD5 y SHA sean falsificados o modificados, se firma digitalmente con la clave privada del cliente (la cual él solo conoce).

Cuando el server reciba este mensaje, debe calcular las hashes MD5 y SHA tal cual lo hizo el cliente, y comparar el resultado calculado con lo que recibió en este mensaje. Si los valores coinciden, entonces se prosigue con el handshake, sino se aborta el handshake.

10) Change Cipher Spec

Este mensaje se utiliza para indicar al server que debe hacer de su estado pendiente (los parámetros de seguridad negociados en este handshake) su estado corriente. Esto significa que los siguientes registros que se envíen desde el cliente al server serán protegidos con el ciphersuite y claves negociados.

11) Finished

Este mensaje se genera para verificar con el server que los parámetros de seguridad fueron correctamente calculados por ambos. Este mensaje consta de 12 bytes generados a partir del secreto previo, y los hashes MD5 y SHA calculados hasta ahora.

En este punto, recién se envían al server todos los mensajes anteriores, desde **Client Certificate** hasta **Finished** inclusive.

El cliente se queda esperando la respuesta del server.

12) Change Cipher Spec

El server recibe la respuesta del cliente.

Primero recibe el mensaje **Certificate** del cliente en el cual viaja la cadena de certificados del cliente. El server autentica al cliente, si puede hacerlo continua con el próximo mensaje recibido del cliente, de lo contrario aborta el handshake.

Suponiendo que pudo autenticar al cliente, procesa el mensaje **ClientKeyExchange**, en el que viaja el secreto previo necesario para generar las claves de sesión, los secretos del MAC y los IVs.
El server desencripta el contenido del mensaje usando su clave privada.

Procesa el mensaje **CertificateVerify**, que contiene los hashes generados por el cliente.

Compara los hashes MD5 y SHA calculados por el server con los enviados por el cliente y si son iguales el server continua con el handshake, sino lo aborta.

Si los hashes coincidieron, el server procesa el mensaje **ChangeCipherSpec** enviado por el cliente, que tiene el efecto de cambiar el estado corriente (sin encriptación ni MAC) por el estado pendiente (encriptación). A partir de este momento, el server enviará y recibirá mensajes protegidos con el ciphersuite negociado.

El server procesa ahora el mensaje **Finished** que envió el cliente. Este mensaje viene protegido con el nuevo ciphersuite, y el cliente lo envía para verificar que los parámetros sean correctos. El server desencripta el mensaje con su clave de lectura, genera el MAC con su secreto MAC de lectura y compara con el del cliente para verificar que el mensaje no ha sido modificado, y genera los 12 bytes random de la misma forma que el cliente, si coinciden, entonces *este sentido* de la conexión ha sido verificado.

Ahora el server genera un mensaje **ChangeCipherSpec**, para que el cliente cambie el estado corriente de lectura con el estado pendiente de lectura, para poder usar el nuevo ciphersuite en este sentido de la conexión.

13) Finished

El server genera 12 bytes random de forma muy similar a como lo hizo el cliente en su mensaje Finished, usando el secreto compartido y las hashes calculadas hasta ahora.

Estos 12 bytes los encapsula en un mensaje **Finished**, genera el MAC con MD5, y encripta todo con la clave de escritura.

Luego envía estos dos últimos mensajes al cliente. En este momento la conexión ya está lista para transportar en forma segura datos de la capa de aplicación, salvo que el cliente envíe un mensaje de Alerta para abortar la conexión.

14) HTTP request

El cliente recibe el mensaje anterior del server, y cambia el estado de lectura corriente con el estado de lectura pendiente.

Ahora, el cliente recibe el mensaje **Finished** del server y chequea que la conexión en este sentido tenga los parámetros de seguridad correctos desencriptando el mensaje, comparando el MAC y chequeando los 12 bytes random generados por el server.

Si no se encontró ningún error, el cliente se dispone a hacer el pedido al server. En caso contrario se aborta la conexión enviando un **mensaje Alert** al server.

No habiendo errores en el handshake, el cliente (Web browser) se dispone ha hacer el pedido al servidor Web.

15) HTTP response

Se envía la información solicitada por el cliente.

El resto de la comunicación es de la misma forma para todos los datos del nivel de aplicación que se quieran transferir.

16) Close Notify Alert

Finalmente, el server manda un **Alert Close Notify** para indicarle al cliente que el server terminó. Por supuesto, este mensaje se envía protegido por el ciphersuite.

Dependiendo del protocolo de aplicación que esté usando la conexión, el servidor podría decidir esperar a que llegue el **Close Notify** del cliente para cerrar el socket TLS, o bien cerrarlo sin esperar el **Close nofity**.

17) Close Notify Alert

El cliente recibe este mensaje del server y responde con otro **Close Notify** para cerrar la sesión. Se envía protegido y se cierra el socket subyacente.

Handshake Abreviado

En la tabla muestra los diferentes mensajes en la conexión abreviada. **Mensajes para la conexión con handshake abreviado,** el contenido de los mismos es de carácter similar al completo detallado anteriormente**:**

Cliente	Server
1) ClientHello	
	2) ServerHello
	3) ChangeCipherSpec
	4) Finished
5) ChangeCipherSpec	
6) Finished	
Datos de Aplicación <----->	<-----> Datos de Aplicación

En una forma resumida, el handshake es como sigue:

El cliente envía un **ClientHello** usando el **session-ID** de la sesión a ser reanudada y otros 48 bytes random. El server luego chequea en su caché de sesiones para ver si encuentra esa sesión. Si se encuentra la sesión, y el server está dispuesto a reestablecer la conexión bajo el estado especificado de sesión, mandará un **ServerHello** con el mismo valor de **session-ID** y otros 48 bytes random. Se debe regenerar el secreto compartido y las claves de sesión, los secretos MAC y los IVs. En este punto tanto el cliente como el server deben enviar directamente mensajes **ChangeCipherSpec** y **Finished**. Una vez que el reestablecimiento está completo, el cliente y server empiezan a intercambiar datos de aplicación.

Si el server no puede encontrar el **session-ID** en su caché, el server genera una nuevo **session-ID** y tanto el cliente como el server TLS ejecutan un handshake completo.

7.11.5. Intercambio de claves, algoritmos de cifrado y resúmenes:

Sin la intención de exponer en este punto estas técnicas, se trata solamente de incorporar los nuevos aspectos que permite TLS, en virtud de nuevas legislaciones de EEUU, las cuales permiten la exportación de claves de 56 bits e intercambio e claves de hasta 1024 bits. Teniendo en cuenta esta liberación se incorporan las siguientes posibilidades:

Intercambio de Claves	Clave Intercambio	Cipher	Clave Cipher	Exportable	Hash
RSA	1024 bits	DES (CBC)	56 bits	SI	SHA
RSA	1024 bits	RC4	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	1024 bits	DES (CBC)	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	1024 bits	RC4	56 bits	SI	SHA
Diffie-Hellman (efimera, firma DSS)	sin límite	RC4	128 bits	NO	SHA

También existe documentación para el agregado a TLS del Elliptic Curve Cryptosystem (**ECC**). Que en la actualidad se considera como un algoritmo muy robusto y más veloz que RSA.

Para la implementación de Curvas elípticas se define toda la cipher suite, empleando los siguientes algoritmos de establecimiento de clave: **ECES** (Elliptic Curve Encryption Scheme), **ECDSA** (Elliptic Curve Digital Signature Algorithm), **ECNRA** (Elliptic Curve Nyberg-Rueppel Signature Scheme with Appendix), **ECDH** (Elliptic Curve Diffie-Hellman Key Agreement), **ECMQV** (Elliptic Curve Menezes-Qu-Vanstone Key Agreement).

También se contempla la incorporación de Kerberos. Las credenciales Kerberos se usan para llevar a cabo una autenticación mutua y para establecer un secreto compartido usado subsecuentemente para asegurar la comunicación cliente-servidor.

7.11.6. Puertos definidos:

Teóricamente TLS puede asegurar cualquier protocolo de la familia TCP/IP ejecutándose sobre todo puerto, si ambos lados conocen que en el otro extremo se está ejecutando TLS, sin embargo, en la práctica un grupo de puertos han sido reservados para cada uno de los protocolos comúnmente empleados en Internet, facilitando con esto la tarea a los firewalls. En el año 1998, IANA designó los siguientes puertos para SSL/TLS:

Keyword	Decimal	Description
Nsiiops	261/tcp	IIOP Name Service over TLS/SSL
Https	443/tcp	http protocol over TLS/SSL
Ddm-ssl	448/tcp	DDM-SSL
Smtps	465/tcp	smtp protocol over TLS/SSL
Nntps	563/tcp	nntp protocol over TLS/SSL
Sshell	614/tcp	SSLshell
Ldaps	636/tcp	ldap protocol over TLS/SSL
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
Ftps	990/tcp	Ftp, control, over TLS/SSL
Telnets	992/tcp	telnet protocol over TLS/SSL
Imaps	993/tcp	imap4 protocol over TLS/SSL
Ircs	994/tcp	irc protocol over TLS/SSL
Pop3s	995/tcp	pop3 protocol over TLS/SSL

La lista de puertos y sus detalles puede ser encontrada en:

http://www.isi.edu/in-notes/iana/assignments/port-numbers

7.12. Establecimiento, mantenimiento y cierre de sesiones.

Como mencionamos al principio del libro, el modelo de capas TCP/IP, engloba en el nivel de "Aplicación", los tres niveles superiores del modelo de capas propuesto por OSI, que eran: Sesión, presentación y aplicación. Por esta razón todos los conceptos relacionados al manejo de "sesiones" hemos querido presentarlos en este capítulo, y será lo que trataremos a continuación.

Con excepción del tráfico ocurrido por DHCP, WINS y DNS, la comunicación entre ordenadores requiere el establecimiento de una sesión. Este proceso en redes Microsoft produce en total 11 tramas y 1300 Byte. Debe quedar claro que si un cliente se conecta a múltiples recursos de un servidor, sólo necesitará establecer una sesión con este, repitiendo únicamente el último paso que se verá que es el establecimiento de la conexión.

Se entenderá bajo la arquitectura cliente-servidor, que un host se comportará como cliente durante una sesión con otro que será el servidor, y que luego de la misma podrá encontrarse en la situación inversa pues los roles de cliente y servidor se pueden invertir permanentemente en todos los recursos de una red.

Si bien 1300 Byte parece ser una cantidad respetable, es importante notar que la mayoría del tráfico ocurre después del establecimiento de la sesión, durante la transferencia de información o archivos, cuyo volumen en la generalidad de los casos es inmensamente superior.

7.12.1. Pasos para el establecimiento de sesiones.

En este capítulo nos centraremos en la arquitectura de redes Microsoft, que es la más difundida en este tipo de control de sesiones.

Existen seis pasos que deben ser realizados antes de poder ejecutar una sesión de archivos y que se tratarán a continuación:

a) Resolución NetBIOS/IP:

Este es el primer paso que genera un cliente para conectarse a un servidor, este proceso se llama resolución de nombres NetBIOS y fue tratado con anterioridad.

b) Resolución de dirección de Hardware (MAC):

Una vez que el cliente resolvió el nombre del servidor, debe resolver cual es la dirección de Hardware o dirección MAC, tema que también desarrollamos en capítulos anteriores y que produce las siguientes tramas:

a. Trama de solicitud: La genera el cliente por medio de un Broadcast dentro de la cual especifica la dirección IP destino la cual la obtuvo en el paso anterior. Se trata de una trama de 60 Byte.

b. Trama de respuesta: El ETD que reconoce su dirección IP en la solicitud, envía esta trama especificando en ella su propia dirección de Hardware o MAC, por medio de una trama también de 60 Byte pero ya se trata de una trama dirigida pues cuenta con toda la información para hacerlo.

Se debe recordar que existe previo a este proceso, la consulta a su propia tabla caché ARP en la cual por el lapso de diez minutos cuenta con esta información de haber establecido esta sesión con anterioridad, por lo tanto si encuentra allí estos datos, estas dos tramas no se generan.

c) Establecimiento de la sesión TCP:

Una vez resuelta la dirección MAC e IP, el cliente puede comenzar a establecer la sesión TCP. Dos host deben tener establecida su sesión TCP antes de poder realizar cualquier proceso de comunicación. Basado en un esquema TCP/IP, pues será este el protocolo que garantizará la comunicación de EXTREMO a EXTREMO (Nivel 4 OSI).

Este proceso genera tres tramas de 60 Byte cada una todas dirigidas, y se lleva a cabo una sola vez entre cliente y servidor, pues una vez ejecutada múltiples procesos de capas superiores podrán ejecutarse sobre ésta:

a. Trama de solicitud SYN: La genera el cliente para establecer la sesión.

b. Trama de SYN-ACK Servidor: Respuesta del servidor que acepta el establecimiento de una sesión TCP.

c. Trama de ACK Cliente: Respuesta del cliente para dejar establecida la sesión, garantiza al servidor haber recibido toda la información de establecimiento.

d) Establecimiento de sesión NetBIOS:

Establecida la sesión TCP, el cliente puede continuar con el establecimiento de la sesión NetBIOS. De igual forma que anteriormente, dos host deben tener establecida esta sesión para llevar a cabo una comunicación, se trata de dos tramas dirigidas que en total suman 186 Byte:

a. Solicitud de sesión: Generada por el cliente para iniciar la sesión.

b. Sesión positiva: Responde el servidor comunicando que fue aceptada la solicitud.

Se puede apreciar que no existe una tercer trama que confirme que la aceptación fue recibida; esto se debe a que la comunicación de extremo a extremo la garantiza el TCP, por lo tanto, esta trama llegará.

e) Negociación del protocolo SMB (Server Message Block Protocol).

En este proceso, el cliente envía la lista de todos los protocolos SMB llamados dialectos. El Server, al recibir la lista selecciona el más alto nivel entendido por el cliente y el servidor y responde al cliente inicializando el dialecto con el cual se llevará a cabo la comunicación. Estos dialectos fijarán características como longitudes de nombres soportados, tipos de UNIcodes, etc.

Este paso generará dos tramas dirigidas cuyo tamaño oscila entre 339 y 379 Byte dependiendo del nivel de SMB que entiendan. Esta secuencia sólo necesita ser negociada una vez entre cliente y servidor.

f) Conexión y desconexión:

Realizados los cinco pasos anteriores, recién podrá ocurrir la conexión a los recursos compartidos en la red. En este punto, los archivos son accedidos y transferidos por la red hasta que finalmente la sesión sea terminada.

El cliente envía una configuración de sesión SMB y solicita conectarse a un servidor indicando el nombre del recurso compartido, la conexión deseada, su nombre de usuario y contraseña para esta conexión.

El servidor al recibir la solicitud, valida el usuario y contraseña, y si está habilitado a recibir otra conexión sobre este recurso, envía una trama de respuesta con un mensaje de sesión satisfactoria.

Estas dos tramas generan una cantidad de tráfico variable, dependiendo del nombre de usuario, servidor y del recurso compartido, los valores límites se encuentran entre los 360 y 500 Byte.

7.12.2. Transferencia de datos.

Una vez que la conexión ha sido establecida, se generará la mayor cantidad de tráfico en la red producida por la transferencia de archivos. El análisis de este tipo de tráfico es de vital importancia para la determinación de cuán a menudo son accedidos determinados archivos, qué servidores son los más consultados, qué tamaño de archivos es el óptimo, etc.

7.12.3. Terminación de sesión:

Cuando un usuario ha terminado el acceso a archivos sobre un directorio remoto, la conexión puede ser terminada. Esta desconexión es un simple proceso de solicitud de desconexión al cual el servidor responderá con un mensaje de desconexión aceptada. En esta solicitud de desconexión, el cliente especifica la identidad del directorio remoto al cual desea desconectarse, llamado habitualmente TID (Tree Identify). La desconexión genera dos tramas totalizando 186 Byte entre ambas. Cuando la última de las conexiones de archivos es desconectada, el cliente y el servidor finalizarán su sesión TCP. Ese proceso involucra tres tramas con un total de 180 Byte, las cuáles son de solicitud, respuesta y acknowledgement. Finalizadas estas tres tramas, si el cliente desea establecer otra sesión necesitará repetir los seis pasos.

7.12.4. Tráfico de validación de LOGON.

Una de las primeras funciones que una red necesita proveer a un usuario es su validación de LOGON. Esta suele presentarse por medio de un cuadro donde el usuario ingresa su nombre y contraseña, luego de lo cual el, o los servidores lo validarán o no.

Para que esta actividad pueda ser llevada a cabo, lo primero que el cliente necesita es encontrar un servidor de validación. Una vez que este es identificado tiene lugar una conversación de validación.

El mínimo tráfico que esto implica es como mínimo 24 tramas con un total de 3.105 Byte, producido por cuatro tareas:

- Búsqueda de servidor de validación (mínimo 4 tramas >= 700 Byte).
- Validación (4 a 20 tramas = 765/3725 Byte).
- Preparación (11 tramas = 1280/1370 Byte).
- Terminación de sesión (5 tramas = 360 Byte).

Se debe tener en cuenta especialmente, que en una organización, este tráfico presenta un cuello de botella en el horario de inicio de actividades diariamente. A continuación trataremos con más detalle este tráfico.

1) Búsqueda de un servidor de validación:

Esta actividad se puede llevar a cabo de dos formas distintas:

- Broadcast: Esta solicitud es un broadcast a nivel Ethernet, a nivel 4 emplea UDP, haciendo referencia al port 138 (servicio de datagramas NetBIOS). Este nombre de destino NetBIOS es el nombre de domino en el cual se desea validar seguido de <00> en el 16to carácter.

  Cada servidor de validación que tenga iniciado el servicio de Net Logon, responderá al cliente a través de una respuesta dirigida al nombre de cliente. En esta respuesta incluirá la dirección IP y el nombre del servidor de validación

- Usando WINS: Si el cliente está configurado como un (nodo-h) cliente WINS, este se comunicará con el servidor WINS que tenga configurado para encontrar el servidor de validación de la siguiente manera:

  - Enviará una consulta al servidor WINS anexando <1C> como 16to carácter.
  - En respuesta a esta consulta el servidor WINS entregará una trama de respuesta que incluirá la dirección IP de todos los servidores del dominio. Esta trama variará de tamaño dependiendo del número de controladores de dominio que existan, para dos controladores de dominio, la trama tiene 116 Byte. Esta respuesta llegará a incluir los 25 primeros controladores de dominio, siendo el primero el PDC.
  - El cliente enviará un mensaje dirigido a cada servidor listado en la respuesta WINS, preguntando si alguno puede validar su solicitud. El cliente envía cada solicitud e ignora cualquier respuesta hasta completar el envío a todos los servidores.
  - Cada servidor responde a la validación incluyendo su dirección IP y su nombre.

2) Validación: Esta ocurre como se detalló en el punto anterior, acorde del tipo de búsqueda.

3) Preparación (o establecimiento de la sesión):

El cliente resuelve el nombre NetBIOS del servidor seleccionado por broadcast o dirigido. Una sesión TCP es establecida con el servidor empleando un proceso triple de "Handshake". Una sesión NetBIOS se establece a continuación con el servidor de validación. Luego ocurre una negociación del servicio de bloques de mensajes (SMB), lo cual da por finalizado el establecimiento de la sesión.

4) terminación de sesión: Esta actividad implica 5 tramas, y permite la liberación de los distintos niveles que fueron.

7.13 Tráfico entre clientes y servidores

En toda arquitectura de red donde exista algún tipo de jerarquías entre "Clientes" y "Servidores", se generará un cierto tráfico en el nivel de aplicación que permitirá mantener esta relación jerárquica. Este tráfico se producirá entre "Clientes y Servidores" (el cual un poco ya fue tratado en el punto anterior) y también entre "Servidor y Servidor".

Consideramos importante incluir este tema aquí pues puede ser una de las grandes causas que afecten al rendimiento, estabilidad y disponibilidad de una red.

7.13.1 Tráfico "Cliente – Servidor".

El tráfico cliente – servidor es iniciado por un host cliente comunicándose con un servidor, el análisis de este tráfico está principalmente centrado en tres tipos de comunicaciones: Browser, Sistemas de nombres de dominio y intranet browsing.

1) Servicio de Browser cliente:

Es el tráfico generado por un cliente que se registra asimismo como un posible proveedor de recursos sobre la red.

Después que un cliente se ha validado exitosamente sobre una red, generalmente el próximo paso es acceder a los recursos de la misma. Para asistir a los usuarios en la ubicación de los recursos, las redes Microsoft implementan un servicio llamado "Browser". El proceso de Browsing de un cliente implica los siguientes pasos:

a. Anuncio (1 trama = 243 Byte): Los servidores de recursos son adicionados a la lista de Browse anunciándose ante el master browser. Toda computadora que pueda proporcionar recursos en una red, se anunciará a si misma cada 12 minutos.

b. Backup browser (2 tramas >= 445 Byte): Los master browser comparten la lista de servidores con los backup browser. Para encontrar el master browser local, el cliente envía una "Solicitud de obtención de backup browser" al nombre de dominio con un <1D> como 16to carácter. El master browser responde con una lista de los backup browser disponibles, esta lista puede variar en tamaño, como ejemplo, una lista de 2 backup browser tendrá 230 Byte.

c. Elección (1 trama = 225 Byte): Una computadora cliente recibe una lista de backup browser desde el master browser. Una vez recibida la lista, se conecta a uno de los listados

d. Obtención de lista de servidores (19 tramas = 2150 Byte): La computadora cliente contacta un backup browser para recibir una lista de servidores.

e. Obtención de recursos compartidos (16/19 tramas = 1900/3300 Byte): El cliente contacta al servidor para recibir una lista de recursos compartidos sobre ese servidor.

2) Resolución DNS.

Cuando un usuario desea acceder a un ordenador empleando los comandos y utilidades estándar de red de Windows, como entorno de red o los comandos "net", el nombre Net BIOS del host destino, debe ser resuelto a una dirección IP. Esto es generalmente resuelto, como ya dijimos, por WINS. Cuando un usuario, desea acceder a una computadora empleando utilidades TCP/IP, como Internet Explorer o "ping", el nombre de esa deberá ser resuelto también a una dirección IP; este proceso es llamado resolución de host name y puede ser ejecutado por medio del Sistema de Nombres de Dominio (DNS: también tratado en puntos anteriores). Estos servidores, también impactan sobre la red, y tienen la capacidad de ir pasando las solicitudes a otros servidores DNS o WINS si ellos no pueden resolverlos. Por lo cual, una simple solicitud cliente, puede causar múltiples tramas hasta poder ser resueltas.

Cuando un cliente necesita resolver un TCP/IP host name a una dirección IP, este envía una consulta a su servidor DNS, generalmente de tamaño 256 Byte. DNS emplea el UDP port 53. Cuando el servidor DNS recibe la solicitud determina si tiene una entrada para el nombre solicitado, si la tiene, envía una trama de respuesta que incluye la dirección IP del nombre solicitado. En caso de no poseer esa entrada, pueden suceder dos cosas: enviar un mensaje de "Nombre no existente", o enviar la solicitud a otro servidor DNS, esto último sólo sucederá si tiene configurada la "Recursión" por el administrador. Si el segundo servidor DNS posee esta información, responderá al primer servidor DNS, el cual a su vez responderá al cliente. Caso contrario continuará la recursión (si lo tiene configurado). Otra posibilidad consiste en la consulta del servidor DNS a un servidor WINS, también si así fue configurado, permitiendo hacer uso de las entradas estáticas que posee el servicio WINS.

3) Tráfico de Intranet Browsing (Web).

Cada vez más los usuarios emplean el método de consultas a páginas Web como un medio de obtener información estática o establecer conexiones a archivos. La interfaz gráfica de estas y el rápido y fácil manejo de las mismas incrementan su uso.

El tráfico que genera es quizás el más alto iniciado por un cliente de red. Todo el proceso, desde encontrar una Web, hasta conectarse a ella, implica una enorme cantidad de tráfico, y a su vez los grandes volúmenes de información que se "bajan" (generalmente interfaces gráficas) son considerables.

La conexión a una Web es un proceso simple, el primer paso es resolver el nombre del servidor Web; este puede ser realizado por el estándar DNS visto anteriormente o cualquier otro método.

Una vez resuelto el nombre el cliente deberá establecer una sesión TCP con el servidor a través del Port TCP 80, como ya vimos.

Una vez establecida la sesión, el cliente puede comenzar a solicitar información. El cliente solicita páginas al servidor de Web empleando comandos del protocolo HiperText Transfer Protocol (HTTP). El comando que solicita páginas es "GET", inicialmente se bajarán páginas, luego de las cuales podrá bajarse también gráficos o imágenes a través de solicitudes adicionales HTTP GET.

Cuando un servidor Web recibe una solicitud HTTP y determina que esta información está disponible, responde con un número apropiado de "respuestas HTTP" para completar la solicitud del cliente. Si este archivo es superior a 1.238 Byte existirá más de una respuesta.

7.13.2 Tráfico "Servidor - Servidor".

Este tráfico se genera entre servidores para tareas de mantenimiento del sistema operativo de red. Consta de distintos tipos de mensajes que se tratarán a continuación.

1) Sincronización de cuentas:

Tráfico generado para sincronizar las bases de datos de las cuentas de clientes entre el controlador principal de dominio (PDC) y el o los de backup (BDC).

En una red Microsoft, la validación de los usuarios, puede ser realizada por el controlador primario de dominio o por los de Backup indistintamente. Los cambios realizados en cualquier cuenta de usuario, son ejecutados únicamente en el PDC el cual se encargará de replicarlos en todos los BDC de su dominio.

La sincronización de las cuentas de usuarios ocurre en tres bases de datos que mantiene el sistema:

- La Security Accounts Manager (SAM): Contiene las cuentas de usuarios y grupos que el administrador crea, y también incluye toda computadora adicionada al dominio como los BDC, los servidores independientes y las NT workstation.
- Las construcciones internas de la SAM: Contiene todas las construcciones de grupos locales como administradores, usuarios, invitados, etc.
- LSA: Contiene las claves que son usadas para las relaciones de confianza y contraseñas de los controladores de dominio, también incluye las políticas de seguridad configuradas por el administrador.

La sincronización de las cuentas de usuario ocurre:

- Cuando un BDC es instalado o reiniciado en el dominio:
  - La primera actividad que necesita realizar un BDC es descubrir al PDC, esta actividad se implementa con cuatro tramas y un total de 745 Byte (Solicitud del BDC al servidor WINS de nombre de dominio con el 16to carácter <1B> único del PDC, respuesta del servidor WINS, solicitud para

PDC con la IP y el nombre de dominio dado por WINS, y por último respuesta del PDC.

- Una vez que el PDC fue encontrado, se necesita establecer una sesión TCP.

  El establecimiento de la sesión se realiza de igual manera como fue tratado anteriormente (Resolución de nombre, dirección IP, sesión TCP sesión NetBIOS y servicio de bloques de mensajes).

- El paso final es el establecimiento de un canal seguro, el cual no se cerrará hasta que uno de los controladores sea apagado, este tráfico sólo ocurre durante el inicio del controlador e implica 8 tramas con un total de 1.550 Byte.

- Una vez establecido el canal seguro, el controlador de backup puede comenzar a verificar su base de datos de usuario. Esta verificación consta de tres llamadas a procedimientos remotos (una por cada base de datos) en las cuales le informa al controlador primario el número de serie o ID de versión de cada una de las bases de datos del controlador de backup. Por cada solicitud, existe una "Respuesta RPC" enviada desde el PDC. Estas 6 tramas generan un mínimo de 1.344 Byte dependiendo de la longitud de los nombres y de la cantidad de actualizaciones que se deban realizar en las bases de datos.

Cuando es forzada por el administrador empleando el administrador de servidores. Desde el panel de Administrador de servidores, se pueden realizar distintos tipos de configuraciones dentro de los servidores de red, para que las mismas sean actualizadas en los mismos, la orden para esta tarea se genera desde aquí.

Automáticamente por los controladores de dominio dependiendo del registro de configuración. Por defecto, el PDC verifica sus bases de datos cada 5 minutos comparando cambios en las tres bases de datos, cuando los cambios son notificados, este envía un mensaje a todos los BDC informando que un cambio ha sido realizado en una de las bases de datos. El PDC mantiene una tabla de cada BDC y la ID de versión de cada una de sus bases de datos, si un BDC tiene al día la información, este no envía ningún mensaje. Si envía el mensaje de actualización, es dirigido al nombre NetBIOS de cada BDC identificando este mensaje como un "Anuncio de cambios a UAS o SAM" incluyendo:

- Número de serie.
- Datos cifrados y tiempo.
- Valores de pulso y parámetros random.
- Nombre del PDC y valores de nombres de dominio.
- Las ID de versión o números de serie de cada una de las tres bases de datos y los identificadores de seguridad (SID).

Cuando el BDC recibe un anuncio de actualización desde el PDC, este chequea y verifica el número de versión referenciada en el mensaje, si este es posterior al de sus bases de datos, solicita la actualización (Se conecta, establece una sesión y recibe usando SMB o RPC dependiendo del tamaño de la actualización).

Esta sincronización puede generar una gran cantidad de tráfico si las actualizaciones son muy frecuentes. Esta frecuencia dependerá de la configuración del servicio de NetLogon.

2) Relaciones de confianza:

Tráfico generado durante el establecimiento de una relación de confianza, y también incluye el pasaje de autenticación en dominios diferentes en los que existe relación de confianza. Existen cuatro áreas donde esta actividad genera tráfico:

- El proceso de establecimiento de la relación de confianza genera cerca de 110 tramas y 16.000 Byte. Este proceso ocurre únicamente al establecerse la relación.
- El empleo de una cuenta de confianza también genera tráfico. Este tráfico ocurre si el administrador del dominio en el que se confía necesita asignar permisos a una cuenta del otro dominio para un recurso local, o adición de una cuenta en la que se confía a un grupo local.
- El pasaje de autenticaciones es el más frecuente de este tipo de tráfico. Este tráfico ocurre cuando un usuario de un dominio en el que se confía desea validarse físicamente en el dominio que brinda su confianza, y también cuando un usuario de un dominio confiado desea acceder a recursos del dominio en el que se confía.
- Mantenimiento de una relación de confianza: Existe una pequeña cantidad de tráfico generado por el mantenimiento de una relación de confianza. Cada vez que un PDC de un dominio en el que se confía es reiniciado, necesita verificar la relación. Este controlador consulta un servidor WINS para obtener la lista de todos los servidores del dominio confiado, a continuación intenta un logon en este dominio empleando la cuenta "Usuario de confianza inter dominio", si esta tiene éxito entonces la relación de confianza ha sido verificada. Adicionalmente cada siete días la contraseña asignada a la relación de confianza es cambiada (si durante este lapso, el servidor no se hace presente, la relación de confianza se pierde).

3) Servicio de Browser Servidor:

Tráfico generado por anuncios de servidores, elecciones de master browser y listas de intercambio entre servidores de browser. En una red Microsoft por defecto el rol de master browser es asumido por los servidores NT, generando una cierta cantidad de tráfico entre ellos. Los procesos básicos que se ejecutan son:

- Al iniciar un PDC en un dominio, este asume el rol de master browser del dominio.

- Al iniciar cada BDC asume el rol de backup browser o local master browser, dependiendo si existe ya un master browser en esa subred.
- Cada 15 minutos cada master browser (sobre cada subred) se anuncia a los master browser de otros dominios.
- Cada 12 minutos cada master browser contacta al servidor WINS para obtener la lista de todos los dominios (16to carácter <1B> NetBIOS).
- Cada 12 minutos cada backup browser contacta al master browser para actualizar la lista de browser.
- Los master browser en diferentes dominios, se comunican entre sí para determinar que recursos y servidores pueden ser accedidos.

Todos los host de la red (Como Windows 95, 98, NT workstation y NT server) que tienen componentes de servicios o recursos para compartir en la red, se anuncian cada 12 minutos al local master browser. Esto permite al host encontrarse incluido en la lista de browser para ese dominio, cada anuncio consta de una trama de tamaño mínimo 243 Byte. Si dentro de estos lapsos (12 minutos) este host tiene configuradas capacidades para constituirse como master browser, también generará una trama de solicitud de anuncio de browser (Broadcast de 220 Byte); si existe un master browser , esta responderá (también Broadcast) con una trama de respuesta llamada Anuncio de Local Master. Si no recibe respuesta, espera un período de tiempo específico y se genera un algoritmo de elección por el más alto criterio dentro de los potenciales master browser, luego de cuatro respuestas de elección, el host con el más alto valor gana la elección e inicia un Anuncio de Local Master, para dar a conocer a todos los host que él es el master browser.

El master browser determinará si un potencial browser debe o no constituirse como backup browser, en base al tamaño de las listas y las consultas que sufra. Por defecto todos los BDC son backup browser.

En un entorno TCP/IP un master browser de dominio es elegido en cada dominio. Si está disponible, es el PDC. Uno de los roles del master browser es mantener la lista de los otros dominios que existen en toda la red, esto se realiza de dos modos:

- Cada 15 minutos cada master browser se anuncia a los de los otros dominios por medio de una “Trama de Anuncio de Workgroup” de tamaño mínimo 250 Byte y dirigida a un nombre especial NetBIOS <01> <02>_MSBROWSE_<02> <01>.
- Cada 12 minutos cada Master browser contacta al servidor WINS y consulta por todos los nombres de dominio registrados, estos dominios son adicionados a la lista de browser.

Una vez que los master browser poseen las listas de dominios, cada 12 minutos intercambiarán las listas de browser de cada uno de ellos, consolidando las listas completas de browser de toda la red. Cuando un master browser ha recibido una lista de un par de otro dominio, está en condiciones de distribuirla a todos los de backup proceso que realizará también cada 12 minutos. Los clientes ahora pueden conformar sus listas de browser cuando las necesiten.

4) Replicaciones WINS:

Tráfico generado por replicación de bases de datos WINS en otros servidores WINS.

En una red, si bien un servidor WINS puede soportar hasta 10.000 usuarios, suele ser conveniente contar con más de uno para consistencia de la red y a su vez para agilizar las consultas de usuarios.

Cada entrada en la base de datos WINS varía en su tamaño, dependiendo del tipo de entrada.

- La cantidad de datos guardados para un cliente normal (único) con un simple adaptador de red es entre 40 y 280 Byte.
- SI el cliente es multitarjeta, la cantidad de datos dependerá de la cantidad de direcciones IP que tenga configuradas, este rango también oscilará entre 40 y 280 Byte.
- Si el nombre registrado es un nombre de grupo, como un nombre de dominio <1C> este puede tener hasta 480 Byte.

El conocimiento del tamaño de la base de datos ayudará a determinar la cantidad de tráfico que será generado durante la replicación de los WINS.

Para ingresar otro servidor WINS, lo primero es establecer una relación de replicación, la cual se realiza desde el administrador de WINS e implica los siguientes pasos:

- El WINS local establece una sesión TCP/IP con el WINS destino sobre el puerto TCP 135, esta sesión es un triple handshake de tres paquetes de 180 Byte.
- Se establece un RPC y la base de datos WINS es inicializada, esto genera cuatro tramas y 656 Byte de tráfico.
- Se validan las credenciales y el nombre del WINS a ingresar, esta actividad provoca 580 Byte de tráfico.
- Finalmente se establece una nueva sesión TCP usando el port TCP 42 para verificar la replicación y solicitud inicial de réplica.

Todo este proceso genera aproximadamente 20 tramas con 2.000 Byte, si todo se realizó correctamente, la replicación comienza en este momento.

Una vez que la relación ha sido establecida, las replicaciones pueden ser ejecutadas a intervalos predefinidos o luego que un determinado número de registros ha sido actualizado. Una vez que la replicación es disparada, la verificación de la relación y la base de datos es requerida generando 12 tramas y un total de 900 Byte usando una sesión estándar TCP, solicitudes, respuesta y terminación de la sesión. Finalizada la verificación, los datos pueden ser transferidos a través del puerto TCP 42, teniendo en cuenta que solo serán los que se encuentran modificados desde la replicación anterior.

5) Replicaciones de directorios:

Tráfico generado durante la replicación automática de la estructura de directorios entre computadoras.

El servicio de replicación de directorios permite la duplicación o replicación automática de la estructura del árbol de directorios entre múltiples computadoras sin la intervención del administrador de red.

El proceso de replicación ocurre cada vez que el servidor de exportación detecta que algún cambio ha ocurrido en su árbol de exportación por medio de los siguientes pasos:

- El servidor de exportación notifica a todos en su lista de exportación que han ocurrido cambios en su árbol de exportación, este es un anuncio broadcast que emplea el puerto UDP 138 (Servicio de datagramas NetBIOS), su tamaño oscila generalmente en los 330 Byte.
- Las computadoras de importación realizan una conexión SMB con el servidor de exportación, esta actividad constará de 9 tramas y un total de 1.286 Byte.
- La computadora de importación consulta a la de exportación con un llamado NetRemoteTOD, el cual permite determinar si existen menos de 10 minutos desde la última exportación, generando 330 Byte de tráfico.
- La computadora de importación verifica si necesita replicar todos los archivos o solo los modificados y si tiene que esperar un tiempo específico de replicación, esta tarea genera 18 tramas y aproximadamente 3.500 Byte.
- Existe un paso más de verificación de directorios que implica 30 tramas y cerca de 5.000 Byte.
- Comienza la actualización o la replicación inicial desde el servidor de exportación hasta el de importación. Esta actividad dependerá de la cantidad de archivos a ser transferidos.

El proceso completo de replicación de dos archivos de 1.500 Byte genera 160 tramas y casi 26.000 Byte de datos

6) Servidor DNS:

Tráfico generado por replicación de zonas entre servidores DNS.

En grandes organizaciones un solo servidor DNS no es suficiente para proveer resolución de nombres a la totalidad de los clientes, en estos casos puede ser conveniente implementar un servidor primario y uno secundario por cada zona (Una zona es una base de datos que contiene los registros de un dominio en particular).

El DNS primario en una zona es el que mantiene la base de datos de la misma, el secundario recibe una copia en forma muy similar a los PDC y BDC.

La replicación de la información de la zona produce tráfico en la red.

- Cuando un DNS secundario se reinicia, este contacta al primario e inicia la replicación de la zona, este proceso es llamado transferencia de zona. El primar paso en este proceso es consultar la configuración del DNS primario para verificar que esté activo y conformar los parámetros de la zona. Esta trama es una pequeña consulta llamada SOA (Start Of Authority), en general de tamaño 69 Byte.
- El DNS primario responde con los parámetros de configuración de zona, número de versión de la base de datos e intervalos de replicación.
- El servidor secundario establece una sesión estándar TCP en el puerto 53 del servidor primario, una vez establecida, solicita la transferencia de información de zona.
- El servidor primario responde con la información de la zona. No hay actualización de registros, sino que todos los registros son transferidos. El tamaño de esto variará acorde a la cantidad de registros.
- La sesión TCP es desconectada por medio de cuatro tramas.

Una vez que la zona es transferida, el DNS secundario se conectará con el primario para determinar si la zona necesita ser transferida nuevamente, por defecto esto ocurre cada 60 minutos.

7.14. Detección de Vulnerabilidades

Una vez más presentamos dentro del nivel "Aplicación", el empleo de software que puede ser considerado en otros capítulos, pero hemos decidido hacerlo en este nivel pues en realidad abarca o incluye el control y manejo de tareas que se corresponden a todos los niveles del modelo de capas y por esa razón es que aplica perfectamente a los conceptos del nivel que estamos tratando aquí.

7.14.1. Presentación.

Un "Detector de Vulnerabilidades" es una herramienta (generalmente de software) que básicamente va a lanzar diferentes tipos de ataques hacia uno o varios host, y luego informará a cuáles de ellos el blanco al que fue dirigido presenta debilidades.

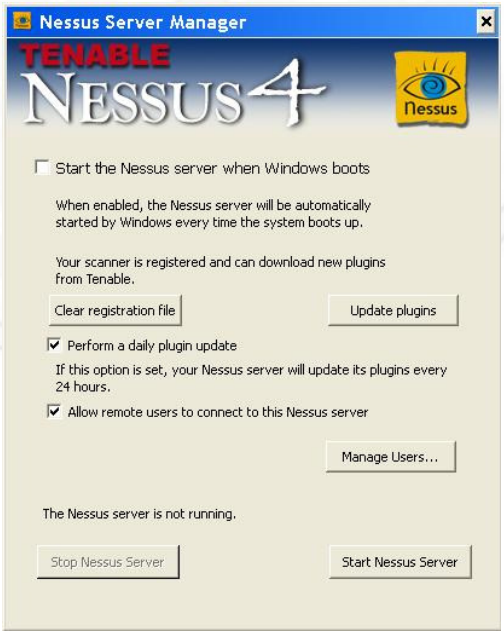
Como su uso habitual es justamente este, independientemente que se emplee para "el lado del mal", también es una muy buena estrategia emplearlo para detectar lo mismo en nuestros propios sistemas y luego de ello analizar las causas para minimizar o evitar su explotación por personas no deseadas.

En el mundo de código abierto, existe desde hace muchos años la mejor herramienta de vulnerabilidades. Muchos podrán opinar que hay herramientas comerciales que lo superan, como siempre son criterios subjetivos y se pueden aceptar también. La herramienta que desarrollaremos aquí es "**Nessus**". Su historia lleva ya varias décadas y nació a través de línea de comandos (que aún hoy puede seguir empleándose) y tal vez el mayor hito lo haya sufrido hace poco con su bifurcación (fork en inglés) a "**OpenVAS"** (Open Vulnerability Assessment System). Este Fork, se produjo cuando "Nessus" fue adquirido por la empresa Tenable Network Security y si bien sigue siendo gratuito para su descarga y empleo, oferta versiones de pago con algunas pocas diferencias y lo más importante es que ya no se tiene acceso a su código. A partir de esta bifurcación, comienzan ciertas diferencias, aunque ambos software son muy similares. En nuestro caso trabajaremos con **Nessus 4.0**, pues es cierto que aún a OpenVAS le quedan algunos detalles para igualar esta versión, pero aconsejamos que evalúes y le sigas el rastro a ambos.

Casi siempre esta herramienta se ha presentado con su versión "Cliente" y su versión "Servidor", esta metodología de trabajo suele ser muy útil a la hora de dejar instalado y actualizado el servidor, y poder conectarnos a él desde diferentes lugares o hosts. En el caso de Nessus, será el servidor quien almacene todo el motor de trabajo, los "plugins" y lo que es más importante: será quien lance el ataque (Si bien existen estrategias de "puente"), este aspecto es vital, pues desde ese host es desde donde deberá poder ser alcanzable la dirección destino del análisis.

7.14.2. Metodología de trabajo con el servidor.

Una vez instalado (Actualmente tanto para Linux como para Windows) el software, nos presentará una consola "Servidor" como la que vemos a continuación.



Lo primero que se debe hacer con ella es crear una cuenta de usuario y luego actualizar la lista de plugins, siguiendo lo s pasos que presentan los respectivos "botones" que podemos apreciar en la imagen anterior. Una vez realizado estos pasos, estaremos en condiciones de "iniciar el Nessus Server" tal cual lo indica también la imagen anterior en el botón de abajo a la derecha.

En esta parte del texto no nos detendremos a ejercitar estas tareas, lo haremos en la parte de "ejercicios" de este capítulo, en lo que centraremos la atención es en la metodología con la cuál deberíamos ser capaces de desenvolvernos con esta herramienta.

Cuando se emplea este tipo de software, no nos podemos quedar simplemente con "saber" que somos vulnerables a cierta cantidad de ataques, sino que lo que en realidad nos interesa es poder llegar al fondo de cada uno de ellos para poder ofrecer la mejor solución a nuestros sistemas, que no siempre podrá ser erradicarlo. Este concepto merece que nos detengamos un poco. En muchísimos casos nos encontraremos con el problema que hemos detectado una vulnerabilidad (o varias) y que la solución para erradicar la misma no es posible de implementar en nuestro sistema. Aunque nos llame la atención nos sucederá muchas más veces de la que imaginamos, pues suele suceder que tenemos instaladas versiones de SSOO, protocolos, hardware o software, aplicaciones, diseños, herramientas, etc... que no soportan que instalemos ese parche, esa actualización, que cerremos ese puerto, que cambiemos de protocolo, etc... Es decir, habremos detectado perfectamente que somos vulnerables a ese patrón de ataque, pero no podemos erradicarlo de raíz. Aquí está la clave: "**el patrón de ataque**", sobre este concepto es que estará basada nuestra metodología. **Si somos capaces de detectar, asilar y entender el "patrón de ataque", tenemos una visión clara de la totalidad de las potenciales soluciones a aplicar**.

Los patrones de ataque los guarda el servidor nessus, son los denominados "plugins" y por cada ataque tiene uno de ellos (si bien hay plugins que se emplean o forman parte de módulos de más de un tipo de ataque), todos estos están redactados en un lenguaje llamado "NASL" (Nessus Attack Scripting Lenguaje). Este lenguaje, se trata de una sencilla forma de programación y muy parecida a otros lenguajes de línea de comandos, por lo tanto todo aquel que tenga experiencia en "C", "perl", "Pitón", etc... le resultará sencillo, y esta es justamente una de las críticas que encontraréis del mismo, su sencillez que lo lleva a acotar su potencia. Para nosotros será un muy buen apoyo, y os invitamos a que busquéis en Internet pues hay suficiente información al respecto. Una vez que se actualizan los "Plugins", encontraremos en el directorio homónimo varias decenas de miles de archivos "*.nasl", cada uno de ellos es un "script" que permitirá llevar a cabo uno o varios ataques".

A continuación presentamos el formato de una regla "NASL", en este caso hemos elegido "**telnet.nasl**" por ser un protocolo que ya conocemos (Hemos agregado en rojo algunos comentarios):

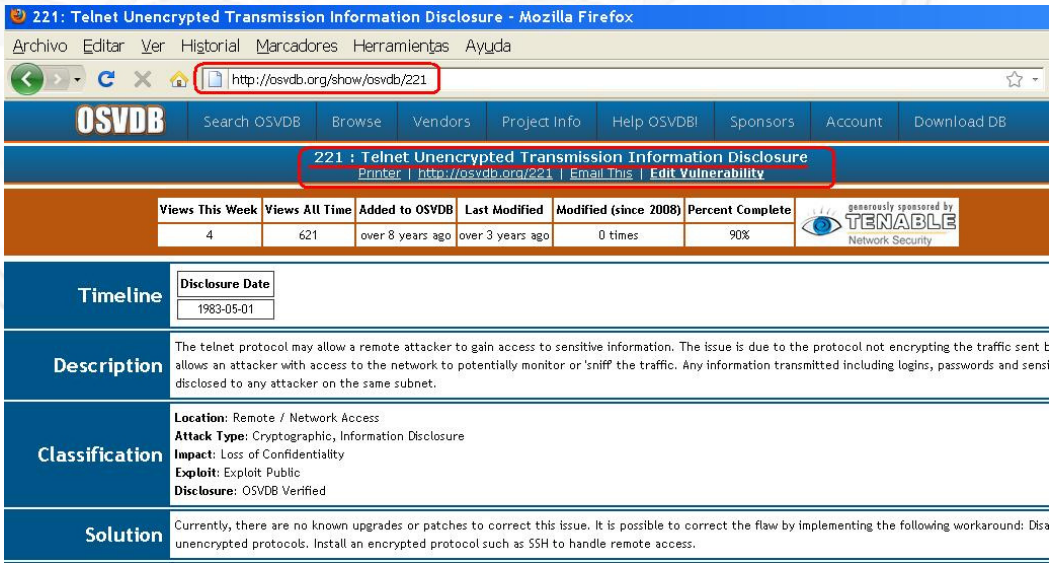
```
#
# (C) Tenable Network Security, Inc.
#
include("compat.inc");

if(description)
{  [inicio de la descripción del plugin]
 script_id(10280);
 script_version ("$Revision: 1.39 $");
 script_osvdb_id(221); [Base de datos MUNDIAL]
```

```
 script_cvs_date("$Date: 2011/07/02 19:59:01 $");

 script_name(english:"Telnet Service Detection");

 script_set_attribute(attribute:"synopsis", value:
"Telnet service appears to be running on the remote system."
);
 script_set_attribute(attribute:"description", value:
"The Telnet service is running. This service is dangerous in
the sense that it is not ciphered - that is, everyone can
sniff the data that passes between the telnet client and
the telnet server. This includes logins and passwords." );
 script_set_attribute(attribute:"solution", value:
"If you are running a Unix-type system, OpenSSH can be used
instead of telnet. For Unix systems, you can comment out the
'telnet' line in /etc/inetd.conf. For Unix systems which use
xinetd, you will need to modify the telnet services file in
the
/etc/xinetd.d folder.  After making any changes to xinetd or
inetd configuration files, you must restart the service in
order
for the changes to take affect.

In addition, many different router and switch manufacturers
support SSH as a telnet replacement. You should contact your
vendor
for a solution which uses an encrypted session." );
 script_set_attribute(attribute:"risk_factor", value:"None" );

 script_set_attribute(attribute:"plugin_publication_date",
value: "1999/08/22");
 script_set_attribute(attribute:"vuln_publication_date",
value: "1983/05/01");
script_set_attribute(attribute:"plugin_type", value:"remote");
script_end_attributes();

 script_summary(english: "Checks for the presence of Telnet");
 script_category(ACT_GATHER_INFO);

 script_copyright(english:"This script is Copyright (C) 1999-
2011 Tenable Network Security, Inc.");
 script_family(english: "Service detection");
 script_dependencie("find_service1.nasl");
 script_require_ports("Services/telnet", 23);
 exit(0);
}[Fin de la descripción del plugin]
#
# The script code starts here
#[inicio del código del plugin]
include("global_settings.inc");
include("telnet_func.inc");
include("misc_func.inc");

port = get_service(svc: "telnet", default: 23, exit_on_fail:
1);

soc = open_sock_tcp(port);
```

```
if (! soc) exit(1, "Cannot connect to TCP port "+port+".");

  r = telnet_negotiate(socket:soc);
  close(soc);
  if(r) {
    set_telnet_banner(port: port, banner: r);
    register_service(port:port, proto:"telnet");
  }
```

Si prestamos atención al formato de "telnet.nasl" anterior, vemos que tenemos una primera parte "descripción" y una segunda parte "código", tratemos con un poco más de detalle a ambas.

- Descripción: Nos presenta toda la información necesaria para comprender, correlacionar y solucionar ese plugin. El "id" es un valor único para cada plugin, la versión nos indica su última modificación, luego tenemos todos los atributos para entenderlo, y también para buscar soluciones, por último las dependencias que posee y la "familia" en la que está comprendido y veremos luego desde la interfaz "cliente". Lo que para nuestro trabajo consideramos muy importante es la línea que indica: "**script_osvdb_id**", pues es allí el punto de encuentro común donde comenzar a analizar Vulnerabilidades, **osvdb** quiere decir: Open Source Vulnerability DataBase y su página Web es: www.osvdb.org. Esta Web concentra identificadores únicos de vulnerabilidades y cuando veamos en el capítulo siguiente el tema de "Sistemas de Detección de Intrusiones" comprenderemos que la mejor forma de controlar las mismas, es justamente pudiendo "currelarlas". Esta Web nos permite hincar cualquier trabajo con vulnerabilidades, en nuestro caso, hicimos una búsqueda por el "id: 221" que es el del anterior plugin, y como podemos ver en la imagen siguiente, nos proporciona toda la información que deseemos.



NOTA: Hace unos años, con motivo de una análisis que realizamos sobre los diferentes productos de detección de intrusiones, escribimos un artículo que fue muy conocido en Internet y aún puede encontrar que se llamó "**Nivel de inmadurez de los NIDS**" y justamente esta inmadurez se debía a que cada producto trataba las

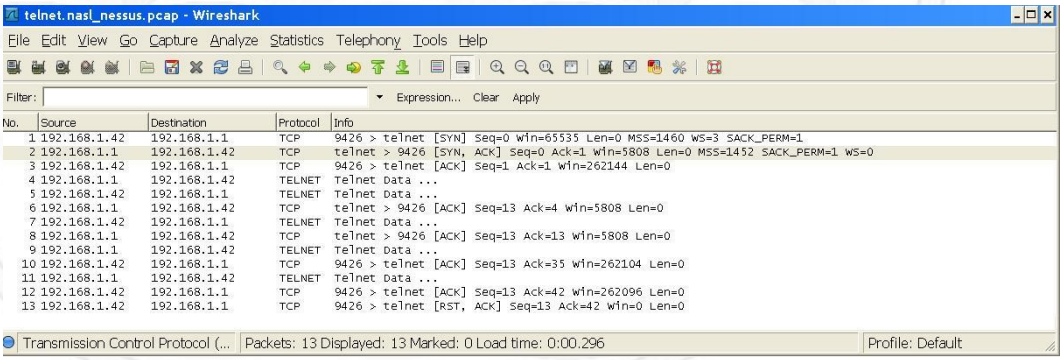
vulnerabilidades como quería por no existir una base de datos común, y como resultado de estas diferencias cada producto detectaba lo que consideraba vulnerabilidad o no según su propio criterio dando como resultado una diferencia abismal entre lo que detectaba cada uno, por lo tanto no era serio el trabajo con este tipo de herramientas pues no era aceptable que uno detectara una cosa y otro otras. Hoy en día gracias a esta posible "correlación" entre lo que un "Detector de vulnerabilidades" reconoce y un "Sistema de Detección de Intrusiones" captura y analiza, se ha convertido en una "dupla" de herramientas INDISPENSABLE para la seguridad de los sistemas.

- Código: Esta parte del script es la que verdaderamente ejecuta el ataque, por esa razón es que nos interesa especialmente, pues aquí está lo que debemos analizar. En nuestro ejemplo de "telnet.nasl", vemos que al principio "incluye" (incluye) algunas librerías al mejor estilo "c", luego llama la función "get_service" que verificará si el puerto 23 está abierto o no, si lo está abrirá ese socket por medio de la función "open_sock_tcp", si este triple "hanshake" es exitoso, negociará una sesión "telnet" por medio de la función "telnet_negotiate", finalmente si pudiera conectarse por telnet nos entregaría el valor de su banner y lo registraría por medio de las funciones "set_telnet_banner" y "register_service".

  Todo esto para los que queráis profundizar en este lenguaje de programación, puede lanzarse vía línea de comandos de la siguiente forma:

  ```
  #nasl –t  Direccion_IP_destino  telnet.nasl
  ```

  Y si lo capturáramos con Wireshark, veríamos las siguientes tramas:

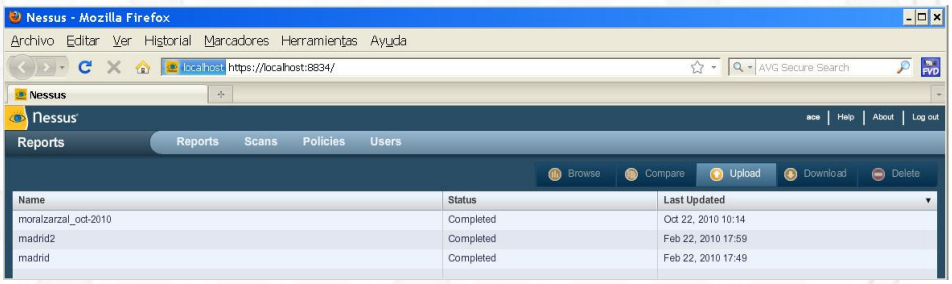


De esta forma, podemos practicar e inclusive generar nuestros propios plugins, también os invitamos a que investiguéis el comando "nessus" por línea de comandos. .Si deseas profundizar en el formato de las reglas "nasl" creemos que la referencia más sólida para que empieces es el viejo manual "**nasl2_reference.pdf**" que si lo buscas, aún está presente en varios sitios de Internet

7.14.3. Metodología de trabajo con el cliente.

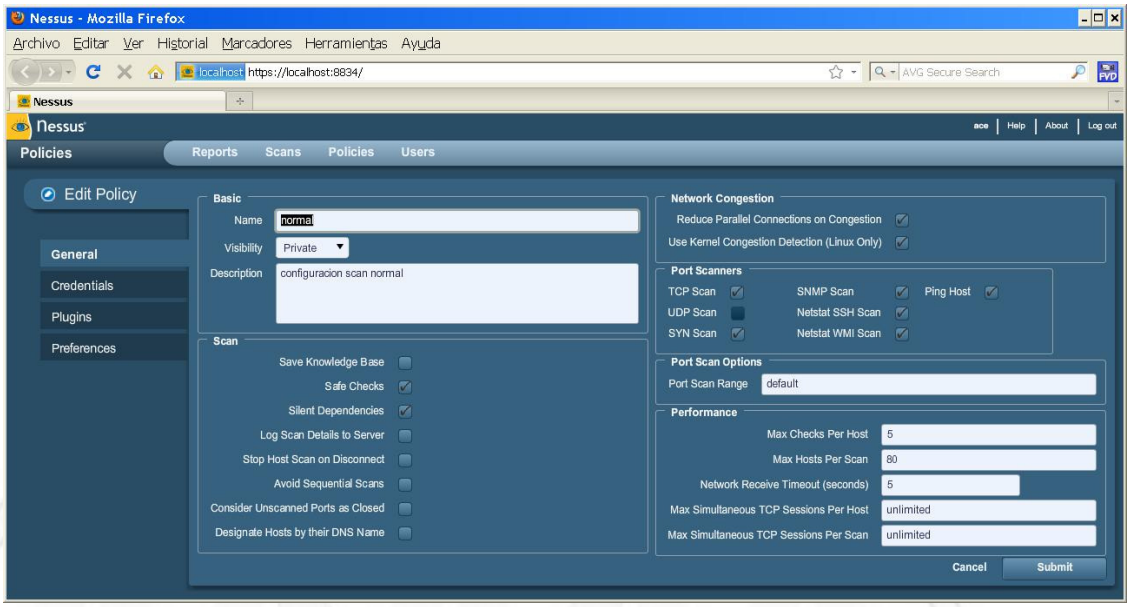
Una vez iniciado el servidor, para poder conectarnos a él desde un entorno gráfico, lo debemos hacer desde el cliente nessus, el cual una vez ejecutado, nos presentará una página Web conectada vía https al puerto 8834 del servidor que acabamos de iniciar, el cual puede ser en local o en la dirección IP en que se encuentre.

Como vemos en la imagen anterior, nos pedirá el usuario y contraseña (que habíamos creado en el servidor), una vez ingresados estos datos se nos presentará la consola principal del cliente nessus:

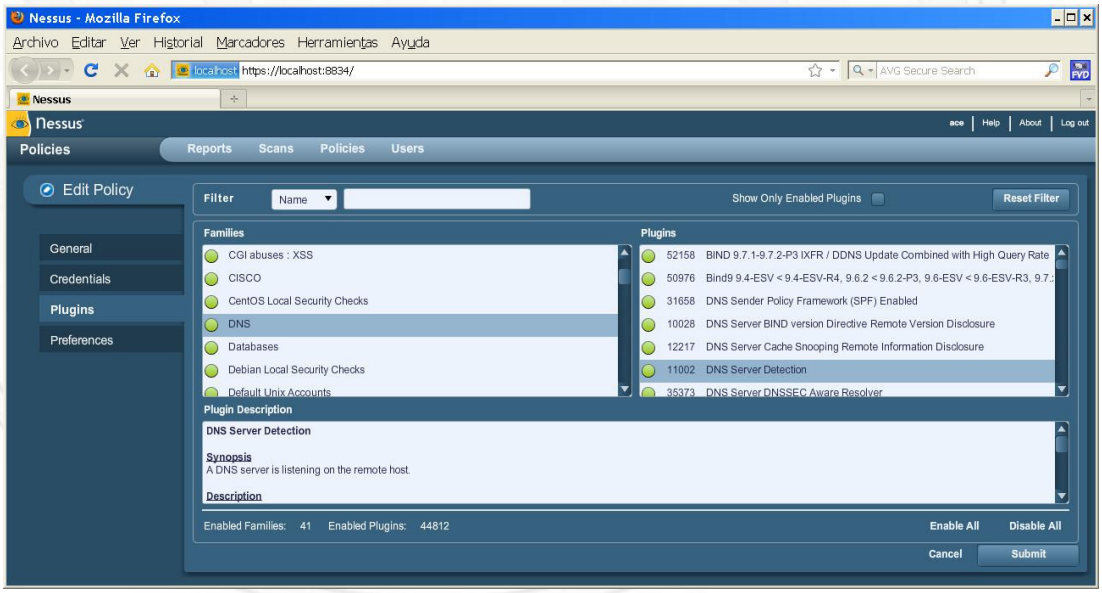


Nuestra metodología de trabajo desde el cliente inicialmente es la de lograr identificar cada "plugin" de Nessus hasta lograr aislarlo de forma que al lanzar un ataque podamos llegar a seleccionar el "mínimo indispensable", es decir, lograr configurar nuestra consola cliente para poder lanzar ataques de la forma más puntual que sea posible, y de esta forma poder capturar únicamente ese patrón de tráfico para poder compararlo con el "código" de ese plugin y de forma práctica analizar "trama a trama" como opera.

Para seleccionar cada uno de los plugins, se debe trabajar a través de la "política de seguridad" que es uno de los nombres que vemos en el menú superior de esta interfaz gráfica:



Como podemos ver, esta interfaz ofrece una configuración general a todos los plugins, unas "credenciales" que podemos configurar para diferentes tipos de conexiones, y luego la serie de "plugins" que acabamos de actualizar, si seleccionamos esta última, se nos presenta la totalidad de los mismos ordenados por las "familias" que mencionamos en la descripción anterior.



Lo que nos interesa por ahora es que en esta vista, nos permite "habilitar" o "deshabilitar" puntualmente cualquier plugin, familia o la totalidad de ellos y a su vez nos presenta toda la información respectiva. Si buscamos nuestro ejemplo "telnet.nasl" podremos verificarlo.

En la parte de ejercicios realizaremos todas las practicas necesarias con esta herramienta, por ahora nuestra intención es que la instales, evalúes, te familiarices con ella y sobre todo comiences a comprender la importancia que más adelante verás de poder trabajar en "dupla"

con un IDS y para ello, lo mejor que puedes ir haciendo para avanzar en nuestra metodología es:

- Buscar, y analizar diferentes "plugins" desde su código.
- Practicar y avanzar en la comprensión del lenguaje "nasl".
- Comprobar el funcionamiento de varios de ellos, capturándolos con "Wireshark" y comparándolos con su código.
- Practicar con la interfaz cliente, pero no "a lo bestia", es decir lanzando ataques así porque sí, sino practicando "acotar" los mismos exclusivamente a lo que tú deseas estudiar, es decir generando la cantidad "mínima e indispensable" de tramas, las cuales por supuesto deberás ir capturando con "Wireshark para evaluar."

7.15. Sistemas de Detección de Intrusiones

Al igual que el punto anterior, hemos decidido incluir este tema en el nivel de aplicación, pues si bien hoy existe Hardware que ya posee preinstalado este tipo de herramientas (se los suele llamar "appliance"), en realidad lo que está haciendo es ejecutar módulos de software que de una u otra forma interactúan a nivel de aplicación con el usuario que los administra.

7.15.1. ¿Qué es un IDS (Intrusion Detection System)?

Un IDS es básicamente un sniffer de red, que se fue optimizando, para poder seleccionar el tráfico deseado, y de esta forma, poder analizar exclusivamente lo que se configura, sin perder rendimiento, y que luego de ese análisis en base a los resultados que obtiene, permite generar las alarmas correspondientes.

La primera clasificación que se debe tener en cuenta es que los hay y de red (Network IDSs o NIDS) y los hay de host (Host IDSs o HIDS). A lo largo de este texto, se tratarán exclusivamente los NIDS, pues son el núcleo de este trabajo, pero conociendo el funcionamiento de estos, es muy fácil pasar a los HIDS.

Luego existen otros criterios más que permiten catalogar estas herramientas, pero no se va a entrar en estos detalles. En este texto se va a emplear "Snort", que es claramente uno de los productos líderes de esta tecnología (a criterio del autor, es el mejor) y es preferible dedicar la atención a aprender brevemente el funcionamiento del mismo.

Otro concepto es el de DIDS (Distributed IDSs), que es la evolución natural del trabajo con NIDS en redes complejas, donde es necesario armar toda una infraestructura de sensores, con la correspondiente arquitectura de administración, comunicaciones y seguridad entre ellos

Por último cabe mencionar que está naciendo el concepto de ADSs (Anomaly Detection Systems) que es una nueva variante de todo lo que se tratará en este texto.

Como todo elemento de seguridad, los NIDS pueden ser vulnerados, engañados, "puenteados" y atacados. Existen muchas estrategias y publicaciones de cómo evadir NIDS, por lo tanto a lo largo del libro se irá tratando de remarcar estos conceptos, para tenerlos en cuenta.

La reflexión final de esta introducción debería ser: que así como hace unos años atrás, esta técnica estaba en pañales (el autor de este texto escribió un artículo aún presente en Internet denominado "*Nivel de Inmadurez de los NIDS"*, que explicaba detenidamente este hecho), hoy se debe considerar como un elemento imprescindible de todo sistema, es más se aprecia que sin estos, "sería como montar, en pleno siglo XXI una operación militar defensiva de noche y sin visores nocturnos".

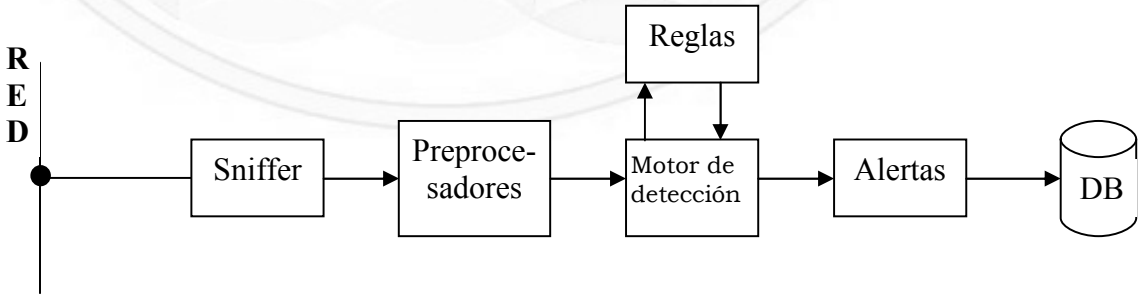
7.15.2. Breve descripción de Snort.

Esta herramienta fue creada por Marty Roesch (Actual Director de Sourcefire que es la versión comercial de Snort) en 1998. Nace simplemente como un Sniffer (o analizador de protocolos) al que luego se le fueron incorporando muchas otras opciones y en la actualidad cuenta con preprocesadores, plugins para bases de datos, varias opciones de envío de alarmas, diferentes esquemas de evaluar paquetes, conectores con Windows, consolas de administración gráficas, etc.

En concreto, Snort consiste de cuatro componentes básicos:

- El Sniffer.
- Los preprocesadores.
- El motor de detección.
- Las salidas.

El funcionamiento es el que se grafica a continuación:



Se detalla brevemente a continuación cada una de las partes:

a.El Sniffer:

Es el paso inicial del funcionamiento de Snort, se relaciona directamente con la tarjeta de red, a la que coloca en modo "promiscuo", es decir, captura la totalidad del tráfico que circula por el cable, independientemente que vaya dirigido a su tarjeta de red o no. Con este primer paso se logra "escuchar" la totalidad de la información del sistema (se debe tener en cuenta el segmento en el que es colocado el dispositivo, pues si hubiere un switch de por medio, este dividiría los dominios de colisión y por lo tanto sólo se capturaría el tráfico correspondiente al segmento en el que se encuentre). El modo promiscuo se puede verificar con el comando "ifconfig", el cual indicará a través de la palabra *promisc*, si la tarjeta se encuentra en este modo

Para el funcionamiento del sniffer se apoya en la librería "libpcap", que es la misma que emplea el programa tcpdump (Snort hace uso de muchos programas ya existentes en el mundo GNU) o la herramienta "Wireshark" con los que venimos trabajando desde el inicio del libro. Una vez capturado cada paquete, se pasa al decodificador (esto lo realiza "decode.c") que es el responsable de interpretar la totalidad de los encabezados de cada nivel, desde enlace hasta aplicación.

b. Los preprocesadores:

Los preprocesadores son un elemento fundamental para el rendimiento de Snort. Como su nombre lo indica, realizan un análisis previo de los paquetes capturados, confrontándolos con sus "plugins" (similares a los vistos con Nessus), para evitar seguir escalando todo el volumen de información y poder realizar evaluaciones más simples. Se tratarán en detalle más adelante, pero en resumen sus tareas son la estandarización de formatos, decodificación, seguimiento de conexiones, análisis scan, etc.

c. El motor de detección:

Esta parte es el corazón de Snort, toma la información que proviene de los preprocesadores y sus plugins y se verifica con el conjunto de reglas, si existe alguna correspondencia con estas últimas, envía una alerta. El tratamiento de las reglas se hará más adelante en este texto.

e. Las salidas.

En la actualidad, Snort permite varios tipos de salidas al detectar una alerta. Pueden ser manejadas en forma local, a través de los logs, enviadas a otro equipo por medio de sockets de UNIX, SMB de Windows, protocolo SNMP, e-mails (SMTP), SMSs, etc.

En cuanto al formato de las alertas es también muy variado y su almacenamiento en diferentes tipos de bases de datos (Se inició con MySQL, actualmente se aconseja Postgres).

La presentación visual de las mismas ofrece a su vez varias alternativas, y existen muchos plugins para Perl, PHP, y todo tipo de servidores Web (La aplicación más tradicional es **ACID** y en este texto se empleó la integración de la misma con la consola **Snort Center**).

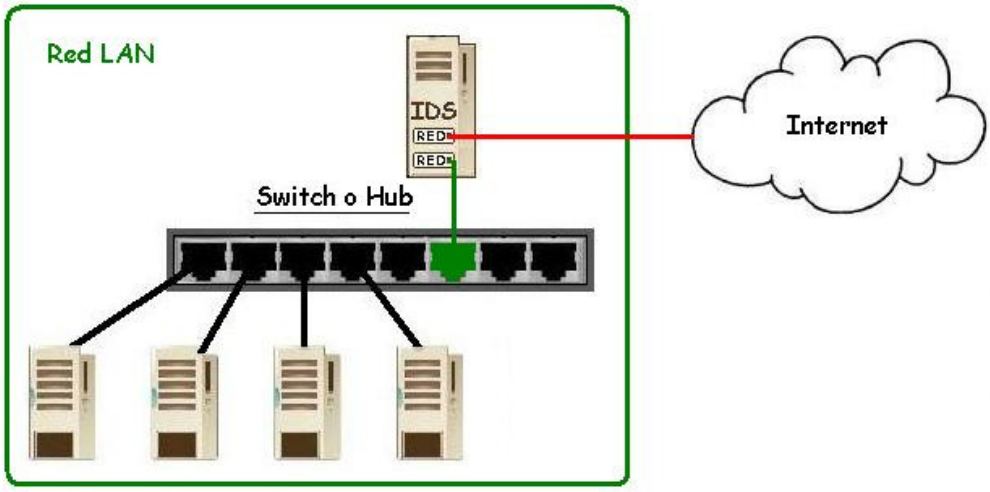
7.15.3. ¿Dónde instalar físicamente Snort?

El software de Snort, si bien hoy existen muchas alternativas, en este texto trataremos su operación bajo Linux, así que básicamente necesitamos tener un Hardware con Linux (de cualquier distribución) ya instalado, pero en realidad lo que nos interesa remarcar en este punto es su ubicación física, pues recordemos que para poder trabajar, necesariamente debe poder capturar la totalidad del tráfico que se desea evaluar como potencial intrusión. Si algún tipo de tráfico no pasa por esta "libpcap" que mencionamos, pues entonces no podrá ser analizado por el IDS, por lo tanto detengámonos a analizar este aspecto.

La tarjeta de red sobre la que operará la captura debe estar recibiendo el 100% del tráfico a analizar, independientemente que venga dirigido a su dirección MAC o IP, o hasta que no posea una dirección MAC o IP destino, pues puede ser un flujo que no responda a la pila TC/IP, y sin embargo nos interese analizar. Básicamente existen cuatro tipos de configuraciones:

a. Opción "casera":

Esta configuración es la más básica, pero no por ello deja de ser eficiente. Consiste en instalar el IDS y conectarlo "interceptando" la totalidad del tráfico que circula a través de él por medio de dos tarjetas de red (o más), tal cual se presenta a continuación poniendo como ejemplo una red LAN en un extremo e Internet en otro:



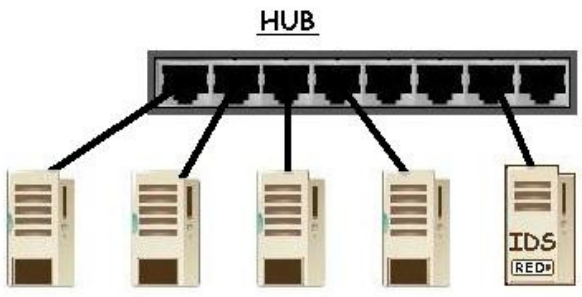
Como se puede apreciar, se presenta una red LAN con cuatro hosts conectados a través de un Switch o Hub al cual se conecta también el IDS (en nuestro ejemplo con el cable verde) y a través de una segunda tarjeta de red (en nuestro ejemplo en rojo) se realiza la conexión a Internet.

En este ejemplo el IDS capturaría la totalidad del tráfico desde y hacia Internet y en el caso de tratarse de un "Hub" también capturaría todo el tráfico interno de la LAN, pero si fuera un "Switch" se perdería del diálogo dirigido entre los hosts de la LAN (si dudamos por qué, repasar el nivel de enlace), en los siguientes puntos veremos cómo se puede hacer también para escuchar este tráfico empleando un "Switch".

- Ventajas de esta configuración:
  1. Sencillez.
  2. Economía.
  3. Facilidad de configuración.
- Desventajas de esta configuración:
  1. Ralentiza el tráfico. Dependiendo de la capacidad del hardware se sentirá más o menos, pero se debe tener en cuenta que el tráfico que ingresa por una tarjeta de red, debe ser "desarmado" en todos los niveles, analizado y luego vuelto a "armar" para sacarlo por la segunda tarjeta.
  2. Punto de fallo: si se cae por cualquier causa el host sobre el que está montado el IDS, queda toda la Red LAN aislada.
  3. Necesita tener una dirección IP expuesta hacia Internet, por lo tanto ese IDS es "detectable, alcanzable y atacable".

b. Empleo de "Hub":

Otra estrategia, es conectar el IDS con una sola tarjeta de red al mismo Hub al que se encuentre conectada la LAN, o esta a Internet (Sería igual). Recordemos que un Hub "explota" toda señal que ingrese a él por todas sus bocas (menos la que ingresó), por lo tanto el IDS estaría capturando la totalidad del tráfico que llega al Hub.

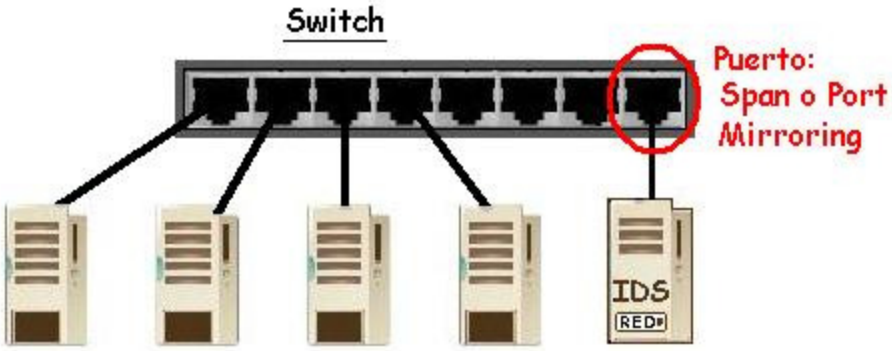


Como se puede apreciar en la imagen anterior, el IDS se puede conectar a cualquiera de las bocas del Hub, pues operan todas de la misma forma.

- Ventajas de esta configuración:
  1. Sencillez.
  2. Economía.
  3. Facilidad de configuración.
  4. Cualquier fallo del IDS no afecta en absoluto a la LAN.
  5. Podemos considerar como una ventaja aún presente, si no olvidamos que la línea de salida hacia Internet, suele operar a velocidades mucho más bajas que la LAN, por lo tanto, si este "Hub" se "pincha" cortocircuitando la salida del Router hacia Internet, es poco probable que merme el rendimiento de la red.
- Desventajas de esta configuración:
  1. En la actualidad es difícil encontrar un Hub que opere a alta velocidad (igual o superior a 100 Mbps). De no soportar la velocidad de la red LAN, ocasionaría un serio cuello de botella o descartaría tramas (de no estar ubicado en el segmento que se mencionó como ventaja).
  2. El IDS puede ser detectado, aunque se puede llegar a configurar para que trabaje sin dirección IP para evitar este hecho.

c. Empleo de "Switch":

Esta configuración físicamente es igual a lo que acabamos de mencionar con el empleo de un Hub, pero la gran diferencia radica en que un Switch no explota la señal por todas sus bocas (excepto los broadast y multicast), sino que las conmuta de acuerdo a lo que tenga almacenada en sus tablas MAC (Como se trató en el capítulo del nivel de enlace), por lo tanto en este caso el IDS por el solo hecho de conectarlo a un Switch, no implica que esté capturando la totalidad del tráfico. Para poder capturar la totalidad del tráfico, debemos realizar cierta configuración en el Switch. No todos los switch lo soportan (aunque en la actualidad la mayoría sí lo hacen), pero en el caso de permitirlo, estos dispositivos tienen un puerto (o a veces más) sobre el que se puede "Espejar" el tráfico de las bocas que se deseen, esto es lo que se denomina "**Port mirroring**" (o también "**Span**"). Para esta configuración hay que conectarse al Switch con permisos de administración (generalmente por telnet, SSH, http o https) y crear los grupos de bocas que se desean "espejar", cada fabricante tiene sus propios pasos, pero no difieren mucho entre ellos. Una vez configurado el "port mirroring" la totalidad del tráfico de los puertos que hayamos designado se replicará en el puerto destino, que por supuesto será donde "pinchemos" el IDS.



- Ventajas de esta configuración:
  1. Permite operar al IDS un poco más rápido (* ver en desventaja).
  2. Continúa siendo sencilla y económica.
  3. Se mantiene el concepto que los fallos del IDS no afectan al resto de la red.
- Desventajas de esta configuración:
  1. MUY IMPORTANTE: Se debe tener en cuenta que, por ejemplo, si el Switch opera a 100 Mbps (si el fabricante es serio....), esto implica que está en capacidad de recibir y entregar tráfico a 100Mbps POR CADA BOCA, por lo tanto, si la red tiene alto tráfico y en determinados momentos está recibiendo tráfico a 100 Mbps por varias bocas..... ¿Qué entregará por la única boca del port mirroring?.....(*)

     Ten muchísimo, pero muchísimo cuidado con esto, pues si bien es un hecho muy poco frecuente, la realidad es que sí sucede, y la única forma que tiene un Switch de procesar este hecho es sencillamente DESCARTANDO TRAMAS hacia esa boca destino, por lo tanto nuestro IDS no estará recibiendo el 100% del tráfico pues parte del mismo ni siquiera le será enviado.

     En la actualidad existen (y aconsejamos su empleo para IDSs) muchos modelos de Switches que traen uno o dos bocas que operan a mayor velocidad que el resto (Ej: un Switch de 24 bocas 10/100 Mbps con 2 bocas de 1 Gbps). En estos casos lo ideal es que el IDS se conecte a una de ellas y se haga el port mirroring a esta.
  2. En este tipo de configuración, el IDS continúa pudiendo ser alcanzado (y/o atacado) por otros hosts de la red.

d. Splitter o TAP:

Un "Splitter" como su nombre inglés lo indica viene del verbo "split: partir" y más específicamente "Splitter: divisor", es un dispositivo que "parte" una señal en dos o más, y para ser estrictos esta debería ser la diferencia con un TAP, el cuál teóricamente debería partir la señal estrictamente en dos (y no más), aunque la realidad cotidiana hace que hoy casi se empleen indistintamente ambos términos. Para nosotros un "Splitter o un TAP" será un dispositivo que ante una señal de entrada, la dividirá al menos en dos señales de salida IDÉNTICAS.

En la actualidad existen muchos modelos con mayores o menores prestaciones cada uno de ellos, pero lo más importante, es que un Splitter, es un dispositivo que tiene una boca de entrada y otra de salida que van conectadas "interceptando" (o cortocircuitando) el tráfico que se desea capturar. Estas dos bocas tienen la particularidad que operan como si fuera un conmutador "normal cerrado", es decir que ante cualquier fallo o caída del Splitter, la red queda "cerrada" con lo que queremos decir que la comunicación se mantiene al 100%, no pudiendo presentar puntos de fallo. A estas dos bocas, es donde va conectada la tercera y/o cuarta boca que es por la cual/cuales se "Bifurca" el tráfico PERO EN UN SOLO SENTIDO, es decir sólo el tráfico entrante de cada una de las dos bocas, no el saliente. Esto está pensado de esta forma, para que justamente el IDS sea un dispositivo PASIVO, que sólo puede RECIBIR el tráfico que pasa a través de ese cable o fibra óptica, pero no pueda emitir, pues directamente a nivel físico, no posee conexión hacia los pines de "envío".

A continuación presentaremos las dos mayores opciones (hay más) que nos interesan conocer para poder trabajar con estos elementos.

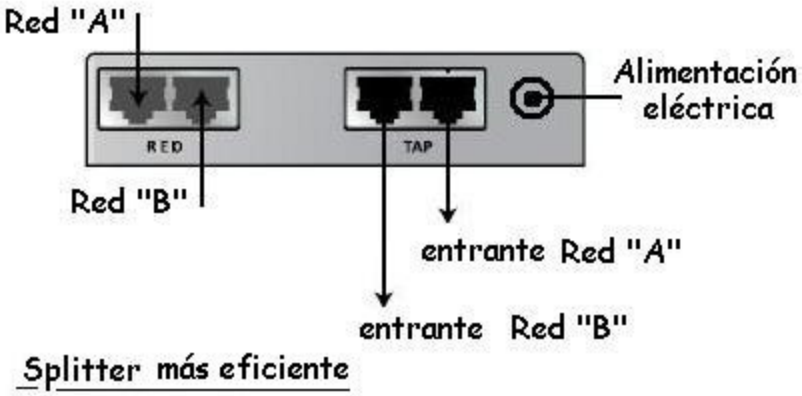
a. Splitter básico:

Se trata del caso más simple, en el cual, como podemos apreciar en la imagen, el tráfico de ambas entradas se replica en una sola boca de salida. Como se deduce, el IDS que esté conectado a esa boca debe ser capaz de operar al doble de la velocidad de cada entrada, pues en el caso de ser "full dúplex" podrá llegar a recibir simultáneamente el máximo de velocidad por cada entrada. En la imagen siguiente se presenta este caso, pudiendo identificar claramente la salida con el nombre de "TAP" (como habitualmente suele estar marcada).

Splitter Básico

b. Splitter más eficiente:

En este caso, la diferencia radica en que por cada entrada posee una boca de TAP, por lo tanto se pueden colocar dos IDS (uno en cada salida) o uno sólo con dos tarjetas de red identificando con total claridad el tráfico que ingresó por cada boca y operando a la misma velocidad real con que ingresaron los datos.



En ambos casos vemos que este dispositivo debe ser alimentado con corriente eléctrica, pues como todo Hardware de red, debe "reconstruir" tramas entrantes y salientes, y esto requiere una lógica de operación en la cual interviene la electrónica.

- Ventajas de esta configuración:

1. Es totalmente transparente a la red monitorizada. Es decir, es imposible detectar la presencia de un IDS por ningún host que forme parte de este circuito. Por supuesto que para la administración de los IDS, se debe poseer otra tarjeta de red totalmente aislada de las redes monitorizadas, esta red suele denominarse "Red de administración o de monitorización", y será por medio de la cual se llega hasta los IDSs.
2. Es tolerante a cualquier tipo de fallos en la totalidad de los elementos de captura y monitorización.
3. Permite operar a los IDSs al máximo de velocidad que soporte la red sin pérdida de tramas.

- Desventajas de esta configuración:

1. Mayor coste.
2. Mayor infraestructura de monitorización (para aislar correctamente todo el sistema).
3. No siempre se tiene permiso, acceso o capacidad de administración sobre los gabinetes de comunicaciones, como para alterar la capacidad de los mismos colocando en ellos nuevos dispositivos.

7.15.4. ¿Cómo se usa Snort?

Si bien en muchos textos se suele clasificar el empleo de Snort, como Sniffer, Logger e IDS y se tratan por separado, aquí para simplificar el empleo y llegar a entender su uso más básico, se comenzará trabajando por línea de comandos y detallaremos más comunes para emplear Snort sin estas subdivisiones.

"**-v**" coloca a snort en modo sniffer (Sólo encabezados de nivel transporte).

"**-d**" incluye los encabezados de nivel red.

"**-e**" incluye nivel de enlace.

EJEMPLO 1:

```
# snort -dev
    Running in packet dump mode
    Log directory = /var/log/snort

    Initializing Network Interface eth0

           --== Initializing Snort ==--
    Initializing Output Plugins!
    Decoding Ethernet on interface eth0

           --== Initialization Complete ==--

    -*> Snort! <*-
    Version 2.0.0 (Build 72)
    By Martin Roesch (roesch@sourcefire.com, www.snort.org)
```

Se puede apreciar en el ejemplo anterior, un inicio de Snort con los mensajes de que todo ha sido realizado satisfactoriamente.

"**-l**" se emplea para indicar el directorio en el que se desean almacenar las alarmas.

"**-b**" almacena datos en formato binario (Igual que tcpdump), es muy útil para luego ser leídos con cualquier analizador de protocolos (como Ethereal).

"**-L**" Se debe emplear junto con la opción -b para indicar el nombre del archivo en el que se almacenarán los datos.

"**-r**" para leer cualquier archivo guardado en formato binario.

"**port**" (puede operar sólo o también con src y dst): Indica el puerto deseado.

"**host**" indica un host para detectar únicamente este rango (también se puede emplear src y dst).

"**net**" indica una red para detectar únicamente este rango (también se puede emplear src y dst).

NOTA: como en casi todos los comandos UNIX, la opción "not", niega la ejecución del mismo. Los operadores lógicos "or" y "and" también son válidos, por lo tanto puede ser empleados con la mayoría de las opciones de Snort.

Se debe tener en cuenta que Snort está fuertemente relacionado al lenguaje de comandos BPF (Berkeley Packet Filter), es decir la masa de las opciones empleadas en sistemas UNIX, este lenguaje permite especificar, hosts, redes, puertos, protocolos, etc. (la lista de estos comandos puede consultarse en www.tcpdump.org).

EJEMPLOS 2:

```
# snort -dev -l /var/snort/log   (Almacenará en formato log en ese path).
# snort -dev host 10.1.1.1 (sólo operará con esa dirección IP)
# snort -dev net 10.1.0.0/16 (Sólo operará con la red 10.1.x.x)
# snort -b -L /var/archivo1    (Almacenará en formato binario en ese path).
# snort -b -r /var/archivo1    (leerá ese archivo desde  binario y en ese path).
# snort -dev host 10.1.1.1 and port 80 (operará sólo con esa dirección IP y sólo el
  puerto origen o destino 80).
# snort -dev host 10.1.1.1 and dst port 80 (operará sólo con esa dirección IP y sólo el
  puerto destino 80).
# snort -dev  not net 10.1.0.0/16 (ignorará la red 10.1.x.x)
```

El empleo más potente es cuando se llama al archivo de configuración de Snort (Aquí es donde algunos textos lo denominan uso como IDS), para esto se emplea la opción "**-c"** y se aclara a continuación el path en donde se encuentra el archivo "**snort.conf**".

EJEMPLO 3:

```
# snort -dev  -l /var/snort/logs net 10.1.1.0/24 –c /var/snort/snort.conf (operará como
  IDS sólo con esa red, guardará los logs en el path indicado y su configuración será
  la guardada en el archivo snort.conf).
```

Para poder hacer funcionar a Snort como IDS de forma eficiente es importante entonces, poder entender y dominar el archivo **"snort.conf"**, que es el que le marca todas las funciones que lo caracterizan como IDS. Este archivo queda configurado por defecto al instalar Snort, y con esta configuración básica ya puede emplearse la herramienta, pero durante este texto, se tratará de centrar la atención sobre este archivo, lo que será el pilar fundamental de este trabajo y lo trataremos con detalle en la parte de ejercicios de Snort.

Por ahora mencionaremos solamente de qué se trata la configuración de este archivo, y que consta de cuatro pasos:

- Paso 1: Configuración de variables.
- Paso 2: Configuración de preprocesadores.
- Paso 3: Configuración de reglas.
- Paso 4: Configuración de salidas.

En nuestra experiencia, un fenómeno muy común que hemos encontrado es la instalación de IDSs casi por defecto y con sólo ello querer obtener resultados, y a nuestro juicio es justamente lo contrario.

> Estamos absolutamente convencidos que recién a esta altura del libro podrás obtener los resultados que deseas, (*pero gracias a que has avanzado metódicamente y nivel a nivel*), y los IDS serán una de las mayores satisfacciones de poder encontrar plasmado plenamente en la práctica todos estos protocolos teóricos que has visto en cada capa. Sin esta base, un IDS para ti sería una "**caja negra**".

Un IDS requiere "Know How", sin esta base es la típica herramienta que no sirve para nada, pero con conocimientos de protocolos de comunicaciones es un apoyo vital para la seguridad de una red.

Para seguir siendo metódicos, continuaremos con el trabajo de ajuste de esta herramienta, pero basados en la "técnica estricta". Así como mencionamos en su momento con los Firewalls que podemos tener una política estricta u holgada, en este caso es similar, podemos dejar que el IDS nos sature (en el mejor sentido de la palabra) con eventos que es lo que suele suceder cuando se instala por defecto, o podemos ser "estrictos" y dejarlo trabajando en donde "nos duele el zapato", es decir, continuar con lo que empezamos con Nessus y volcarlo paso a paso en nuestros IDSs.

Esta metodología nos invita ir al terreno de poder estar en condiciones de evaluar nuestras propias debilidades desde adentro y desde fuera de nuestras redes (con herramientas de detección de vulnerabilidades, igual que haría un intruso), y paulatinamente ir minimizando el impacto de las mismas, con nuestras "Soluciones", "parches", "barreras" y "alarmas".

Nos va a suceder en muchos casos que no podemos hacer frente a estas debilidades, ni encontrarles solución. Como mencionamos en su momento, seguramente nos encontraremos en nuestra infraestructura, con elementos de hardware y/o software que no podremos migrar, actualizar, emparchar, etc... pero sobre los mismos deberemos tomar alguna decisión para evitar que sean explotados por quien no debe, la primera opción sería colocar una barrera, pero recuerda que ya lo hemos comentado, una barrera u obstáculo que no está vigilado no sirve (Hasta es un viejo proverbio militar), y también sucederá que no podemos ponerle una barrera pues se debe llegar al mismo (por ejemplo a servidores que están expuestos en Internet). Ya sabemos que a nadie le gusta dejar vulnerabilidades expuestas en Internet, como tampoco nos gusta tener ventanas en nuestros domicilios si vivimos en un barrio marginal, pero necesitamos luz, mirar hacia fuera, etc...y nos encontramos en la necesidad de asumir ciertos riesgos. . Por lo tanto si quiero tener ventanas en mi casa del barrio marginal, al menos podré poner las alarmas correspondientes para que salten en el momento que alguien se acerca, o intenta romper su cristal... pero ¡ojo! No pongo la alarma apuntando al muro de

hormigón de la pared trasera, ni al suelo interior de mi casa, pues allí no tengo problemas de seguridad, tampoco la pongo apuntando a los árboles de mi jardín, pues desde allí los pájaros la harían saltar tantas veces por día que dejaría de responder cada vez que salta.

Toda esta analogía aunque parezca trivial no lo es, pues la masa de la gente pretende que al instalar un IDS pro defecto trabaje, y al igual que si llenara mi casa de sensores inútiles sería un desperdicio, en nuestras redes y sistemas de información sucede igual, por esa razón es que somos unos acérrimos defensores de trabajar con "técnicas estrictas", es decir, identificar nuestros puntos más débiles y poco a poco comenzar a poner las alarmas, muy bien alineadas con cada punto, con cada detalle. De esta forma estaremos seguros que si "salta una alarma" es seria y se corresponde sin dudarlo a alguna **actividad anómala**.

En el caso de los IDSs, se diferencian claramente dos fallos muy molestos:

- Falsos positivos: Cuando nos salta una alarma que en realidad no nos afecta o es justamente una "falsa alarma".
- Falsos negativos: Cuando una verdadera debilidad de nuestros sistemas o redes está siendo explotada y no nos enteramos.

Para evitar ambos, el mejor camino pasa por dos acciones:

- Estar en capacidad de detectar y mantener actualizada la detección de nuestras debilidades.
- Estar en capacidad de detectar fehacientemente una intrusión sobre cualquiera de ellas (si es que no podemos solucionarla).

La primera de estas acciones es responsabilidad de hacer bien los deberes que tratamos en el punto anterior, y la segunda de ellas pasa por operar adecuadamente los IDSs con "técnicas estrictas", para ello en los ejercicios de este capítulo practicaremos tres partes de los componentes de un IDS (Variables, preprocesadotes y salidas), pero en este desarrollo teórico, avanzaremos un poco más en el cuarto componente, que son las "**Reglas**".

7.15.5. Las reglas de Snort.

Este es un aspecto fundamental de Snort, y que se tratará de aclarar lo mejor posible.

Una regla de Snort, se divide en dos secciones: **Encabezado** y **Cuerpo**.

a. **Encabezado:** Es la parte principal de una regla, especifica qué es lo que debería hacerse al encontrarse una correspondencia con ella, qué protocolo emplear y las direcciones y/o puertos. El encabezado se divide en cuatro categorías: Acción, Protocolo, Fuente y Destino.

1) Acción: Esta le dice a Snort qué debe hacer cuando la regla se cumple.

Posee cinco opciones:

- Pass: Ignora el paquete.

- Log: Almacena el paquete hacia dónde se haya especificado el formato Loging mode (explicado más adelante en el punto 7.).
- Alert: Esta acción almacena igual que el formato log y a su vez también envía una alerta (en modo alerta (explicado más adelante en el punto 7.), acorde a lo configurado en el archivo "snort.conf".
- Dynamic: Esta se emplea de manera combinada con la siguiente acción (activate), y en realidad permanece inactiva hasta que es disparada por un paquete que se corresponde con una regla "activate" que apunta a esta dinámica, a partir de ese momento, esta regla (Dynamic) opera como cualquier otra de tipo "Log".
- Activate: Al cumplirse una regla de este tipo, lo primero que hace es generara una alerta y luego inmediatamente activa la regla dinámica a la que esté dirigida esta regla.

Aparte de estas cinco categorías, Snort ofrece también la posibilidad de programar cualquier tipo que se necesite. Esta actividad no será tratada en este texto, pero se puede buscar la metodología (que es realmente simple) en los mismos manuales de Snort.

2) Protocolo: En la actualidad, Snort soporta cuatro opciones de protocolo: IP, ICMP TCP y UDP. Para especificar el tipo de protocolo, simplemente se coloca este nombre luego de la acción y separado por un espacio. El ritmo de avance de Snort es realmente vertiginoso, así que es conveniente consultar su página web al hacer uso de la herramienta para corroborar el desarrollo de nuevos protocolos, los cuales se están probando cotidianamente.

   Solo se puede declarar un protocolo por regla (y no admite más).

3) Fuente: Esta es la tercera parte del encabezado y admite varias opciones, puede hacerse nombrando las variables estáticas, dinámicas o empleando la notación CIDR, permite también el uso de "any". Puede emplearse también la negación, a través del operador "!" delante de la dirección.

   Dentro del campo Fuente, se encuentra incluido también el puerto origen, el cual puede ser expresado por medio de un número (1 – 65535), o a través del nombre del protocolo al cual se está accediendo por medio de ese puerto. Otra alternativa que presenta es la de declarar o excluir rangos de puertos, se coloca el valor de inicio y el final, separado por ":"

4) Destino: Ídem a Fuente.

   Entre los campos Fuente y Destino, se emplea el operador de dirección. Este operador indica hacia que sentido se dirige el tráfico, y por lo tanto cómo será analizada la regla, las tres opciones son:

   - Fuente -> Destino
   - Fuente <- Destino
   - Fuente<-> Destino

Ejemplos de encabezados de reglas:

```
log tcp 10.10.0.0/16 any <-> 192.168.10.0/24 any
alert icmp any any -> $HOME_NET any
alert tcp any any <-> any any
pass tcp $INTERNAL_NET any -> $EXTERNAL_NET any
alert tcp $EXTERNAL_NET any -> $INTERNAL_NET 137:139
```

b. **Cuerpo:** El cuerpo de una regla no es obligatorio, pero es donde se definen parámetros concretos que pueden ajustar al trabajo de una regla al proceder deseado, describir la misma, los mensajes que debe enviar, etc.

El cuerpo de una regla comienza abriendo un paréntesis y finaliza cerrándolo. El cuerpo se divide en dos partes, separadas por ":": El **cuerpo principal** y las **opciones**.

- El cuerpo principal: Especifica el contenido.
- Las opciones: Consisten en una gran cantidad de parámetros opcionales que se pueden configurar.

El contenido de las opciones se puede escribir en ASCII (Se encierra entre "", y es necesario incluir ". | ´ " se encapsula entre " ") o binario (Se lo representa en su valor hexadecimal, precedido por "|").

Ejemplos:

```
alert tcp any any <-> any any (content: "esto es una opcion";)
alert tcp any any <-> any any (content: "|0105 0BFF|")
```

Existen muchas opciones que permite configurar Snort, las mismas serán tratadas a continuación:

- depth: esta opción permite especificar hasta cuantos bytes después del encabezado IP serán analizados por esta regla. Superados estos bytes, no continúa el análisis. Con esto se reduce tiempo de CPU, pues un paquete cualquiera, independientemente del tamaño que tenga, para este regla, solo contarán los byte que se hayan impuesto en la opción depth.
- offset: Esta opción permite declarar una posición determinada a partid de la cual empezará a buscar dentro del paquete.
- nocase: A través de esta opción, se puede aclarar que la regla no sea sensible a mayúsculas y minúsculas.
- session: Esta opción se emplea para guardar todos los datos de una sesión. Es muy útil en los casos en que se emplea texto plano. Se debe aclarar una de dos acciones que toma esta opción: "all" o "printable".

- uricontent: Esta opción es muy útil para evitar que se analice todo el contenido de un paquete, en los casos en que solo se desea observar dentro del "URI content", es decir al emplear un path determinado dentro de la sección URI de una solicitud.
- stateless: Esta opción en las versiones actuales de Snort, queda incluida dentro de la opción flow (que se trata más adelante).
- regex: Esta opción (en estado experimental), habilita la posibilidad de emplear los conocidos comodines "?" y "*", para reemplazar uno o varios caracteres dentro de la búsqueda.
- flow: La característica importante de esta regla, es que no necesita definir el sentido del tráfico a nivel IP, sino que puede ser útil para seguir el flujo bajo un concepto Cliente-Servidor. El empleo más conocido es cuando un cliente genera tráfico malicioso hacia un servidor, se puede emplear esta opción para verificar ¿qué es? lo que el servidor responde. Se debe especificar cuál es el valor que se le asigna, separado por ":" y posee varias opciones: to_server, from_server, to_client, from_client, only_stream, no_stream, established, stateless.
- "Opciones IP": existe un gran conjunto de opciones para IP, casi todas relacionadas con el formato del encabezado IP: D, M y R (Bits fragmentación y reservados) – sameip (misma IP origen y destino) – ipopts (IP options) – tos (Type Of Service) – ttl (Time TO Life)
- "Opciones TCP": también casi todas relacionadas a su encabezado.

7.15.6. El trabajo con Snort.

Hasta ahora, has visto de forma sencilla la operación básica de Snort. Dejaremos para la sección "ejercicios" todas las prácticas necesarias para que llegues a ser un experto en esta herramienta, pero por ahora seguiremos avanzando con lo que más nos interesa, es decir, que seas "metodológico" en su empleo y que puedas llegar a generar las alarmas vitales para tus sistemas de información. Para este objetivo, lo que nos interesa ahora es llegar a implantar las "técnicas estrictas" que venimos mencionando desde el principio, lo haremos mediante el aprendizaje de las "**local.rules**".

Cuando tratamos "Cómo se usa Snort" mencionamos que para que trabaje en modo IDS, debemos llamar con la opción "-c" a su archivo de configuración: "**snort.conf**". Si abrimos una consola y lo editamos (generalmente se encuentra en: /var/snort/snort.conf), veremos muchas líneas de código y al final de este archivo comienza la "llamada (incluye)" hacia cada una de las "reglas" que posee Snort, las cuales se actualizan prácticamente de forma diaria, abajo presentamos el listado de ellas en la fecha en que estamos escribiendo estas líneas:

```
# site specific rules
include $RULE_PATH/local.rules
```

```
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/blacklist.rules
include $RULE_PATH/botnet-cnc.rules
include $RULE_PATH/chat.rules
include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/imap.rules
include $RULE_PATH/info.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/multimedia.rules
include $RULE_PATH/mysql.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/nntp.rules
include $RULE_PATH/oracle.rules
include $RULE_PATH/other-ids.rules
include $RULE_PATH/p2p.rules
include $RULE_PATH/phishing-spam.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/pop2.rules
include $RULE_PATH/pop3.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/scada.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/snmp.rules
include $RULE_PATH/specific-threats.rules
include $RULE_PATH/spyware-put.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/virus.rules
include $RULE_PATH/voip.rules
include $RULE_PATH/web-activex.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-client.rules
include $RULE_PATH/web-coldfusion.rules
```

```
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-php.rules
include $RULE_PATH/x11.rules
```

Como puedes apreciar la segunda línea de todas ellas es la llamada:

```
include $RULE_PATH/local.rules
```

En breve vamos a comenzar a trabajar exclusivamente sobre ella.

En los puntos anteriores has lanzado Snort con diferentes opciones de comandos **"-l, -b, -r, -L,** etc…" y por último la opción "**-c**" para verificar su funcionamiento en modo IDS, si luego de ello miraste las salidas de consola o el directorio hacia el que estabas dirigiendo los "**Logs**" habrás notado que el IDS si estaba conectado a cualquier red, comenzó a capturar alarmas (insistimos que todo el tema de salidas, preprocesadotes, variables, etc… lo trataremos en detalle en la sección de "ejercicios", por ahora deseamos seguir esta metodología de trabajo), esto se debe a que por defecto ya trae incorporados y configurados los "preprocesadotes" y llama (include) todas las reglas que acabamos de presentar. En estos momentos no nos interesa que haga esto, sino todo lo contrario, deseamos que no capture absolutamente nada que no le ordenemos nosotros. Para ello, el primer paso será comentar ("#") todas las reglas excepto las "**local.rules**", es decir en tu archivo "snort.conf" deberías agregar al principio de cada uno de los "Include" el signo "#", tal cual lo presentamos a continuación:

```
# site specific rules
include $RULE_PATH/local.rules

#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
. . . . . .
. . . .
```

También puedes hacer una copia de seguridad del archivo "snort.conf" y luego directamente borrar estas líneas y dejar únicamente la llamada a "local.rules" y sería igual.

Ahora nuestro objetivo será generar una regla para que nos permita capturar un patrón de tráfico sencillo, por ejemplo "**hola**". Es decir nuestra intención será que Snort escuche tráfico y en el caso de que cualquiera de las tramas tenga en cualquier lado de sus "Bytes" una secuencia de caracteres iguales a "hola", salte una alarma.

Para comenzar con ello, debes ir al directorio donde estén guardadas las "rules", desde allí editar las "local.rules" (que generalmente están en blanco) y crear tu primer regla, para que puedas comenzar este trabajo, te facilitamos el paso inicial, con una regla lo más amplia posible, como la que te presentamos a continuación:

```
alert any any any -> any any (msg:"capturamos la palabra hola";
content:"hola"; nocase; classtype:attempted-user; sid:1001001; rev:1;)
```

Si agregamos la misma y guardamos "local.rules", le acabamos de "ordenar" lo siguiente.

- `alert`: Que genere una alerta.
- `Any`: Cualquier protocolo (TCP, UDP, ICMP, etc…)
- `Any`: Desde Cualquier dirección origen.
- `Any`: Desde Cualquier puerto origen.
- `->`: Sólo en sentido entrante (es decir "desde" la IP-puerto anteriores).
- `Any`: Hacia cualquier dirección destino.
- `Any`: Hacia cualquier puerto destino.
- `(`: Abrimos el "Cuerpo" de la regla.
- `msg:"capturamos la palabra hola";`: Este es el mensaje que recibiremos si salta esta regla.
- `content:"hola";`: Este es el patrón de tráfico que buscará en cualquier posición de esta trama.
- `nocase;`: Para que NO reconozca entre mayúsculas o minúsculas, es decir sería igual que en la trama esté "HOLA", Hola" u "hOLA", etc…
- `classtype:attempted-user;`: El concepto de "classtype" es obligatorio para poder incluirlo dentro de una categoría de alarmas, en nuestro caso hemos elegido este pero podría haber sido cualquier otro (y hasta se puede generar las propias).
- `sid:1001001;`: Es el "Security ID" de Snort, se aconseja que para evitar cualquier solapamiento con reglas propias de Snort que puedan ser generadas por la Comunidad Snort" y podamos siempre diferenciarlas de las "local.rules" se emplee un número de "sid" superior a un millón (1.000.000), también es un campo obligatorio.
- `rev:1;`: Es el número de la revisión de esta regla, que como acabamos de crearla, es la revisión número 1.
- `)`: Cierra el cuerpo de la regla.

Ahora sólo nos quedaría lanzar Snort con la opción "-c" apuntando hacia la ubicación del archivo "snort.conf" y quedar en escucha.

Si deseamos hacer "saltar" esta regla, ya sabemos hacerlo con las diferentes herramientas que hemos visto en los capítulos anteriores para "generar" o "inyectar" tráfico, una de ellas por ejemplo, podría ser "hping3" pero a esta altura del libro ya no te diremos nuevamente cómo tienes que hacerlo, pues deberías estar en perfecta capacidad para inyectar este patrón de tráfico en la red, y si tienes "física o lógicamente" nuestro IDS conectado a este segmento tal cual lo explicamos en el punto anterior, deberías perfectamente hacer saltar esta alarma, ¿o no?.....

Este trabajo, por supuesto no es nada más que el punta pie inicial de la metodología. Sobre esta sencilla regla, te animamos a que empieces a "ajustarla" para que escuche sólo una red externa, un protocolo determinado, hacia una dirección IP específica, un puerto específico. Y si quieres ser más serio, en una determinada posición de la trama, dentro de un determinado protocolo (Ej: ftp, telnet., hettp, etc…), que pueda ser en ASCII y/o binario, etc…, etc…, etc…

Si has logrado lo que te propusimos, creemos que ya estarás visualizando a dónde queremos apuntar. Nuestro objetivo final de la metodología es que, como ya sabes detectar a qué eres vulnerable en tus sistemas (con herramientas de detección de vulnerabilidades), y entiendes los patrones de tráfico que hacen saltar estas debilidades (analizadores de protocolos, reglas ".nasl", scritps de ataque, etc…), ahora emplees estos mismos patrones dentro del "encabezado y cuerpo" de tus "local.rules", para ir generando TODAS las "local.rules" que permitan detectar lo que verdaderamente te afecta **Y SÓLO ESO**.

Es decir el próximo paso es que sigas avanzando en cada una de las vulnerabilidades que posees, generando una local.rule que detecte SÍ o SÍ, cualquier intento de explotación de las mismas. Ya sabes configurar Nessus y otras herramientas para que pueda generar "ese patrón único", por lo tanto, si lo capturas y/o analizas su patrón, ahora con el mismo debes ir probando la creación de una "local.rule" específica para su detección, y una vez que la hayas conseguido ya queda activa en tu IDS.

Si logras dejar configurado tu IDS, en la detección de la totalidad de las vulnerabilidades que no has podido bloquear en tus sistemas, ya tienes un sistema de alertas en tiempo real "ajustado al 100%" a tu infraestructura, y eso te garantizamos que es raro (muy raro) de encontrar en las organizaciones que trabajan con IDSs, y sin embargo es la metodología óptima para trabajar con ellos.

Ahora que has aprendido la parte dura de este trabajo, te damos una pista que te allanará mucho la tarea: En realidad es muy probable que alguna (o muchas) de las reglas que acabas de crear, en realidad ya existan dentro de alguna de las familias que llama "snort.conf" en la lista de "include". Si existe, la verdad es que te ahorras una gran parte del trabajo (aunque queríamos que primero "te lo curres"), para encontrarla, cuando conozcas el manejo de reglas de Snort, te resultará muy fácil, pero el primer paso es tener claro al menos de qué familia se trata, es decir, si la vulnerabilidad es del protocolo "http" y afecta a IIS, pues únicamente busca allí y no en el resto de las reglas de Snort, de esta forma lo importante es primero tener claro a que tipo de "familias" se está atacando para acotar el trabajo.

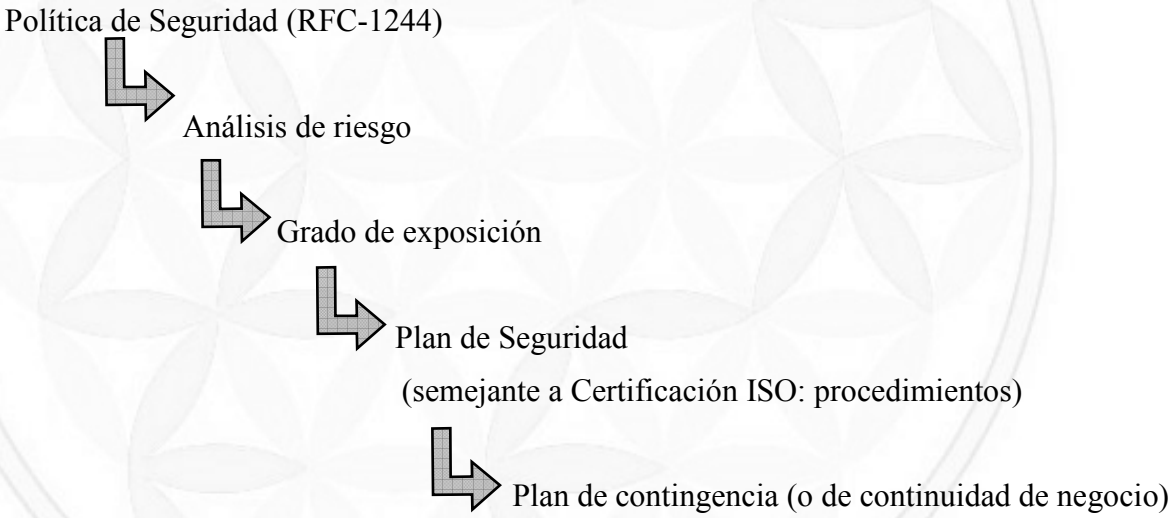
En la sección ejercicios, continuarás avanzando en esta metodología y en el resto de los parámetros de configuración de Snort, también verás una interfaz gráfica de visualización de eventos con "**ACID**" y una consola gráfica de administración de sensores con "**SnortCenter**".

7.16. Honey Pots.

En muchos casos y organizaciones, este continúa siendo un tema aún no implantado con toda la rigurosidad que creemos debe ser tomado. En este punto desarrollaremos el "por qué" consideramos que estas herramientas son imprescindibles y luego la parte práctica para poder implantarla.

7.16.1. ¿Por qué honey pots?

Si se implementa un sistema informático bajo la arquitectura o el modelo de referencia TCP/IP, existen varias RFC que regulan o estandarizan metodologías y procedimientos para asegurar el mismo. La actual política de seguridad (**RFC–2196** Site Security Handbook) y también la anterior (**RFC-1244**, que si bien queda obsoleta por la primera es muy ilustrativa), planten una metodología muy eficiente de feedback partiendo desde el plano más alto de la organización hasta llegar al nivel de detalle, para comparar nuevamente las decisiones tomadas y reingresar las conclusiones al sistema evaluando los resultados y modificando las deficiencias. Se trata de un ciclo permanente y sin fin cuya característica fundamental es la constancia y la actualización de conocimientos. Esta recomendación plantea muy en grande los siguientes pasos:



La política es el marco estratégico de la organización, es el más alto nivel. El análisis de riesgo y el grado de exposición determinan el impacto que puede producir los distintos niveles de clasificación de la información que se posee. Una vez determinado estos conceptos, se pasa al Cómo que es el Plan de seguridad, el cual si bien no guarda relación con las normas ISO, se mencionan en este texto por la similitud en la elaboración de procedimientos de detalle para cada actividad que se implementa.

Sobre el punto en el cual se desea prestar especial atención en esta investigación es, dentro de esta RFC, el 2.5. (SIC):

"Protect and Proceed:

1. *If assets are not well protected.*
2. *If continued penetration could result in great financial risk.*
3. *If the possibility or willingness to prosecute is not present.*

4. *If user base is unknown.*
5. *If users are unsophisticated and their work is vulnerable.*
6. *If the site is vulnerable to lawsuits from users, e.g., if their resources are undermined.*

Pursue and Prosecute:

1. *If assets and systems are well protected.*
2. *If good backups are available.*
3. *If the risk to the assets is outweighed by the disruption caused by the present and possibly future penetrations.*
4. *If this is a concentrated attack occurring with great frequency and intensity.*
5. *If the site has a natural attraction to intruders, and consequently regularly attracts intruders.*
6. *If the site is willing to incur the financial (or other) risk to assets by allowing the penetrator continue.*
7. *If intruder access can be controlled.*
8. *If the monitoring tools are sufficiently well-developed to make the pursuit worthwhile.*
9. *If the support staff is sufficiently clever and knowledgable about the operating system, related utilities, and systems to make the pursuit worthwhile.*
10. *If there is willingness on the part of management to prosecute.*
11. *If the system adminitrators know in general what kind of evidence would lead to prosecution.*
12. *If there is established contact with knowledgeable law enforcement.*
13. *If there is a site representative versed in the relevant legal issues.*
14. *If the site is prepared for possible legal action from its own users if their data or systems become compromised during the pursuit."*

Es decir:

"Proteger y proceder:

a. *Si los activos no están bien protegidos.*
b. *Si la penetración puede originar un gran riesgo financiero.*
c. *Si la posibilidad o la voluntad para llevarlos a juicio no está presente.*
d. *Si la base (o listado) de usuarios es desconocida.*
e. *Si los usuarios no están concienciados en seguridad y su trabajo es vulnerable.*
f. *Si el sitio es vulnerable a las demandas de los usuarios, por ejemplo, si sus recursos no se encuentran debidamente claros o configurados.*

Seguir y Perseguir:

1. *Si los activos y los sistemas están bien protegidos.*
2. *Si se dispone de una correcta metodología de copias de seguridad.*

3. *Si el riesgo de los activos (o recursos) está compensado (es equivalente)a los problemas que puede causar una intrusión en el presente o futuro.*
4. *Si los ataques ocurren con gran frecuencia e intensidad.*
5. *Si el sitio tiene un atractivo natural para los intrusos, o suele atraer a los intrusos.*
6. *Si la intrusión al sitio puedo ocasionar pagos financieros (u otros) al permitir que la intrusión continúe.*
7. *Si el acceso de intrusos puede ser controlado.*
8. *Si las herramientas de monitorización están adecuadamente desarrolladas para el seguimiento de la intrusión.*
9. *Si el personal de soporte está lo suficientemente capacitado e instruido para realizar el seguimiento de la intrusión.*
10. *Si hay voluntad por parte de la organización (o Dirección) de llevar a cabo una acción judicial.*
11. *Si los administradores del sistema conocen, en general, qué tipo de pruebas o evidencias se deben tener en cuenta.*
12. *Si hay un contacto establecido con las fuerzas legales o policiales.*
13. *Si personal capacitado o formado en las cuestiones jurídicas pertinentes.*
14. *Si el sitio está preparado para una posible acción legal sobre sus propios usuarios si sus datos o sistemas se ven en peligro durante la persecución. "*

Los párrafos que acabamos de presentar de esta RFC, nos parecen fundamentales a esta altura del libro, pues en definitiva nos dicen con total claridad que si en realidad tenemos un sistema asegurado con seriedad, no podemos andar con "medias tintas", ya no nos sirve y no es eficiente "desconectar todo" ante el primer intento de intrusión. Si hasta ahora hemos hecho las cosas bien, nuestro siguiente paso es ser capaces de "**Convivir con el enemigo**".

En este punto es donde claramente la RFC hace referencia al proceder ante incidentes, proponiendo dos estrategias:

- **Proteger y proceder.**
- **Seguir y perseguir**.

La primera de ellas es un curso de acción bajo el cual ante una intrusión, inmediatamente se procede a desconectar sistemas, apagar servidores, negar accesos, etc. Es decir se soluciona el problema actual pero no se puede llegar al fondo del mismo, no permite determinar las causas, ante lo cual cuando se vuelva a su régimen normal, existe una gran posibilidad que la intrusión se produzca nuevamente. Las ventajas que ofrece son que el intruso en ese momento no podrá avanzar más, y la información y recursos serán protegidos. Es una buena metodología a tener en cuenta si no se posee un alto grado de capacitación, soporte especializado, ni recursos suficientes.

La segunda metodología es más audaz, permitiendo llegar al origen de la vulnerabilidad, determinar las causas, los pasos que siguió el intruso, obtener toda la información probatoria, e inclusive hasta generar ataques inversos. Lo que es evidente aquí es que, como ya dijimos, se está "Jugando con fuego", es decir se debe tener un adecuado nivel de

conocimientos, herramientas, especialistas en apoyo, hasta soporte legal y de difusión de noticias.

Este es el punto clave pues para llevar a cabo la actividad de "Seguimiento de intrusiones" con un cierto grado de efectividad, se debe plantear una nueva línea de pensamiento para la planificación e implementación de los sistemas informáticos que oriente paso a paso al administrador de los mismos.

Hasta ahora hemos intentado desarrollar la teoría y práctica necesarias para poder implantar una infraestructura de Sistemas Informáticos segura. Si esto lo comparáramos con operaciones militares o policiales, podríamos pensar que en estos momentos tenemos una muy buena fortaleza, pero en la actualidad ¿es eso suficiente?

Desde hace varios cientos de años, toda organización militar cuenta con una especialidad que se denomina "Inteligencia militar" que es la responsable de mantener actualizada toda la información sobre el enemigo, esto es imprescindible hoy en día para saber a qué debemos atenernos, qué tácticas o estrategias emplea, que capacidades tiene, cuántos son, cuál es su origen, su intención, su meta, etc… Sin estos datos una defensa actual queda bastante limitada.

En el caso de los sistemas de información, por supuesto que existen muchos sitios desde donde obtener información "genérica" de estos intrusos, pero a cada administrador debe interesarle también la información "específica" de quién está intentando atacar a NUESTROS sistemas.

Al detectar a través de una alarma temprana o bloquear la presencia de intrusos, una medida activa de velo y engaño es la desviación hacia zonas de sacrificio. Este es el motivo de estudio de este punto.

7.16.2. ¿Qué es y cómo se implementa una honey pot?

Desde principios de los años 90 se están realizando varias pruebas para poder realizar el seguimiento de intrusiones y obtener información suficiente de las mismas para erradicarlas. El concepto que se ha impuesto de este conjunto de actividades es el de **Honeynet**, el cual abarca toda la topología de red, junto con el hardware y las medidas a adoptar para esta actividad. Existen varios laboratorios ya que tienen implementada esta metodología y comparten listas de discusión bajo esta denominación.

El punto clave de las mismas es lo que se desarrolla a continuación denominado Honeypot (o Jailing, o encarcelamiento).

Para implementar una zona de sacrificio (o Honey Pots), es necesario tener en cuenta los siguientes elementos:

- Equipo puente que monitorice todo el tráfico (por lo menos 3 interfaces de red).
- Equipo de control para limitar/bloquear el ancho de banda de los equipos víctima (único con salida a Internet), su misión será:

- Saturar el vínculo o generar colisiones.
- Reducir el ancho de banda.
- Modificar los "time out" de TCP.
- Alterar o eliminar paquetes para forzar el reenvío.
- Sobrecargar la capacidad de procesamiento de las víctimas (reducir los archivos de paginación, forzar el paginado a disco, abrir muchos procesos).

- Equipos trampa (o de sacrificio, o víctimas), los aspectos a tener en cuenta son:
  - puertos abiertos.
  - usuarios ficticios.
  - login y password fáciles de romper.
  - Sin parches de actualización.
  - Compartiendo información (Acorde al impacto de la zona).
  - Mala configuración de Logs.
- Equipos de generación de tráfico de usuario falsos o de información de bajo impacto (conexiones telnet, ftp, pop3, login, password, etc.).
- Equipo de resguardo de información.

Una vez implementada la infraestructura comienza la tarea de configuración de detalle, en la cual se debe tener en cuenta lo siguiente:

- Sincronización horaria de detalle de todo el sistema (puede ser un servidor NTP)
- Análisis de ataques.
- Creación de scripts para todo tipo de actividades de engaño, demora, derivación, enmascaramiento, spoof, etc.
- Empleo de Herramientas, como pueden ser:
  - Honeyd (que es la que más emplearemos en la sección "ejercicios").
  - Backofficer
  - TCT (The Coroners Toolkit)(Paquete de análisis forense).
  - TCTUTILs (Adiciona ventajas al anterior).
  - Tcpdump.
  - Snort.
  - Ethereal o Wireshark.
  - IPtables.
  - Dd (permite copias a nivel de bit de archivos o ficheros).
  - NetCat.

- Tripwire (integridad de archivos).
- Otras ya vistas en este libro…

- Herramientas que realizan comprobaciones automatizadas de vulnerabilidades, pudiendo algunas corregir de forma automática dichas vulnerabilidades como las que ya hemos tratado "nieto", "wikto", "nmap", "nessus", etc… O comerciales: Cybercop Scanner, ISS, Retina.
- Herramientas que capturan todo el tráfico visible en un segmento de red, siendo capaces de capturar las contraseñas de protocolos que no empleen esquemas de cifrado. Comerciales: Nai Sniffer. Gratuitos: Analyzer, Ethereal, Wireshark, etc...
- Password crackers: Herramientas que intentan obtener las contraseñas de acceso a un sistema mediante técnicas de fuerza bruta. Comerciales: Lopht Crack. Gratuitos: John the Ripper, Crack.

7.16.3. Metodología de trabajo.

La mejor forma de iniciar la tarea con Honey Pots es aislar esta infraestructura en un laboratorio, y comenzar a aprender el empleo de todas las herramientas mencionadas. De esta forma se empieza a familiarizar con los patrones de tráfico habituales que generan estos productos y las diferentes metodologías de respuesta que tienen configuradas el hardware y software de cada fabricante que se posee en el sistema a controlar. Esta tarea previa es de suma importancia, y lleva su tiempo, pues se debe asegurar que cada uno de esos pasos, son bien conocidos en la propia red para poder identificarlos sin lugar a dudas, cuando todo esto pase a producción.

Continuando con la fase de laboratorio, se debe seguir avanzando en las pruebas de vulnerabilidades reales del sistema en producción. En esta fase, se puede comenzar a atacar el sistema en producción (con todas las precauciones para evitar errores), y capturar las respuestas. Una vez capturadas, se replica la metodología en laboratorio y se procede a evaluarla con el sistema Honey pots aislado. Esta tarea pueda dar lugar a muchos cursos de acción, desde quitar o colocar parches hasta comenzar a jugar con la posibilidad de "ceder información", acción que se deberá realizar en algún momento. Se debe tomar todo el tiempo que haga falta para la realización de todas las pruebas necesarias, hasta poseer un alto grado de confiabilidad de lo que se está haciendo. Cuando se domine esta tarea, se pueda iniciar la implantación del sistema de sacrificio pero sólo en la periferia, es decir en lo que podríamos denominar como "primera línea de retardo", es decir, una zona en la cual tenemos posibilidades de "ralentizar" y "observar" la actividad del intruso.

Aquí comienza el verdadero ciclo de trabajo, pues por tratarse de una zona con enorme grado de exposición, la actividad de intrusiones es la más alta y por lo tanto la que más experiencia aportará. La gran ventaja que se posee al comenzar aquí es que el grado de impacto que se posee es mínimo, y por lo tanto en esta etapa aún de aprendizaje, cualquier error que se cometa (si bien se deben minimizar, pues ya se posee experiencia de todo el tiempo de trabajo en laboratorio), no debería causar mayores problemas.

A medida que se vaya aprendiendo el funcionamiento de cada zona y la táctica de los intrusos, se puede ir avanzando hacia el interior de la red, pero tratando de mantener en todo momento la metodología de trabajo laboratorio-producción, es decir, realizando todas las pruebas necesarias en un entorno aislado y seguro, y una vez dominado el tema allí ir volcando las experiencias a producción.

El comportamiento habitual de intrusos que se detecta a través de Honey Pots es el que se presenta a continuación:

a. Escaneo de puertos.
b. Finger printing (con ICMP, TCP, UDP o IP).
c. Escaneo de vulnerabilidades conocidas.
d. Empleo de exploits o troyanos para abrir puertas traseras.
e. Instalación de rootkit (conjunto de programas de nombre y características similares a los del sistema operativo, pero con modificaciones que facilitan el acceso al intruso. Estos programas suelen ser muy difíciles de detectar si no se pueden comparan con su versión original.
f. Instalación de herramientas que le permitan atacar otros equipos de la red desde la máquina infectada.
g. Borrado de huellas en los archivos de registro.

Se presentan a continuación dos tablas obtenidas en segmentos diferentes de red (Internet e Intranet) para que puedas notar la diferencia en la cantidad de actividad analizada entre una y otra. Todas estas capturas son reales y tomadas de una importante red LAN conectada a Internet y muy conocida.

TABLA 1: Ataques producidos en un día (Internet)

Prior	Nº	nombre del ataque	Ataques	
Prioridad ALTA	1	NETBIOS DCERPC ISystemActivator bind attempt	1501	
	2	POLICY PPTP Start Control Request attempt	392	
	3	WEB-MISC Lotus Notes .exe script source download attempt	126	
	4	WEB-MISC perl post attempt	64	
	5	WEB-IIS cmd.exe access	46	
	6	MULTIMEDIA Windows Media Video download	23	
	7	WEB-FRONTPAGE fourdots request	22	
	8	WEB-IIS asp-dot attempt	19	
	9	FINGER remote command ; execution attempt	7	totales ALTA:
	10	WEB-IIS ISAPI .ida attempt	7	**2207**

Prioridad MEDIA	1	BAD-TRAFFIC loopback traffic	139674	
	2	DDOS shaft synflood	20905	
	3	FINGER version Quero	18297	
	4	FINGER . Quero	18197	
	5	MISC source port 53 to <1024	13670	
	6	SNMP public access udp	7499	
	7	ICMP webtrends scanner	4634	
	8	SCAN SYN FIN	1346	
	9	WEB-FRONTPAGE /_vti_bin/ access	94	totales MEDIA:
	10	WEB-IIS nsiislog.dll access	64	**224380**
Prioridad BAJA	1	ICMP Destination Unreachable (Communication with Destination Network is Administratively Prohibited)	2124	
	2	POLICY FTP anonymous login attempt	321	
	3	POLICY SMTP relaying denied	293	
	4	POLICY poll.gotomypc.com access	176	
	5	MISC MS Terminal server request (RDP)	67	
	6	ICMP PING BSDtype	38	
	7	MISC MS Terminal server request	31	
	8	INFO FTP No Password	24	
	9	POLICY VNC server response	24	totales BAJA:
	10	POLICY PCAnywhere server response	12	**3110**

TABLA 2: Ataques producidos en un día (Intranet)

Prior	Nº	nombre del ataque	Ataques	
Prioridad ALTA	1	WEB-IIS cmd.exe access	11	
	2	WEB-FRONTPAGE fourdots request	9	
	3	WEB-IIS ISAPI .ida attempt	5	
	4	WEB-IIS WEBDAV nessus safe scan attempt	5	
	5	WEB-MISC Cisco /%% DOS attempt	4	
	6	WEB-ATTACKS cc command attempt	1	
	7	WEB-ATTACKS rm command attempt	1	
	8	WEB-IIS asp-dot attempt	1	totales ALTA:
	9	WEB-MISC cross site scripting attempt	1	**38**
Prioridad MEDIA	1	SCAN nmap TCP	96	
	2	WEB-FRONTPAGE /_vti_bin/ access	50	
	3	WEB-IIS nsiislog.dll access	40	
	4	ATTACK-RESPONSES 403 Forbidden	10	
	5	WEB-IIS ISAPI .printer access	5	

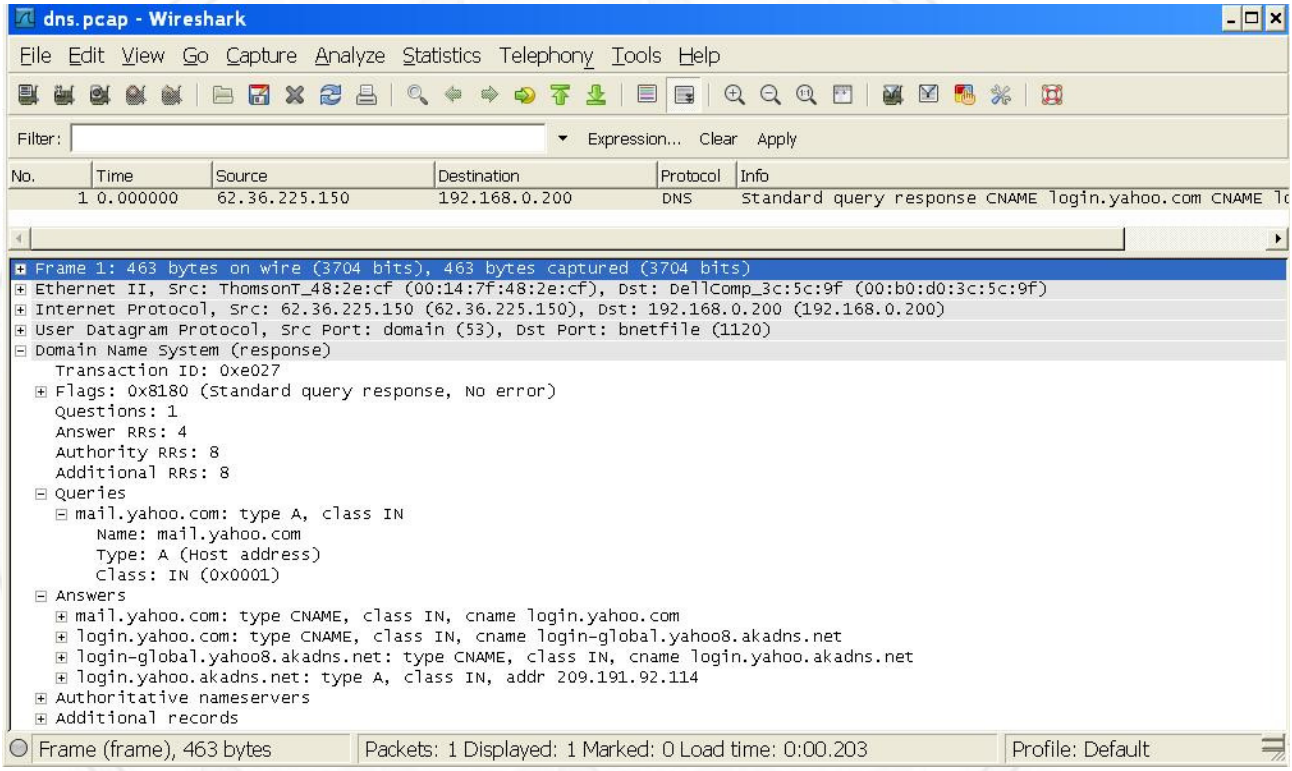
	6	WEB-MISC sadmind worm access	5	
	7	SCAN myscan	3	
	8	WEB-MISC /.... access	3	
	9	WEB-MISC cat%20 access	3	totales MEDIA:
	10	WEB-MISC ultraboard access	1	**216**
Prior BA-JA	BAD-TRAFFIC tcp port 0 traffic		**13**	

Como has podido apreciar en las tablas, la actividad cambia substancialmente en una red interna, respecto a otra que se encuentra expuesta a Internet.

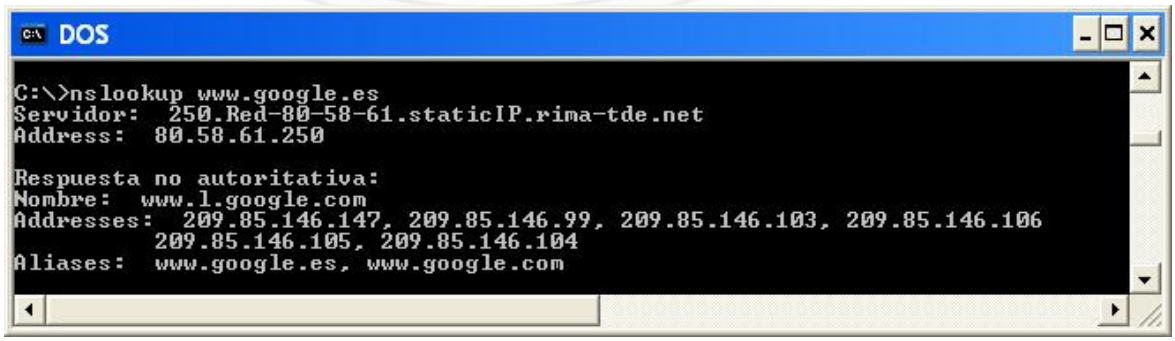
EJERCICIOS DEL CAPÍTULO 7 (Nivel de Aplicación)

1. **Ejercicios con DNS**.

   1) Lo primero que haremos, siguiendo con nuestra metodología de análisis de tráfico será que captures tráfico DNS, para ello basta con que una vez lanzado Wireshark, abras cualquier navegador hacia una URL concreta. Deberás haber capturado tráfico DNS y su formato será similar a la imagen que te presentamos a continuación. Compara el dato obtenido en la práctica con lo que hemos tratado en la teoría. Investiga su encabezado, tanto en solicitud como en respuesta.



   2) En la imagen de abajo, acabamos de ejecutar desde una consola MSDOS ">nslookup www.google.es, ¿Porqué nos devuelve seis direcciones IP?



Ejecuta el mismo comando y captúralo con Wireshark, ¿Es una sola respuesta o varias?, ¿Cómo es el encabezado de la, o las respuestas?

3) Averigua la dirección IP de alguna página Web, agrega una línea en los archivos "hosts" de Linux y/o Windows, tal cual lo tratamos en la teoría, con esta dirección IP y el nombre www.pp.com. Lanza Wireshark y comienza haciendo "ping www.pp.com" ¿A qué dirección IP hace el ping?, ¿Has capturado tráfico DNS con Wireshark? Ahora abre un navegador y coloca la URL "www.pp.com", ¿Qué página Web te abre?

4) Desde una máquina Linux y desde otra Windows, practica e investiga las diferentes opciones que ofrecen para obtener información DNS.

5) Consulta diferentes bases de datos "Whois" en Internet, averigua información sobre dominios que conozcas, presta atención a los datos que figuran en ellos: Nombres, correos, responsables, teléfonos, direcciones, etc... Este es un muy buen punto de partida para "fingerprinting".

2. Ejercicios con Telnet.

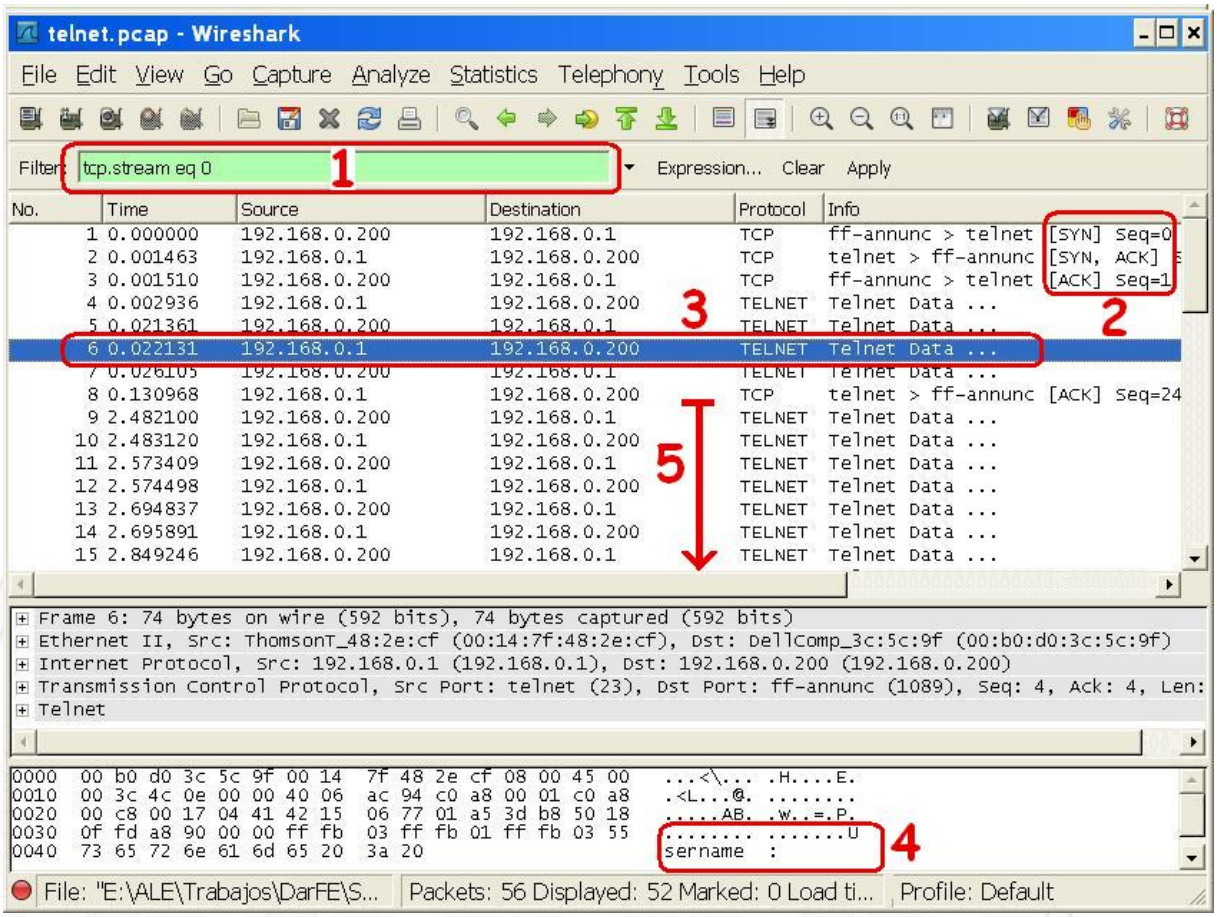
1) En el ejercicio con herramientas "5.Empleo de la herramienta **iptables**" del capítulo anterior (nivel de Transporte), te propusimos que instales el demonio "**telnetd**" para verificar la apertura y cierre del puerto 23 con el FW "iptables". Si has cumplido con esta ejercitación, ya debes tener funcionando nuestro servidor "telnet".

   En este primer ejercicio te proponemos que una vez más lances Wireshark y desde otro host establezcas una conexión telnet hacia este servidor.

   El comando es sencillamente: "telnet direccion_IP_Servidor", desde cualquier consola (Windows o Linux), recuerda que debes tener en escucha "Wireshark".

   Al iniciar esta conexión te pedirá "usuario y contraseña", debes poner el de cualquier usuario de ese Linux. Si todo está en orden, te habilitará un "prompt >", que indica que estás conectado a ese servidor. Para este ejercicio, sólo escribe "exit" y presiona [Enter], con lo cual te habrás desconectado.

   Nuestro objetivo es que logres verificar que tanto el usuario como la contraseña viajaron en texto plano, tal cual te presentamos en las siguientes imágenes:



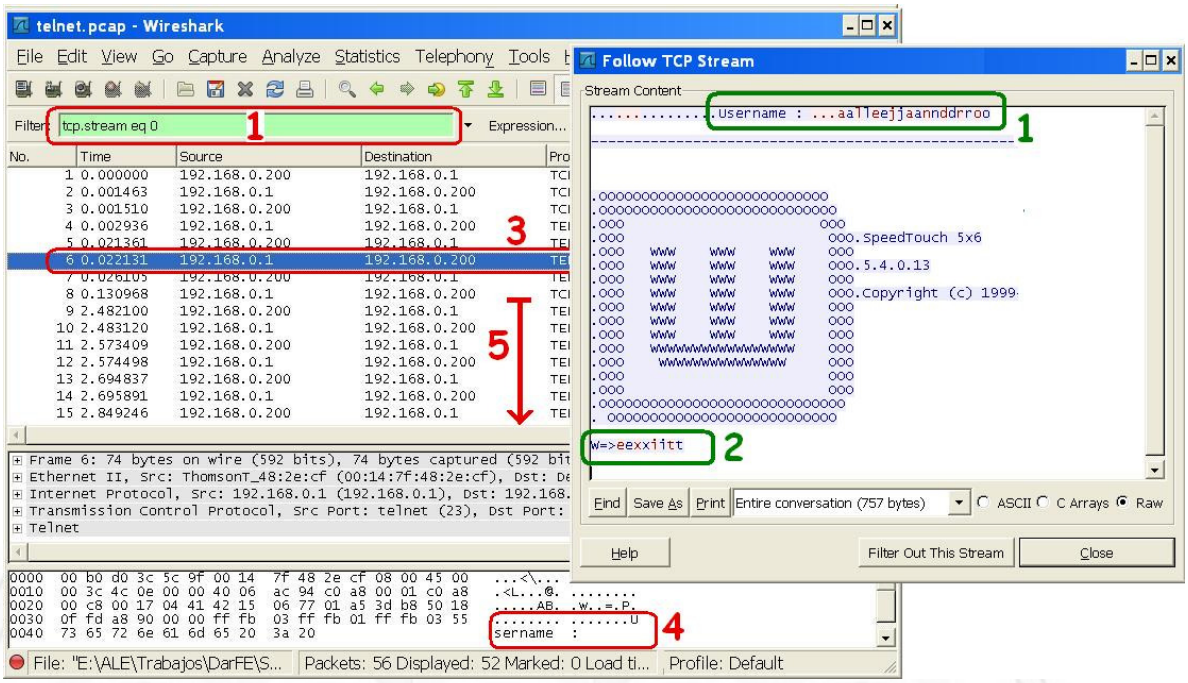
Lo primero que debes observar es que hemos destacado en rojo con el *número 1*, que estamos siguiendo esta secuencia tcp (ya lo hemos visto, seleccionando cualquier trama de esta secuencia y desde el menú "Analyze" → "Follow TCP stream").

En el *número 2* (también en rojo), puedes ver los tres segmentos TCP del "triple Handshake". A partir de este, se abrió el puerto 23 y puedes ver que en la columna "protocolo" ya se aprecia "TELNET", pues acabamos de abrir esta puerta.

En el *número 3* hemos destacado la trama número 6, a partir de la cual comienza el diálogo de "Autenticación", el cual como puedes ver en el *número 4* (en rojo y debajo de la imagen), el servidor nos está pidiendo que ingresemos el "Username" **en texto plano** (por esa razón es que lo estamos viendo en la captura).

A partir de allí, como indicamos con el *número 5*, comienza a pasarse "carácter a carácter" nuestro nombre de usuario en cada una de las tramas que indicamos con la flecha roja.

Como hemos seguido este flujo TCP, Wireshark nos ha abierto otra ventana que es la que pegamos a continuación:

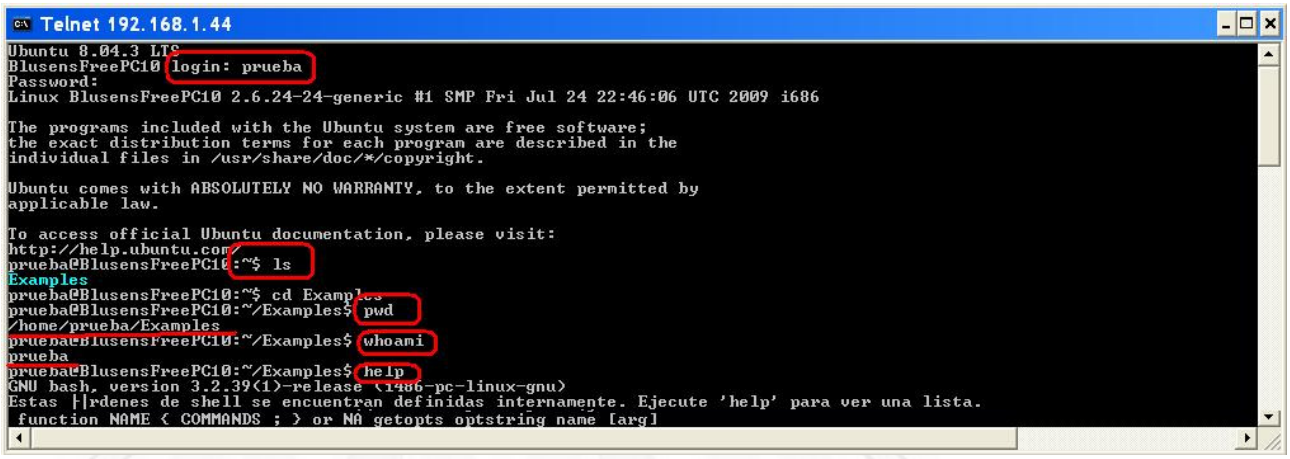


En esta segunda imagen, destacamos en <u>verde</u> con el *<u>número 1</u>*, el nombre de usuario que viajó (carácter a carácter) en cada una de las tramas. Es importante destacar, que como terminal remota que es "telnet", cada uno de estos caracteres, se ejecutaba concretamente en el servidor, y este nos hacía un "**echo**" (eco) hacia nuestro cliente "telnet", por esa razón cada carácter que viajó se ve por duplicado, pues el cliente, en este caso por ejemplo, escribió como <u>usuario "**alejandro**"</u>, por lo tanto la primer trama fue con el caracter "**a**", y el servidor le respondió con el "echo" de "**a**", luego el cliente, escribió la letra "**l**", y el servidor le respondió con el "echo" de "**l**"... y así hasta completar la última letra, luego de la cual se presionó [Enter].

Como te propusimos al principio, este ejercicio, sólo consistía en verificar la conexión y luego desconectarse, por lo tanto como puedes ver, hemos remarcado con el *número 2*, el comando "exit" que te dijimos.

Te proponemos que hagas todo esto por ti mismo, pero esto no termina aquí, pues si has prestado atención, en estos momentos te estarás preguntando ¿Y la password?, pues justamente es lo que no te hemos mostrado ni comentado intencionalmente en las imágenes anteriores... ¿te atreves a investigar, capturar y analizar cómo viaja la misma?

2) Continuando con el demonio "telnetd" que tenemos instalado, ahora te proponemos que analices y navegues un poco por los comandos desde una consola. Para ello nuevamente desde la otra máquina realiza la conexión telnet hacia la que tienes "telnetd" instalado (debes tener una cuenta de usuario, así que ni no la tienes deberás crearla... recuerda "adduser") y prueba la ejecución de algunos comandos, tal cual te presentamos en la siguiente imagen:



En el host donde está el servidor "telnetd" hemos creado un usuario llamado "prueba". En nuestro caso realizamos la conexión desde una consola MSDOS (como puedes ver en el ángulo superior izquierdo de la imagen "C:\". En la imagen (y remarcado en rojo), el usuario "prueba" es con el que se realiza la conexión. A partir de allí, a pesar que nuestra máquina origen es Windows, puedes ver que los comandos que estamos ejecutando son comandos Linux, en la imagen se aprecia "ls", "pwd" y "whoami".

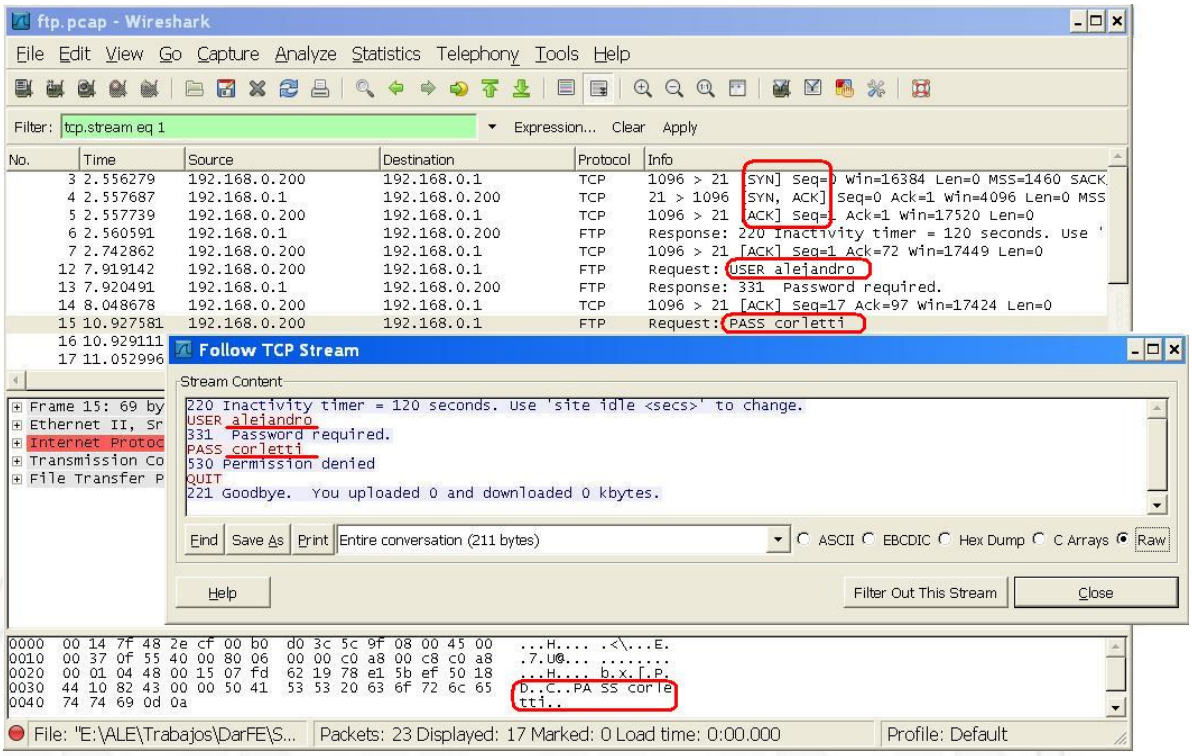
Te proponemos que ejecutes tú también "help" y pruebes diferentes opciones del listado de comandos.

3) Lo último que te proponemos es que investigues y pruebes diferentes opciones de configuración del servidor "telnetd", todas ellas están muy claras en su manual ("man telnetd").

3. Ejercicios con FTP.

1) Primero veamos con Wireshark una captura de establecimiento de sesión FTP para ver la apertura del puerto y comprobar la transferencia del "usuario" y "contraseña" en texto plano.

   A continuación, te pegamos un ejemplo de captura:

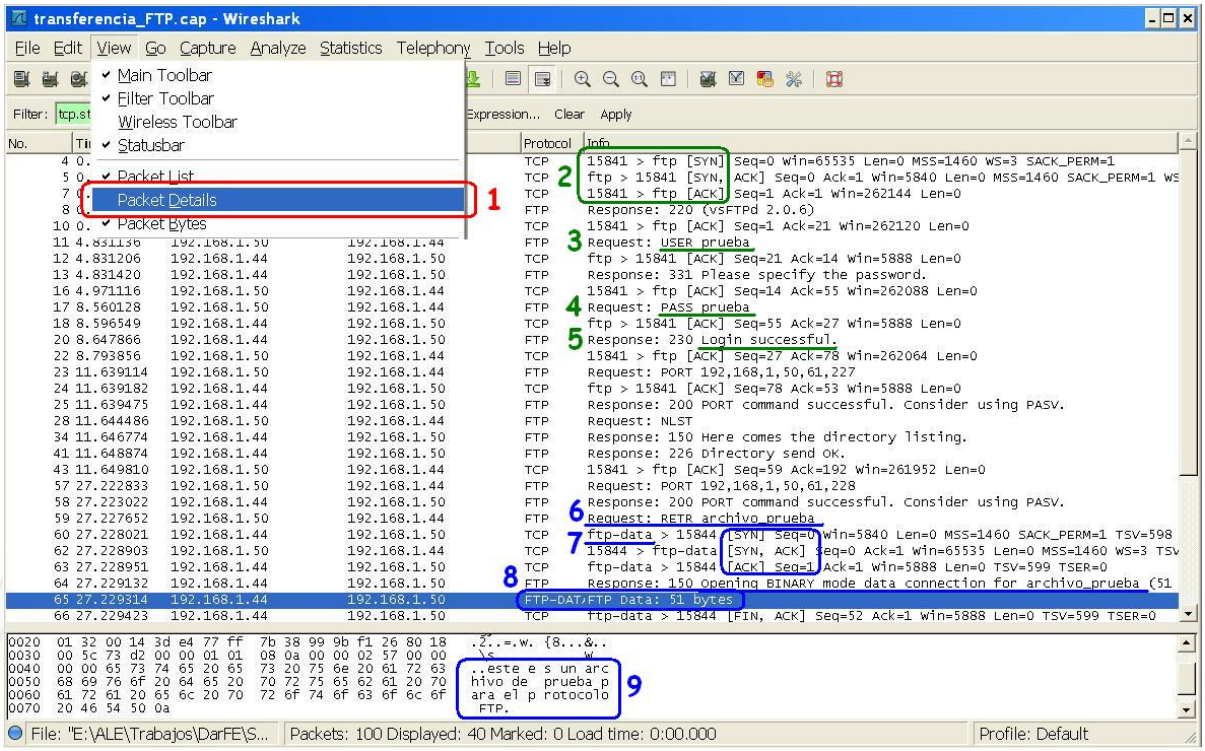


Como puedes apreciar, al principio remarcamos en rojo el "triple handshake" para abrir el puerto TCP 21, una vez abierto, para la validación FTP, el servidor requiere "USER" y el cliente en ese campo escribe "alejandro" a continuación es requerida la "PASS", donde el cliente escribe "corletti", pudiendo verificar en la captura que ambos viajan en texto plano, a su vez hemos hecho un seguimiento de sesión TCP que aparece en primer plano en otra ventana, donde también se aprecian estos datos.

2) En esta otra captura os proponemos analices la secuencia de apertura de puertos de una conexión FTP "activa". Como siempre te presentamos una imagen de referencia para que puedas comparar, pero el trabajo te invitamos a que lo hagas tú y puedas verificar:

- En qué momento se abre el otro puerto.
- Evalúa con máximo detalle el "Sentido" de ambas conexiones.
- Cuál es el número de los puertos origen y destino de ambas conexiones.
- ¿Es pasiva o activa?
- ¿En qué momento se define si es activa o pasiva?
- Cómo se realiza el cierre de la conexión.

A continuación te presentamos la imagen:



En nuestra imagen te presentamos varias cosas. Lo primero que puedes notar es que arriba y a la izquierda, hemos remarcado en rojo con el *número 1* "Packet Details", esa parte del menú "View" está compuesta por las tres ventanas que presenta "Wireshark" (arriba: lista de paquetes, al centro: detalle de paquetes y abajo: Bytes de los paquetes), te hemos remarcado esto, pues si prestas atención está "desmarcado Packet Details" por esa razón no se ve la ventana central, quedando en la imagen únicamente la parte superior (Listado) y la inferior (Bytes) que son las que queríamos mostrar.

Pasando ya a la captura, hemos remarcado con verde lo que ya conoces, con el *número 2* el triple Handshake hacia el puerto 21, con los *números 3 y 4* cómo se ven en texto plano el usuario y la contraseña, y por último con el *número 5* cómo nos indica que nos hemos validado con éxito (login successful).

Por último hemos remarcado en azul, lo que nos interesa en este ejercicio que es analizar la transferencia de un archivo por FTP. Con el *número 6*, puedes ver el momento en el que el cliente (Dirección IP 192.168.1.50), solicita la transferencia del "archivo_prueba", lo importante aquí es que el cliente escribió "get", pero el COMANDO que viajó es RETR, si repasas la teoría verás en el punto 7.3.4. Comandos:

RETR <filename>	Retrieve (copy) file from server.

Lo que nos interesar destacar aquí es cómo trabaja este protocolo, una cosa es lo que tú escribes y otra es el intérprete de comandos, que lo traduce a lo que el protocolo entiende entre cliente y servidor (Comandos y Mensajes).

Hasta ahora si prestáis atención, únicamente estaba abierto el puerto TCP 21, pero a partir del momento en que el cliente ejecutó un comando relacionado con "**Datos**" se inició el triple handshake para abrir el puerto TCP 20 (ftp-data) como puedes ver en el *número 7*.

Con el *número 8*, remarcamos la trama en la cual el servidor (Dirección IP 192.168.1.44) nos indica que transfiere 51 Byte de contenido de este archivo, el cuál con el *número 9* (en la ventana Byte de abajo) puedes ver la totalidad del contenido en texto plano de este archivo que está viajando por la red: "**este es un archivo de prueba para el protocolo FTP**".

3) Continuando con nuestra práctica de FTP, en esta parte vamos a trabajar con el servidor "**VSFTP**" y el cliente nativo de Linux o Windows, a través de línea de comandos.

Configuración y ajuste de vsftp:

Una vez instalado (no deberías tener ninguna dificultad para hacerlo), el demonio se lanza de forma automática, es decir ya te ha quedado el **puerto 21** abierto. El primer detalle que se debe considerar es cómo configurarlo, todo aspecto de su configuración se puede realizar desde el archivo: "`sftpd.conf`" (se suele encontrar en "*/etc/*")

El cual si lo editas verás algo similar a lo que presentamos a continuación:

```
/etc# vi sftpd.conf

# Example config file /etc/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# daemon started from an initscript.
. . . . .
. . . .
listen=YES
#
# Run standalone with IPv6?
# Allow anonymous FTP? (Beware - allowed by default if you comment this
out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES   ← (Esta línea la debes "des comentar", es decir quitarle "#", para
                      poder acceder con cualquier cuenta de usuario local).
```

Una vez que hayas modificado este archivo (Recuerda guardarlo) debes reiniciar el demonio, pues recuerda que se está ejecutando, y se inició con el archivo "vsftpd.conf" anterior

para ello las instrucciones son:

```
/etc# cd init.d/
etc/init.d# ./vsftpd restart
```

Podrás verificar su correcto reinicio, pues te deberían aparecer las siguientes líneas en tu consola:

```
* Stopping FTP server: vsftpd
[ OK ]
 * Starting FTP server:
vsftpd                                          [ OK ]
```

- Para verificar el funcionamiento de nuestros puertos TCP, vamos a lanzar "**nmap localhost**" y analizar en qué situación está el puerto 21.
- Ejecutar ahora: "**./vsftpd stop**"
- lanzar "**nmap localhost**" nuevamente y verificar en qué situación está el puerto 21.
- ejecutar: "**./vsftpd start**" y verificar nuevamente.
- ¿Qué diferencias encuentra entre restart y reload?

4) Ahora os proponemos aanalizar y comprobar con más detalle el archivo /etc/vsftp.conf:
   - Configurar otro banner desde (ftpd_banner=Bienvenido.....(Lo que queráis)
   - Probar, habilitar y deshabilitar el acceso anónimo (anonymous_enable=YES/NO)
   - Combinar la opción anterior con: local_enable=YES/NO (para verificar el acceso o no de usuarios locales) ¿Qué combinaciones puedes comprobar? ¿Qué sucede en cada una de ellas?
   - Verificar las opciones de escritura con: write_enable=YES/NO, ¿Qué sucede?
5) Vamos a seguir avanzando con este servidor, ahora transfiriendo archivos, para ello vamos a hacer lo siguiente:
   - Generar un archivo de texto plano en el servidor de archivos.
   - Crear más usuarios (para conectarse por ftp).
   - Realizar conexiones ftp y monitorizar qué puertos trabajan.
   - Desde el analizador de protocolos, verificar cómo es la apertura y mantenimiento de puertos de control y de datos.
   - Transferir el archivo creado de texto plano y capturarlo.
   - Analizar especialmente los comandos y los códigos de control (Comparar lo capturado con la teoría).
   - Verificar en la captura cómo se ve en texto plano el usuario y contraseña.
   - Verificar el contenido del archivo (debe verse textualmente todo su contenido)
   - Verificar el establecimiento y cierre del puerto de datos, describir el detalle de pasivo o activo.
6) Sigamos más aún con esta práctica, ahora probando lo que en la teoría desarrollamos como "**ftp pasivo**".
   - Verificar las opciones de "**pasv_enable**" en el servidor (vsftpd.conf).
   - Desde el lado cliente verificar el funcionamiento de "**ftp -p**" ¿Para qué sirve esta opción?
   - Transferir un archivo y capturar este tráfico, ¿Qué diferencia puedes apreciar?

- Verificar las opciones de "**pasv_max_port**" y "**pasv_min_port**" en el servidor (vsftpd.conf).
- Transferir un archivo y capturar este tráfico, ¿Qué diferencia puedes apreciar?

7) Otro de los temas peligrosos de "ftp" es la posibilidad que un usuario tiene para poder salirse del directorio en el que se conectó inicialmente, para evitar ello en este punto analizaremos qué podemos hacer a través de las siguientes tareas:

- Analizar y comprobar el funcionamiento de "**Jailing**" (enjaulamiento) desde: "**/etc/vsftpd.user_list**" (*creando este archivo*).

El concepto de "**Jailing**" es la posibilidad que ofrece **vsftp** para que los diferentes usuarios, puedan navegar por la estructura de directorios o que queden confinados (enjaulados) a uno en particular.

- Los parámetros que entran en juego para esto son:

  * **chroot_local_user** (habilitará la función de chroot())

  * **chroot_list_enable** y **chroot_list_file** (establecen el fichero con la lista de usuarios que quedarán excluidos de la función chroot()).

- Probar con las diferentes combinaciones mencionadas, crear varios usuarios y "enjaularlos" en diferentes directorios, etc.

  (Ej: chroot_local_user=YES
  chroot_list_enable=YES
  chroot_list_file=/etc/vsftpd/vsftpd.chroot_list *(se deberá crear este archivo).*

- Analizar y probar: "**userlist_enable**".
- Analizar y probar: "**userlist_file**".
- Comprobar las opciones para permisos de archivos con: "**local_umask=nnn**" (nnn = valores de permisos unix)
- Probar la creación de usuarios con las opciones "**-d**" (directorio), (Ej: useradd -d /home/Nombre_usuario).

8) Como desafía de "vsftpd", y relacionado con el próximo tema, te invitamos a que investigues qué opciones seguras ofrece este servidor (una pista "ftps:..").

4. Ejercicios con SSH.

Como dijimos en la teoría SSH también responde a una arquitectura "Cliente – Servidor", es decir que necesitamos contar con ambas partes para poder trabajar con este protocolo.

Para la parte cliente, desde Linux su empleo es muy sencillo pues ya viene instalado, sólo debes ejecutar:

"# ssh IP_Servidor_remoto (o nombre del servidor)"

Si no se especifica nada más, por defecto esta instrucción abre una conexión con la cuenta de cliente correspondiente al usuario de Linux con el que estemos conectados en este momento en la máquina local, pero si se desea establecer la conexión con otra cuenta de usuario el comando debería ser:

"# ssh usuario_remoto@IP_Servidor remoto (o nombre del servidor)"

Al establecerse la conexión, nos pedirá la contraseña para ese usuario, y si es la correcta, veremos una consola remota correspondiente al perfil del usuario que se acaba de validar. Si es la primera vez que nos conectamos a ese servidor, es muy probable que nos presente una ventana de aceptación de la clave que posee el mismo (ya lo veremos en criptografía), pero una vez aceptado el mismo, a partir de allí nunca más volverá a aparecer la misma. A partir de ese momento estamos operando físicamente en el host remoto, es decir que podemos realizar cualquier actividad que ese usuario tenga autorizada sobre esa shell.

Siguiendo con Linux, dentro de las opciones gráficas, la más sencilla es directamente, por ejemplo en Ubuntu, ir a "**Lugares**" → "**Conectar con el servidor**", y desde allí se nos abrirá una interfaz gráfica muy sencilla que nos ofrece varios protocolos para esta conexión, uno de ellos es SSH, nos pide también la configuración del puerto, que para este protocolo por defecto es el TCP 22.
En el caso de Windows, deberás instalar algún software cliente para realizar esta conexión, el cual lo presentamos en la parte de "Herramientas" de estos ejercicios.

1) Servidor SSH:

   Para trabajar con este protocolo, como mencionamos en la teoría, emplearemos "**OpenSSH**", en el caso de distribuciones Debian su instalación es sencillamente: "**apt-get install openssh-server**".
   Por defecto se instala en el directorio: "**/etc/ssh**", y el archivo que más nos interesa es: "**/etc/ssh/sshd_config**".

   Nuestro primer paso será habilitar la conexión a los usuarios que deseemos (los cuales ya deben estar dados de alta en Linux "adduser") para ello, en cualquier lugar del archivo "**sshd_config**" se deberá añadir la línea:

   "**AllowUsers usuario1 usuario2 usuarion'**" (separados por espacio, todos los usuarios que se desee).

En nuestro caso lo hemos añadido al final, abajo te pegamos un ejemplo de nuestro archivo "**sshd_config**":

```
# This is the ssh client system-wide configuration file.  See
# ssh_config(5) for more information.  This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.
. . . . .
. . .
#   Tunnel no
#   TunnelDevice any:any
#   PermitLocalCommand no
    SendEnv LANG LC_*
    HashKnownHosts yes
    GSSAPIAuthentication yes
    GSSAPIDelegateCredentials no
AllowUsers prueba       ← (Esta es la línea que hemos agregado).
```

Una vez instalado **"Openssh**, por defecto queda iniciado (y el puerto 22 abierto), por lo tanto cada vez que se haga un cambio en su configuración deberá lanzarse nuevamente este proceso, para ello se ejecuta desde:

"**/etc/ini.t/./ssh restart**" (con permisos de root)

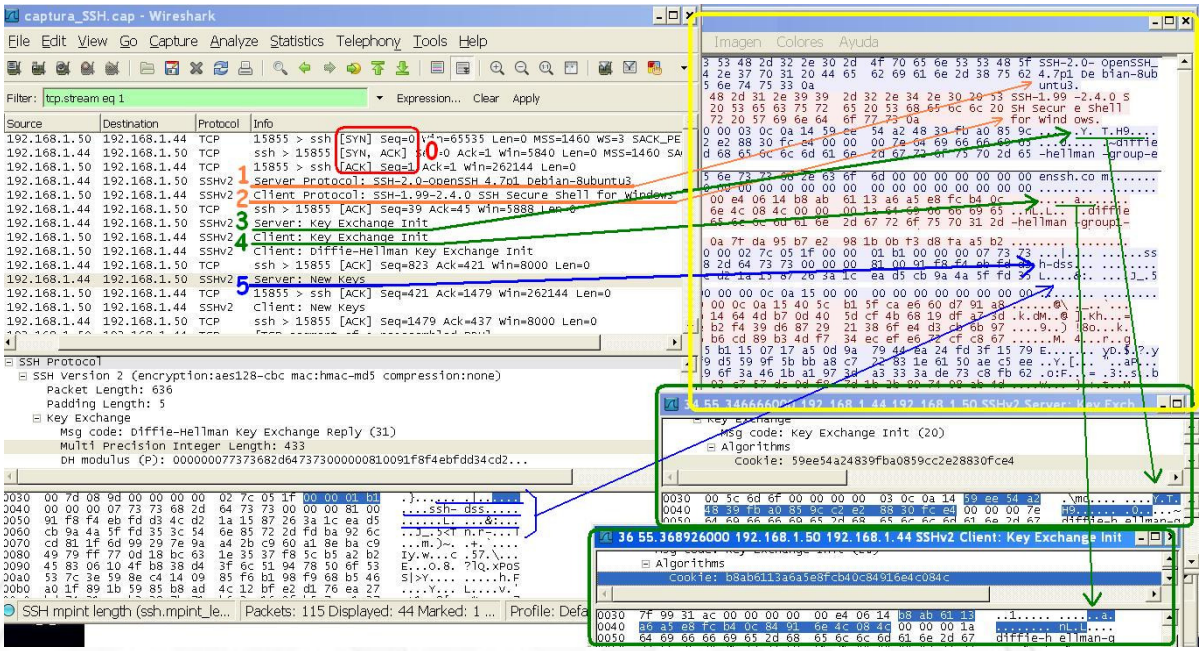
Nos aparecerá el mensaje:

*** Restarting OpenBSD Secure Shell server...**

Y si todo está en orden al final se verá **[OK]**

A partir de este momento, cualquiera de los usuarios que acabamos de incluir en la línea "**AllowUsers**".

Por defecto, una vez que se establece el triple "Handshake TCP" desde el cliente al puerto TCP 22 del servidor, lo primero que sucede es el inicio del intercambio de claves según el algoritmo de **Diffie Hellman** (verlo en la parte de criptografía), por medio del cual como mencionamos en la teoría, se logra establecer un secreto compartido en una red no segura, y por medio de este se establece la contraseña de **esa sesión**, sin que nadie ajeno a estos dos extremos haya podido "escuchar" ni un sólo bit en texto plano. Las únicas tramas que llegará a ver son las 2 primeras de este intercambio, en las cuales sencillamente se "presentan" uno al otro, luego las dos siguientes en las cuales cada uno indica las diferentes opciones de criptografía y versiones que ambos pueden soportar, pero en ningún momento quedará visible dato alguno que pueda hacer perder la confidencialidad de esta sesión, a partir de este momento todo viajará cifrado. Esto lo ponemos de manifiesto en la imagen que presentamos a continuación, pero como siempre te invitamos a que lo puedas hacer tú mismo:



Lo primero que hemos remarcado en rojo con el *número "0",* es el triple Handshake hacia el puerto TCP22, inmediatamente del mismo puedes ver en los *números 1 y 2* de color naranja, la "presentación" entre "servidor" (SSH-2.0-OpenSSH_4.71 p1 Debian....) y el "cliente" (SSH1.99-2.4.0 SSH Secure Shell for Windows....), estas dos tramas son muy pequeñas y son una mera presentación de las versiones de SSH de ambos hosts.

En las tramas *número 3 y 4* de color verde. como puedes apreciar, se inicia el intercambio de claves, estas sí ya son tramas extensas pues en ella viajan todos los posibles algoritmos que se podrían implementar entre ambos, y lo más importante ya viajan los parámetros para generar la clave de sesión, la cual la hemos subrayado en verde y a su vez de cada una de ellas hemos presentado en la parte inferior derecha la primera parte del contenido del protocolo SSH de las mismas. Sería muy importante que las generes tú y te detengas sobre estas dos, para analizar cómo se inicia este intercambio y puedas confrontar la teoría de lo que expusimos sobre Diffie-Hellman con lo que "viaja" en estas tramas.

Por último hemos remarcado en azul con el *número 5*, la trama en la cual desde el servidor, ya viaja ese secreto compartido para la sesión ("**Server: New Keys**"), el cual lo puedes ver pues ya figura como "**ssh-dss**" (Digital Standard Signature), pues lo está firmando. Más abajo no lo hemos marcado, pero puedes ver también, la aceptación por parte del cliente: "**Client: New Keys**".

No dejes pasar la oportunidad de generar tu propia captura sobre este tráfico y analizarlo, pues hemos intentado resumirlo todo lo posible en la imagen anterior para que lo comprendas, pero el contenido completo de cada una de esas tramas, estamos seguros que te será de mucho interés.

2) Para trabajar con este protocolo y el servidor OpenSSH, te proponemos que realices los siguiente ejercicios:

- Configurar varias cuentas de usuario en el servidor SSH.
- Generar un archivo de texto plano en el servidor de archivos.
- Conectarse por SSH.
- Monitorizar qué puertos trabajan.
- Desde el analizador de protocolos, verificar cómo es la apertura y mantenimiento de puertos y el intercambio Diffie-Hellman. ¿Se llega a visualizar algún dato del usuario (nombre passw, etc...)?
- Transferir el archivo creado de texto plano y capturarlo.
- Analizar especialmente los datos que están viajando.
- Verificar en la captura cómo se ve diferente que en FTP.

5. Ejercicios con SMTP y POP3.

1) Como siempre, primero veamos el funcionamiento a través de una captura. Os recordamos que el protocolo SMTP, se emplea para enviar correo hacia un servidor smtp, por lo tanto debemos estar en capacidad de capturar tráfico hacia ese servidor, en nuestro caso lo haremos desde una consola Linux, con los comandos que pegamos a continuación. A su vez en ese mismo host teníamos lanzado Wireshark para capturar el tráfico generado, así que te pegamos abajo también la captura para que la tomes como referencia, y una vez más te reiteramos que no te quedes con esto solamente, sino que tú mismo realices esta actividad para obtener también tu propia visión del trabajo.

   Los comandos en nuestra shell de Linux fueron:

```
root@BlusensFreePC10:/etc# telnet 127.0.0.1 25
      Trying 127.0.0.1...
      Connected to 127.0.0.1.
      Escape character is '^]'.
      220 BlusensFreePC10 ESMTP Postfix (Ubuntu)
mail from: acorletti@hotmail.com
      250 2.1.0 Ok
rcpt to: acorletti@darfe.es
      250 2.1.5 Ok
data
      354 End data with <CR><LF>.<CR><LF>
esta es una prueba de SMTP
.
      250 2.0.0 Ok: queued as 1713B2A40D4
Subject: prueba
.
quit
      Connection closed by foreign host.
```

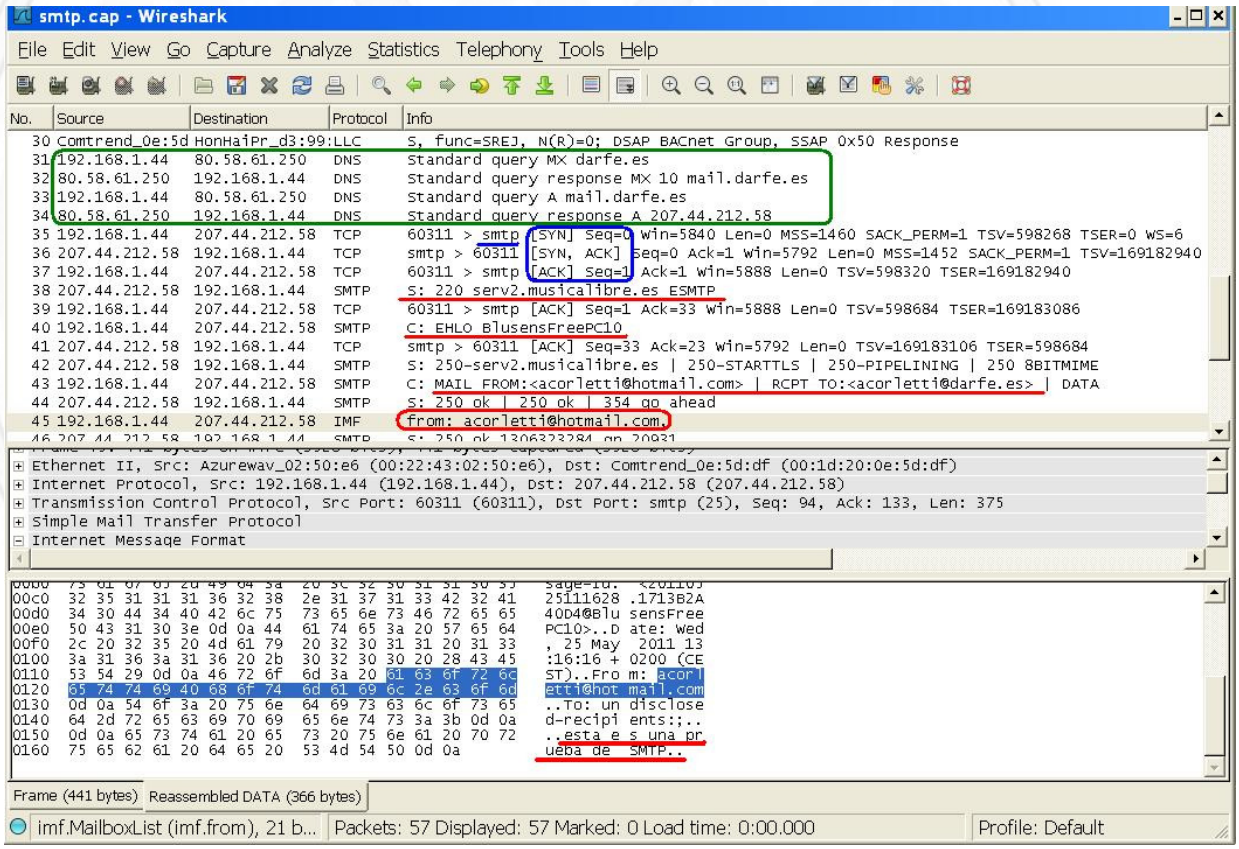
   Hemos “tabulado” las respuestas de la consola para que veas con más detalle los “Comandos” introducidos por el usuario (sin tabular) y las respuestas de la consola, pero en realidad están todos en la misma columna.

Lo que puedes notar del diálogo anterior es que primero hicimos una conexión "**telnet**" a nuestra dirección IP de "loopback" y al puerto **TCP 25** (SMTP), también podríamos haber escrito "telnet localhost 25" que es equivalente. Una vez conectados el Servidor nos respondió, tal cual vimos en la teoría el **código 220** (Quería decir: "*Service ready*"), a través del comando "**mail from:"** pusimos el correo del EMISOR del mismo, aquí debemos prestar atención también a otro detalle visto en la teoría pues si SMTP soporta las extensiones de servicio, responde con un mensaje multi-línea **250 OK,** que como en esta caso nos indica, incluye una lista de las extensiones de servicio que soporta "**2.1.5**".

A continuación escribimos "**rcpt to:**" hacia quien lo dirigimos, luego "**data**:" con el contenido del mensaje (el cual se cierra cuando colocamos el "."), el servidor replica con *"354 Start mail input, end with <CRLF>.<CRLF>"* (también tratado en la teoría).

Para finalizar pusimos un "**Subject:**" y para salir de nuestra conexión telnet, escribimos "**quit**".

Analicemos a continuación a través de la captura de esta actividad qué es lo que hizo nuestro Linux.



Lo primero que hemos remarcado en un rectángulo verde, son las dos peticiones DNS y sus respectivas respuestas, primero el registro "**MX**" (Mail Exchange), y luego el "**A**" (Authoritative).

Una vez resueltas, inmediatamente podemos ver el triple Handshake hacia el puerto TCP 25 (smtp), que lo recuadramos en azul.

A continuación vemos un diálogo entre servidor y cliente que es cuando se decide el empleo de ESMTP (lo subrayamos en rojo, dos tramas "**S: y C:**"), se pone de manifiesto que en la primera de estas tramas el servidor propone "ESMTP" y en la segunda de ellas el cliente contesta con EHLO (en vez de HELO), lo que implica que soporta también ESMTP.

Luego ya podemos ver cada uno de los comandos que acabamos de presentar en la conexión telnet hacia este puerto 25: **MAIL FROM**: , **RCPT TO**: y **DATA**, con las correspondientes respuestas "**OK**" que también vimos en nuestra consola Linux.

Por último vemos recuadrado en rojo, la trama donde se envían los datos, en la cual se aprecia perfectamente en texto plano el contenido de este correo: "esta es una prueba de SMTP", así que recordad que el protocolo ESMTP no considera la confidencialidad en absoluto...

En nuestro ejemplo, habíamos habilitado la capacidad de "relay" en nuestro servidor SMTP, por lo tanto este correo, fue recibido por el mismo y reenviado a su destinatario.

2) Ejercicio con POP-3

Para este ejercicio, vamos a emplear la herramienta "**qpopper**" que es un muy sencillo servidor de correo POP-3, si trabajamos con distribuciones Debian de Linux, no tendremos ningún tipo de dificultad para instalarlo, sencillamente: "apt-get install qpopper", no dedicaremos tiempo a su configuración, pues con sólo instalarlo ya se ejecuta y para nuestra práctica es más que suficiente.

Como nos interesa únicamente ver su funcionamiento en local, esta vez no emplearemos Wireshark, sino que sencillamente nos enviaremos un correo y lo leeremos desde la misma máquina Linux, para ello, primero debemos enviarnos un correo. En nuestro host Linux, hemos creado un usuario ("adduser") de nombre: prueba, y con contraseña: "prueba", así que nos conectamos al servidor SMTP local y le enviamos un correo, como figura a continuación:

```
root@BlusensFreePC10:/etc# telnet localhost 25
      Trying 127.0.0.1...
      Connected to localhost.
      Escape character is '^]'.
      220 BlusensFreePC10 ESMTP Postfix (Ubuntu)
HELO blusens
      250 BlusensFreePC10
MAIL FROM: root
      250 2.1.0 Ok
rcpt to: prueba
      250 2.1.5 Ok
data
      354 End data with <CR><LF>.<CR><LF>
hola prueba
.
      250 2.0.0 Ok: queued as 7B7B02A40D6
Subject: correo de prueba.
quit
.
      Connection closed by foreign host.
```

Como acabamos de presentar, lo único que hicimos, fue conectarnos al servidor SMTP ("**telnet localhost 25**"), y enviar un mail desde "**root**" hacia el usuario "**prueba**" cuyo contenido es "hola prueba" y su **Subject**: "correo de prueba", luego salimos de telnet con "**quit**".

Ahora a continuación, veamos el empleo de POP-3, lo primero que haremos será conectarnos por "**telnet localhost 110**" (que acabamos de abrir al instalar "**qpopper**"):

```
root@BlusensFreePC10:/etc# telnet localhost 110
      Trying 127.0.0.1...
      Connected to localhost.
      Escape character is '^]'.
      +OK Qpopper (version 4.0.5) at BlusensFreePC10 starting.
      <6980.1306324654@BlusensFreePC10>
user prueba
      +OK Password required for prueba.
pass prueba
      +OK prueba has 1 visible message (0 hidden) in 466 octets.
stat
      +OK 1 466
list
      +OK 1 visible messages (466 octets)
      1 466
      .
retr 1
      +OK 466 octets
      Return-Path: <root@BlusensFreePC10>
      X-Original-To: prueba
      Delivered-To: prueba@BlusensFreePC10
      Received: from blusens (localhost [127.0.0.1])
            by BlusensFreePC10 (Postfix) with SMTP id 7B7B02A40D6
            for <prueba>; Wed, 25 May 2011 13:55:51 +0200 (CEST)
      Message-Id: <20110525115612.7B7B02A40D6@BlusensFreePC10>
      Date: Wed, 25 May 2011 13:55:51 +0200 (CEST)
      From: root@BlusensFreePC10
      To: undisclosed-recipients:;
      X-UIDL: ;/2"!GP;!!3Z@"!&T,"!

      hola prueba

quit
      +OK Pop server at BlusensFreePC10 signing off.
      Connection closed by foreign host.
```

Me solicita usuario y contraseña (ambos "prueba"), una vez autenticado este usuario, ejecutamos el comando "**stat**" para ver el estado de mis correos, me dice que tengo "1 466", es decir uno sólo de 466 Byte, ejecutamos "**retr**" (entregar) **1** (se coloca el número del mensaje que se desea ver), y nos muestra todo el contenido del mismo. Para finalizar ejecutamos "**quit**" y salimos de telnet.

En los ejemplos anteriores, tratamos de presentaros desde la práctica, cómo trabajan estos dos protocolos, tanto para la entrega como para la recepción de correos, fueron sencillamente dos casos básicos de su empleo, pero puedes ir probando todas las opciones que te ofrecen los

dos. Lo mejor que puedes hacer para ejercitar al máximo estos protocolos es trabajar todo lo que puedas a través de línea de comandos pues es a través de ella dónde podrás obtener toda la potencia de los mismos (siempre capturando lo que hagas), y verás que luego cualquier interfaz gráfica que emplees no tendrá ningún misterio.

6. Ejercicios con SNMP.

Al protocolo SNMP le dedicaremos bastantes ejercicios de práctica, pues es de suma importancia para la monitorización de redes y hemos detectado que suele ser una gran falencia a nivel empresarial y también que está poco desarrollado didácticamente. Como irás viendo en esta parte de ejercicios, si bien iremos practicando bastante con diferentes herramientas y comandos algunos más seguros y otros no tanto, nuestro objetivo final es que llegues a desenvolverte bien con SNMP versión 3, que tal cual expresamos en la teoría, está diseñado para entornos seguros tanto en su autenticación como en su confidencialidad.

1) Para los ejercicios snmp, lo primero que haremos es verificar si existen puertos 161 abiertos.

   " #nmap -sU 161 192.168.1.0/24"

   En este ejemplo, se estaría escaneando todo el rango de la red tipo "C" 192.168.1.0, pero hazlo con el que desees.

2) También puedes hacerlo con "netstat" para tu propia máquina, como presentamos a continuación:

   "#netstat -au |grep snmp"

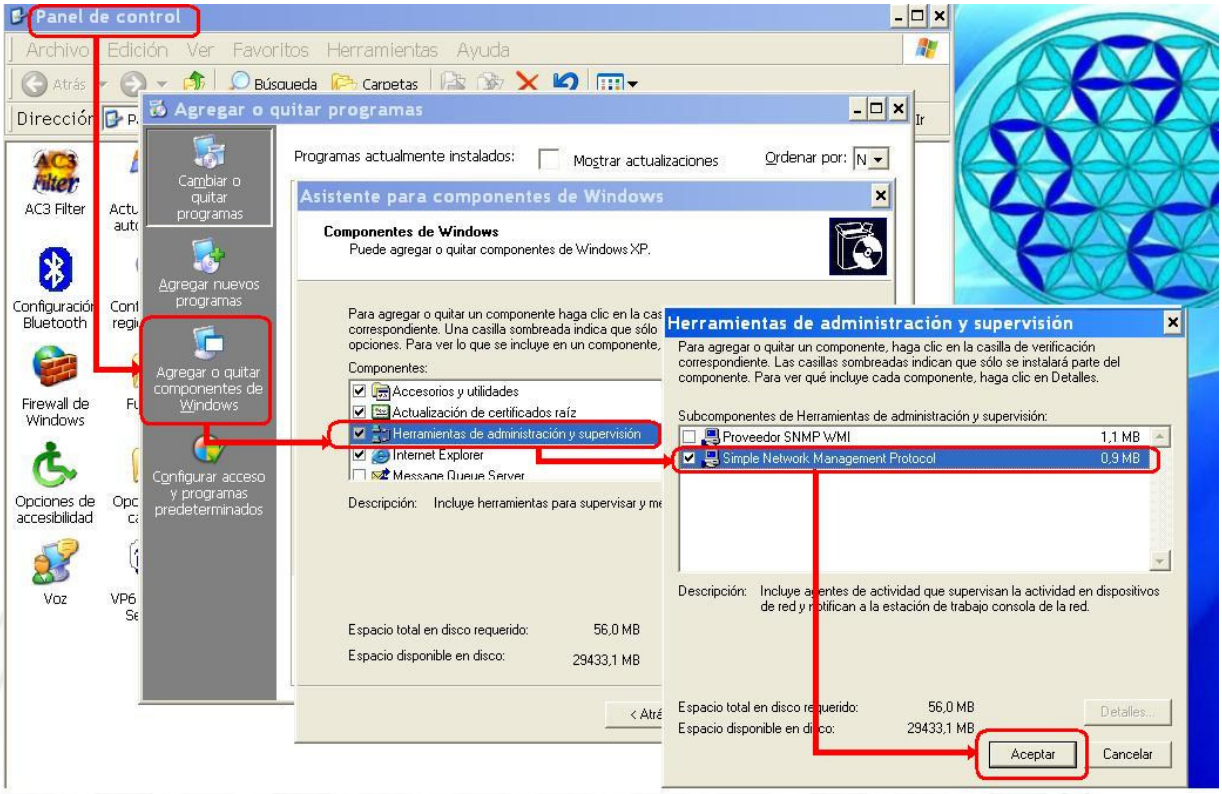
   udp 0 0 *:snmp

   (En nuestro ejemplo nos demuestra que sí está escuchando), ¿A ti que te responde?

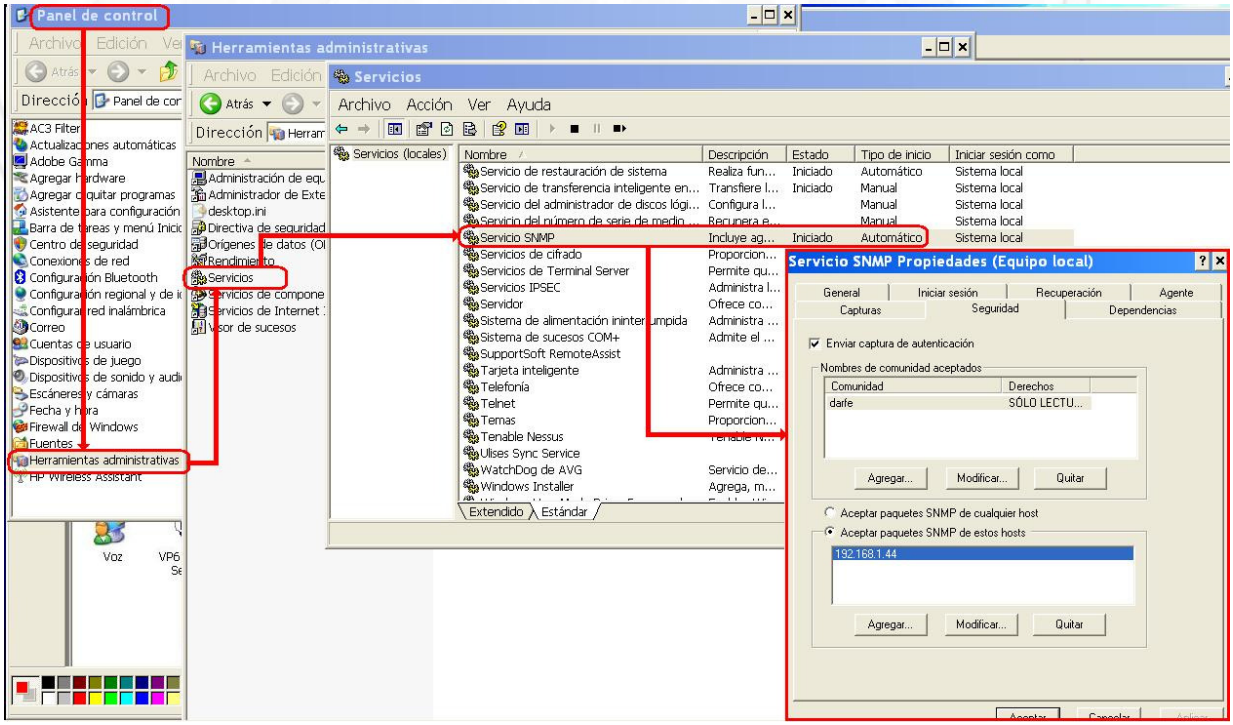
3) Luego iniciaremos con los agentes y servidores snmp. Los servidores los implantaremos desde Linux, y los agentes en Linux y Windows.

   Para el agente Windows, Se debe instalar el servicio (que por defecto no viene instalado), para ello los pasos son desde "Panel de control → Agregar o quitar componentes de Windows → Herramientas de administración y supervisión → Simple Network Monitor Protocol → Aceptar (nos pedirá que ingresemos el CD de Instalación).

   Los pasos se detallan en la imagen siguiente:



Una vez instalado el protocolo, el paso siguiente es activar el servicio, como también se puede apreciar en la siguiente imagen:



La secuencia se indica con flechas rojas, igual que en la imagen anterior, lo que deseamos remarcar aquí es la configuración de seguridad (mínima) que ofrece, en nuestro ejemplo, modificamos la comunidad por defecto "public" por la nuestra: "**darfe**" y a su vez acotamos el host desde el cual acepta conexiones, en nuestro caso al "192.168.1.44" que es el que emplearemos en esta práctica.
Debemos mencionar que para poder emplear seguridad de autenticación y cifrado, la alternativa que ofrece la familia "Windows" es la de hacerlo a través de túneles IPSec, no lo desarrollaremos en este texto para centrarnos en SNMPv3, pero puedes encontrar los pasos a seguir en cualquier buscador de Internet.

4) La herramienta que se empleará para el resto del trabajo será la más conocida de Linux que viene en el paquete "**Net-snmp**". Su Web es: http://www.net-snmp.org.

   Se instala a través de dos paquetes, para distribuciones Debian sería:

   "#**apt-get install snmp snmpd**" (*no tiene ninguna dificultad*)

   Este paquete una vez instalado veremos que posee muchas aplicaciones:

   Aplicaciones básicas de NET-SNMP:

   - **encode_keychange**: produce la cadena para el intercambio de claves en SNMPv3
   - **snmpcmd**: comandos para "opciones y comportamiento" que son comunes a la mayoría (casi todos) las líneas de comandos de Net-SNMP
   - **snmptranslate**: traduce los identificadores (OID) de la MIB entre las formas numéricas y las textuales
   - **snmpget**: comunica con una entidad de red utilizando una solicitud "SNMP GET"
   - **snmpgetnext**: ídem utilizando "SNMP GETNEXT"
   - **snmpbulkget**: ídem utilizando "SNMP GETBULK"
   - **snmpwalk**: Entrega una parte de la estructura del árbol de la MIB mediante peticiones "SNMP GETNEXT"
   - **snmpbulkwalk**: Entrega una parte de la estructura del árbol de la MIB mediante peticiones "SNMP GETBULK"
   - **snmpset**: comunica con una entidad de la red utilizando peticiones "SNMP SET"
   - **snmptest**: comunica con una entidad de la red utilizando peticiones "SNMP"
   - **tkmib**: navegador gráfico interactivo de la MIB para SNMP

   "segundo nivel" de aplicaciones de NET-SNMP:

   - **snmptable**: recupera una tabla SNMP y la muestra en forma de tabla

- **snmpdelta**: Monitoriza diferencias o incrementos (delta) del en los valores de contadores
- **snmpsum**: crea y mantiene los usuarios SNMPv3 sobre una entidad de red
- **snmpvacm**: crea y mantiene el control de acceso basado en vistas (VACM)para SNMPv3 sobre una entidad de red
- **snmpstatus**: recupera un conjunto fijo de información sobre la gestión de una entidad de red
- **snmpnetstat**: Muestra el estado de la red y su configuración a través de SNMP
- **snmpdf**: Muestra el uso de espacio en disco de una entidad de red a través de SNMP

5) Antes de iniciar el detalle con "**net-snmp**" y ver su funcionamiento, queríamos presentar cómo se podría "ver" básicamente la configuración que acabamos de hacer en el agente "**Windows XP**". Para ello desde nuestro host **Linux** en el que acabamos de instalar "**net-snmp**" (cuya dirección IP es la 192.168.1.44, tal cual autorizamos a acceder en Windows), podemos realizar la siguiente consulta:

```
root@BlusensFreePC10:/home/blusens# snmpwalk -c public -v 1 192.168.1.50
system
     SNMPv2-MIB::sysDescr.0 = STRING: Hardware: x86 Family 6 Model 15
     Stepping 13 AT/AT COMPATIBLE - Software: Windows 2000 Version 5.1
     (Build 2600 Multiprocessor Free)
     SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-
     SMI::enterprises.311.1.1.3.1.1
     DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (44528) 0:07:25.28
     SNMPv2-MIB::sysContact.0 = STRING:
     SNMPv2-MIB::sysName.0 = STRING: DARFE2
     SNMPv2-MIB::sysLocation.0 = STRING:
     SNMPv2-MIB::sysServices.0 = INTEGER: 79
```

Este punto ha sido una sencilla práctica para que puedas verificar que se instaló correctamente el agente en Windows, pero el detalle de estos comandos es lo que comenzamos a tratar a continuación

6) Configuración de "**net-snmp**".

En nuestro caso vamos a configurar todo el agente desde el archivo **"/etc/snmp/snmpd.conf"**, para que cada vez que se lanza el demonio "**snmpd**" busque allí. Debemos editar el archivo "**/etc/default/snmpd**" y dejarlo como figura a continuación (sólo debes comentar (#) la línea que viene por defecto y agregar la de abajo que apunta a donde acabamos de decir).

```
# snmpd options (use syslog, close stdin/out/err).
#SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid'
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid -
c /etc/snmp/snmpd.conf'
```

Debemos aclarar que "Net-snmp" dentro de sus aplicaciones, ofrece otra herramienta que es "**snmpconf**", la cual nos va guiando paso a paso para la configuración de toda la plataforma. Como puede presentar alguna dificultad y deseamos desde ya comenzar a orientarnos hacia snmpv3, aunque es larga decidimos pegarla a continuación con cada uno de los pasos que hemos ejecutado nosotros para que lo puedas ver también (Pusimos en "negrita" las líneas en las que ingresamos respuestas):

```
# snmpconf

The following installed configuration files were found:

   1:  /etc/snmp/snmp.conf
   2:  /etc/snmp/snmpd.conf
   3:  /etc/snmp/snmptrapd.conf

Would you like me to read them in?  Their content will be merged
with the output files created by this session.

Valid answer examples: "all", "none","3","1,2,5"

Read in which (default = all):

I can create the following types of configuration files for you.
Select the file type you wish to create:
(you can create more than one as you run this program)

   1:  snmpd.conf
   2:  snmp.conf
   3:  snmptrapd.conf

Other options: quit

Select File: 2

The configuration information which can be put into snmp.conf is
divided into sections.  Select a configuration section for snmp.conf
that you wish to create:

   1:  Debugging output options
   2:  Default Authentication Options
   3:  Textual mib parsing
   4:  Output style options

Other options: finished

Select section: 1

Section: Debugging output options
Description:
  This section allows debugging output of various kinds to
  be turned on or off.

Select from:

   1:  Turns debugging output on or off (0|1)
```

```
   2:  Debugging tokens specify which lines of debugging
   3:  Print packets as they are received or sent
   4:  Silence warnings about unknown tokens in configuration files

Other options: finished, list

Select section: 3

Configuring: dumppacket
Description:
  Print packets as they are received or sent
    arguments: (1|yes|true|0|no|false)
    command line equivelent: -d

Print packets as they are received or sent: 1

Finished Output: dumppacket  1

Section: Debugging output options
Description:
  This section allows debugging output of various kinds to
  be turned on or off.

Select from:

   1:  Turns debugging output on or off (0|1)
   2:  Debugging tokens specify which lines of debugging
   3:  Print packets as they are received or sent
   4:  Silence warnings about unknown tokens in configuration files

Other options: finished, list

Select section: list
Lines defined for section "Debugging output options" so far:
  dumppacket  1

Section: Debugging output options
Description:
  This section allows debugging output of various kinds to
  be turned on or off.

Select from:

   1:  Turns debugging output on or off (0|1)
   2:  Debugging tokens specify which lines of debugging
   3:  Print packets as they are received or sent
   4:  Silence warnings about unknown tokens in configuration files

Other options: finished, list

Select section: finished


The configuration information which can be put into snmp.conf is
divided into sections.  Select a configuration section for snmp.conf
that you wish to create:
```

```
   1:  Debugging output options
   2:  Default Authentication Options
   3:  Textual mib parsing
   4:  Output style options

Other options: finished

Select section: 2

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 4

Configuring: defsecurityname
Description:
  The default snmpv3 security name to use when using snmpv3
    override: with -u on the command line.
    arguments: securityname

Enter the default security name to use: darfe

Finished Output: defsecurityname  darfe

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
```

```
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 5

Configuring: defcontext
Description:
  The default snmpv3 context name to use
    override: with -n on the command line.
    arguments: contextname

Enter the default context name to use: empresa

Finished Output: defcontext  empresa

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 6

Configuring: defsecuritylevel
Description:
  The default snmpv3 security level to use
    override: with -l on the command line.
    arguments: noAuthNoPriv|authNoPriv|authPriv

Enter the default privacy pass phrase to use: authPriv

Finished Output: defsecuritylevel  authPriv
```

```
Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 7

Configuring: defauthtype
Description:
  The default snmpv3 authentication type name to use
    override: with -a on the command line.
    arguments: authtype

Enter the default authentication type to use (MD5|SHA): SHA

Finished Output: defauthtype  SHA

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list
```

```
Select section: 8

Configuring: defauthpassphrase
Description:
  The default snmpv3 authentication pass phrase to use
    Note: It must be at least 8 characters long.
    override: with -A on the command line.
    arguments: passphrase

Enter the default authentication pass phrase to use: 12345678

Finished Output: defauthpassphrase  12345678

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 9

Configuring: defprivtype
Description:
  The default snmpv3 privacy (encryption) type name to use
    override: with -x on the command line.
    arguments: privtype

Enter the default privacy type to use (DES|AES): DES

Finished Output: defprivtype  DES

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:
```

```
   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: 10

Configuring: defprivpassphrase
Description:
  The default snmpv3 privacy pass phrase to use
    Note: It must be at least 8 characters long.
    override: with -X on the command line.
    arguments: passphrase

Enter the default privacy pass phrase to use: 23456789

Finished Output: defprivpassphrase  23456789

Section: Default Authentication Options
Description:
  This section defines the default authentication
  information.  Setting these up properly in your
  ~/.snmp/snmp.conf file will greatly reduce the amount of
  command line arguments you need to type (especially for snmpv3).

Select from:

   1:  The default port number to use
   2:  The default snmp version number to use.
   3:  The default snmpv1 and snmpv2c community name to use when
needed.
   4:  The default snmpv3 security name to use when using snmpv3
   5:  The default snmpv3 context name to use
   6:  The default snmpv3 security level to use
   7:  The default snmpv3 authentication type name to use
   8:  The default snmpv3 authentication pass phrase to use
   9:  The default snmpv3 privacy (encryption) type name to use
  10:  The default snmpv3 privacy pass phrase to use

Other options: finished, list

Select section: finished
```

7) Explicación de parámetros

Una vez configurado, y antes de comenzar a usar "**Net-snmp**" debes tener en cuenta que hay un conjunto de parámetros que son comunes a "*casi*" todas las aplicaciones o comandos, el conjunto de ellos puedes consultarlo con el "**man snmpcmd**", los que por ahora nos interesa tener en cuenta son:

- "**-v**" Versión (actualmente acepta [1|2c|3]).
- "**-c**" Nombre de la comunidad.
- "**-t**" Tiempo de espera.
- "**-r**" Número de intentos de conexión.
- "**-O**" Opciones de salida.
- "**-a**" Algoritmo (o resumen) de autenticación (actualmente acepta [SHA|MD5]).
- "**-A**" Contraseña para autenticación (debe ser mayor a 8 caracteres).
- "**-x**" Algoritmo de cifrado de datos (actualmente acepta [DES|AES]).
- "**-X**" Contraseña para cifrado de datos (debe ser mayor a 8 caracteres).
- "**-l**" Nivel de seguridad (actualmente acepta [NoauthNoPriv|authNoPriv|authPriv|]) (Se explica más adelante)
- "**-n**" Configura en parámetro "contextName" usado para SNMPv3. El ContextName por defecto es la cadena vacía, es decir que se puede emplear como "-n "". Sobre escribe el valor "**defContext**" que figura en el "**snmp.conf**"..... *En lo personal, es uno de los enigmas de mi vida (y que más horas me ha llevado en SNMPv3 con cientos de errores, como veréis en todos los ejemplos me rendí llamándolo en esos casos como "-n "" y de esta forma es la ÚNICA QUE LOGRÉ QUE FUNCIONE. Ruego a algún lector que logre "domarlo" que sea tan amable de explicárnoslo.....*

8) Empleo de "**snmpwalk**"

Podemos consultar de deferentes formas, en este caso con una opción de texto como es "interface"

```
#snmpwalk -v 1 -c "darfe" localhost interface
      IF-MIB::ifNumber.0 = INTEGER: 4
      IF-MIB::ifIndex.1 = INTEGER: 1
      IF-MIB::ifIndex.2 = INTEGER: 2
      IF-MIB::ifIndex.3 = INTEGER: 3
      IF-MIB::ifIndex.4 = INTEGER: 4
      IF-MIB::ifDescr.1 = STRING: lo
      IF-MIB::ifDescr.2 = STRING: eth2
      IF-MIB::ifDescr.3 = STRING: wmaster0
      IF-MIB::ifDescr.4 = STRING: wlan1
      IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
      IF-MIB::ifType.2 = INTEGER: ethernetCsmacd(6)
      IF-MIB::ifType.3 = INTEGER: other(1)
      IF-MIB::ifType.4 = INTEGER: ethernetCsmacd(6)
      IF-MIB::ifMtu.1 = INTEGER: 16436
      IF-MIB::ifMtu.2 = INTEGER: 1500
```

```
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifMtu.4 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 10000000
IF-MIB::ifSpeed.2 = Gauge32: 10000000
IF-MIB::ifSpeed.3 = Gauge32: 0
IF-MIB::ifSpeed.4 = Gauge32: 10000000
IF-MIB::ifPhysAddress.1 = STRING:
IF-MIB::ifPhysAddress.2 = STRING: 0:1a:13:b3:ec:b0
IF-MIB::ifPhysAddress.3 = STRING: 0:22:43:2:50:e6
IF-MIB::ifPhysAddress.4 = STRING: 0:22:43:2:50:e6
IF-MIB::ifAdminStatus.1 = INTEGER: up(1)
IF-MIB::ifAdminStatus.2 = INTEGER: up(1)
IF-MIB::ifAdminStatus.3 = INTEGER: up(1)
```

También se puede realizar esta consulta hacia las interfaces directamente

```
#snmpwalk -v 1 -c "darfe" localhost  ifDescr
    IF-MIB::ifDescr.1 = STRING: lo
    IF-MIB::ifDescr.2 = STRING: eth2
    IF-MIB::ifDescr.3 = STRING: wmaster0
    IF-MIB::ifDescr.4 = STRING: wlan1
```

(o lo que es lo mismo)

```
#snmpwalk -v 1 -c "darfe" localhost
    IF-MIB::ifName
    IF-MIB::ifName.1 = STRING: lo
    IF-MIB::ifName.2 = STRING: eth2
    IF-MIB::ifName.3 = STRING: wmaster0
    IF-MIB::ifName.4 = STRING: wlan1
```

9) El objetivo fundamental de estos ejercicios es que puedas llevar a la práctica el modelo seguro de SNMP, es decir la versión 3, en este texto aplicaremos el modelo **USM** Especificado en la **RFC-2574** y explicado en detalle en la teoría.

   Para comenzar a emplear **snmpv3**, lo primero que debemos hacer es crear los usuarios que podrán acceder, el formato general para ello es:

```
#net-snmp-config --create-snmpv3-user [-ro] [-A
contraseña_Autenticacion] [-X Contraseña_Cifrado]
                        [-a MD5|SHA] [-x DES|AES] [nombre_usuario]
```

   En nuestros ejemplos crearemos los siguientes usuarios:

```
#net-snmp-config --create-snmpv3-user -ro -a SHA 12345678 alejandro
-x DES -X 34567890

#net-snmp-config --create-snmpv3-user -ro -a MD5 23456789 javier
```

   Al ejecutar estos comandos se crean líneas para guardar estos datos, pero como corresponde no puede ser en texto plano, por lo tanto, Linux guarda justamente el **HASH** de los mismos

(así lo especifica la **RFC-2574**). En el caso de distribuciones Debian lo hace en "**/var/lib/snmp/snmpd.conf**", el cual NO SE DEBE CONFUNDIR con el del mismo nombre que está en "**/etc/snmp**", pues este último sí que podemos configurar a gusto, pero el anterior, que presentamos a continuación justamente ¡NO SE DEBE TOCAR!

```
##############################################################################
# STOP STOP STOP STOP STOP STOP STOP STOP STOP
#
#          **** DO NOT EDIT THIS FILE ****
#
# STOP STOP STOP STOP STOP STOP STOP STOP STOP
##############################################################################
#
# DO NOT STORE CONFIGURATION ENTRIES HERE.
# Please save normal configuration tokens for snmpd in
SNMPCONFPATH/snmpd.conf.
# Only "createUser" tokens should be placed here by snmpd administrators.
# (Did I mention: do not edit this file?)
#

usmUser 1 3 0x80001f8880dc94b7367dfce54d 0x726f6f7400 0x726f6f7400 NULL
.1.3.6.1.6.3.10.1.1.2 0x6537145fb6c638918f763ca8c692b826
.1.3.6.1.6.3.10.1.2.2 0x6537145fb6c638918f763ca8c692b826 0x00
usmUser 1 3 0x80001f8880dc94b7367dfce54d 0x6a617669657200 0x6a617669657200
NULL .1.3.6.1.6.3.10.1.1.2 0x0e3a0d458b512f940a83f24d7b7db233
.1.3.6.1.6.3.10.1.2.1 "" ""
usmUser 1 3 0x80001f8880dc94b7367dfce54d 0x6d6f6e69746f7200
0x6d6f6e69746f7200 NULL .1.3.6.1.6.3.10.1.1.3
0x62b87803df63e7a983a8b98b53c7d845c75bfc4d .1.3.6.1.6.3.10.1.2.2
0x62b87803df63e7a983a8b98b53c7d845 0x00
```

Por defecto este comando añade también una línea en **"/etc/snmp/snmpd.conf"** con permisos de lectura y NO de escritura (las últimas del que presentamos a continuación), luego puedes editarlos y cambiar lo que desees.

10) A continuación presentamos un archivo "**snmpd.conf**" con una configuración básica para que puedas comenzar a trabajar (en este caso ya estamos preparando todo para que en los ejercicios podamos ver la diferencia entre SNMP versión 1 y versión 3)

```
#====================================
#definiciones

defcontext darfe
defsecurityName alejandro
defsecuritylevel authNoPriv

#Creación de usuarios

createUser alejandro SHA 12345678 DES 34567890
createUser javier MD5 23456789

#Características de acceso
```

```
rwuser alejandro noauth .1.3.6.1.2.1
rouser javier auth .1.3.6.1.2.1
#====================================
```

Ante cualquier cambio que se haga en SNMP se debe reiniciar el demonio: "**/etc/init.d/snmpd restart**"

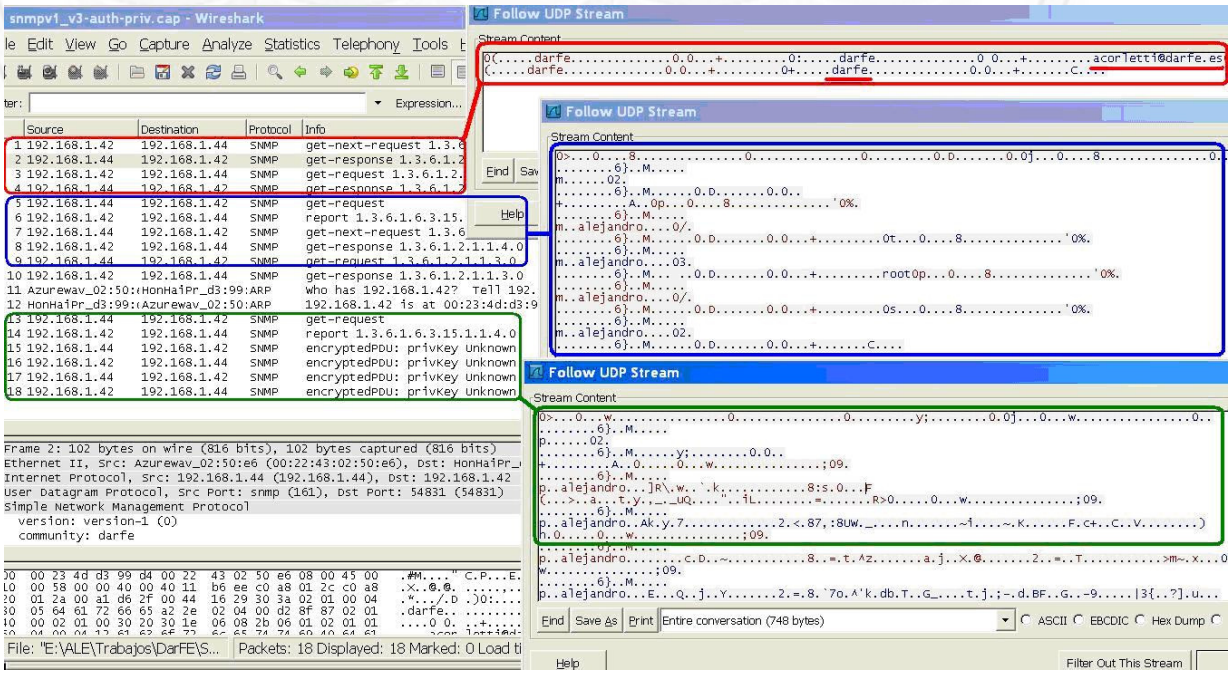
11) En esta serie de ejercicios vamos a ir trabajando tanto en **snmpv1** como en **snmpv3**, pues queremos que veas bien las diferencias entre ambos y que te convenzas que en un entorno serio y en producción no tienes otra opción que emplear la versión 3 sino tendrás permanentemente puertas abiertas muy peligrosas. La parte de versión 1 la presentamos como algo didáctico, muy útil a la hora del aprendizaje PERO NADA MÁS, reiteramos que es peligroso.

Cuando empecemos el trabajo con SNMPv3, algo importante a considerar (tal cual lo dijimos en la teoría) es la posibilidad que ofrece de "**autenticar**" y "**cifrar**". Como iremos viendo en estos ejercicios, esto se controla a través de tres opciones:

- **NoauthNoPriv**: no se realiza autenticación, ni se cifran los datos en tránsito.
- **authNoPriv**: Sí se realiza autenticación, pero no se cifran los datos en tránsito.
- **authPriv**: Se realiza autenticación y se cifran los datos en tránsito.

Este tema lo iremos viendo y practicando en reiterados ejercicios.

12) Presentaremos es una captura empleando diferentes consultas, primero SNMPv1, luego SNMPv3 sin autenticación ni criptografía y finalmente SNMPv3 con autenticación y criptografía.



El primer recuadro (rojo), nos presenta un flujo SNMPv1, en el mismo hemos subrayado en rojo, como se puede observar la comunidad "darfe" y el contacto del usuario que se conecta acorletti@darfe.es, y en la ventana principal de "Wireshark" en la columna "info" se aprecian las solicitudes y respuestas con sus valores en texto plano ("*1.3.6.1...*")

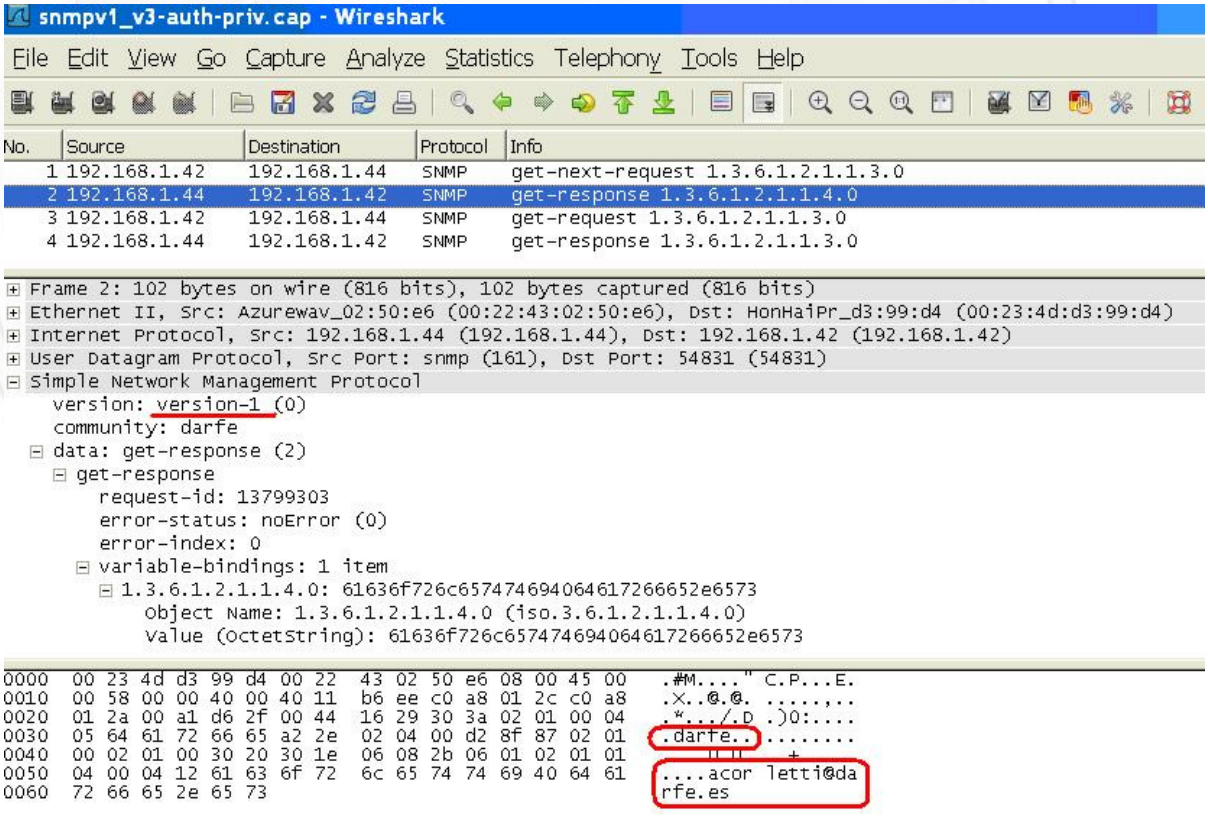
En el segundo recuadro (azul) se presenta un flujo SNMPv3, pero sin emplear autenticación ni privacidad. Ya se puede apreciar que no se ve ni la comunidad ni el correo electrónico, si bien se ve el usuario que solicita esta información "alejandro", pero en la ventana principal de Wireshark, se continúa viendo la información de al MIB en texto plano ("*1.3.6.1...*").

Por último, remarcamos en verde un flujo SNMPv3, ahora sí empleando autenticación y criptografía. El seguimiento del flujo puede parecernos similar, pero en la ventana principal de Wireshark se nota claramente que ahora, una vez que el usuario "alejandro" se hace presente, se establece un secreto compartido y todo viaja de forma cifrada ("*Encrypted PDU: Privkey Unknown*").

Para la solicitud de **SNMPv1** recuadrada en rojo en la imagen anterior, la consulta realizada fue:

```
#snmpwalk -v 1 -c darfe 192.168.1.44 1.3.6.1.2.1.1.3.0
```

Si desplegamos al completo esa captura de SNMPv1 el contenido de la misma es:



Como se puede observar, el encabezado es tal cual se desarrolló en la teoría, y en su contenido hexadecimal, comprobamos cómo toda la información viaja en texto plano.

13) A continuación presentamos algunos ejercicios en los que puedes practicar y verificar el empleo de "**auth**" y "**priv**":

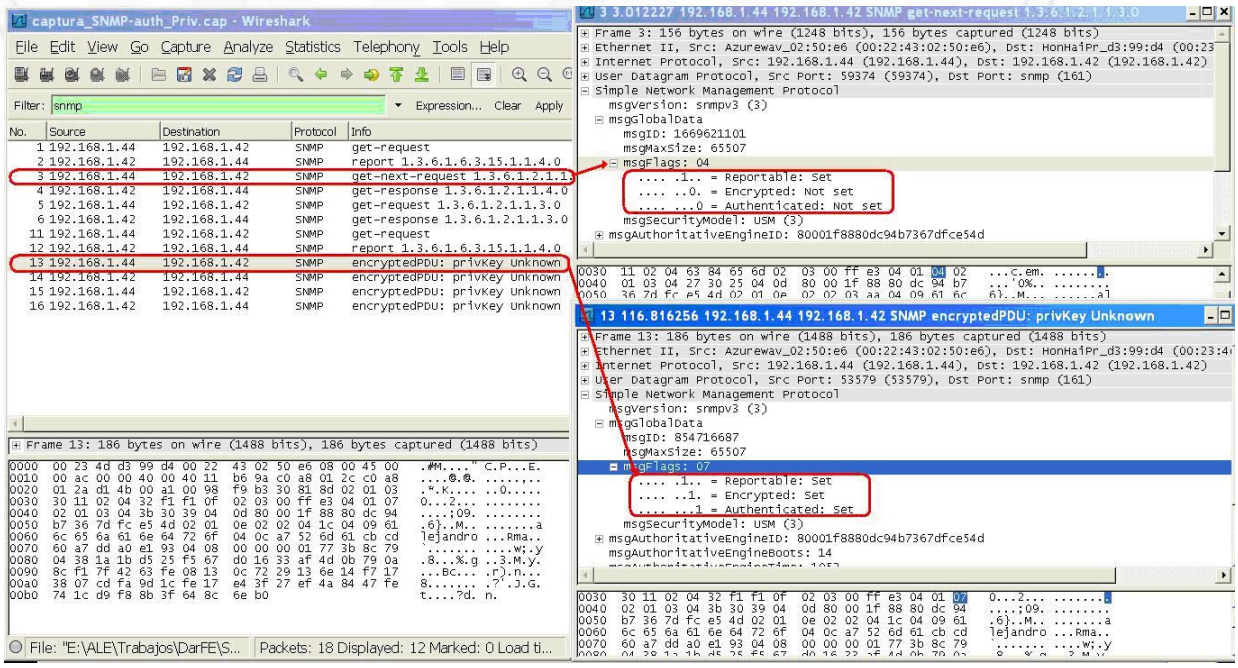
```
#snmpwalk -v 3 -n '' -u alejandro -a SHA -A 12345678 -x DES -X
34567890 -l authNoPriv localhost -Oa

#snmpwalk -v 3 -n '' -u javier -a MD5 -A 23456789 -l authNoPriv
localhost -Oa
```

En el ejemplo anterior, se presenta cómo se realizaría una consulta para el caso de una cuenta con "SHA" y otro con "MD5", en el caso que tú también hayas creado un par de usuarios con ese resumen de autenticación, no deberías tener problemas si ejecutas este tipo de consulta, recibiendo toda la respuesta de la MIB consultada.

A continuación te proponemos un ejemplo en el cual puedes verificar la diferencia entre información y autenticación que viajará cifrada e información y autenticación que viajará en texto plano. Para ello, como mencionamos anteriormente, la opción es "aut" y "noAuth" junto con "NoPriv" o "Priv", por lo tanto para el primer análisis, las consultas deberían ser como las que te pegamos a continuación:

```
#snmpwalk -v 3 -n '' -c "darfe" -u alejandro -a SHA -A 12345678 -x DES
-X 34567890 -l NoauthNoPriv 192.168.1.42 sysUpTime.0

#snmpwalk -v 3 -n '' -c "darfe" -u alejandro -a SHA -A 12345678 -x DES
-X 34567890 -l authPriv 192.168.1.42 sysUpTime.0
```

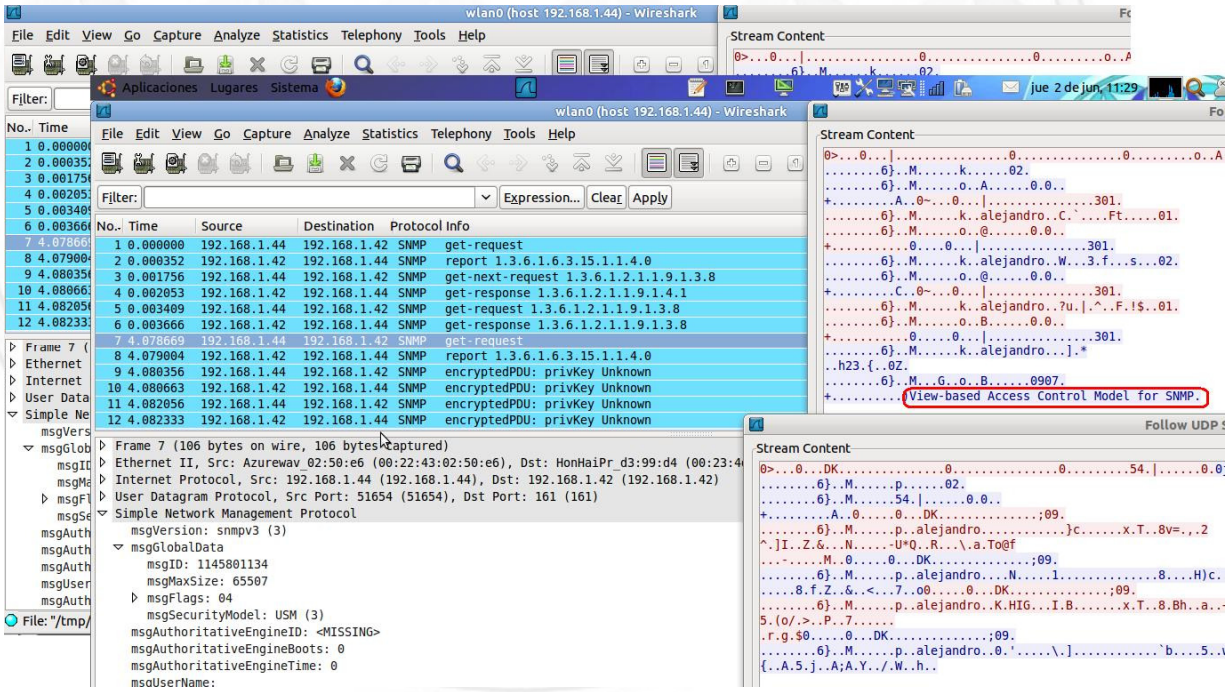
En nuestro ejemplo, como puedes ver, primero se hace una petición con "**authPriv**" y luego otra con "**NoauthNoPriv**" solicitando en tiempo que lleva funcionando el sistema. Al ejecutar estos comandos teníamos "Wireshark" escuchando, y las tramas capturadas son las que se presentan a continuación:



En la imagen anterior tienes: desde la trama 1 a la 6 la petición "**NoautNoPriv**" y desde la trama 11 a la 16 la "**authPriv**". Lo primero que deseamos destacar es que veas la diferencia en la columna "**Info**" de **Wireshark**, en las seis primeras, queda muy claro que lo que viaja son los valores exactos de solicitud y respuesta de esa **MIB** (*1.3.6.1....*), en cambio en las "**authPriv**", puedes notar cómo en a partir de la trama 13 ya viaja todo cifrado a través de una clave que es desconocida para "Wireshark" ("*Privkey Unknown*"), como lo sería también para cualquier otra persona que esté escuchando este tráfico y no forme parte de este "par de claves" que generaron el secreto compartido que está cifrando esos datos. Por lo tanto nadie ajeno a este diálogo puede obtener ningún dato de quien se autentica ni de la MIB consultada.

Lo segundo que hemos remarcado en la imagen anterior con recuadros y líneas rojas, son los parámetros que han sido enviados a partir del momento en que el comando fue "**NoauthNoPriv**" o "**authPriv**". Estas órdenes, se reflejan en el campo "MsgFlags" del encabezado SNMP, los cuales como puedes apreciar, en el caso de "**NoauthNoPriv**" están a "cero" (*Encrypted: Not set, Authenticated: Not set*), y sin embargo en el caso de "**authPriv**" están a "uno" (*Encrypted: set, Authenticated: set*).

Si desde de Wireshark, hacemos un seguimiento de ambos flujos UDP ("Analyze → Follow UDP Stream"), podemos verificar como en el caso del flujo "**auth priv**", se genera una entrada hacia VACM (modelo de control de accesos basado en vistas), como se presenta a continuación:



14) Opción "**Syscontact**".

Este es un parámetro que se puede incluir en "**snmpd.conf**":

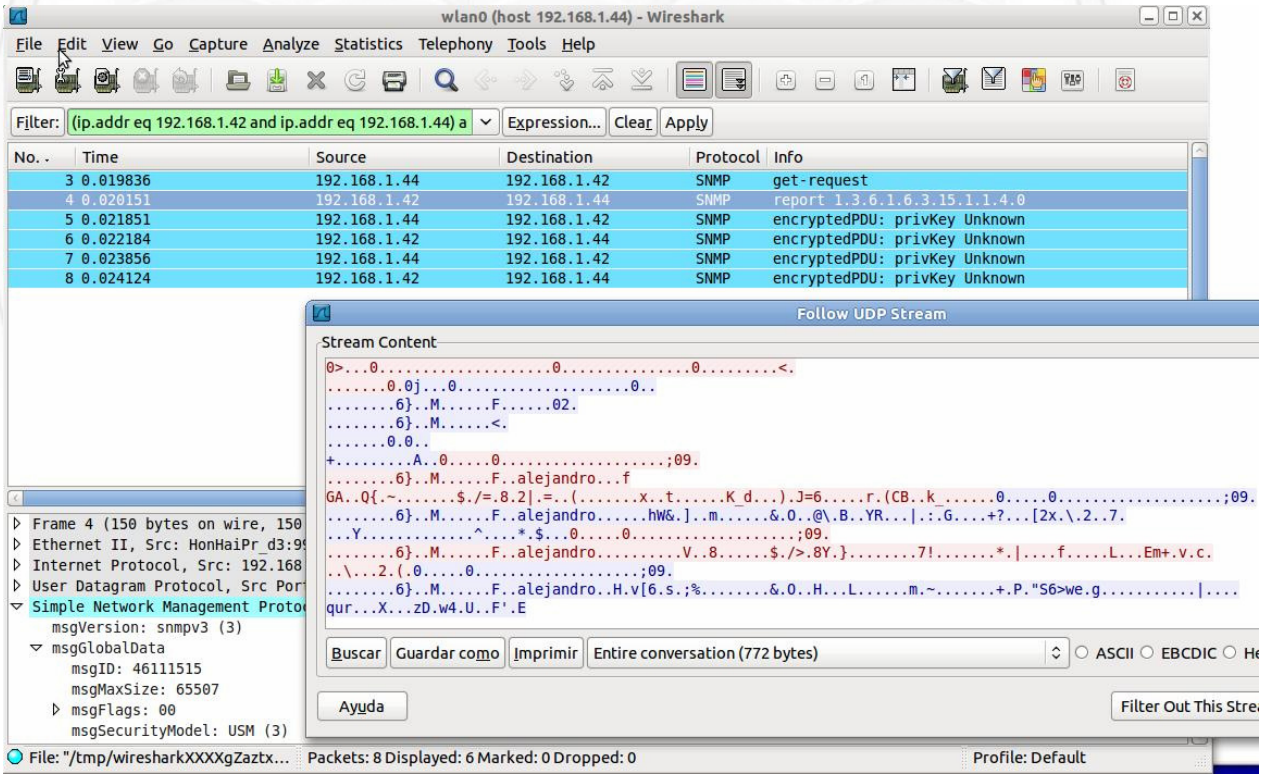
A continuación presentamos dos solicitudes **snmpv3**, la primera de ellas se realizó cuando en nuestro archivo "**snmpd.conf**" no existía el parámetro "syscontact" configurado, por lo tanto como podéis ver la respuesta fue el usuario por defecto de mi Linux que es "**root**" (ya sé que

está mal validarse como "root", pero es un vicio que tengo desde hace muchos, muchos años.....). Luego ingresamos el parámetro "**syscontact**" a nuestro archivo "**/etc/snmp/snmpd.conf**" tal cual figura abajo:

```
rwcommunity  darfe
syslocation  "Madrid, Oficinas Centrales"
syscontact   acorletti@darfe.es
```

Una vez hecho esto, reiniciamos "snmpd" y al lanzar nuevamente la solicitud snmpv3, la respuesta fue "acorletti@darfe.es", tal cual podemos ver en la segunda solicitud. Creímos importante destacar este hecho, pues no es necesario dar a conocer los usuarios del sistema, aunque como puedes ver en la captura, por ser snmpv3 no figura en ningún momento en tránsito.

```
#snmpwalk -v 3 -n '' -c "darfe" -u alejandro -a SHA -A 12345678 -x DES -X
34567890 -l authPriv 192.168.1.42 1.3.6.1.2.1.1.4.0
      SNMPv2-MIB::sysContact.0 = STRING: root

#snmpwalk -v 3 -n '' -c "darfe" -u alejandro -a SHA -A 12345678 -x DES -X
34567890 -l authPriv 192.168.1.42 1.3.6.1.2.1.1.4.0
      SNMPv2-MIB::sysContact.0 = STRING: acorletti@darfe.es
```



Tanto en la ventana principal de "**Wireshark**", como en la del seguimiento de este flujo **SNMPv3** presentado en la imagen anterior (correspondiente a una de las solicitudes "syscontact"), se puede apreciar que los datos viajan cifrados, sólo se puede ver el nombre del usuario "alejandro" correspondiente a la opción "-u alejandro" de la solicitud, pero en ningún caso vemos pasar ni "root" ni "acorletti@darfe.es", sin embargo nuestro agente y servidor SNMPv3, lo han podido cifrar y descifrar sin problemas, tal cual se muestra en la respuesta por consola de ambas solicitudes (que se presentan al principio de este ejercicio)

15) Lo mismo sucede con el parámetro "**syslocation**" el cual se puede incluir en el mismo archivo de configuración y consultar como se ve abajo:

```
#snmpwalk -v 1 -c darfe 192.168.1.44  1.3.6.1.2.1.1.6.0
        iso.3.6.1.2.1.1.6.0 = STRING: "\"Madrid, Oficinas Centrales\""
```

16) Ejercicios con "**snmptranslate**".

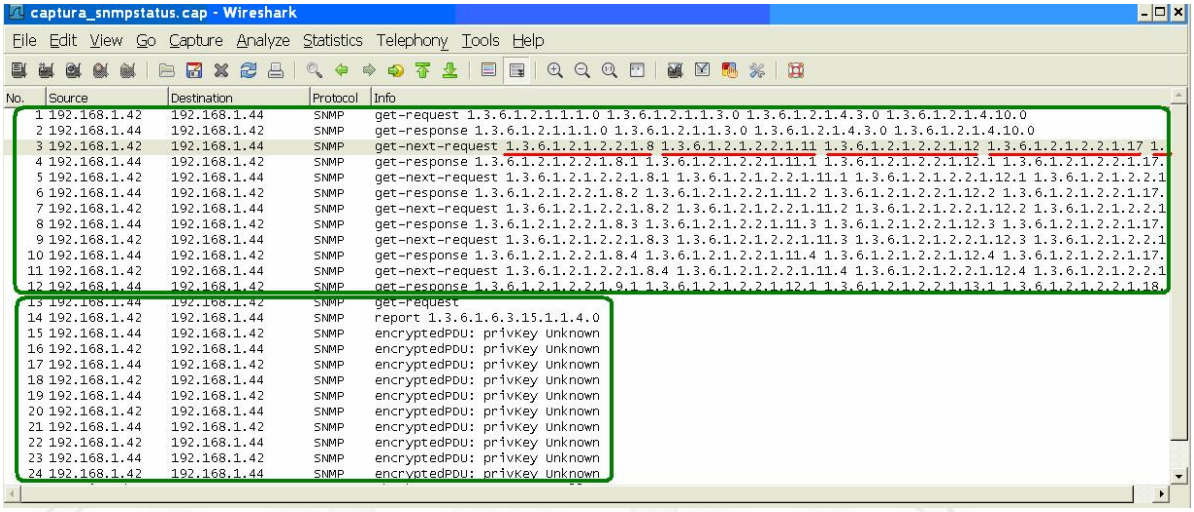
También podemos movernos convirtiendo los parámetros de la MIB y ejecutándolos bajo un formato u otro, como vemos a continuación:

```
#snmptranslate 1.3.6.1.2.1.1.3.0
        DISMAN-EVENT-MIB::sysUpTimeInstance
#snmptranslate 1.3.6.1.2.1.1.4.0
        SNMPv2-MIB::sysContact.0
#snmptranslate 1.3.6.1.2.1.1.4.0
        SNMPv2-MIB::sysContact.0
#snmptranslate 1.3.6.1.2.1.1.5.0
        SNMPv2-MIB::sysName.0
```

17) Ejercicio con "**snmpstatus**"
En este ejercicio quisimos poner de manifiesto ambas versiones de SNMP, para ello como puedes ver hemos ejecutado solicitudes en un sentido con "**snmpv1"** y en el otro con "**snmpv3".**

Las dos direcciones IP son 192.168.1.44 y 192. 168.1.42.

```
root@ace-DarFE:/etc/snmp# snmpstatus -v 1 -c darfe 192.168.1.44
  [UDP: [192.168.1.44]:161->[0.0.0.0]]=>[Linux BlusensFreePC10 2.6.24-
  24-generic #1 SMP Fri Jul 24 22:46:06 UTC 2009 i686] Up: 1:53:30.06
  Interfaces: 4, Recv/Trans packets: 12914/12813 | IP: 12819/12729
  1 interface is down!

root@BlusensFreePC10:/home/blusens# snmpstatus -v 3 -n '' -c "darfe" -u
alejandro -a SHA -A 12345678 -x DES -X 34567890 -l authPriv 192.168.1.42
  [UDP: [192.168.1.42]:161]=>[Linux ace-DarFE 2.6.35-28-generic #50-
  Ubuntu SMP Fri Mar 18 19:00:26 UTC 2011 i686] Up: 0:16:17.00
  Interfaces: 3, Recv/Trans packets: 155020/137811 | IP: 146853/136360
  1 interface is down!
```



En la imagen anterior, podemos apreciar la captura de los dos comandos ejecutados en este ejercicio, destacamos cada uno de ellos con recuadros color verde. En el primero, vemos que tanto las solicitudes como las respuestas se ven en texto plano (*1.3.6.1....*), en cambio en la segunda una vez más vemos que una vez generado el secreto compartido, a partir de la tercer trama viaja todo cifrado ("*encryptedPDU: Privkey unknown*").

En esta captura quisimos poner de manifiesto a su vez, una capacidad de SNMP que es la de poder "agrupar" varias peticiones en un solo envío, en este caso vemos claramente cómo cinco solicitudes "**snmpv1**" diferentes del árbol de la MIB, viajan en cada una de las tramas (las subrayamos en rojo). Lo mismo sucede con lo capturado en "**snmpv3**", pero por supuesto que en el envío cifrado no se puede distinguir.

18) Más ejercicios con "**snmpwalk**".

El comando snmpwalk permite obtener información de las variables de la MIB, si no ponemos opciones adicionales "camina" por toda la MIB como se muestra a continuación (pegamos solamente unas pocas líneas de la respuesta, pues es inmensa).

```
#snmpwalk -v 1 -c darfe 192.168.1.44
      iso.3.6.1.2.1.88.1.4.3.1.3.6.95.115.110.109.112.100.95.109.116.101.8
      4.114.105.103.103.101.114.82.105.115.105.110.103 = STRING:
      "_triggerFire"
      iso.3.6.1.2.1.92.1.1.1.0 = Gauge32: 1000
      iso.3.6.1.2.1.92.1.1.2.0 = Gauge32: 1440
      iso.3.6.1.2.1.92.1.2.1.0 = Counter32: 0
      iso.3.6.1.2.1.92.1.2.2.0 = Counter32: 0
      ............
      ...
```

19) Ejercicios con "**snmpgetnext**".

Por defecto el comando "**snmpgetnext**" nos dará el siguiente valor de la MIB, para ello un buen ejercicio es "caminar" por la **MIB** como se muestra a continuación:

Como podemos ver, si coloco un parámetro me responde con el siguiente:

```
#snmpgetnext -v 1 -c "darfe" localhost  sysUpTime.0
      SNMPv2-MIB::sysContact.0 = STRING: acorletti@darfe.es
```

Ahora, si vamos consultando con "**snmpgetnext**" con el valor que nos da la respuesta anterior, se pueda ir recorriendo toda la **MIB** como se presenta a continuación:

```
#snmpgetnext -v 1 -c "darfe" localhost  sysUpTime.0
      SNMPv2-MIB::sysContact.0 = STRING: acorletti@darfe.es

#snmpgetnext -v 1 -c "darfe" localhost  sysContact.0
      SNMPv2-MIB::sysName.0 = STRING: BlusensFreePC10

#snmpgetnext -v 1 -c "darfe" localhost  sysName.0
      SNMPv2-MIB::sysLocation.0 = STRING: "Madrid, Oficinas Centrales"

# snmpgetnext -v 1 -c "darfe" localhost  sysLocation.0
      SNMPv2-MIB::sysORLastChange.0 = Timeticks: (1) 0:00:00.01
```

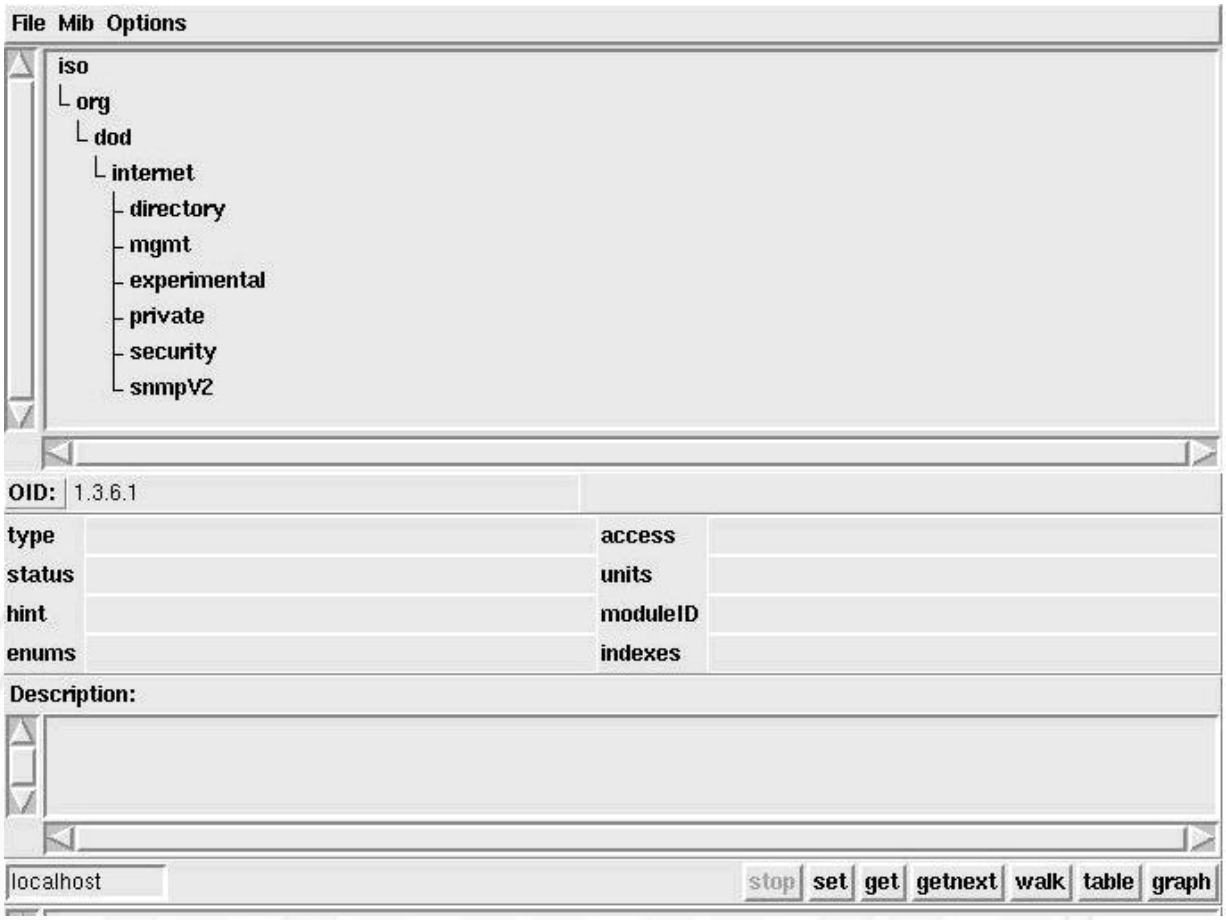
Ya que hemos trabajado con "**snmptranslate**" verifiquemos que se está siguiendo con "el próximo valor" de la MIB:

```
#snmptranslate -On SNMPv2-MIB::sysUpTime.0
      .1.3.6.1.2.1.1.3.0

#snmptranslate -On SNMPv2-MIB::sysContact.0
      .1.3.6.1.2.1.1.4.0

#snmptranslate -On SNMPv2-MIB::sysName.0
      .1.3.6.1.2.1.1.5.0

#snmptranslate -On SNMPv2-MIB::sysLocation.0
      .1.3.6.1.2.1.1.6.0
```

Como se puede verificar con las consultas anteriores, cada valor que nos respondió "snmpgetnext" era el que seguía en el árbol de la MIB.

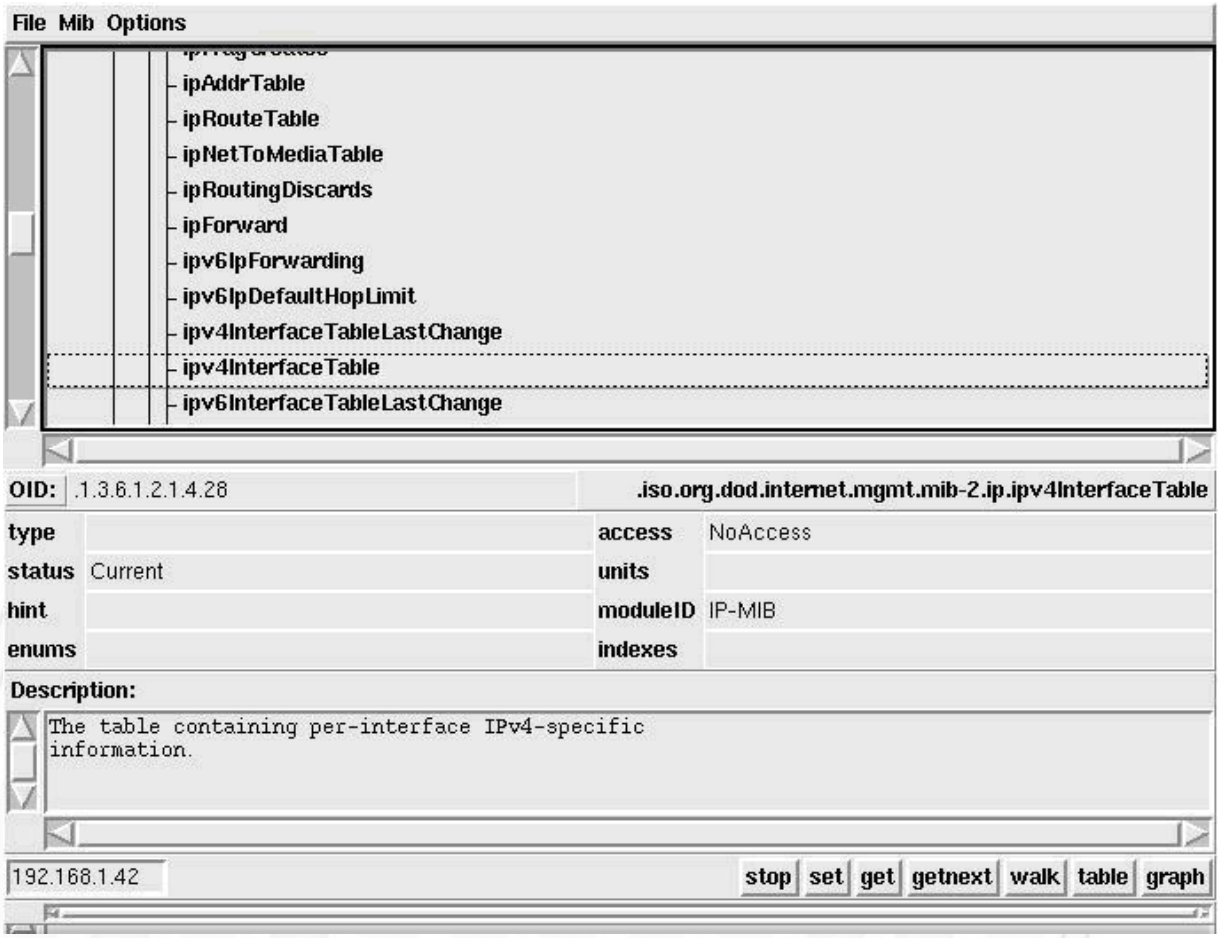
20) La interfaz gráfica "**tkmib**".

El paquete "**net-snmp**" nos ofrece de forma nativa una interfaz gráfica sencilla y bastante amigable. Para acceder a ella se ejecuta el comando "**tkmib**", automáticamente se abre una ventana como la que se presenta a continuación:

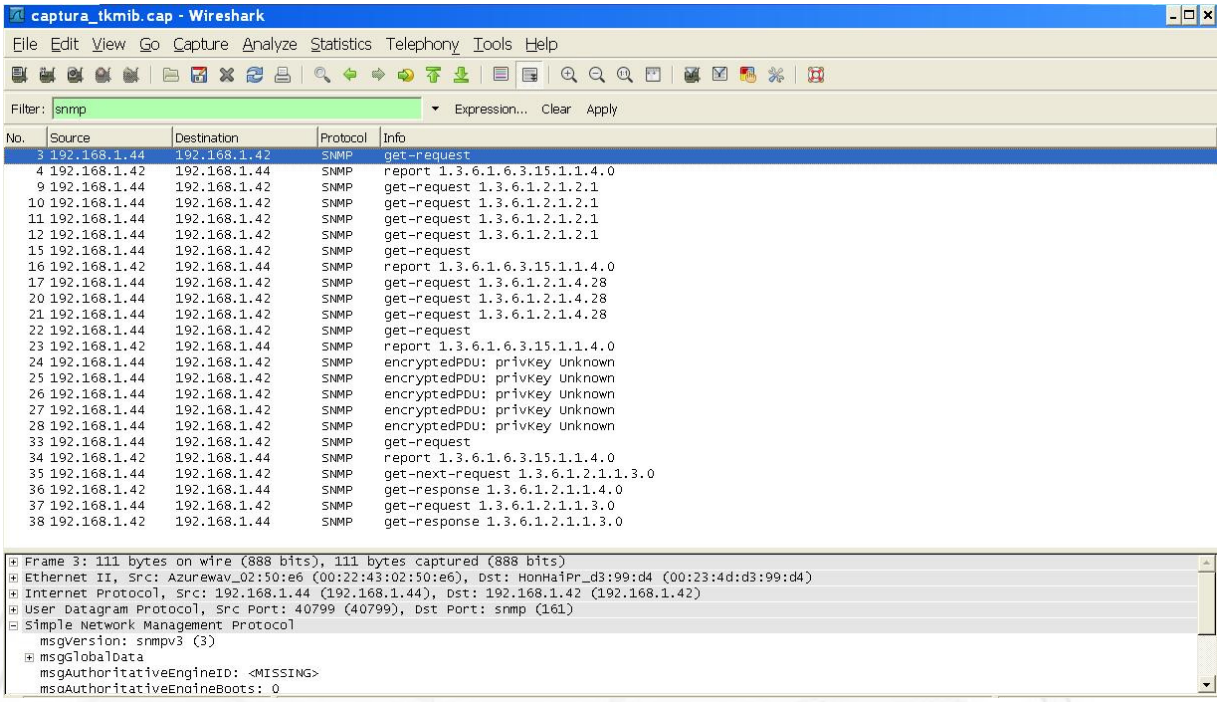


Como puedes apreciar ya nos presenta el árbol de la MIBII al cual podemos agregar todas las que deseemos. A medida que vamos desplegando el árbol (haciendo "doble clic" en cualquier nodo del mismo), podemos seleccionar el objeto concreto o rama sobre la que se la que se lanzará la consulta, a su vez en la parte inferior izquierda, se puede seleccionar a qué agente se dirigirá la misma, por defecto, tal cual puedes ver en nuestra imagen, viene configurado "**localhost**".

A continuación presentamos el árbol desplegado y el resultado de una consulta básica sobre el mismo lanzada hacia el agente 192.168.1.42:



Un aspecto muy importante de "**tkmib**" es que desde la ventana "**Options**" del menú superior, te permite configurar y ajustar al máximo de detalle todos los parámetros de SNMPv3, cosa que como mencionamos desde el principio debe ser una cuestión fundamental en toda red en producción, así que te invitamos a que lo pruebes y verifiques. Aquí abajo te pegamos una captura, justamente de la imagen anterior en la que enviamos consultas con "Priv" "auth" y "NoPriv" (que son parte de esta configuración de "options"), desde la cual puedes apreciar una vez más que está viajando información cifrada o sin cifrar, en base a lo que hayamos configurado:



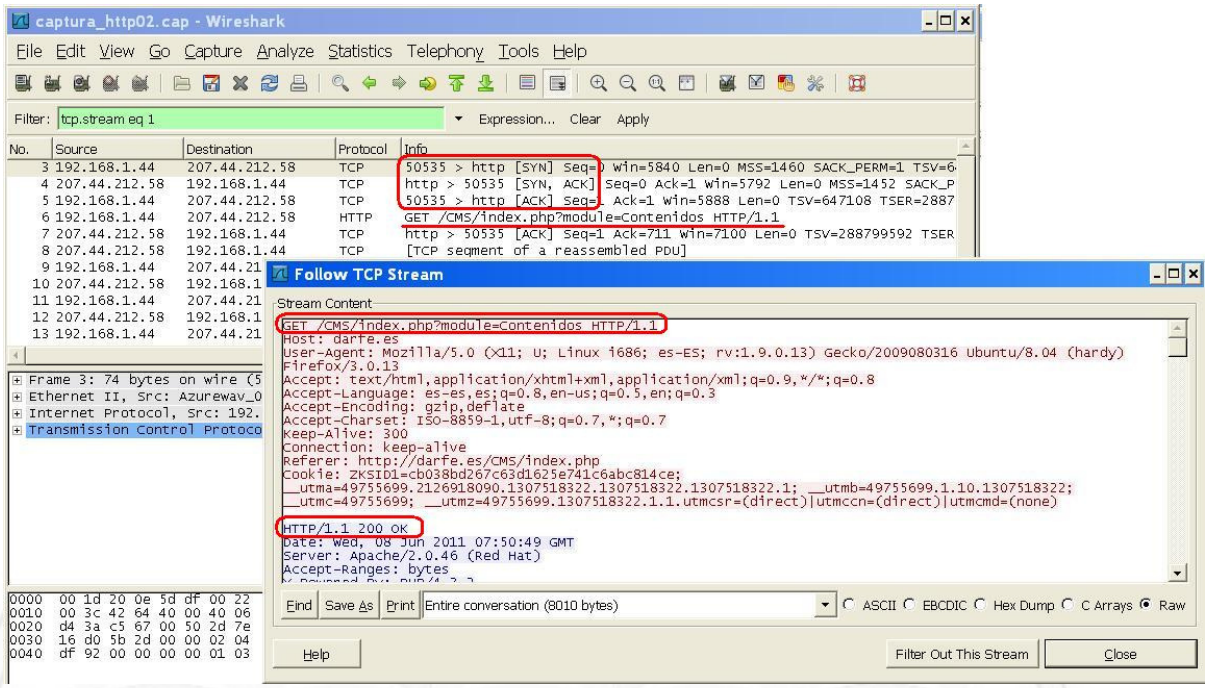
Puedes ver en la imagen anterior, diferentes valores de consultas, todas ellas lanzadas desde "tkmib" y en la columna "info" de Wireshark se ven los datos en algunos casos cifrados y en otros no. En la parte inferior, se nota que la consulta es SNMPv3.

Sobre "**tkmib**", no es necesario que nosotros te sigamos dando información, la forma más eficiente es que le dediques un tiempo y vayas probando con todas las diferentes opciones que ofrece.

7. Ejercicios con HTTP.

1) Capturar tráfico http.

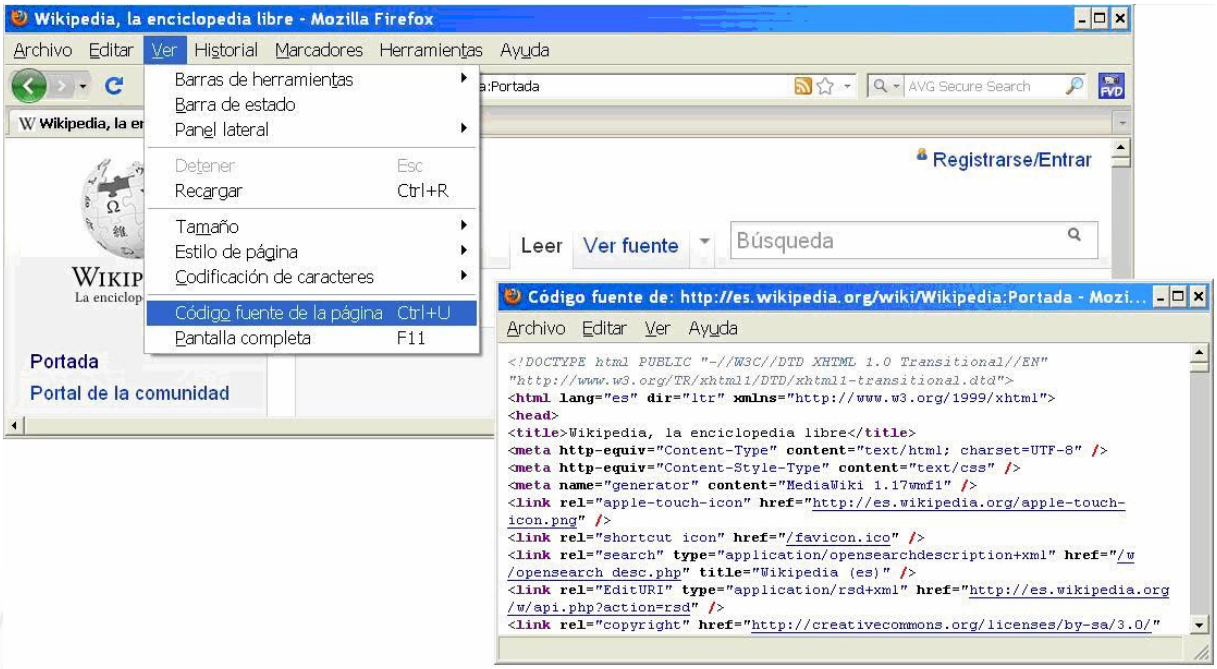
Como en los ejercicios anteriores, te proponemos que comiences aquí analizando tráfico "**http**" sencillamente realizando una conexión con cualquier navegador y capturando con "**Wireshark**", abajo te pegamos un primer ejemplo:



En la imagen anterior, vemos el triple "handshake" desde el puerto 50535 hacia el puerto 80 (que Wireshark ya lo presenta como http), luego hemos remarcado la solicitud "GET" y la respuesta con código "200 OK" siguiendo este flujo. Te invitamos a que repases la teoría en cuanto a solicitudes y respuestas, y generes diferentes tipos de ellas, siempre capturando las mismas y comparando la teoría con la práctica.

2) Código fuente de la página.

Una vez que estás conectado a cualquier página Web, lo primero que tienes a tu disposición es el código fuente, el cual en muchas ocasiones nos puede sorprender con la información que ofrece, abajo te mostramos un ejemplo:



Como puedes ver en la imagen, en este navegador, desde la opción "Ver" tienes el acceso a "Código fuente de la página" y allí se abre la segunda ventana.

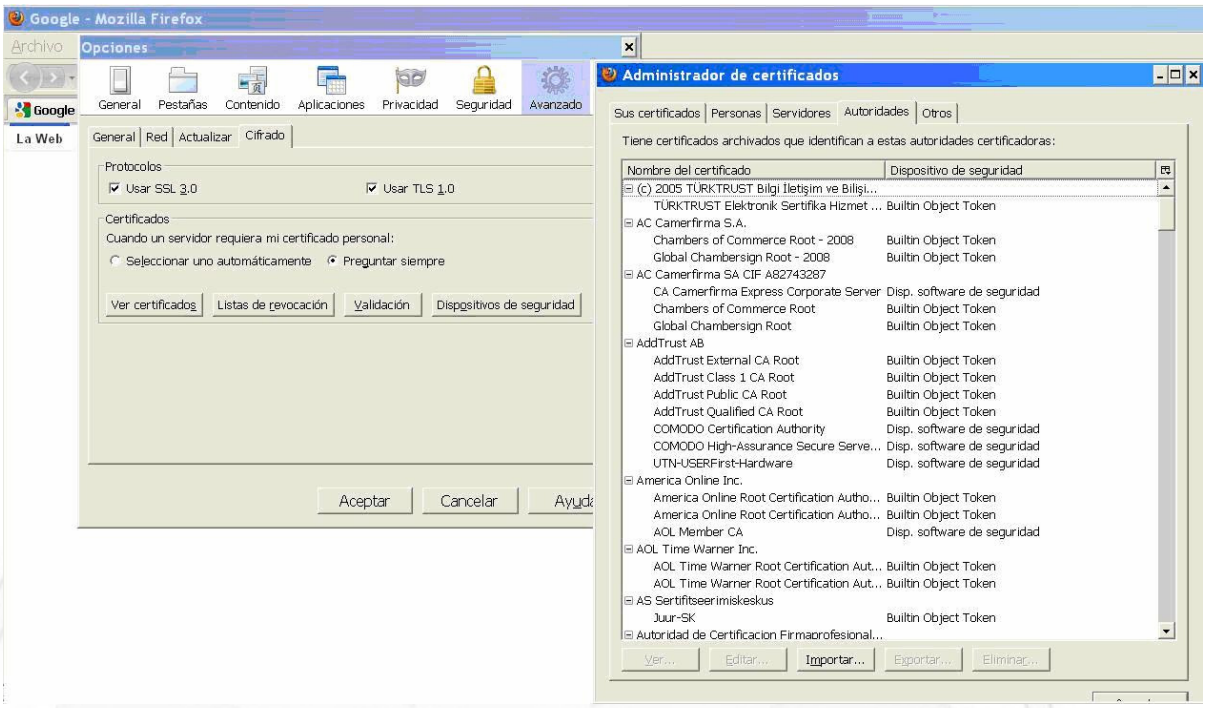
Investiga el código fuente de varias páginas analizando sus contenidos.

3) hacer seguimiento de sesiones

8. **Ejercicios con TLS**.

1) Ejercicio de conexión a una Web con certificado reconocido.

Cuando nos conectamos a un servidor seguro por "**https**", lo que estamos haciendo es aceptar el certificado que este nos presenta, si el mismo fue emitido por una entidad reconocida, entonces el trámite es transparente para el usuario, es decir, nuestro navegador tiene "pre-cargadas" las autoridades de certificación Internacionalmente reconocidas y que por supuesto han firmado convenios con ellos. Cuando instalamos Firefox por ejemplo, este ya trae incorporadas las autoridades de certificación que mencionamos, estas podemos verlas desde "*Herramientas → Opciones → Avanzado → Ver certificado*", tal cual presentamos en la imagen siguiente:



Podemos apreciar el listado de los que ya están configurados en este caso en Firefox. Casi la totalidad de ellos se encontrarán también en cualquier otro navegador. Mencionamos que si está "pre-cargado" será transparente pues no nos preguntará absolutamente nada, pero en el momento en que la página a la que queremos conectarnos, nos ofrece su certificado (el cual ha sido comprado por esa empresa a uno de las autoridades de certificación "pre-cargadas"), nuestro navegador automáticamente, se conecta a la página Web de esta autoridad de certificación validando este certificado, en realidad lo que hace es sencillamente presentar el número de serie del mismo y su "Huella digital", la cual en el caso de ser correcta, la autoridad de certificación lo reconocerá como válido y continuará la transacción con la empresa a la cual nos estemos conectando, sin que el usuario se haya percatado de ninguno de estos pasos.

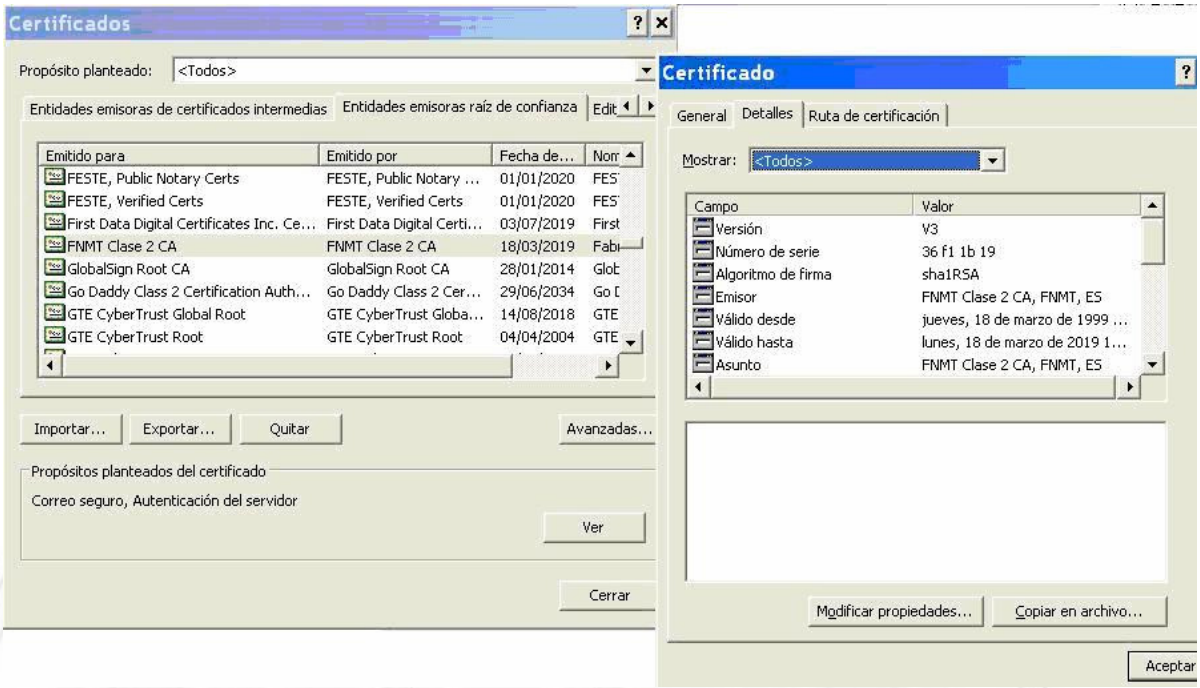
Te invitamos a que realices esta conexión a cualquier página segura, y que la captures, siguiendo esta secuencia.

2) Conexión a una página con un certificado no reconocido.

El segundo ejemplo es cuando nos conectamos por primera vez a una página que posee un certificado emitido por una entidad que no se encuentra "pre-cargada" en nuestro navegador.

En este caso nos presentará (sólo la primera vez) una ventana informándonos que este certificado no tiene quien lo valide, y si deseamos aceptar o no (dependiendo del navegador el mensaje puede tener distinta forma, pero el mensaje que nos transmite es el mencionado), a partir del momento que lo aceptamos, ese certificado queda cargado en nuestro navegador.

A continuación presentamos por ejemplo un certificado emitido por la Fábrica Nacional de Monedas y Timbres (FNMT) de España, que ya ha sido incorporado a nuestro navegador.



Este certificado no viene "pre-cargado", pero una vez que lo aceptamos ya queda incorporado a nuestro navegador, y a partir de ese momento no nos volverá a preguntar nada más en ninguna de las siguientes conexiones. Por supuesto que si la primera vez no aceptáramos estos certificados, la conexión no seguiría hacia delante, es decir no nos podríamos conectar.

Te proponemos que navegues hacia algunas páginas seguras de este tipo y permitas la incorporación de estos certificados y también que rechaces alguno, verificando cada caso.

También sería importante que captures tráfico para verificar el establecimiento del canal seguro, los pasos y la criptografía que se aplica para la transmisión.

3) **Validarse como usuario reconocido**.

Hasta ahora en los dos casos planteados, es el servidor el que nos está garantizando que "**es quien dice ser**" y luego que aceptamos que verdaderamente lo sea, se establece un canal seguro, por medio de "compartir un secreto", y toda la transmisión se realiza de forma cifrada. Pero en ninguno de ellos, nos hemos presentado nosotros como usuarios, es decir, el servidor en los casos anteriores no nos ha pedido ningún tipo de "aval" para reconocer quines somos.

Para este caso, existen dos posibilidades:

- Que la organización responsable del servidor, tenga algún mecanismo para haberme reconocido con anterioridad y permitirme el acceso mediante cualquier método de autenticación (password, token, par de claves, etc).
- Que no me conozca.

El primero de los casos es sencillamente un tema entre ambas partes. Puedo ser un cliente de ese Banco al cual le enviaron por correo postal el mecanismo, usuario y contraseña de

acceso; puedo ser un empleado, partner o proveedor de esa empresa, y me han entregado un token para que acceda; puedo ser un cliente habitual registrado previamente al cual le han pasado por teléfono o mail un sistema de autenticación, etc...

El segundo caso es el que deseamos centrarnos, pues la empresa no tiene la menor idea si verdaderamente soy yo el que se está haciendo presente o no. En este caso, nuevamente entra en juego el "**tercero de confianza**", que no deja de ser una especie de "notario virtual" que da constancia que el que se está presentando soy yo y no otro. Pueden haber varias posibilidades, pero las dos más cercanas (y gratuitas), en el caso de España, son a través del DNI electrónico, o de un certificado emitido por un tercero confiable, que nuevamente en el ámbito Español está reconocida la FNMT. Cualquier persona física en España puede obtener gratuitamente un certificado emitido por esta entidad. Su URL es: http://www.cert.fnmt.es/, y como puedes ver abajo te permite obtener un certificado, cuentes o no con el DNI electrónico.

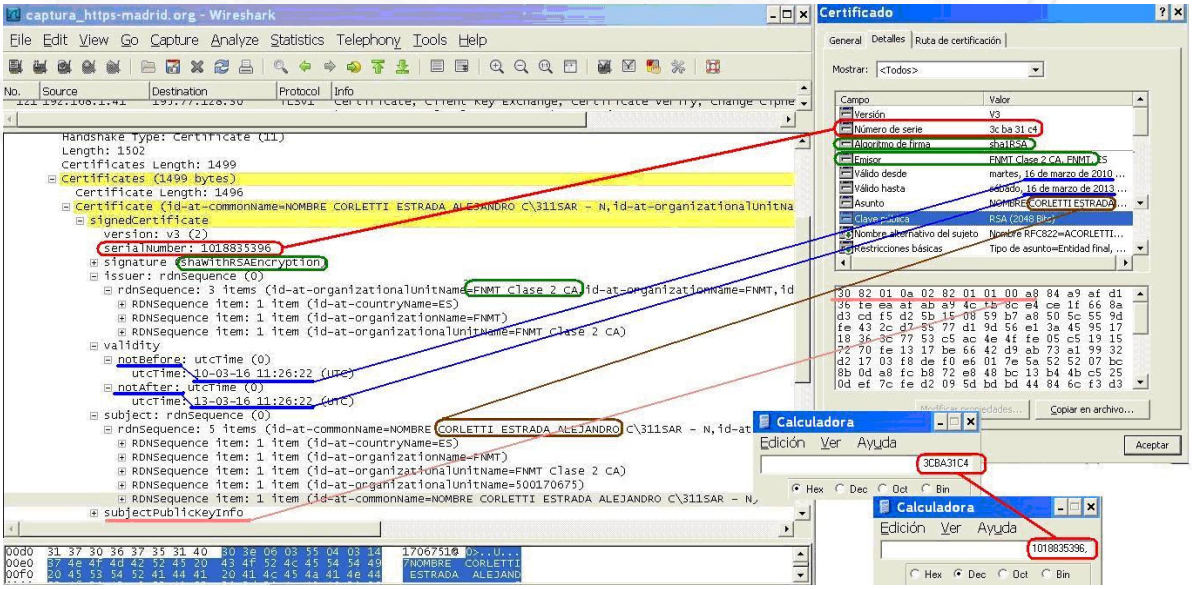
Aquí te pedimos que prestes mucha atención, pues si no presentas el DNI electrónico, ¿Cómo saben en definitiva que eres tú y no otra persona el que está solicitando este certificado?, pues ten en cuenta que una vez que te emitan este certificado, podrás hacer cualquier trámite vía electrónica en España y tendrás los mismos derechos que cuando los haces "por ventanilla"...

El mecanismo funciona muy bien, pues una vez que comienzas con el trámite, la Web de la FNMT te adjudica un número o código, con el mismo te debes hacer presente físicamente y con tu DNI en cualquiera de las reparticiones públicas que tienen implantado este mecanismo, en la misma página Web, te lo indica y puedes buscar la más cercana a tu domicilio.

Como puedes apreciar en la imagen anterior debes "acreditar tu identidad". Este tema es muy importante que lo comprendas bien, pues de no "acreditarte" no existe ninguna forma con la que pueda una entidad oficial avalar que realmente eres tú y no otro.

Una vez que te presentas en cualquiera de estas dependencias, una persona que tiene su ordenador conectado al sistema de la FNMT, te pide el DNI y el código que te han adjudicado anteriormente, verifica estos datos y te "valida en el sistema" (que en definitiva te está reconociendo como que "eres quien dices ser"), inmediatamente DESDE EL MISMO ORDENADOR y con EL MISMO NAVEGADOR con que iniciaste el trámite, puedes descargar tu propio certificado, el cual te aconsejamos que inmediatamente hagas una copia en más de un sitio seguro para ti, pues si lo pierdes deberás obtener otro, revocando el anterior.

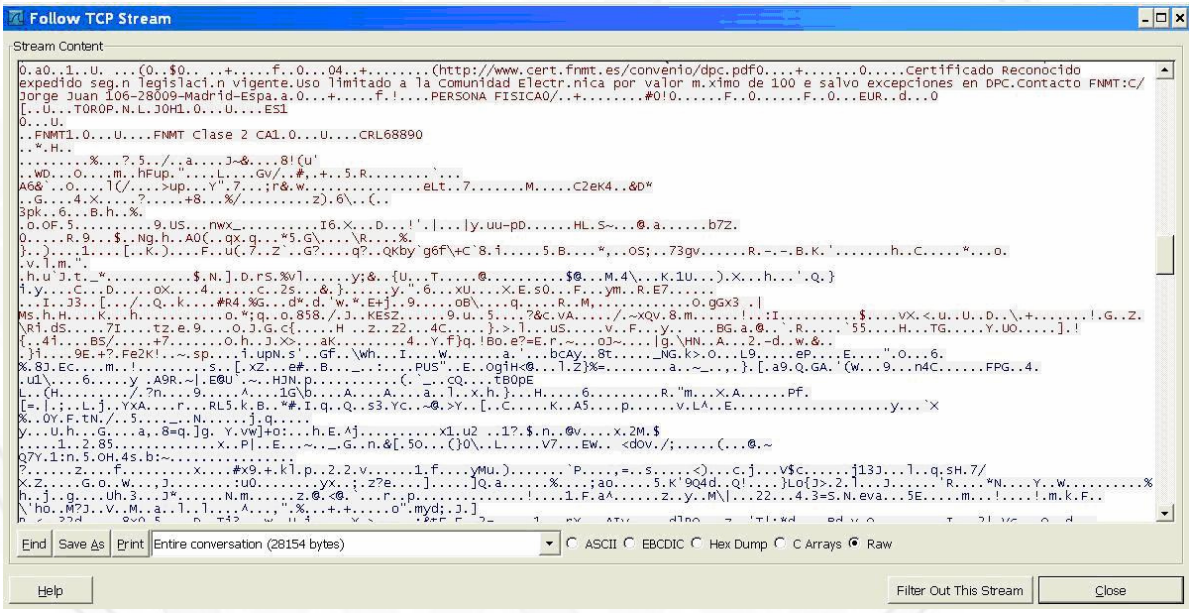
A continuación te presentamos una imagen de una conexión a través de este tipo de certificados, a una página oficial, en este caso www.madrid.org.



En la imagen anterior, estamos viendo ventanas totalmente diferentes: a la izquierda una captura de este tráfico con **Wireshark**, a la derecha y arriba, el **certificado** personal emitido por la FNMT, abajo por último dos imágenes de la **calculadora** (resaltadas en rojo) para que puedas verificar que el número de serie que nos presenta en "hexadecimal" la ventana del certificado es el mismo que el que nos presenta en "decimal" Wireshark.

En la imagen también puedes verificar (resaltado en verde que el emisor es: FNMT (figura CA, por Autoridad de Certificación) y que es un certificado clase 2, con algoritmo RSA y resumen SHA1. En azul subrayamos la fecha de emisión y vencimiento (16 de marzo de 2010 hasta el 16 de marzo del 2013). En marrón a quién fue emitido (Alejandro Corletti Estrada), y por último que no entró en la imagen de Wireshark, pero vendría inmediatamente abajo el valor de la "clave pública" emitida para este certificado, la cual será con la que se establecerá toda la criptografía y firma electrónica (para que valide el servidor) de los documentos que se vayan a presentar en esta Web.

Por último, si seguimos el flujo TCP de esta conexión desde Wireshark, podemos apreciar que una vez que se hizo presente este certificado, todos los datos viajan cifrados, como puedes apreciar en la imagen que sigue:

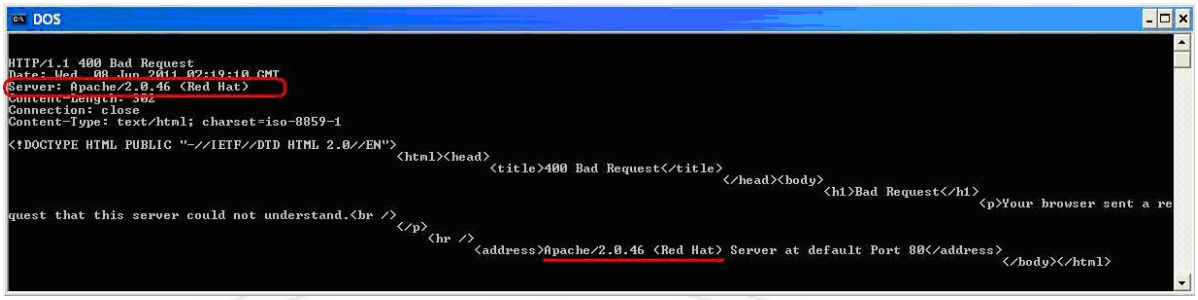


Te invitamos a que generes tu propio certificado ante la FNMT y que ejercites todos estos pasos mencionados, como siempre capturando el tráfico, y comparando la práctica con el desarrollo teórico.

4) Un ejercicio sencillo es la obtención de la información del servidor donde está alojado esa aplicación Web, puedes hacerlo a través de los siguientes pasos:

- Ping a un servidor Web www.cualquiera.com
- registramos su dirección IP
- ejecutamos: "**telnet dirección_IP 80**"
- Al producirse la conexión, escribimos: "**get http /1.1**" (y pulsamos 2 veces ENTER)

- Nos mostrará toda la información de ese servidor.....



Como puedes ver en nuestro ejemplo se trata de un sistema operativo "**Linux**" con distribución "**Red Hat**", y el servidor Web es "**Apache 2.0**".

9. Ejercicios con netBIOS.

1) Como siempre hemos hecho, te invitamos a que comiences estos ejercicios con capturas de tráfico y análisis de las mismas con Wireshark, particularmente centrando la atención en los siguientes puertos:

   - UDP port 137 (name services)
   - UDP port 138 (datagram services)
   - TCP port 139 (session services)
   - TCP port 445 (Windows 2k en adelante)

2) Captura y analiza la información que te ofrecen los siguientes "décimo sextos" caracteres del protocolo Netbios. Verifica de qué tipo de host o server se trata, que servicios ofrece, por qué se hace presente en la red, con quién establece conexiones o dialoga, etc…

   - [03h] Messenger Service
   - [06h] RAS Server Service
   - [1Fh] NetDDE Service
   - [20h] Server Service
   - [21h] RAS Client Service
   - [BEh] Network Monitor Agent
   - [BFh] Network Monitor Application
   - [03] Messenger Service
   - [1Dh] Master Browser
   - [1Bh] Domain Master Browser

   Nombres de grupo

- [10h] Domain Name
- [1Ch] Domain Controllers
- [1Eh] Browser Service Elections

3) Captura solicitudes y respuestas WINS y compáralas con lo que tratamos en la teoría.

4) investiga y emplea las diferentes opciones de comandos "net" (help, view, use.....).

5) capturar establecimiento de sesiones "netBIOS", confrontar las capturas con las tramas descriptas en la teoría.

6) Capturar tráfico de "logon Windows" Repasar protocolo "nodo".

EJERCICIOS CON HERRAMIENTAS

1. Ejercicios con DNS:

1) **Nslookup**: Es un cliente DNS que sirve para obtener direcciones IP a través del dominio y viceversa (Ej: nslookup www.google.es o nslookup 209.85.229.104).
2) **Dig**: (Domain Information groper) otro comando flexible para interrogar DNSs
3) **host**: Sencilla
4) **BIND** (Berkeley Internet Name Domain).

   BIND es el servidor DNS más difundido en el mundo, y sobre el que se sustentan la inmensa mayoría de los servidores que implantan este protocolo. Es un software de libre difusión, que en la actualidad lo administra el Internet System Consortium.

   En este ejercicio lo damos por instalado en cualquier arquitectura Linux (no tiene ninguna dificultad). En el caso de familia Debian, encontrarás los archivos de configuración en "/etc/bind/" y en "/var/cache/bind/" es donde se guardan todos los valores con los que se encuentra trabajando una vez iniciado.

   Te proponemos que investigues en este ejercicio los siguientes archivos:

   - "**named.conf**" y sus variantes.

- "**zones.rfc1918**": en este archivo se guardan las direcciones IP privadas que tratamos en el capítulo de red y que las regula justamente la **RFC-1918**.

Por defecto al instalar BIND, se desempeñará como servidor caché, es decir que deberá consultar a otros servidores DNS para responder a sus peticiones. El próximo ejercicio que te planteamos es ¿Desde dónde debes configurar este BIND para que pueda responder una petición cliente? (es decir ¿dónde le decimos cuáles son los servidores DNS que nuestro BIND debe consultar?), deja nuestro BIND apuntando a uno de ellos, puede ser el que nos ofrece nuestro ISP o cualquiera que conozcas de Internet.

Te planteamos una inquietud: ¿Para qué piensas que puede servir instalar un servidor caché dentro de tu red local?.....(una pista pasa por las consultas UDP... ¿Te acuerdas que mencionamos que UDP es un protocolo "peligroso"...).

Las consultas cliente, es decir, apuntar un cliente hacia este servidor, es un tema que seguramente ya conocerás y lo habréis hecho (en Windows desde "propiedades de red", y en Linux, desde el archivo "resolv.conf"), apunta algún cliente hacia este BIND.

¿Cuánto tiempo tarda nuestro BIND en realizar una respuesta?, ¿Y si la reiteras, cuánto tarda ahora?, ¿Por qué cambió este valor?

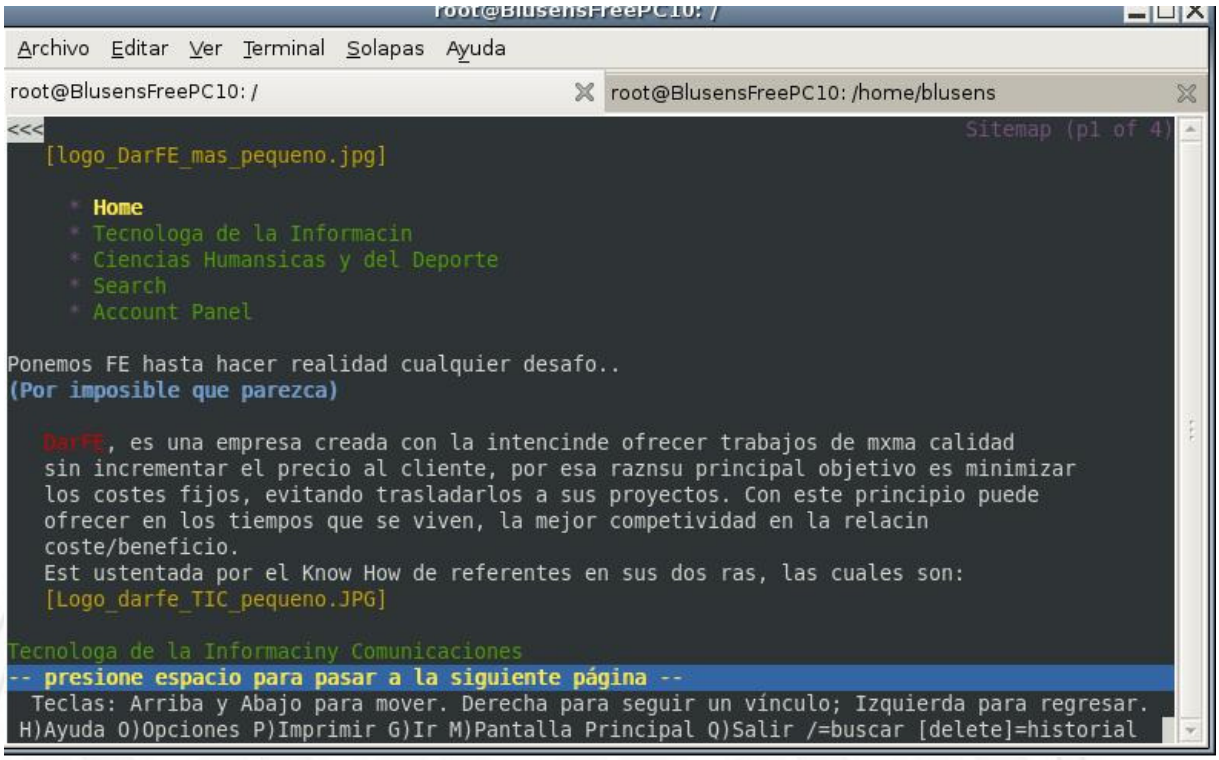
Navega por Internet desde el cliente por algunas páginas. Ahora analiza cómo está nuestro directorio "/var/cache/bind/", y en "/var/log" ¿Encuentras algo relacionado a este servicio BIND?

El resto de la configuración de BIND te lo dejamos más adelante como "Desafío".

2. Ejercicios con HTTP.

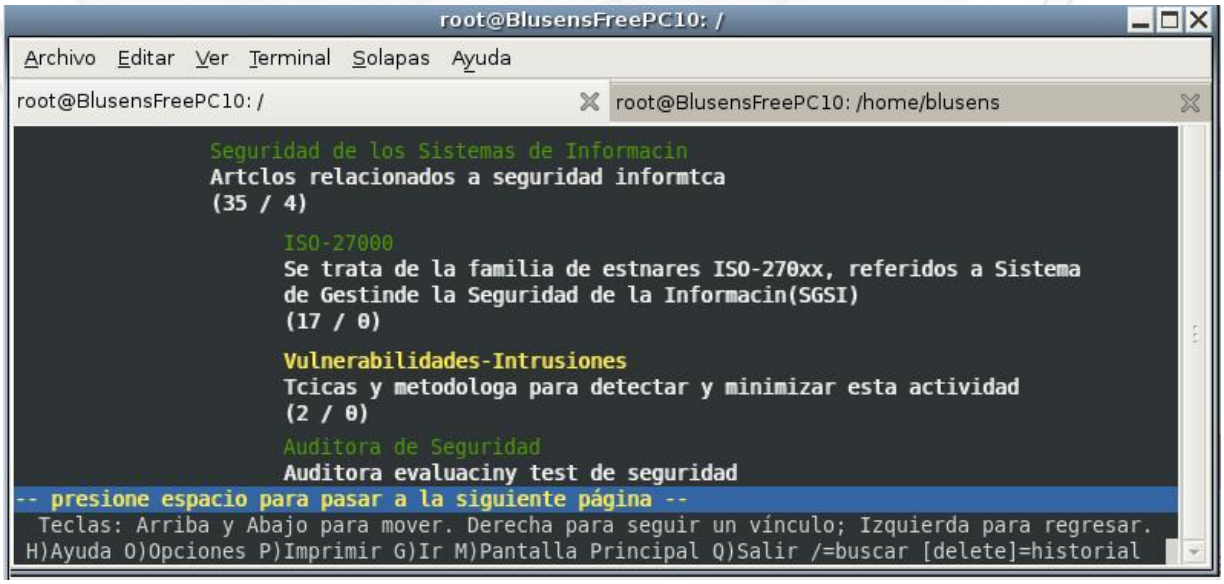
1) Herramienta "**lynx**".

Esta herramienta es un navegador Web por línea de comandos, por supuesto que no te pedimos que la emplees hoy para navegar, pero sí te invitamos a que la investigues, pues justamente por la razón de su relación directa con la programación en bash, es que con pequeños programas (o scripts) podrás llegar a sacarle muchísimo provecho (tal mucho más de lo que te imaginas...)

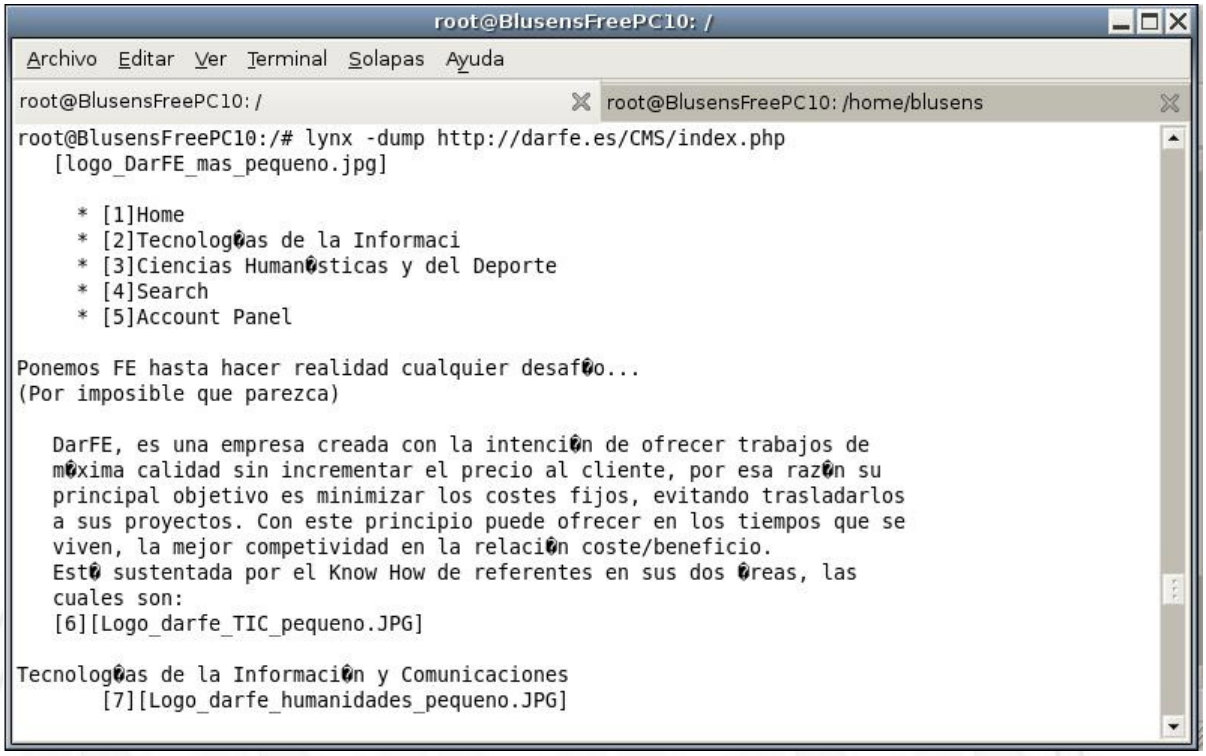


En la imagen anterior, puedes ver una conexión a nuestra web, y cómo te presenta el menú de la misma bajo una sencilla forma de navegación dentro de la web.

Puedes ir navegando por sus páginas e interactuar, por ejemplo descargando cualquier contenido que desees. Abajo te presentamos la sección descargas de nuestra Web y cómo esta herramienta te permite llegar hasta ella y descargar un archivo a través de las "flechas" de navegación de nuestro teclado.



Por último te presentamos alguna de sus opciones "potentes" como es el caso de "-dump" que te ofrece la opción de descargar el contenido de cualquier parte del "árbol" de una Web, a través de línea de comandos:



2) Herramienta "**wget**"

"wget" tal vez sea una de las mayores herramientas para trabajar con http, por defecto viene instalado en la mayoría de las distribuciones Linux, sino no tendrás ningún problema para encontrarlo e instalarlo, se opera desde línea de comandos.

Como puedes ver en la imagen anterior, su empleo es sencillo y te ofrecerá un sinnúmero de posibilidades, te proponemos que investigues su empleo con recursividad "**-r**", "**-i**" para ficheros, "**-l**" para profundidad, "**-o**" para salidas, etc.

3) herramienta "**tcpxtract**".

Esta herramienta nos permite reconstruir el tráfico capturado y realizar el seguimiento de la sesión TCP. El ejercicio que te invitamos a hacer es lanzar el analizador de protocolos, capturar tráfico con Wireshark, abrir un navegador cualquiera desde la máquina local o cualquier otra que se encuentre en ese "ámbito de captura", guardar lo capturado en formato "pcap" y luego con esta herramienta reconstruir lo que se ha capturado, para poder verificar la navegación que se ha realizado en esa sesión.

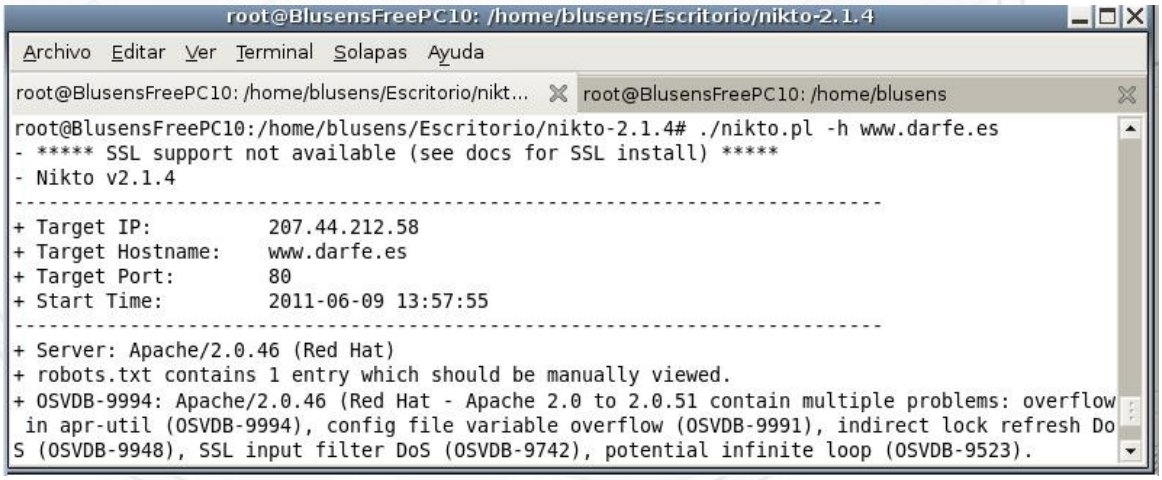
Capturar, guardar y reconstruir sesiones TCP de distinto tipo.

4) Herramienta "**nikto**"

Nikto es un scanner de vulnerabilidades gratuito bajo licencia GPL, que emplea el protocolo http y https para la búsqueda de debilidades en entornos Web. Lleva muchos años en el mercado, por lo que actualmente se encuentra bastante maduro tanto en su eficiencia como en su amigabilidad. Su instalación es sumamente sencilla (apt-get install nikto, en entornos Debian), una vez instalado ya está "preconfigurado" para funcionar, aunque aconsejamos que veas su configuración en "**/etc/nikto/nikto.txt**".

- El primer ejercicio será muy sencillo, simplemente puedes ejecutar:
  - "#nikto.pl -h Direccion_IP_del_host_destino"

  La opción "-h" indica el host que se atacará pudiendo poner su dirección IP o también su URL (es indistinto), el llamado "nikto.pl" puede ser también "nikto" dependiendo de cómo tengas configurado "perl" y los permisos de ese archivo.



- Si el servidor Web, se encuentra configurado para trabajar sobre otro puerto que no sea el 80 (que es el que por defecto emplea nikto), se puede también atacar a cualquier puerto que se desee con el comando:
  - "#nikto.pl –h URL_del_host **–p** puerto_deseado"
- Ejercicio 1: Investiga cómo puedes hacer para escanear más de un puerto con una sola instrucción.
- Ejercicio 2: Investiga cómo puedes hacer para escanear más de un host con una sola instrucción.
- Ejercicio 3: Investiga cómo se hace un escaneo exclusivamente para SSL o TLS.

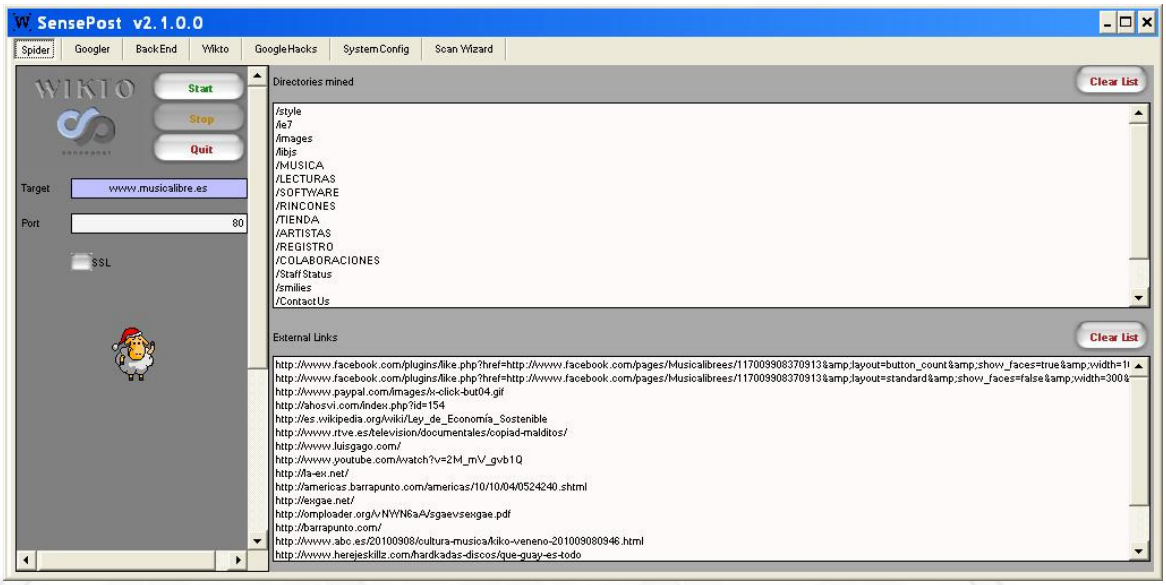
- Ejercicio 4: ¿Puedes trabajar a través de un proxy?
- Ejercicio 5: ¿Qué sucedería si deseas trabajar con diferentes archivos de configuración?, ¿Puedes usar más de uno?, ¿Cómo los llamarías?
- Ejercicio 6: Investiga cómo puedes integrarlo con "Nessus".
- Ejercicio 7: ¿Qué tipo de opciones te permiten interactuar con nikto mientras este se ejecuta?
- Ejercicio 8: continúa probando las opciones "-verbose, -format, -output, -cookies".

5) Herramienta **"wikto".**

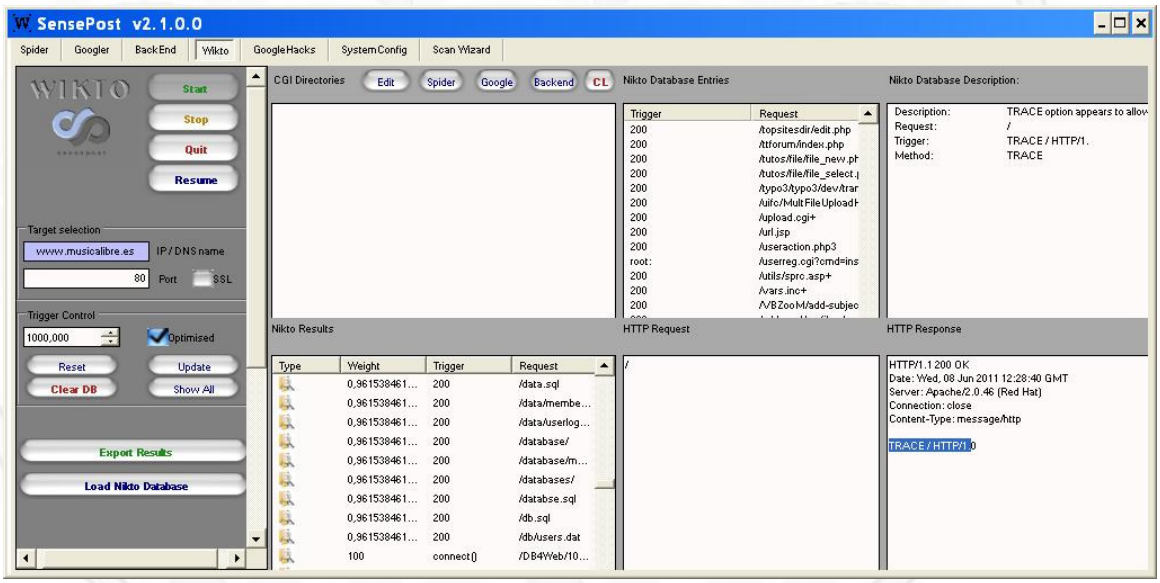
Wikto es una herramienta que deriva de "nikto", empleando la misma base de datos de vulnerabilidades. Está escrita en lenguaje "**.NET**" y que opera bajos entornos Windows, está desarrollado por la empresa **SensePost.com** y lo distribuye de forma gratuita previo registro en su Web. Actualmente podemos asegurar que es muy eficiente y como veremos a continuación ofrece una interfaz gráfica muy amigable.

Una vez instalado queda listo para ser empleado, la primera pantalla que podemos tener en cuenta es justamente la selección de la Web que se desea escanear, para ello sencillamente ponemos su URL y el puerto que queremos atacar, por último nos ofrece la posibilidad de hacerlo a través de un proxy, seleccionando su dirección IP. Abajo presentamos esta pantalla.

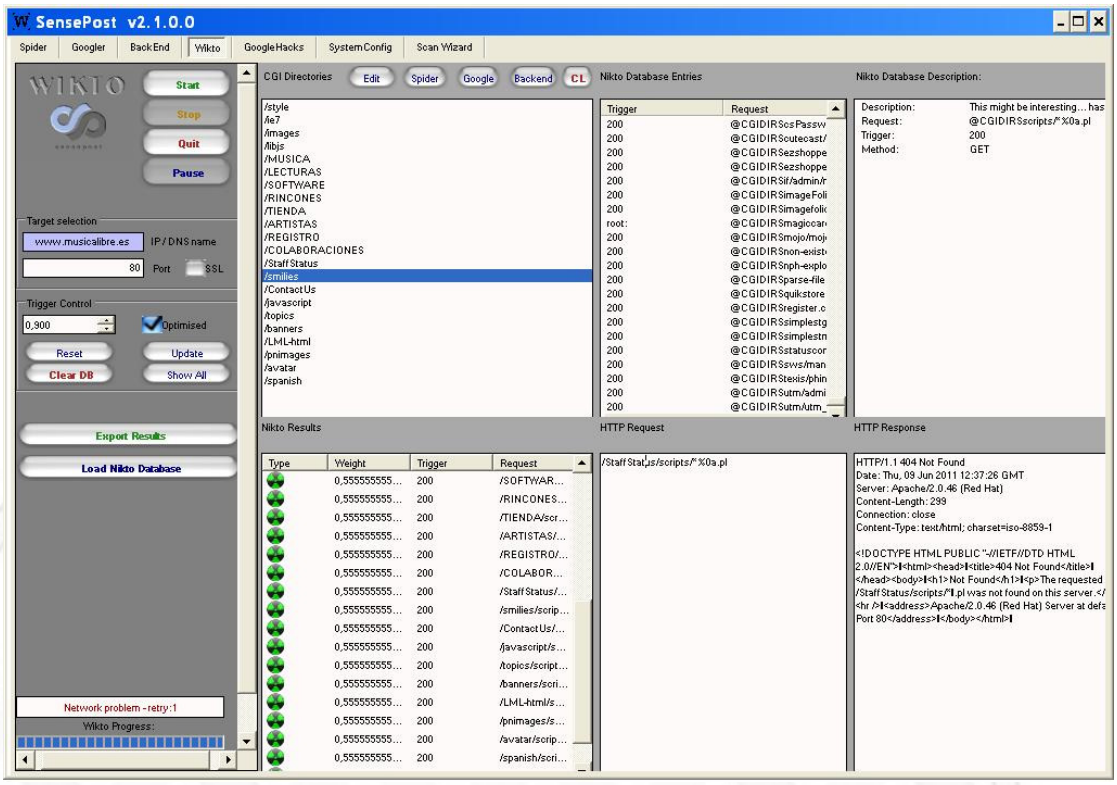
Una vez seleccionado estos campos, sólo nos queda confirmar algún parámetro de configuración (opcional), e inmediatamente si continuamos ("next") con el proceso, se abre la ventana que presentamos abajo, donde al presionar "**Start**" da comienzo al scan, comenzando a presentar inmediatamente los resultados del mismo.



Una vez finalizado, en la ventana "Wikto" del menú superior, nos presentará todas las debilidades encontradas.



Por último, seleccionando cualquiera de ellas nos desplegará todo el detalle necesario de cada una de ellas:



- Ejercicio 1: Investiga el empleo del botón "**Spider**".
- Ejercicio 2: Investiga el empleo del botón "**Googler**".
- Ejercicio 3: Investiga el empleo del botón "**BackEnd**".
- Ejercicio 4: Investiga los formatos que ofrece para exportar la información obtenida.

3. Ejercicios con Nessus.

Instalación de Nessus (o si se prefiere OpenVAS) con Debian.

http://www.nessus.org/products/nessus/select-your-operating-system

Seleccionamos:

Debian 5.0 (32 bits):

Nessus-4.4.1-debian5_i386.deb (12393 KB)

Descargamos el paquete anterior y luego ejecutamos, desde el directorio en el cual lo descargamos:

```
#dpkg -i Nessus-4.4.1-debian5_i386.deb
```

Luego debemos registrarnos en:

http://www.nessus.org/products/nessus/nessus-plugins/obtain-an-activation-code

(Se rellenan los campos, utilizaremos la opción "home" que es gratuita, y el código se envía por mail a la dirección que se rellene allí)

Luego se debe ejecutar el siguiente comando para adjuntar el número de registro:

```
/opt/nessus/bin/nessus-fetch --register <Código de activación>
```

Si todo está en orden nos mostrará un mensaje como el que figura a continuación y comenzará a actualizar los "plugins" (como son más de 40.000 puede demorar unos minutos), luego nos presentará un mensaje similar al siguiente:

```
Your activation code has been registered properly - thank you.
Now fetching the newest plugin set from plugins.nessus.org...
```

Ya se encuentra activado y actualizado Nessus, ahora el primer paso para su ejecución es la creación del usuario de administración, para ello ejecutamos:

```
/opt/nessus/sbin/nessus-adduser
```

Nos irá guiando con algunas preguntas:

```
Login : admin
Login password :
Login password (again) :
Do you want this user to be a Nessus 'admin' user ?
(can upload plugins, etc...) (y/n) [n]: y

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that admin has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser manual for the rules syntax

Enter the rules for this user, and enter a BLANK LINE once you are done :
(the user can have an empty rules set)

Login           : admin
Password        : ***********
This user will have 'admin' privileges within the Nessus server
Rules           :
Is that ok ? (y/n) [y] y
User added
```

Una vez finalizado debemos iniciar (o reiniciar el demonio nessus server):

/etc/init.d/nessusd restart (o start si aún no está iniciado)

"**nessusd**" es el proceso servidor, este se rige por lo que establezca el archivo "**nessusd.conf**", el cual en la instalación por defecto, se encuentra en:

/opt/nessus/etc/nessus#

Para verlo podemos ejecutar, por ejemplo: "#vi nessusd.conf"

Lo primero que nos interesa observar, es que por defecto este demonio, queda configurado para que el "cliente" se conecte por https al puerto 8834, esto podemos verlo en las siguientes líneas de "nessusd.conf":

```
# Port for the Nessus Web Server to listen to (new XMLRPC protocol) :
xmlrpc_listen_port = 8834
```

(También deja abierto el puerto 1241, para clientes "viejos" de nessus, se puede ver unas líneas más debajo de las que acabamos de presentar)

Por lo tanto, si deseamos conectarnos a ese servidor, deberemos abrir una sesión https a la dirección IP del servidor y a ese puerto. Si estamos en local, podemos hacerlo con:

https://127.0.0.1:8834/, (si es en forma remota será https://dir_IP_Servidor:8834)

Nos pedirá usuario y contraseña, los cuales serán los que acabamos de crear con el comando "**nessus-adduser**"

Podemos seleccionar **"Scan" --> add** (y colocando la IP deseada es posible lanzar un escaneo hacia ella), nos pedirá qué política aplicar.

1) Formas de operar:

a. desde consola:

Práctica a realizar:

Consola: empleo de las opciones "**-a, -c, -p, -D, -v, -h**"

Ejemplo:

./nessusd -v -a 10.64.130.195 -D (ejecuta el demonio en background)
Verificación de la ejecución del demonio (ps -ef).

./nessus -q localhost 1241 nessus nessus host_dest_nessus_ejemplo resultados (ejecuta el cliente en bacground, se conecta al servidor localhost por el puerto 1241, USER:nessus, PASSWORD:nessus, obtiene los rangos a monitorizar

de host_dest_nessus_ejemplo y guarda los resultados en el archivo "resultados" con formato estándar.

./nessus -q localhost 1241 nessus nessus host_dest_nessus_ejemplo resultados -c /usr/local/etc/nessus/nessusd.conf -T text (idem anterior, pero especificando el archivo de configuración y con salida en formato texto)

b. interfaz gráfica:

Práctica a realizar:

- Activación de Interfaz gráfica.
- Actualización de plugins.
- Selección de plugins.
- Selección de Scan.
- Determinar el lanzamiento de plugins puntuales.
- Analizar cuando "Conecta" y cuando "no" un plugins.
- Entender los formatos de ".NASL" (en los plugins).
- Determinar cuál es el patrón que busca o que lanza.
- Visualización de informes en diferentes formatos.

En la parte de desarrollo teórico hemos presentado bastante sobre la interfaz gráfica en el modo "cliente", así que ya estás en condiciones de realizar perfectamente todas las tareas que te proponemos en los párrafos anteriores, así que dejamos en tus manos este desafío para que lo practiques e investigues, pues seguramente no encontrarás ningún inconveniente en realizarlo.

4. Ejercicios con Snort:

En los ejercicios que presentamos a continuación, hemos presentado un archivo "snort.conf" que se corresponde con la fecha de este texto, pero como el mismo varía constantemente, es muy probable que encuentres nuevas opciones, variables, parámetros, etc… Nuestra intención es que a través de estos ejercicios, comprendas la lógica de su funcionamiento para que puedas estar en capacidad de configurar cualquier nueva línea que aparezca a futuro.

1) Variables:

Práctica a realizar:

Configuración de : var HOME_NET, ETERNAL_NET y algunos SERVERS.

En la sección variables (del archivo "Snort.conf") es posible agrupar varios tipos de elementos bajo el concepto de variables, las cuáles podrán emplearse luego para cualquier otra opción de configuración.

La sintaxis de las variables es: "**var <nombre_de_variable> <valor>"**

Para ser estrictos, se pueden emplear dos tipos de variables:

- Variables estáticas: son las que detallan rangos de direcciones.
- Variables dinámicas: Son las que tienen como <valor>, el nombre de una o varias variables estáticas, precedidas por el signo "$".

A continuación se presenta esta sección tal cual es en el archivo "snort.conf" y luego se describe cada variable.

```
###################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect
# your local network. The variable is currently
# setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
#
var HOME_NET 10.64.130.6/32
#
# or use global variable $<interfacename>_ADDRESS
# which will be always initialized to IP address and
# netmask of the network interface which you run
# snort at.  Under Windows, this must be specified
# as $(<interfacename>_ADDRESS), such as:
# $(\Device\Packet_{12345678-90AB-CDEF-1234567890AB}_ADDRESS)
#
# var HOME_NET $eth0_ADDRESS
#
# You can specify lists of IP addresses for HOME_NET
# by separating the IPs with commas like this:
#
# var HOME_NET [10.1.1.0/24,192.168.1.0/24]
#
# MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
# or you can specify the variable to be any IP address
# like this:

#var HOME_NET any

# Set up the external network addresses as well.
# A good start may be "any"

var EXTERNAL_NET any
```

```
# Configure your server lists.  This allows snort to only look
for attacks
# to systems that have a service up.  Why look for HTTP attacks
if you are
# not running a web server?  This allows quick filtering based
on IP addresses
# These configurations MUST follow the same configuration scheme
as defined
# above for $HOME_NET.

# List of DNS servers on your network
var DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
var SMTP_SERVERS $HOME_NET

# List of web servers on your network
var HTTP_SERVERS $HOME_NET

# List of sql servers on your network
var SQL_SERVERS $HOME_NET

# List of telnet servers on your network
var TELNET_SERVERS $HOME_NET

# Configure your service ports.  This allows snort to look for
attacks
# destined to a specific application only on the ports that
application
# runs on.  For example, if you run a web server on port 8081,
set your
# HTTP_PORTS variable like this:
#
# var HTTP_PORTS 8081
#
# Port lists must either be continuous [eg 80:8080], or a single
port [eg 80].
# We will adding support for a real list of ports in the future.

# Ports you run web servers on
var HTTP_PORTS 80

# Ports you want to look for SHELLCODE on.
var SHELLCODE_PORTS !80

# Ports you do oracle attacks on
var ORACLE_PORTS 1521

# other variables
#
# AIM servers.  AOL has a habit of adding new AIM servers, so
instead of
# modifying the signatures when they do, we add them to this
list of
# servers.
var AIM_SERVERS
[64.12.24.0/24,64.12.25.0/24,64.12.26.14/24,64.12.28.0/24,64.12.
```

```
29.0/24,64.12.161.0/24,64.12.163.0/24,205.188.5.0/24,205.188.9.0
/24]

# Path to your rules files (this can be a relative path)
var RULE_PATH ../rules

# Configure the snort decoder:
# ============================
#
# Stop generic decode events:
#
# config disable_decode_alerts
#
# Stop Alerts on experimental TCP options
#
# config disable_tcpopt_experimental_alerts
#
# Stop Alerts on obsolete TCP options
#
# config disable_tcpopt_obsolete_alerts
#
# Stop Alerts on T/TCP alerts
#
# config disable_ttcp_alerts
#
# Stop Alerts on all other TCPOption type events:
#
# config disable_tcpopt_alerts
#
# Stop Alerts on invalid ip options
#
# config disable_ipopt_alerts


# Configure the detection engine
# ==============================
#
# Use a different pattern matcher in case you have a machine
with very
# limited resources:
#
# config detection: search-method lowmem
#
##################################################
```

- **var HOME_NET**: Permite establecer el o los rangos de todas las direcciones de la propia red. Se emplea notación CIDR. Separadas por coma (sin espacios) y encerradas entre [ ], permite contener todas las redes que se necesiten. Si se desea incluir cualquier valor, existe el parámetro “any”

- **var EXTERNAL_NET**: permite aclarar rangos de redes que se interpretarán como externas (suele emplearse “any”, pero en algunas ocasiones puede ser un parámetro importante a asignar).

Las variables que siguen, responden a la misma lógica presentada en los dos anteriores y su misión es poder aplicarlas luego en cada una de las reglas.

- Var DNS_SERVERS
- Var SMTP_SERVERS
- Var HTTP_SERVERS
- Var SQL_SERVERS
- Var TELNET_SERVERS

Todos estos servidores es importante incluirlos si se poseen las direcciones de los mismos, pues como se verá más adelante, muchas de las reglas de Snort, operan sobre servicios que prestan estos servidores, y en el caso de no conocer sus direcciones, emplea el valor por defecto que es $HOME_NET obligando al motor a trabajar con todas las IPs pertenecientes a los mismos. En cambio si se acotaron los valores de estos servidores, cada una de esas reglas, entrarán en juego únicamente cuando la dirección IP del datagrama que se esté tratando, se corresponda con uno de ellos, caso contrario, directamente se descarta mejorando sensiblemente el rendimiento de Snort.

- Var HTTP_PORTS
- Var SHELLCODE_PORTS
- Var ORACLE_PORTS

Como se puede apreciar las variables anteriores están referidas a puertos. La lógica es la misma que la de los servidores, es decir, si se emplean determinados puertos en la red, para qué se va a obligar a Snort a analizar todos los existentes, si se puede reducir su espacio de búsqueda. Con estas variables, se acotan los servicios y puertos presentes en la red.

- Var AIM_SERVERS: Esto especifica la lista de los servidores de AOL, ya viene en la configuración por defecto y se aconseja dejarlo como está.
- Var RULE_PATH: Permite definir dónde Snort debe buscar las reglas. Puede ser un path absoluto o relativo.

En esta práctica con variables, te proponemos que configures cada una de ellas para ir ajustando el funcionamiento de Snort.

2) Preprocesadores:

Práctica a realizar:

Configuración de algunos preprocesadores y pruebas de detección.

Con anterioridad se definió la función de estos módulos, ahora se tratarán un poco más en detalle cada uno de ellos.

El primer concepto a comprender, es que cada uno de ellos se activará y configurará a través del archivo “snort.conf” (una vez analizados los preprocesadores y las reglas,

se verá en detalle este archivo) en la sección correspondiente a preprocesadores, la misma se presenta a continuación (Se resalta en negrita cada uno de ellos).

```
##################################################
# Step #2: Configure preprocessors
#
# General configuration for preprocessors is of the form
# preprocessor <name_of_processor>: <configuration_options>

# frag2: IP defragmentation support
# ------------------------------
# This preprocessor performs IP defragmentation.  This plugin will also detect
# people launching fragmentation attacks (usually DoS) against hosts.  No
# arguments loads the default configuration of the preprocessor, which is a
# 60 second timeout and a 4MB fragment buffer.

# The following (comma delimited) options are available for frag2
#    timeout [seconds] - sets the number of [seconds] than an unfinished
#                        fragment will be kept around waiting for completion,
#                        if this time expires the fragment will be flushed
#    memcap [bytes] - limit frag2 memory usage to [number] bytes
#                     (default:  4194304)
#
#    min_ttl [number] - minimum ttl to accept
#
#    ttl_limit [number] - difference of ttl to accept without alerting
#                         will cause false positves with router flap
#
# Frag2 uses Generator ID 113 and uses the following SIDS
# for that GID:
#  SID     Event description
# -----   -------------------
#   1       Oversized fragment (reassembled frag > 64k bytes)
#   2       Teardrop-type attack

preprocessor frag2

# stream4: stateful inspection/stream reassembly for Snort
#--------------------------------------------------------------------
# Use in concert with the -z [all|est] command line switch to defeat
# stick/snot against TCP rules.  Also performs full TCP stream
# reassembly, stateful inspection of TCP streams, etc.  Can statefully
# detect various portscan types, fingerprinting, ECN, etc.

# stateful inspection directive
# no arguments loads the defaults (timeout 30, memcap 8388608)
# options (options are comma delimited):
```

```
# detect_scans - stream4 will detect stealth portscans and generate alerts
#               when it sees them when this option is set
# detect_state_problems - detect TCP state problems, this tends to be very
#                     noisy because there are a lot of crappy ip stack
#                     implementations out there
#
# disable_evasion_alerts - turn off the possibly noisy mitigation of
#                     overlapping sequences.
#
#
# min_ttl [number]  - set a minium ttl that snort will accept to
#                   stream reassembly
#
# ttl_limit [number] - differential of the initial ttl on a session
#                       versus the normal that someone may be playing
 #                      games.
#
# keepstats [machine|binary] - keep session statistics, add "machine" to
#                   get them in a flat format for machine reading, add
#                   "binary" to get them in a unified binary output
#                   format
# noinspect - turn off stateful inspection only
# timeout [number] - set the session timeout counter to [number] seconds,
#                   default is 30 seconds
# memcap [number] - limit stream4 memory usage to [number] bytes
# log_flushed_streams -if an event is detected on a stream the option will
#                  cause all packets that are stored in the stream4
#                  packet buffers to be flushed to disk.  This only
#                  works when logging in pcap mode!
#
# Stream4 uses Generator ID 111 and uses the following SIDS
# for that GID:
#  SID     Event description
# -----   -------------------
#   1       Stealth activity
#   2       Evasive RST packet
#   3       Evasive TCP packet retransmission
#   4       TCP Window violation
#   5       Data on SYN packet
#   6       Stealth scan: full XMAS
#   7       Stealth scan: SYN-ACK-PSH-URG
#   8       Stealth scan: FIN scan
#   9       Stealth scan: NULL scan
#   10      Stealth scan: NMAP XMAS scan
#   11      Stealth scan: Vecna scan
#   12      Stealth scan: NMAP fingerprint scan stateful detect
#   13      Stealth scan: SYN-FIN scan
```

```
#   14      TCP forward overlap

preprocessor stream4: detect_scans, disable_evasion_alerts

# tcp stream4_reassembly directive
# no arguments loads the default configuration
#   Only reassemble the client,
#   Only reassemble the default list of ports (See below),
#   Give alerts for "bad" streams
#
# Available options (comma delimited):
# clientonly - reassemble traffic for the client side of a connection only
# serveronly - reassemble traffic for the server side of a connection only
# both - reassemble both sides of a session
# noalerts - turn off alerts from the stream reassembly stage of stream4
# ports [list] - use the space separated list of ports in [list], "all"
#               will turn on reassembly for all ports, "default" will turn
#               on reassembly for ports 21, 23, 25, 53, 80, 143, 110, 111
#               and 513

preprocessor stream4_reassemble

# http_decode: normalize HTTP requests
# ----------------------------------
# http_decode normalizes HTTP requests from remote
# machines by converting any %XX character
# substitutions to their ASCII equivalent. This is
# very useful for doing things like defeating hostile
# attackers trying to stealth themselves from IDSs by
# mixing these substitutions in with the request.
# Specify the port numbers you want it to analyze as arguments.
#
# Major code cleanups thanks to rfp
#
# unicode          - normalize unicode
# iis_alt_unicode  - %u encoding from iis
# double_encode    - alert on possible double encodings
# iis_flip_slash   - normalize \ as /
# full_whitespace  - treat \t as whitespace ( for apache )
#
# for that GID:
#  SID     Event description
# -----   -------------------
#   1       UNICODE attack
#   2       NULL byte attack
```

```
preprocessor http_decode: 80 unicode iis_alt_unicode double_encode iis_flip_slash
full_whitespace

# rpc_decode: normalize RPC traffic
# ---------------------------------
# RPC may be sent in alternate encodings besides the usual
# 4-byte encoding that is used by default.  This preprocessor
# normalized RPC traffic in much the same way as the http_decode
# preprocessor.  This plugin takes the ports numbers that RPC
# services are running on as arguments.
# The RPC decode preprocessor uses generator ID 106
#
# arguments: space separated list
# alert_fragments - alert on any rpc fragmented TCP data
# no_alert_multiple_requests - don't alert when >1 rpc query is in a packet
# no_alert_large_fragments - don't alert when the fragmented
#                            sizes exceed the current packet size
# no_alert_incomplete - don't alert when a single segment
#                       exceeds the current packet size

preprocessor rpc_decode: 111 32771

# bo: Back Orifice detector
# -------------------------
# Detects Back Orifice traffic on the network.  Takes no arguments in 2.0.
#
# The Back Orifice detector uses Generator ID 105 and uses the
# following SIDS for that GID:
#  SID     Event description
# -----   ------------------
#   1      Back Orifice traffic detected

preprocessor bo

# telnet_decode: Telnet negotiation string normalizer
# ---------------------------------------------------
# This preprocessor "normalizes" telnet negotiation strings from
# telnet and ftp traffic.  It works in much the same way as the
# http_decode preprocessor, searching for traffic that breaks up
# the normal data stream of a protocol and replacing it with
# a normalized representation of that traffic so that the "content"
# pattern matching keyword can work without requiring modifications.
# This preprocessor requires no arguments.
# Portscan uses Generator ID 109 and does not generate any SID currently.

preprocessor telnet_decode
```

```
# Portscan: detect a variety of portscans
# -------------------------------------
# portscan preprocessor by Patrick Mullen <p_mullen@linuxrc.net>
# This preprocessor detects UDP packets or TCP SYN packets going to
# four different ports in less than three seconds. "Stealth" TCP
# packets are always detected, regardless of these settings.
# Portscan uses Generator ID 100 and uses the following SIDS for that GID:
#  SID     Event description
# -----   -------------------
#   1       Portscan detect
#   2       Inter-scan info
#   3       Portscan End

# preprocessor portscan: $HOME_NET 4 3 portscan.log

# Use portscan-ignorehosts to ignore TCP SYN and UDP "scans" from
# specific networks or hosts to reduce false alerts. It is typical
# to see many false alerts from DNS servers so you may want to
# add your DNS servers here. You can all multiple hosts/networks
# in a whitespace-delimited list.
#
#preprocessor portscan-ignorehosts: 0.0.0.0

# arpspoof
#---------------------------------------
# Experimental ARP detection code from Jeff Nathan, detects ARP attacks,
# unicast ARP requests, and specific ARP mapping monitoring.  To make use
# of this preprocessor you must specify the IP and hardware address of hosts on # the same
layer 2 segment as you.  Specify one host IP MAC combo per line.
# Also takes a "-unicast" option to turn on unicast ARP request detection.
# Arpspoof uses Generator ID 112 and uses the following SIDS for that GID:
#  SID     Event description
# -----   -------------------
#   1       Unicast ARP request
#   2       Etherframe ARP mismatch (src)
#   3       Etherframe ARP mismatch (dst)
#   4       ARP cache overwrite attack

#preprocessor arpspoof
#preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00

# Conversation
#----------------------------------------
# This preprocessor tracks conversations for tcp, udp and icmp traffic.  It
# is a prerequisite for running portscan2.
#
# allowed_ip_protcols 1 6 17
```

```
#     list of allowed ip protcols ( defaults to any )
#
# timeout [num]
#     conversation timeout ( defaults to 60 )
#
#
# max_conversations [num]
#     number of conversations to support at once (defaults to 65335)
#
#
# alert_odd_protocols
#     alert on protocols not listed in allowed_ip_protocols
#
# preprocessor conversation: allowed_ip_protocols all, timeout 60, max_conversations 3000
#
# Portscan2
#-----------------------------------------
# Portscan 2, detect portscans in a new and exciting way.  You must enable
# spp_conversation in order to use this preprocessor.
#
# Available options:
#      scanners_max [num]
#      targets_max [num]
#      target_limit [num]
#      port_limit [num]
#      timeout [num]
#      log [logdir]
#
#preprocessor portscan2: scanners_max 256, targets_max 1024, target_limit 5, port_limit
20, timeout 60

# Too many false alerts from portscan2? Tone it down with
# portscan2-ignorehosts!
#
# A space delimited list of addresses in CIDR notation to ignore
#
# preprocessor portscan2-ignorehosts: 10.0.0.0/8 192.168.24.0/24
#

# Experimental Perf stats
# ----------------------
# No docs. Highly subject to change.
#
# preprocessor perfmonitor: console flow events time 10
#
##################################################
```

Todos los preprocesadores se activan en el archivo "snort.conf", en el "paso 2", dentro de este archivo existe una breve descripción de cada uno de ellos, y el formato general es:

preprocessor <nombre> : <Opciones_de_configuración>

A continuación se presenta cada uno de ellos:

Preprocesador frag2:

La fragmentación de paquetes es una técnica que emplea la familia TCP/IP para optimizar el tráfico y permitir adaptar el tamaño del mismo para pasar por distintos tipos de redes que pueden exigir un tamaño fijo o máximo de los mismos.

Una técnica conocida de ataques es justamente el aprovechamiento de esta operación. Se lleva a cabo armando el código del ataque en pequeños fragmentos, los cuales no son sospechosos pues el código maligno al completo se verá recién al rearmarse la totalidad de los fragmentos, y cada uno de ellos en sí mismo puede ser interpretado como válido. Un ejemplo de este ataque puede ser a través del empleo del programa "*fragroute*", que hace exactamente esto. Si un IDS o FW, no llevan el control de esa secuencia, el ataque pasa de largo y se ejecuta en el destino final.

Para evitar este tipo de acciones es que se implementó este preprocesador, el cual se encarga de reensamblar los fragmentos correspondientes al protocolo IP para entregarle al motor de detección el paquete completo (y no cada uno de los fragmentos).

Un detalle de particular interés, es que una mala configuración de este preprocesador puede ser empleada para realizar un ataque DoS hacia el IDS, pues si se generan muchos paquetes fragmentados, podría llegar el caso que el IDS esté tan ocupado con todos esos reensambles que no esté en capacidad de procesar el resto del tráfico. Es por ello que este preprocesador posee parámetros que se configuran para este tipo de acciones:

- timeout [segundos] (Define el número de segundos que se almacenará en memoria un fragmento no completo a nivel IP. Superado este tiempo, se descartan todos los fragmentos correspondientes a ese datagrama. Si no se emplea esta opción, por defecto el valor es 60 segundos).
- Memcap [bytes] (define el número de bytes disponibles en memoria para esta operación. Si no se emplea esta opción, por defecto el valor es 4MB = 4194304 bytes).
- Min ttl [número] (Define el mínimo valor de "time to life" empleado en ese fragmento IP a aceptar, cualquier valor menor a este se descarta. Si no se emplea esta opción, este valor no es tenido en cuenta).

Preprocesador stream4:

Este preprocesador permite llevar el control de estado de las secuencias TCP. Esta actividad tiene muchos aspectos a destacar, pues permite:

- Verificar la apertura y cierre de sesiones.

- Realizar el seguimiento de tráfico de cada sesión (es decir, entre cada cliente y servidor individualmente).
- Colaborar con la identificación de scan de puertos.

Stream4 tiene 2 características fundamentales:

- Control de estados TCP: Se basa en los flags de TCP (SYN, ACK y FIN), con los cuales se establecen y cierran sesiones y a su vez en los números que permiten realizar el secuenciamiento (Ns y Nr: concepto de ventana deslizante) y en los pasos establecidos para toda esta actividad.
- Reensamble de sesiones: permite analizar el paquete completo y no cada una de sus partes.

Este preprocesador posee varias opciones (por defecto tiene timeout=30 segundos y reserva 8Mbyte de memoria de captura), las opciones son:

- z [est | all]: permite almacenar el estado de todas las conexiones TCP y alertar sólo cuando no se respeta la secuencia de conexión y/o desconexión adecuada.
- detect_scan: como su nombre lo indica permite detectar scan de puertos.
- detect_state_problems: problemas con estados TCP.
- disable_evasion_alerts: deshabilita la posibilidad de generar ruido de secuencias mal configuradas, para evadir otros eventos.
- Min_ttl [múmero]: perite configurar el valor mínimo de ttl para aceptar el reensamble de paquetes.
- Ttl_limit [número]: Diferencia máxima de ttl que puede ser tolerada entre paquetes de una misma sesión.
- Keepstats [machine | binary]: guarda estadísticas de cada sesión TCP en formato binario o texto.
- Noinspect: deshabilita el control de estados.
- Timeout [número]: configura el tiempo de almacenamiento de cada sesión.
- Memcap [número]: configura la cantidad de memoria disponible para stream4.
- Log_flushed_streams: si se detecta un evento dentro de un flujo, esto causa que sea guardado el paquete completo. Sólo opera en modo pcap.

Preprocesador stream4_reassembly:

Este preprocesador permite especificar cuáles sesiones se desea reensamblar. La configuración por defecto, solo reensambla los paquetes del cliente y la lista de puertos por defecto (21, 23, 25, 53, 80, 143, 110, 111 y 513). Las opciones son:

- clientonly: reensambla solo del lado cliente.
- Serveronly: solo del lado servidor.

- Both: ambos.
- Noalerts: deshabilita las alertas.
- Ports [lista]: permite especificar qué puertos reensamblar (se coloca la lista separada por espacios, o las palabras "all" o "default").

Preprocesador http_decode:

Este preprocesador, normaliza el formato de las solicitudes http, substituyendo todos los caracteres del tipo %xx (unicode), iis (UTF-8), caracteres de escape, etc.. a su equivalente ASCII. Permite especificar la lista de puertos que se deseen, separados por espacios

Preprocesador rpc_decode:

El protocolo RPC permite que un programa de un host llame a otro programa en un host diferente (se encuentra bien documentado en la RFC 1831). RPC se ejecuta a nivel de aplicación, y emplea 32 bits para cada registro, de los cuales el primer bit indica si el registro continúa o si es el último, y los restantes 31 bits describen el tamaño de datos del fragmento. Por encontrarse a nivel de aplicación, se puede fácilmente generar tráfico que sea fragmentado en los niveles inferiores, y de esta forma, disfrazar el contenido de cada registro, es decir, partir esos 32 bits en fragmentos diferentes. Con este método, un IDS, que no pueda "normalizar" cada uno de estos grupos de 32 bits, pasaría por alto el contenido de la información que contiene a continuación.

El preprocesador_rpc entonces, de manera similar al preprocesador anterior, normaliza el tráfico RPC, pero su actividad es el control y armado de sus característicos 4-bytes de codificación, para entregarlo agrupados al motor de detección siempre en formato 4-bytes. Acepta como argumentos, la lista de puertos sobre los que esté trabajando RPC separados por espacios.

Preprocesador Back Oriffice:

Este preprocesado examina todos los paquetes UDP mayores de 18 bytes, verificando los primeros 8 caracteres con (en realidad lo hace primero con 2, y luego posee un proceso que decide si continuar o no...), para verificar la existencia del cifrado que emplea BO y lo confronta contra un tabla que Snort carga al iniciarse.

Detecta tráfico Back Orifice. No posee argumentos.

Preprocesador telnet_decode:

Este preprocesador, también se dedica a normalizar el tráfico, en este caso el de las secuencias de negociación de telnet, buscando paquetes que no respeten la misma. No requiere argumentos, pero se pueden especificar los puertos que se deseen, separados por espacios.

Preprocesador portscan:

Este preprocesador detecta los tipos de scan lanzados desde un host hacia varios puertos durante un cierto período de tiempo, por defecto lo hace sobre 4 diferentes puertos en menos de 3 segundos . Trabaja sobre scan UDP y TCP y sus flags u opciones correspondientes.

El problema que plantean los detectores de scan es el llamado "slow scan", es decir qué sucede si se realiza un scan, pero por ejemplo lanzando un paquete cada diez minutos. Nuevamente se debe decidir entre el límite de detectar y/o dejar el IDS fuera de servicio, pues para esta actividad el IDS necesita almacenar en memoria RAM, la totalidad del tráfico dirigido hacia los diferentes puertos de la red que monitoriza, y por lo tanto, si se almacenan intervalos de tiempo muy grandes, es peligroso, y si son pequeños, no estaría en capacidad de detectar un "Slow scan".

Para configurar este preprocesador es necesario evaluar los siguientes parámetros:

- Red a monitorizar: Suele ser la variable $HOME_NET, pero puede especificarse lo que se desee en formato CIDR separado por espacios (Ej:10.0.0.0/8).
- Número de puertos: Cantidad de puertos en el intervalo de tiempo que se especifica a continuación.
- Intervalo de tiempo en segundos: Determina el umbral de tiempo en el que se escanean los puertos del punto anterior.
- Archivo donde se almacenarán los eventos.

Ejemplo: preprocessor portscan: $HOME_NET 4 3 portscan.log

Preprocesador portscan-ignorehosts:

Es un complemento del anterior y permite especificar qué host se desea que sean ignorados para el procesador de scan. No posee parámetros, simplemente se colocan las redes o host que se desea ignorar en formato CIDR, separados por espacios.

Preprocesador portscan2:

Este amplía al anterior (y se prevé que será el único a emplear a futuro). Se debe combinar con el preprocesador "conversation".

Preprocesador arpspoof

Permite comparar la correspondencia entre una respuesta ARP y una tabla cargada en Snort, que posea la lista de IP-MAC de la propia red. Este preprocesador actualmente se encuentra en estado experimental y puede bajar sensiblemente el rendimiento de la herramienta si la red es considerable. Se aclara aquí que existen otras herramientas que realizan específicamente esta actividad, independientemente de Snort, como puede ser

ARPWATCH, que en muchos casos debería analizarse su empleo en otro hardware diferente de donde se ejecuta Snort.

- **Preprocesador conversation:**

  También se encuentra en estado experimental y su actividad es la de guardar registros de cada comunicación entre dos hosts, permitiendo al preprocesador scan2 realizar el seguimiento y potencial alerta de una conversación.

3) REGLAS (o firmas):

Práctica a realizar:

Te proponemos que dediques un tiempo para realizar las siguientes tareas (ya estás en capacidad de hacerlo perfectamente):

- Activación de reglas.
- Creación de reglas de detección.
- Análisis de tráfico y evaluación de parámetros (con tcpdump y Ethereal).
- Creación de local.rules, para detectar con Snort el tráfico capturado.
- Ejecución de Snort con sus propias local.rules

Para que puedas practicar con diferentes patrones de tráfico, a continuación ponemos varias reglas que fueron creadas en una red en producción para la detección de vulnerabilidades que en su momento no eran detectadas por Snort. Como puedes ver en ellas, dentro del campo "msg" encontrarás varias en las que figura {nessus}, en esos casos se trata de patrones de tráfico que fueron generados con "Nessus" y que las reglas de Snort no detectaban (esta tarea con "nessus-snort" es el tema de la siguiente práctica).

EJEMPLOS de LOCAL.RULES:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Anonymous FTP enabled
{nessus}"; flags:A+; content:"USER null"; nocase; depth: 10;
reference:CVE, CAN-1999-0452; classtype:attempted-user; sid:1001001;
rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Linux FTP Backdoor
{nessus}";flags: AP; content:"PASS null"; nocase; depth: 10;
reference:CVE, CAN-1999-0452; classtype:attempted-user; sid:1001015;
rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Writeable FTP root
{nessus}"; flags:A+; content:"STOR nessus_test"; depth: 20; reference:CVE,
CAN-1999-0527; classtype:attempted-user; sid:1001002; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:" writeable FTP root
{Comando CWD / - nessus}"; flags:A+; content:"CWD /"; depth: 10;
```

```
reference:CVE, CAN-1999-0527; classtype:attempted-user; sid:1001003;
rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 513 (msg:"rlogin {nessus}";
flags:A+; content:"root"; nocase; depth: 10; reference:CVE, CAN-1999-0651;
classtype:attempted-user; sid:1001004; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"EXPN and VRFY commands
{nessus}"; flags:A+; content:"HELO nessus.org"; nocase; depth: 20;
reference:CVE, CAN-1999-0531; classtype:successful-recon-largescale;
sid:1001005; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP Server type and
version {nessus}"; flags:A+; content:"HELP"; depth: 10; classtype:network-
scan; sid:1001006; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"WFTP login check
{nessus}"; flags:A+; content:"bogusbogus"; depth: 25; reference:CVE, CAN-
1999-0200; classtype:attempted-user; sid:1001007; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"IIS 5.0 PROPFIND
Vulnerability {nessus}"; flags:A+; content:"PROPFIN"; depth: 25;
classtype:attempted-user; sid:1001008; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 513 (msg:"SysV /bin/login buffer
overflow (rlogin) {nessus}"; flags:A+; content:"nessus"; depth: 10;
reference:url, www.cert.org/advisories/CA-2001-34.html;
classtype:attempted-user; sid:1001009; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP Service Allows Any
Username {nessus}"; flags:A+; content: "user pp * pass pp"; regex; nocase;
depth: 20; classtype:attempted-user; sid:1001010; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"lpd, dvips and remote
command execution {nessus}"; flags:A+; content:"|F702 0183 82C0 1C3B 0000
0000 03E8 1B20 5463 5820 6F75 7470 7574 2032 3030| "; depth: 30;
reference:CVE,CAN-2001-1002; classtype:attempted-user; sid:1001011;
rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP antivirus filter
{nessus}"; flags:A+; content:"HELO nessus"; nocase; depth: 15;
classtype:attempted-recon; sid:1001012; rev:1;)
#
alert tcp $EXTERNAL_NET 20 -> $HOME_NET 8888 (msg:"BenHur Firewall active
FTP firewall leak {nessus}"; classtype:attempted-recon; sid:1001013;
rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"EXPERIMENTAL WEB-
MISC OpenSSL Worm traffic"; content:"TERM=xterm"; nocase; classtype:web-
application-attack; reference:url,www.cert.org/advisories/CA-2002-27.html;
sid:1001014; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"deteccion 2 WEB-
MISC OpenSSL Worm traffic"; content:"|4745 5420 2F20 4854 5450 2F31 2E30
0D0A 0D0A|"; flags: AP; classtype:web-application-attack; reference:url,
www.cert.org/advisories/CA-2002-27.html; sid:1001016; rev:1;)
#
```

```
alert udp $EXTERNAL_NET any -> $HOME_NET 137 (msg:"Using NetBIOS to
retrieve information from a Windows host {nessus}"; content:"|0000 0001
0000 0000 0000 2043 4b41 4141 4141 4141 4141 4141 4141 4141 4141|"; depth:
33; classtype: attempted-dos; sid: 1001017; rev:1;)
#
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"gusano referente a
lsass.exe"; content:"lsass.exe"; offset: 40; depth: 50; classtype:web-
application-attack; sid:1001017; rev:1;)
#
alert tcp any any -> $HOME_NET 24 (msg: "aaaaaaaaa"; content: "a\*sh";
regex; classtype: web-application-attack; sid: 1001018; rev: 1;)
#
Alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"Alibaba 2.0 buffer
Overflow {nessus}"; content: "POST XXXXXXXX"; depth: 15; classtype:web-
application-attack;reference:CVE,CAN-200-0626; sid:1001019; rev:1;)
#
Alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"Axent Raptor DoS
{nessus}"; tos: 123; id:1234; ttl: 255; ip_proto: 6; reference: CVE,CVE-
1999-0905; classtype: attempted-dos; sid:1001021;rev:1;)
```

4) Salidas y Logs:

Práctica a realizar:

- Activación de diferentes salidas: formato alert y log.
- Pruebas con formato unificado, CSV y texto.
- Pruebas con formatos fast y full.
- Configuración de salidas en línea de comandos y en el archivo snort.conf
- Análisis de salidas: ID, Clasificación, prioridad y referencias.

Las salidas difieren del resto de los componentes de Snort, pues no existe un único punto de entrada para las opciones de salida (aunque parezca raro), es decir que existen diferentes partes de Snort que pueden generar salidas, estas son:

- El decodificador de paquetes: Puede ejecutarse para generar salidas tipo tcpdump o texto.
- Algunos preprocesadores: existen algunos de estos módulos, que tienen sus propias salidas (Ej: portscan).
- El motor de detección: emplea los plug-ins de salida a alert y syslog.

El primer concepto que se debe tener en cuenta es algo que ya se ha mencionado, pero vale la pena reiterar. Si se desea guardar información en formato log hacia el disco duro, se debe emplear la opción "**-l**" y aclarar luego el path hacia donde se dirigirá este archivo y su nombre (Ej: snort –de –l /var/log/snort_log).

En muchos casos es conveniente dejar la opción de almacenar estos datos en formato binario (también llamado formato "pcap"), lo cual es una excelente alternativa para visualizarlos

con alguna herramienta de análisis de tráfico (como también se mencionó con anterioridad), para este caso se emplea la opción "**-b**" (Ej: snort –de –l /var/log/snort_log –b). Otra gran ventaja de esta opción es que es extremadamente rápida. Si se desea leer este archivo, se emplea la opción "**-r**".

La configuración por defecto de Snort, cuando se lo emplea como IDS, es almacenar los eventos en formato denominado **"alert"** en un subdirectorio denominado **"log".**

El mejor modo de configurar múltiples salidas es hacerlo a través del paso 3 en el archivo "snort.conf", el cual se presenta a continuación:

```
#--------------------------------------------------------------------
# Step #3: Configure output plugins
#
# Uncomment and configure the output plugins you decide to use.
# General configuration for output plugins is of the form:
#
# output <name_of_plugin>: <configuration_options>
#
# alert_syslog: log alerts to syslog
# ----------------------------------
# Use one or more syslog facilities as arguments.  Win32 can also
# optionally specify a particular hostname/port.  Under Win32, the
# default hostname is '127.0.0.1', and the default port is 514.
#
# [Unix flavours should use this format...]
# output alert_syslog: LOG_AUTH LOG_ALERT
#
# [Win32 can use any of these formats...]
# output alert_syslog: LOG_AUTH LOG_ALERT
# output alert_syslog: host=hostname, LOG_AUTH LOG_ALERT
# output alert_syslog: host=hostname:port, LOG_AUTH LOG_ALERT

# log_tcpdump: log packets in binary tcpdump format
# -------------------------------------------------
# The only argument is the output file name.
#
# output log_tcpdump: tcpdump.log

# database: log to a variety of databases
# ---------------------------------------
# See the README.database file for more information about configuring
# and using this plugin.
#
# output database: log, mysql, user=root password=test dbname=db
host=localhost
# output database: alert, postgresql, user=snort dbname=snort
# output database: log, unixodbc, user=snort dbname=snort
# output database: log, mssql, dbname=snort user=snort password=test

# unified: Snort unified binary format alerting and logging
# ---------------------------------------------------------
# The unified output plugin provides two new formats for logging
# and generating alerts from Snort, the "unified" format.  The
# unified format is a straight binary format for logging data
# out of Snort that is designed to be fast and efficient.  Used
```

```
# with barnyard (the new alert/log processor), most of the overhead
# for logging and alerting to various slow storage mechanisms
# such as databases or the network can now be avoided.
#
# Check out the spo_unified.h file for the data formats.
#
# Two arguments are supported.
#    filename - base filename to write to (current time_t is appended)
#    limit    - maximum size of spool file in MB (default: 128)
#
# output alert_unified: filename snort.alert, limit 128
# output log_unified: filename snort.log, limit 128

# You can optionally define new rule types and associate one or
# more output plugins specifically to that type.
#
# This example will create a type that will log to just tcpdump.
# ruletype suspicious
# {
#   type log
#   output log_tcpdump: suspicious.log
# }
#
# EXAMPLE RULE FOR SUSPICIOUS RULETYPE:
# suspicious $HOME_NET any -> $HOME_NET 6667 (msg:"Internal IRC Server";)
#
# This example will create a rule type that will log to syslog
# and a mysql database.
# ruletype redalert
# {
#   type alert
#   output alert_syslog: LOG_AUTH LOG_ALERT
#   output database: log, mysql, user=snort dbname=snort host=localhost
# }
#
# EXAMPLE RULE FOR REDALERT RULETYPE
# redalert $HOME_NET any -> $EXTERNAL_NET 31337 (msg:"Someone is being
LEET"; \
#   flags:A+;)

#
# Include classification & priority settings
#

include classification.config

#
# Include reference systems
#

include reference.config

##################################################################
```

Como se puede apreciar existen diferentes formatos de salida par snort, estos son:

- CSV (Comma Separated Values): permite fácilmente importar datos desde varias

aplicaciones.

- Syslog: Similar a la opción desde "alert facility", permite personalizar la facilidad de syslog.
- Database: Permite la salida a distintos formatos de bases de datos: MySGL, PostgreSQL, UnixODBC, Oracle y SQL Server.
- Null: Permite a Snort enviar a "alert facility", pero no crea ningún archivo de log.
- Tcpdump: Salida en formato tcpdump.
- SnmpTrap: envía traps SNMP a un servidor SNMP.
- Unified: Es el futuro de Snort y es la salida más rápida. Emplea el formato binario con Fast alert. Emplea el programa Barnyard para leer posteriormente.
- XML: Salidas en formato SNML (Simple Network Markup Lenguage).

Existen dos "MODOS" de salida para Snort:

a. Modo alerta: Cuando el "alert" de una regla genera una alarma, se tienen en cuenta dos acciones:

- Salida del evento (denominada "facilidad"):

  La facilidad controla el formato de la alerta, algunos aspectos y su destino. Las opciones de facilidad son:

  - Full (por defecto): contenido completo de todos los encabezados.
  - Fast: Formato simple, con tiempo, mensaje de alerta, direcciones fuente y destino y puertos.
  - Syslog: Envía a syslog. (LOG_AUTH_PRIV y LOG_ALERT).
  - Unsock: Envía a un socket UNIX.
  - SMB: Mensaje winpopup de windows.
  - None: desactiva alertas.
  - Console: envía alertas en modo fast a la consola.
  - Enviar un log acorde a la configuración de detalle deseada: Esto se trata en el punto siguiente (b. Modo logging).

- Modo logging:

  Almacena la información completa, pero sin generar una alerta. Este modo puede ser activado, empleando directamente la palabra "log", dynamic" o "alert" en las reglas de snort (Se debe tener en cuenta que si no se inicia Snort con la opción –l este modo sólo lo activan las "alert" de las reglas, como se trató en el apartado anterior) . Por defecto Snort lo hará siempre en /var/log/snort, pero puede ser cambiado con la opción "-l" a la ruta y archivo que se desee. Se crea un directorio por cada dirección IP que detecte Snort y dentro de cada uno de ellos se almacenarán las alertas correspondientes.

Como se mencionó con anterioridad, la mejor forma de trabajar con las salidas es a través del archivo **"snort.conf"** en el paso 3.

Las salidas se declaran con la forma: output <nombre>: <opción>

Se tratarán a continuación algunas de ellas:

1) Output log_null: Configura las salidas de Snort para que no creen los subdirectorios por cada dirección IP que escucha.

2) output alert_CSV: <filename> <format>: Esta es quizás la salida más flexible para operar posteriormente, pues permite varias opciones, las que entregará separadas por comas. Lo realmente útil es el campo <format>, en el cual se puede especificar, separados por coma, los siguientes parámetros: *_timestamp, sig_generator, sig_id, sig_rev, msg, proto, src, srcport, dst, dstport, ethsrc, ethdst, ethlen, tcpflags, tcpseq, tcpack, tcplen, tcpwindow, ttl, tos, id, dgmlen, iplen, icmptype, icmpcode, icmpid, icmpseq.* SIN DEJAR ESPACIOS!!!!!

   Ejemplos:

   ```
   output alert_CSV: /var/log/alert.csv default
   output alert_CSV: /var/log/alert.csv timestamp,sid,proto,src,dst
   ```

3) output log_unified: filename snort.log : Se trata del método más rápido para almacenar eventos, permite hacerlo a "alert y log files", acorde al nivel de detalle que se desee. Los eventos se almacenan en formato binario, pero no es el mismo formato de pcap (es decir no se lo puede leer directamente con tcpdump o ethereal), este nuevo formato, se aprecia que será el futuro de Snort y fue pensado para trabajar con la herramienta BARNYARD (que puede descargarse de www.snort.org), la cual permite varios tipos de configuración. Otro programa más simple para transformar formatos es logtopcap.c (que puede descargarse de http://dragos.com/logtopcap.c), este programa una vez compilado (gcc –o nombre_final logtopcap.c) permite transformar rápidamente cualquier archivo en formato unificado a formato binario (pcap), se ejecuta de la siguiente forma:

   ```
   ./logtopcap snort.log.nnnnnnnn archivo_bin_destino
   ```

4) output log_tcpdump: almacenará paquetes "log" en formato tcpdump en la ruta especificada. Y crea archivos del tipo "snort.log.nnnnnn". Para leer estos archivos, se puede emplear tcpdump – r nombre_arch o Ethereal. Ejemplo:

   ```
   output log_tcpdump: /usr/local/bin/log/snort.log
   ```

EJEMPLOS:

`./snort -c snort.conf` (sale por defecto con formato "full", hacia /var/log/snort/alert y también crea un directorio por cada dirección IP que detecta).

`./snort -c snort.conf - A fast` (sale con formato "fast", hacia /var/log/snort/alert y también crea un directorio por cada dirección IP que detecta).

`./snort -c snort.conf -b` (sale por defecto con formato "full", hacia /var/log/snort/alert y crea un archivo en binario con nombre "snort.log.nnnnnnn").

`./snort -l /var/tmp/ -c snort.conf` (sale por defecto con formato "full", hacia /var/tmp/alert y también crea un directorio por cada dirección IP que detecta).

`./snort -l ./log -c snort.conf` (sale por defecto con formato "full", hacia el directorio local /log y también crea un directorio por cada dirección IP que detecta). (DEBE EXISTIR EL DIRECTORIO LOG, sino dará un error)

```
output log_tcpdump: /usr/local/bin/log/snort.log
output alert_CSV: /var/log/alert.csv default
output alert_CSV: /var/log/alert.csv timestamp,sid,proto,src,dst
```

`./logtopcap snort.log.nnnnnnnn archivo_bin_destino` (empleo de logtocap)

5) Instalación de Snort + MySQL + ACID.

Estas tres herramientas presentan la gran ventaja de una visualización muy eficiente con la interfaz gráfica "ACID" y una consulta rápida a todos los eventos detectados a través de una base de datos "MySQL", por esta razón hemos creído conveniente que puedas instalarlas y practicar con ellas.

A continuación intentamos redactar con el mayor detalle que pude los pasos a seguir para la instalación del ACID que pueden presentar alguna dificultad.

En esta práctica os invitamos a que sigáis los pasos tal cual los vamos poniendo a continuación hasta poder levantar esta interfaz gráfica. Veréis que es necesario también instalar y configurar la base de datos "**mysql**" para que trabaje con Snort, y desde ella pueda enviarse los eventos a ACID.

Los pasos para una distribución de Linux "Debian" son:

Desde una consola "COMO ROOT":

#**apt-get install mysql-server** (nos pedirá la contraseña para el usuario root de esta DB, os aconsejamos que pongáis un usuario sencillo y la misma password, pues estamos de prácticas y aprendizaje)

#**apt-get install snort-mysql**

Si necesitamos incorporar la password de root en Mysql deberíamos ejecutar el comando que figura a continuación (en mi caso no lo hice pues siempre me valido como root en mi máquina, pero en el vuestro no lo sé *"ya sé que no se debe hacer……"*):

#rootmysqladmin -uroot password nueva-pass

#mysqladmin -uroot -p create snort (Se crea la nueva DB con nombre "snort")

Para crear las tablas:

#gunzip /usr/share/doc/snort-mysql/create_mysql.gz

#mysql -uroot -p snort < /usr/share/doc/snort-mysql/create_mysql

Para crear el usuario para snort:

#mysql -uroot -p

(os pedirá la password que pusisteis en la instalación de mysql..... para salir del prompt">" escribid "quit")

Desde el gestor de paquetes "Synaptics" instalar "**acidbase**" (con todos los complementos que el mismo seleccione)

Ejecutamos:

#chown -R www-data:www-data /etc/acidbase

Editamos

#vi /etc/acidbase/base_conf.php

En nuestro caso agregamos (o modificamos dentro de todo el archivo) a continuación de la línea (todo lo que viene abajo):

```
###### End of variables configured through dbconfig-common
    $alert_dbname = "snort";
    $alert_host = "localhost";
    $alert_port = "8080";
    $alert_user = "root";
    $alert_password = "root";

    /* Archive DB connection parameters */
    $archive_exists   = 0; # Set this to 1 if you have an archive DB
    $archive_dbname   = 'snort_archive';
```

```
$archive_host     = 'localhost';
$archive_port     = '8080';
$archive_user     = 'root';
$archive_password = 'root';
```

(Tened en cuenta que nuestro usuario y password para esta DB es "root")

(Como veis aparece también una DB "snort_archive" que aún no fue creada, pero que debemos crear por si el mismo ACID, genera algún mensaje, para ello ejecutamos nuevamente:

#mysqladmin -uroot -p create snort_archive

(Se crea la nueva DB con nombre "snort_archive")

En nuestro caso, no estamos seguros si venía por defecto o lo llegamos a configurar nosotros, pero en el archivo "/etc/acidbase#vi database.php", también quedó configurado al final como figura a continuación (en vuestro caso controladlo):

```
## you'll probably also want to edit the configuration file mentioned
## above too.
##
$alert_user='root';
$alert_password='root';
$basepath='';
$alert_dbname='snort';
$alert_host='';
$alert_port='';
$DBtype='mysql';
```

Ahora en snort edita:

#/etc/snort/vi snort.conf

y en la sección de "Output" (salidas) agrega:

output alert_full : alert

Esto configura el log sobre MySQL

output database: log, mysql, user=root password=root dbname=snort host=localhost

(recuerda que debe ser TU user y passwd).

Por las dudas antes de terminar reiniciaremos apache y mysqld:

#/etc/init.d/apache2 stop

#/etc/init.d/apache2 start

Lanzamos snort:

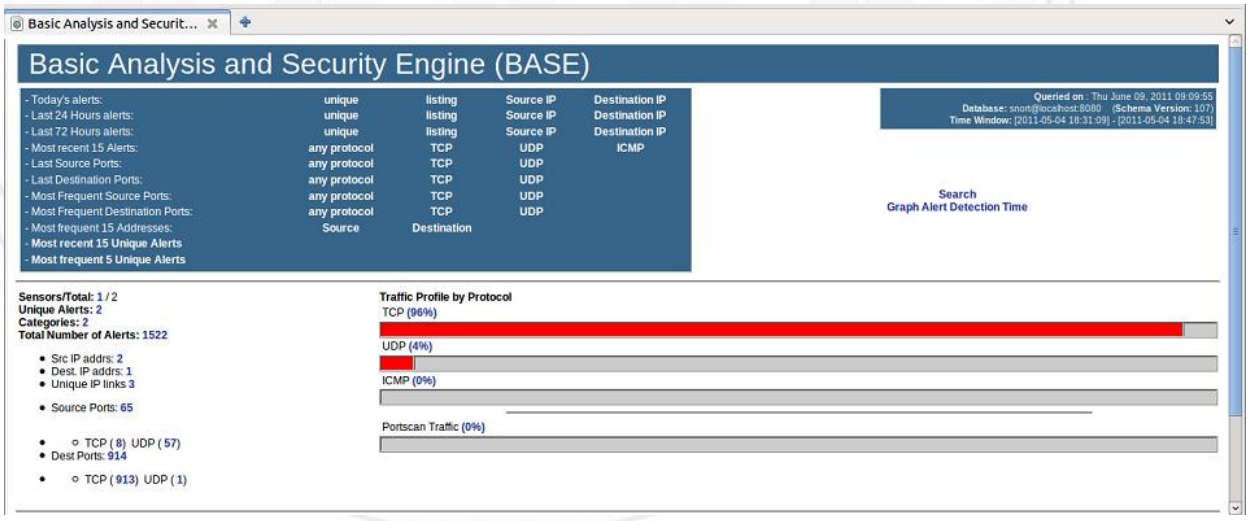
#snort -c /etc/snort/snort.conf

Y si todo está en orden, desde cualquier navegador podríamos ver nuestra consola ACID colocando:

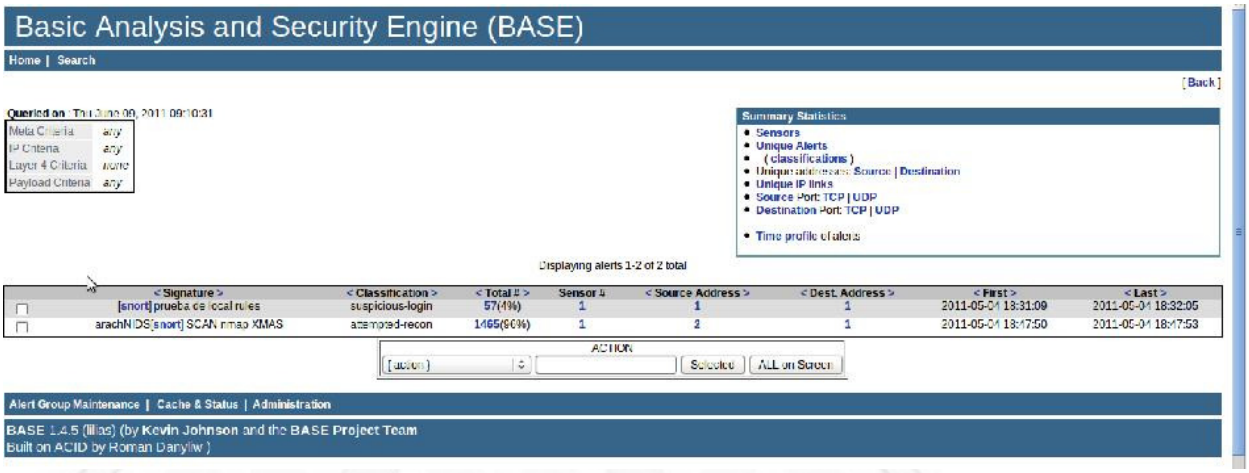
http://localhost/acidbase/

En nuestro caso "apache2" lo tenemos instalado desde hace tiempo, pero la instalación no debería generar ningún conflicto (lo hemos probado sin problemas) vamos a probar sin instalar apache2, pero si falla algo, sólo debéis ejecutar:

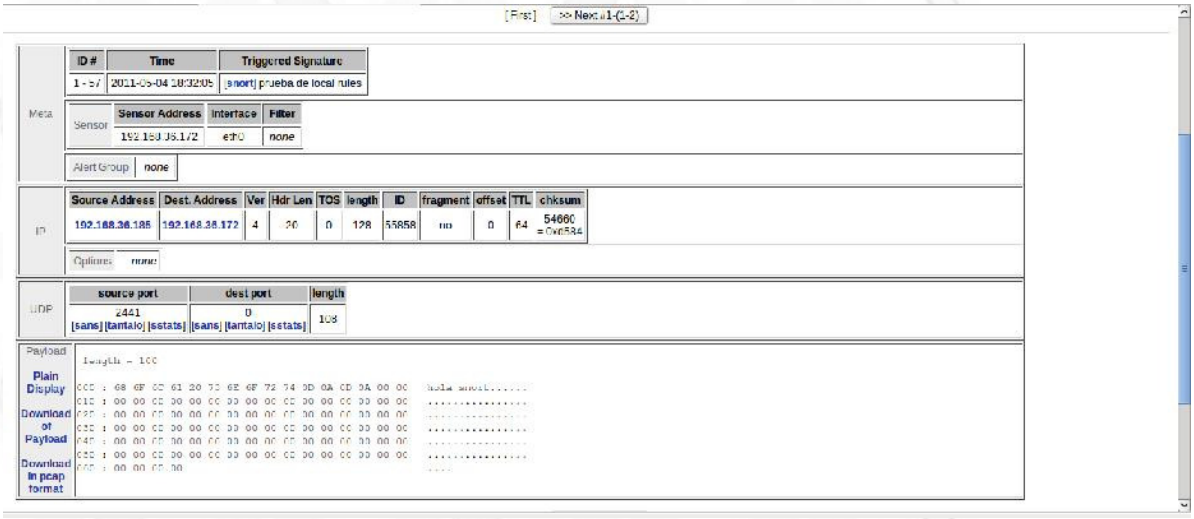
#apt-get install apache2



Como puedes ver en la imagen anterior, esta consola gráfica ya está conectada con la base de datos de Snort y nos comienza a presentar la información detectada por el IDS que configuramos. Desde de la página inicial, puedes ir seleccionando varios campos que te irán ampliando la información de detalle de cada uno de los eventos.



También si lo deseas puedes ampliar la información hasta llegar a ver la totalidad de su contenido, prácticamente como si la visualizaras con un analizador de protocolos, como puedes ver a continuación:



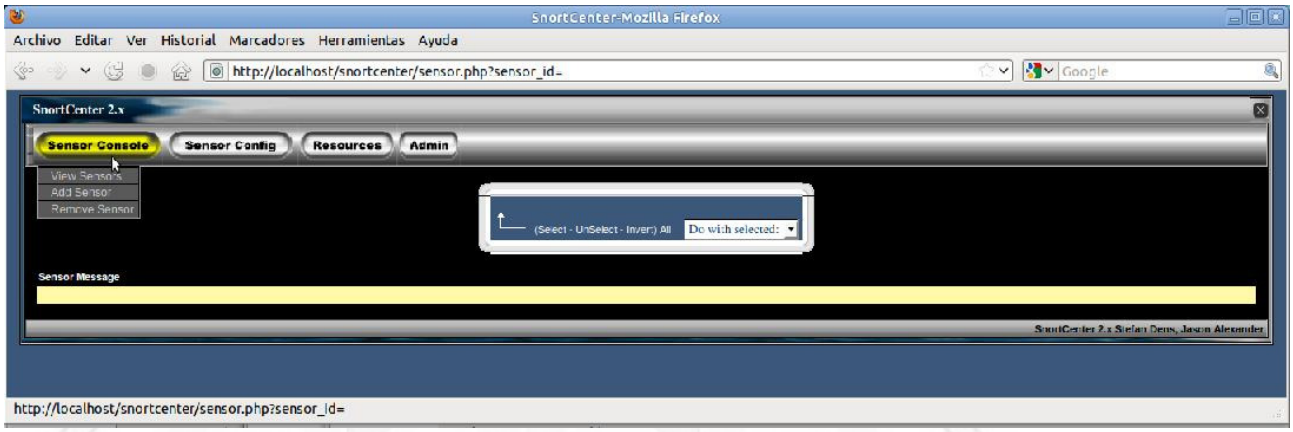
Te proponemos que trabajes y practiques con todas las opciones que ofrece ACID, pues esta será la mejor herramienta que encontrarás para visualizar el estado de tus IDSs.

6) Práctica con Snort Center.

Para finalizar con esta práctica de IDSs, lo último que te proponemos es el trabajo con "Snort Center". Se trata de la herramienta natural que emplea Snort para su administración y configuración. Ya has realizado todas las prácticas con línea de comandos para conocer este IDS con el máximo detalle, con ello has podido comprender todos los aspectos desde su raíz, ahora cuando empieces a trabajar con estas tecnologías en producción, te hará falta una herramienta que te facilite el control de una verdadera infraestructura, la cual suele ser de más de un IDS, para esta tarea es que te invitamos a trabajar con Snort Center.

Los pasos de su instalación no difieren metodológicamente de lo que acabamos de hacer con ACID, así que no los repetiremos, estamos seguros que los encontrarás fácilmente en Internet

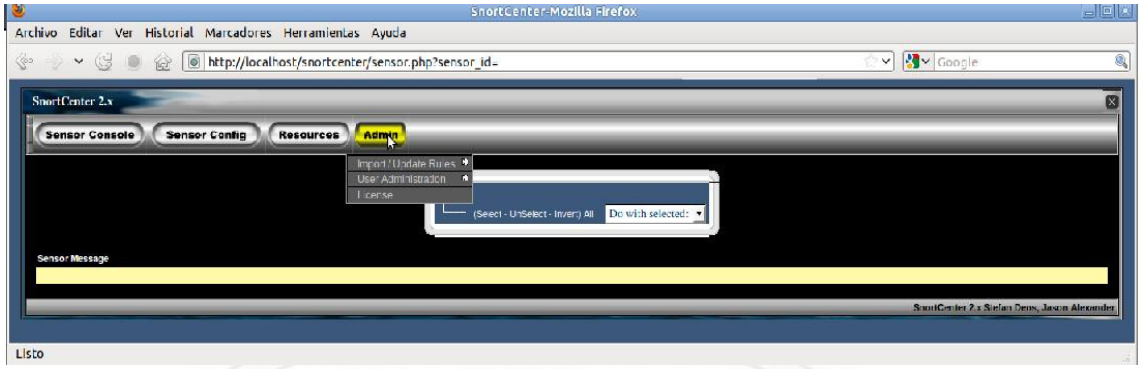
y que no tendrás problemas, por lo tanto te presentamos a continuación la interfaz gráfica de esta herramienta ya instalada.

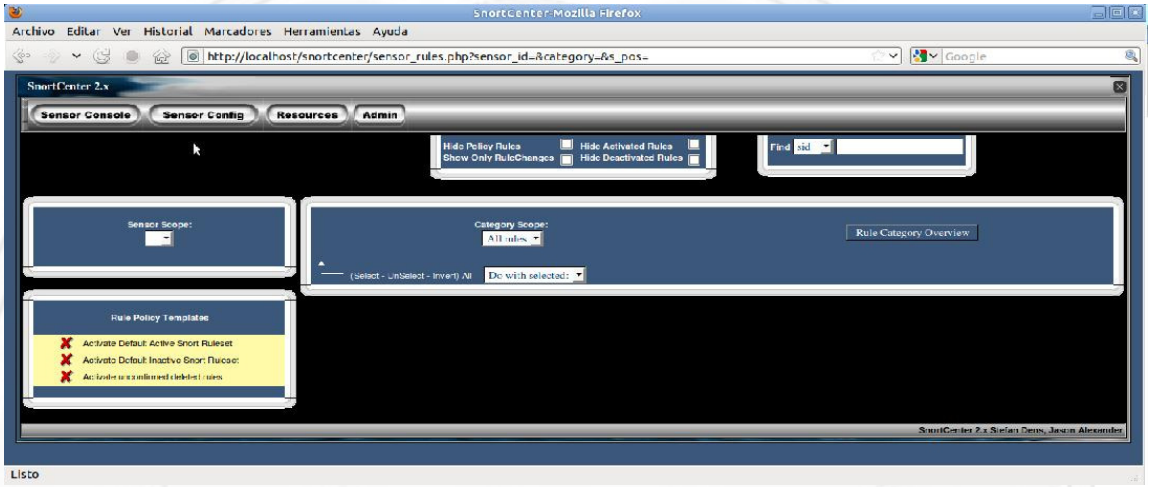


Desde esta interfaz gráfica, puedes realizar absolutamente todas las tareas de configuración de cada uno de los sensores que cuentes en tu infraestructura. En la imagen anterior, hemos seleccionado el botón de "Sensor Console" desde donde puedes ir agregando cada uno de los sensores que desees. Una vez que los has agregado, puedes editarlos y verás al completo el archivo "snort.conf".

Para su configuración está el segundo botón de la barra de menú "Sensor Config". En la imagen siguiente puedes apreciar que desde esta opción puedes configurar todos los "pasos" que te ofrece "snort.conf" (Variables, preprocesadores, reglas, etc…).

El tercer botón "Resources" es el que te permite generar plantillas que puedes aplicar de forma genérica o puntual para uno o varios sensores, y el último botón "Admin" es con el que "activarás" las reglas y administrarás usuarios de esta consola, en este punto debes tener en cuenta algo muy importante: todos los cambios que estés realizando, en realidad los estás haciendo sobre la "consola Snort Center", por lo tanto para que se actualicen en el sensor, debes "seleccionar" el sensor al cual le aplicarás los cambios y actualizarlo sino estos cambios no pasarán a producción, te lo presentamos en las dos imágenes que siguen:





Este tipo de prácticas con interfaz gráfica, consideramos que la mejor forma de entenderlas es justamente probando con ellas en un entorno de maqueta o de prueba, así que te proponemos que le dediques un tiempo a ello e investigues todas las opciones que esta herramienta ofrece, verás que no presenta mayores dificultades y a su vez que podrás llevar a la práctica amigablemente todos los conceptos que has visto, tanto en la teoría como a través de línea de comandos.

5. Ejercicios con metodología "Nessus – Snort".

Ejercicios (lo ideal es trabajar al menos por parejas o más personas):

- Lanzamos un primer "Scan" con Nessus hacia la máquina objetivo, empleando la Política para "Scan de red Interna" por defecto (lo llamamos "scan_01").
- De ser posible, en la máquina destino, capturamos con Wireshark y analizamos el tráfico que se está generando.
- Analizamos el "Report" de Nessus, lo guardamos en diferentes formatos, y analizamos los mismos.
- Creamos una nueva política de "Nessus" para que lance lo mínimo posible (es decir, desactivamos todos los plugins, las preferencias y lo que esté seleccionado en General). La

llamamos "Politica_inicial_nula". En este ejercicio dedicad todo el tiempo que podáis hasta que verdaderamente el scan lanzado sea el mínimo posible (revisad muy bien todas las opciones de configuración de Nessus, pues encontraréis que hay muchos parámetros que podemos desactivar), para verificar en detalle la cantidad de tramas lanzadas, antes de lanzar el scan, lanzad Wireshark filtrando únicamente la captura para estos hosts.

- Abrimos Wireshark en la máquina local.
- Lanzamos un nuevo scan completo (lo llamamos "scan_02"). Nuevamente lo guardamos y analizamos los Reports y lo capturado con Wireshark.
- Identificamos cuáles son los IDs de Nessus de las vulnerabilidades detectadas en el "scan_01". Editamos la "Politica_inicial_nula", filtramos los plugins por ID **y habilitamos solamente uno** de las vulnerabilidades detectadas en el "scan_01".
- Repetimos todo el proceso de lanzamiento de Nessus y captura con Wireshark.
- Analizamos una vez más su Report e intentamos confrontarlo con lo capturado con Wireshark ¿Notas cuál fue el patrón que detecta la vulnerabilidad?
- Para entender un poco más cómo funcionan las reglas de Nessus, nos situamos en el directorio: **/opt/nessus/lib/nessus/plugins**. Si haces un "**ls -l**" allí verás todas las reglas que en este momento tiene definido Nessus.
- Para encontrar una regla por su ID en esta consola, por ejemplo si fuera el ID 52611, puedes ejecutar:

  /opt/nessus/lib/nessus/plugins# grep 52611 *

- Allí te indicará cuál es la "regla.....**.nasl**" que contiene es ID.
- Puedes visualizarla completamente, por ejemplo con "**vi regla.....nasl**". Analiza la misma, mira su familia, descripción, el formato "nasl" (Nessus Attack Scripting Language, o Lenguaje de Scripting de Ataque Nessus), su correlación con Bugtraqs, CVE, etc. si lo posee). Si esta regla en concreto es muy compleja, busca cualquier otra u otra u otra que sea más sencilla y logres entender un poco más su lógica.
- Una vez más compara esta regla con lo capturado con Wireshark para aclarar más estos conceptos hasta que no tengas ninguna duda de cuáles son los bytes, tramas, o procesos que detectan esta vulnerabilidad. El paso final de esta secuencia es tener bien claro de qué forma Nessus llega a la conclusión que su objetivo es vulnerable a esa secuencia o patrón de ataque.

  Esta es la pieza clave, pues es lo mismo que hará un intruso sobre tu sistema, por lo tanto si tu sabes lo que hará este intruso, estarás en capacidad de solucionarlo o al menos de activar la alarma que desees en tu IDS.

Comencemos ahora de a poco a trabajar con Snort (todos estos ejercicios están basados en la teoría y prácticas que hemos venido haciendo con Snort, así que te invitamos a que los sigas con estas páginas a mano para ir consultándolo (si estas leyendo este texto desde un ordenador, lo mejor será que abras otra ventana con el texto, si es en un libro impreso, coloca alguna marca en la teoría y otra en la práctica que acabamos de hacer con Snort).

- Comenzaremos repasando el uso de Snort para que capture tráfico en modo "Sniffer" (que ya conocemos).
- Consulta una vez más si lo deseas en el "**man**" las opciones "**-d**", "**-e**", "**-v**" y ejecuta:

  #snort -dev

- ¿Qué sucede?
- Prueba también las opciones "**-b**" y "**-L**" para que guarde información. una vez guardada, prueba de abrir esos archivos con Wireshark. También puedes probar de abrirlos con la opción "**snort -r**".
- ¿Te acuerdas aún del comando "**tcpdump**". Prueba también con la opción "**-r**" pero con "tcpdump". Compara cómo es la visualización con estos diferentes comandos o modos.
- Verifica el empleo de la opción "**-l**" para especificar el directorio donde se guardan las alarmas. ¿Cómo las guarda?, ¿Con qué comando puedes visualizarlas?
- ¿Qué sucede si lo abres con un editor de texto (por ejemplo "**vi**")?, ¿Qué ves?
- Puedes configurar la interfaz sobre la que va a "escuchar" snort con la opción "**-i**", pruébalo.
- Lanza "Snort" con alguno de los comandos practicados hasta ahora y desde otra máquina, lanza "**Nessus**" hacia esta dirección IP con la política por defecto de "**scan de redes internas**" que acabamos de ver en Nessus. (Si puedes también captúralo con Wireshark).
- Analiza detenidamente los archivos generados por "Snort" con el report de "Nessus". Te invitamos también a que hagas el esfuerzo de "***correlarlo***" con lo capturado con "Wireshark".
- Elige una o dos de las vulnerabilidades detectadas en el report de "Nessus", activa sólo estas en la "Politica_inicial_nula", lanza nuevamente el Scan.
- Analiza una vez más todo lo capturado y generado.
- Como decíamos en este texto, lo más importante de "Snort" es cuando llamamos al archivo "**snort.conf**". El mismo por defecto, te recordamos, se encuentra en: "**/etc/snort/**", vamos a editarlo y analizar los pasos que presenta para su configuración (es bastante claro en cada uno de ellos).
- Para evitar errores, nuestra primera medida será copiar el archivo "**snort.conf**" a otro, por ejemplo "**snort.conf.original**" (podemos hacerlo ejecutando: "**cp snort.conf snort.conf.original**")
- Siguiendo con la secuencia de estos ejercicios, nos centraremos sólo en la sección de "**rules**" y en "habilitar" únicamente las reglas que aplican a las vulnerabilidades que acabamos de detectar con "Nessus". Es decir, un par de líneas más arriba, acabamos de elegir una o dos vulnerabilidades de las detectadas con "Nessus" y las habilitamos en "Politica_inicial_nula", estas vulnerabilidades, seguramente forman parte de una determinada "familia de Nessus", por ejemplo: ftp, telnet, smtp, etc...

  En realidad lo que haremos, más que "habilitar", será "Deshabilitar" pues por defecto, "snort.conf" trae habilitadas todas las reglas. Editemos entonces el archivo "snort.conf" y comentamos (#), o borramos todas las ".rules" que no formen parte de estas familias

mencionadas, dejando sólo estas y también las "local.rules" que veremos un poco más adelante.

- Lanzamos nuevamente "Snort", pero esta vez en modo "IDS", es decir con la opción "-c" ("snort -i eth0 -dev -c /etc/snort/snort.conf").
- Lanzamos nuevamente el Scan con "Nessus" con la "Politica_inicial_nula" hacia este host, de ser posible también capturamos con "Wireshark". Una vez finalizado el Scan, detenemos "Snort" **[Ctrl + Z]**.
- Ahora para visualizar los resultados nos situamos en el directorio que por defecto está configurado "snort.conf" para almacenar los eventos, que es: **/var/log/snort**.
- Seguramente encontremos más de un tipo de archivos. ¿Qué diferencia encuentras entre ellos?
- Analízalos y compara estos con los report de "Nessus" y lo capturado con "Wireshark".
- Puedes seguir adelante con esta práctica, habilitando más plugins de "Nessus" en tu "Politica_inicial_nula" y habilitando más ".rules" en "Snort.conf", lanzar ambos y analizarlos como hicimos hasta ahora.

El objetivo más importante de lo practicado hasta aquí es que si se avanza metódicamente con esta práctica, llegaremos a un punto en el cual **todas** las vulnerabilidades que detecta "Nessus" (y/o absolutamente cualquier otra herramienta) quedarán configuradas en nuestro IDS de forma tal que llevando esto a todos nuestros servidores en producción, llegaremos a tener la certeza de poder detectar cualquier intento de explotación sobre lo que sabemos que es una debilidad.

Muchas de estas vulnerabilidades podrán ser solucionadas, cerradas, parcheadas, etc... pero muchas otras NO y tendremos que convivir con ellas, por esta razón es que es fundamental poseer IDSs que inmediatamente nos puedan informar de un intento de aprovechamiento de las mismas.

Aún nos queda un punto más sobre el que avanzar, y es el caso de una vulnerabilidad que no posea ".rules" nativa de "Snort". En este caso es que debemos ser capaces de generar nuestras propias "**local.rules**", para ello trabajaremos de la siguiente forma:

- En la máquina desde la que lanzábamos "Nessus", crearemos un archivo sencillo de texto plano, lo llamaremos "prueba_snort_01" (el texto que contenga ese archivo será nuestro patrón inicial de detección para Snort, así que por ahora, no lo llenes de texto complicado, coloca sólo un par de palabras, por ejemplo: prueba de snort).
- Vamos a crear nuestra nueva y primera "**local.rules**". Abre con cualquier editor de texto el archivo "local.rules", verás que está creado pero vacío. Inserta una primera regla, algo similar a la que figura abajo:

```
alert udp any any -> any any (msg:"prueba de local rules"; content:"prueba de snort";
classtype:suspicious-login; sid:1000000; rev:1;)
```

- Sus partes se desarrollaron en la teoría y en las prácticas anteriores de Snort, pero presta atención al protocolo "**udp**", a las **IP fuente y destino**, a los **puertos fuente y destino**, al **sentido (->)**. Todo eso es lo que se denomina "Encabezado" de la regla, luego viene el "Cuerpo" (*todo lo que está entre paréntesis*), en nuestro ejemplo inicial, lo que nos interesa es

el parámetro "**content**" que contiene el patrón de búsqueda que queremos localizar: "**prueba de snort**".

- Por ahora practica con lo más básico, es decir, incluye esta primer regla, cierra y guarda el archivo "local.rules" y vuelve a lanzar snort: "snort -i eth0 -dev -c /etc/snort/snort.conf".
- Desde la máquina en la que generaste el archivo "prueba_snort_01" puedes ahora atacar al IDS enviándole el patrón de tráfico que está intentando detectar. Para ello recurriremos, nuevamente a otra vieja herramienta para nosotros "**hping3**" y podemos ejecutar:

```
#hping3 -2 direccion_IP-IDS -d 100 -E prueba_snort_01
```

- Detén ambos comandos, y analiza todo lo sucedido.
- ¿Qué ha detectado por la pantalla de la consola desde la que se ha lanzado Snort?, ¿Compara esto con lo que figura en /var/log/snort?, ¿Qué conclusiones puedes obtener de esto?.
- Te invitamos ahora a que poco a poco vayas ajustando todo lo que desees las reglas de detección, tanto en su "encabezado" como en su "Cuerpo", tienes muchísimas opciones para configurar y llegar a ser un experto en la "detección de intrusiones por medio de tus propias **local.rules**". Si verdaderamente profundizas en este trabajo, encontrarás la máxima potencia que puedes emplear en el trabajo "DETECCIÓN DE VULNERABILIDADES <--> DETECCIÓN DE INTRUSIONES, y verás que es la mejor forma de asegurar un sistema informático, luego de haber pasado por todos los ajustes de nivel de red y de nivel de transporte (con la configuración de direcciones MAC, IP, listas de control de accesos y reglas de Firewalls).
- Te invitamos a ejercitar esta metodología por medio del empleo y generación de tráfico ICMP, UDP, TCP, etc... con las herramientas que fuimos viendo durante este texto: icmpush, hping3, nmap, nikto, etc...
- Puedes ampliar estos ejercicios configurando los distintos tipos de "salidas" de "Snort". Practica los dos modos: "alerta" y "logging", y prueba las diferentes opciones que ofrecen ambos.
- Para finalizar esta práctica sólo te queda profundizar en el ajuste de los "preporcesadores" de "Snort", tema que ya hemos visto también.

6. Más prácticas con "Snort".

1) Prueba con diferentes salidas:

```
snort -c /etc/snort/snort.conf -A fast
snort -l /var/log/snort -c /etc/snort/snort.conf
snort –c /etc/snort/snort.conf -b
```

2) Configurando "snort.conf" con:

```
output alert_fast: alert.log
```

output log_tcpdump: tcpdump.log

output alert_CSV: /var/log/snort/alert.csv default

3) Generar una local.rules para que detecte un patrón de tráfico sencillo (Ej: "hola").

4) Ajustar con mayor detalle esa local.rules

5) Genera una local.rules que detecte un intento de conexión ftp "anonymous" y comprueba su funcionamiento.

6) Activar en "snort.conf", las siguientes reglas:

include $RULE_PATH/scan.rules

include $RULE_PATH/dos.rules

include $RULE_PATH/icmp.rules

7) Generar patrones de tráfico (nmap, hping3, etc...)para hacer saltar las reglas que se detallan a continuación (lo importante de este ejercicio es que aprendas a analizar los patrones de tráfico que están siendo detectados por Snort, es decir que comprendas el significado de esas reglas:

icmp.rules

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP redirect host"; icode:1; itype:5; reference:arachnids,135; reference:cve,1999-0265; classtype:bad-unknown; sid:472; rev:4;)

alert icmp $EXTERNAalert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Source Quench"; icode:0; itype:4; classtype:bad-unknown; sid:477; rev:2;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP redirect net"; icode:0; itype:5; reference:arachnids,199; reference:cve,1999-0265; classtype:bad-unknown; sid:473; rev:4;)

scan.rules

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; flow:stateless; flags:SF,12; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:7;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS"; flow:stateless; flags:SRAFPU,12; reference:arachnids,144; classtype:attempted-recon; sid:625; rev:7;)

alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap XMAS"; flow:stateless; flags:FPU,12; reference:arachnids,30; classtype:attempted-recon; sid:1228; rev:7;)

dos.rules

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"DOS Jolt attack"; dsize:408; fragbits:M; reference:cve,1999-0345; classtype:attempted-dos; sid:268; rev:4;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"DOS ath"; itype:8; content:"+++ath"; nocase; reference:arachnids,264; reference:cve,1999-1228; classtype:attempted-dos; sid:274; rev:5;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 135:139 (msg:"DOS Winnuke attack"; flow:stateless; flags:U+; reference:bugtraq,2010; reference:cve,1999-0153; classtype:attempted-dos; sid:1257; rev:10;)

8) Practica, analiza y verifica alguno de los ejemplos de "local.rules" que figuran a continuación:

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Linux FTP Backdoor {nessus}";flags: AP; content:"PASS null"; nocase; depth: 10; reference:CVE, CAN-1999-0452; classtype:attempted-user; sid:1001015; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Writeable FTP root {nessus}"; flags:A+; content:"STOR nessus_test"; depth: 20; reference:CVE, CAN-1999-0527; classtype:attempted-user; sid:1001002; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:" writeable FTP root {Comando CWD / - nessus}"; flags:A+; content:"CWD /"; depth: 10; reference:CVE, CAN-1999-0527; classtype:attempted-user; sid:1001003; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 513 (msg:"rlogin {nessus}"; flags:A+; content:"root"; nocase; depth: 10; reference:CVE, CAN-1999-0651; classtype:attempted-user; sid:1001004; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"EXPN and VRFY commands {nessus}"; flags:A+; content:"HELO nessus.org"; nocase; depth: 20; reference:CVE, CAN-1999-0531; classtype:successful-recon-largescale; sid:1001005; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP Server type and version {nessus}"; flags:A+; content:"HELP"; depth: 10; classtype:network-scan; sid:1001006; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"WFTP login check {nessus}"; flags:A+; content:"bogusbogus"; depth: 25; reference:CVE, CAN-1999-0200; classtype:attempted-user; sid:1001007; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"IIS 5.0 PROPFIND Vulnerability {nessus}"; flags:A+; content:"PROPFIN"; depth: 25; classtype:attempted-user; sid:1001008; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 513 (msg:"SysV /bin/login buffer overflow (rlogin) {nessus}"; flags:A+; content:"nessus"; depth: 10; reference:url, www.cert.org/advisories/CA-2001-34.html; classtype:attempted-user; sid:1001009; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"FTP Service Allows Any Username {nessus}"; flags:A+; content: "user pp *pass pp"; regex; nocase; depth: 20; cla sstype:attempted-user; sid:1001010; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 515 (msg:"lpd, dvips and remote command execution {nessus}"; flags:A+; content:"|F702 0183 82C0 1C3B 0000 0000 03E8 1B20 5463 5820 6F75 7470 7574 2032 3030| "; depth: 30; reference:CVE,CAN-2001-1002; classtype:attempted-user; sid:1001011; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"SMTP antivirus filter {nessus}"; flags:A+; content:"HELO nessus"; nocase; depth: 15; classtype:attempted-recon; sid:1001012; rev:1;)

alert tcp $EXTERNAL_NET 20 -> $HOME_NET 8888 (msg:"BenHur Firewall active FTP firewall leak {nessus}"; classtype:attempted-recon; sid:1001013; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"EXPERIMENTAL WEB-MISC OpenSSL Worm traffic"; content:"TERM=xterm"; nocase; classtype:web-application-attack; reference:url,www.cert.org/advisories/CA-2002-27.html; sid:1001014; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 443 (msg:"deteccion 2 WEB-MISC OpenSSL Worm traffic"; content:"|4745 5420 2F20 4854 5450 2F31 2E30 0D0A 0D0A|"; flags: AP; classtype:web-application-attack;reference:url,www.cert.org/advisories/CA-2002-27.html; sid:1001016; rev:1;)

alert udp $EXTERNAL_NET any -> $HOME_NET 137 (msg:"Using NetBIOS to retrieve information from a Windows host {nessus}"; content:"|0000 0001 0000 0000 0000 2043 4b41 4141 4141 4141 4141 4141 4141 4141 4141|"; depth: 33; classtype: attempted-dos; sid: 1001017; rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"gusano referente a lsass.exe"; content:"lsass.exe"; offset: 40; depth: 50; classtype:web-application-attack; sid:1001017; rev:1;)

alert tcp any any -> $HOME_NET 24 (msg: "aaaaaaaaa"; content: "a*sh"; regex; classtype: web-application-attack; sid: 1001018; rev: 1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"Alibaba 2.0 buffer Overflow {nessus}"; content: "POST XXXXXXXX";

depth: 15; classtype:web-application-attack;reference:CVE,CAN-200-0626; sid:1001019; rev:1;)

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"Axent Raptor DoS {nessus}"; tos: 123; id:1234; ttl: 255; ip_proto: 6; reference: CVE,CVE-1999-0905; classtype: attempted-dos; sid:1001021;rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 514 (msg:" possible rsh scan with null username {nessus}"; content:"|0204|"; flags: S; reference: CVE,CVE-1999-0180; classtype: attempted-recon; sid:1001022;rev:1;)

alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"NT IIS Malformed HTTP request Header DoS Vulnerability {nessus}"; content:"|686f 7374 6e61 6d65 203a 2058 5858 5858|"; depth:20; reference: CVE,CVE-1999-0867; classtype: attempted-dos; sid:1001023; rev:1;)

alert ip $EXTERNAL_NET any -> $HOME_NET any (msg:"pimp {nessus}"; id: 69; ip_proto: 2; ttl: 255; content: "|5858 5858 5858 5858 5858|";offset: 10; depth:15; classtype: attempted-dos; sid:1001024; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"Sendmail redirection check {nessus}"; content:"|5243 5054 2054 4f3a 2072 6f6f 7440 686f 7374 3140|"; classtype: attempted-recon; sid:1001025; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"Sendmail from piped program {nessus}"; content:"|4d41 494c 2046 524f 4d3a 207c 7465 7374 696e 67|"; classtype: attempted-recon; sid:1001026;rev: 1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"Sendmail mailing to files {nessus}"; content:"|5243 5054 2054 4f3a 202f 746d 702f 6e65 7373 7573 5f74 6573 74|"; classtype: attempted-recon; sid:1001027;rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 25 (msg:"Sendmail mailing to programs {nessus}"; content:"|5243 5054 2054 4f3a 207c 7465 7374 696e 67|"; classtype: attempted-recon; sid: 1001028; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Anonymous FTP enabled {nessus}"; flags:A+; content:"USER null"; nocase; depth: 10; reference:CVE, CAN-1999-0452; classtype:attempted-user; sid:1001001; rev:1;)

7. Práctica con Honey net

Toda esta práctica, la haremos una vez más desde Linux con distribución Debian, empleando la herramienta "**honeyd**" que consideramos muy eficiente para nuestros ejercicios.

1) Instala honeyd (directamente desde Synaptics)

2) Mira los archivos que se instalaron en /etc/honeypot

3) Instala un archivo de configuración sencillo (copia y pega el que figura aquí abajo) y dale un nombre cualquiera "nombre_archivo"

```
route entry 192.168.36.129
route 192.168.36.129 link 192.168.36.128/25
```

```
route 192.168.36.248 unreach 32.0.0.0/3

### Default Template
create default
# Set default behavior
#set default personality "Windows NT4 / Win95 / Win98"
set default default tcp action reset
set default default udp action reset
set default default icmp action open

# Add specific services
add default tcp port 139 open
add default tcp port 137 open
add default udp port 1337 open
add default udp port 135 open
add default tcp port 31337 open

create router
#set router personality "Cisco IOS 12.1(5)-12.2(1)"
set router default tcp action reset
set router default udp action reset
set router uid 32767 gid 32767
set router uptime 1327650
add router tcp port 23 "perl scripts/router/cisco/router-telnet.pl"
add router tcp port 80 open
add router tcp port 443 open
add router udp port 161 "perl scripts/unix/general/snmp/fake-snmp.pl public
private --config=scripts/unix/general"
bind 192.168.36.254 router

### Dynamic honeypot
dynamic magichost
add magichost use router if time between 12:00am - 5:00pm
add magichost otherwise use default
bind 192.168.36.150 magichost
bind 192.168.36.151 magichost
```

4) Ejecuta el comando: "farpd 192.168.36.128/25" ¿Para qué sirve?, ¿Por qué hemos elegido este rango y máscara?

5) Posiciónate en el directorio **/etc/honeypot**.

6) Ejecuta el comando "honeyd -d -f nombre_archivo"

- ¿Qué sucede?
- ¿Para qué es la opción "**-d**"
- ¿Para qué es la opción "**-p**"

7) Desde otro host, realiza ping a direcciones de ese rango de red.

8) Desde otro host, realiza escaneos a las direcciones que figuran en el archivo de configuración de honeyd, ¿Qué respuestas obtenemos?

9) Detiene el programa "Honeyd" (con **[ctrl]** + **"C"**).

10) Modifica algunos puertos, agrega otros, etc. y lanza nuevamente el demonio.

11) Desde otro host, ejecuta "**telnet**" hacia alguna de las direcciones de ese rango de red. ¿Qué sucede?

12) Detén el demonio, edita nuestro archivo de configuración y agrega una nueva línea:

```
add default tcp port 23 "perl /usr/share/honeyd/scripts/router-telnet.pl"
```

y comenta la línea existente:

```
#add router tcp port 23 "perl scripts/router/cisco/router-telnet.pl"
```

Lanza nuevamente el demonio.

13) Desde otro host, ejecuta "**telnet**" nuevamente hacia alguna de las direcciones de ese rango de red. ¿Qué sucede ahora?

14) Vete a la URL: **http://www.honeyd.org** (navega por ella e investiga opciones)

15) Desde esta misma URL descarga "**Sample Configuration**" y poco a poco ve agregando a nuestro archivo inicial de configuración nuevas opciones.

16) Ejecuta el comando "honeyd -d -f pp -p nmap.prints"

17) Prueba con diferentes tipos de escaneos nmap sobre ese rango de red.

DESAFÍOS DE NIVEL APLICACIÓN.

1. "Desafío extremo": En el capítulo de red hablamos del empleo de NAT. Luego en el de transporte tratamos puertos y sockets. En este capítulo tratamos el protocolo "FTP" y dijimos que abre dos puertos (20 y 21). ¿Te atreves a investigar entonces, cómo se hace en NAT en este caso?..... te damos una pista para empezar: la **RFC-3022**.
2. Luego de instalar BIND, ¿Te atreves ahora a configurarlo como DNS master?
3. Referido a SNMP, hemos visto ejercicios y comandos a través de "**net-snmp**", una sencilla interfaz gráfica con el comando "**tkmib**". Te proponemos que investigues las siguientes interfaces gráficas para emplearlas con SNMP:
   - MRTG.
   - NAGIOS.
   - CACTI.
4. Investiga otras herramientas de trabajo para http (generadores de tráfico, medidores de tráfico, cookies, posicionamiento, etc).
5. Investiga y prueba el funcionamiento de proxies (Te aconsejamos SQUID).
6. Evalúa diferentes herramientas de trabajo con el protocolo "netBIOS" y con los comandos "net".
7. Participa en los diferentes grupos de discusión de Snort y Nessus, hasta puedes llegar a proponer nuevas reglas para ambos (hay miles de colaboradores que lo hacen, ¿te atreves?).

CAPÍTULO 8: ALGUNOS CONCEPTOS MÁS

8.1. Breves conceptos de criptografía

La criptografía es la ciencia que permite convertir un determinado conjunto de códigos generalmente comprensibles en otro cuyas características lo hacen de difícil interpretación.

Este proceso se lleva a cabo a través de algoritmos matemáticos combinados con claves. El mayor o menor grado de complejidad del algoritmo, combinado con una determinada longitud de clave, harán que el nivel de dificultad para poder descifrar o interpretar un mensaje sin ser parte de los miembros que realizan estos pasos (es decir intruso) sea una tarea simple, dificultosa o prácticamente imposible.

Este arte es milenario y se empleó generalmente en el transporte de información de carácter militar o política. En sus comienzos se utilizaba para mensajes en tránsito, pero casi asociado a la aparición de la informática se comienza a hacer necesario su empleo también en el almacenamiento de la información, pues la misma se presenta ahora de una manera más accesible a individuos que no necesariamente deberían tener acceso a ella. Es evidente que con la aparición de las redes y la posibilidad de interconexión que existe hoy, la importancia de mantener la confidencialidad de la información por todos los métodos existentes, hace necesario el uso de esta herramienta en muchas actividades cotidianas.

Uno de los conceptos a tener en cuenta es que la información en tránsito en todos los casos es escuchada por muchas personas, ya sea dentro de una red LAN como cuando se desplaza a través de vínculos WAN tanto dedicados como por medio de redes X.25 o Frame Relay. Cualquiera que posea las herramientas adecuadas puede decodificar los distintos encabezados de estas tramas o paquetes y acceder a la información que está viajando.

Al cifrar información, si se intenta interpretarla, será necesario poder revertir el proceso, para volver al mensaje original. Como se mencionó con anterioridad, el empleo de la clave es lo que hace posible esta actividad, la cual es análoga a la llave de un candado o cerradura y el algoritmo es el conjunto de engranajes o mecanismos. La cantidad de trabas o bolillas que posea esta cerradura, determinará la cantidad de posibilidades que existen para abrir o cerrar ese acceso. Si se mantiene este ejemplo para el caso de un candado de clave numérica, éste podría tener cuatro ruedas de diez dígitos de las cuales una sola combinación liberaría el mecanismo, generando una posibilidad entre diez mil. Criptográficamente hablando, tendría un **ESPACIO DE CLAVES** de 10.000 posibilidades.

Si alguien tuviera acceso a este candado sin conocer la combinación (Clave), podría probar desde el 0000, luego el 0001, el 0002..... y así sucesivamente hasta lograr abrir el candado. Nuevamente en terminología criptográfica, esto se llamaría **FUERZA BRUTA**. Se debe tener en cuenta que si este intruso pudiera tener algún indicio respecto a esta combinación, como por ejemplo que la misma empieza con el dígito cuatro, el espacio de claves queda reducido a cien, con lo cual el tiempo necesario para abrir el candado sería diez veces menor. Esta última es de especial interés pues en informática existe un sinnúmero de métodos para reducir el ESPACIO DE CLAVES, minimizando el tiempo necesario para vulnerar textos cifrados. Por ejemplo, un método

muy conocido de cifrado (que se mencionará posteriormente) es el **DES** que tiene una clave de 64 bit. A primera vista se pensaría que su espacio de claves es 2^{64}, pero como primer detalle, esta clave emplea 8 bit de paridad, con lo cual ya se tiene un espacio de claves de 2^{56} claves independientes. Como segunda medida se puede emplear por ejemplo un método llamado criptoanálisis diferencial que reduce las posibilidades a 2^{34}, con lo cual se reducen los tiempos exponencialmente. Esto se menciona sólo a título de ejemplo, pues si se tiene en cuenta que cada intento de estos trae aparejado la comparación de un texto para verificar si es entendible, la realidad hace que si se trata de un mensaje extenso esta actividad sea poco rentable, existiendo métodos de análisis mucho más eficientes.

Otro detalle significativo es que la **ENTROPIA** de los códigos reales no es máxima en ningún caso, es decir que la frecuencia de aparición de los distintos símbolos no es equiprobable. Si se analiza cualquier texto escrito en Castellano, se podría apreciar que determinadas letras tienen mayores ocurrencias de aparición que otras, por ejemplo la E, N, S, A, etc. y por el contrario es poco frecuente encontrar X, W. Esta característica es de sumo interés pues un mensaje cifrado si se puede determinar su alfabeto fuente (por ejemplo Castellano) allanará las posibilidades de criptoanálisis. Es evidente que este detalle causa más impacto cuanto mayor sea la longitud del texto, pues en unas pocas palabras esta frecuencia de aparición puede no ser cierta, y absolutamente lo será si la longitud tiende a infinito. También se tiene en cuenta el **CODIGO EN SU EXTENSION**, pues volviendo al mismo ejemplo, la secuencia de extensión dos TH en Castellano tiene una probabilidad excesivamente baja pero en Inglés es cotidiana; más aún si se extiende en grado tres THR en Castellano no existe y en Inglés sí. Estos breves conceptos sirven para ilustrar escuetamente cómo se puede reducir de distintas maneras la tarea de **criptoanalizar** un texto cifrado, desde ya que estos conceptos son básicos, y que en la actualidad esta actividad se encuentra altamente avanzada y tremendamente favorecida por las altas velocidades de procesamiento que existen hoy.

8.1.1. Algoritmos de autenticación y cifrado

A continuación presentamos algunos de estos algoritmos, los cuales se irán desarrollando a lo largo de este capítulo:

- HMAc con MD5 [RFC-2403]
- HMAC con SHA-1 [RFC-2404].
- HMAC [RFC-2104].
- Diffie-Hellman.
- DES (Data Encryption Standard) [ANSI X3.106] en modo CBC.
- MD5 (Message Digest Algorithm Versión 5) [RFC-1321] y SHA (Secure Hash Standard) [FIPS- 180-1, de NIST].
- 3DES (triple DES) para cifrado.
- Tiger para Hash.

- DSS (Digital Standard Signature).
- RSA (Rivest, Shamir and Adleman).

8.1.2. Empleo y conceptos de clave simétrica y asimétrica

El principio de funcionamiento de los distintos algoritmos criptográficos y sus claves correspondientes, se puede clasificar clásicamente de tres formas: **Simétrico, asimétrico e híbrido**, siendo este último una combinación de los dos anteriores. Se podría considerar una cuarta forma que se debería llamar **irreversible** (como se puede considerar el sistema de claves de Unix)**.** A continuación se detallan cada una de ellas.

El concepto de "**clave**" en general se confunde con la "**contraseña**" que en la mayoría de estas aplicaciones debe ingresar el usuario para acceder a la "aplicación de la clave", es decir la clave es un valor que en casi todos los programas de cifrado, se genera a través de un algoritmo y/o función matemática y su valor es de compleja determinación, en cambio la contraseña que ingresa el usuario, por más que se recomiende que sea un valor de cierta longitud, combinando mayúsculas, números y caracteres especiales, así y todo se debe elegir un valor "recordable" y en general no llegará a tener la magnitud criptográfica de una "clave de cifrado". Tal vez aún nos suene confusa esta diferencia, por esa razón creemos que es mejor seguir adelante con este capítulo y que poco a poco podamos ampliar el tema.

8.1.3. Cifrado simétrico

Este tipo de algoritmo, emplea la misma clave para cifrar que para descifrar. Este algoritmo es el primero de todos, y se empleó casi con exclusividad hasta principios de los ochenta. Los ejemplos más conocidos son el DES, Triple DES, CAST, RC 4 y RC 5, Blowfish, IDEA, y CAST.

Como ejemplo simple, se desea cifrar el siguiente mensaje (con módulo 256):

código ASCII: $Mje_{fuente} = \{77, 69, 83, 65\} = \{MESA\}$

El algoritmo consiste en sumarle siete a cada símbolo (Clave = 7).

$Mensaje_{Cripto} = \{84, 76, 90, 72\} = \{T, L, Z, H\}$

Para descifrarlo, el algoritmo será la resta, y la Clave será la misma = 7 .

Este ejemplo que a simple vista parece trivial, de hecho no lo es tanto pues fue utilizado durante muchos siglos, y se llamaba el algoritmo del César. Se puede hacer la prueba de redactar un texto cualquiera y emplear distintas claves, comprobando que no es tan simple descifrarlo, también se puede incrementar la complejidad del algoritmo con distintas operaciones simples e inclusive con combinaciones de ellas, logrando paso a paso un grado de dificultad cada vez mayor.

Como conclusión, se puede apreciar que se emplea la misma clave para cifrar que para descifrar, y el algoritmo es la inversa. Si se desea incrementar el grado de dificultad, se realiza fácilmente, incrementando la cantidad de operaciones y la longitud de la clave.

El gran problema radica en la forma en la cual se hace llegar la clave pues debería ser por otro medio de comunicaciones debido a que justamente éste no es seguro. Este método posee la gran debilidad que esta clave no puede ser difundida pues a medida que más de dos personas conocen un secreto, éste poco a poco va dejando de serlo. El gran inconveniente radica en que se está creyendo que la información es confidencial y sobre este concepto se fundamenta la toma de decisiones, siendo esto más peligroso que si se es consciente que la información puede ser escuchada y se opera al respecto.

8.1.4. Cifrado asimétrico

A mediados de los años 70‘ se descubre esta nueva técnica que permite el cifrado de una manera diferente, haciendo especial hincapié en la preservación del secreto, el cual como se mencionó anteriormente, sólo es seguro si lo conoce una sola persona (es decir el propietario) pues si ya se difundió a alguien más se pierde la certeza de su no distribución, los algoritmos más empleados en la actualidad son **RSA** y **Diffie-Hellman**. Se basan todos en un par de claves denominados “Pública y privada”. Hoy básicamente, existen tres tipos de problemas matemáticos que fundamentan este cifrado:

- Logaritmos entero discretos.
- Factorización de números.
- Curvas elípticas.

Esta técnica se basa en el empleo de dos claves, una **PÚBLICA** la cual se difunde sin ninguna limitación, como se hace con un número telefónico en la guía correspondiente. Esta clave es la que se emplea para emitir un mensaje cifrado, es decir la emplea cualquier persona que desee enviar un mensaje seguro hacia un determinado remitente (por supuesto que cada remitente tendrá su propia clave pública, la cual dará a conocer a todas las personas que necesiten emplearla sin limitación). La segunda es la clave **PRIVADA**, la cual es conocida únicamente por su propietario y nadie más, y es la que se emplea para descifrar el mensaje cifrado con su correspondiente clave pública. Como se puede suponer, ambas claves están asociadas de alguna manera, conformando un PAR DE CLAVES, causa por la cual no es imposible partiendo desde la clave pública obtener la privada, pero en la actualidad aún es excesivamente cara la inversión de tiempo, recursos y algoritmos necesarios para hacerlo, y más aún a medida que se emplean claves extensas; esta cierta complejidad es la que hace a esta técnica altamente confiable y la convierte en la más segura que se emplea en la actualidad.

La lógica es la siguiente:

Mje $_{fuente}$ = {M}

CLAVE PÚBLICA <Mje $_{fuente}$ = {M}> = Mje $_{cripto}$ ⟶ Emite

Recibe ⟶ CLAVE PRIVADA <Mje $_{cripto}$> = Mje $_{fuente}$

En este método se soluciona el problema de la distribución de claves, pues no se necesita otro canal o procedimiento para esta tarea, pues no es necesario.

El primer problema que se plantea es cómo se puede estar seguro que la clave pública que dice ser, realmente es; es decir una persona puede confiar en la guía telefónica que publica anualmente la o las empresas de telefonía local porque sabe que las mismas fueron impresas por esta empresa y es su responsabilidad ser veraces, también se puede confiar en un teléfono o dirección que me suministra una persona conocida siempre y cuando la misma esté considerada como "confiable"; pero qué sucedería si la clave pública que se obtiene no es en realidad la que se corresponde con el remitente al cual se le desea enviar un mensaje cifrado, como ejemplo se presenta el siguiente:

PROCEDER CORRECTO

A desea enviar un mensaje a **B**.

A busca en la guía **G** la clave pública de **B**, la cual será $\mathbf{K_{B(Pub)}}$

Redacta el mensaje **M** = **Mje M**.

Lo cifra con la clave pública de **B**, ⟶ $\mathbf{K_{B(Pub)}}$ {**Mje M**} = **Mje** $_{\mathbf{cripto}}$

Lo envía a **B**.

B lo recibe.

B lo descifra con su clave privada $\mathbf{K_{B(Priv)}}$ ⟶ $\mathbf{K_{B(Priv)}}$ {**Mje** $_{\mathbf{cripto}}$} = **Mje M**

Obteniendo el mensaje original, **Mje M.**

PROCEDER INCORRECTO

J publica su lista falsa de claves públicas.

A desea enviar un mensaje a **B**.

A busca en la guía **J** la clave pública de **B**, la cual será $\mathbf{K_{B(Pub\ falsa)}}$

Redacta el mensaje **M** = **Mje M**.

Lo cifra con la clave púB. falsa de **B**, ⟶ $\mathbf{K_{B(Pub\ falsa)}}$ {**Mje M**} = **Mje** $_{\mathbf{cripto\ falso}}$

Lo envía a **B**.

J lo escucha y lo descifra con la clave privada falsa de **B**, ⟶

$\mathbf{K_{B(Priv\ falsa)}}$ {**Mje** $_{\mathbf{cripto\ falso}}$} = Mje M

J cifra el mensaje con la verdadera clave pública de **B**, ⟶

$\mathbf{K_{B(Pub)}}$ {**Mje M**} = **Mje** $_{\text{cripto}}$

J lo envía a **B**.

B lo recibe.

B lo descifra con su clave privada $\mathbf{K_{B(Priv)}}$, ⟶ $\mathbf{K_{B(Priv)}}$ {**Mje** $_{\text{cripto}}$} = **Mje M**

Obteniendo el mensaje original, **Mje M**.

Nótese cómo un intruso obtuvo el mensaje original, el cual desde ya que si se realiza el proceder inverso, también se tomará conocimiento de la respuesta de B hacia A.

Este como se mencionó es el primer problema que plantea este algoritmo, y es por esta razón que es de suma importancia **garantizar la veraz distribución de claves públicas**, la cual se realizará a través de listas conocidas de distribución que puedan garantizar la consistencia de sus datos o lo que es más eficiente a través de la seguridad individual de cada emisor, el cual deberá estar plenamente convencido de la confiabilidad de las fuentes de obtención, lo cual puede ser a través del propio receptor telefónicamente, vía terceros que son de confianza, dependencias dentro de la organización que garanticen, etc.

El segundo gran problema que presenta este método es la demora que introduce (llegando a ser en algunos casos hasta mil veces más lentos que las técnicas simétricas), y el mayor volumen de información que genera. Ambos problemas son naturales en todo proceso que se desee optimizar la seguridad, como regla general:

SIEMPRE QUE SE INCREMENTA LA SEGURIDAD, SE INTRODUCEN DEMORAS.

Esto da origen al cifrado híbrido.

8.1.5. Cifrado híbrido.

Esta técnica es la que se emplea en la mayoría de las aplicaciones de software comercial, y se basa en el empleo de los dos algoritmos (simétrico y asimétrico), para mejorar la velocidad y el volumen de datos, se procede a cifrar todo el mensaje con clave simétrica, luego se toma esta clave y se la cifra con clave pública del destinatario enviando todo el conjunto; el receptor entonces recibe un mensaje que en realidad está compuesto de dos partes, la primera de ellas es la clave simétrica cifrada a través de una clave pública; sobre esta parte se aplica la clave privada obteniendo como resultado la clave simétrica original, una vez obtenida esta, se aplica sobre la segunda parte (mensaje cifrado), obteniendo el texto original.

NOTA: Esta técnica en general para optimizar la seguridad se suele emplear generando en cada sesión una clave simétrica en forma aleatoria, es decir para cada sesión se implementará una clave simétrica diferente. Esta mejora permite incrementar el algoritmo pues aún en el caso de lograr descifrar la clave simétrica, esta sola sería de utilidad para esa sesión y nada más.

Cifrado Irreversible:

Este procedimiento, en particular nos gusta denominarlo así aunque es posible que no encuentres este nombre en otros textos. Consiste en poder cifrar código, pero si bien puede existir un método de descifrado (o no), éste no se difunde o no se emplea. Su aplicación más común se puede observar en la forma en que los sistemas operativos de red suelen tratar las cuentas de usuario y contraseñas, como por ejemplo Unix o Windows a partir de NT. Estos sistemas operativos, al crear una cuenta de usuario de red, la guardan en archivos dentro de alguna estructura de directorios en forma cifrada, no es un mero resumen, pues allí interviene justamente este secreto que ingresó el usuario. Al hacerse presente este usuario en la red, solicita validarse ante un servidor, colocando su nombre de usuario y contraseña. El o los servidores que reciben esta petición, aplican el mismo algoritmo de cifrado y comparan los resultados, si son los mismos códigos, lo reconocen como usuario de red, caso contrario le niegan el acceso. Lo importante a tener en cuenta es que en ningún momento se compara texto plano, sino código cifrado, por lo tanto no se necesita el procedimiento inverso para descifrar.

Lamentablemente este tipo de cifrado tiene poca difusión, pues tiene la enorme potencia de no poder volver atrás. Si se tiene en cuenta este detalle, se podría implementar en discursos políticos, programas de TV, etc, en los cuales sería importantísimo que ni siquiera el que generó el discurso o diálogo pueda volver a repetirlo.

8.1.6. Función HASH (o resúmenes)

Una función **HASH** o también llamados resúmenes, tiene por objetivo lograr un extracto (siempre de igual tamaño) de cualquier tipo de archivo binario, con el propósito que se puede generar una relación vinculante EN UN SOLO SENTIDO desde el documento hacia el HASH.

Propiedades de una función HASH

Una función HASH "h" aplicada a un archivo "M", es decir, h(M) será segura si tiene las siguientes características:

a. **Unidireccionalidad**: conocido un resumen h(M), debe ser computacionalmente imposible encontrar M a partir de dicho resumen.

b. **Compresión**: a partir de un mensaje de cualquier longitud, el resumen h(M) debe tener una longitud fija. Lo normal es que la longitud de h(M) sea menor que el mensaje M.

c. **Facilidad de cálculo**: debe ser fácil calcular h(M) a partir de un mensaje M.

d. **Difusión**: el resumen h(M) debe ser una función compleja de todos los bits del mensaje M: si se modifica un solo bit del mensaje M, el hash h(M) debería cambiar la mitad de sus bits aproximadamente.

e. **Colisión simple**: será computacionalmente imposible conocido M, encontrar otro M' tal que h(M) = h(M'). Esto se conoce como *resistencia débil a las colisiones*.

f. **Colisión fuerte**: será computacionalmente difícil encontrar un par (M, M') de forma que h(M) = h(M'). Esto se conoce como *resistencia fuerte a las colisiones*.

Tipos de resúmenes

- **MD5**: Ron Rivest 1992. Mejoras al MD4 y MD2 (1990), es más lento pero con mayor nivel de seguridad. Resumen de 128 bits. (RFC:1321)
- **SHA-1**: Del NIST, National Institute of Standards and Technology, 1994. Similar a MD5 pero con resumen de 160 bits. Existen otras nuevas versiones conocidas como SHA-256 y SHA-512.
- **RIPEMD**: Comunidad Europea, RACE, 1992. Resumen de 160 bits.
- **N-Hash**: Nippon Telephone and Telegraph, 1990. Resumen: 128 bits.
- **Snefru**: Ralph Merkle, 1990. Resúmenes entre 128 y 256 bits. Ha sido criptoanalizado y es lento.
- **Tiger**: Ross Anderson, Eli Biham, 1996. Resúmenes de hasta 192 bits. Optimizado para máquinas de 64 bits (Alpha).
- **Panama**: John Daemen, Craig Clapp, 1998. Resúmenes de 256 bits de longitud. Trabaja en modo función hash o como cifrador de flujo.
- **Haval**: Yuliang Zheng, Josef Pieprzyk y Jennifer Seberry, 1992. Admite 15 configuraciones diferentes. Hasta 256 bits.

Comparativa entre MD5 y SHA-1

SHA-1 genera una salida de 160 bits de longitud mientras que MD5 genera sólo 128 bits.

- La dificultad de generar un mensaje que tenga un resumen dado es del orden de 2^{128} operaciones para MD5 y 2^{160} para SHA-1.
- La dificultad de generar dos mensajes aleatorios distintos y que tengan el mismo resumen (ataques basados en paradoja del cumpleaños) es del orden de 264 operaciones para MD5 y 280 para SHA-1.

Esta diferencia de 16 bits a favor de SHA-1 lo convierte en más seguro y resistente a ataques por fuerza bruta que el algoritmo MD5. Aunque es más lento que MD5, SHA-1 es hoy el estándar como función hash. En la actualidad ya se está empezando a trabajar con SHA-256.

Ataques a las funciones HASH

- Ya a finales del año 2004 científicos chinos de la Shandong University presentan trabajos en los que se analizan las debilidades reales de las funciones hash como MD5 y SHA-1 ante colisiones.

- Aunque no está claro que este tipo de ataques pudiese derivar en acciones de fraude, como sería suplantar un hash por otro igual y que en recepción se aceptase como válido si bien este último proviene de un mensaje distinto, es un motivo de preocupación actual.
- El problema de estas vulnerabilidades estriba en que muchos servidores Web presentan un certificado digital X.509 firmado a partir de una función hash MD5 y, en el mejor de los casos aunque todavía en muy pocos, con un hash SHA-1.... pero éste también ha sido criptoanalizado.

Como se verá más adelante las funciones hash vistas (MD5, SHA-1, etc.) pueden usarse además para autenticar a dos usuarios.

- Entre ellos está HMAC, una función que usando los hash vistos y una clave secreta, autentica a dos usuarios mediante sistemas de clave secreta. Las funciones MAC, Message Authentication Code, y HMAC se tratarán en el próximo capítulo dedicado a la autenticación y firma digital.
- HMAC se usa en plataformas IP seguras como por ejemplo en Secure Socket Layer, SSL.

8.1.7. Métodos de autenticación y no repudio

1) Diffie_Hellman.

El problema del intercambio de claves es poder ponerse de acuerdo sobre un secreto compartido a través de un canal de comunicaciones no seguro (o público). El esquema propuesto por Diffie-Hellman es el siguiente:

- El **Equipo A** elige dos valores de clave pública un **número primo m (grande)** y otro número **g (más pequeño)** el cual es un generador de módulo m, es decir que el resto de g^a/m (o en lenguaje informático: g^a mod m) generará un número n tal que $0 \leq n < m$ para cualquier valor de a (y todos distintos).
- A elige también una **clave secreta x** < m, luego calcula X = resto (g^x/m) y le envía al Equipo B los siguientes tres valores: m, g, X.
- B recibe estos valores y elige una clave secreta y < m, calcula Y = resto (g^y/m), y le transmite el valor de Y a A.

NOTA: tener en cuenta que en ningún momento se han transmitido las claves secretas **x** e **y.**

- A calcula un nuevo valor S = resto (Y^x/m).
- B calcula un nuevo valor S = resto (X^y/m).

NOTA: Tener en cuenta que cada uno lo hace con su clave secreta y empleando un cálculo que lo relaciona con la clave secreta del otro (X e Y).

- La característica (que se demuestra matemáticamente) que presenta S es que:

$$\mathbf{S}=resto(X^y/m)=resto(Y^x/m)=resto[(g^x)^y/m]=resto[g^{(xy)}/m]=resto[(g^y)^x/m]=resto[g^{(yx)}/m]$$

- Por lo tanto el valor **de S es el secreto compartido** entre A y B.
- En el caso que alguien deseara poder descubrir este secreto, contaría con la información que fue enviada por el canal de comunicaciones es decir: m. g, X e Y. Para participar de este secreto debería resolver una ecuación del tipo Z = resto (g^z/m) para un z desconocido. Esto es conocido como un problema de logaritmos discretos, lo cual en la actualidad es **computacionalmente imposible de resolver para algoritmos que empleen números primos suficientemente extensos**.
- Los parámetros soportados son hasta 512 bit.

  NOTA: Con 75 bit existen $5{,}2*10^{72}$ números primos.

  Con 512 bit existen $3{,}1*10^{151}$ números primos.

  Se estima que existen **$8{,}3*10^{77}$ electrones en el Universo**.

- EJEMPLO: Se desarrolla a continuación un ejemplo empleando esta función a través de un valor muy conocido y didáctico de **g =3 y m =17**, por lo tanto la teoría tratada anteriormente se puede explicar a través de pequeños números primos de la siguiente forma:

f(x) = resto (3^x/17)

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
f(x)	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

Esta función con sus primeros valores presenta la didáctica característica de permitir asociar todo el conjunto de 16 valores a 16 restos con propiedad biyectiva.

Como primer detalle es claro que dado un valor de x es muy fácil descubrir f(x), pero dado un valor de f(x), para poder realizar la operación inversa, se debe realizar el cálculo de las 16 posibilidades. Si se tiene en cuenta lo expresado en la teoría sobre una clave de 512 bit empleando números primos, se deberían realizar $3{,}1*10^{151}$ operaciones matemáticas, aunque existen algunas técnicas para reducir este espacio, en el mejor de los casos puede llegar a ser la mitad, con lo cual seguirá **siendo computacionalmente imposible**.

Volviendo al ejemplo:

- A elige dos valores de clave pública **m = 17** y otro generador **g = 3**.

A elige también una **clave secreta x = 4** < m, luego calcula **X** = resto (3^4/17) = **13** y le envía a B los siguientes tres valores:

m = 17, g = 3, X = 13.

- B recibe estos valores y elige una **clave secreta y = 7** < m,

  calcula **Y** = resto (3^7/17) = **11**, y le transmite el valor de **Y = 11** a A.

- A calcula un nuevo valor **S**= resto (Y^x/m) = resto (11^4/17) = **4**.

14.641	**17**
.....	**861**
R = 4	

- B calcula un nuevo valor **S** = resto (X^y/m) = resto (13^7/17) = **4**.

62.748.517	**17**
.....	**3.691.089**
R = 4	

- Si existiera un tercero que escucha este diálogo, obtendría los valores que fueron transmitidos por el canal de comunicaciones, es decir: m = 17, g = 3, X = 13 e Y = 11. Si tratara de averiguar el secreto compartido, los cálculos que podría realizar son:

  Elegir un número secreto z < m, por ejemplo z = 2.

  Obtener Z = resto (g^z/m), por lo tanto Z = resto (3^2/17) = 9.

  Con este valor intentaría resolver S. Sólo puede hacer dos operaciones:

  Sz = resto (Y^z/m) = resto (11^2/17) = **2**.

121	**17**
- 119	**7**
R=2	

  Sz = resto (X^z/m) = resto (13^2/17) = **16**.

169	**17**
- 153	**9**
R=16	

Como puede apreciarse, mediante estos cálculos no puede obtener el secreto compartido.

- La característica que se demuestra matemáticamente que presenta S es que:

S=resto(X^y/m)=resto(Y^x/m)=resto[$(g^x)^y$/m]=resto[$g^{(xy)}$/m]=resto[$(g^y)^x$/m] = resto[$g^{(yx)}$/m]

S=resto(13^7/17)=resto(11^4/17)=resto[$(3^4)^7$/17]=resto[$3^{(4*7)}$/17]

= resto[$(3^7)^4$/17]=resto [$3^{(7*4)}$/17] = **4.**

Para cerrar este ejemplo, téngase en cuenta la siguiente hipótesis:

Para calcular x con los datos que circularon por el canal de comunicaciones, es decir g, m, X e Y; se debería resolver la siguiente ecuación:

X = resto (g^x/m), que en el ejemplo es 13 = resto (3^4/17)

81	**17**
13	**4**

g^x	**M**
X	**Result**

Por lo tanto: **Result * m + X = g^x** , es decir 4 * 17 + 13 = 81.

Para obtener x desde esta función se despejaría: $x = \log_g(\text{Result} * m + X)$

Si se tiene en cuenta que con 75 bit existen $5{,}2*10^{72}$ números primos, y se eligiera este tamaño de claves, teniendo en cuenta valores extremos como podrían ser:

- Procesadores de 100 GHz.
- Capacidad de poder realizar procesamiento en paralelo de manera tal que a través de n ordenadores, se pudiera realizar el cálculo de x para cada variable en un solo ciclo de reloj, es decir:

En un segundo se obtendrían 10^{11} valores de x.

Si basado en cálculo de probabilidades se tiene en cuenta que la probabilidad de acierto sería a la mitad del espacio de valores, entonces:

$(5{,}2*10^{72})/2 = 2{,}6*10^{72}$

por lo tanto si: 1 seg → 10^{11} valores de x

para $2{,}6*10^{72}$ = **$2{,}6*10^{61}$ segundos**

$2{,}6*10^{61}$ segundos es el tiempo que demoraría calcular la cantidad de valores para acertar el valor que corresponda a **x secreto**.

Como última reflexión se puede calcular lo siguiente:

$2{,}6*10^{61}$ segundos

$= 4{,}333*10^{59}$ minutos

$= 7{,}222*10^{57}$ horas

$= 3{,}009*10^{56}$ días

= **$8{,}244*10^{53}$ años**.

Quedando claro que es computacionalmente imposible de realizar.

2) RSA

Este criptosistema lleva el nombre de sus inventores R. Rivest, A. Shamir y L. Adleman. Permite ser empleado para compartir un secreto y para firma digital. Su seguridad está basada en el problema de factorización de enteros.

Cada entidad crea un par de claves (Pública y Privada) de la siguiente forma:

- Genera dos números primos largos y distintos **p** y **q**.
- Se calcula **n** = p*q, y **ø** = (p –1) * (q – 1).
- Elige un entero (random) **e**, tal que 1<e< ø, y que el máximo común divisor (mcd) de e y ø sea 1, es decir: mcd (e, ø) = 1.
- Se calcula a través del algoritmo Euclídeo extendido un valor entero **d**, 1<d< ø, y tal que el Resto (e*d / ø) = 1.
- Se obtiene de esta forma la **clave pública** (n,e) y la **clave privada** (d).

En la terminología RSA, los enteros **e** y **d** son llamados exponentes de cifrado y descifrado, y **n** es llamado módulo.

El mecanismo completo funcionaría de la siguiente manera:

Si A le enviara un mensaje cifrado a B:

a. Debería obtener la cave pública de B, es decir ($\mathbf{n_b}$, $\mathbf{e_b}$).
b. Representar el mensaje **m** como un entero en el intervalo {0, n-1}.
c. Procesar c = Resto (m^{eb} /n_b).
d. Enviar el mensaje cifrado **c** a B.
e. Al llegar a B, éste debería usar su clave privada $\mathbf{d_b}$, a través de la siguiente fórmula: **m = Resto (c^d / n)**

EJEMPLO:

Generación de claves por parte del equipo A:

a. Se propone dos números primos pequeños: **p** = 5 y **q** = 11 y se calcula n = **p*q** = 55. De estos valores se puede calcular también **ø = (p – 1) * (q – 1)** = 40.
b. A también genera el valor **e = 3**, tal que **1<e< ø** y que el **mcd (e, ø) = 1**. Como puede verificarse 1<3<40 y mcd (3,24) = 1.
c. Se emplea el algoritmo Euclídeo extendido para calcular **d** = 27, teniendo en cuenta que 1<d< ø, (1<27<49); y que el **Resto (e*d / ø) = 1**; Resto (3*27 / 40) = 1.
d. Con estos parámetros entonces la clave pública de A es (**n** = 55 y **e** = 3) y la clave privada de A es d = 27.

Si un equipo B deseara enviarle un mensaje al equipo A, debería primero obtener la clave pública de A, es decir (**n** = 55 y **e** = 3), para luego poder cifrar el mensaje de la siguiente forma:

a. Se toma como ejemplo un valor m de mensaje que se representa por un número entero en el intervalo {0, n-1}, **m** = 12.
b. Calcula **c** = Resto (m^e/n) = Resto (12^3 / 55) = 23.
c. Envía el valor 23 al equipo A.

El equipo A para descifrarlo emplearía su clave privada (**d**) procesándolo de la siguiente forma:

$$\mathbf{m = Resto\ (c^d / n)} = 23^{27} / 55) = 12.$$

8.1.8. Métodos de verificación de integridad (HMAC – SHA – MD5)

1) **HMAC (Hashing for Message Authentication Codes) [RFC-2104]**

Para proveer un modo de chequear integridad de la información transmitida o almacenada en un medio no confiable es necesario un mecanismo que permita compararla contra algo que se considere válido. Para esto se estandarizó un procedimiento basado en una clave secreta usualmente llamado **Código de Autenticación de Mensajes** (**MAC**). El empleo típico de estos mecanismos es a través de dos partes que comparten una clave secreta para validar la información transmitida entre ellas. HMAC propone el empleo de criptografía aplicada a funciones Hash (resúmenes). En la RFC, estandariza el empleo de HMAC con las funciones Hash definidas como **MD5** (Message Digest versión 5) [RFC-1321] y **SHA-1** (Standard Hash Algorithm Versión 1) [FIPS 180-1]. También hace mención al algoritmo propuesto por RIPE denominado RIPEMD-128/160. Hoy ya ha salido la nueva versión SHA-256, de 256 bits y también de 512.

HMAC requiere una función Hash (H) que se encargará de comprimir un texto de longitud finita por medio de iteraciones de una función de compresión básica sobre los bloques de datos (B = 64 Byte), y una clave secreta (K); y por medio de ambas se obtendrá un resumen de longitud fija (L), que será de 16 Byte para MD5 y 20 Byte para SHA-1.

Esta función Hash es llamada "**One Way**" pues no es posible a través del resumen de salida obtener el texto de entrada, también resultará computacionalmente imposible obtener un valor de salida igual a través de otro valor de entrada, como así tampoco desde un valor de salida ya calculado, obtener otro valor de entrada diferente al verdadero.

Se definen dos cadenas de longitud fija diferentes una de la otra llamadas:

- Ipad = repetición del Byte 36 B veces.

- Opad = repetición del Byte 5C B veces.

Luego se ejecuta: H { K_B xor Opad, H (K_B xor Ipad, texto)}

Para esta tarea se debe tener en cuenta:

a. Rellenar con ceros la clave K hasta obtener una longitud de 64 Byte (llamada K_B).

b. Realizar: K_B *xor* Ipad , (Result1).

c. Anexar el texto completo al resultado de b, (Result2).

d. Aplicar la función Hash (H) al resultado de c, (Result3).

e. Realizar: K_B *xor* Opad, (Result4).

f. Anexar el resultado de d, (Result3). Al resultado de e, (Result4).

g. Aplicar la función Hash (H) al resultado de f. Obteniendo el resultado, que acorde a la función Hash empleada será de 16 o 20 Byte.

2) The MD5 Message-Digest Algorithm (RFC- 1321)

Este algoritmo toma un mensaje de entrada de longitud arbitraria y entrega una salida de 128 bit de longitud fija. Llamado "Huella digital" o "Recopilación" de mensaje (Message Digest). Es computacionalmente imposible producir dos mensajes que posean la misma recopilación, como tampoco regenerar el mensaje a través de la recopilación. Este algoritmo puede ser empleado para aplicaciones de firma digital, donde un texto debe ser comprimido de manera segura antes de ser cifrado con sistemas de clave pública.

El algoritmo MD5:

Se implementa por medio de 5 pasos:

Paso 1: Anexa bit de relleno.

Se anexan bit de relleno para que el mensaje dividido 512 tenga resto 448, es decir 448 (mod 512). El primer bit de relleno será un uno y luego se continuará con una cadena de ceros.

Paso 2: Anexa longitud.

Una representación de 64 bit del mensaje original (antes del paso 1) es anexada al resultado. En este paso el mensaje resultante es un múltiplo exacto de 512 bits, es decir que es múltiplo exacto de 16 palabras de longitud 32 bit.

Paso 3: Inicialización del buffer MD.

Un buffer de cuatro palabras (A, B, C, D) es empleado para procesar el mensaje generado luego del paso 2. Cada uno de estos buffer es un registro de 32 bit. Esos registros son inicializados con los siguientes valores en hexadecimal:

A:01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 54 32 10

Paso 4: Procesar el mensaje en bloques de 16 bit.

Se definen cuatro funciones auxiliares donde cada una tiene una entrada de tres palabras de 32 bit y produce una palabra de salida de 32 bit. Las funciones son:

F(X,Y,Z) = X.Y OR not(X).Z

G(X,Y,Z) = X.Z OR Y.not(Z)

F(X,Y,Z) = X XOR Y XOR Z

I(X,Y,Z) = Y XOR (X OR not(Z))

Como ejemplo de la función F, su tabla es:

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Se procesa cada bloque de 32 bit bajo una serie de operaciones tabuladas en esta RFC-1321 y queda como resultado un valor de 32 bit en cada uno de los buffer A, B, C y D.

Se repiten los pasos a través de asignaciones de variables temporales (AA, BB, CC, DD) hasta el último bloque de texto, repitiendo sucesivamente la asignación que se detalla a continuación.

A = A + AA

B = B + BB

C = C + CC

D = D + DD

Al finalizar todos los ciclos (recordar que el texto de entrada es múltiplo exacto de 16 palabras de longitud 32 bit) quedan en los buffer A, B, C y D el resultado final del proceso.

Paso 5: Salida.

El mensaje generado es una salida de los Buffer A, B, C y D, los cuales se anexan desde el bit de menor orden de A hasta el último de D, por lo tanto esta salida es un mensaje de 128 bit de longitud.

8.1.9. Firma digital

Ya hemos tratado toda la lógica, mecanismos y funcionamiento de la criptografía con clave simétrica. También el empleo de resúmenes o "hash", ahora avanzaremos sobre su empleo para "firma digital".

La idea de firma digital, está asociada al concepto de "**no repudio**", es decir cómo poder vincular cualquier tipo de documento o transacción para que quede "**vinculado**" a la persona responsable del mismo. Si comparamos esta metodología digital con el mundo convencional, se trata de poder recibir o procesar un archivo "firmado" como si fuera de puño y letra.

Bajo los conceptos que hemos desarrollado a través del "par de claves" (pública y privada), hicimos hincapié en que las mismas se generan en un mismo paso, que guardan relación entre ambas, y que si aplico una para cualquier proceso, debo aplicar inexorablemente la otra para revertirlo. Este par de claves una vez entregado a su responsable, se difunde únicamente la clave pública a través de un mecanismo que evite la posibilidad de falsificarla (para evitar el hombre del medio), pero la clave privada la única persona que debe conocerla y que la posee es su propietario (y nadie más), por lo tanto ¿Quién es la única persona que puede aplicar esa clave?, la respuesta es obvia: ÚNICAMENTE SU PROPIETARIO (y nadie más). Por lo tanto el planteamiento que da origen a la firma digital es: ¿Qué sucede si a un archivo, se le genera su resumen (o hash) y luego se le aplica una clave privada?...Evidentemente se obtendría un segundo resumen. Pues si a ese mismo resumen inicial se le aplica la clave pública correspondiente a esa clave privada, debería obtenerse el mismo hash o resumen final, pero con cualquier otra clave pública el hash sería diferente pues sólo se corresponde UNÍVOCAMENTE con la clave privada que está enlazada con esta pública.

Concepto de Firma: resumen del texto cifrado con la parte secreta de la clave.

Verificación:

- ¿Es igual el texto a lo que sale de descifrar con la parte pública?
- Garantía matemática de integridad
- Asociación matemática al conocimiento del secreto (por el concepto desarrollado que el "par de claves" mantienen un algoritmo matemático que se demuestra, las vincula)

Identidad del signatario: alguien fiable debe ratificar la pertenencia de la clave pública

Requisitos de una firma digital:

a. Debe ser fácil de generar.

b. Será irrevocable, no rechazable por su propietario.

c. Será única, sólo posible de generar por su propietario.

d. Será fácil de autenticar o reconocer por su propietario y los usuarios receptores.

e. Debe depender del mensaje y del autor (Esta última propiedad es muy importante pues protege ante la falsificación de los mensajes).

Son condiciones mucho más fuertes que las de una firma manuscrita.....

Estándares de firma digital

Los más importantes a tener en cuenta son:

1991: National Institute of Standards and Technology (NIST) propone el DSA, Digital Signature Algorithm, una variante de los algoritmos de ElGamal y Schnoor.

1994: Se establece como estándar el DSA y se conoce como DSS, Digital Signature Standard.

1996: La administración de los Estados Unidos permite la exportación de Clipper 3.11 en donde viene inmerso el DSS, que usa una función hash de tipo SHS, Secure Hash Standard.

No repudio: Es crear una "relación vinculante" entre las partes involucradas, es decir que quien firme un archivo electrónicamente (con su clave privada), no pueda luego aducir que no fue él quien lo hizo. Teniendo en cuenta el concepto de "Confianza" para la distribución y verificación de claves y la relación "Unívoca" entre clave pública y privada, resulta computacionalmente imposible que al aplicar la clave pública del firmante del documento, una vez verificado el proceso, no haya sido éste quien aplicó la clave privada para firmar el mismo, por lo tanto sólo responderá el proceso a la correspondencia con una única clave privada, lo que demuestra y vincula el origen del mismo.

Para representarlo por pasos:

PROCEDER CORRECTO DE FIRMA DIGITAL

A desea enviar un documento firmado a **B**.

A busca el archivo "**ArchX**" a enviar y genera una "hash" aplicando su "clave privada", la cual será $\mathbf{K_{A(Priv)}}$

Quedan dos archivos: "**ArchX**" y "**Hash_1_ArchX**"(generado con $\mathbf{K_{A(Priv)}}$).

Supongamos que **A** le envía a **B** AMBOS archivos.

B, recibe ambos archivos, busca la "clave pública" de **B** y a "**ArchX**" le aplica la $\mathbf{K_{B(Pub)}}$

B obtiene un nuevo "Hash" aplicando su "clave pública" de **A**:

"**Hash_2_ArchX**"(generado con $\mathbf{K_{A(Pub)}}$).

Si: **Hash_1_ArchX = Hash_2_ArchX** → NO HAY DUDA QUE LO FIRMÓ "**A**", pues es la única persona que posee esa clave privada para generar el mismo "Hash".

8.1.10. Sellado de tiempos.

La técnica de sellado de tiempos o "**time stamp**", es otra da las grandes aplicaciones que pueden ser llevadas a cabo mediante la combinación de clave asimétrica y resúmenes. La analogía con el mundo convencional es la misma que cuando se entrega un documento por mesa de entradas o se envía un correo postal de forma "certificada", casos en los cuales nos queda una constancia legal o sello de que esa información la recibió "alguien" en esa fecha.

Para tratar este tema de forma eminentemente práctica, nos basaremos en dos desarrollos que están a nuestro alcance, son públicos y sin fines de lucro:

- Criptolab de la UPM: https://add.mec.es
- Musicalibre: www.musicalibre.es

El "Criptolab" o laboratorio criptológico de la Universidad Politécnica de Madrid, ofrece un servicio de Sellado de tiempos gratuito desde el año 2004. Está basado en el "encolado" de resúmenes "**Hash 256**" avalados con su propio certificado digital, como lo desarrollaremos a lo largo de este punto.

www.musicalibre.es es un portal Web, que ofrece también desde el año 2004, un servicio sin fines de lucro a todo artista que desee publicar sus obras con licencia "Copyleft". Este portal ha desarrollado un mecanismo gratuito de registro de temas denominado "Licencias de Música Libre" (LML), que a través del empleo de criptografía y sellado de tiempos, permite al autor "subir" su obra, dejando constancia de "**Autoría – Integridad y Sellado de tiempo**", cosa que para un autor es esencial. Cabe mencionar que con anterioridad a Musicalibre, esta garantía era complicada para autores que no tienen recursos (económicos y/o técnicos) para apoyar su arte en el Registro de la Propiedad Intelectual, por ejemplo, en muchos casos se "auto enviaban" un correo postal con aviso de entrega a su propio domicilio, y en el sobre ponían un CD (o cinta) con su tema, este sobre jamás era abierto, y en caso de un litigio, podían comprobar que ese sobre había sido enviado tal día, y que aún estaba cerrado, por lo tanto si un Juez ordenaba abrirlo, se encontraría con una canción grabada en esa fecha sobre la cual apela sus derechos de autor esa persona... como podéis ver, este tema de la propiedad intelectual da para mucho, y el "sellado de tiempos" cobra un sentido real y concreto en nuestra vida. Por supuesto que el sellado de tiempo de "algo" (cualquier cosa), no tiene ningún sentido, si "esa cosa" no está íntegra. Volviendo a nuestra analogía con el mundo del papel, de nada nos sirve tener sellado un documento por mesa de entrada, si el mismo tiene tachaduras, enmiendas, agregados, no se sabe cuántas hojas son, está ilegible, etc... Es decir, el factor clave en todo esto es su "**Integridad**": lo que está sellado, debe ser "irrefutable", sino no tendría sentido.

Vamos a ir avanzando de forma práctica.

Criptolab:

El servidor de sellado de tiempos del critpolab, como ya mencionamos, funciona en base a resúmenes "Hash-256". Cuando te conectas al mismo, primero te presentará la aceptación o no de su certificado para entrar en "https" (como se trató en ese capítulo), una vez que aceptas el mismo, ya entras en la sesión segura y verás la siguiente interfaz gráfica:

Como puedes ver posee una breve explicación de su servicio y luego en la parte inferior nos ofrece una serie de alternativas, comenzaremos analizando sus "**Ficheros Históricos**". Si pinchas aquí se te despliega la siguiente ventana:

En la imagen anterior de nuestro ejemplo hemos seleccionado el día "**02/02/2010**" y que se envíe a la cuenta de correo acorletti@hotmail.com, es decir que nos enviará todos los registros de tiempo de esa fecha. En nuestro caso, si bien no pegamos las otras dos imágenes pues no tiene sentido, también hemos pedido el del mes "**02/2010**" y el del año "**2010**" (solamente repitiendo esta solicitud, dejando en blanco esos campos).

Para comenzar a estudiar el tema, pegamos abajo las tres solicitudes:

Registro diario 2010/02/02

2010/02/02 00:05:02,020181 UTC
0000000000000000000000000000000000000000000000000000000000000000
2010/02/02 00:05:02,020278 UTC
79522C4763AA35783BE9E5953686EB7EA9E2515BC5CF0E3CF2A1D439C7C4F414
2010/02/02 01:05:01,310722 UTC
E27D8AE1B5AF0B6FBECADDDE7F3D578B2CA28A3FC9F65DEE924AF21FEB01A9DE
2010/02/02 15:11:55,071243 UTC
609134A19FEB1B78C3E8512A81C4D2A3F34C8C18AD58D83F40B7E4EA214DF98B
2010/02/02 15:22:23,393439 UTC
982923732BC8775A71CB292B9EE292EB4B0DFB6498B7C6AC5120DD23BD4105FD

2010/02/02 15:32:53,074430 UTC
8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
2010/02/02 15:50:19,161434 UTC
3EF3E4B730C39494E86C1A43E72E77C28F821E428C776515BB2BEE1CB8B40146
2010/02/02 21:05:02,038257 UTC
B58705BF7C85ABA86D914890EF10CABB1AF000A5FC8C0C3AE837DEA9F15D7B31
2010/02/02 22:05:01,225798 UTC
AAC553223833C9D3A8DC2BFA38ACBF3D07B4CE0D051249A44BA6BE1D4FDBA24A
2010/02/02 23:05:02,010294 UTC
B529FE51BDA674CA04E130A6903CCF9516353DBD5BB88D677985740965C140D9

Registro diario cerrado a 2010/02/03 00:05:01,313984 UTC

Registro mensual 2010/02

2010/02/01 05DC732C170949AA20726F43EACF5BF933F5921A372FF488B3BD26538EB930A5
2010/02/02 3A233C5771B5C63F6755CA4D84D153790C63B1F92AF0F471595D899A84EEB695
2010/02/03 8D21D8C78DB2F65585B5A9CC7B71B669B508778D1DBFCCABA5344651DE7FD8C1
2010/02/04 49DBB2C723D27A259AE21F869EEFBE03F3441F86F380C3A44EA584E2F2018B52
2010/02/05 660D4B9D49E1B643F1CD054A3020922955375769992F5871909FABFB61FD14DD
2010/02/06 B980623C33A200B7E9FD6376634BCEAA0DB0D99446B2906A6E1108F542D82B3D
2010/02/07 7E412323CF42DDA9FD83216BD30DB0064549E0F93DCA343E76881024E318ACCD
2010/02/08 2224C874009D59A9A76840216931B3F5FE088AA57D5F1CBD5A8F5FD8144D1A6A
2010/02/09 57549A3D60D60701F131017CD54B1D9EC452FDE79C3B46E18EC8DC0A45875A1F
2010/02/10 677F7C4A5D56D1667859FAD3EE10929EB39812E268E78DD6CA8505D5A51F4731
2010/02/11 24B695E6DBDCE7E464A9567E6794025EE6D60403A7474644B3000E6A36E5A519
2010/02/12 72560A7B4F0D0ECBDBA68360646CF92C13F24687778EFC3E63FD19397A457436
2010/02/13 F284696A2FBABD5BE42205E6CDCAB2D0A444DA58FD917B9559401C0C79CE26D5
2010/02/14 EC81D386C8789187C8217D0199A1FC23A0910586CE81990CF91453943B9EF77E
2010/02/15 02EFCBD6EA587719910C44BF4B09067A476B9C6388D74329104A8FD63AF97922
2010/02/16 4D362846AF29EC46F0A6B5C809B801F4C3B5866C165C0396343FAB7525AE57EE
2010/02/17 779CDE8A377F1D4F4780DFFC1359EC95F0A1EA52BD838275180AC70FBD3D910E
2010/02/18 75E8376D56CF40FFF9F68A06B109F26429ADE3AAFB297991D96007AA179ACCF9
2010/02/19 C29A53CCA1C1B3F7CCDD25DDBD394775D867674E5DC407A4698423940657333B
2010/02/20 268523621D2250917AC2267E6AB827B418F3C5D597EB219D423209A95ABB141A
2010/02/21 3E4C1FD905C478092C97EAA5C486013D17E7B04207B8B4B47188BA03519944C3
2010/02/22 5AF20D774724DFC93E9B6D58E54D89609962F40FD34B78A71B134AE7C002CD8F
2010/02/23 996D18F4A0A53B10CF88DEA69F8836B1CE5776E47C97499F851250FACF20710F
2010/02/24 091CB851DA36629E0B5AD9719D83A2E4E2E9515D0871385A132226687777B8AB
2010/02/25 D48C83CF1A175BF50596E09813BC9544D1B4ACD08BE93E9C3274E670787E3C18
2010/02/26 966CC6CD03D3EE92BF04CCA314687BD6D08FC9C601E6551B2384E3ACDC6DD12D
2010/02/27 2FFA26317C4F883341EF63FE0EE1A00F1FE764C3BDD60171C2104790FDDDE57A
2010/02/28 D61ED008B1A135B02FD3753AECBB881A9E14DFE490266671AB8F98568E48BC41

Registro mensual cerrado a 2010/03/01 00:05:01,096550 UTC

Registro anual 2010

2010/01 54CD48C0AAF80EE1FA04EEE959AE76DE84AB6E4E6FDACBBFE8D6D254BEE06D65
2010/02 A1099F3D6D06A8C863D80169E16229C0BAD11D73572AF9115A8F9A2389F138D3
2010/03 2C5BA0A4A1FD4BF75E1400923059D572ACEF4B4E3547CCD1D7E484336F43F424
2010/04 3AEB0E878098B4FD7B2A3D5B155870E992B10A7E009767AB6291C9695032B5F0
2010/05 E76ED3A8C43FA09EBAA4E1511FF67B2A7C3FA437E2F43D510D762A8298D787A5
2010/06 0BB156E86ADB5FB4F9BD8B71EC0AC4D5E90458A5C341FAD3BC1C2784F132F244
2010/07 8DD5E63182C45C8251540211E5479B020AB6B5352D2A350C6AAC9440B6DF2B04
2010/08 D121BF392B1BEC4DE60E51D7D3D3610EF8E66102B3444A26F227D4C5CCE9D543
2010/09 8092ECF94C5595B509E3365D21A58A0852D6192562EB24A33E1102DF3A573BAC
2010/10 A3C045077C93EB0DB74C6C3BBA7DD7D8DAAA0D1330B4B2011E6BCCBC6FB16B66
2010/11 6DE17D86A62A73484365B7F714929785889DD436C420EF934AC23BC2795ACC7D

2010/12 D9F1B9DE40E1EE0F4313F73F138C2F537967B9B22B94823F264741859B73F87E

Registro anual cerrado a 2011/01/01 00:05:02,050778 UTC

Como podéis ver en el primero de los registros "Registro Diario", cada día se inicia con todos ceros:

2010/02/02 00:05:02,020181 UTC
0000000000000000000000000000000000000000000000000000000000000000

A medida que ingresan las solicitudes, se van ordenando secuencialmente con todas las demás de ese día (ya veremos cómo). Por ahora para comprender el tema de los históricos, sigamos adelante hasta finalizar ese día:

"Registro diario cerrado a 2010/02/03 00:05:01,313984 UTC"

Unos minutos comenzado el nuevo día (para evitar ambigüedades de 00:00:00) se cierra el "Registro diario", ¿qué implica ello? En ese preciso instante, se genera un nuevo HASH que es la concatenación de todos los sellados del día, firmado con la clave privada del "Servidor de tiempos", lo que genera un nuevo resumen que será el histórico de ese día. En nuestro caso se corresponde con:

2010/02/02 3A233C5771B5C63F6755CA4D84D153790C63B1F92AF0F471595D899A84EEB695

Este mismo proceso se repite al finalizar cada mes, generando cada uno de los doce registros mensuales que puedes ver también en la histórico de "R**egistro mensual**" que pegamos arriba, con los cuales unos minutos después de finalizar el año se repite el proceso obteniendo el Registro "histórico Anual" que en nuestro caso fue:

Registro anual cerrado a 2011/01/01 00:05:02,050778 UTC

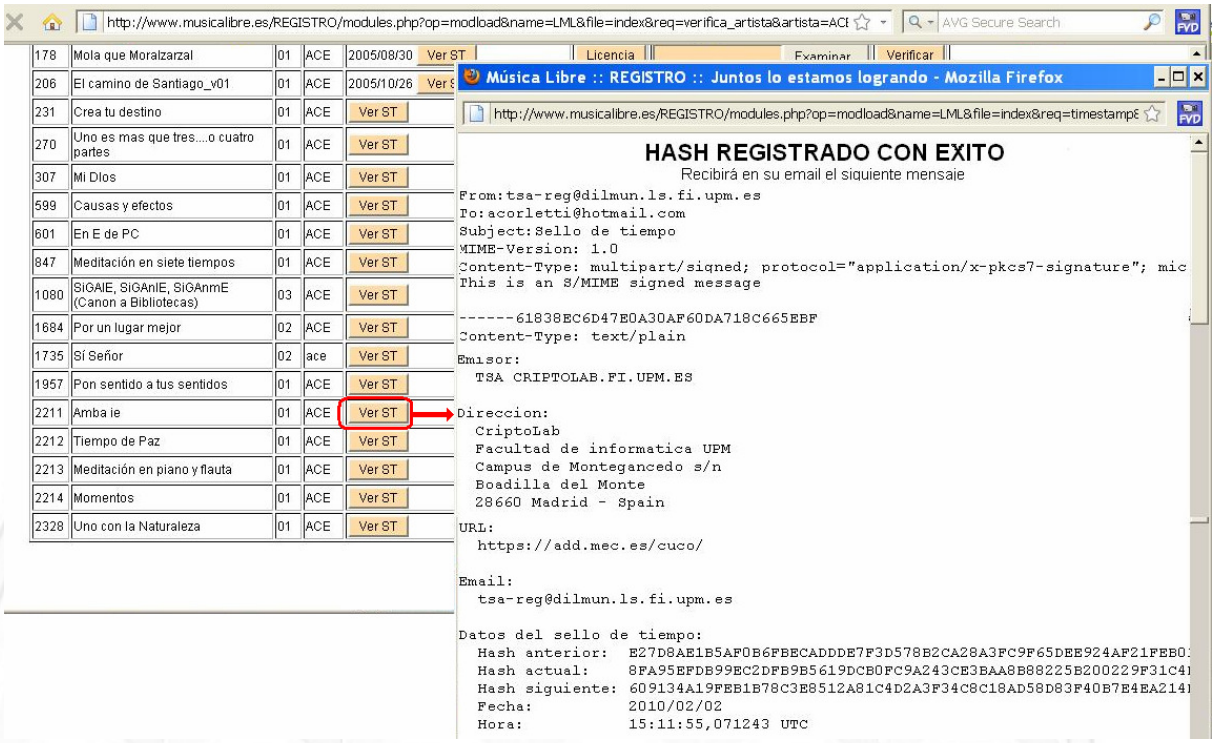
Todos los "Hash 256" que has visto, se presentan en formato "hexadecimal", el cual te recordamos que puede representar con un dígito a cuatro "bits", por lo tanto la longitud de estos registros es de 64 dígitos hexadecimales, que equivalen a 256 bits (64x4).

Tal vez te puedas preguntar ¿Por qué tantos registros?

El aspecto más importante de una entidad de sellado de tiempos, aparte de su reloj y firma digital, es garantizar su "**imparcialidad**" respecto a los resúmenes. Lo que tiene que ofrecer es un sistema que no permita alterar nada de lo que históricamente procesó.

Analicemos ahora su funcionamiento, pues básicamente para poder garantizar esta "No alteración", **cada sellado de tiempo depende directamente del anterior**. Para verificar este hecho, es que vamos a trabajar con datos reales generados por la Web www.musicalibre.es.

En esta página Web, desde la opción "REGISTRO → Validar una canción", puedes acceder justamente a estos resúmenes Hash que tienen también su sellado de tiempo.



En nuestra imagen anterior, hemos seleccionado uno de sus usuarios (conocido...) "ACE" y del mismo, como puedes ver "haciendo clic" en el botón "**Ver ST**" (por Sellado de Tiempo) se despliega una ventana con el sellado que emitió en su momento el "Criptolab", en el mismo puedes ver todos los campos que posee este sellado.

Para nuestro estudio nos vamos a centrar en los valores que figuran debajo de la ventana "Hash anterior, Hash actual, Hash siguiente", los cuales como puedes apreciar en la parte inferior de la imagen, se emitieron el "**2010/02/02**" a las "**15:11:55,071243 UTC**". Si prestas atención nuevamente a la ventana de Musicalibre, verás que este "Sellado de Tiempo" se corresponde al registro de la canción "*Amba ie*" con el número **2211**, y que los siguientes temas registrados por este autor son el **2212**, **2213**, y **2214**. Como conocemos a su autor, nos ha autorizado a presentarlos en este texto, por lo tanto los descargamos y te presentamos a continuación el primero de ellos completo (el de la imagen anterior), luego sólo los Hash anterior, actual y siguiente de todos ellos:

HASH REGISTRADO CON EXITO

(Recibirá en su email el siguiente mensaje))
From:tsa-reg@dilmun.ls.fi.upm.es
To:acorletti@hotmail.com
Subject:Sello de tiempo
MIME-Version: 1.0
Content-Type: multipart/signed; protocol=”application/x-pkcs7-signature”; micalg=sha1; boundary=”----61838EC6D47E0A30AF60DA718C665EBF”
This is an S/MIME signed message
------61838EC6D47E0A30AF60DA718C665EBF
Content-Type: text/plain

Emisor:

TSA CRIPTOLAB.FI.UPM.ES
Direccion: CriptoLab
Facultad de informatica UPM
Campus de Montegancedo s/n
Boadilla del Monte
28660 Madrid - Spain

URL: https://add.mec.es/cuco/
Email: tsa-reg@dilmun.ls.fi.upm.es

Datos del sello de tiempo:
Hash anterior: E27D8AE1B5AF0B6FBECADDDE7F3D578B2CA28A3FC9F65DEE924AF21FEB01A9DE
Hash actual: 8FA95EFDB99EC2DFB9B5619DCB0FC9A243CE3BAA8B88225B200229F31C4B812C
Hash siguiente: 609134A19FEB1B78C3E8512A81C4D2A3F34C8C18AD58D83F40B7E4EA214DF98B
Fecha: 2010/02/02
Hora: 15:11:55,071243 UTC

------61838EC6D47E0A30AF60DA718C665EBF
Content-Type: application/x-pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"

MIIG3gYJKoZIhvcNAQcCoIIGzzCCBssCAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3
DQEHAaCCA/YwggPyMIIC2qADAgECAgEBMA0GCSqGSIb3DQEBBQUAMIHOMRIwEAYD
.....[cortada la firma del servidor]
NrlVxn5cBERMppFzcASZEM7kTWheWQlnE5tKNZf1aA7FRA==
------61838EC6D47E0A30AF60DA718C665EBF—

El extracto de los cuatro "Hash" es el siguiente:

Datos del primer sello de tiempo *(el que figura al completo en el texto anterior)*:
Hash anterior:
E27D8AE1B5AF0B6FBECADDDE7F3D578B2CA28A3FC9F65DEE924AF21FEB01A9DE
Hash actual:
8FA95EFDB99EC2DFB9B5619DCB0FC9A243CE3BAA8B88225B200229F31C4B812C
Hash siguiente:
609134A19FEB1B78C3E8512A81C4D2A3F34C8C18AD58D83F40B7E4EA214DF98B
Fecha: 2010/02/02
Hora: **15:11:55**,071243 UTC

Datos del segundo sello de tiempo:
Hash anterior:
609134A19FEB1B78C3E8512A81C4D2A3F34C8C18AD58D83F40B7E4EA214DF98B
Hash actual:
010D7423552EE96747349B2BD515A1CA091CB6C25D5C6CAE192428158802556F
Hash siguiente:
982923732BC8775A71CB292B9EE292EB4B0DFB6498B7C6AC5120DD23BD4105FD
Fecha: 2010/02/02
Hora: **15:22:23,393439 UTC**

Datos del tercer sello de tiempo:
Hash anterior:
982923732BC8775A71CB292B9EE292EB4B0DFB6498B7C6AC5120DD23BD4105FD
Hash actual:
7537B24F82C064D1F229983B5A34FE23448DB11D7A614E5456F0B6C4696D36D3

Hash siguiente:
8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
Fecha: 2010/02/02
Hora: **15:32:53**,074430 UTC

Datos del cuarto sello de tiempo:
Hash anterior:
8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
Hash actual:
56B0BCB528771396C8434982C48FCD507695A11BECE4F2320ACC38E1789F45BA
Hash siguiente:
3EF3E4B730C39494E86C1A43E72E77C28F821E428C776515BB2BEE1CB8B40146
Fecha: 2010/02/02
Hora: **15:50:19**,161434 UTC

Vamos a analizar con detalle como funciona este "servidor de tiempos" sobre la base de los cuatro sellados anteriores (que tienen la característica de ser consecutivos en su registro y sellado).

Como puedes apreciar, el Hash siguiente del primer sello, es el mismo que el Hash anterior del segundo sello. El Hash siguiente del segundo sello es el mismo que Hash anterior del tercero y por último, el Hash siguiente del tercero es el mismo que el Hash anterior del cuarto.

El primer concepto entonces es que: **cada sello se inicia con el valor resultante del anterior**.

¿Pero cómo se obtiene este valor resultante?

Este tema se explica con total claridad en la Web del Criptolab:

Para comprobar que el valor hash registrado realmente está en la secuencia del registro histórico diario, concatenamos los valores de texto del hash anterior y del hash registrado y el documento de texto resultante se le aplica la función hash de referencia (en nuestro caso el sha256). El resultado de esta operación tendrá que coincidir precisamente con el valor hash posterior.

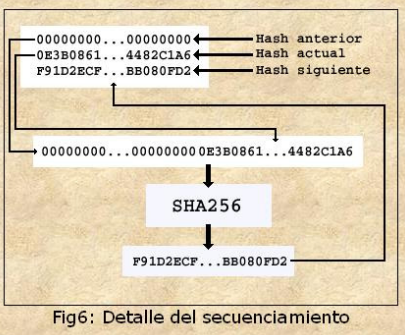


Fig6: Detalle del secuenciamiento

A la vista de dos o más sellos de tiempo válidos, no sólo se pueden ordenar cronológicamente según las horas que constan en ellos, sino que se pueden localizar en el correspondiente registro histórico y conocer, además de su orden, cuantos sellos más se generarón entre cada uno de ellos. Si se disponen de todos los sellos de tiempo que forman parte de un registro histórico, éste puede ser reconstruido completamente aunque la firma digital del registro ya queda suficientemente probada con los diferentes estados que aparecen dentro de él.

Es decir, se concatena el "**Hash anterior**" con el Hash que ha "subido el usuario" y a esa concatenación, se le aplica la función "Hash-256", dando como resultado un nuevo resumen de 256 bits (expresado en 64 dígitos hexadecimales) que es el valor del "**Hash siguiente**".

Esto lo puede comprobar cualquier persona que aplique la función hash. En nuestro análisis vamos a emplear el comando "hash256sum" de Linux.

Vamos a trabajar con este cuarto sellado de los anteriores:

Datos del cuarto sello de tiempo:
Hash anterior: 8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
Hash actual: 56B0BCB528771396C8434982C48FCD507695A11BECE4F2320ACC38E1789F45BA
Hash siguiente: 3EF3E4B730C39494E86C1A43E72E77C28F821E428C776515BB2BEE1CB8B40146
Fecha: 2010/02/02
Hora: **15:50:19**.161434 UTC

La explicación del Criptolab, nos indica que para obtener el "Hash siguiente" debemos concatenar el Hash anterior con el **Hash actual**. Es decir que para trabajar con estos datos reales, debemos concatenar:

8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
Con:
56B0BCB528771396C8434982C48FCD507695A11BECE4F2320ACC38E1789F45BA

Para poder ejecutar el comando "**hash256sum**" desde Linux es necesario contar con el "archivo" al que se le aplicará la función, por lo tanto a esta "concatenación" debemos guardarla en un archivo, esto lo podemos hacer con la siguiente instrucción:

```
# echo -n
"8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E556B0BC
B528771396C8434982C48FCD507695A11BECE4F2320ACC38E1789F45BA" >prueba_hash.txt
```

Con esta orden hemos logrado "concatenar" ambos hash (anterior y actual) y direccionarlos hacia un archivo que quedó creado llamado "prueba_hash.txt". Ahora nos falta aplicarle la función hash 256 a ese archivo. Esto lo hacemos con el siguiente comando:

```
# sha256sum prueba_hash.txt

3ef3e4b730c39494e86c1a43e72e77c28f821e428c776515bb2bee1cb8b40146  prueba_hash.txt
```

Como puedes apreciar, el valor que nos devolvió es EXACTAMENTE el "Hash siguiente: 3EF3E4B730C39494E86C1A43E72E77C28F821E428C776515BB2BEE1CB8B40146" de ese sellado de tiempo (número cuatro de nuestro ejemplo) emitido por el Criptolab.

CONCLUSIÓN PARCIAL: Hemos verificado que el servidor de sellado de tiempos, "concatena" el "Hash último" que posee con el "Hash actual", que un usuario presenta, a esta cadena le aplica la función "Hash 256" y obtiene el "Hash siguiente", que luego será en "Hash anterior para el próximo usuario que solicite un sellado de tiempo. Todo esto podemos verificarlo siguiendo esta secuencia (en colores) de nuestros 4 sellados de tiempo.

Basados en esta conclusión anterior es que podemos inferir en que si cualquiera intentara modificar o alterar uno de esos "Hash", el "Hash siguiente" ya será diferente, y el que sigue, y el que sigue y el siguiente... Es decir, si cualquiera pudiera tener acceso a este servidor y hacer una modificación, debería calcular todos y cada uno de los "Hashes siguientes" y falsificarlos también... esto ya sería complicado, pero lo es aún mucho más porque justamente existen los "Históricos" (y para ello están). Como vimos al principio, al finalizar cada día el "servidor de tiempos", genera un resumen Hash de TODOS los sellos

emitidos ese día Y LO FIRMA, luego lo hace mensualmente con cada uno de esos históricos diarios y finalmente lo hace de forma anual con lo de cada uno de los meses. Por esta razón es que elegimos cuatro sellos emitidos en el 2010, para que puedas ver todos los históricos.

Estos cuatro registros que hemos descargado de la Web www.musicalibre.es, pueden verificarse desde el "**resumen Histórico**" de día **02/02/2010** que hemos descargado del "Criptolab" (de dos sitios diferentes):

Registro diario 2010/02/02 *(del Criptolab)*
2010/02/02 00:05:02,020181 UTC
0000000000000000000000000000000000000000000000000000000000000000
2010/02/02 00:05:02,020278 UTC
79522C4763AA35783BE9E5953686EB7EA9E2515BC5CF0E3CF2A1D439C7C4F414
2010/02/02 01:05:01,310722 UTC
E27D8AE1B5AF0B6FBECADDDE7F3D578B2CA28A3FC9F65DEE924AF21FEB01A9DE
2010/02/02 15:11:55,071243 UTC
609134A19FEB1B78C3E8512A81C4D2A3F34C8C18AD58D83F40B7E4EA214DF98B
2010/02/02 15:22:23,393439 UTC
982923732BC8775A71CB292B9EE292EB4B0DFB6498B7C6AC5120DD23BD4105FD
2010/02/02 15:32:53,074430 UTC
8E57AE9F14A1A4B7265ECE1C5CFB1339BAA2826D4A9C4143D34A0C3AFE52B6E5
2010/02/02 15:50:19,161434 UTC
3EF3E4B730C39494E86C1A43E72E77C28F821E428C776515BB2BEE1CB8B40146
2010/02/02 21:05:02,038257 UTC

Como puedes ver, este "encolado" o concatenado de resúmenes tienen una dependencias absoluta del anterior, en nuestro caso se trata de sellados del año 2010, que sería absolutamente imposible de falsificar.

Cabe mencionar un hecho particular del Critolab. Como puedes apreciar en el histórico anterior, quedan asentados los "**Hash anterior**" y los "**Hash siguiente**", pero en ningún caso el "Criptolab" almacena el "**Hash actual**" que es el que subió el usuario. Este es una decisión que se adoptó en su momento para no almacenar nada de lo que es "propiedad" del usuario. En realidad para este servidor y para cumplir su función es un dato que no le hace falta para nada, pues su deber es "Garantizar el encolado y el tiempo", si se debe hacer uso del "Hash" del usuario, es éste quien deberá presentarlo ante la instancia que lo requiera, y al momento de presentar su "Hash", este (y sólo este) será el que pueda generar el "Hash siguiente" exactamente igual al que figura en el histórico. NO PASÉIS POR ALTO ESTE TEMA, pues es de suma importancia: ese "**Hash actual**" que subió en su momento el usuario, es el que generó UNÍVOCAMENTE el archivo que ese usuario deseaba registrar. En el caso de ser una canción de Músicalibre, lo que se está presentando como "Hash actual" al Criptolab, es un resumen de la canción con todas sus etiquetas "embebidas" e inclusive un secreto que el usuario ha ingresado y que lo vincula, de todos esos datos, se genera el "Hash actual" que presentará el usuario y POR DEFINICIÓN de HASH, ningún otro archivo sería capaz de generar este mismo "Hash actual", por lo tanto, una vez que ese "Hash actual" queda "encolado" en el servidor de tiempos, independientemente de ser una prueba de "Fechas", **también es una prueba de Integridad**, pues ese "Hash actual", es la prueba fehaciente (**o huella digital**) de un archivo que lo generó.

CONCLUSIÓN FINAL: La "Imparcialidad e Integridad" de cada uno de los sellados queda garantizada por estos históricos, pues sería "Computacionalmente imposible" alterar

cualquier tipo de datos que haya sido "encolado" en este servidor, y en el remoto caso que pudiera hacerse, dejaría huellas en cada uno de los usuarios que recibieron su sellado "legal" y luego resulta modificado, estaríamos hablando de un tema verdaderamente complicado.

Hemos querido presentar este tema de forma eminentemente práctica y trabajando sobre datos reales e históricos que puedes verificar en el momento en que lo desees en las páginas Web mencionadas. Estamos seguros que es la forma más didáctica de presentar el tema, si deseas profundizar más en su teoría verás que existen también otras técnicas de sellado y otro tipo de servicios y servidores también gratuitos y otros de pago, pero lo importante aquí es que puedas haber visto de forma práctica cómo "Cierra" todo este tema de la clave asimétrica y los resúmenes Hash.

En la sección "ejercicios", te propondremos diferentes formas de trabajar con este tema.

8.1.11. PGP y GPG

PGP (Pretty Good Privacy: Pivacidad bastante buena) es un programa desarrollado por Phil Zimmerman y presentado en el año 1991 para libre distribución, lo que le ocasiona un serio problema legal por las leyes criptográficas de EEUU. Combina las mejores características del empleo de clave asimétrica y simétrica, permitiendo con ello todo tipo de trabajo con técnicas criptográficas, desde aplicar confidencialidad e integridad a archivos en tránsito o almacenados (comprimiéndolos también), firmar digitalmente, establecer conexiones seguras, emplear túneles, etc. En toda actividad criptográfica PGP puede emplear cualquier técnica explicada anteriormente en este capítulo pero por defecto empleará siempre técnicas "híbridas", creando una clave secreta única para cada archivo, permitiendo seleccionar entre casi todos los algoritmos existentes.

En el mismo año, el **IETF** (Internet Engineering Tsak Force) publica la **RFC-2440** "**OpenPGP**" que luego en el año 2007 queda obsoleta por la actual **RFC-4880**, creando la "OpenPGP Alliance" cuya Web es: http://www.openpgp.org/, tal vez sea el mayor estándar para confidencialidad e integridad en correo electrónico y transferencia de archivos.

En el año 1996 Zimmermann crea la compañía "PGP Inc." Que en 1997 es adquirida por NAI (Network Associates Inc.), finalmente en el año 2002 nuevamente la adquiere otra compañía "PGP Corpoation" hasta la fecha. En la actualidad existe también una versión de PGP gratuita para uso no comercial.

En nuestro texto, nos basaremos en **GnuPG** (GNU Privacy Guard), también conocido como "**GPG**" que es la implementación libre de OpenPGP (RFC-4880) desarrollada inicialmente por Werner Koch, con la única limitación de no emplear el algoritmo **IDEA** que tiene patente comercial. GPG es Software Libre, por lo tanto puede ser modificado y distribuido libremente bajo los términos de la GNU General Public License. La primera versión de este software fue publicada en el año 1999. En la actualidad ofrece soporte para todos los SSOO. Su Web es: http://www.gnupg.org

A continuación presentamos una captura

CAPTURA DE UN TRAMA CIFRADA EN SUBRED PGP

```
   ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol

   ETHERNET: Destination address : 0020185751DC

   ETHERNET: .......0 = Individual address
   ETHERNET: ......0. = Universally administered address
   ETHERNET: Source address : 0020185751D2

   ETHERNET: .......0 = No routing information present
   ETHERNET: ......0. = Universally administered address
   ETHERNET: Frame Length : 158 (0x009E)
   ETHERNET: Ethernet Type : 0x0800 (IP:  DOD Internet Protocol)
   ETHERNET: Ethernet Data: Number of data bytes remaining = 144 (0x0090)

   IP: ID = 0x190B; Proto = 0x32; Len: 144

IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Service Type = 0 (0x0)
  IP: Precedence = Routine
  IP: ...0.... = Normal Delay
  IP: ....0... = Normal Throughput
  IP: .....0.. = Normal Reliability
IP: Total Length = 144 (0x90)
IP: Identification = 6411 (0x190B)
IP: Flags Summary = 0 (0x0)
  IP: .......0 = Last fragment in datagram
  IP: ......0. = May fragment datagram if necessary
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 128 (0x80)
IP: Protocol = 0x32 → 32h = 50decimal(IMPLICA AUTENTICATION HEADER-IPSec)
IP: CheckSum = 0xA107
        IP: Source Address = 10.190.10.214    → *** DIRECCIONES DE SUBRED**
        IP: Destination Address = 110.250.10.83 →** QUE FORMAN TUNEL PGP **
IP: Data: Number of data bytes remaining = 124 (0x007C)

   00000:  00 20 18 57 51 DC 00 20 18 57 51 D2 08 00 45 00   . .WQ.. .WQ...E.
   00010:  00 90 19 0B 00 00 80 32 A1 07 C0 A8 FF 6E C0 A8   .......2.....n..
   00020:  FF 69 87 16 96 FA 00 00 08 65 F3 F5 67 7A A1 7F   .i.......e..gz..
   00030:  59 63 51 DC 5B 69 6C 28 12 5E 73 84 00 AF 47 A6   YcQ.[il(.^s...G.
   00040:  86 24 8A 1D A9 8C 38 A1 4A C8 B0 4A FC 90 90 9A   .$....8.J..J....
   00050:  FF 3A AB 8C D0 1B CE 70 14 18 5B 9A 8D 3C 6F 92   .:.....p..[..<o.
   00060:  9F DD BA F9 EB 87 41 AE 96 CA F6 FA FF 34 6A E0   ......A......4j.
   00070:  B7 41 94 1D FD 86 29 9F 28 5F 18 AE 47 00 CC 0A   .A....).(_..G...
   00080:  68 81 26 67 FB 5B F9 05 DC FA B8 96 86 12 BB FC   h.&g.[..........
   00090:  BD 99 E6 56 14 D1 EB 31 D7 6F 0D E9 AC 8E         ...V...1.o....
```

TEXTO CIFRADO

Como se puede apreciar, en la captura anterior, en este envío de datos, directamente PGP emplea criptografía a nivel de "red", es decir sólo estamos viendo en texto plano hasta el encabezado "IP", luego todo viaja cifrado. En este caso se está empleando el protocolo "IPSec" que se tratará más adelante, pero en la captura se puede apreciar este valor "32"

(resaltado en rojo) que implica "Autentication Header" que es una de las posibilidades que ofrece justamente IPSec.

En la parte de ejercicios trataremos con todo detalle y en forma eminentemente práctica GPG, por lo tanto no creemos necesario más desarrollo teórico.

8.1.12. Sistema de autenticación Kerberos

Este sistema nace en el Instituto Tecnológico de Massachusetts y su nombre se remonta a la mitología Griega donde así se denominaba el perro guardián de los Dioses. Este sistema de autenticación hoy es soportado por la masa de los sistemas operativos y componentes de red.

El sistema Kerberos está basado en un "Servidor despachador de boletos", al cual se encarga de validar la identidad de los "Principales" los cuales pueden ser:

- Usuarios.
- Servicios.

En cualquiera de los dos casos, un "Principal" queda definido por un "Trío" cuyos componentes son:

- Nombre primario: Nombre de persona o servicio.
- Instancia: Para usuarios es nula o contiene información de ésta. Para un servicio es el nombre de la máquina.
- Reino: Define distintos Dominios de autenticación.

Si un "Principal" obtiene un boleto, éste tendrá un tiempo de vida limitado por el Servidor, y a partir de este poseerá una clave privada que sólo conocerán el Principal y el Servidor, por lo tanto será considerada auténtica y a través de esta podrá acceder a los recursos del sistema.

8.1.13. RADIUS (Remote Authentication Dial-In User Server)

Este protocolo nace en 1991, aunque se estandariza recién en 1997 con las RFC-2138 y RFC-2139, las cuales quedan obsoletas en el año 2000 por las RFC-2865 y RFC-2866 en el año 2000. Como su nombre lo indica, se trata de un protocolo de "Autenticación", que a su vez amplía también su alcance sobre "control de accesos" operando por defecto sobre el puerto **UDP 1812**. En la actualidad el mayor provecho debe ser su empleo para esta actividad en las redes WiFi como parte del proceso de autenticación que propone "802.1x".

RADIUS es una arquitectura cliente-servidor, en la cual, un denominado "**Network Access Server**" (NAS) opera como un "cliente" del servidor RADIUS. Este NAS es el

responsable de pasar la información de usuario hacia el servidor RADIUS de forma segura. El servidor por su parte, es el responsable de autenticar al usuario y retornarle toda la información de configuración que éste necesite, toda esta comunicación se llevará a cabo por medio de un "secreto compartido" que nunca se enviará por la red en texto plano. Cualquier usuario podría conectarse al NAS para que éste gestione su autenticación.

El servidor puede soportar diferentes métodos de autenticación de usuario, la RFC-2865 nos indica PPP, PAP o CHAP, o login tipo UNIX u otros mecanismos.

La RFC-2865 es un ejemplo claro del empleo de UDP y/o TCP, en el punto 2.4 de la misma se hace la pregunta ¿Por qué UDP?, la cual queda explicada en cuatro aspectos:

- Si la solicitud de autenticación a un servidor RADIUS primario falla, un servidor secundario debe ser requerido, y no se desea entrar en retransmisiones o nuevos establecimientos de sesión.
- El tiempo de respuesta de UDP frente a TCP es significativo y no es importante la pérdida de datos.
- La técnica de "no" control de estados simplifica el proceso.
- UDP simplifica también la implementación del servidor

Si bien aclara que esto no es la panacea, técnicamente se ha optado por esta mayor eficiencia, frente a la "segundad de entrega" que nos daría TCP.

Los paquetes RADIUS son "encapsulados" dentro del campo de datos de UDP en este puerto 1812. EL formato de su encabezado es el siguiente:

8 bit	8 bit	8 bit	8 bit
Código	Identificador	Longitud	
Autenticador			
Atributos.....			

- Código (8 bits): Identifica el tipo de paquete RADIUS. Si posee un valor no contemplado (inválido) se descarta automáticamente sin generar respuesta (silenciosamente). Los códigos válidos son:

  1 : Solicitud de acceso

  2 : Acceso aceptado

  3: Acceso rechazado

  4: Solicitud de autenticación

  5: Respuesta de autenticación

  11: desafío de acceso

  12: Estado del Servidor (experimental)

  13: Estado del cliente (experimental)

255: Reservado

- Identificador (8 bits): Se emplea para mantener la correspondencia entre solicitudes y respuestas.
- Longitud (16 bits): Indica la longitud de todo el paquete, incluyendo el código, identificador, longitud, autenticador y atributos. La mínima longitud es de 20 octetos y la máxima es 4096. Si el paquete fuera más corto que lo que aquí se indica, también deberá ser descartado silenciosamente.
- Autenticador (16 octetos): Este valor se emplea para autenticar la respuesta desde el servidor RADIUS y para esconder el algoritmo de la contraseña.
    - Cuando se trata de una Solicitud de acceso (es decir código 1) este valor es un número "random" de 16 bytes (u octetos), se recomienda que este valor debería ser impredecible. Tanto el NAS, como el servidor RADIUS comparten previamente un secreto, por lo tanto sobre esta solicitud y el secreto se genera un "Hash" MD5 para crear un resumen de 16 bytes al cual se le aplicará la función "XOR" concatenando la clave del usuario.
    - Cuando se trata de una respuesta (códigos: acceso aceptado = 2, o acceso rechazado = 3), este campo contiene también un "Hash" MD5 calculado sobre el flujo de octetos consistentes en: El paquete completo, los atributos de la respuesta, seguidos del secreto compartido.
- Atributos: La RFC-2865 en el punto 5 establece todos los atributos especificados para cada tipo de paquete, todos ellos responden a un mismo formato de tres campos:
    - tipo (8 bits) presenta un listado con 63 de ellos.
    - longitud (8 bits): define la longitud de todos los atributos presentes en este paquete.
    - Valor: desde cero a más octetos, y contienen información acerca de estos atributos.

En nuestro caso, como es costumbre, proponemos que si deseáis profundizar en el tema, lo hagáis a través del "**Free RADIUS Project**" cuya Web es: http://freeradius.org. Allí encontrarás todas las herramientas y documentación para implantar esta arquitectura de forma gratuita.

8.2. PKI (Infraestructura de clave pública)

El concepto de PKI es el conjunto de documentos, servicios y funciones que garanticen el trabajo confiable a través de certificados digitales.

Este sencilla descripción como iremos viendo a lo largo de este punto, implica una serie de medidas, hardware y software que ya no son tan simples y que a su vez requieren un constante mantenimiento y actualización, por esta razón es que iremos tratando cada uno de sus aspectos e incluyendo también en este punto algunas soluciones para trabajar con "pares de claves" que pueden resultar bastante más sencillas en su implantación.

8.2.1. Situación, casos y empleos

Hasta aquí hemos tratado las diferentes opciones de empleo de clave asimétrica, en el capítulo anterior ya habíamos presentado SSL como metodología segura de acceso al nivel de aplicación, ahora empecemos a plantearnos ¿cómo se logra establecer esta comunicación segura si nos toca administrarla a nosotros?

Una muy buena práctica será siempre el empleo de claves asimétricas. Cuando se decide este tipo de empleo, lo primero que hará falta es una autoridad de emisión de claves, es decir un servidor desde el cual se puedan generar el par "Clave Pública – Clave Privada" para cada usuario de nuestro sistema. Hasta aquí la idea es poder instalar este servidor, y "bastionarlo" lo suficientemente bien como para que nadie pueda llegar a comprometerlo, pues si sucediera algún problema sobre el mismo, como éste será la "**raíz**" de todo el sistema, se nos caerá toda nuestra infraestructura de seguridad. Es más, una muy buena medida es mantenerlo aislado físicamente y cada vez que se genere un par de claves, distribuir las mismas "fuera de línea", es decir, por medio de una memoria USB, un CD, etc, pero negando todo tipo de acceso a este servidor mediante la red.

El interrogante más importante ahora será ¿Quiénes serán los usuarios de esta infraestructura de seguridad?, de esta respuesta dependerá todo lo que pueda seguir haciendo.

Si los usuarios de este sistema es personal de la propia empresa, el camino es más sencillo, si a su vez es también "partners o clientes" conocidos, aún podemos administrarlo nosotros, pero este tema nos excede o se nos va de las manos, cuando a nuestro sistema debe acceder alguien desconocido, por ejemplo contamos con una página Web a la cual se puede acceder desde cualquier lugar del mundo y a su vez realizar una transacción de comercio electrónico. En este último caso, evidentemente ambas partes deberán poder ser "confiables" y para este ejemplo no nos queda más remedio que obtener un certificado de una "**entidad de certificación**" reconocida internacionalmente, esas que hemos mencionado y vienen precargadas en todos los navegadores.

Cada uno de estos supuestos son los que desarrollaremos a continuación.

8.2.2. Certificados digitales

La diferencia entre emitir un "par de claves" y emitir un "certificado digital" básicamente radica en que el certificado está emitido por una "**Autoridad de Certificación**" que FIRMA el mismo, luego veremos que se incluyen una serie de campos, pero su diferencia radica fundamentalmente en el concepto mencionado.

En virtud de lo que acabamos de expresar, se deduce que para trabajar con certificados digitales, es necesario contar con una "**Autoridad de certificación**", como expresamos en el párrafo anterior. Dependiendo de los usuarios que hagan uso de estos certificados, esa "Autoridad de certificación" puede ser implantada por nosotros (para empleados, clientes, partners, etc.), o deberemos recurrir a una internacionalmente reconocida si los usuarios son desconocidos. Este aspecto lo seguiremos tratando en los párrafos siguientes, en éste nos limitaremos a describir qué es un certificado digital.

Existen varios tipos de certificados estandarizados, pero el mercado (y casi el 100% de Internet) se ha decantado por la familia **UIT-T X.509** (Unión Internacional de Telecomunicaciones- sector Telecomunicaciones).

Los certificados X.509 nacen en el año 1988, como parte de la familia X.500 que fue un trabajo conjunto entre **ISO** e **ITU-T**, esta familia abarca aspectos referentes a servicios de directorio, y justamente dentro de esta familia (o serie), el que más éxito llega a tener es X.509 que no trata de servicios de directorio, sino de certificados digitales.

Una vez más sucedió que la velocidad de Internet superó la de estas grandes organizaciones y la **IETF** a través de su grupo de trabajo **PKIX (**para infraestructura de clave pública), tomó las riendas de este estándar publicando en el año 1999 la **RFC-2459** (Internet X.509 Public Key Infrastructure Certificate and CRL Profile), que sucesivamente va quedando obsoleta por las RFC: 3280 y luego la actual **RFC-5280** en el año 2008, que es la que verdaderamente regula los aspectos de lo que hoy llamamos "**X.509 versión 3**". Esta última versión según la RFC está en capacidad de soportar cualquier aplicación del tipo WWW, correo electrónico, autenticación de usuarios y protocolo IPSec (que trataremos más adelante).

Los Certificados digitales son documentos (digitales) que sirven para asegurar la veracidad de la Clave Pública perteneciente al propietario del certificado o de la entidad, con la que se firman digitalmente documentos. Deben proporcionar las más absolutas garantías de seguridad respecto a cuatro elementos fundamentales:

- La autentificación del usuario/entidad (es quien asegura ser).
- La confidencialidad del mensaje (que sólo lo podrá leer el destinatario).
- La integridad del documento (nadie lo ha modificado).
- El no repudio (el mensaje una vez aceptado, no puede ser rechazado por el emisor).

Otro dato a tener en cuenta, es que un certificado no puede falsificarse ya que va firmado por una Autoridad de Certificación **(CA).** Si algún dato se modificase la firma no correspondería con el resumen (Hash) que se obtendría de los datos modificados. Por tanto al utilizarlo, el software que los gestiona daría un mensaje de invalidez.

Un certificado electrónico contiene una **clave pública**, y una **firma digital.**

Para su correcto funcionamiento, los certificados contienen además la siguiente información:

- Un identificador del propietario del certificado, que consta de su nombre, sus apellidos, su dirección e-mail, datos de su empresa como el nombre de la organización, departamento, localidad, provincia y país, etc.
- Otro identificador de quién asegura su validez (AC), que será una Autoridad de Certificación.
- Dos fechas, una de inicio y otra de fin del período de validez del certificado, es decir, cuándo un certificado empieza a ser válido y cuándo deja de serlo, fecha a partir de la cual la clave pública que se incluye en él, no debe utilizarse para cifrar o firmar.
- Un identificador del certificado o número de serie, que será único para cada certificado emitido por una misma Autoridad de Certificación. Esto es, identificará inequívocamente a un certificado frente a todos los certificados de esa Autoridad de Certificación.
- Firma de la Autoridad de Certificación de todos los campos del certificado que asegura la autenticidad del mismo.

Los navegadores actuales gestionan y almacenan las Claves Públicas de los certificados que permiten al emisor de mensajes firmarlos y cifrarlos utilizando las claves públicas de los destinatarios. Para estar completamente seguros en cualquier transacción es necesario utilizar, al menos dos tipos de certificados:

- uno general para comunicaciones seguras (X.509).
- otro especifico para transacciones económicas (SET).

Además de servir como mecanismo confiable y seguro de identificación en la red, un certificado de identidad digital permite disfrutar de otra serie de beneficios:

- Enviar y recibir información confidencial, asegurándose que sólo el remitente pueda leer el mensaje enviado.
- Acceder a sitios Web de manera segura con su identidad digital, sin tener que usar el peligroso mecanismo de passwords.
- Firmar digitalmente documentos, garantizando la integridad del contenido y autoría del documento; y todas aquellas aplicaciones en que se necesiten mecanismos seguros para garantizar la identidad de las partes y confidencialidad e integridad de la información intercambiada, como comercio electrónico, declaración de impuestos, pagos provisionales, uso en la banca, etc.

Las aplicaciones de Internet, como navegadores (por ejemplo, Internet Explorer, Firefox o Netscape Navigator), programas para correo electrónico, etc. ya traen incorporados los elementos que les permiten utilizar los certificados de identidad digital, por lo que los usuarios no necesitan instalar ningún software adicional.

Los campos de un certificado X.509, están especificados en el punto "**4.**" De la **RFC-5280**, y establece los siguientes (mantendremos la misma numeración de la RFC):

4.1. Campos básicos de un certificado:

4.1.1. Certificate Fields (Campos del certificado)

Los campos básicos de un certificado son una secuencia de 3 "subcampos" básicos:

4.1.1.1. tbsCertificate (Estimo que "tbs" es por The Basic Syntax)

Son una serie de campos que se detallan en el punto 4.1.2 y que los presentaremos con más detalle a continuación.

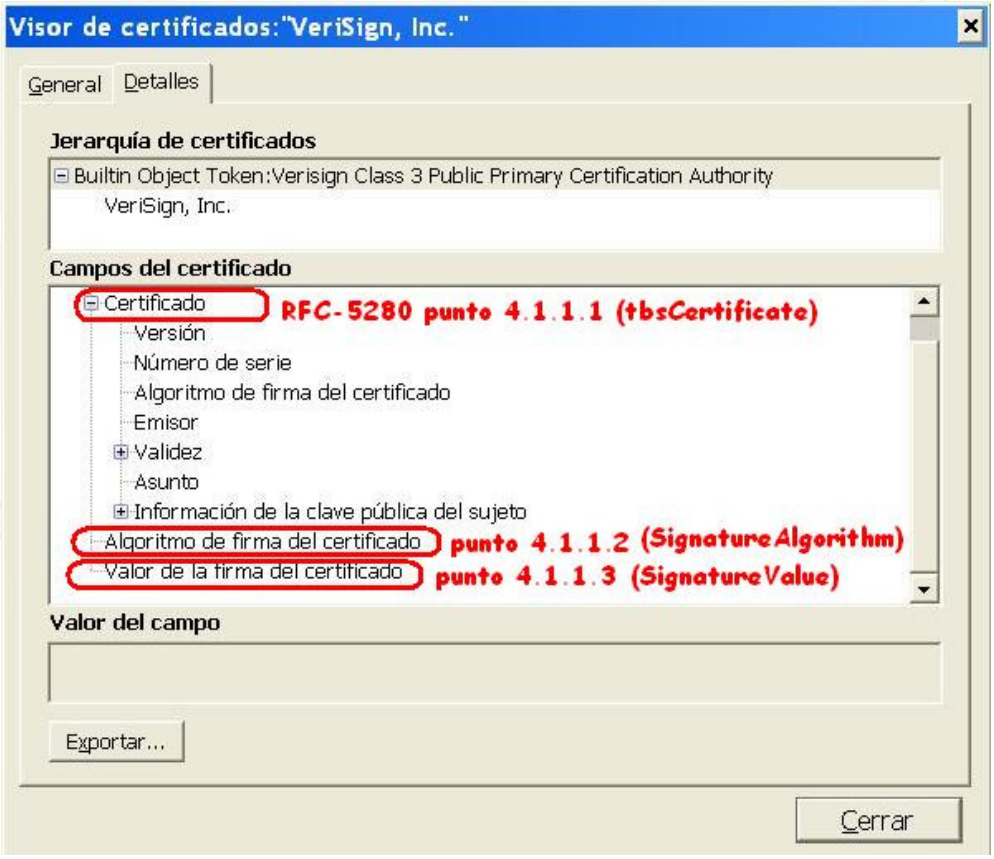
4.1.1.2. signatureAlgorithm (Algoritmo de firma)

Contiene el identificador de este algoritmo criptográfico, la lista de los soportados figura en las RFCs: 3279, 4055 y 4491.

4.1.1.3. signatureValue (Valor de la firma)

Se trata de una cadena de bits que codifica el valor de la firma, los detalles de este proceso están listados en las mismas RFCs citadas en el punto anterior.

Estos tres campos son los que presentamos a continuación en la imagen que puedes ver desde cualquier navegador si consultas uno de los certificados pre-cargados que trae, en nuestro caso podemos apreciar los de un certificado de la propia empresa Verisign.



4.1.2. TBSCertificate

La secuencia TBS contiene la información que está asociada con el sujeto del certificado y la CA que lo emitió, su contenido es el siguiente:

4.1.2.1. Version (Versión)

Puede tener el valor 1, 2 o 3

4.1.2.2. Serial Number (Número de serie)

Debe ser un valor entero positivo asignado a la autoridad de certificación para cada certificado. Para cumplir con este valor único por cada certificado, las CAs DEBEN soportar valores hasta 20 octetos

4.1.2.3. Signature (Firma)

Contiene la identificación del algoritmo empleado por la CA para firmar este certificado

4.1.2.4. Issuer (Emisor)

Este campo identifica a la entidad que firmó este certificado y DEBE contener un valor no vacío de "Distinción de nombre (DN)" definido por el estándar X.501, hace referencia también a poder hacer especificaciones definidas por la RFC-4519, referidas a DNS. Se trata de un campo definido con extremo detalle en esta RFC y también en anexos de la misma.

4.1.2.5. Validity (Validez)

Define el período o intervalo de tiempo durante el cual la AC garantiza el mantenimiento de la información acerca del estado de este certificado. Se representa por una secuencia de dos datos: "no antes de" y "no después de", expresados en formato UTC (Universal Time) o GMT (Generalized Time).

4.1.2.6. Subject (Asunto)

Identifica la entidad asociada con la clave pública guardada en este certificado en el campo "Clave pública". Responde al mismo formato que el campo "Emisor"

4.1.2.7. Subject Public Key Info (Información de clave pública del sujeto).

Identifica el algoritmo empleado para la generación de la clave pública de este certificado, responde a la misma estructura que el algoritmo de firma que trata el punto 4.1.1.2

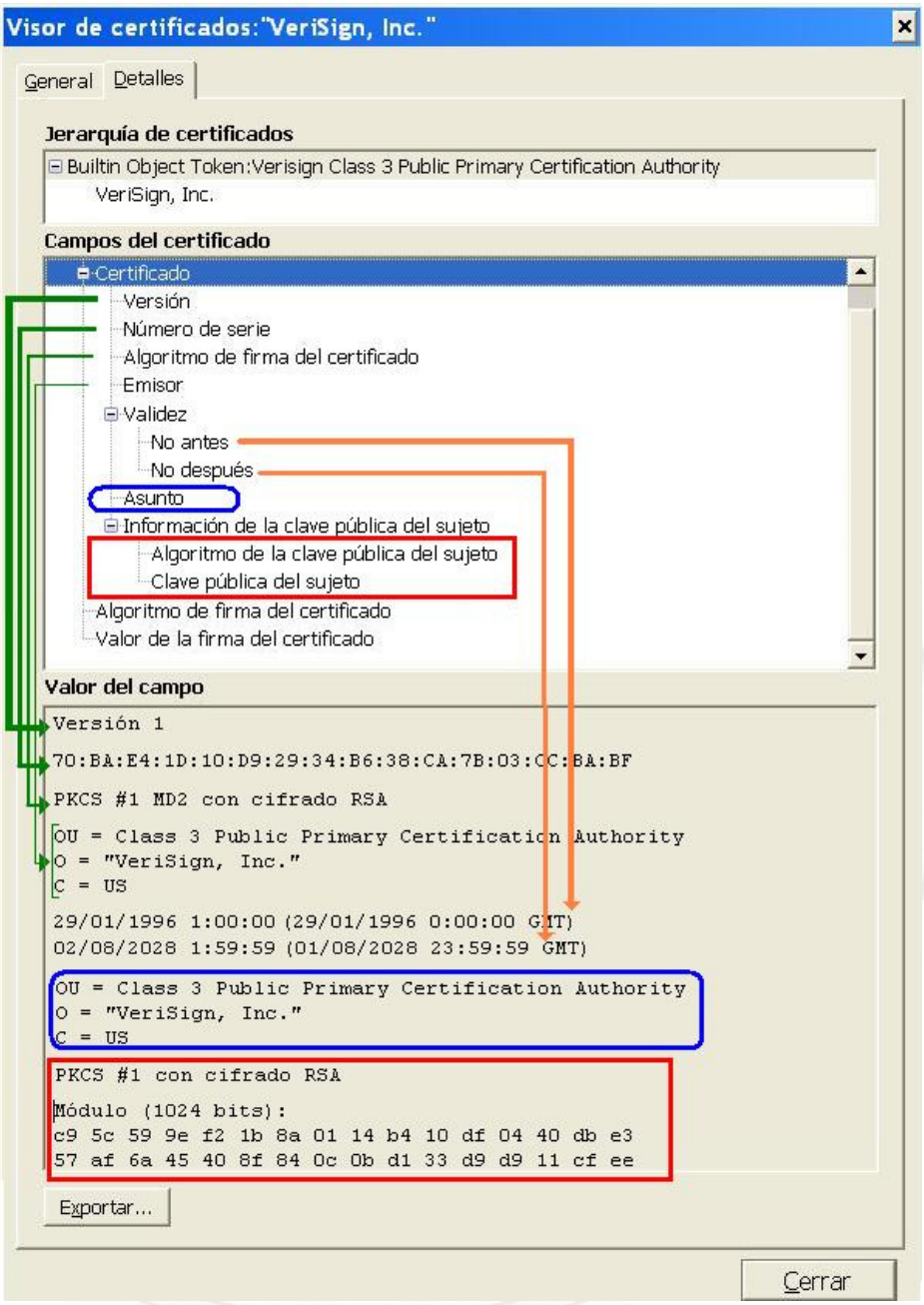
4.1.2.8. Unique Identifiers (Identificadores únicos).

Este campo solo DEBE aparecer en las versiones 2 o 3 y NO DEBE aparecer en la versión 1. Permite la posibilidad de reusar los campos de asunto y emisor, aunque esta RFC RECOMIENDA que no sean reusados

4.1.2.9. Extensions (Extensiones)

Este campo solo DEBE aparecer en la versión 3, y si está presente se trata de una secuencia de uno o más extensiones del certificado y se definen en la sección 4.2.

A continuación presentamos una imagen con el detalle de cada uno de estos campos sobre el mismo certificado de la imagen anterior.



Hasta aquí hemos presentado el formato y los aspectos que podemos encontrar de forma estandarizada, pero sin embargo veremos que diferentes proveedores, aplicaciones o autoridades de certificación, han decidido "Clasificar" diferentes tipos de certificados. Sobre esta clasificación no hay una regulación específica y clara, en general se puede estar de acuerdo en que existen certificados:

- Personales.
- Profesionales.

- De empresa, organización y/o entidad.
- De productos o componentes.

A su vez se han propuesto algunas "**Clases**", las más frecuentes son:

- Clase 1: son los más fáciles de adquirir y no requieren verificación de datos, en general sólo se solicita el nombre y la dirección de correo electrónico del titular.
- Clase 2: En estos casos ya se comprueba de forma fidedigna alguna documentación adicional, suele ser el DNI.
- Clase 3: Se añade a la Clase 2 la verificación de antecedentes empresariales, créditos, volumen de negocio, rendiciones financieras o de hacienda, etc.
- Clase 4: a lo anterior se añade la verificación del cargo o posición de una persona dentro de la empresa u organización.

X.509 es la pieza central de la infraestructura PKI, y es la estructura de datos que enlaza la clave pública con los datos que permiten identificar al titular. Su sintaxis, se define empleando el lenguaje **ASN.1** (Abstract Syntax Notation One), y los formatos de codificación más comunes son **DER** (Distinguish Encoding Rules) o **PEM** (Privacy Enhanced Mail). Siguiendo la notación de ASN.1, un certificado contiene diversos campos.

SET (Secure Electronic Translation)

Al principio de este punto mencionamos que para operar de forma segura desde cualquier navegador es importante considerar dos tipos de certificados X.509 y SET, este último es el que desarrollaremos a continuación.

Se trata de un protocolo especialmente diseñado para comercio electrónico con tarjetas de crédito en redes abiertas (incluyendo a Internet). Nace impulsado por Mastercard y VISA, con la colaboración de IBM, GTE, Microsoft, Verisign, SAIC, ect. El formato de sus mensajes está basado en el estándar RSA (**PKCS-7**). La especificación de SET v0.1 consta de 3 volúmenes publicados en 1997 y es de libre distribución. Actualmente se encuentra la versión 2 del mismo, uno de los puntos de mayor interés de esta nueva versión es la inclusión de tarjetas ICC (se verán a continuación).

El protocolo SET se puede transportar directamente en TCP, mediante correo electrónico con SMTP o MIME y en sitios Web con HTTP.

La gran diferencia que ofrece es la posibilidad de autenticar todas las partes mediante certificados digitales, detalle que minimiza enormemente el fraude.

Para operar SET necesita cuatro piezas de software:

- Software cartera del titular: Aplicación que permite a los compradores almacenar información acerca de sus datos personales para el envío de las mercancías compradas, así como información de pago, como número de tarjeta de crédito y banco emisor. Debe ser compatible con SET, ya que constituye el medio a través del cual se transmite la información de su certificado digital en los pagos por Internet. Para garantizar la seguridad de sus datos, el monedero los protege

mediante una contraseña. Microsoft distribuye una aplicación monedero con su navegador Internet Explorer 4.0 ó superior (Herramientas, Opciones de Internet..., Contenidos, Pagos). SafeLayer (www.safelayer.com) comercializa en España una aplicación de cartera digital. Si su banco emite certificados SET, distribuirá también software de monederos digitales. Puedes consultarlo con él.

- El software de punto de venta del comerciante: para que el sitio Web del comerciante acepte pagos con SET necesitará instalar una aplicación de Terminal de Punto de Venta (**TPV**) compatible con SET en su servidor, que acepte los pedidos y procese los pagos con el banco.
- El software del servidor de la pasarela de pagos: realiza el procesamiento automatizado de los pagos. La pasarela recibe peticiones de autorización/liquidación/reconciliación de pagos de los sistemas del comerciante (TPV) en Internet y las encamina hacia los sistemas de pago propietarios (sistemas de autorización tradicionales).
- El software de la autoridad de certificación: las entidades financieras que decidan soportar el estándar SET necesitarán este software para que sus respectivos clientes (titulares de tarjetas y comerciantes que aceptan pago con tarjeta) puedan participar en el juego. Permite registrar a los usuarios y emitir certificados digitales para ellos, que aseguren la confianza entre las partes.

Los puntos fuertes de SET son también su mayor debilidad: la autentificación de todas las partes exige rígidas jerarquías de certificación, ya que tanto clientes, como comerciantes, deben adquirir certificados distintos para cada tipo de tarjeta de crédito, trámites que resultan engorrosos para la mayoría de los usuarios. Se añade el problema de la revocación de certificados, la portabilidad de los mismos cuando el usuario trabaja en distintas máquinas y las cadenas de certificación. En definitiva, SET descansa sobre una infraestructura de clave pública (PKI) que en la actualidad dista mucho de ser perfecta.

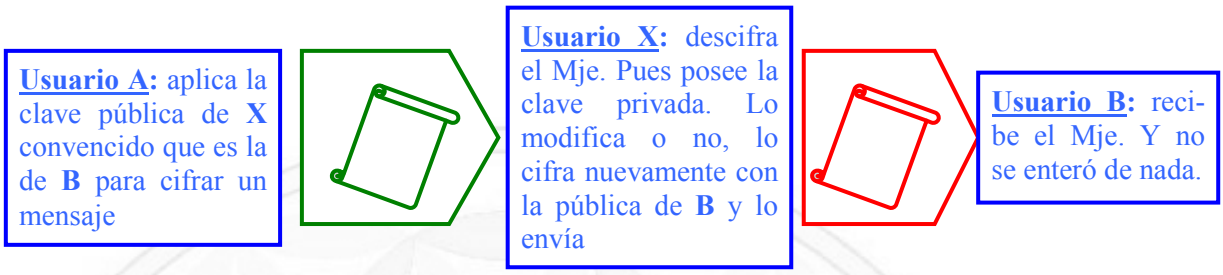
SET seguirá coexistiendo con SSL durante mucho tiempo, hasta que se alcance una masa crítica de usuarios que propicien su utilización a gran escala, o caiga en el olvido superado por otra nueva iniciativa más ágil y mejor adaptada. Las opiniones de los analistas se encuentran divididas acerca de su futuro. En lo que todos coinciden es que aún le queda un largo camino por recorrer.

8.2.3. Estructuras de confianza

Cuando tratamos el tema de criptografía de clave asimétrica, pusimos dos ejemplos de empleo correcto e incorrecto, el segundo de ellos ponía de manifiesto la única debilidad que en la actualidad sufre esta metodología: el "ataque del hombre del medio", el cual sencillamente se trataba de generar un par de claves falsas y lograr que cuando alguien desee cifrar un archivo hacia un destinatario, en vez de obtener su clave pública verdadera, lo hiciera con esta falsa, con lo cual la única persona que lograría descifrarlo sería el poseedor de esa clave privada falsa (el intruso u hombre del medio), luego podría a su vez reenviarla al destinatario original, esta vez sí aplicando la clave pública verdadera de este, el cual le

respondería con otra clave falsa hacia el emisor, etc. Quedando siempre en el medio de esta comunicación ese que falsificó las claves.

Un ejemplo muy simplificado sería el siguiente:



Para evitar este problema, la solución más sencilla es desarrollar una metodología segura para la obtención de las claves públicas de la organización, en algunos casos es tan sencillo como redactar un documento y que el mismo se cumpla, enviando las claves por CDs, descargándolas de un único servidor (y verificando su huella digital), preguntando por teléfono al usuario destinatario cuál es la huella digital de su clave pública, teniendo un listado único de huellas digitales, etc... La idea de esta propuesta es lograr una "Estructura de confianza", es decir como dijimos al principio de este punto, deberemos contar con una autoridad de emisión de claves (segura), pero tal vez nada más, siempre y cuando la metodología que diseñemos pueda garantizar que NADIE de nuestros usuarios pueda dar cabida al hombre del medio.

En la práctica podemos asegurar que se han desarrollado un sinnúmero de sistemas que gracias a técnicas creativas y sencillos documentos, con un buen control de los administradores funcionan a la perfección, y sobre todo sin la necesidad de incrementar toda la infraestructura con mayores componentes que este simple emisor de claves.

Esta metodología de trabajo, se complica cuando empiezan a aparecer usuarios que están fuera del control de nuestra organización (clientes, partners, socios, etc.), en estos casos, nadie podrá obligarlos a cumplir con un determinado procedimiento, y hasta tal vez no contemos con accesos seguros, métodos de control de accesos desde el exterior, y nuestra sencilla "estructura de confianza" no será suficiente, o inclusive será inaplicable, en estos casos deberemos empezar a pensar en una PKI.

8.2.4. Componentes de una PKI

Una PKI como iniciamos este tema, es un conjunto de documentos, servicios y funciones que garanticen el trabajo confiable a través de certificados digitales. Para poder conformar toda esta infraestructura, como veremos a continuación es necesario una serie de elementos.

Si bien dentro de un mismo Hardware y/o Software puede ejecutarse más de un componente, e inclusive puede en algunos casos estar ausente alguno de ellos, conceptualmente los componentes de una PKI son:

- **Autoridad de Certificación**: Es uno o varios servidores cuya misión es la de emitir y revocar certificados.
- **Autoridad de registro**: Es el elemento responsable de verificar la correspondencia entre una clave pública y una privada y las identidades asociadas a los certificados correspondientes.
- **Repositorios**: Es donde se debe almacenar la información de los certificados, los dos repositorios que son indispensables son:
  - Repositorio de certificados: desde donde se deberían descargar los certificados o claves públicas de los usuarios.
  - Repositorio de Listados de revocación: es donde se lleva el listado de todos los certificados que por alguna razón han sido revocados, es decir no están vigentes en esta arquitectura. Se la suele reconocer por sus siglas en inglés CRL: "Certificate Revocation List"
- **Autoridad de sellado de tiempo**: responsable de emitir estos sellos

8.2.5. ¿PKI o estructuras de confianza?

Como tratamos de expresar en los dos puntos anteriores, en la mayoría de los casos estaremos obligados a decidir por una u otra, pero en realidad el factor más importante es si tenemos a nuestro alcance el cumplimiento estricto de una "Estructura de confianza" o no, si lo tenemos no hay ninguna duda que lo más sencillo e igualmente eficiente es la implantación de una de ellas, y en caso contrario, no nos queda más recurso que lanzar una PKI.

¿Por qué razones elegir una u otra?

- Por coste de implementación.
- Por mantenimiento.
- Por magnitud de toda la infraestructura.
- Por imagen.
- Por necesidad de operación con usuarios y/o empresas desconocidas.
- Por necesidad o exigencias de operación.

8.2.6. ¿Certificados de terceros o propios?

El segundo interrogante será ¿Es necesario adquirir certificados de pago?

En el momento de comenzar a trabajar con extraños, nos quedan aún dos posibilidades:

- Que acepten acceder a un sitio "no verificable".
- Que no lo acepten.

Seguramente ya nos ha sucedido en muchas organizaciones públicas y/o privadas, que al acceder a un sitio Web o una aplicación determinada, nos aparezca una ventana aclarando que "este certificado no pertenece a ninguna entidad de certificación" (o algo por el estilo) y luego "¿Desea añadir una excepción?". Este tema lo presentamos en el capítulo anterior cuando vimos SSL, pero en realidad aplica a este concepto de certificados digitales, pues lo que nos está anunciando específicamente esa ventana es que vamos a elegir o no acceder a un sitio, cuyo certificado fue emitido por el administrador de ese sistema y no por una "entidad de certificación" pre-cargada en nuestro navegador. Si la actividad que voy a realizar con ellos no implica un serio riesgo, normalmente uno selecciona "Añadir excepción" → "Obtener certificado" → "Aceptar", pero si conocemos un poco sobre temas de seguridad, y por ejemplo este acceso es al Banco Santander, nos llamará la atención que esa página Web no sea propietaria de un certificado emitido por una "entidad de certificación" reconocida internacionalmente, y seguramente no lo aceptemos y declinemos de seguir adelante.

Por lo tanto, si nuestro sistema posee la característica de ser importante su presencia hacia cualquier lugar del mundo, y que al mismo deben acceder de forma "confiable" usuarios que no nos conocen, no tendremos otra alternativa que adquirir un certificado emitido por una "entidad de emisión de certificados" que ya esté pre-cargada en TODOS los navegadores y que haga de tercera parte confiable (como ya lo desarrollamos en el capítulo de SSL).

8.2.7. Ventajas y desventajas.

Las ventajas de no tener que pagar un certificado digital son meramente económicas, las de montar una PKI son también del mismo tipo y a su vez se le suma todo el trabajo adicional que implica montar, mantener y actualizar muy periódicamente toda la infraestructura. Por último las ventajas de una estructura de confianza son su sencillez y practicidad, y su desventaja es el limitado ámbito al que generalmente aplica.

8.2.8. Estándares PKCS

PKCS: Public-Key Cryptography Standards, son un conjunto de especificaciones técnicas desarrolladas por Netscape, RSA y otros desarrolladores de informática cuyo objeto es uniformizar las técnicas y protocolos de la criptografía pública. La primera publicación (versión 1.0) se hace en el año 1991.

PKCS forma parte de distintos estándares de hecho como ANSI PKIX, X9, SET, S/MIME y SSL.

A la fecha existen 14 documentos con títulos genéricos que van desde PKCS #1 a PKCS #15.

El de mayor trascendencia podría ser PKCS #11 llamado CRYPTOKI. A continuación se presentan cada uno de ellos:

- PKCS #1: RSA Cryptography Standard
- PKCS #2: Incluido ahora en PKCS #1
- PKCS #3: Diffie-Hellman Key Agreement Standard
- PKCS #4: Incluido ahora en PKCS #1
- PKCS #5: Password-Based Cryptography Standard
- PKCS #6: Extended-Certificate Syntax Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #8: Private-Key Information Syntax Standard
- PKCS #9: Selected Attribute Types
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard
- PKCS #15: Cryptographic Token Information Format Standard

8.3. Qué busca y cómo opera un intruso

En general se debe tener en cuenta que un INTRUSO dedica todo o la masa de su tiempo a nuestra red, pues es esta su actividad, y casi con seguridad está al tanto de las últimas novedades encontradas en Internet.

La mejor comparación (Sin ofender a "los buenos") es el caso inverso que plantea una prisión; todo el personal de seguridad día a día analiza el problema de seguridad global, es más, se nutre de los desarrollos realizados en otras entidades similares de todo el mundo, pero dentro de cada una de ellas existen muchas personas que les SOBRA TIEMPO y que observan no lo global, sino el detalle fino, esas muy pequeñas cosas que se pueden llegar a pasar por alto, a esto le dedican todo su tiempo pues vulnerarlas es su desafío, y tarde o temprano lo logran.

8.3.1. Cómo se autodenominan

- **Hackers:** Estos personajes son expertos en sistemas avanzados. En la actualidad se centran en los sistemas informáticos y de comunicaciones. Dominan la programación y la electrónica para lograr comprender sistemas complejos como la comunicación móvil. Su objetivo principal es comprender los sistemas y el funcionamiento de ellos. Les encanta entrar en computadores remotos, con el fin de decir "Su sistema ha sido vulnerado" pero no modifican ni se llevan nada del computador atacado. Normalmente son quienes alertan de un fallo en algún programa comercial, y lo comunican al fabricante. También es frecuente que un buen Hacker sea finalmente contratado por alguna importante empresa de seguridad.

- **Crackers:** Es el siguiente eslabón y por tanto el primero de una familia "rebelde". Cracker es aquel fascinado por su capacidad de romper sistemas y software y que se dedica única y exclusivamente a crackear sistemas. Para los grandes fabricantes de sistemas y la prensa este grupo es el más rebelde de todos, ya que siempre encuentran el modo de romper una protección. Pero el problema no radica ahí, si no en que esta "rotura" es difundida normalmente por medio de Internet para conocimiento de otros; en esto comparten la idea y la filosofía de los Hackers.

- **Lamers:** Este grupo es quizás el que más número de miembros posee y quizás son los que mayor presencia tienen en la red. Normalmente son individuos con ganas de hacer Hacking, pero que carecen de cualquier conocimiento. Este es quizás el grupo que más peligro representa en la red ya que ponen en práctica todo el software de hacking que encuentran, sin tener experticia sobre ellos.

- **Copyhackers:** Es una nueva raza sólo conocida en el terreno del crackeo de hardware, mayoritariamente del sector de tarjetas inteligentes empleadas en sistemas de televisión de pago.

- **Bucaneros:** Son peores que los Lamers, ya que no aprenden nada ni conocen la tecnología. Comparados con los piratas informáticos, los bucaneros sólo buscan el comercio negro de los productos entregados por los Copyhackers. Los bucaneros sólo tienen cabida fuera de la red, ya que dentro de ella, los que ofrecen productos "Crackeados" pasan a denominarse "piratas informáticos".

- **Phreaker:** Este grupo es bien conocido en la Red por sus conocimientos en telefonía. Un Phreaker posee conocimientos profundos de los sistemas de telefonía, tanto terrestres como móviles. En la actualidad también poseen conocimientos de tarjetas prepago, ya que la telefonía móvil ha cobrado un terreno importante en las comunicaciones.

- **Newbie:** Es un novato o más particularmente es aquel que navega por Internet, tropieza con una página de Hacking y descubre que existe un área de descarga de buenos programas de Hacking. Después se baja todo lo que puede y empieza a trabajar con los programas. Al contrario que los Lamers, los Newbies aprenden el Hacking siguiendo todos los cautos pasos para lograrlo y no se mofa de su logro, sino que aprende.

- **Script Kiddie:** Denominados Skid kiddie o Script kiddie, son el último eslabón de los clanes de la red. Se trata de simples usuarios de Internet, sin conocimientos sobre Hack o el Crack en su estado puro. En realidad son devotos de estos temas, pero no los comprenden. Simplemente son internautas que se limitan a recopilar información de la red. En realidad se dedican a buscar programas de Hacking en la red y después los ejecutan sin leer primero los archivos "Readme" de cada aplicación. Esta forma de actuar, es la de total desconocimiento del tema, lo que le lleva a probar y probar aplicaciones de Hacking. Podrían llamarse los

"pulsabotones" de Internet. Los Kiddies en realidad no son útiles en el progreso del Hacking.

8.3.2. Razones por las que un intruso desea ingresar a un sistema informático

En general las razones por las que un intruso desea ingresar a un sistema son las siguientes:

- Por diversión.
- Para mirar o investigar.
- Para robar.
- Para alterar información.
- Por desafío personal, ego o "prestigio en su ámbito".
- Por Ciber terrorismo.

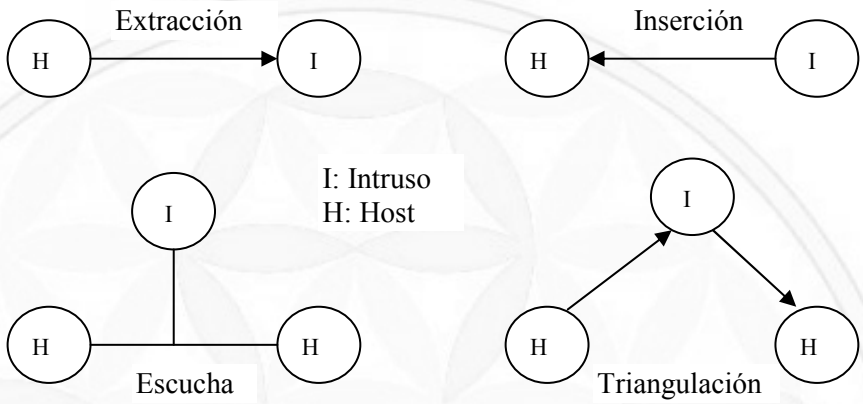
8.3.3. El proceder o los pasos que esta gente suele emplear

- Búsqueda de información publicada: Se trata de una investigación de todo tipo de información del "target" u objetivo, que se encuentre disponible, generalmente a través de Internet (Organigrama, cargos, nombres, teléfonos, direcciones, correos, infraestructura, mails, datos fiscales, accionistas, etc.).
- Localización de rangos: Averiguación de nombres de responsables, direcciones IP y DNSs, generalmente a través de RIPE, ARIN, APNIC, etc.
- Passive Fingerprinting: Identificación de sistemas sin generar tráfico.
- Active Fingerprinting: Identificación de máquinas activas y de sistemas, inyectando tráfico estratégicamente generado en la red, para verificar las diferentes respuestas de los mismos.
- Footprinting: Se trata de todo el proceso de acumular datos observando un entorno de red específico, normalmente tiene el objetivo de comenzar a encontrar brechas de seguridad.

Uno de los primeros desafíos de un intruso una vez que ha logrado penetrar en una red es obtener las listas de usuarios y contraseñas, los cuales si bien suelen estar criptografiadas, al obtenerse una copia de estas es sólo cuestión de tiempo el resolverlas. En general se suelen utilizar programas "buscadores" que en virtud del diccionario que posean y la velocidad de la CPU empleada, demoran más o menos en realizar esta tarea. La gran ventaja es que estos

tiempos no suelen ser lo breves que se desean siempre y cuando la estrategia de contraseñas está bien implementada.

8.3.4. Tipos de ataques



Otra forma de plantearlos es:

- **Interrupción**: Un recurso del sistema es destruido o se vuelve inutilizable. Este es un ataque a la *disponibilidad*.
- **Interceptación**: Un tercero no autorizado obtiene acceso a un recurso. Este es un ataque a la *confidencialidad*.
- **Modificación**: Un tercero sin autorización, no solo obtiene acceso sino también corrompe el recurso. Este es un ataque a la *integridad*.
- **Fabricación**: Un tercero no autorizado inserta objetos falsos en el sistema. Este es un ataque a la *autenticidad*.

Tipo de ataque	**¿Qué ataca?**
Interrupción	*disponibilidad*
Interceptación	*confidencialida*
Modificación	*integridad*
Fabricación	*autenticidad*

8.3.5. Cómo pueden clasificarse los ataques

Los ataques a sistemas informáticos pueden clasificarse como:

- **pasivos**: el objetivo es obtener información que está siendo transmitida, pueden ser:

- Análisis de tráfico: Se emplea para determinar protocolos, puertos direcciones de hardware y Software, nombres, contraseñas, servidores, bases de datos, horarios, dispositivos, segmentos, subredes, permisos, etc.
- Obtención del contenido de los mensajes: Para interpretación de los mismos.

activos: involucran tomar participación en la transmisión; modificando o generando datos falsos, estos pueden ser de cuatro tipos:

- Falsificación de la identidad: Previamente se debe obtener cuentas, permisos o identidades de usuarios de una red para posteriormente hacerse pasar por ellos.

  Se debe tener en cuenta especialmente que cuanta mayor autoridad administrativa posea su identidad, mayor será la capacidad de este ataque.
- Retransmisión: Se trata de obtener un mensaje (Triangulándolo), para luego colocarlo nuevamente en el canal de comunicaciones. Esta actividad puede llevar involucrada o no la modificación, como así también podrá ser retransmitido al destino real o algún otro.
- Modificación de mensajes: Alterar los datos del mismo.
- Negación de servicio: Tomar medidas para que un determinado servicio no esté accesible en la red, generalmente se basa en la generación de un alto tráfico aparentemente legítimo. Se los conoce como *"ataques de inundación*"

8.3.6. Problemas que pueden ocasionar

Si bien es inimaginable la cantidad de problemas que esta actividad puede ocasionar, se pueden agrupar ellos dentro de los siguientes conceptos:

- Destrucción de Software y datos.
- Extracción de información.
- Análisis pasivo de tráfico.
- Generación de tráfico, ocasionando baja performance o paralización de la red.
- Negación de servicios o recursos.
- Inserción de virus.
- Modificación de información.
- Modificación de rutas.

8.3.7. Esquema resumen de pasos y tipos de ataques

¿Deseas conocer algunos casos?

1) Espías empresariales en la Red (Fuente: **www.larioja.com** - 22.06.05)

Las empresas son conscientes del peligro del spyware pero no saben qué medidas tomar al respecto.

Una veintena de ejecutivos y directivos de compañías claves de Israel han sido detenidos este mes acusados de espionaje industrial mediante el uso de software espía, fundamentalmente un 'troyano', introducido en los ordenadores de sus competidores. En el escándalo están implicados, entre otros, el canal de televisión por satélite 'Yes', que podría haber espiado a la cadena de televisión por cable 'HOT', las compañías de teléfono 'Pelephone' y 'Cellcom', que podrían haber espiado a su rival mutuo 'Partner'; y 'Mayer' encargada de exportar a Israel vehículos Volvo y Honda que podría haber espiado a 'Champion Motors', exportadores de vehículos Audi y Volkswagen.

Un estudio realizado por Panda Software entre más de 650 empresas de todo el mundo, pone de manifiesto que las corporaciones son conscientes de la amenaza que el spyware representa para sus sistemas informáticos (de hecho, el 99% de ellas reconoce a este tipo de malware como peligroso para el correcto funcionamiento de los ordenadores). Sin embargo, el dato más preocupante es que, pese a tratarse de una grave amenaza, un número significativo de las empresas no saben qué medidas tomar para combatir el software espía.

El mismo estudio revela que el 53% de las empresas han sufrido ataques por parte de ejemplares de spyware en alguna ocasión. Por su parte, el 74% de las empresas conoce los efectos directos del spyware: el robo de datos relacionados con los hábitos de navegación de los usuarios.

Por último, entre las medidas que las empresas consideran adecuadas para protegerse del spyware figuran la utilización de un antivirus actualizado (70%) y/o la instalación de una herramienta antispyware específica.

Según Luis Corrons, director de Pandalabs, es muy significativo el hecho de que casi todas las empresas encuestadas se encuentren concienciadas sobre la amenaza que el spyware supone. Sin embargo, el hecho de buena parte de ellas no sepan que medidas tomar ante este malware debe ser un motivo de preocupación. Asimismo, aunque el resto de encuestados dicen saber que tipo de herramientas de seguridad emplear, vemos que confían únicamente en las tecnologías reactivas, como son los antivirus tradicionales y las aplicaciones específicas anti-spyware. Sin embargo, dada la gran proliferación de software espía que estamos viviendo actualmente, se hace también necesario el empleo de tecnologías capaces de detectar ejemplares desconocidos. De esta manera, se evita que un software espía de muy reciente aparición, o que aún no haya sido identificado por las empresas fabricantes de antivirus, pueda instalarse en el sistema.

2) La becaria espía de Valeo (Fuente: Expansión - 04.05.05)

Una joven china ha sido detenida en Francia por posible delito de espionaje industrial, mientras hacía prácticas en el fabricante de componentes de automoción

La policía francesa ha detenido a una estudiante china de 22 años por supuesto espionaje industrial mientras realizaba prácticas en Valeo, uno de los mayores fabricantes europeos de componentes de automoción. El arresto podría ensombrecer la visita a París esta semana del ministro de Comercio de China.

Li Li ha sido detenida después de que la policía registra su casa el pasado viernes y encontrara varios disquetes y diverso material que contiene información secreta y confidencial sobre Valeo y sus productos. La fiscalía asegura que los disquetes incluyen información "confidencial" sobre diseños de automóviles que todavía no están en el mercado.

La noticia de la detención de la becaria, que fue ofrecida en exclusiva por el diario francés Liberation, es un revés embarazoso para las intenciones del Gobierno Chino de estrechar los lazos comerciales con Francia. A este suceso puntual se suma la creciente preocupación política por el impacto del aumento de las importaciones de productos textiles chinos más baratos que los facturados en Francia y en el resto de Europa.

Bo Xilai, el ministro de Comerció de China, de visita en París debido a la celebración de una conferencia de ministros de la OCDE, afirmó ayer en rueda de prensa que ignoraba la detención de la joven estudiante Li." Si esto es verdad, es absolutamente rechazable", afirmó Xilai.

Espionaje

Yves Colleu, un fiscal del Estado en el juzgado de primera instancia de Versalles, en el que Li está arrestada, señalo: "Lo que nos preocupa es si ella ya ha enviado esta información o planeaba hacerlo, a una tercera parte". Colleau añadió que el juzgado ha designado a un magistrado para que examine el caso.

Todavía no está claro si la joven estudiante actuaba sola o en nombre de una empresa o Gobierno, indicó Colleu. "los estudiantes son objetivos potenciales de grupos industriales para manejarlos como espías", añadió.

Brillante estudiante

El fiscal dijo que Li, que está licenciada en Ingeniería, Matemáticas y Física, y que estaba de prácticas en Valeo, ha rechazado los cargos de robo de información, entrada ilegal en un sistema informático ajeno y ruptura de confianza. Colleau afirmó que la becaria indicó que ella copió erróneamente la información tras reemplazar el disco duro del ordenador que le suministró.

Li Li, que fue calificada por fuentes de Valeo como una estudiante "brillante", cursó sus estudios durante tres años en una Universidad al norte de París, y estaba completando unas prácticas laborales en el fabricante de componentes. La joven china se benefició de un acuerdo entre la universidad y Valeo.

Tras comenzar a trabajar en Valeo el pasado febrero en su división de aire acondicionado, Li levantó rápidamente sospechas entre los directores de personal de la empresa por las muchas horas que pasaba trabajando con los ordenadores. Valeo presentó una denuncia la semana pasada por el caso en el juzgado de Versalles.

Plan de negocio

Valeo afirmó recientemente su deseo de crecer en el mercado chino tras firmar un acuerdo para crear una sociedad mixta en el segmento de aire acondicionado en la ciudad de Changchun. El consejero delegado de la sociedad, Thierry Morin, dijo la semana pasada que prevé aumentar sus ventas en China de 250 millones a 1.300 millones de euros en 2010 a través de acuerdos locales.

3) Las nuevas tecnologías multiplican la deslealtad (Fuente: La Gaceta - 30.09.04)

La propiedad intelectual se usa para crear empresas paralelas

Las infracciones que cometen los empleados en España a través de Internet se han incrementado el pasado año casi el 50%, según un estudio del bufete Landwell, filial de PricewatherhouseCopers, a partir del análisis de casi 400 casos y sentencias judiciales. Aunque la pérdida económica por este tipo de delitos no suele ser catastrófica (la media no supera los 60.000 euros), su generalización y acumulación si está poniendo en jaque a muchas compañías. A pesar de ello, sólo una cuarta parte de las infracciones acaban en los tribunales y, el resto, se resuelven con acuerdos.

La infracción más habitual es el uso de la red corporativa para intercambiar música, películas o software a través de las redes peer to peer (P2P) accesibles a través de Internet. Durante los dos últimos años el número de usuarios de estas redes de intercambios ha aumentado de tal manera que "es difícil encontrar una empresa que no tenga instalado un programa de ese tipo", según el citado informe.

Una adecuada configuración del firewall pude impedir el uso de programas P2P como Kazaa eKonkey o eMule, pero las nuevas versiones permiten acceder a la red a través de puertos de comunicación no bloqueados por la empresa.

Un delito habitual es la explotación de la propiedad intelectual de la empresa paralela de nueva creación. Según el informe, éste se ha convertido en un problema habitual en las empresas que concentran las actividades de investigación y desarrollo en equipos muy reducidos o unipersonales "Cuando la tecnología está controlada por pocas personas, existe el riesgo de que minusvaloren el papel de la empresa en la creación del producto y decidan explotarlo por su cuenta, o con la ayuda de un inversor externo". Este riesgo puede minimizarse con una adecuada segregación de las tareas de un proyecto y con cláusulas penales disuasorios.

También se está generalizando la utilización del correo electrónico corporativo para enviar mensajes amenazantes, injuriosos o calumniosos.

Confidencial

El informe también destaca el crecimiento de la revelación de información confidencial de forma no intencionada, sino debido a la proliferación de programas espías (spyware) que se instalan en el ordenador del usuario cuando navega por Internet.

En el 55% de los casos, el ánimo de lucro es la motivación fundamental que lleva a los trabajadores a cometer este tipo de infracciones, aunque el 41% de los empleados lo que buscan es vengarse ante lo que consideran un despido injusto. De hecho, los conflictos laborales son la principal causa de los sabotajes y de la introducción de virus.

La mayoría de las infracciones (36%) son cometidas por los usuarios generales de la red de la empresa, aunque suelen ser los empleados que mejor dominan las tecnologías los más proclives a cometer este tipo de delitos: por ejemplo, los analistas o programadores (31%) y el administrador de sistemas (9%). También se destacan los representantes de los trabajadores (12%) en los actos de sabotaje.

Por sectores, las empresas más afectadas por este tipo de deslealtades son las de desarrollo de software (55%), las de servicios a empresas (28%) seguros (7%) y laboratorios farmacéuticos (7%).

4) Twitter sucumbe a un ataque informático (Fuente: http://www.elpais.com/ -07/08/2009)

La red social deja 'colgados' a millones de usuarios tras cerrar su servidor durante más de tres horas

Un ataque informático provocó ayer la caída de la popular red social de Internet Twitter. Durante más de tres horas la página web dejó de dar servicio a sus millones de usuarios de todo el mundo. Según el blog de la compañía, Twitter había sido víctima de un ataque pirata que enviaba peticiones masivas a su servidor. Esto les obligó al cierre. Al final de la tarde, y con bastante lentitud, la página volvió paulatinamente a la normalidad. "Estamos trabajando para tener el servicio al 100% tan pronto como nos sea posible", aseguraron desde la firma.

"Nos defenderemos", aseguraba ayer uno de sus creadores y directivos, Biz Stone, a los fieles seguidores de su blog en plena vorágine electrónica. Admitía así que por primera vez el popular sistema es vulnerable. El ataque de los hackers (piratas informáticos) comenzó en torno a las 9.00 en la costa atlántica de EE UU (15.00, hora peninsular), en forma de un bombardeo electrónico masivo lanzado desde varios ordenadores y dirigido hacia un servidor para interrumpir el tráfico electrónico.

Debido a la lentitud del servicio, y para preservar a los usuarios, Twitter cerró a cal y canto su portal. Según el blog de Biz Stone, la caída se debe a un ataque masivo de peticiones a su servidor que lo bloquea hasta impedir el servicio. Stone indicaba que este tipo de ataques son comunes en pasarelas de pago o cuando se intenta atacar a bancos en la Red, pero no en servicios dedicados a la comunicación, como es su caso.

Biz Stone es una de las figuras más cotizadas del momento en el universo tecnológico por la expansión espectacular de esta comunidad electrónica.

Hace unas semanas, los hackers robaron documentos internos de Twitter. Aunque Stone dejó claro que este ataque no tiene que ver con aquel incidente. "Es por pura saturación", reiteró. Este bombardeo suele realizarse infectado miles de ordenadores con virus informáticos, que lanzan peticiones de información a un servidor. El pasado fin de semana, la web de Gawker estuvo bloqueada por este motivo.

La red social Live Journal también fue víctima ayer de un bombardeo electrónico, y Facebook parecía experimentar problemas en el tráfico normal hacia sus servidores. Por eso fueron analizados por sus ingenieros. Hace unas semanas, los portales de la Casa Blanca, la Reserva Federal, el Pentágono o la Bolsa de Nueva York también sufrieron ataques, pero sin mayores consecuencias.

Mientras Twitter trataba de recuperarse del incidente, el apagón contribuía a echar más leña al debate sobre la seguridad de este tipo de redes sociales, a las que millones de internautas confían sus datos y experiencias personales. En Wall Street no tardaron en hacerse sentir los abogados que aconsejaban a sus clientes corporativos no utilizar Twitter para intercambiar información dentro o fuera de sus trabajos. Se evitarían así, afirmaron, los problemas legales.

5) Sony confirma haber sido objeto de un nuevo ataque informático (Fuente: http://chismososdetv.blogspot.com/ - 03/06/2011)

Nueva York. El grupo japonés Sony confirmó el viernes en un comunicado haber sido objeto de un nuevo ataque informático a través de su filial Sony Pictures.

"El cibercrimen que afectó a Sony y a cierto número de agencias de gobierno, de empresas y de individuos durante los últimos meses afectó asimismo a Sony Pictures. Ayer (jueves) a la tarde un grupo de piratas informáticos llamado 'LulzSec' afirmó haber penetrado algunos de nuestros sitios", indicó Sony Pictures en un comunicado.

"Tuvimos la confirmación de que tuvo lugar un ataque y hemos tomado medidas contra intrusiones ulteriores sobre nuestros sitos", agregó el comunicado, precisando que el grupo llamó a un equipo de expertos para analizar el ataque, un trabajo que aún está en curso.

Un grupo de piratas informáticos afirmó el jueves haber robado más de un millón de palabras clave, identificaciones de mensajería electrónica y otros datos al sitio Sony Pictures.com unas semanas después de un ataque similar conducido contra el gigante japonés.

El ataque fue reivindicado por piratas que se hicieron llamar "Lulz Security" en su cuenta twitter @LulzSec.

6) "Ingenioso" ataque informático accede a datos del FMI (Fuente:New York Times - 12/06/2011)

El Fondo Monetario Internacional (FMI) fue víctima de un "complejo" ataque cibernético recientemente, según ha revelado este sábado el diario The New York Times. El rotativo asegura el organismo comunicó el miércoles a su personal y a su junta directiva que era sujeto de un "complejo" ataque de piratas informáticos "cuyas dimensiones aún se desconocen".

El ataque no motivó ningún anuncio público por parte de la institución, que atraviesa un momento delicado ya que está pendiente de seleccionar a un nuevo director gerente tras el arresto de Dominique Strauss-Kahn el mes pasado en Nueva York, acusado de un delito sexual.

El ataque informático que sufrieron durante varios meses los servidores empleados por el organismo tenía como objetivo la instalación de un software que diera a un Estado "presencia digital interna" en la organización, según ha asegurado el experto en ciberseguridad Tom Kellerman, que ha trabajado para el FMI y el Banco Mundial.

"Fue un ataque dirigido", ha señalado Kellerman, que conoce la arquitectura de ambas instituciones financieras y que trabaja en la junta directiva de un grupo conocido como la Alianza Internacional de Ciberseguridad. "El código fue desarrollado y enviado con ese propósito", ha recalcado. El FBI también se ha unido a la investigación sobre el caso, según ha anunciado a Reuters la portavoz del Pentágono, April Cunningham.

Varios funcionarios de alto rango indicaron al diario neoyorquino que el ataque cibernético fue "serio y complejo", aunque no han revelado de qué país procedía. "Se trata de una brecha [de seguridad] muy grande", dijo uno de los funcionarios, quien precisó que había sucedido a lo largo de los últimos meses, incluso antes del arresto de Strauss-Kahn.

Preguntado sobre el asunto en la noche del viernes, un portavoz del FMI, David Hawley, dijo a The New York Times que el organismo estaba "investigando el incidente" y destacó que "el Fondo opera plenamente", sin ofrecer más detalles.

El FMI ha estado al frente de los programas de rescate para Portugal, Grecia e Irlanda, y tiene en su poder información sensible sobre otros países que también se encuentran al borde de una crisis, así como otros datos capaces de afectar a los mercados. La institución también tiene información sobre las negociaciones de rescates financieros que, según indicó al diario un funcionario, son "dinamita" en el mercado y en muchos países.

Aunque se desconoce cuántos y qué datos han resultado afectados, el incidente ha causado preocupación. El Banco Mundial, cuya sede se encuentra frente a la del FMI, desactivó por "precaución" el sistema que permite a ambos instituciones el intercambio de información, siempre según el periódico estadounidense.

El pasado 1 de junio, el grupo de ciberactivistas Anonymous anunció en Twitter la "Operación Grecia", en la que invitaba a atacar la página web del FMI por su desacuerdo con las condiciones del rescate a la República Helénica, aunque no precisó cuándo se realizarían esos ataques.

El mensaje del grupo remitía a una web en la que se critican las condiciones del plan de austeridad impuesto por el FMI y la Unión Europea a cambio de un paquete de rescate que asciende a 110.000 millones de euros. En esa ocasión, el organismo económico dijo estar al tanto de las amenazas del grupo Anonymous y aseguró que tomaría "las medidas apropiadas". (ElPais.com)

7) Los ataques informáticos más peligrosos para el bolsillo (Fuente: http://www.portafolio.co – 21/07/2011)

El objetivo principal de los ciberpiratas son las cuentas y los datos bancarios.

Según un estudio de la firma Gemalto, especializada en seguridad, se crearon diariamente un promedio de 73.000 ejemplares de malware (software malintencionado que roba datos) y, en comparación con 2010, los ciberdelincuentes han lanzado un 26 por ciento más de nuevas amenazas informáticas.

Los criminales de Internet son cada vez más agresivos y ya han escogido sus canales de ataque: el 'hackeo' de tarjetas bancarias; el spam (correo basura), por ejemplo, el que

invita a ayudar a Japón; y el robo interno de datos a empresas de seguridad son algunos ejemplos.

El sistema operativo Android, para celulares y tabletas; la red social Facebook y el grupo de personas conocido como Anonymus, por su parte, son protagonistas de los incidentes de seguridad más graves; aunque surge OddJob, un revolucionario virus troyano especializado en atacar entidades bancarias, con gran capacidad para robar datos de clientes.

100 DÓLARES POR 'HACKEAR' UNA TARJETA BANCARIA

La cuota que pide un pirata informático para robar la información de una tarjeta de crédito puede llegar a los 100 dólares, dependiendo de la dificultad del ataque, según cifras de Symantec, la empresa número uno en seguridad informática del mundo.

Las vacaciones y el periodo navideño son las dos épocas preferidas para los 'hackers', debido a que las transacciones aumentan de manera considerable en estos momentos del año.

Para evitar que sea víctima de los ataques, tenga en cuenta estos consejos:

- Revise los estados de cuenta bancarios para detectar irregularidades.
- No tire a la basura sin destruir recibos bancarios, facturas y demás papeles que tenga su información bancaria.
- No acepte llamadas telefónicas para renovar sus tarjetas bancarias.
- Revise una vez al año su estado en las centrales de riesgo (como Datacrédito) para que verifique que no se haya presentado un robo de identidad a nombre suyo y que probablemente haya hecho que tales entidades lo reporten a usted como deudor moroso.

ODDJOB, EL NUEVO TROYANO

Este virus, enfocado a atacar entidades bancarias, fue detectado la empresa Trusteer, es considerado el más peligroso hasta ahora.

Lo más grave es que les permite a los delincuentes acceder a una cuenta sin necesidad de robar los datos de acceso.

Este virus le da la posibilidad al delincuente de compartir la sesión con la víctima; es así como obtiene toda la información sobre su cuenta e identidad.

Pero lo más preocupante es que al intentar salir, bloquea el cierre de la sesión sin que el usuario lo sepa, momento en el cual el delincuente vaciará la cuenta. Los países afectados hasta ahora son EE. UU., Polonia y Dinamarca.

ATAQUES DE SPAM

En el 2010, la mayor parte del spam (74 por ciento) estuvo ligado a promociones de productos farmacéuticos.

En lo que va de este año, la tendencia parece estar centrada en temáticas amarillistas, por ejemplo:

- Enlaces y videos sobre noticias amarillistas de muertes notorias y conflictos, y otras de actualidad en las redes sociales Twitter y Facebook, ya que circulan como spam cargado de virus.

- La catástrofe natural de Japón aún está generando oleadas de robos, enlaces y archivos maliciosos distribuidos por spam, buscadores y desde Facebook y Twitter.

ENTRETENIMIENTO, GANCHO PARA ATACAR

Las personas siguen siendo víctimas de un enlace que promete ver el estreno de una película, el álbum musical más reciente, un video 'espectacular' o bajar el último capítulo de alguna popular serie de TV.

El consejo es no dar clic sobre enlaces desconocidos, no bajar actualizaciones o programas adicionales, y nunca descargar aplicaciones o registrarse en formularios o sitios de dudosa procedencia, pues a menudo son registros falsos para obtener información.

Según datos de la firma PandaLabs sobre los intentos fraudulentos más usados, el 25 por ciento utiliza su URL para la descarga de video y multimedia; el 21,63 por ciento utiliza instaladores o actualizaciones de programas; y el 16 por ciento son direcciones dentro de redes sociales.

RÁNKING DE MALWARE

Los tres códigos más detectados en marzo, según los datos estadísticos proporcionados por el servicio Threatsense.Net, de la compañía Eset, son Win32/PSW.OnLineGames.OUM, INF/Autorun y Win32/Conficker, troyanos orientados a capturar contraseñas de juegos online, y Conficker; este tipo de software maligno sigue ocupando las primeros puestos del ránking, junto con el veterano del malware, Agent.

ATAQUES EN FACEBOOK

El auge de la red social más famosa del mundo también la hace una de las más vulnerables en seguridad. Cada día se encuentran más agujeros en esta red, además de que es el canal favorito para la distribución de 'malware' y 'phishing' (suplantación de identidad). Por eso, tenga en cuenta lo siguiente:

- Si le llega un mail informándole del cambio de contraseña en su cuenta, no lo abra, contiene un archivo que descarga el troyano Oficla.

- Aquellos que se han dado de alta con su cuenta de correo electrónico en Hotmail pueden estar comprometidos. Cambie su contraseña de Facebook. Aunque la red social anunció que se ha solucionado el problema, su cuenta puede haber sido robada.

- Tenga cuidado con el virus de la "chica que se suicidó por webcam".

Este lleva circulando desde marzo y roba las claves de acceso a todos aquellos que abren el video o uno de sus enlaces.

- Aléjese de las 'seductoras' campañas que ofrecen aplicaciones para Facebook. Aparecen cada semana, son falsas y prometen algo que no está permitido; roban su información con cuestionarios y lo infectan al descargarlas a su PC. También

circula una oferta maliciosa para bajar una aplicación Facebook Profile Creeper Tracker / Profile Creeps, para ver quién visita tu perfil.

VULNERABILIDAD DE ANDROID; SKYPE, EN LA MIRA

Las aplicaciones para el sistema operativo Android siguen siendo vulnerables. Se detectan 58 aplicaciones maliciosas en Android Marketplace, que Google aún está quitando de su tienda y de las 260.000 unidades que se calculan infectadas.

Ahora es el turno de Skype para Android, ya que se ha comprobado que es vulnerable a maliciosos usuarios o aplicaciones que podrían acceder a su información almacenada.

¿Quieres ver un esquema resumen (más completo) de pasos y tipos de ataques?

a) pasos de intrusiones:

- Descubrimiento de direcciones.
- Descubrimiento de sistemas y servicios.
- Descubrimiento de vulnerabilidades.
- Descubrimiento de cuentas y contraseñas.
- Descubrimiento de topologías.

HASTA AQUÍ SUELE LLAMARSE → FINGERPRINTING (Huella digital)

- Aprovechamiento de vulnerabilidades (Exploits).
- Infecciones.
- Saltos.
- Empleo.
- Borrado de huellas y puertas traseras.

b) tipos de ataques:

- Por su metodología:
  - ataques pasivos
  - ataques activos
- Por su origen:
  - Interno
  - Externo
- Por su efecto:
  - Negación de Servicio (puntual o distribuido).
  - Obtención de Información.

Inserción o suplantación de Información.

Posesión de sistemas.

Ataques de correo (Spam, bombardeo de mail, virus).

- Por su Target (Objetivo):

  Amenazas en red

  Amenazas en sistemas

Empleo práctico de estos ataques:

- Barrido de máquinas (Relevamiento)
- Negación de Servicio
- Negación de Servicio distribuido
- Bombardeos de mail.
- Aprovechamiento de bugs (Exploits)
- Obtención de Información.
- Posesión de sistemas (Bouncers).
- Escucha de tráfico.
- Cracking de password
- Ingeniería Social
- Scanner de puertos.
- Virus
- Gusanos
- Troyanos
- Modificación de mensajes
- Evasión de detecciones (o borrado de huellas).
- Divulgación de contenidos.
- Spam
- Phishing

¿Quieres ver el detalle de los pasos de un ataque real del gusano SLAMMER?

Se presenta la captura de una infección a través del gusano SLAMMER. Como se puede apreciar, la infección se hace a través del puerto UDP 1434, y el código comienza a partir del octeto 04 en hexadecimal y tiene una longitud de 376 octetos:

1 10:58:38.691 00D059322092 00065BA7C216 UDP Src Port: Unknown, (1051); Dst Port: Unknown (1434); Length = 384 10.64.130.201 10.64.131.205 IP

```
FRAME: Base frame properties
ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
IP: ID = 0x17D9; Proto = UDP; Len: 40

   IP: Version = 4 (0x4)
   IP: Header Length = 20 (0x14)
     • IP: Service Type = 0 (0x0)
   IP: Total Length = 404 (0x194)
   IP: Identification = 6105 (0x17D9)
     • IP: Flags Summary = 2 (0x2)
   IP: Fragment Offset = 0 (0x0) bytes
   IP: Time to Live = 64 (0x40)
   IP: Protocol = UDP - User Datagram
   IP: CheckSum = 0x066A
   IP: Source Address = 10.64.130.201
   IP: Destination Address = 10.64.131.205
   IP: Data: Number of data bytes remaining = 384 (0x0180)
UDP: Src Port: Unknown, (1051); Dst Port: Unknown (1434); Length = 384
(0x180)

00000:  00 06 5B A7 C2 16 00 D0 59 32 20 92 08 00 45 00   ..[.....Y2 ...E.
00010:  01 94 17 D9 40 00 40 11 06 6A 0A 40 82 C9 0A 40   ....@.@..j.@...@
00020:  83 CD 04 1B 05 9A 01 80 16 1A 04 01 01 01 01 01   ................
00030:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
00040:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
00050:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
00060:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
00070:  01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01   ................
00080:  01 01 01 01 01 01 01 01 01 01 01 DC C9 B0 42 EB   ..............B.
00090:  0E 01 01 01 01 01 01 01 70 AE 42 01 70 AE 42 90   ........p.B.p.B.
000A0:  90 90 90 90 90 90 90 68 DC C9 B0 42 B8 01 01 01   .......h...B....
000B0:  01 31 C9 B1 18 50 E2 FD 35 01 01 01 05 50 89 E5   .1...P..5....P..
000C0:  51 68 2E 64 6C 6C 68 65 6C 33 32 68 6B 65 72 6E   Qh.dllhel32hkern
000D0:  51 68 6F 75 6E 74 68 69 63 6B 43 68 47 65 74 54   QhounthickChGetT
000E0:  66 B9 6C 6C 51 68 33 32 2E 64 68 77 73 32 5F 66   f.llQh32.dhws2_f
000F0:  B9 65 74 51 68 73 6F 63 6B 66 B9 74 6F 51 68 73   .etQhsockf.toQhs
00100:  65 6E 64 BE 18 10 AE 42 8D 45 D4 50 FF 16 50 8D   end....B.E.P..P.
00110:  45 E0 50 8D 45 F0 50 FF 16 50 BE 10 10 AE 42 8B   E.P.E.P..P....B.
00120:  1E 8B 03 3D 55 8B EC 51 74 05 BE 1C 10 AE 42 FF   ...=U..Qt.....B.
00130:  16 FF D0 31 C9 51 51 50 81 F1 03 01 04 9B 81 F1   ...1.QQP........
00140:  01 01 01 01 51 8D 45 CC 50 8B 45 C0 50 FF 16 6A   ....Q.E.P.E.P..j
00150:  11 6A 02 6A 02 FF D0 50 8D 45 C4 50 8B 45 C0 50   .j.j...P.E.P.E.P
00160:  FF 16 89 C6 09 DB 81 F3 3C 61 D9 FF 8B 45 B4 8D   ........<a...E..
```

Si se presta atención, la trama de infección fue enviada a las 10:58:38.691 horas, **UN MILISEGUNDO después** la víctima comenzó a generar tráfico pseudoaleatorio en la red tratando de infectar a cualquier otro SQL que encuentre sin parchear.

A continuación se presentan las primeras tramas generadas, para poner de manifiesto la velocidad con la que es infectada y la tasa promedio de transmisión, pues lo hace a razón de tres tramas por milisegundo, es decir 3000 tramas por segundo.

Si se tiene en cuenta que la longitud es de 384 Byte, y a esto se le suma los 20 del encabezado IP, más los 18 de Ethernet, y los 8 de preámbulo e inicio, hace un total de

430 Byte = 3.440 bit, por lo tanto: 1 seg → 3.000 tramas, cada una de 3.440 bit → 3.440 * 3000 = **10.320.000 bps**

Este tráfico quedó registrado en los analizadores de protocolos y dejó la red totalmente fuera de servicio CON UNA SOLA MAQUINA INFECTADA.

Se presentan las primeras tramas:

1 **10:58:38.691** 00D059322092 00065BA7C216 UDP Src Port: Unknown, (1051); Dst Port: (1434); Length = 384 10.64.130.201 10.64.131.205
(LA TRAMA NÚMERO UNO ES LA DE LA INFECCIÓN)

2 **10:58:38.692** 00065BA7C216 USC 01A38A UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 235.1.163.138
3 **10:58:38.692** 00065BA7C216 USC 615865 UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 225.97.88.101
4 **10:58:38.693** 00065BA7C216 USC 5A35AD UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 229.90.53.173
5 **10:58:38.693** 00065BA7C216 USC 587FC4 UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 238.216.127.196
-6 **10:58:38.693** 00065BA7C216 USC 069AF7 UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 234.134.154.247
7 **10:58:38.694** 00065BA7C216 USC 256722 UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 236.37.103.34
8 **10:58:38.694** 00065BA7C216 USC 66C385 UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 237.230.195.133
9 **10:58:38.694** 00065BA7C216 USC 1BDBDA UDP Src Port: Unknown, (2359); Dst Port: (1434); Length = 384 10.64.131.205 227.27.219.218
10 **10:58:38.695** 00065BA7C216 USC 126ADA UDP .....

8.4. Auditorías de seguridad

8.4.1. Lo que el cliente verdaderamente necesita.

La hipótesis que se presenta trata de reflejar una problemática de seguridad que se encuentra en exponencial crecimiento y con una vorágine tal que no permite a los responsables de sistemas de las "PyMEs", mantener personal al tanto de lo que sucede día a día. Es más, este hecho se podría considerar casi como asumido por esta línea de empresas, es decir, los gerentes de sistemas ya son plenamente conscientes que un alto nivel de capacitación en seguridad es una cuestión cara y cuya relación coste/beneficio, tiene un cierto límite marcado por el conocimiento básico de seguridad de sus administradores y un claro umbral, superado el cual (por situaciones puntuales o por periodicidad), se debe solicitar el apoyo externo.

Este apoyo externo, cada vez más frecuente (por la simple relación coste/beneficio planteada), se podría englobar en dos grandes causas:

- Problemas puntuales de seguridad: Cuando ocurren hechos que superan el conocimiento básico de sus administradores.
- Periódicos: Cuando se ha llegado a una situación que hace necesaria una cierta evaluación de alguna plataforma, un nuevo servicio o una "Cuantificación del nivel de riesgo"

Al solicitar este apoyo de consultoría a empresas o personas especializadas, es casi una norma general que le indiquen que el primer paso a seguir es la realización de una auditoría de seguridad. Este consejo puede considerarse válido, pues es muy difícil poder evaluar o tomar cualquier acción con escaso conocimiento de la infraestructura que se posee, pero aquí es donde hay que detenerse seriamente para plantear lo que el cliente verdaderamente necesita y cómo llevarlo a cabo.

El cliente necesita:

- Soluciones.
- Garantías.
- Índices (o parámetros).

.....y en ese orden.....

¡¡¡ Y NADA MÁS !!!, pues partimos de la hipótesis que no es un especialista en seguridad, y para eso confía en la empresa consultora.

- **Soluciones:** El cliente es consciente que tiene un problema, es muy frecuente que no tenga claro de qué se trata o de dónde proviene, pero está seguro que lo tiene. Independientemente de todo el trabajo de análisis, detección y evaluación que se realice, el resultado final del mismo debe proporcionar descripciones muy claras de cómo solucionarlo, pues caso contrario, no tendría sentido la totalidad del trabajo. Este punto es de vital interés, pues es difícil para el experto, bajar al nivel de alguien que no tiene por qué tener idea de seguridad y explicarle con todas las letras los pasos que debe seguir para solucionar el mismo.
- **Garantías:** Todo el trabajo que se realice debe culminar ofreciendo dos tipos de garantías:
  - Que se ha detectado la masa de los problemas de seguridad: Este aspecto no es trivial, pues acorde al tipo de trabajo que se realice y al tiempo dedicado, se podrá profundizar más o menos. Lo que no se puede dudar, es que durante el proceso de contratación, hay que hablar claro y dejar constancia de hasta dónde llegará el trabajo a realizar, pues no se puede aducir al finalizar esta actividad que determinadas actividades no se han realizado, o peor aún, dejar

dudas sobre el nivel de seguridad de su infraestructura, pues esa parte no se había contratado, etc...

- Que al aplicar las soluciones recomendadas, el nivel de riesgo se reduce a los índices deseados, o mejor aún, que lo que se propone es "la mejor solución" a sus problemas, pues es lo que recomiendan los especialistas del tema.

En definitiva, con garantías se quiere expresar que luego de la actividad que se realice, el cliente puede encarar las soluciones recomendadas, confiado en que es su mejor opción y que al aplicar las mismas, su nivel de riesgo ha mejorado sensiblemente.

NOTA: Una excusa "Omnipresente" y bastante desagradable para evadir garantías (aunque lamentablemente no deja de ser cierta), es que surgen nuevas vulnerabilidades día a día, por lo tanto una vez finalizada toda actividad, "No se puede garantizar la seguridad absoluta aplicando las soluciones propuestas", pues mañana habrá algo nuevo que afecte a la infraestructura.....Qué pena.....(Confianza, seriedad, sinceridad.....etc, etc, etc).

8.4.2. Indicadores o parámetros de seguridad.

Este especto, se desarrolla intentando "cuantificar" la forma en que deberíamos evaluar el estado de seguridad de un sistema, el cual podemos analizarlo desde varios aspectos, que presentamos a continuación.

- **Índices (o parámetros):** Desde mi enfoque personal, creo que uno de los problemas más grandes que tiene un gerente de sistemas es "Cuantificar la seguridad", pues como todos pueden apreciar es un "bien intangible". SE DEBE HACER TODO ESFUERZO POSIBLE PARA PONERLE NÚMEROS a la misma. Ya hay varias estrategias a seguir al respecto y en Internet se puede encontrar mucho de esto (ESPACIO PARA PUBLICIDAD: Recomiendo que mires un método que he propuesto hace tiempo que lo denominé "**Matriz de Estado de Seguridad**", está publicado en varias web. Su objetivo es aplicar todos los indicadores objetivos posibles, dejando de lado la subjetividad). El no contar con índices o parámetros, ocasiona dos grandes perjuicios:
  - Desconocimiento del grado de seguridad y de la evolución del mismo, no pudiendo plantear objetivos o umbrales a cumplir (¡¡muy negativo!!).
  - Imposibilidad de demostrar el ROI (Retorno a la Inversión) en temas de seguridad ante la dirección de la empresa (¡¡¡Catastrófico!!!).

Un trabajo de auditoría externa es la mejor oportunidad para cuantificar el nivel de seguridad, pues todo especialista en el tema posee la "expertiz" necesaria para jugar con ellos, promediando valores. Los parámetros más importantes a considerar son:

- **Criticidad:** Este parámetro refleja el daño que puede causar a ese sistema la explotación de esa vulnerabilidad por quien no debe.
- **Impacto:** Independientemente de la criticidad de una vulnerabilidad encontrada, esta puede causar daño a sistemas que son el sustento de la empresa, que mantienen datos de alta clasificación (Ley Orgánica de Protección de Datos: LOPD), o una fuerte pérdida de imagen de empresa, etc. O por el contrario, puede ser Crítica para ese servicio, pero el mismo no cobra mayor interés para el buen funcionamiento de la empresa.
- **Visibilidad:** Este indicador puede servir como multiplicador de los anteriores, pues no posee el mismo riesgo un servidor de Internet "Front End", que uno interno de la empresa con accesos restringidos.
- **Popularidad:** Este parámetro, si bien puede ser muy discutido, permite indicar el grado de "visitas, conexiones o sesiones" que posee un sistema. El empleo o no de este indicador permite considerar, que si existiere una vulnerabilidad sobre el mismo, se puede suponer que tiene mayo grado de "exposición" que el resto. Se admite aquí que puede ser valorado o no.
- **Magnitud:** Cuántos sistemas afecta. Este parámetro es muy interesante tenerlo en cuenta por niveles, es decir una misma plataforma puede estar conformada por varios sistemas, pero también existen sistemas que forman parte de varias plataformas, que permiten el paso hacia ellas, que autentican, que filtran, que monitorizan, etc. Es decir, hay plataformas cuya magnitud "Directa" es muy clara y dependen únicamente de ellas, pero hay otras que para su funcionamiento necesitan la participación de otros elementos. En concreto, una cadena se corta por el eslabón más débil, por lo tanto, si no se considera la magnitud de una infraestructura, y se solucionan o evalúan únicamente los aspectos puntuales de cada host, el resultado no es el óptimo.
- **Facilidad de explotación:** Una determinada vulnerabilidad, puede presentar desde la ejecución de una simple herramienta pública en Internet y desde allí mismo, hasta una elaborada técnica de intrusión, que conlleva amplios conocimientos y pasos por parte del ejecutante. En este valor entra también en juego el grado de visibilidad del sistema, el grado de segmentación interna, la autenticación, el control de accesos, etc.
- **Facilidad o coste de solución:** Este valor es muy subjetivo y debe ser evaluado con mucho cuidado pues es el punto de partida para planificar las soluciones. Se presentan aquí muchas combinaciones y a mi juicio es el parámetro que determina la "Expertiz" de una auditor, pues si realmente sabe, será capaz de interpretar con mayor claridad la problemática del cliente y proponerle un plan de acción "realista y eficiente" para los recursos del cliente. Se debe tener en cuenta aquí no solo la simple recomendación, sino como cada una de ellas puede o no ser aplicada (pues habrá aplicaciones o servicios que no lo permitan), puede involucrar a muchos dispositivos más, puede ocasionar actualizaciones de hardware y software, rediseños, caídas de sistemas, etc....
- **Tiempo de solución:** Es un parámetro muy relacionado con el anterior, y nuevamente dependerá de la "Expertiz" del auditor y de sus ganas de involucrarse con el cliente para tener en cuenta todos sus sistemas. Una de las mayores satisfacciones para el auditor (lo digo por experiencia propia) es poder entregarle al

cliente un cronograma de cómo emprender las soluciones, con todo el nivel de detalle posible (Gant, hitos, valores a alcanzar mes a mes, recursos, prioridades, objetivos, etc), si se llega a esto es porque el auditor, se ha involucrado en tal medida con la empresa, que conoce hasta el último detalle a considerar para poder estimar este "Project". Esto para el cliente es lo máximo que puede desear, pues le permite organizar su plan de acción, presentarlo a la dirección de la empresa y acorde a los recursos que obtenga, definirá su estrategia al corto/medio plazo, para cumplir las acciones aceptadas. Se debe considerar también que es la mejor forma que posee la dirección de realizar el seguimiento del dinero que invirtió, pues los hitos serán todo lo claros que hagan falta. Este aspecto sin lugar a dudas, si se hacen bien las cosas, incrementará la confianza y los fondos que la gerencia informática tendrá para el próximo ejercicio.

Por último, relacionado a las métricas de seguridad, es muy interesante la lectura del modelo que propone NIST a través del documento *"Security Metrics Guide for Information Technology Systems"*, el mismo desarrolla una métrica de seguridad basada en el alcance de objetivos y metas, lo que se plasma en resultados, de forma muy precisa. El mismo puede ser descargado en: http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf

8.4.3. Cómo se puede clasificar lo que habitualmente se engloba bajo "Auditorías de seguridad".

Para tratar de diferenciar bien las opciones que se puede tener en cuenta a la hora de solicitar este tipo de actividades, se deben considerar al menos tres grandes grupos:

- Penetration Test: Se trata de una actividad con objetivo específico y acotado empleando técnicas de hacking y en general aplicando metodologías de "Caja Negra" (Sin ningún tipo de información, mas allá de la que puede contar cualquier individuo ajeno a la empresa)

  Según la definición de "**Open Source Security Testing Methodology Manual**" (OSSTMM): *Test de seguridad con un objetivo definido que finaliza cuando el objetivo es alcanzado o el tiempo ha terminado*.

- Diagnóstico o evaluación de Seguridad: Comprende una actividad más amplia, en general tanto desde fuera como desde dentro de la empresa, siempre relacionado a actividades eminentemente técnicas, y puede realizar por medio de "Caja Negra o Blanca" (con total conocimiento de la información de la empresa)

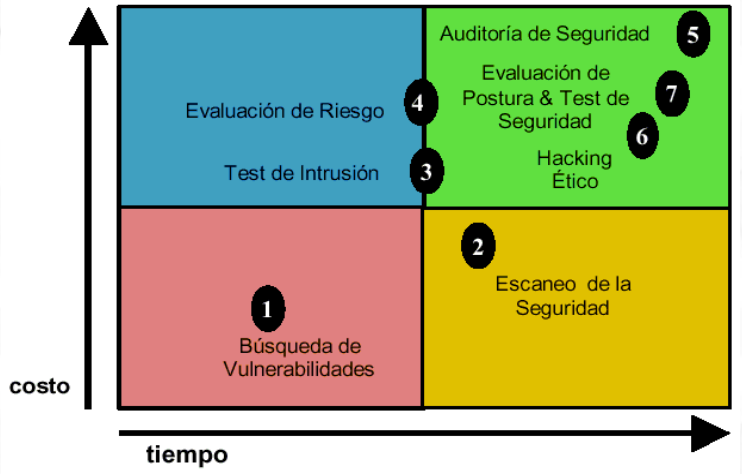
  Según la definición de OSSTMM: *Una visión general de la presencia de seguridad para una estimación de tiempo y horas hombre*.

- Auditoría de seguridad: Comprende a lo anterior y también una visión más amplia en cuanto a planes y políticas de seguridad, revisión de normativas, aplicación de LOPD, procedimientos, planos, inventarios, audiencias, etc. Es decir involucra la visión más amplia a considerar, sin dejar ningún aspecto librado al azar.

> Según la definición de OSSTMM: *Inspección manual con privilegios de acceso del sistema operativo y de los programas de aplicación de un sistema. En los Estados Unidos y Canadá, "Auditor" representa un vocablo y una profesión oficiales, solamente utilizado por profesionales autorizados. Sin embargo, en otros países, una "auditoría de seguridad" es un término de uso corriente que hace referencia a Test de Intrusión o test de seguridad.*

Para completar un poco el enfoque que ISECOM (Institute for Security and Open Methodologies), entidad responsable del proyecto OSSTMM, ofrece sobre estas clasificaciones se presenta a continuación una gráfica que simboliza con mayor detalle las actividades que pueden ser llevadas a cabo (La misma se presenta textualmente figura en sus manuales).

ISECOM aplica los siguientes términos a los diferentes tipos de sistemas y de testeos de seguridad de redes, basados en tiempo y costo para el Testeo de Seguridad de Internet:



1. **Búsqueda de Vulnerabilidades:** se refiere generalmente a las comprobaciones automáticas de un sistema o sistemas dentro de una red.
2. **Escaneo de la Seguridad:** se refiere en general a las búsquedas de vulnerabilidades que incluyen verificaciones manuales de falsos positivos, identificación de los puntos débiles de la red y análisis profesional individualizado.
3. **Test de Intrusión:** se refiere en general a los proyectos orientados a objetivos en los cuales dicho objetivo es obtener un trofeo, que incluye ganar acceso privilegiado con medios pre-condicionales.
4. **Evaluación de Riesgo:** se refiere a los análisis de seguridad a través de entrevistas e investigación de nivel medio que incluye la justificación negocios, las justificaciones legales y las justificaciones específicas de la industria.
5. **Auditoría de Seguridad:** hace referencia a la inspección manual con privilegios administrativos del sistema operativo y de los programas de aplicación del sistema o sistemas dentro de una red o redes.

6. **Hacking Ético**: se refiere generalmente a los tests de intrusión en los cuales el objetivo es obtener trofeos en la red dentro del tiempo predeterminado de duración del proyecto.
7. **Test de Seguridad y su equivalente militar, Evaluación de Postur**a, es una evaluación de riesgo con orientación de proyecto de los sistemas y redes, a través de la aplicación de análisis profesional mediante escaneos de seguridad donde la intrusión se usa generalmente para confirmar los falsos positivos y los falsos negativos dentro del tiempo permitido de duración del proyecto.

8.4.4. ¿Es posible respetar algún método que permita repetir esta tarea y obtener índices de evolución?

En este punto es donde se plantea a título de guía lo que propone OSSTMM. Se reitera, no porque sea mejor o peor que cualquier otra que pueda emplear una empresa consultora, sino simplemente por ser una referencia gratuita y sobre todo porque tiene su punto de partida en respetar la mayoría de los estándares, tal cual lo expresa en sus primeras páginas, estando en plena conformidad con los mismos (ISO-17799 o BS-7799, GAO y FISCAM, NIST, CVE de Mitre, etc.).

Resumidamente, esta metodología propone un proceso de evaluación de una serie de áreas que reflejan los niveles de seguridad que posee la infraestructura a auditar, a estos los denominará "Dimensiones de seguridad", y consisten en el análisis de lo siguiente:

- Visibilidad.
- Acceso.
- Confianza.
- Autenticación.
- No repudio.
- Confidencialidad.
- Privacidad.
- Autorización.
- Integridad.
- Seguridad.
- Alarma.

Para un trabajo metódico y secuencial, describe seis secciones que abarcan el conjunto de los elementos que componen todo sistema actual, ellas son:

1 Seguridad de la Información

2 Seguridad de los Procesos

3 Seguridad en las tecnologías de Internet

4 Seguridad en las Comunicaciones

5 Seguridad Inalámbrica

6 Seguridad Física

En cada sección se especifican una serie de módulos a ser evaluados, teniendo en cuenta si aplica o no cada uno de ellos a la infraestructura en cuestión, el resultado de la observación de todos ellos es lo que permitirá "pintar" el mapa de seguridad.

Otro aspecto que trata con bastante detalle es la **Evaluación de riesgo**, teniendo en cuenta que dentro de cada módulo se encuentran los valores adecuados (RAVs: Risk Análisis Values) para obtener las métricas finales, lo cual como se recalcó en este texto es uno de los principios que debe tener en cuenta todo auditor si es consciente de las necesidades del cliente.

Al final de este manual, se ofrece el formato de todas las plantillas que pueden ser necesarias durante la auditoría, muchas de las cuales pueden no ser cumplimentadas en virtud de que no apliquen al sistema en cuestión, pero lo verdaderamente importante es que las que SI apliquen, proporcionan un verdadero estándar abierto, para que cuando sea necesario repetir cualquier aspecto de este trabajo se posea una

Referencia clara, para que cualquier otra persona pueda evaluar y tomar como punto de partida de un nuevo análisis, el cual si respeta estos formatos será un claro índice de evolución en ese aspecto. Este tipo de acciones y sobre todo cuando son abiertas, es una de las cosas que más valoro en todo sistema informático, pues le dejan total libertad de acción a su verdadero dueño (el cliente), para tomar la decisión que más le guste a futuro, sin ningún tipo de compromiso u obligatoriedad de caer nuevamente en manos de la empresa anterior, la cual si fue de su agrado podrá hacerlo y sino no. (Bendito software libre!!!!).

Por último presenta la Licencia de Metodología Abierta (OML), cuyas líneas introductorias deseo expresarlas textualmente:

PREÁMBULO

"Una metodología es una herramienta que detalla QUIÉN, QUÉ, CUÁL Y CUÁNDO. Una metodología es capital intelectual y está a menudo enérgicamente protegido por instituciones comerciales. Las metodologías abiertas son actividades comunitarias que transforman todas las ideas en un solo documento de propiedad intelectual que está disponible sin cargo para cualquier individuo.

Con respecto a la GNU General Public License (GPL), esta licencia es similar con la excepción del derecho de los desarrolladores de software a incluir las metodologías abiertas que están bajo esta licencia en los programas comerciales. Esto hace que esta licencia sea incompatible con la licencia GLP.

La principal preocupación de los desarrolladores de metodologías abiertas que esta licencia tiene en cuenta, es que ellos recibirán el debido reconocimiento por su contribución y desarrollo, así como también el reservarse el derecho de permitir las

publicaciones y distribuciones gratuitas cuando las metodologías abiertas no sean utilizadas en material comercial impreso del cual las ganancias se deriven ya sea de su publicación o distribución.

Como se pudo apreciar, esta última parte del trabajo no trata de ser un desarrollo de la metodología OSSTMM ni mucho menos, por eso justamente es que no la compara tampoco con otras, simplemente trata de remarcar que es posible aplicar técnicas o metodologías estándar, que permitan con total transparencia, mostrar resultados y dejar libertad de acción al cliente. Y este aspecto, remarco una vez más, se debe tener en cuenta como uno de los más importantes en TI para los tiempos que se avienen, pues ya no existe ninguna excusa sincera u honesta que de pie a dejar aferrado en algo a nadie cuando se trate de seguridad informática. La mejor y más sana solución que se debe ofrecer hoy en día a todo cliente, es justamente proporcionar un servicio y aferrar al mismo por el nivel de excelencia y no por otros nebulosos métodos, dejándole con absoluta sinceridad todas las herramientas que necesite para que pueda optar por quien lo desee, pero en virtud de la calidad con que se han hecho las cosas y el grado de satisfacción, no le quepan dudas si tiene que volver a levantar el teléfono pidiendo apoyo.

Independientemente de este nivel de excelencia, este tipo de metodologías estándar, lo que permiten es repetir la experiencia con la magnitud y la cantidad de veces que se desee, pudiendo en cada una de ellas evaluar el desvío que el sistema esta sufriendo, de forma totalmente numérica y objetiva.

8.4.5. Guía de pasos para la realización de una Auditoría de seguridad o Penetration Test.

A continuación presentamos una serie de aspectos que merece la pena sean tenidos en cuenta a la hora de realizar esta actividad o también cuando se contrate a terceros.

1) Introducción (Definición de la actividad):

- Penetration Test.
- Diagnóstico o evaluación de Seguridad.
- Auditoría de seguridad.

Definición del alcance del proyecto.
Penetration Test o evaluación Externo e Interno.
Penetration Test vía Internet.
Duración del proyecto
Objetivos del proyecto.

2) Pasos para realizar un Penetration Test o evaluación

Definición del alcance.
Definición de la metodología a utilizar.
Aplicación de la metodología.
Evaluación de los resultados obtenidos.
Corrección de los expuestos detectados.

3) Metodologías y Estándares en proyectos de Penetration Testing:

OSSTMM (Open Source Security Testing Methodology Manual.)
Metodología de Penetration Test o evaluación de Seguridad.:

- Descubrimiento.
- Exploración.
- Evaluación.
- Intrusión.

4) Fase de Descubrimiento

Recolección de información.
Escucha y análisis de tráfico.
Descubriendo la red (hardware y software de red).
Fuentes de información en Internet.
Direcciones físicas.
Detección de Redes WiFi y/o bluetooth.
Números telefónicos.
Nombres de personas, usuarios, cargos y cuentas de correo electrónico.
Rango de direcciones IP.
Información de prensa sobre el lugar.
Análisis de las páginas Web Institucionales y/o Intranet Corporativa.
Evaluación del código fuente.

5) Fase de Exploración:

Scanning telefónico.
Detección de hosts activos.
Detección y Análisis de servicios activos
Detección remota de sistemas operativos.
Determinación de mecanismos de criptografía en redes Wi-Fi.
Relevamiento de aplicaciones Web, correo, telnet, ftp, SSH, https, proxies, firewalls, IDSs, honey pots, etc.

6) Fase de Evaluación:

Detección de vulnerabilidades en forma remota.
Herramientas de detección de vulnerabilidades.

Testing de seguridad en Switches, Routers / Firewalls/ Dispositivos de Comunicaciones.
Testing de seguridad de servidores UNIX.
Testing de seguridad de servidores Windows.
Testing de seguridad de otros servidores.
Testing de eficacia de Sistemas de Detección de Intrusiones.
Testing de seguridad de Bases de Datos.
Testing de seguridad de aplicaciones (Web, correo, telnet, ftp, SSH, https, proxies, etc).

7) Fase de Intrusión:

Planificación de la intrusión.
Utilización de ingeniería social para obtención de información.
Explotación de las vulnerabilidades detectadas.
Acceso vía módems o accesos remotos detectados.
Intrusiones vía switch, router, punto de acceso, web, ftp. telnet, servidor de autenticación o de acceso, SSH, https.
Escalada de privilegios.
Combinación de vulnerabilidades para elevar el control.
Acceso a información interna.
Generación de evidencia de expuestos detectados.

8) Evaluación y corrección:

Evaluación de los resultados obtenidos.
Determinación de niveles de riesgo.
Propuesta de soluciones de seguridad.
Corrección de las vulnerabilidades detectadas.

8.5. Familia ISO 27000 (Sistema de Gestión de la Seguridad de la Información)

Como otros grupos de estándares, la serie 27xxx, forman un conjunto de normas cuyo objetivo es normalizar las técnicas, actividades y medidas a considerar en para la implantación de un Sistema de Gestión de la Seguridad de la Información.

En esta sección iremos tratando este tema, pues lo consideramos un hito fundamental para la seguridad de los sistemas de información del siglo XXI, y a su vez para intentar "desmitificar" todo el proceso que se debe llevar a cabo hasta su certificación, pues en nuestra experiencia de más de diez empresas certificadas, estamos convencidos que cualquier persona que cuente con los conocimientos necesarios y la decisión de hacerlo, puede perfectamente llegar a implantar un SGSI y certificarlo sin ningún tipo de problema.

Nuestra intención es presentar la teoría correspondiente y luego proponerte la secuencia de pasos que te pueden allanar el camino para esta actividad. No es necesario tener el objetivo de "Certificar" para implantar un SGSI, lo importante a nuestro juicio, es ser consciente que esta norma tiene una historia de más de 20 años y en la actualidad está tan madura que no deja "Huecos" en la seguridad de tus sistemas, por lo tanto, si haces el esfuerzo de alinear tus sistemas con lo que aquí se propone, no tengas duda que haz hecho un gran trabajo, y luego si en algún momento se presenta la oportunidad de solicitar la certificación, mejor que mejor, pero no dejes de lado lo que aquí trataremos, pues con toda sinceridad creemos que la palabra "Gestión" y el ciclo continuo que este estándar ofrece es la mejor solución a la mayoría de los problemas que hoy se sufren en la seguridad de los sistemas informáticos.

8.5.1. Presentación de los estándares que la componen

El conjunto de estándares que aportan información de la familia ISO-27xxx que se puede tener en cuenta son:

- **ISO/IEC 27000**: Es una visión general de las normas que componen la serie 27000. Actualmente sólo en inglés y puede descargarse gratuitamente en la Web: www.iso.org *(Publicada en mayo de 2009)*.
- **ISO/IEC 27001** (Sistema de Gestión de la Seguridad de la Información): Esta norma fue publicada en octubre de 2005, dejando obsoleta a la **ISO-17799**. Se trata de una norma certificable y es la que establece todos los requerimientos de un **SGSI.** Como anexo de la misma, pone de manifiesto el listado de los 133 "**Controles**" pero sin entrar en detalle sobre los mismos (misión de la 27002). A finales de 2009 **AENOR** la traduce al español y la publica como **UNE-ISO/IEC 27001:2007**. En 2009 AENOR también publica otro documento con ciertas modificaciones a la norma, que lo denominó **UNE-ISO/IEC 27001:2007/1M:2009.**
- **ISO/IEC 27002**: Código o Guía de buenas prácticas para la Seguridad de la Información, fue publicado el 15 de junio del 2005 y detalla los 133 controles reunidos en 11 grupos, más 39 "Objetivos de control". en España AENOR la publicó como **UNE-ISO/IEC 27002:2009** en diciembre de 2009.
- **ISO/IEC 27003**: Guía de Implementación. Describe los aspectos a tener en cuenta para la implantación de un SGSI, fue publicada en febrero de 2009 y aún no existe traducción al español.
- **ISO/IEC 27004**: Es la norma que describe todos los aspectos de métricas, indicadores y mediciones que deben realizarse sobre un SGSI. Se publicó en diciembre de 2009. Cabe mencionar aquí que según toda la familia, si un "Control" no es medible no sirve para nada (pues no nos permitirá evaluar su estado ni evolución), así que es muy recomendable considerar esta norma, sobre todo a la hora de ir avanzando en los ciclos de vida de un SGSI.
- **ISO/IEC 27005**: Trata los aspectos relacionados a la "Gestión de riesgos" tema de suma importancia en toda esta familia, fue publicada en junio de 2008 y aún no está traducida

al español. Cabe mencionar que para la certificación en ISO-27001, no se exige ninguna metodología en concreto para el "Análisis de Riesgo", siempre y cuando esta actividad sea coherente y metodológica, pero insistimos que es la primera y tal vez más importante actividad de toda esta familia.

- **ISO/IEC 27006:** Especifica los requisitos que debe reunir cualquier organización que desee acreditarse como "Entidad certificadora" de ISO 27001, fue publicada en marzo de 2007.
- **ISO/IEC 27007:** (Borrador) Guía para auditoría de un SGSI.
- **ISO/IEC 27008:** (Borrador) Guía para auditoría de los controles de un SGSI.
- **ISO/IEC 27010:** (Borrador) Guía para la gestión de la seguridad de Sistemas de Información entre organizaciones.
- **ISO/IEC 27011:** Guía de implementación de un SGSI para el sector de Telecomunicaciones, se publicó en diciembre de 2008, aún no está disponible en español.
- **ISO/IEC 27012:** (Borrador) SGSI para el sector de e-administración.
- **ISO/IEC 27013:** (Borrador) Integración con ISO-20000.
- **ISO/IEC 27014:** (Borrador) Gobierno corporativo de un SGSI.
- **ISO/IEC 27015:** (Borrador) Sector financiero.
- ISO/IEC 27016: (Borrador) Relacionado a finanzas en las organizaciones.
- **ISO/IEC 27031:** Directrices para la preparación de las TIC en la Continuidad de Negocio,.de reciente publicación, orientada a aspectos específicos de las TICs, en particular hacia el Plan de Continuidad de Negocio.
- **ISO/IEC 27032:** (Borrador) Ciberseguridad.
- **ISO/IEC 27033:** Seguridad en redes, consta de 7 partes, de las cuáles la 1 y 2 ya están disponibles.
- **ISO/IEC 27034:** (Borrador) Guías de seguridad para aplicaciones informáticas.
- **ISO/IEC 27035:** (Borrador) Guía para la gestión reincidentes de seguridad.
- **ISO/IEC 27036:** (Borrador) Guía de seguridad para externalización de prestaciones.
- **ISO/IEC 27037:** (Borrador) Relacionada a evidencias digitales.
- **ISO/IEC 27038:** (Borrador) Redacción digital.
- **ISO/IEC 27039:** (Borrador) Sistemas de detección de intrusiones (IDSs).
- **ISO/IEC 27040:** (Borrador) Seguridad en almacenamiento de información.
- **ISO 27799:** Publicada en el 2008 y está orientada a la aplicación de un SGSI en el ámbito sanitario. Desde el año pasado ya está disponible en AENOR su versión en español.

8.5.2. Breve historia

ISO (Organización Internacional de Estándares) e **IEC** (Comisión Internacional de Electrotecnia) conforman un sistema especializado para los estándares mundiales. Organismos nacionales que son miembros de ISO o IEC participan en el desarrollo de Normas Internacionales a través de comités técnicos establecidos por la organización respectiva para tratar con los campos particulares de actividad técnica. Los comités técnicos de ISO e IEC colaboran en los campos de interés mutuo. Otras organizaciones internacionales, gubernamentales y no gubernamentales, en relación con ISO e IEC, también forman parte del trabajo.

En el campo de tecnología de información, ISO e IEC han establecido unirse en un comité técnico, ISO/IEC **JTC 1** (Join Technical Committee Nº1). Los borradores de estas normas Internacionales adoptadas por la unión de este comité técnico son enviados a los organismos de las diferentes naciones para su votación. La publicación, ya como una Norma Internacional, requiere la aprobación de por lo menos el 75% de los organismos nacionales que emiten su voto.

La historia de esta familia ISO-2700, tiene más de veinte años, naciendo como Estándar Internacional **ISO/IEC 17799** el cual a su vez tuvo el antecesor preparado inicialmente por el Instituto de Normas Británico (como **BS 7799**) y fue adoptado, bajo la supervisión del grupo de trabajo "Tecnologías de la Información", del Comité Técnico de esta unión entre ISO/IEC JTC 1, en paralelo con su aprobación por los organismos nacionales de ISO e IEC.

El estándar ISO/IEC 27001 es el nuevo estándar oficial, su título completo inicial fue: **BS 7799-2:2005 (ISO/IEC 27001:2005)**. También preparado por este JTC 1 y en el subcomité **SC 27**, IT "Security Techniques".

1870 organizaciones en 57 países han reconocido la importancia y los beneficios de esta nueva norma. Actualmente el **ISO-27001** es el único estándar aceptado internacionalmente para la administración de la seguridad de la información y aplica a todo tipo de organizaciones, tanto por su tamaño como por su actividad

A los efectos de la certificación, hubo una **transición** entre ambas normas quedando propuesta (o establecida) por el **TPS-55** de **UKAS** (United Kingdom Acreditation Service): "*Transition Statement Regarding Arrangements for the Implementation of ISO 27001:2005"*. Establecía que las empresas (en realidad los auditores, lo cual afecta directamente a las empresas) durante los primeros seis meses (desde que se firmó el acuerdo "MoU: Memorandum of Understanding" entre UKAS y el Departamento de Comercio e Industria de Reino Unido), pueden elegir acerca de qué estándar aplicar, a partir del 23 de julio del 2006, la única certificación que se deberá aplicar será la ISO/IEC 27001:2005. Ante cualquier no conformidad con la aplicación de la misma motivada claramente por su transición, se estableció un plazo de un año para solucionarla, es decir, hasta el 23 de julio de 2007.

Como mencionamos al principio, uno de los aspectos que consideramos más acertados de esta norma es la palabra "Gestión". Evidentemente la experiencia que se venía acumulando con la familia **ISO-9000** "Gestión de Calidad", **ISO-14000** "Gestión medioambiental", etc… demostraba que un sistema actual, no puede ser sólo un hito, un umbral superado.

Necesita inexorablemente un "Ciclo de vida" un ciclo de mejora continua, que no es más que lo que se conocía como "ciclo de Deming" (PDCA: Plan – Do – Check – Act). En pleno siglo XXI, la evolución es permanente y vertiginosa, por lo tanto lo más importante es "mantener vivas" las infraestructuras, y no podía ser menos con las más aceleradas de todas, las infraestructuras de Sistemas de Información (SSII).

¿Gestionar la Seguridad?

Si bien, a nivel físico, siempre existieron inconvenientes de acceso a los SSII. La seguridad de la información es un concepto que nació hace unos veinte años, cuando las redes comenzaron a interconectarse, y se presentó por primera vez la posibilidad de tener acceso lógico y a distancia a los recursos. Estas redes, inicialmente de investigación universitaria, se integraron con redes militares, y luego de empresas. Esta "melange" de objetivos, hizo muy tentador el acceso a las mismas. El objetivo de los intrusos de los años 80`y 90`era mayoritariamente militar y de investigación, hoy en día es netamente económico, y por esta razón el blanco predilecto es cualquier empresa que de alguna u otra forma pueda ser rentable a un intruso.

Cuando se empiezan a producir estos problemas de seguridad, casi como una acelerada carrera, aparece las primeras medidas de seguridad, las cuales día a día son violadas, y se generan contramedidas, las que entran en el mismo proceso y aparecen las contra–contra medidas, y las… en forma de carrera cada vez más acelerada y sin un final predecible.

Este conjunto de acciones técnicas que aparecen día a día para mitigar los problemas de seguridad, es lo que hoy en día se denomina "Administrar seguridad". La palabra clave es justamente "administrar", esto indica claramente una organización de recursos para un objetivo específico concreto (Salvaguardar los recursos de la organización).

Como todo directivo conoce bien, si se hila muy fino, no es lo mismo "**Administrar**" que "**Gestionar**", como tampoco es lo mismo el rol de "Administrador" que el de "Director".

La tendencia actual en todo ámbito relacionado con recursos de la organización es la de Gestionarlos. En las organizaciones, ya no hay duda al respecto si se piensa en trabajar con "calidad", y para ello apareció ISO–9000, que ganó todo el terreno en Europa. En el caso de las TI, esta diferencia es tan radical que rápidamente, se han elaborado dos familias de estándares que se están imponiendo exponencialmente, para SSII: ISO–20000 y para Seguridad de los SSII: ISO–27000. Cualquiera de los tres mencionados, tiene como objetivo final lograr la calidad por medio de la "GESTION".

A continuación se presenta una tabla, donde se reflejan las diferencias entre "Administrar y Gestionar" Seguridad:

Administrar Seguridad	Gestionar Seguridad
• Dispersión de Plataformas. • Especialización Puntual. • Ausencia de Estándares. • Mayor Cantidad de Recursos. • Mayor Tiempo de Peticiones. • Dificultad de Medición. • Riesgo Difícil de Controlar y Gestionar.	• Ahorro en Inversiones y Tiempo Formación de un Equipo Técnico y Multidisciplinar. • Organización Dedicada a su Negocio. • Facilidad Control e Implantación Políticas de Seguridad. • Delegación de Tareas. • Economía de Escala. • Gestión Administración Completa e Integral. • Procedimientos Actualizados según disposiciones Legales Vigentes y Evolución de Necesidades. • Gestión del Riesgo y Alineación de Inversiones y Negocio. • Alineación de Seguridad con Calidad.

8.5.3. ISO 27001

La propuesta de esta norma, no está orientada a despliegues tecnológicos o de infraestructura, sino a aspectos netamente organizativos, es decir, la frase que podría definir su propósito es "Organizar la seguridad de la información", por ello propone toda una secuencia de acciones tendientes al "establecimiento, implementación, operación, monitorización, revisión, mantenimiento y mejora del **SGSI** o en Inglés **ISMS** (Information Security Management System)". El SGSI, es el punto fuerte de este estándar.

Los detalles que conforman el cuerpo de esta norma, se podrían agrupar en **tres grandes líneas**:

- **SGSI**
- **Valoración de riegos (Risk Assesment)**
- **Controles**

El desarrollo de estos puntos y la documentación que generan, es lo que se tratará en este texto.

Los párrafos siguientes son una breve descripción de los puntos que se considerarán en este texto para poder llegar finalmente y avalando la importancia de la documentación que es necesaria preparar y mantener.

Hemos considerado importante mantener la misma puntuación que emplea el Estándar Internacional, para que, si fuera necesario, se pueda acceder directamente al mismo, para ampliar cualquier aspecto, por lo tanto, la numeración que sigue a continuación, no respeta la de este texto, pero sí la de la norma.

0. Introducción:

0.1 General:

Este estándar fue confeccionado para proveer un modelo para el establecimiento, implementación, operación, monitorización, revisión, mantenimiento y mejora del SGSI, la adopción del SGSI debe ser una decisión estratégica de la organización, pues el mismo está influenciado por las necesidades y objetivos de la misma, los requerimientos de seguridad, los procesos, el tamaño y la estructura de la empresa, la dinámica que implica su aplicación, ocasionará en muchos casos la escalada del mismo, necesitando la misma dinámica para las soluciones.

0.2. Aproximación (o aprovechamiento) del modelo:

Este estándar internacional adopta un proceso para establecer, implementar, operar, monitorizar, revisar, mantener y mejorar el SGSI en una organización.

Una organización necesita identificar y administrar cualquier tipo de actividad para funcionar eficientemente. Cualquier actividad que emplea recursos y es administrada para transformar entradas en salidas, puede ser considerada como un "proceso". A menudo, estas salidas son aprovechadas nuevamente como entradas, generando una realimentación de los mismos.

Este estándar internacional adopta también el modelo "Plan-Do-Check-Act" (**PDCA**), el cual es aplicado a toda la estructura de procesos de SGSI, y significa lo siguiente:

- **Plan** (Establecer el SGSI): Implica, establecer a política SGSI, sus objetivos, procesos, procedimientos relevantes para la administración de riesgos y mejoras para la seguridad de la información, entregando resultados acordes a las políticas y objetivos de toda la organización.
- **Do** (Implementar y operar el SGSI): Representa la forma en que se debe operar e implementar la política, controles, procesos y procedimientos.
- **Check** (Monitorizar y revisar el SGSI): Analizar y medir donde sea aplicable, los procesos ejecutados con relación a la política del SGSI, evaluar objetivos, experiencias e informar los resultados a la administración para su revisión.
- **Act** (Mantener y mejorar el SGSI): Realizar las acciones preventivas y correctivas, basados en las auditorías internas y revisiones del SGSI o cualquier otra información relevante para permitir la continua mejora del SGSI.

1.2. Aplicación:

Los requerimientos de este estándar internacional, son genéricos y aplicables a la totalidad de las organizaciones. La exclusión de los requerimientos especificados en las cláusulas 4, 5, 6, 7 y 8, no son aceptables cuando una organización solicite su conformidad con esta norma.

Estas cláusulas son:

4. SGSI.
5. Responsabilidades de la Administración
6. Auditoría Interna del SGSI
7. Administración de las revisiones del SGSI
8. Mejoras del SGSI.

(Estas cláusulas realmente conforman el cuerpo principal de esta norma)

Cualquier exclusión a los controles detallados por la norma y denominados como "necesarios" para satisfacer los criterios de aceptación de riegos, debe ser justificada y se debe poner de manifiesto, o evidenciar claramente los criterios por los cuales este riesgo es asumido y aceptado. En cualquier caso en el que un control sea excluido, la conformidad con este estándar internacional, no será aceptable, a menos que dicha exclusión no afecte a la capacidad y/o responsabilidad de proveer seguridad a los requerimientos de información que se hayan determinado a través de la evaluación de riesgos, y sea a su vez aplicable a las regulaciones y legislación vigente.

2. Normativas de referencia:

Para la aplicación de este documento, es indispensable tener en cuenta la última versión de:

"ISO/IEC 17799:2005, *Information technology — Security techniques — Code of practice for information security management"*

3. Términos y definiciones:

La siguiente terminología aplica a esta norma:

3.1. Recurso (Asset): Cualquier cosa que tenga valor para la organización.

3.2. Disponibilidad (availability): Propiedad de ser accesible y usable bajo demanda por una entidad autorizada.

3.3. Confidencialidad (confidentiality): Propiedad que la información no esté disponible o pueda ser descubierta por usuarios no autorizados, entidades o procesos.

3.4. Seguridad de la información: Preservación de la confidencialidad, integridad y disponibilidad de la información, en adición también de otras propiedades como autenticación, autorización, registro de actividad, no repudio y confiabilidad pueden ser también consideradas.

3.5. Eventos de seguridad de la información: Ocurrencia de un evento identificado sobre un sistema, servicio o red, cuyo estado indica una posible brecha en la política de seguridad de la información o fallo en el almacenamiento de la misma, también cualquier situación previa desconocida que pueda ser relevante desde el punto de vista de la seguridad.

3.6. Incidente de seguridad: uno o varios eventos de seguridad de la información, no deseados o inesperados que tienen una cierta probabilidad de comprometer las operaciones de la empresa y amenazan a la seguridad de la información.

3.7. Sistema de administración de la seguridad de la información (SGSI: Information Security Management System): Parte de los sistemas de la empresa, basado en el análisis de riesgo de negocio, cuya finalidad es establecer, implementar, operar, monitorizar, revisar, mantener y mejorar la seguridad de la información.

NOTA: el SGSI incluye las políticas, planes, actividades, responsabilidades, prácticas, procedimientos, procesos y recursos.

3.8. Integridad: Propiedad de salvaguardar la precisión y completitud de los recursos.

3.9. Riesgo residual: El riesgo remanente luego de una amenaza a la seguridad.

3.10. Aceptación de riesgo: Decisión de aceptar un riesgo.

3.11. Análisis de riego: Uso sistemático de la información para identificar fuentes y estimar riesgos.

3.12. Valoración de riesgo: Totalidad de los procesos de análisis y evaluación de riesgo.

3.13. Evaluación de riesgo: Proceso de comparar los riesgos estimados contra los criterios de riesgo establecidos o dados, para determinar el grado de significativo del riesgo.

ACLARACIÓN AJENA A LA NORMA: En definitiva la "Evaluación del riesgo", es el resultado final de esta actividad, pero no debe ser pensada únicamente con relación a "Análisis y Valoración", sino también a los criterios de riesgo que la organización haya definido a lo largo de toda su política empresarial.

3.14. Administración del riesgo: Actividades coordinadas para dirigir y controlar las medidas necesarias para la observación del riesgo dentro de la organización.

3.15. Tratamiento del riesgo: Proceso de selección e implementación de mediciones para modificar el riesgo.

NOTA: el término "**control**" en esta norma es empleado como sinónimo de "Medida o medición".

3.16. Declaración de aplicabilidad: Documento que describe los objetivos del control, y los controles que son relevantes y aplicables a la organización del SGSI.

NOTA: Estos controles están basados en los resultados y conclusiones de la valoración y los procesos de tratamiento de riesgo, los requerimientos y regulaciones legales, las obligaciones contractuales y los requerimientos de negocio para la seguridad de la información que defina la organización.

4. SGSI (Sistema de Gestión de Seguridad de la Información).

4.1. Requerimientos generales:

La organización, establecerá, implementará, operará, monitorizará, revisará, mantendrá y mejorará un documentado SGSI en el contexto de su propia organización para las

actividades globales de su negocio y de cara a los riesgos. Para este propósito esta norma el proceso está basado en el modelo PDCA comentado en el punto 0.2.

4.3.2. Control de documentos:

Todos los documentos requeridos por el SGSI serán protegidos y controlados. Un procedimiento documentado deberá establecer las acciones de administración necesarias para:

- ➔ Aprobar documentos y prioridades o clasificación de empleo.
- ➔ Revisiones, actualizaciones y reaprobaciones de documentos.
- ➔ Asegurar que los cambios y las revisiones de documentos sean identificados.
- ➔ Asegurar que las últimas versiones de los documentos aplicables estén disponibles y listas para ser usadas.
- ➔ Asegurar que los documentos permanezcan legibles y fácilmente identificables.
- ➔ Asegurar que los documentos estén disponibles para quien los necesite y sean transferidos, guardados y finalmente dispuestos acorde a los procedimientos aplicables a su clasificación.
- ➔ Asegurar que los documentos de origen externo sean identificados.
- ➔ Asegurar el control de la distribución de documentos.
- ➔ Prevenir el empleo no deseado de documentos obsoletos y aplicar una clara identificación para poder acceder a ellos y que queden almacenados para cualquier propósito

5. Responsabilidades de administración:

5.1. La administración proveerá evidencias de sus compromisos para el establecimiento, implementación, operación, monitorización, mantenimiento y mejora del SGSI a través de:

- Establecimiento de la política del SGSI
- Asegurar el establecimiento de los objetivos y planes del SGSI.
- Establecer roles y responsabilidades para la seguridad de la información.
- Comunicar y concienciar a la organización sobre la importancia y apoyo necesario a los objetivos propuestos por la política de seguridad, sus responsabilidades legales y la necesidad de una continua mejora en este aspecto.
- Proveer suficientes recursos para establecer, operar, implementar, monitorizar, revisar, mantener y mejorar el SGSI (5.2.1).
- Decidir los criterios de aceptación de riesgos y los niveles del mismo.

- Asegurar que las auditorías internas del SGSI, sean conducidas y a su vez conduzcan a la administración para la revisión del SGSI (ver 7.)

5.2.2. Formación, preparación y competencia:

La organización asegurará que todo el personal a quien sean asignadas responsabilidades definidas en el SGSI sea competente y esté en capacidad de ejecutar las tareas requeridas, para ello deberá proveer las herramientas y capacitación necesaria (Documento: Planificación, guías y programas de formación y preparación).

6. Auditoría interna del SGSI:

La organización realizará auditorías internas al SGSI a intervalos planeados para determinar si los controles, sus objetivos, los procesos y procedimientos continúan de conformidad a esta norma y para analizar y planificar acciones de mejora. Ninguna persona podrá auditar su propio trabajo, ni cualquier otro que guarde relación con él.

La responsabilidad y requerimientos para el planeamiento y la conducción de las actividades de auditoría, los informes resultantes y el mantenimiento de los registros será definido en un procedimiento (Ver: Procedimiento de Revisión del SGSI - Periódicas y aperiódicas

7. Administración de las revisiones del SGSI:

Las revisiones mencionadas en el punto anterior deberán llevarse a cabo al menos una vez al año para asegurar su vigencia, adecuación y efectividad. Estas revisiones incluirán valoración de oportunidades para mejorar o cambiar el SGSI incluyendo la política de seguridad de la información y sus objetivos. Los resultados de estas revisiones, como se mencionó en el punto anterior serán claramente documentados y los mismos darán origen a esta actividad.

Esta actividad está constituida por la revisión de entradas (7.2.) y la de salidas (7.3.) y dará como resultado el documento correspondiente (Ver: Documento de administración de las revisiones del SGSI).

8. Mejoras al SGSI

La organización deberá mejorar continuamente la eficiencia del SGSI a través del empleo de la política de seguridad de la información, sus objetivos, el resultado de las auditorías, el análisis y monitorización de eventos, las acciones preventivas y correctivas y las revisiones de administración.

8.2. Acciones correctivas:

La organización llevará a cabo acciones para eliminar las causas que no estén en conformidad con los requerimientos del SGSI con el objetivo de evitar la recurrencia de los mismos. Cada una de estas acciones correctivas deberá ser documentada (Ver: Documento de acciones correctivas)

El anexo A de esta norma propone una detallada tabla de los controles, los cuales quedan agrupados y numerados de la siguiente forma:

A.5 Política de seguridad

A.6 Organización de la información de seguridad

A.7 Administración de recursos

A.8 Seguridad de los recursos humanos

A.9 Seguridad física y del entorno

A.10 Administración de las comunicaciones y operaciones

A.11 Control de accesos

A.12 Adquisición de sistemas de información, desarrollo y mantenimiento

A.13 Administración de los incidentes de seguridad

A.14 Administración de la continuidad de negocio

A.15 Cumplimiento (legales, de estándares, técnicas y auditorías)

El anexo B, que es informativo, a su vez proporciona una breve guía de los principios de OECD (guía de administración de riesgos de sistemas de información y redes - París, Julio del 2002, "www.oecd.org") y su correspondencia con el modelo PDCA.

Por último el **Anexo C**, también informativo, resume la correspondencia entre esta norma y los estándares ISO 9001:2000 y el ISO 14001:2004

8.5.4. ISO 27002

Esta norma como mencionamos al principio es la que describe con máximo de detalle cada uno de los controles y objetivos de control que se deben considerar. A la hora de implantar un SGSI, se debe tener en cuenta cada uno de ellos desde el principio, pues hay un documento que se denomina "Declaraciones de aplicabilidad" en el cual se presenta cómo tratará la organización a cada uno de ellos. Esto tiene mucha lógica, pues sobre cada uno de los controles, es que se debe hacer un minucioso análisis para ver cuáles "aplican" y cuales "no aplican", no tendría ningún sentido dedicar tiempo a "comercio electrónico o áreas de carga y descarga, etc…" si mi empresa no lo hace o no las tiene. Una vez hecho el análisis

se deben justificar (aplicando la lógica) las causas por las cuales esos "controles" no serán de aplicación en este SGSI.

A continuación presentamos una tabla resumen de cada uno de los establecidos en esta norma:

DOC	OBJETIVO	CONTROL
5	POLÍTICA DE SEGURIDAD	
5.1	Política de Seguridad de la Información.	**5.1.1** Documento de Política de Seguridad de la Información.
		5.1.2 Revisión de la Política de Seguridad de la Información.
6	ORGANIZACIÓN DE LA SEGURIDAD DE LA INFORMACIÓN	
6.1	Organización Interna.	**6.1.1** Comisión de Gestión de la Seguridad de la Información.
		6.1.2 Coordinación de la Seguridad de la Información.
		6.1.3 Asignación de Responsabilidades Sobre Seguridad de la Información.
		6.1.4 Proceso de Autorización de Recursos Para el Tratamiento de la Información.
		6.1.5 Acuerdos de Confidencialidad.
		6.1.6 Contacto con las Autoridades.
		6.1.7 Contacto con Grupos de Interés Especial.
		6.1.8 Revisión Independiente de la Seguridad de la Información.
6.2	Partes Externas.	**6.2.1** Identificación de Riesgos Relativos a Partes Externas.
		6.2.2 Consideración de la Seguridad en el Trato con los Clientes.
		6.2.3 Consideración de la Seguridad con Contratos con Terceros.
7	GESTIÓN DE ACTIVOS	
7.1	Responsabilidad Sobre los Activos.	**7.1.1** Inventario de Activos.
		7.1.2 Propietarios de los Activos.
		7.1.3 Uso Aceptable de los Activos.
7.2	Clasificación de la Información.	**7.2.1** Guías de Clasificación.
		7.2.2 Marcado y Tratamiento de la Información.
8	SEGURIDAD LIGADA A LOS RECURSOS HUMANOS	
8.1	Previo a la Contratación.	**8.1.1** Funciones y Responsabilidades.
		8.1.2 Investigación de Antecedentes.

		8.1.3 Términos y Condiciones de Contratación.
8.2	Durante la Contratación.	**8.2.1** Gestión de Responsabilidades.
		8.2.2 Concienciación, Educación y Formación en Seguridad de la Información.
		8.2.3 Proceso Disciplinario.
8.3	Finalización o Cambio de Puesto de Trabajo.	**8.3.1** Finalización de Responsabilidades.
		8.3.2 Retorno de Activos.
		8.3.3 Retirada de los Derechos de Acceso.
9	**SEGURIDAD FÍSICA Y DEL ENTORNO**	
9.1	Áreas Seguras.	**9.1.1** Perímetro de Seguridad Física.
		9.1.2 Controles Físicos de Entrada.
		9.1.3 Seguridad de Oficinas, Despachos y Recursos.
		9.1.4 Protección Contra las Amenazas Externas y del Entorno.
		9.1.5 Trabajando en Áreas Seguras.
		9.1.6 Áreas de Acceso Público, de Carga y Descarga.
9.2	Seguridad de los Equipos.	**9.2.1** Instalación y Protección de Equipos.
		9.2.2 Instalaciones de Suministro.
		9.2.3 Seguridad del Cableado.
		9.2.4 Mantenimiento de Equipos.
		9.2.5 Seguridad de los Equipos Fuera los Locales de la Organización.
		9.2.6 Seguridad en la Reutilización o Eliminación de Equipos.
		9.2.7 Extracción de Pertenencias.
10	**GESTIÓN DE COMUNICACIONES Y OPERACIONES**	
10.1	Responsabilidades y Procedimientos Operativos.	**10.1.1** Procedimientos Operacionales Documentados.
		10.1.2 Gestión del Cambio.
		10.1.3 Segregación de Tareas.
		10.1.4 Separación de los Recursos de Desarrollo, Ensayo y Operacionales.
10.2	Gestión de Entrega del Servicio por Tercera Parte.	**10.2.1** Entrega del Servicio.
		10.2.2 Control y Revisión de los Servicios por Tercera Parte.
		10.2.3 Gestión de Cambios en los Servicios por Tercera Parte.
10.3	Sistema de Planificación y Aceptación.	**10.3.1** Gestión de la Capacidad.
		10.3.2 Aceptación del Sistema.

10.4	Protección Contra el Código Malicioso y Ambulante.	**10.4.1** Controles Contra el Código Malicioso.
		10.4.2 Controles Contra el Código Ambulante.
10.5	Copias de Seguridad.	**10.5.1** Información de Copias de Seguridad.
10.6	Gestión de la Seguridad de las Redes.	**10.6.1** Controles de Red.
		10.6.2 Seguridad de los Servicios de Red.
10.7	Manejo de Soportes.	**10.7.1** Gestión de Soportes Extraíbles.
		10.7.2 Retirada de los Soportes.
		10.7.3 Procedimientos de Tratamiento de la Información.
		10.7.4 Seguridad de la Documentación del Sistema.
10.8	Intercambio de Información.	**10.8.1** Políticas y Procedimientos de Intercambio de Información.
		10.8.2 Acuerdos de Intercambio.
		10.8.3 Soportes Físicos en Tránsito.
		10.8.4 Envío Correo Electrónico.
		10.8.5 Sistemas de Información del Negocio.
10.9	Servicios de Comercio Electrónico.	**10.9.1** Comercio Electrónico.
		10.9.2 Transacciones On–Line.
		10.9.3 Información Pública Disponible.
10.10	Seguimiento.	**10.10.1** Registros de Auditoría.
		10.10.2 Seguimiento del Uso del Sistema.
		10.10.3 Protección de los Registros de Información.
		10.10.4 Diario de Operaciones y Administración.
		10.10.5 Registro de Fallos.
		10.10.6 Sincronización del Reloj.
11	**CONTROL DE ACCESO**	
11.1	Requisitos de Negocio Para el Control de Acceso.	**11.1.1** Política de Control de Acceso.
11.2	Gestión de Acceso de Usuario.	**11.2.1** Registro de Usuario.
		11.2.2 Gestión de Privilegios.
		11.2.3 Gestión de Contraseñas de Usuario.
		11.2.4 Revisión de los Derechos de Acceso de Usuario.
11.3	Responsabilidades de Usuario.	**11.3.1** Uso de Contraseña.
		11.3.2 Equipo de Usuario

		Desatendido.
		11.3.3 Política de Puesto de Trabajo Despejado y Pantalla Limpia.
11.4	Control de Acceso de Red.	**11.4.1** Política de Uso de los Servicios de Red.
		11.4.2 Autenticación de Usuario Para Conexiones Externas.
		11.4.3 Identificación de Equipos en las Redes.
		11.4.4 Diagnóstico Remoto y Configuración de la Protección del Puerto.
		11.4.5 Segregación de las Redes.
		11.4.6 Control de la Conexión a Red.
		11.4.7 Control del Direccionamiento de Red.
11.5	Control de Acceso al Sistema Operativo.	**11.5.1** Procedimientos de Entrada Seguros.
		11.5.2 Identificación y Autenticación de Usuario.
		11.5.3 Sistema de Gestión de Contraseñas.
		11.5.4 Uso de los Recursos del Sistema.
		11.5.5 Tiempo de Conexión de la Sesión.
		11.5.6 Limitación del Tiempo de Conexión.
11.6	Control de Acceso a la Aplicación y a la Información.	**11.6.1** Restricción del Acceso a la Información.
		11.6.2 Aislamiento del Sistema Sensible.
11.7	Ordenadores Portátiles y Tele trabajo.	**11.7.1** Ordenadores y Comunicaciones Móviles.
		11.7.2 Tele trabajo.
12	**ADQUISICIÓN DESARROLLO Y MANTENIMIENTO DE LOS SISTEMAS DE INFORMACIÓN**	
12.1	Requisitos de Seguridad de los Sistemas de Información.	**12.1.1** Análisis y Especificación de los Requisitos de Seguridad.
12.2	Procesamiento Correcto en las Aplicaciones.	**12.2.1** Validación de los Datos Introducidos.
		12.2.2 Control de Procesamiento Interno.
		12.2.3 Integridad de los Mensajes.
		12.2.4 Validación de los Datos

		Resultantes.
12.3	Controles Criptográficos.	**12.3.1** Política Acerca del Uso de Controles Criptográficos.
		12.3.2 Gestión de las Claves.
12.4	Seguridad de los Archivos del Sistema.	**12.4.1** Control del Software Operativo.
		12.4.2 Protección de los Datos de Prueba del Sistema.
		12.4.3 Control de Acceso al Código Fuente de los Programas.
12.5	Seguridad en el Desarrollo y Procesos de Asistencia Técnica.	**12.5.1** Procedimientos de Control de Cambios.
		12.5.2 Revisión Técnica de las Aplicaciones Tras Efectuar Cambios en el Sistema Operativo.
		12.5.3 Restricciones de los cambios a los Paquetes de Software.
		12.5.4 Fugas de Información.
		12.5.5 Externalización del Desarrollo de Software.
12.6	Gestión de la Vulnerabilidad Técnica.	**12.6.1** Control de las Vulnerabilidades Técnicas.
13	**GESTIÓN DE INCIDENTES EN LASEGURIDAD DE LA INFORMACIÓN**	
13.1	Notificación de Eventos y Puntos Débiles de la Seguridad de la Información.	**13.1.1** Comunicación de los Eventos de Seguridad de la Información.
		13.1.2 Comunicación de Puntos Débiles de Seguridad.
13.2	Gestión de Incidentes de Seguridad de la Información y Mejoras.	**13.2.1** Responsabilidades y Procedimientos.
		13.2.2 Aprendizaje de los Incidentes de Seguridad de la Información.
		13.2.3 Recopilación de Pruebas.
14	**GESTIÓN DE LA CONTINUIDAD DEL NEGOCIO**	
14.1	Aspectos de Seguridad de la Información en la Gestión de la Continuidad del Negocio.	**14.1.1** Inclusión de la Seguridad de la Información en el Proceso de la Continuidad del Negocio.
		14.1.2 Continuidad de Negocio y Evaluación de Riesgos.
		14.1.3 Desarrollo e Implementación de Planes de Continuidad que incluyan Seguridad de la Información.
		14.1.4 Marco de Referencia Para la Planificación de la Continuidad del Negocio.
		14.1.5 Pruebas, Mantenimiento y

		Reevaluación de los Planes de Continuidad del Negocio.
15	CUMPLIMIENTO	
15.1	Cumplimiento de los Requisitos Legales.	**15.1.1** Identificación de la Legislación Aplicable.
		15.1.2 Derechos de la Propiedad Intelectual (IPR).
		15.1.3 Protección de los Registros de la Organización.
		15.1.4 Protección de Datos y Privacidad de la Información Personal.
		15.1.5 Prevención del Uso Indebido de las Instalaciones de Procesamiento de la Información.
		15.1.6 Regulación de los Controles Criptográficos.
15.2	Cumplimiento de Políticas y Normas de Seguridad y Cumplimiento Técnico.	**15.2.1** Cumplimiento de las Políticas y Normas de Seguridad.
		15.2.2 Comprobación del Cumplimiento Técnico.
15.3	Consideraciones de la Auditoría de los Sistemas de Información.	**15.3.1** Controles de Auditoría de los Sistemas de Información.
		15.3.2 Protección de las Herramientas de Auditoría de los Sistemas de Información.

8.5.5. Análisis de riesgo

El Análisis de Riesgos es el paso inicial para conocer el nivel de seguridad en que se encuentra la compañía, estableciéndose una relación entre la misma y su entorno. Se identifican cuáles son sus puntos fuertes y cuáles no lo son, cuáles son las oportunidades que tiene, y qué amenazas pueden provocar algún desastre.

Existen varias metodologías para realizar un Análisis de Riesgos. Nosotros hemos utilizado MAGERIT porque consideramos que es una de las metodologías más completas que existen y además es gratis (No así su herramienta EAR para las empresas privadas, y sí lo es su herramienta PILAR para empresas públicas españolas).

De lo que se trata es de analizar los riesgos para los activos críticos. Una vez evaluado, se estima qué medidas de seguridad son necesarias para mitigar el riesgo (no debemos olvidar que el riesgo nunca desaparece) y se implantan o se planifica una implantación progresiva año a año. Este es un tema de especial interés pues una certificación ISO 27001, no nos exige que todo lo evaluado en el análisis de riesgo deba ser solucionado "ya", en muchos casos puede suceder que el conjunto de medidas a implementar sea de un coste no asumible por la empresa, en esos casos lo que se debe hacer es una planificación progresiva de estas implementaciones y "**Asumir el riesgo**". Esta última frase es

trascendental, pues una vez realizado un metódico análisis de riesgo, se debe presentar a la dirección de la empresa la situación real con sus fortalezas, debilidades, riesgo e impacto. Como consejo, no suele ser eficiente "intimidar" a la Dirección con una solución única (como si fuera: "Pagas esta solución o tus sistemas serán peligrosísimos"), es aconsejable que una vez que la misma haya comprendido la situación real y concreta en que se encuentra, se le propongan dos, tres o cuatro "Cursos de acción" posibles:

- El de máxima: Máximo gasto – mínimo riego.
- Intermedios: Gastos medios – Riesgos medios.
- El de mínima: Mínimo gasto – alto riesgo.

La dirección es la que conoce al detalle el concepto de "coste-beneficio" y si le hemos sabido explicar con claridad la situación, comprenderá perfectamente a qué se atiene, a su vez sabe qué dinero puede invertir o no ese año en seguridad (pues debemos ser conscientes que existen cientos de gastos e inversiones más para toda la empresa). Aquí aparece uno de los primeros GRANDES ACIERTOS de esta norma, pues una vez que la dirección adopta su "Curso de Acción", ¡**Lo Firma**! asumiendo ese riesgo….. esto para los que llevamos años en seguridad es casi como haber tocado el cielo, pues ahora existe una responsabilidad desde la más alta dirección de esta organización en la cual, si no quiso invertir en una determinada medida, nos exime de toda culpa si esta incidencia llegara a suceder (recordad este concepto pues es importantísimo). Luego de todo esto, no nos olvidemos que si decidimos certificar este SGSI cada año, la entidad certificadora, nos exigirá que demostremos que esta planificación del riesgo se cumpla, de otra forma no nos renovará nuestra certificación.

Las actividades que se realizan en un Análisis de Riesgos son:

- Planificación del Análisis de Riesgos.
- Inventario de activos.
- Valoración de activos.
- Dependencias.
- Análisis de impacto.
- Análisis de Riesgos.
- Tratamiento y Gestión de Riesgos.

8.5.6. Controles

Los controles serán seleccionados e implementados de acuerdo a los requerimientos identificados por la valoración del riesgo y los procesos de tratamiento del riesgo. Es decir, de esta actividad surgirá la primera decisión acerca de los controles que se deberán abordar, recordando que luego deberemos analizar cuáles de ellos aplicarán y cuáles no.

La preparación y planificación de SGSI, se trató en el artículo anterior, pero en definitiva, lo importante de todo este proceso es que desencadena en una serie de **controles** (o mediciones) a considerar y documentar, que se puede afirmar, **son uno de los aspectos fundamentales del SGSI** (junto con la Valoración de riesgo). Cada uno de ellos se encuentra en estrecha relación a todo lo que especifica la norma ISO/IEC 17799:2005 en los puntos 5 al 15, y tal vez estos sean el máximo detalle de afinidad entre ambos estándares.

La evaluación de cada uno de ellos debe quedar claramente documentada, y muy especialmente la de los controles que se consideren excluidos de la misma.

"DESCONCEPTO": Al escuchar la palabra "**Control**", automáticamente viene a la mente la idea de alarma, hito, evento, medición, monitorización, etc…., se piensa en algo muy **técnico o acción**. En el caso de este estándar, el concepto de "**Control**", **es mucho** (pero mucho) **más que eso**, pues abarca todo el conjunto de acciones, documentos, medidas a adoptar, procedimientos, medidas técnicas, etc…….

> Un "**Control**" es lo que permite garantizar que cada aspecto que se valoró con un cierto riesgo, queda cubierto y auditable
> ¿Cómo? → De muchas formas posibles.

El estándar especifica en su "Anexo A" el listado completo de cada uno de ellos, agrupándolos en los once rubros que mencionamos antes y a su vez la norma 27002, como dijimos, los detalla y para cada uno de ellos define el objetivo y lo describe brevemente.

Cabe aclarar que el anexo A de la 27001 proporciona una buena base de referencia, no siendo exhaustivo, por lo tanto se pueden seleccionar más aún. Es decir, estos 133 controles (hoy) son los mínimos que se deberán aplicar, o justificar su no aplicación, pero esto no da por completa la aplicación de la norma si dentro del proceso de análisis de riesgos aparecen aspectos que quedan sin cubrir por algún tipo de control. Por lo tanto, si a través de la evaluación de riesgos se determina que es necesaria la creación de nuevos controles, la implantación del SGSI impondrá la inclusión de los mismos, sino seguramente el ciclo no estará cerrado y presentará huecos claramente identificables.

Los controles que el anexo A de esta norma propone quedan agrupados y numerados de la siguiente forma:

A.5 Política de seguridad

A.6 Organización de la información de seguridad

A.7 Administración de recursos

A.8 Seguridad de los recursos humanos

A.9 Seguridad física y del entorno

A.10 Administración de las comunicaciones y operaciones

A.11 Control de accesos

A.12 Adquisición de sistemas de información, desarrollo y mantenimiento

A.13 Administración de los incidentes de seguridad

A.14 Administración de la continuidad de negocio

A.15 Cumplimiento (legales, de estándares, técnicas y auditorías)

A continuación vamos a profundizar un poco más en cada uno de ellos y, para ser más claro, se respetará la puntuación que la norma le asigna a cada uno de los controles.

A.5 Política de seguridad.

Este grupo está constituido por dos controles y es justamente el primer caso que se puede poner de manifiesto sobre el mencionado "Desconcepto" sobre lo que uno piensa que es un control, pues aquí se puede apreciar claramente la complejidad que representa el diseño, planificación, preparación, implementación y revisiones de una Política de Seguridad (la revisión es justamente el segundo control que propone)…….como se mencionó *"un Control es mucho (pero mucho), mas que eso…"*

Todo aquel que haya sido responsable alguna vez de esta tarea, sabrá de lo que se está hablando. La Política de Seguridad, para ser riguroso, en realidad debería dividirse en dos documentos:

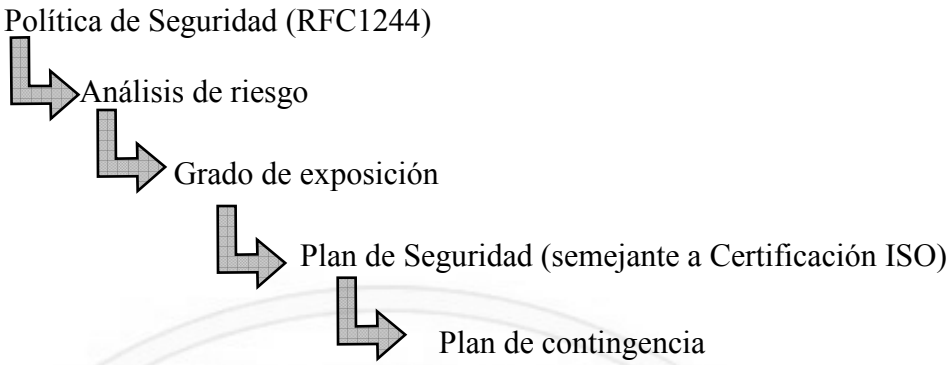
- Política de seguridad (Nivel político o estratégico de la organización): Es la mayor línea rectora, la alta dirección. Define las grandes líneas a seguir y el nivel de compromiso de la dirección con ellas.
- Plan de Seguridad (Nivel de planeamiento o táctico): Define el "Cómo". Es decir, baja a un nivel más de detalle, para dar inicio al conjunto de acciones o líneas rectoras que se deberán cumplir.

Algo sobre lo que generalmente no se suele reflexionar o remarcar es que:

Una "Política de Seguridad" bien planteada, diseñada, y desarrollada cubre la gran mayoría de los aspectos que hacen falta para un verdadero SGSI.

Haciendo abuso de la avanzada edad de este autor, es que se van a citar dos puntos de partida para la mencionada actividad que, a juicio del mismo, siguen siendo grandes referentes metodológicos a la hora de la confección de estos controles.

Se trata de lo que hemos tratado en el capítulo anterior referido a las siguientes RFCs (Request For Comments). Política de seguridad (**RFC – 2196** Site Security Handbook) y también la anterior (**RFC-1244**, que si bien queda obsoleta por la primera es muy ilustrativa) ambas, planten una metodología muy eficiente de feedback partiendo desde el plano más alto de la Organización hasta llegar al nivel de detalle, para comparar nuevamente las decisiones tomadas y reingresar las conclusiones al sistema evaluando los resultados y modificando las deficiencias. Se trata de un ciclo permanente y sin fin cuya característica fundamental es la constancia y la actualización de conocimientos. Esta recomendación os recordamos, plantea muy en grande los siguientes pasos:



La política es el marco estratégico de la Organización, es el más alto nivel. El análisis de riesgo y el grado de exposición determinan el impacto que puede producir los distintos niveles de clasificación de la información que se posee. Una vez determinado estos conceptos, se pasa al Cómo que es el Plan de Seguridad, el cual, si bien en esta RFC no está directamente relacionado con las normas ISO, se mencionan en este texto por la similitud en la elaboración de procedimientos de detalle para cada actividad que se implementa, y porque se reitera, su metodología se aprecia como excelente.

A.6 Organización de la información de seguridad.

Este segundo grupo de controles abarca once de ellos y se subdivide en:

- Organización Interna: Compromiso de la Dirección, coordinaciones, responsabilidades, autorizaciones, acuerdos de confidencialidad, contactos con autoridades y grupos de interés en temas de seguridad, revisiones independientes.
- Partes externas: Riesgos relacionados con terceros, gobierno de la seguridad respecto a clientes y socios de negocio.

Lo más importante a destacar de este grupo son dos cosas fundamentales que abarcan a ambos subgrupos:

- Organizar y Mantener actualizada la cadena de contactos (internos y externos), con el mayor detalle posible (Personas, responsabilidades, activos, necesidades, acuerdos, riesgos, etc.).
- Derechos y obligaciones de cualquiera de los involucrados.

En este grupo de controles, lo ideal es diseñar e implementar una simple base de datos, que permita de forma amigable, el alta, baja y/o modificación de cualquiera de estos campos. La redacción de la documentación inicial de responsables: derechos y obligaciones (para personal interno y ajeno) y el conjunto de medidas a adoptar con cada uno de ellos. Una vez lanzado este punto de partida, se debe documentar la metodología de actualización, auditabilidad y periodicidad de informes de la misma.

A.7 Administración de recursos

Este grupo cubre cinco controles y también se encuentra subdividido en:

- Responsabilidad en los recursos: Inventario y propietario de los recursos, empleo aceptable de los mismos.
- Clasificación de la información: Guías de clasificación y Denominación, identificación y tratamiento de la información.

Este grupo es eminentemente procedimental y no aporta nada al aspecto ya conocido en seguridad de la información, en cuanto a que todo recurso debe estar perfectamente inventariado con el máximo detalle posible, que se debe documentar el "uso adecuado de los recursos" y que toda la información deberá ser tratada de acuerdo a su nivel. En el caso de España, tanto la LOPD como la LSSI han aportado bastante a que esta tarea sea efectuada con mayor responsabilidad en los últimos años. También se puede encontrar en Internet varias referencias a la clasificación de la información por niveles.

Tal vez sí valga la pena mencionar aquí el problema que se suele encontrar en la gran mayoría de las empresas que cuentan con un parque informático considerable, sobre el cual, se les dificulta mucho el poder mantener actualizado su sistema de inventario. El primer comentario, es que este aspecto debe abordarse "**sí o sí**", pues es imposible pensar en seguridad, si no se sabe fehacientemente lo que se posee y cada elemento que queda desactualizado o no se lo ha inventariado aún, es un **hueco concreto en la seguridad de todo el sistema**, y de hecho suelen ser las mayores y más frecuentes puertas de entrada, pues están al margen de la infraestructura de seguridad.

El segundo comentario, es que se aprecia que las mejores metodologías a seguir para esta actividad, son las que permiten **mantener "vivo"** el estado de la red y por medio de ellas inventariar lo que se "escucha". Esta metodología lo que propone es, hacer un empleo lógico y completo de los elementos de red o seguridad (IDSs, Firewalls, Routers, sniffers, etc.) y aprovechar su actividad cotidiana de escucha y tratamiento de tramas para mantener "vivo" el estado de la red. Es decir, nadie mejor que ellos saben qué direcciones de la "Home Net" se encuentran activas y cuáles no, por lo tanto, aprovechar esta funcionalidad para almacenar y enviar estos datos a un repositorio adecuado, el cual será el responsable de mantener el inventario correspondiente. Sobre este tema, se propone la lectura de dos artículos publicados hace tiempo en Internet por este autor que se denominan "**Metodología Nessus-Snort**" y "**Matriz de Estado de Seguridad**", si bien los mismos deben ser actualizados al día de hoy, de ellos se puede obtener una clara imagen de cómo se puede realizar esta tarea y aprovechar las acciones de seguridad para mejorar el análisis de riesgo y el inventario.

A.8 Seguridad de los recursos humanos.

Este grupo cubre nueve controles y también se encuentra subdividido en:

- Antes del empleo: Responsabilidades y roles, verificaciones curriculares, términos y condiciones de empleo.
- Durante el empleo: Administración de responsabilidades, preparación, educación y entrenamiento en seguridad de la información, medidas disciplinarias.
- Finalización o cambio de empleo: Finalización de responsabilidades, devolución de recursos, revocación de derechos.

Este grupo, en la actualidad, debe ser el gran ausente en la mayoría de las organizaciones. Se trata de un serio trabajo a realizar entre RRHH y los responsables de Seguridad de la Información de la organización.

Se debe partir por la redacción de la documentación necesaria para la contratación de personal y la revocación de sus contratos (por solicitud, cambio o despido). En la misma deberán quedar bien claras las acciones a seguir para los diferentes perfiles de la organización, basados en la responsabilidad de manejo de información que tenga ese puesto. Como se pueda apreciar, tanto la contratación como el cese de un puesto, es una actividad conjunta de estas dos áreas, y cada paso deberá ser coordinado, según la documentación confeccionada, para que no se pueda pasar por alto ningún detalle, pues son justamente estas pequeñas omisiones de las que luego resulta el haber quedado con alta dependencia técnica de personas cuyo perfil es peligroso, o que al tiempo de haberse ido, mantiene accesos o permisos que no se debieran (casos muy comunes).

Tanto el inicio como el cese de cualquier tipo de actividad relacionada con personal responsable de manejo de información de la organización, **son actividades muy fáciles de procedimentar**, pues no dejan de ser un conjunto de acciones secuenciales muy conocidas que se deben seguir "a raja tabla" y que paso a paso deben ser realizadas y controladas……..se trata simplemente de ¡¡¡ESCRIBIRLO!!! (y por supuesto de cumplirlo luego), esto forma parte de las pequeñas cosas que cuestan poco y valen mucho ¿Por qué será que no se hacen????

En cuanto a formación, para dar cumplimiento al estándar, no solo es necesario dar cursos. Hace falta contar con un "Plan de formación". La formación en seguridad de la información, no puede ser una actividad aperiódica y determinada por el deseo o el dinero en un momento dado, tiene que ser tratada como cualquier otra actividad de la organización, es decir se debe plantear:

- Meta a la que se desea llegar.
- Determinación de los diferentes perfiles de conocimiento.
- Forma de acceder al conocimiento.
- Objetivos de la formación.
- Metodología a seguir.
- Planificación y asignación de recursos.
- Confección del plan de formación.
- Implementación del plan.
- Medición de resultados.
- Mejoras.

Si se siguen estos pasos, se llegará a la meta, pero no solo a través de la impartición de uno o varios cursos o la distribución de documentos de obligada lectura, sino con un conjunto de acciones que hará que se complementen e integren en todo el SGSI como una parte más, generando concienciación y adhesión con el mismo.

A.9 Seguridad física y del entorno

Este grupo cubre trece controles y también se encuentra subdividido en:

- Áreas de seguridad: Seguridad física y perimetral, control físico de entradas, seguridad de locales edificios y recursos, protección contra amenazas externas y del entorno, el trabajo en áreas e seguridad, accesos públicos, áreas de entrega y carga.
- Seguridad de elementos: Ubicación y protección de equipos, elementos de soporte a los equipos, seguridad en el cableado, mantenimiento de equipos, seguridad en el equipamiento fuera de la organización, seguridad en la redistribución o reutilización de equipamiento, borrado de información y/o software.

Como todo este libro propone, uno de los mejores resultados que se pueden obtener en la organización de una infraestructura de seguridad de la información, está en **plantearla siempre por niveles**. Tal vez no sea necesario hacerlo con el detalle de los siete niveles del modelo ISO/OSI, pero sí por lo menos de acuerdo al modelo TCP/IP.

Apreciamos que es correcto considerar separadamente el nivel físico con el de enlace, pues presentan vulnerabilidades muy diferentes. Si se presenta entonces el modelo de cinco niveles, se puede organizar una estructura de seguridad contemplando medidas y acciones por cada uno de ellos, dentro de las cuales se puede plantear, por ejemplo, lo siguiente:

- **Aplicación:** Todo tipo de aplicaciones.
- **Transporte:** Control de puertos UDP y TCP.
- **Red:** Medidas a nivel protocolo IP e ICMP, túneles de nivel 3.
- **Enlace:** Medidas de segmentación a nivel direccionamiento MAC, tablas estáticas y fijas en switchs, control de ataques ARP, control de broadcast y multicast a nivel enlace, en el caso WiFi: verificación y control de enlace y puntos de acceso, 802.X (Varios), empleo de túneles de nivel 2, etc.
- **Físico:** Instalaciones, locales, seguridad perimetral, CPDs, gabinetes de comunicaciones, control de acceso físico, conductos de comunicaciones, cables, fibras ópticas, radio enlaces, centrales telefónicas, etc.

Como se puede apreciar, este es una buena línea de pensamiento para plantear cada una de las actividades y evitar que se solapen algunas de ellas y/o que queden brechas de seguridad.

En el caso físico, es conveniente también separar todas ellas, por lo menos en los siguientes documentos:

- Documentación de control de accesos y seguridad perimetral general, áreas de acceso y entrega de materiales y documentación, zonas públicas, internas y restringidas, responsabilidades y obligaciones del personal de seguridad física.
- Documentación de CPDs: Parámetros de diseño estándar de un CPD, medidas de protección y alarmas contra incendios/humo, caídas de tensión, inundaciones, control de climatización (Refrigeración y ventilación), sistemas vigilancia y control de accesos, limpieza, etc.

- Documentación y planos de instalaciones, canales de comunicaciones, cableado, enlaces de radio, ópticos u otros, antenas, certificación de los mismos, etc.
- Empleo correcto del material informático y de comunicaciones a nivel físico: Se debe desarrollar aquí cuales son las medidas de seguridad física que se debe tener en cuenta sobre los mismos (Ubicación, acceso al mismo, tensión eléctrica, conexiones físicas y hardware permitido y prohibido, manipulación de elementos, etc.). No se incluye aquí lo referido a seguridad lógica.
- Seguridad física en el almacenamiento y transporte de material informático y de comunicaciones: Zonas y medidas de almacenamiento, metodología a seguir para el ingreso y egreso de este material, consideraciones particulares para el transporte del mismo (dentro y fuera de al organización), personal autorizado a recibir, entregar o sacar material, medidas de control. No se incluye aquí lo referido a resguardo y recuperación de información que es motivo de otro tipo de procedimientos y normativa.
- Documentación de baja, redistribución o recalificación de elementos: Procedimientos y conjunto de medidas a seguir ante cualquier cambio en el estado de un elemento de Hardware (Reubicación, cambio de rol, venta, alquiler, baja, destrucción, compartición con terceros, incorporación de nuevos módulos, etc.).

A.10 Administración de las comunicaciones y operaciones

Este grupo comprende treinta y dos controles, es el más extenso de todos y se divide en:

- Procedimientos operacionales y responsabilidades: Tiene como objetivo asegurar la correcta y segura operación de la información, comprende cuatro controles. Hace especial hincapié en documentar todos los procedimientos, manteniendo los mismos y disponibles a todos los usuarios que los necesiten, segregando adecuadamente los servicios y las responsabilidades para evitar uso inadecuado de los mismos.

  Esta tarea en todas las actividades de seguridad (no solo informática), se suele realizar por medio de lo que se denomina Procedimientos Operativos Normales (PON) o Procedimientos Operativos de Seguridad (POS), y en definitiva consiste en la realización de documentos breves y ágiles, que dejen por sentado la secuencia de pasos o tareas a llevar a cabo para una determinada función. Cuanto mayor sea el nivel de desagregación de esta función, más breve será cada PON (también habrá mayor cantidad de ellos) y a su vez más sencillo y comprensible. Luego de trabajar algún tiempo en esta actividad, se llegará a comprender que la mayoría de las actividades relacionadas con seguridad, son fácilmente descriptibles, pues suelen ser una secuencia de pasos bastante "mecanizables", y allí radica la importancia de estos procedimientos. La enorme ventaja que ofrece poseer todo procedimentado es:

  - Identificar con absoluta claridad los responsables y sus funciones.
  - Evitar la "imprescindibilidad" de ciertos administradores.
  - Evitar ambigüedades de procedimientos.

- Detectar "zonas grises" o ausencias procedimentales (futuras brechas de seguridad).

- Administración de prestación de servicios de terceras partes: Abarca tres controles, se refiere fundamentalmente, como su nombre lo indica, a los casos en los cuales se encuentran externalizadas determinadas tareas o servicios del propio sistema informático. Los controles están centrados en tres aspectos fundamentales de esta actividad:

  - Documentar adecuadamente los servicios que se están prestando (acuerdos, obligaciones, responsabilidades, confidencialidad, operación, mantenimiento, etc.).
  - Medidas a adoptar para la revisión, monitorización y auditoría de los mismos
  - Documentación adecuada que permita regularizar y mantener un eficiente control de cambios en estos servicios.

- Planificación y aceptación de sistemas: El objetivo es realizar una adecuada metodología para que al entrar en producción cualquier sistema, se pueda minimizar el riesgo de fallos. De acuerdo a la magnitud de la empresa y al impacto del sistema a considerar, siempre es una muy buena medida la realización de maquetas. Estas maquetas deberían "acercarse" todo lo posible al entorno en producción, para que sus pruebas de funcionamiento sean lo más veraces posibles, simulando ambientes de trabajo lo más parecidos al futuro de ese sistema (Hardware y Software, red, carga de operaciones y transacciones, etc.), cuanto mejor calidad y tiempo se dedique a estas maquetas, menor será la probabilidad de fallos posteriores, es una relación inversamente proporcional que se cumple en la inmensa mayoría de los casos.

  Los dos aspectos claves de este control son el diseño, planificación, prueba y adecuación de un sistema por un lado; y el segundo, es desarrollar detallados criterios de aceptación de nuevos sistemas, actualizaciones y versiones que deban ser implantados. Este último aspecto será un documento muy "vivo", que se realimentará constantemente en virtud de las modificaciones, pruebas, incorporaciones y avances tecnológicos, por lo tanto se deberá confeccionar de forma flexible y abierto a permanentes cambios y modificaciones.

- Protección contra código móvil y maligno: el objetivo de este apartado es la protección de la integridad del software y la información almacenada en los sistemas.

  El código móvil es aquel que se transfiere de un equipo a otro para ser ejecutado en el destino final, este empleo es muy común en las arquitecturas cliente-servidor, y se está haciendo más común en las arquitecturas "víctima-gusano", por supuesto con un empleo no tan deseado. Sobre el empleo seguro de Código móvil, recomendamos que el que esté interesado, profundice en una metodología que esta haciendo las cosas bien, que se denomina "**Proof-Carring Code**" (**PCC**), la cual propone la implementación de medidas para garantizar que los programas que serán ejecutados en el cliente lo hagan de forma segura. Se puede encontrar mucha información al respecto en Internet.

  En cuanto al código malicioso, a esta altura no es necesario ahondar en ningún detalle al respecto, pues "quien esté libre de virus y troyanos que tire la primera piedra". El estándar hace referencia al conjunto de medidas comunes que ya suelen ser aplicadas en

la mayoría de las empresas, es decir, detección, prevención y recuperación de la información ante cualquier tipo de virus. Tal vez lo más importante aquí y suele ser el punto débil de la gran mayoría es la preparación y la existencia de procedimientos (Lo que implica practicarlos). En nuestra opinión es donde más frecuentemente se encuentran fallos. **La gran mayoría de las empresas confían su seguridad antivirus en la mera aplicación de un determinado producto y nada más**, pero olvidan preparar al personal de administradores y usuarios en cómo proceder ante virus y, por supuesto, tampoco realizan procedimientos de recuperación y verificación del buen funcionamiento de lo documentado (si es que lo tienen…..). Esto último, es una de las primeras y más comunes objeciones que aparecen al solicitar certificaciones en estos estándares. Lo recalcamos una vez más, no es eficiente el mejor producto antivirus del mercado, sino se realizan estas dos últimas tareas que se han mencionado: Preparación del personal e implementación (con prácticas) de procedimientos.

- Resguardo: El objetivo de esta apartado conceptualmente es muy similar al anterior, comprende un solo control que remarca la necesidad de las copias de respaldo y recuperación. Siguiendo la misma insistencia del párrafo precedente, de nada sirve realizar copias de respaldo y recuperación, sino se prepara al personal e implementan las mismas con prácticas y procedimientos
- Administración de la seguridad de redes: Los dos controles que conforman este apartado hacen hincapié en la necesidad de administrar y controlar lo que sucede en nuestra red, es decir, implementar todas las medidas posibles para evitar amenazas, manteniendo la seguridad de los sistemas y aplicaciones a través del conocimiento de la información que circula por ella. Se deben implementar controles técnicos, que evalúen permanentemente los servicios que la red ofrece, tanto propios como externalizados.
- Manejo de medios: En esta traducción, como "medio" debe entenderse todo elemento capaz de almacenar información (discos, cintas, papeles, etc. tanto fijos como removibles). Por lo tanto el objetivo de este grupo es, a través de sus cuatro controles, prevenir la difusión, modificación, borrado o destrucción de cualquiera de ellos o lo que en ellos se guarda.

  En estos párrafos describe brevemente las medidas a considerar para administrar medios fijos y removibles, su almacenamiento seguro y también por períodos prolongados, evitar el uso incorrecto de los mismos y un control específico para la documentación.

  De todo ello, lo que debe rescatarse especialmente, es el planteo de este problema de los medios, procedimentarlo, practicarlo y mejorarlo con la mayor frecuencia que se pueda.
- Intercambios de información: Este grupo contempla el conjunto de medidas a considerar para cualquier tipo de intercambio de información, tanto en línea como fuera de ella, y para movimientos internos o externos de la organización.

  Aunque a primera vista no lo parezca, son muchos los aspectos que deben ser tenidos en cuenta para esta tarea. No debe olvidarse que la información es **el bien más preciado de una empresa, por lo tanto al igual que en un Banco, cuando la misma se mueve, no es nada más ni nada menos que un "desplazamiento de caudales"**, es decir con todas las prevenciones, vigilancias, procesos, agentes especializados/entrenados

y…… hasta con camión blindado. Los aspectos que no se pueden dejar librados al azar son:

- Políticas, procedimientos y controles para el intercambio de información para tipo y medio de comunicación a emplear.
- Acuerdos, funciones, obligaciones, responsabilidades y sanciones de todas las partes intervinientes.
- Medidas de protección física de la información en tránsito.
- Consideraciones para los casos de mensajería electrónica
- Medidas particulares a implementar para los intercambios de información de negocio, en especial con otras empresas.

- Servicios de comercio electrónico: Este grupo, supone que la empresa sea la prestadora de servicios de comercio electrónico, es decir, no aplica a que los empleados realicen una transacción, por parte de la empresa o por cuenta propia, con un servidor ajeno a la misma.

  La prestación de servicios de comercio electrónico por parte de una empresa exige el cumplimiento de varios detalles desde el punto de vista de la seguridad:

  - Metodologías seguras de pago.
  - Confidencialidad e integridad de la transacción.
  - Mecanismos de no repudio.
  - Garantías de transacción.
  - Conformidades legales, desde el punto de vista de LSSI y LOPD (en España), como así también del código de Comercio.

  Para el cumplimiento de lo expuesto es que este grupo presenta, a través de tres controles, un conjunto de medidas a considerar referidas al control de información que circula a través de redes públicas para evitar actividades fraudulentas, difusión, modificación o mal uso de la misma. Medidas tendientes a evitar transacciones incompletas, duplicaciones o réplicas de las mismas, y por último mecanismos que aseguren la integridad de la totalidad de la información disponible.

- Monitorización: Este apartado tiene como objetivo la detección de actividades no autorizadas en la red y reúne seis controles. Los aspectos más importantes a destacar son:

  - Auditar Logs que registren actividad, excepciones y eventos de seguridad.
  - Realizar revisiones periódicas y procedimientos de monitorización del uso de los sistemas.
  - Implementación de robustas medidas de protección de los Logs de información de seguridad. Se debe considerar que una de las primeras enseñanzas que recibe cualquier aprendiz de intruso, es a borrar sus huellas para poder seguir operando sin ser descubierto el mayor tiempo posible. Por esta razón, es una de las principales tareas de seguridad, la de proteger todo indicio o prueba de

actividad sospechosa. En casos de máxima seguridad, se llega hasta el extremo de tener una impresora en línea, que va registrando sobre papel en tiempo real, cada uno de los logs que se configuran como críticos, pues es uno de los pocos medios que no puede ser borrado remotamente una vez detectada la actividad. También como experiencia personal, hemos llegado a ver realizar la misma actividad en CDs de una sola escritura.

- La actividad de los administradores y operadores de sistemas, también debe ser monitorizada, pues es una de las mejores formas de tomar conocimiento de actividad sospechosa, tanto si la hace un administrador propio de la empresa (con o sin mala intención) o si es alguien que se hace pasar por uno de ellos. Hay que destacar que una vez que se posee acceso a una cuenta de administración, se tiene control total de esa máquina y en la mayoría de los casos, ya está la puerta abierta para el resto de la infraestructura, es decir, se emplea esa máquina como puente o máquina de salto hacia las demás.
- Así como es vital, ser estricto con el control de Logs, lo es también el saber lo antes posible si cualquiera de ellos está fallando, pues esa debe ser la segunda lección de un aprendiz de intruso. Es decir, lograr que se dejen de generar eventos de seguridad por cada paso que da. Por lo tanto, es necesario implementar un sistema de alarmas que monitorice el normal funcionamiento de los sistemas de generación de eventos de seguridad y/o Logs.
- Sincronización de tiempos: Hoy en día el protocolo **NTP** (Network Time Protocol) está tan difundido y fácilmente aplicable que es un desperdicio no usarlo. Se debe implementar una buena estrategia de estratos, tal cual lo propone este protocolo, y sincronizar toda la infraestructura de servidores, tanto si se depende de ellos para el funcionamiento de los servicios de la empresa, como si no. Pues cuando llega la hora de investigar, monitorizar o seguir cualquier actividad sospechosa es fundamental tener una secuencia cronológica lógica que permita moverse pro todos los sistemas de forma coherente. De acuerdo a la actividad de la empresa, se deberá ser más o menos estrictos con la precisión del reloj del máximo estrato (hacia el exterior) el cual puede soportar mayor flexibilidad en los casos que no sea vital su exactitud con las jerarquías internacionales y luego el resto de las máquinas dependerán de este. Pero lo que no debe suceder y así lo exige esta norma, es la presencia de servidores que no estén sincronizados.

A.11 Control de accesos

No se debe confundir la actividad de control de accesos con autenticación, esta última tiene por misión identificar que verdaderamente "sea, quien dice ser". El control de acceso es posterior a la autenticación y debe regular que el usuario autenticado, acceda únicamente a los recursos sobre los cuales tenga derecho y a ningún otro, es decir que tiene dos tareas derivadas:

- Encauzar (o enjaular) al usuario debidamente.

- Verificar el desvío de cualquier acceso, fuera de lo correcto.

El control de acceso es una de las actividades más importantes de la arquitectura de seguridad de un sistema. Al igual que sucede en el mundo de la seguridad física, cualquier persona que ha tenido que acceder a una caja de seguridad bancaria nota como a medida que se va llegando a áreas de mayor criticidad, las medidas de control de acceso se incrementan, en un sistema informático debería ser igual.

Para cumplir con este propósito, este apartado lo hace a través de veinticinco controles, que los agrupa de la siguiente forma:

- Requerimientos de negocio para el control de accesos: Debe existir una Política de Control de accesos documentada, periódicamente revisada y basada en los niveles de seguridad que determine el nivel de riesgo de cada activo.
- Administración de accesos de usuarios: Tiene como objetivo asegurar el correcto acceso y prevenir el no autorizado y, a través de cuatro controles, exige llevar un procedimiento de registro y revocación de usuarios, una adecuada administración de los privilegios y de las contraseñas de cada uno de ellos, realizando periódicas revisiones a intervalos regulares, empleando para todo ello procedimientos formalizados dentro de la organización.
- Responsabilidades de usuarios: Todo usuario dentro de la organización debe tener documentadas sus obligaciones dentro de la seguridad de la información de la empresa. Independientemente de su jerarquía, siempre tendrá alguna responsabilidad a partir del momento que tenga acceso a la información. Evidentemente existirán diferentes grados de responsabilidad, y proporcionalmente a ello, las obligaciones derivadas de estas funciones. Lo que **no** puede suceder es que algún usuario las desconozca. Como ningún ciudadano desconoce por ejemplo, las medidas de seguridad vial, pues el tráfico sería caótico (¿más aún????), de igual forma no es admisible que el personal de la empresa no sepa cuál es su grado de responsabilidad en el manejo de la información de su nivel. Por lo tanto de este ítem se derivan tres actividades.
    - Identificar niveles y responsabilidades.
    - Documentarlas correctamente.
    - Difundirlas y verificar su adecuada comprensión.

  Para estas actividades propone tres controles, orientados a que los usuarios deberán aplicar un correcto uso de las contraseñas, ser conscientes del equipamiento desatendido (por lugar, horario, lapsos de tiempo, etc.) y de las medidas fundamentales de cuidado y protección de la información en sus escritorios, medios removibles y pantallas.
- Control de acceso a redes: Todos los servicios de red deben ser susceptibles de medidas de control de acceso; para ello a través de siete controles, en este grupo se busca prevenir cualquier acceso no autorizado a los mismos.

  Como primera medida establece que debe existir una política de uso de los servicios de red para que los usuarios, solo puedan acceder a los servicios específicamente autorizados. Luego se centra en el control de los accesos remotos a la organización, sobre los cuales deben existir medidas apropiadas de autenticación.

Un punto sobre el que merece la pena detenerse es sobre la identificación de equipamiento y de puertos de acceso. Este aspecto es una de las principales medidas de control de seguridad. En la actualidad se poseen todas las herramientas necesarias para identificar con enorme certeza las direcciones, puertos y equipos que pueden o no ser considerados como seguros para acceder a las diferentes zonas de la empresa. Tanto desde una red externa como desde segmentos de la propia organización. En los controles de este grupo menciona medidas automáticas, segmentación, diagnóstico y control equipamiento, direcciones y de puertos, control de conexiones y rutas de red. Para toda esta actividad se deben implementar: IDSs, IPSs, FWs con control de estados, honey pots, listas de control de acceso, certificados digitales, protocolos seguros, túneles, etc… Es decir, existen hoy en día muchas herramientas para implementar estos controles de la mejor forma y eficientemente, por ello, tal vez este sea uno de los grupos que más exigencia técnica tiene dentro de este estándar.

- Control de acceso a sistemas operativos: El acceso no autorizado a nivel sistema operativo presupone uno de los mejores puntos de escalada para una intrusión; de hecho son los primeros pasos de esta actividad, denominados "*Fingerprintig y footprinting*", pues una vez identificados los sistemas operativos, versiones y parches, se comienza por el más débil y con solo conseguir un acceso de usuario, se puede ir escalando en privilegios hasta llegar a encontrar el de "*root*", con lo cual ya no hay más que hablar. La gran ventaja que posee un administrador, es que las actividades sobre un sistema operativo son mínimas, poco frecuentes sus cambios, y desde ya que no comunes a nivel usuario del sistema, por lo tanto si se saben emplear las medidas adecuadas, se puede identificar rápidamente cuando la actividad es sospechosa, y en definitiva es lo que se propone en este grupo: Seguridad en la validación de usuarios del sistema operativo, empleo de identificadores únicos de usuarios, correcta administración de contraseñas, control y limitación de tiempos en las sesiones y por último verificaciones de empleo de utilidades de los sistemas operativos que permitan realizar acciones "interesantes".
- Control de acceso a información y aplicaciones: En este grupo, los dos controles que posee están dirigidos a prevenir el acceso no autorizado a la información mantenida en las aplicaciones. Propone redactar, dentro de la política de seguridad, las definiciones adecuadas para el control de acceso a las aplicaciones y a su vez el aislamiento de los sistemas sensibles del resto de la infraestructura. Este último proceder es muy común en sistemas críticos (Salas de terapia intensiva, centrales nucleares, servidores primarios de claves, sistemas de aeropuertos, militares, etc.), los cuales no pueden ser accedidos de ninguna forma vía red, sino únicamente estando físicamente en ese lugar. Por lo tanto si se posee alguna aplicación que entre dentro de estas consideraciones, debe ser evaluada la necesidad de mantenerla o no en red con el resto de la infraestructura.
- Movilidad y tele trabajo: Esta nueva estructura laboral, se está haciendo cotidiana en las organizaciones y presenta una serie de problemas desde el punto de vista de la seguridad:
    - Accesos desde un ordenador de la empresa, personal o público.
    - Posibilidades de instalar o no, medidas de hardware/software seguro en el ordenador remoto.
    - Canales de comunicaciones por los cuales se accede (red publica, privada, GPRS, UMTS, WiFi, Túnel, etc.).

- Contratos que se posean sobre estos canales.
- Personal que accede: propio, externalizado, o ajeno.
- Lugar remoto: fijo o variable.
- Aplicaciones e información a la que accede.
- Nivel de profundidad en las zonas de red a los que debe acceder.
- Volumen y tipo de información que envía y recibe.
- Nivel de riesgo que se debe asumir en cada acceso.

Cada uno de los aspectos expuestos merece un tratamiento detallado y metodológico, para que no surjan nuevos puntos débiles en la estructura de seguridad.

La norma no entra en mayores detalles, pero de los dos controles que propone se puede identificar que la solución a esto es adoptar una serie de procedimientos que permitan evaluar, implementar y controlar adecuadamente estos aspectos en el caso de poseer accesos desde ordenadores móviles y/o tele trabajo.

A.12 Adquisición de sistemas de información, desarrollo y mantenimiento

Este grupo reúne dieciséis controles.

- Requerimientos de seguridad de los sistemas de información: Este primer grupo que incluye un solo control, plantea la necesidad de realizar un análisis de los requerimientos que deben exigirse a los sistemas de información, desde el punto de vista de la seguridad para cumplir con las necesidades del negocio de cada empresa en particular, para poder garantizar que la seguridad sea una parte integral de los sistemas.
- Procesamiento correcto en aplicaciones: En este grupo se presentan cuatro controles, cuya misión es el correcto tratamiento de la información en las aplicaciones de la empresa. Para ello las medidas a adoptar son, validación en la entrada de datos, la implementación de controles internos en el procesamiento de la información para verificar o detectar cualquier corrupción de la información a través de los procesos, tanto por error como intencionalmente, la adopción de medidas par asegurar y proteger los mensajes de integridad de las aplicaciones. Y por último la validación en la salida de datos, para asegurar que los datos procesados, y su posterior tratamiento o almacenamiento, sea apropiado a los requerimientos de esa aplicación.
- Controles criptográficos: Nuevamente se recalca este objetivo de la criptografía de proteger la integridad, confidencialidad y autenticidad de la información. En este caso, a través de dos controles, lo que propone es desarrollar una adecuada política de empleo de estos controles criptográficos y administrar las calves que se emplean de forma consciente.

  El tema de claves criptográficas, como se ha podido apreciar hasta ahora, es un denominador común de toda actividad de seguridad, por lo tanto más aún cuando lo que se pretende es implementar un completo SGSI, por lo tanto es conveniente y muy recomendable dedicarle la atención que evidentemente merece, una muy buena medida

es desarrollar un documento que cubra todos los temas sobre los cuales los procesos criptográficos participarán de alguna forma y desde el mismo referenciar a todos los controles de la norma en los cuales se hace uso de claves. Este documento "rector" de la actividad criptográfica, evitará constantes redundancias y sobre todo inconsistencias en la aplicación de claves.

- Seguridad en los sistemas de archivos: La Seguridad en los sistemas de archivos, independientemente que existan sistemas operativos más robustos que otros en sus técnicas de archivos y directorios, es una de las actividades sobre las que se debe hacer un esfuerzo técnico adicional, pues en general existen muchas herramientas para robustecerlos, pero no suelen usarse. Es cierto que los sistemas de archivos no son un tema muy estático, pues una vez que un sistema entra en producción suelen hacerse muchas modificaciones sobre los mismos, por esto último principalmente es que una actividad que denota seriedad profesional, es la identificación de **¿Cuáles son los directorios o archivos que no deben cambiar y cuáles sí?** Esta tarea la hemos visto en muy, pero muy pocas organizaciones, y podemos asegurar que es una de las que mayores satisfacciones proporciona en el momento de "despertar sospechas" y restaurar sistemas. Suele ser el mejor indicador de una actividad anómala, si se ha planteado bien el interrogante anteriormente propuesto. Si se logra identificar estos niveles de "estaticidad y cambio" y se colocan los controles y auditorías periódicas y adecuadas sobre los mismos, esta será una de las alarmas de la que más haremos uso a futuro en cualquier etapa de un incidente de seguridad, y por supuesto será la mejor herramienta para restaurar un sistema a su situación inicial.

  Este grupo de tres controles, en definitiva lo que propone es justamente esto, control de software operacional, test de esos datos y controlar el acceso al código fuente.

- Seguridad en el desarrollo y soporte a procesos: Este apartado cubre cinco controles cuya finalidad está orientada hacia los cambios que sufre todo sistema. Los aspectos calve de este grupo son:

  - Desarrollar un procedimiento de control de cambios.
  - Realización de revisiones técnicas a las aplicaciones luego de realizar cualquier cambio, teniendo especial atención a las aplicaciones críticas.
  - Documentar claramente las restricciones que se deben considerar en los cambios de paquetes de software.
  - Implementación de medidas tendientes a evitar fugas de información.
  - Supervisión y monitorización de desarrollos de software externalizado.

- Administración técnica de vulnerabilidades: Toda vulnerabilidad que sucede en un sistema de información, tarde o temprano se describe con todo lujo de detalles en Internet. Las palabras claves de esto son "tarde o temprano", pues cuanto antes se tenga conocimiento de una debilidad y las medidas adecuadas para solucionarlas, mejor será para la organización.

  Este grupo que solo trata un solo control, lo que propone es adoptar medidas para estar al tanto de estos temas más "temprano" que "tarde". Esta actividad, en la actualidad no requiere esfuerzos económicos si se pone interés en la misma, pero sí requiere mucho

tiempo para poder consultar Web especializadas o leer los mails que llegan si se está subscripto a grupos de noticias de seguridad, o buscar en Internet en foros, etc. Lo importante es que existe un amplio abanico de posibilidades para realizar esta tarea, que va desde hacerlo individualmente hasta externalizarla, y a su vez desde hacerlo "Muy temprano" hasta "demasiado tarde" y en todo este abanico se pueden elegir un sinnúmero de opciones intermedias.

La norma simplemente nos aconseja plantearse formalmente el tema, análisis de la relación coste/beneficio en la empresa para esta tarea y adoptar una decisión coherente dentro del abanico expuesto.

A.13 Administración de los incidentes de seguridad

Todo lo relativo a incidentes de seguridad queda resumido como ya hemos dicho a dos formas de proceder:

- Proteger y proceder.
- Seguir y perseguir.

Tal vez no sea la mejor traducción de estos dos procederes, pero lo que trata de poner de manifiesto es que ante un incidente, quien no posea la capacidad suficiente solo puede "Proceder y proteger", es decir cerrar, apagar, desconectar, etc… con ello momentáneamente solucionará el problema, pero al volver sus sistemas al régimen de trabajo normal el problema tarde o temprano volverá pues no se erradicaron sus causas. La segunda opción, en cambio, propone verdaderamente "Convivir con el enemigo", y permite ir analizando paso a paso su accionar, llegar a comprender todo el detalle de su tarea y entonces sí erradicarlo definitivamente. Por supuesto este último trabajo, requiere estar preparado y contar con los medios y recursos suficientes.

En definitiva, es esto lo que trata de dejar claro este punto de la norma a través de los cinco controles que agrupa, y subdivide en:

- Reportes de eventos de seguridad de la información y debilidades.

  Como su nombre lo indica, este apartado define el desarrollo de una metodología eficiente para la generación, monitorización y seguimiento de reportes, los cuales deben reflejar, tanto eventos de seguridad como debilidades de los sistemas. Estas metodologías deben ser ágiles, por lo tanto se presupone el empleo de herramientas automatizadas que lo hagan. En estos momentos se poseen muchas de ellas.

  En concreto para que estos controles puedan funcionar de manera eficiente, lo mejor es implantar herramientas de detección de vulnerabilidades, ajustarlas a la organización, para saber con total certeza dónde se es débil y donde no, y a través de estas desarrollar un mecanismo simple de difusión de las mismas a los responsables de su administración y solución, los cuales deberán solucionarlas o justificar las causas para no hacerlo, ante lo cual, esta debilidad pasará a ser tratada por el segundo grupo de este control, es decir una metodología de detección de intrusiones, que será la responsable de generar la alerta temprana, cuando una de esas debilidades sea explotada por personal no autorizado. Estas alertas necesitan también un muy buen mecanismo de gestión, para provocar la respuesta inmediata.

- Administración de incidentes de seguridad de la información y mejoras.

  Si se poseen los dos mecanismos mencionados en el punto anterior, la siguiente tarea es disponer de una metodología de administración de incidentes, lo cual no es nada más que un procedimiento que describa claramente: pasos, acciones, responsabilidades, funciones y medidas concretas. Todo esto no es eficaz si no se realiza la preparación adecuada, por lo tanto es necesario difundirlo, practicarlo y SIMULARLO, es decir generar incidentes que no hagan peligrar los elementos en producción, tanto sobre maquetas como en planta y poner a prueba todos los eslabones de la metodología. Seguramente aparecerán fallos, zonas grises o brechas de seguridad metodológicas, las cuales la mejor manera de solucionarlas es en "situaciones de paz" y no durante un conflicto real…… como se pueda apreciar hemos escrito en terminología muy militar, pues esto no es ni más ni menos que lo que hacen (o deberían hacer….) durante todo el tiempo de paz las fuerzas armadas, "prepararse para incidencias", debido a que esta actividad no puede ser improvisada cuando llega la misma, no hace falta ser militar para deducir que será catastrófico. La preparación militar, en los casos defensivos hace principalmente esto, es decir analizar las posibles metodologías que puede aplicar un enemigo y practicar su contramedida, esto es el entrenamiento militar y a su vez son los denominados "ejercicios militares" en el terreno o en mesas de arena (*léase planta y/o maqueta*), que no son otra cosa que simulaciones sobre qué sucedería si reaccionamos de esta forma u otra. La doctrina militar es milenaria, tiene millones de situaciones vividas, practicadas y estandarizadas, por así llamarlas y, en los casos en que su analogía con la informática es evidente, no se debe re inventar la rueda, sino aprovechar lo que ya existe, y la preparación ante incidencias es uno de los casos más evidentes de esto. Existe un muy antiguo refrán que dice "Si quieres vivir en paz, prepárate para la guerra". Es decir, si quieres evitar problemas de seguridad, prepárate para ellos.

A.14 Administración de la continuidad de negocio

Este grupo cubre nuevamente cinco controles y los presenta a través de un solo grupo:

- Aspectos de seguridad de la información en la continuidad del negocio.

  Este grupo tiene como objetivo contemplar todas las medidas tendientes a que los sistemas no hagan sufrir interrupciones sobre la actividad que realiza la empresa. Hoy en día los sistemas informáticos son uno de los pilares fundamentales de toda empresa, independientemente de la actividad que realice, ya se puede afirmar que no existe ninguna que no tenga un cierto grado de dependencia con estas tecnologías. Cualquier anomalía de sus sistemas repercute en el negocio de la empresa y por supuesto esto debería ser lo mínimo posible.

  Lo primero que considera este grupo es que la seguridad de la información se encuentre incluida en la administración de la continuidad de negocio, esto que tal vez parezca muy intangible o impreciso, no es nada más que considerar los puntos o hitos en los cuales debe incluirse "controles" de seguridad dentro de los procesos de la empresa. Es decir, si una empresa conoce bien su actividad, debe ser capaz de redactar su metodología de trabajo a través de flujos o procesos (Simples diagramas de flujo). En cada una de las líneas que unen este grafo, se debe considerar la seguridad, y verificar si esta influye o no en esta secuencia, si influye es un hito de seguridad y debe ser considerado, evaluado

e implementado el control correspondiente. Este punto que se presentó unos renglones arriba como intangible, **podemos asegurar que debe ser el más importante de esta norma**, y es el que desencadenará absolutamente todos los controles de seguridad de la empresa, pues si un control no está relacionado con los procesos de la empresa, no tiene mayor sentido y, peor aún, si no se conocen con exactitud los procesos de la empresa, es muy difícil asegurar los sistemas de la misma.

Una vez detectada y analizada la inclusión de hitos o controles de seguridad en los procesos de la empresa, el segundo paso es evaluar los riesgos que impone este para la interrupción del negocio de la organización, de ese riesgo se derivará un impacto, cuyas consecuencias se deberá determinar cómo asumir.

Las medidas o determinaciones que se adopten para solucionar, minimizar, mejorar o asumir esos riesgos deberán expresarse por medio de planes de continuidad de negocio (o planes de contingencia), los cuales tienen el objetivo de mantener y restaurar el nivel operacional de la empresa, por medio de un conjunto de medidas que reflejen la forma de proceder y/o escalar ante la ocurrencia de cualquiera de los efectos que produciría un fallo en esos hitos.

Por último, al igual que el grupo anterior, todas estas medidas, deberán ser puestas a prueba para mantener "vivo" y mejorar este plan.

A.15 Marco Legal y buenas prácticas (legales, de estándares, técnicas y auditorías)

Este grupo cubre diez controles. Es uno de los aspectos más débiles que en estos momentos posee la norma, pues la aplicación de la misma en cada País, debe estar de acuerdo a las bases y regulaciones legales del mismo, las cuales sólo son consideradas, una vez que las organizaciones de estandarización correspondientes adecuan el estándar Inglés a cada País respectivo. Para poner como ejemplo, en el caso de España, no puede (o no debería) ser posible la certificación de una empresa que no de cumplimiento a la LSSI, LOPD, leyes de regulación de las telecomunicaciones, interceptación legal, etc…Estos aspectos ningún auditor certificado en BSI, ISACA internacionalmente, etc. tiene porqué conocerlos, como tampoco tendrá la base suficiente para controlarlos con la rigurosidad que esto implica, y por lo tanto, puede suceder (¿o ya sucede?....) que existan empresas que se estén certificando en esta norma y no cumplan estrictamente con las bases legales de cada País. ¿Qué sucedería si les cae una auditoría de, por ejemplo, la Agencia Protectora de Datos y resulta que no están bien en este aspecto?, ¿Seguiría siendo válida su certificación??? (huuuuummmmm…..)

Hechas las salvedades respectivas, seguimos adelante con este grupo que se encuentra subdividido en:

- Cumplimiento de requerimientos legales.

  Lo primero a considerar aquí es la identificación de la legislación aplicable a la empresa, definiendo explícitamente y documentando todo lo que guarde relación con estos aspectos. Otro componente de este primer grupo es lo relacionado con los derechos de propiedad intelectual (A ver si SGAE se enoja, cosa que puede ser de gravísimas consecuencias…), debiendo generar procedimientos que aseguren el

cumplimiento de las regulaciones, el punto tal vez más destacable aquí es el referido al empleo de software legal, su concienciación y difusión.

Todos los registros que guarden algún tipo de información clasificada desde el punto de vista legal, deben ser protegidos para evitar pérdidas, alteraciones y un aspecto muy importante: "divulgación inadecuada", en particular, y esto ya es una regulación generalizada en todos los Países, los registros de carácter personal y de ello lo más importante es lo que se puede considerar como datos íntimos en el caso de tener necesidad de almacenarlos, como pueden ser enfermedades, discapacidades, orientación religiosa, sexual, política, etc.

Para todos estos registros, se deben implementar todos los procedimientos necesarios para prevenir su procesamiento incorrecto, pues se pueden haber considerado todos los aspectos legales en su guarda y custodia, pero al momento de ser procesados, quedan expuestos (memorias temporales, permanencia exterior, o transmisión insegura, etc.), o sus resultados, quedan fuera del perímetro o las evaluaciones de seguridad que fueron realizadas sobre los registros. Por lo tanto, para todo registro deberá ser identificado, analizado, implementado y documentado, todos los aspectos legales que le aplican, durante el almacenamiento y también en todo momento en que sea requerido para su procesamiento (incluyendo aquí sus desplazamientos).

El último control de este grupo hace referencia a las regulaciones legales que aplican al uso de controles criptográficos. Hoy en día a nuestro juicio, la ley aplica a tres aspectos de la criptografía:

- El tema de exportación de claves cuyo máximo exponente fue EEUU (hoy en franca decadencia).
- El tema de requerimientos legales sobre registros almacenados y/o en tránsito (Interceptación legal).
- El empleo de claves por parte de los usuarios y administradores de sistemas. Este aspecto es muy pocas veces considerado en las organizaciones, y he conocido ya varios casos de problemas y pleitos legales, por ejemplo, sobre despidos en los cuales una deficiente política de derechos y obligaciones legales de la empresa hacia sus empleados implicó importantes sumas de dinero para poder retomar el acceso/control a sus infraestructuras, y/o descifrar información que sólo estaban en capacidad de hacerlo ciertos empleados. Esto es un aspecto legal que debe ser claramente definido y puesto en conocimiento del personal.

- Cumplimiento de políticas de seguridad, estándares y técnicas de buenas prácticas.

  En este grupo a través de dos controles, la norma trata de hacer hincapié en el control del cumplimiento de estas medidas, pues de nada sirve tener todo en regla con los aspectos legales, si luego el personal involucrado no da cumplimiento a las medidas y en definitiva, la implementación falla. Para evitar estas debilidades y los graves problemas que pueden ocasionar, es que se debe asegurar que todos estos procedimientos se cumplan y verificar periódicamente que las regulaciones estén vigentes, sean aplicables y estén de acuerdo con toda la organización.

- Consideraciones sobre auditorías de sistemas de información.

Las auditorías de los sistemas de información son imprescindibles, las dos grandes consideraciones son realizarla de forma externa o interna. Cuando se contrata este servicio a través de empresas externas, los resultados son mejores, pues son su especialidad y por lo tanto tienen en "Know How" necesario y suficiente para detectar los aciertos y errores, la parte negativa es que por los recursos económicos que implica, no pueden ser todo lo periódicas que se desean. Por otro lado, las auditorías internas, no poseen tal vez tanta "expertiz", pero por realizarse con recursos propios, ofrecen la posibilidad de realizarlas con mayor periodicidad e inclusive realizarlas aleatoriamente lo cual suele ser muy efectivo. El aspecto fundamental de una auditoría interna es que no puede estar involucrado el personal responsable de lo que se audita, es decir, no se puede ser "Juez y parte", remarco esto pues, por evidente que parezca no suele cumplirse muy a menudo.

El último tema que considera el estándar es lo referido al empleo de herramientas de auditoría de seguridad. Este es un tema de vital interés desde varios aspectos:

- Una herramienta de auditoría de seguridad instalada, puede servir para el lado bueno o el "oscuro" de la organización. Por lo tanto, las mismas deberán ser tratadas con todas las precauciones (Inventariadas, identificadas, controladas en su acceso, monitorizadas, desinstaladas, etc.). Pues si un usuario no autorizado, accede a ellas, se le está sirviendo la red en bandeja de plata.
- El empleo de una herramienta de auditoría de seguridad debe ser perfectamente regulado, en cuanto a su alcance, profundidad, potencialidad, horario, fechas, ventanas de tiempo de operación, objetivo, resultados deseados, etc. Pues al igual que en el punto anterior, no puede dejarse librado al azar su uso correcto, caso contrario se puede disparar todo un procedimiento de incidencias, o caerse una infraestructura, etc.
- Se debe coordinar con cada sistema a auditar qué es exactamente lo que se va a hacer sobre este y cuales son los derechos y obligaciones que se poseen en el uso de esa herramienta sobre cada sistema en particular, pues no tiene porqué ser el mismo para todos.
- Se debe regular CONTRACTUALMENTE, en los casos de auditorías externas cuáles son los derechos, obligaciones y responsabilidades en el empleo de las mismas, incluyendo claramente las indemnizaciones por daños y perjuicios que pueden ocasionar en su empleo incorrecto.

8.5.7. Implantación y certificación de ISO 27001

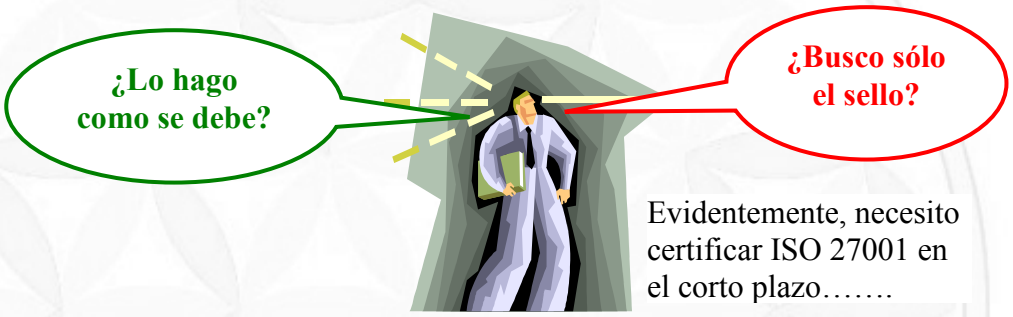
- Propuesta para la implantación de un SGSI.

El principal objetivo de esta sección es ofrecer un claro curso de acción para las PyMEs. No está orientado a las grandes empresas, pues cualquiera de ellas está en condiciones de contratar una consultoría externa y desentenderse del tema (....grave error), asumiendo

también los grandes costes que ello implica. Por esta razón, es que toda PyME debe hacer una fuerte diferencia entre la "**Necesidad de certificar**" y el "**Negocio de la Certificación**", bien entendidas estas posturas es lo que les permitirá implementar a las PyMEs la mayoría de los puntos de este estándar, con gran independencia del "Negocio de la Certificación" que se gesta alrededor de todo estándar certificable.

> Para seros sinceros, si se poseen los conocimientos y capacidades necesarias, afirmaríamos que **se puede "Dibujar" una certificación ISO 27001** (y lo que acabamos de afirmar, es muy, pero muy atrevido….). Lo que también afirmamos y con mucha más contundencia, es **que será imposible de mantener esta mentira**.

Lo que se trata de reflejar en el cuadro anterior es la decisión que deberá adoptar todo responsable de sistemas en los próximos años, es decir:



Evidentemente, necesito certificar ISO 27001 en el corto plazo…….

Tal vez parezca cruel, o poco serio, pero es la cruda realidad (*a veces la realidad supera ampliamente a la ficción, y en este tipo de determinaciones podemos asegurarlo con mucha certeza*).

Se puede encarar esta ardua tarea, con la intención de aprovechar el esfuerzo o simplemente, para cumplir con un requisito que permita a la empresa seguir fielmente las exigencias del mercado y hacer el mínimo esfuerzo posible, tratando de (fría y crudamente) engañar al auditor…..(Lo que recuerda que, también afirmamos y con mucha más contundencia, es **que será imposible de mantener esta mentira**). Podemos garantizar que será humanamente imposible volver a demostrar, año tras año, que el SGSI sigue rodando (la mentira tiene patas cortas). Es decir, no merece la pena tratar de encarar una futura certificación ISO 27001 si no se tiene como objetivo fundamental y sincero:

"Implementar un VERDADERO SGSI"

Esto se desmorona muy rápidamente si se partió de pilares débiles, engañosos o falsos, tratando meramente de obtener el sello de "ISO 27001" como única meta.

Por lo tanto, primer "consejo" (si se puede llamar así):

No os auto engañéis, encarad esta tarea con la sana intención de aprovechar al máximo cada esfuerzo que esta requiera.

Una vez comprendido esto, creemos necesario avanzar un poco más aún, pues esto afecta de lleno a las PyMEs y tal vez no tanto a una gran empresa.

Todo responsable de implementar ISO 27001 en una PyME, en mayor o menor medida, "**SÍ o SÍ**" ¡¡ **debe MOJARSE** !!

Una gran empresa, tal vez pueda darse el lujo de externalizar todo el proceso, el mantenimiento y las acciones a futuro…….una PyME seguro que no. A lo sumo, deberá contratar una consultora que le analice, diseñe, planifique e implemente inicialmente desde el vamos, todo el SGSI, pero es casi seguro que no podrá subcontratar el mantenimiento que un SGSI requiere, esta tarea implica poseer un claro entendimiento de lo que se hizo y el funcionamiento de todo el SGSI, por lo tanto, en cualquier caso alguien, responsable de la PyME deberá intervenir en profundidad. Esto no quiere decir que le implicará abandonar el resto de sus tareas, pero se debe ser consciente, que algo de su tiempo le deberá dedicar.

El responsable de seguridad de la PyME deberá "Mojarse" desde el inicio. Puede hacerlo, embebiéndose del Estándar, e ir preparando poco a poco su empresa con un mínimo apoyo de algún especialista. Este curso de acción, requiere un mayor esfuerzo de los administradores de informática de la empresa (y de su responsable), pero es el que mayor experiencia les aportará y, los resultados, si se ponen ganas, serán muy buenos y dejarán claros los pasos a futuro para mantener el SGSI funcionando perfectamente.

La segunda opción que puede tener el responsable de seguridad de una PyME, es contratar una consultoría para que lo guíe paso a paso en todo el proceso, ¡¡ ojo!! no estoy diciendo que haga todo el trabajo, sino que vaya guiando a la empresa en cómo hacerlo, pues si lo hace la consultora, se cae en la mentira anteriormente mencionada, pues una vez que se retire el consultor, el SGSI será muy duro de mantener. Por lo tanto lo más importante a reflexionar sobre esta segunda opción es que no es "lavarse las manos", sino trabajar codo a codo con el consultor, para aprovechar al máximo la experiencia de éste en cada paso, y ser capaz de tener un claro conocimiento de todo lo realizado, para mantenerlo en funcionamiento, se reitera que de esto se trata: "un ciclo de vida continuo".

En cualquiera de los dos casos, es perfectamente posible preparar una PyME para luego solicitar a los auditores acreditados la certificación ISO 27001.

- ¿Cómo Proponemos realizar esta tarea en una PyME?

Esta actividad de apoyo a las PyMEs que desean encarar un SGSI, independientemente que su objetivo sea certificarse o no (pues muchas lo lanzan únicamente para mejorar la gestión de su seguridad), la hacemos de acuerdo a las siguientes fases:

FASE1: Análisis de la Situación Actual y Evaluación de la Seguridad

Objetivo: Identificar los objetivos de negocio, ya que el propósito de la certificación es garantizar la gestión de la seguridad sin perder de vista que esta ayuda al desarrollo de las actividades comerciales de la PyME

- **Análisis y Estudio del Ámbito de Aplicación (Alcance de la Certificación ISO/IEC 27001:2005).**
- **Identificación de Activos.**
- **Análisis de Riesgos (orientado a procesos de negocio).**
- **Declaración de Intenciones de la Dirección.**
- **Plan de Acción para implementar ISO/IEC 27001:2005.**
- **Inicio del Rodaje:**
- **Selección de Hitos (Medibles, demostrables: RODAJE).**
- **Estándar de Seguridad:**
- **Relación Documental.**
- **LOPD y LSSI (conformidades legales).**
- **Planeamiento y Ejecución de Formación y Concienciación.**
- **Auditoría Interna (plan, realización, resultados, mejoras).**
- **Preparación de Presentación del SGSI a auditores.**
- **Solución de Observaciones y No Conformidades.**

Las **tareas** a desarrollar son:

- Se identifican cuáles son las principales actividades empresariales, reflejándolas en un diagrama de flujo.
- Se selecciona un alcance adecuado para el sistema, ya que el esfuerzo a la hora de implementar el SGSI debe ser proporcional al tamaño del sistema a construir
- En base a la Norma ISO 27002, se comprobará qué controles de dicha norma están implantados, y a qué nivel en base a un checklist. Con esto, se consigue determinar el estado de madurez en el que se encuentra la compañía, para poder identificar el esfuerzo que hay que hacer en la implementación.

FASE2: Análisis y Gestión de Riesgos

Objetivo: Establecer la relación entre la compañía y su entorno, identificando sus puntos fuertes y sus puntos débiles, oportunidades y amenazas.

Las **tareas** a desarrollar son:

- Análisis de Riesgos.
- Tratamiento de Riesgos.

FASE3: Lanzamiento del SGSI

Objetivo: Desarrollar los procedimientos necesarios que permitan implantar los controles seleccionados. En cada procedimiento se detallan los objetivos que se pretenden cubrir, cómo se implantan, y las responsabilidades asociadas.

Las **tareas** a desarrollar son:

Definición del SGSI:

- Se define la Política de Seguridad que establece de forma clara el enfoque de la política de actuación de la compañía, el alcance y los objetivos globales.
- Se recopilan los documentos relativos a la seguridad, existentes en la compañía.
- Se elaboran y estructuran los procedimientos de gestión y funcionamiento del SGSI, que darán soporte a la Política de seguridad definida para la compañía.
- Se desarrolla una bitácora de actividades en donde se van registrando los hitos alcanzados y las actividades que se están desarrollando a lo largo del tiempo.

FASE 4: Implantación y Puesta en Marcha del SGSI

Objetivo: Poner en marcha las políticas y procedimientos definidos en las fases previas, tomando previamente en consideración el necesario aprovisionamiento de fondos y la asignación de responsables.

Las **tareas** a desarrollar son:

a. Formular e implementar un Plan de Tratamiento del Riesgo que identifique las acciones, responsabilidades y prioridades de la Dirección para la gestión de los riesgos de la seguridad.

b. Implantar Planes de Formación y Concienciación:

   1) Impartir **Planes de Formación** sobre los nuevos procedimientos.

   2) Impartir **Planes de Concienciación** sobre los beneficios que tiene implantar un SGSI en la compañía.

c. Implantar el SGSI:

   1) Implantar políticas y procedimientos del SGSI.

   2) Implantar los controles seleccionados en el documento Declaración de Aplicabilidad.

FASE 5: Control y Revisión del SGSI

Objetivo: Realizar revisiones sobre la efectividad del SGSI atendiendo al cumplimiento de la política y objetivos del SGSI, los resultados de las auditorías de seguridad, incidentes, resultados de las métricas, y observaciones de las partes interesadas.

Las **tareas** a desarrollar son:

a. Controlar el SGSI:

- Se detectan los errores en los resultados del tratamiento de riesgos.
- Se identifican y detectan las incidencias de Seguridad.
- Se controla que las actividades de seguridad son realizadas correctamente, tanto por la tecnología implantada, como por las personas en las que se ha delegado la responsabilidad.
- Determinar las acciones para resolver las brechas de seguridad de acuerdo a la prioridad de los negocios.
- Se establecen métricas de seguridad para medir la eficacia y eficiencia del SGSI (ISO 27004).
- Se determina las acciones llevadas a cabo para resolver una incidencia de seguridad son las adecuadas.

b. Revisar el SGSI:

Realizar auditorías y revisiones por la Dirección del SGSI:

- Se revisa la Política de Seguridad y el Alcance del SGSI.
- Se revisa el Análisis de Riesgos.
- Se revisan los controles implantados.
- Se realizan auditorías internas y externas.

FASE 6: Mantenimiento y Mejora del SGSI

Objetivo: Implementar las mejoras identificadas a partir de los resultados obtenidos en las fases anteriores, asegurando que éstas permitan alcanzar los objetivos del SGSI.

Las **tareas** a desarrollar son:

a. Mantenimiento del SGSI:

- Se comunica a las partes interesadas las acciones y mejoras.

- Se ejecutan las acciones correctivas y preventivas.

b. Mejora del SGSI:

- Se implantan las mejoras del SGSI que se han identificado.

8.6. IPSec.

8.6.1. Análisis de IPSec

IPSec está definido por un conjunto de RFCs que especifican una arquitectura básica para implementar varios servicios de seguridad en la familia de protocolos TCP/IP. Contempla su implementación tanto con la Versión 4 como con la 6 del protocolo IP.

IPSec puede ser empleado para proteger uno o más caminos entre pares de host, entre host y Gateway de seguridad o entre pares de Gateway de seguridad. El término *Gateway de seguridad se refiere a un sistema intermedio que implementa IPSec* (Ej: Router, Firewall,etc). El conjunto de servicios que IPSec puede proveer incluye:

- Control de accesos.
- Integridad no orientada a la conexión.
- Autenticación de origen de datos.
- Rechazo o reenvío de paquetes.
- Confidencialidad.
- Negociación de Compresión IP.

Los componentes fundamentales de esta arquitectura son:

- Protocolos de seguridad: Compuestos por **AH** (Autentication Header) [RFC-2402] y **ESP** (Encapsulation Security Payload) [RFC-2406].
- Asociaciones de seguridad (**SA**: Security Asociation).
- **IKE** (Internet Key Exchange) [RFC-2409], para intercambio de claves manual y automático.
- Algoritmos de autenticación y cifrado.

Estos cuatro puntos son los que se tratarán en detalle a continuación.

8.6.2. AH (Authentication Header) [RFC-2402]

AH puede ser implementado solo, en combinación con ESP o anidado en el modo túnel de IPSec.

Los servicios de seguridad que ofrece pueden ser entre:

- Dos Host.
- Un Host y un Gateway de seguridad.
- Dos Gateway de seguridad.

ESP puede se empleado para realizar los mismos servicios y además confidencialidad a través del cifrado de los datos. La principal diferencia entre ambos está dada en que ESP no protege ningún encabezado IP a menos que estos campos estén encapsulados en ESP en modo túnel (como se verá más adelante).

En el caso de IPv4, el campo protocolo del mismo identifica la presencia de AH a través del valor 51d, en el caso de IPv6 en el campo próximo encabezado. El formato del encabezado de AH es el siguiente:

Next Header	Payload Length	Reserved
Security Parameters Index (SPI)		
Sequence Number		
Authentication Data (Variable)		

- Next Header: Este campo de 8 bit define lo que sigue luego del AH Header, es valor de este campo es el mismo que establece IANA para el protocolo IP.
- Payload Length: Expresa la longitud de todo el paquete en palabras de 32 bit.
- Reservado: Se reserva para usos futuros y debe ser puesto a cero.
- Security Parameters Index (SPI): Es un valor arbitrario de 32 bit; en combinación con la dirección IP, el protocolo de seguridad (AH) identifica unívocamente la SA para este datagrama. Queda reservado el rango de valores entre 1 y 255 para IANA para usos futuros.
- Sequence Number: Es un valor de 32 bit que contiene un contador monótono creciente. Debe estar siempre presente aún si el receptor no posee servicio anti-réplica para una SA específica. Los contadores del emisor y receptor son inicializados en 0 cuando se establece la SA.
- Authentication Data: Este es un campo de longitud variable que contiene el Intregrity Check Value (ICV).

Como ESP, AH puede ser implementado en modo transporte o túnel.

En modo transporte, AH es insertado después del encabezado IP y antes de los protocolos de nivel superior (Ej: TCP, UDP, ICMP, etc.), o antes de cualquier otro encabezado IPSec que ya haya sido insertado. A continuación se grafican las distintas opciones:

IPv4	IP Original	**AH**	TCP	Datos

IPv6	IP Original	Hop-by-hop,routing, etc	**AH**	Otros Enc. Ext.	TCP	Datos

Todos los encabezados de extensión de IPv6, si están presentes pueden estar antes, después de AH o ambos casos.

En modo túnel puede ser implementado en Gateway de seguridad y host, pero en el caso de Gateway de seguridad solamente se puede implementar en modo túnel (es decir que no soporta el modo transporte). En el modo túnel aparecen dos encabezados IP, uno externo y otro interno. El interno transporta el origen y destino final del datagrama, mientras que el externo contiene otras direcciones IP (Ej: la de los Gateway de seguridad). En este modo, AH protege la totalidad del paquete incluyendo el encabezado IP interno. La ubicación del mismo se grafica a continuación:

Ipv4	Nuevo Enc. IP	**AH**	IP Original	TCP	Datos

IPv6	Nuevo Enc. IP	Hop-by-hop,routing,etc	**AH**	IP Original	Otros Enc.Ext.	TCP	Datos

El algoritmo empleado para calcular ICV es especificado por la SA. En el caso de comunicaciones punto a punto debe soportar códigos de autenticación de mensajes (MAC: Message authentication Code), tanto algoritmos simétricos (Ej: DES) como funciones "One-Way" (Ej: MD5 o SHA-1). En comunicaciones multicast son adecuadas las combinaciones de funciones "One-Way" con algoritmos de firma electrónica.

Por lo tanto cualquier aplicación de AH debe soportar DES, HMAc con MD5 [RFC-2403] y HMAC con SHA-1 [RFC-2404].

NOTA: Todos estos algoritmos son descriptos en el punto 5. (Algoritmos de autenticación y cifrado).

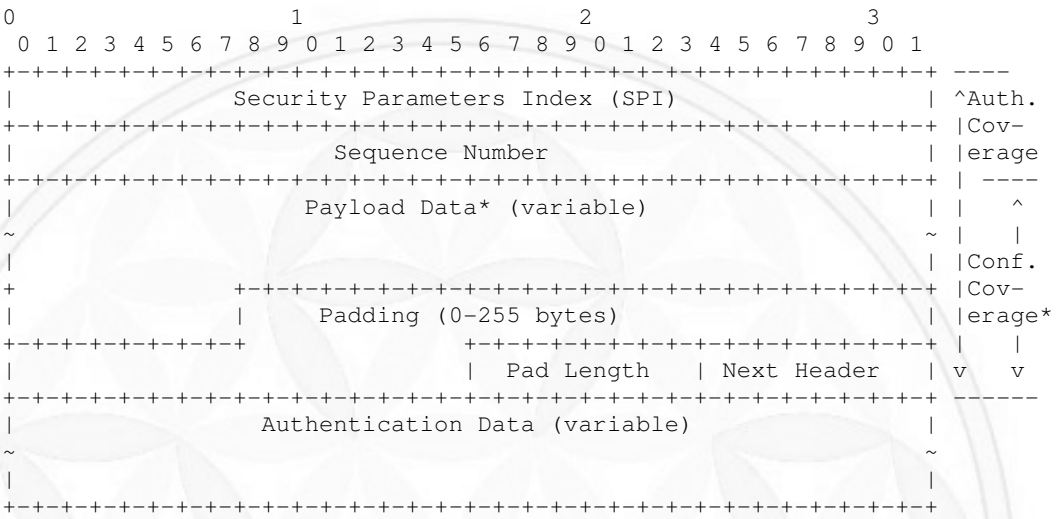
8.6.3. ESP: Encapsulation Security Payload

ESP está diseñado para proveer servicios de seguridad a IPv4 e IPv6. ESP puede ser aplicado solo en combinación con AH o en modo anidado. Los servicios de seguridad pueden ser provistos entre un par de Host, entre un par de Gateway de seguridad o entre un Gateway de seguridad y un Host.

El encabezado del ESP es insertado después del encabezado de IP y antes de los protocolos de nivel superior en modo transporte, y en modo túnel antes del encabezado IP.

ESP provee confidencialidad, autenticación de origen de datos, integridad no orientada a la conexión y servicio anti-réplica. Estos servicios son seleccionados al establecerse la asociación de seguridad (SA) .

El encabezado IP identifica la ocurrencia de este protocolo a través del valor 50d en el campo protocolo de su formato. El encabezado ESP tiene la siguiente estructura:



- Security Parameters Index: Valor arbitrario de 32 bit, en combinación con la dirección IP destino y el ESP identifican unívocamente una SA para este datagrama. El conjunto de valores entre 1 y 255 es reservado por IANA para usos futuros.
- Sequence Numbers: Contador monótono creciente que permite el control de los paquetes enviados y recibidos. Este valor es inicializado a 0 cuando la SA es establecida.
- Payload Data: Campo de longitud variable que define la longitud de los datos que continúan este encabezado. Si el algoritmo empleado para cifrar los datos requiere algún tipo de sincronización (Ej: vector de inicialización (VI)) entonces estos datos pueden ser transportados explícitamente en este campo.
- Padding (Para cifrado): Varios factores requieren o motivan el uso de este campo, como por ejemplo: algoritmos que requieren que el texto sea un múltiplo exacto de cierto número de Bytes, protocolos que necesitan rellenar a múltiplos de 4 Bytes.
- Pad Length: Este campo indica el número de Bytes de relleno que lo preceden.
- Next Header: Este campo de 8 Bytes identifica el tipo de datos que se transporta, por ejemplo encabezado de extensión en IPv6 o protocolo de nivel superior.
- Authentication Data: Este campo de longitud variable contiene un Integrity Check Value (ICV) que procesa el paquete ESP menos la autenticación de datos. Este campo es opcional y es incluido sólo si el servicio de autenticación ha sido seleccionado en esta SA.

Como AH, ESP puede ser empleado en dos modos: modo transporte y modo túnel, dando protección a los niveles superiores pero no al encabezado IP. En modo transporte ESP es insertado después del encabezado IP y antes de los encabezados de nivel superior.

IPv4	IP Original	**ESP**	TCP	Datos	Cola ESP	ESP Auth

IPv6	IP Original	Hop-by-hop,routing, etc	**ESP**	Otros Enc. Ext.	TCP	Datos	Cola ESP	ESP Auth

En modo túnel ESP puede ser empleado en Host o Gateway de seguridad. Cuando ESP es implementado en Gateway de seguridad debe ser empleado el modo túnel, en este caso el encabezado IP "interno" transporta la última dirección fuente y destino IP, mientras el encabezado IP "externo" contiene otra dirección IP. En este modo ESP protege la totalidad del paquete IP interno incluyendo su encabezado.

IPv4	IP Nuevo	**ESP**	IP Original	TCP	Datos	Cola ESP	ESP Auth

Ipv6	IP Nuevo	Hop-by-hop,routing, etc	**ESP**	IP Original	Otros Enc. Ext.	TCP	Datos	Cola ESP	ESP Auth

ESP está diseñado para usarse con algoritmos de clave simétrica, y en virtud que los datagramas IP pueden arribar fuera de orden, cada paquete debe transportar todos los datos requeridos para permitir al receptor establecer el sincronismo necesario para descifrar. Estos datos deben ser explícitamente transportados en el campo Payload, descrito anteriormente.

El algoritmo empleado para calcular ICV es especificado por la SA. En el caso de comunicaciones punto a punto debe soportar códigos de autenticación de mensajes (MAC: Message authentication Code), tanto algoritmos simétricos (Ej: DES) como funciones "One-Way" (Ej: MD5 o SHA-1). En comunicaciones multicast son adecuadas las combinaciones de funciones "One-Way" con algoritmos de firma electrónica.

Toda implementación de ESP debe soportar los siguientes algoritmos:

- HMAc con MD5 [RFC-2403] .
- HMAC con SHA-1 [RFC-2404].
- DES (Data Encriptation Standard) [ANSI X3.106] en modo CBC.
- Algoritmo de autenticación nula.
- Algoritmo de cifrado nulo.

NOTA: Todos estos algoritmos son descriptos en el punto 5. (Algoritmos de autenticación y cifrado).

8.6.4. Asociaciones de seguridad (SA: Security Association)

Una SA es una clase de conexión que permite establecer los servicios de seguridad en el tráfico que transporta. En cada aso SA los servicios de seguridad pueden hacer uso de AH o ESP pero no de ambos, para utilizar los dos, se deberá establecer dos SA.

Una SA es unívocamente identificada por tres valores:

- SPI (Index Parameter Security).
- Dirección IP destino.
- Identificador de protocolo de seguridad (AH o ESP).

Se pueden definir dos tipos de SA:

8.6.4.1. Modo transporte: Se trata de una SA entre dos host. En este tipo, el encabezado del protocolo de seguridad aparece inmediatamente a continuación del encabezado IP en el caso de Ipv4 y como cabecera de extensión en Ipv6. En ambos casos ocurre antes del nivel de transporte.

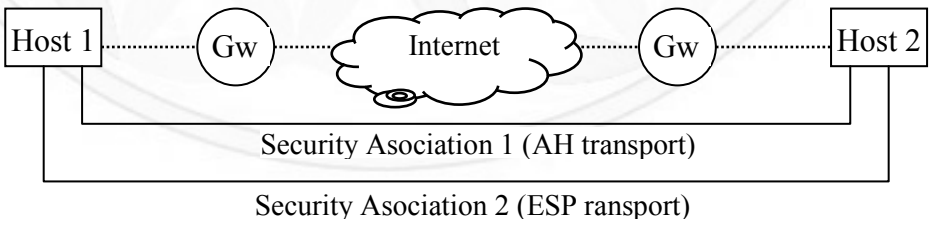
8.6.4.2. Modo túnel: Se trata de una SA aplicada a un túnel IP. Si el extremo de la SA es un Gateway de seguridad, entonces la SA debe ser en modo túnel, es decir que una SA entre dos Gateway de seguridad es siempre en modo túnel.

En este modo existen dos encabezados IP, uno que es el *externo* que especifica los datos para llegar al destino del túnel y otro *interno* a este que detalla el destino final.

Un host debe soportar ambos modos, mientras que un Gateway de seguridad sólo debe soportar modo transporte.

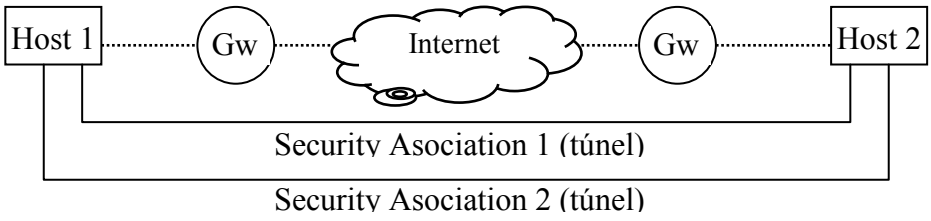
En algunos casos la política de seguridad puede necesitar hacer uso de ambos protocolos de seguridad (AH y ESP), los cuales, como se mencionó anteriormente no pueden estar presentes en la misma SA. En estos casos será necesario emplear múltiples SA. El término empleado en estos casos es Empaquetado (bundle) de SA. Las diferentes SA pueden iniciarse y finalizar en los mismos puntos o no y se pueden combinar de dos formas:

- **Transporte adyacente:** Se trata de aplicar más de un protocolo de seguridad a un mismo datagrama sin invocar un modo túnel, aprovechando la combinación de AH y ESP.

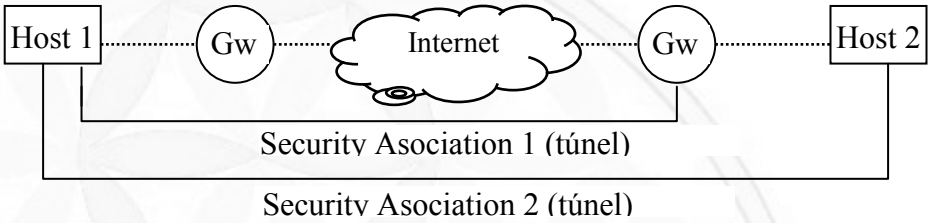


- **Túnel Iterado:** En este caso son también varias SA pero implementadas a través de modo túnel, y se puede llevar a cabo a través de tres formas:

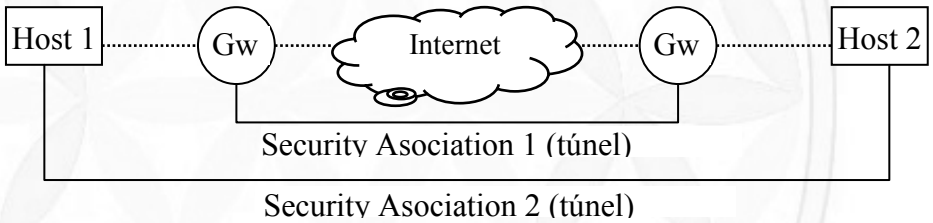
1) **Host, Host <-> Host, Host:** Ambos extremos de las SA son los mismos. Cada túnel podría emplear AH o ESP.



2) **Host, Host <-> Gateway, Host:** Un extremo de las SA es el mismo y el otro no.



3) **Host, Gateway <-> Gateway, Host:** Ninguno de los extremos de las SA son los mismos.

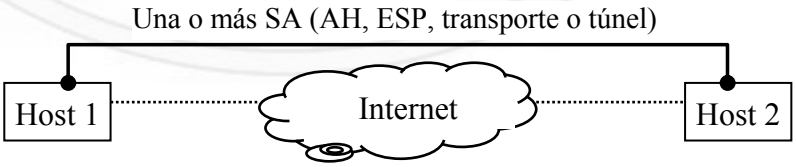


Combinación de asociaciones de seguridad:

Cualquiera de las propuestas anteriores puede ser combinada con otras, generando Empaquetados de SA mixtos.

Hay cuatro casos básicos de estas combinaciones que deben ser soportados por todo host o Gateway de seguridad que implemente IPSec, estos son:

1) Seguridad de extremo a extremo entre 2 Host a través de Internet o Intranet (Host1 <-> Host2).



Como se aprecia en el gráfico, se puede implementar en modo transporte o túnel, acorde a esto, los encabezados de los paquetes pueden adoptar las opciones que se detallan a continuación:

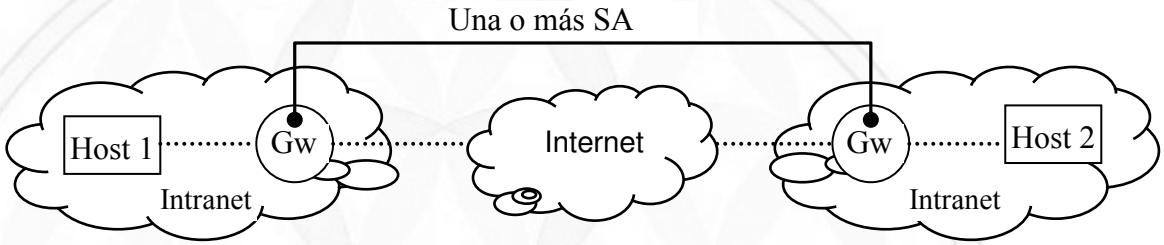
Modo Transporte

1.	IP	AH	Niv. Superior	
2.	IP	ESP	Niv. Superior	
3.	IP	AH	ESP	Niv. Superior

Modo túnel

4.	IPExt	AH	IPInt	Niv. Superior
5.	IPExt	ESP	IPInt	Niv. Superior

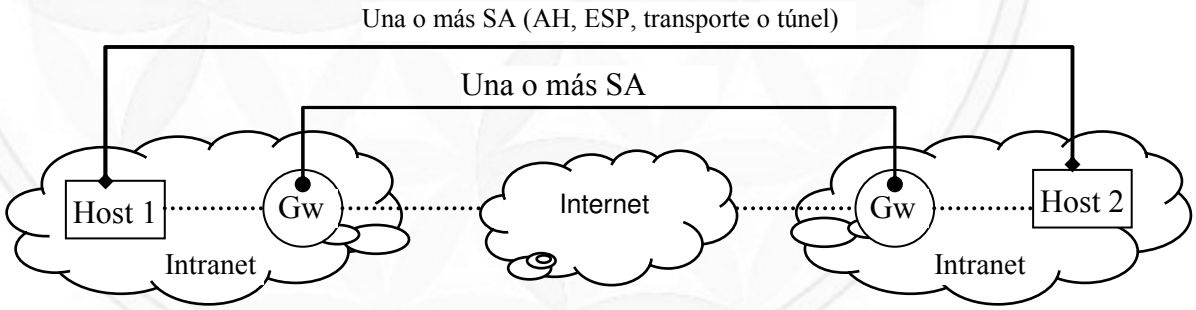
2) Soporte con simple VPN.



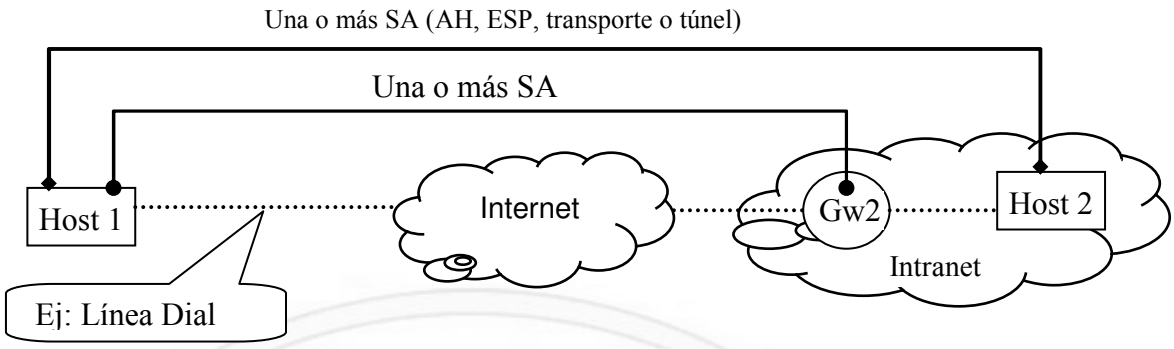
Modo túnel

1.	IPExt	AH	IPInt	Niv. Superior
2.	IPExt	ESP	IPInt	Niv. Superior

3) Combinación de 1) y 2).



4) **Host remoto a través de Internet:** Este caso está contemplado para un host remoto que accede a la organización a través de Internet desde cualquier punto. Accede a la misma a través de Gateway de seguridad 2 (típicamente un Firewall) y a través de este gana el acceso a cualquier host de la misma sobre el cual pueda tener una SA o al mismo Firewall. El detalle particular de este caso es la autenticación, verificación y autorización.
Sólo el modo túnel puede ser empleado entre el host 1 y el Gateway de seguridad, y entre los host cualquiera de ellos.



8.6.5. Administración de claves (IKE: Internet Key Exchange) [RFC-2409]

IPSec impone el soporte para dos tipos de administración de claves aves: Manual y automático. Los protocolos AH y ESP son totalmente independientes de las técnicas empleadas para la administración de claves. La granularidad que se emplee para la distribución de claves determina la granularidad que proveerá la autenticación. Se debe tener en cuenta que la fortaleza de AH y ESP estará dada en gran medida por la administración de claves, pues una debilidad en esta, genera una vulnerabilidad en los secretos empleados en el sistema

8.6.5.1. Manual:

Esta es la forma más simple de administración de claves, en la cual personalmente se configuran las claves de cada componente del sistema y las SA para asegurar la comunicación con otros sistemas.

Estas técnicas son prácticas en pequeños y estáticos entornos, pero no son escalables. En general se emplean técnicas de configuración estáticas con el empleo de claves simétricas, si bien existen varias posibilidades.

8.6.5.2. Automático:

El empleo de IPSec en grandes entornos requiere esta técnica, la cual es fácilmente escalable y automatizada.

El protocolo por defecto que propone IPSec es IKE (Internet Key Exchange), sin embargo otros protocolos pueden ser seleccionados.

Cuando estos protocolos son empleados, la salida de los mismos pueden generar múltiples claves, las cuales sirven para:

- Algoritmos criptográficos que usan múltiples claves.
- Algoritmos de autenticación que usan múltiples claves.
- Combinaciones de ambos.

La RFC-2409 describe un protocolo híbrido cuyo propósito es negociar y proveer material de claves autenticado para SA de una manera protegida.

IKE define tres elementos fundamentales:

- OAKLEY [RFC-2408]: Define una serie de "modos" de intercambio de claves detallando los servicios que provee cada uno.
- SKEME (Secure Key Exchange Mechanism for Internet): Describe una técnica de intercambio de claves muy versátil que provee anonimato, repudio y rápido refresco de claves.
- ISAKMP [RFC-2408] (Internet Security Association and Key Management Protocol): Provee un entorno para autenticación e intercambio de claves, pero no los define, sólo se limita a establecer las fases a seguir. Esta fases son dos, la primera de ellas (Modo principal y agresivo) establece un canal seguro y autenticado entre los extremos; la segunda fase (Modo rápido) establece la negociación de la SA de IPSec

Para negociar y establecer claves, IKE necesita hacer uso de:

- Algoritmos de cifrado.
- Algoritmos Hash (Deben soportar HMAC [RFC-2104].
- Métodos de autenticación.
- Información acerca de un grupo sobre la cual hacer Diffie-Hellman.

Las implementaciones de IKE deben soportar:

- DES (Data Encryption Standard) [ANSI X3.106] en modo CBC.
- MD5 (Message Digest Algorithm Versión 5) [RFC-1321] y SHA (Secure Hash Standard) [FIPS- 180-1, de NIST].
- Autenticación por medio de clave secreta pre-compartida.
- MODP sobre un número de grupo por defecto.

Opcionalmente pueden soportar:

- 3DES (triple DES) para cifrado.
- Tiger para Hash.
- DSS (Digital Standard Signature).
- RSA (Rivest, Shamir and Aldeman).
- MODP grupo 2

NOTA: Todos estos algoritmos son descriptos en el capítulo correspondiente.

IKE propone dos métodos básicos para establecer un intercambio de claves autenticado:

- **Modo Principal** (Obligatorio): Sólo se emplea en la fase uno de ISAKMP. Es una instancia de ISAKMP para proteger el intercambio, y funciona de la siguiente manera:
  1) Los primeros dos mensajes negocian políticas.
  2) Los próximos dos mensajes intercambian los valores públicos de Diffie-Hellman y datos auxiliares.
  3) Los últimos dos mensajes autentican el Intercambio Diffie-Hellman.
- **Modo agresivo** (Optativo): Sólo se emplea en la fase uno de ISAKMP. Es también una instancia de ISAKMP, y funciona de la siguiente manera:
  1) Los primeros dos mensajes negocian políticas, intercambian los valores públicos de Diffie-Hellman y datos auxiliares para intercambio e identidad.
  2) El segundo mensaje también autentica a quien responde.
  3) El tercer mensaje autentica a quien inició el intercambio y provee un perfil de participación en el intercambio.

Existe también un modo rápido (Sólo se emplea en la fase dos de ISAKMP) para la refrescar las claves y SA, el cual no es un intercambio completo, pero es usado como parte de los procesos anteriores.

En modo principal o agresivo están permitidos cuatro métodos de autenticación:

- Firma digital.
- Dos métodos de clave pública.
- Secreto precompartido.

8.6.6. ISAKMP [RFC-2408] (Internet Security Association and Key Management Protocol).

En este texto ya hemos definido de forma práctica el funcionamiento de este protocolo que es el pilar fundamental de toda arquitectura de seguridad, pues cualquier fallo en lo relativo a las claves deja sin sentido toda otra medida.

Este protocolo define los pasos necesarios para establecer una SA (Security Association), el establecimiento y mantenimiento de todas las claves necesarias para la familia de protocolos TCP/IP en modo seguro.

Ya hemos desarrollado antes un ejemplo práctico tomado de la realidad en el establecimiento de una VPN por medio del software PGP, que implementa todos los estándares presentados por ISAKMP.

8.6.7. Procesamiento de tráfico IP

Para el tratamiento del tráfico entrante y saliente en un elemento que implemente IPSec existen dos bases de datos que definen las normas a seguir, la primera de ellas es la *base de datos de políticas del seguridad* (**SPD: Security Policy Database**) la cual define todos los requerimientos del sistema y establece si un paquete se descarta, emplea servicios IPSec o permite "Bypass" IPSec. La segunda es la *base de datos de asociaciones de seguridad* (**SAD: Security Associaton Database**), la cual define los parámetros de todas las SA activas del sistema. Las dos bases de datos toman parámetros llamados *selectores* que son los que hacen posible la decisión sobre las medidas a tomar en el tráfico saliente o entrante. Los selectores definidos por IPSec son, Para SPD: Dirección IP fuente o destino, nombre (Usuario o sistema), nivel de sensibilidad de los datos, protocolo de nivel transporte, puertos fuente o destino. Y para SAD: Contador de número de secuencia en AH o ESP, Contador de secuencia de "overflow", ventana anti-réplica, algoritmo de autenticación AH, algoritmo de cifrado ESP, algoritmo de autenticación ESP, tiempo de vida de la SA, modo del protocolo IPSec (túnel o trasporte) y MTU.

Todo paquete entrante o saliente en un elemento IPSec es confrontado con la SPD para determinar qué procesamiento es requerido para el mismo.

8.6.8. Algoritmos de autenticación y cifrado.

- HMAc con MD5 [RFC-2403]
- HMAC con SHA-1 [RFC-2404].
- HMAC [RFC-2104].
- Diffie-Hellman.
- DES (Data Encryption Standard) [ANSI X3.106] en modo CBC.
- MD5 (Message Digest Algorithm Versión 5) [RFC-1321] y SHA (Secure Hash Standard) [FIPS- 180-1, de NIST].
- 3DES (triple DES) para cifrado.
- Tiger para Hash.
- DSS (Digital Standard Signature).
- RSA (Rivest, Shamir and Aldeman).

8.7. Plan de Continuidad de Negocio (PCN).

Hemos decidido incorporar este tema al final de todo por varias razones, la principal, es que si has llevado a la práctica gran parte de lo tratado hasta ahora verás que tienes andado gran parte del camino. La segunda es que para que esta actividad sea eficiente, es necesario que domines el tema, y por último porque en los últimos años se ha avanzado mucho sobre este tema, presentándolo ya no como una mera estrategia de "Backup", sino como un conjunto de medidas que pasan a integrarse dentro del ciclo de vida de la seguridad, por lo tanto hoy en día ya podemos afirmar que es un tema maduro e imprescindible sobre el que tienes la obligación de realizarlo con la mayor seriedad.

Verás que el PCN, en muchos sitios y textos es también llamado Disaster Recovery Plan (DRP) o en Castellano Plan de Recuperación de Desastres (PRD), otros nombres son Plan de Contingencia, militarmente se suele denominar "Imponderables" y forma parte de un apartado completo en cualquier Orden de Operaciones Militares. Muchos autores hacen diferencias entre ellos, hemos participado de verdaderos debates sobre sus orígenes, alcances, aplicaciones, evoluciones, etc… Por nuestra parte creemos que lo más conveniente es que tengas en cuenta un conjunto de tareas que JAMÁS puedes dejar pasar por alto, para **minimizar el impacto de cualquier anomalía en tus sistemas**, sea cual fuere, prevista o imprevista, natural o artificial, leve o catastrófica, todo ello en definitiva implica un esfuerzo adicional imprescindible cuyo éxito se verá plasmado únicamente el día que ocurra algo, pero si lo has hecho bien será una de las satisfacciones más grandes de tu vida, sino….

8.7.1. Conceptos.

Al final presentaremos brevemente cómo se trata este tema a nivel militar, pero hasta ahora ya hemos visto bastante al respecto, te recordamos que al desarrollar ISO-27001 presentamos dos grupos de controles:

- A.13 Administración de los incidentes de seguridad
- A.14 Administración de la continuidad de negocio

Los mismos tratan con máximo detalle esta actividad y lo más importante es que lo incorporan definitivamente al SGSI, por lo tanto pasan a formar parte del "Ciclo de vida de l seguridad". Tal vez este sea el hito más importante que haya sufrido un PCN, pues con él nos obliga a actualizarlo, medirlo y mantenerlo, cuestión que siempre fue su punto más débil. Hemos visto y sufrido cientos de veces el "error de recuperación" de una copia de seguridad, de restauración de un sistema, el fallo de un CD, la rotura de un disco duro, la pérdida de el "documento ese….", la falta del teléfono justo en el momento clave, la llave que no está, el antivirus que no se actualizó…., etc… Hasta esos cientos de "etcéteras" el PCN era una actividad estática, se iban haciendo copias de seguridad (rutinarias), se compraba una torre de CDs, un sistema de Backup por cintas, se guardaba todo en la caja fuerte tal, en tal despacho, se los llevaba a su casa el administrador. Todas esas actividades

se lanzaban un determinado día e inexorablemente entraban en el "Ciclo de rutina" de la organización (en vez del "Ciclo e vida" de la Organización), todo el problema detonaba cuando al producirse algún incidente, debía hacerse uso de la rutinaria tarea y ahí comenzaban los dolores de cabeza. Al estar formando parte de un SGSI, es necesario hacer pruebas periódicas, documentarlas, solucionar los fallos que se produzcan, proponer mejoras, actualizar planes y procedimientos, informar los resultados, y todo ello nos lleva a minimizar el impacto, que como empezamos en esta sección es el verdadero objetivo de un PCN.

En la actualidad existen varias normas que tratan el concepto de PCN, de las cuales las más importantes son:

- ISO/IEC 27031: Directrices para la preparación de las TIC en la Continuidad de Negocio
- BS25999-1 (Código de prácticas) de diciembre 2006
- BS25999-2 (Especificación) de noviembre 2007
- UNE 71599-1:2010 Gestión de la continuidad del negocio. Parte 1: Código de práctica (Prácticamente traducción del BS-25999-1).
- UNE 71599-2:2010 Gestión de la continuidad del negocio. Parte 2: Especificaciones (Prácticamente traducción del BS-25999-1).
- ISO/IEC 24762:2008 Se trata de una guía para la recuperación de desastres en el ámbito de la informática y las tecnologías de comunicaciones.

8.7.2. El Plan de escalada.

Más adelante veremos una descripción del estándar BS-25999 que podríamos decir que hoy en día debe ser el mayor referente metodológico de esta actividad, pero en esta introducción no queríamos dejar pasar por alto algo de nuestra experiencia en este ámbito, mezclada con el mundo militar y que tal vez no se menciona en esta ni en otras normas:

A un desastre se llega por dos caminos:

- **Inmediato.**
- **Escalada de incidentes leves.**

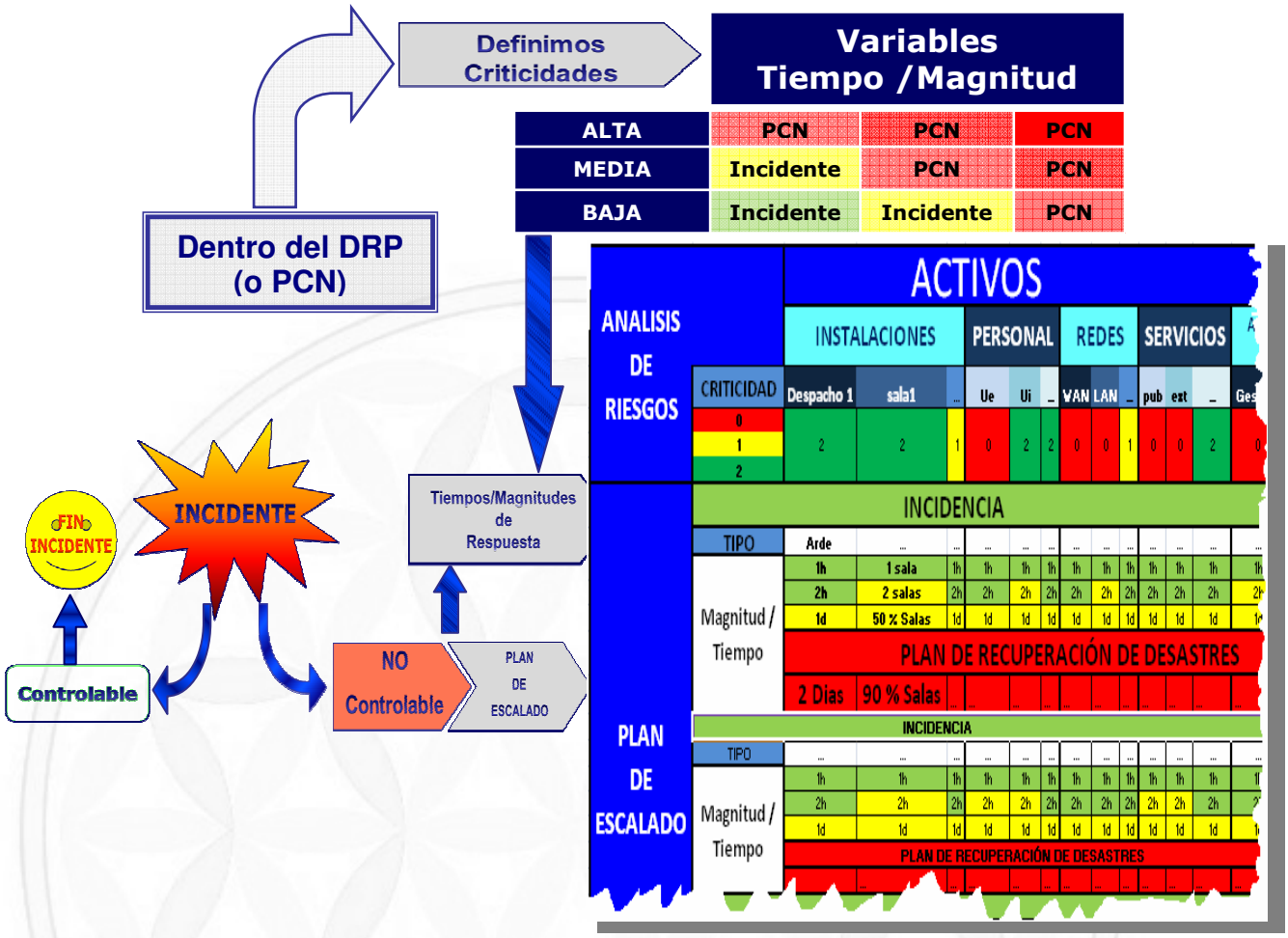
Es decir, un desastre para un sistema informático no es sólo el de las Torres Gemelas, existe un caso conocido de una empresa donde sus empleados almorzaban en el mismo restaurante, hasta que un día enfermaron todos de Salmonera estando allí absolutamente todos los responsables de sistemas. Si hubiera caído enfermo uno de ellos, podríamos haberlo considerado un incidente leve, ya dos tal vez nos causen más problemas, tres, cuatro….todos y a su vez el resto de los empleados no hay duda que es una verdadera catástrofe para la empresa. .El mismo ejemplo podría ser un virus pero esta vez informático, si este afecta a

una PC de un empleado, es un incidente leve, pero si se propaga a todo el hardware y software de la organización esto ya es gravísimo.

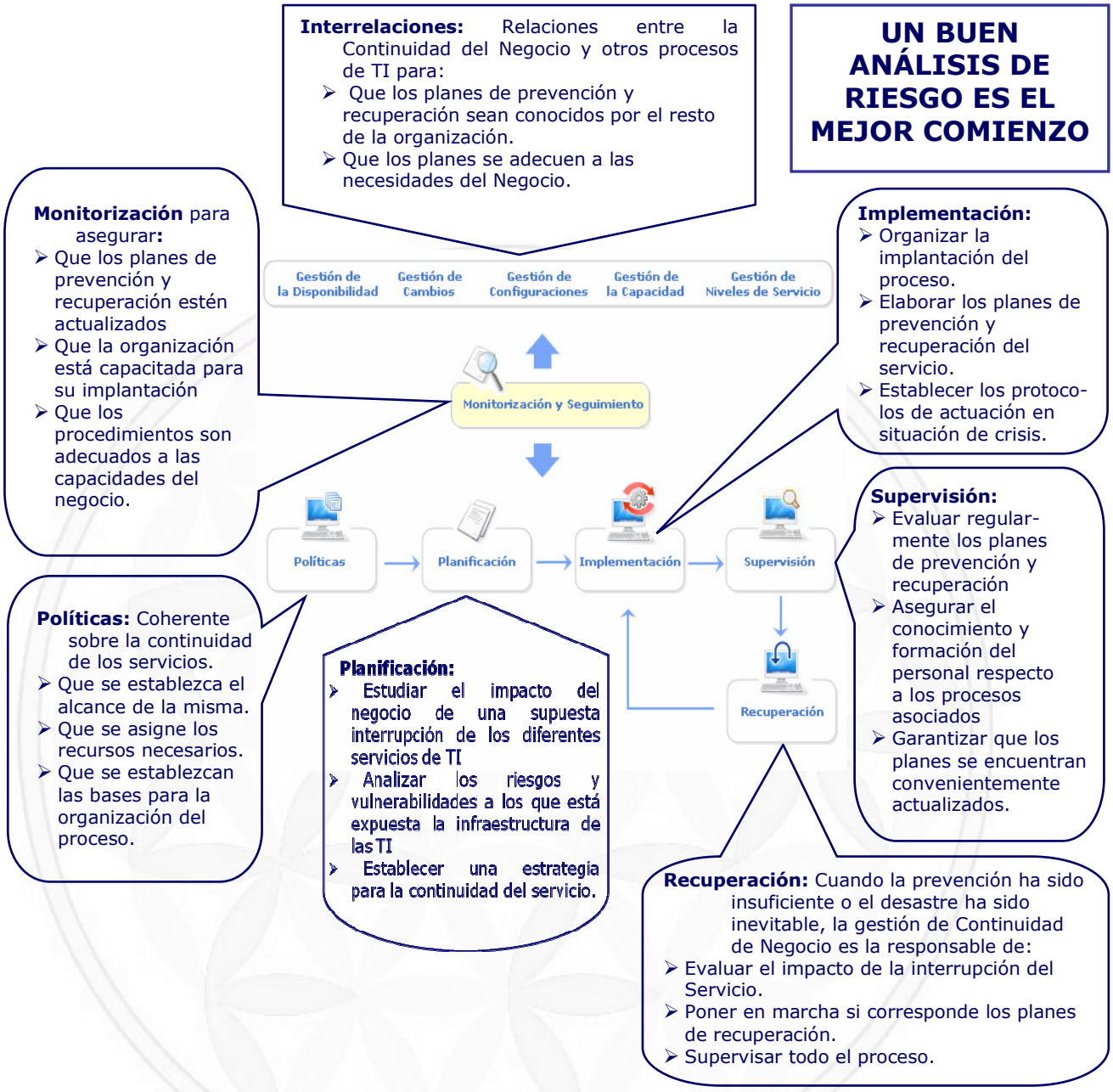
En los ejemplos anteriores hemos tratado el concepto de "Recursos", pero lo mismo sucede con el concepto de "Tiempo". Si un servidor de nuestra organización, sufrió una interrupción y se reinició, lo más probable es que no sea más que una mera "Incidencia", ahora, si nuestro servidor de "Comercio electrónico" se cae y al minuto no lo logramos levantar, ni a la hora, ni al día siguiente, esto empieza a preocupar a la empresa pues dejó de vender "On line" y eso toma el rostro de un "desastre". Este ejemplo concreto nos tocó vivir con una importante y muy conocida empresa de telefonía española (salió en todos los medios de comunicación) cuando se cayó durante casi dos días uno de los nodos principales de conmutación, y por supuesto todos los usuarios conectados al mismo "No hablaban, ni podía conectarse a Internet", este tipo de fallos son mucho más frecuentes de lo que la gente supone, pero duran instantes, lo feo es cuando comienzan a pasar los segundos sin facturación, toca bolsillos muy altos y esto empieza a ser grave e imperdonable (el vil dinero…..).

Lo que intentamos transmitir es que el PCN se puede "Activar" porque se prendió fuego el edificio, o también por una suma de desperfectos que simple vista pueden parecer inofensivos, pero que en conjunto pasan a ser "Letales". Siempre que hemos desarrollado este aspecto, tanto en teoría como en la práctica insistimos en la elaboración de lo que denominamos "Plan de escalada".

El Plan de escalada para nosotros es un proceso fundamental que no está desarrollado por otras normas, y que creemos importante que forme parte del PCN. Una forma de elaboración es definir cuáles pueden ser los escenarios en los que ocurren o pueden ocurrir incidencias, si nuestros sistemas llevan tiempo en producción este es un dato estadístico muy conocido; luego evaluar diferentes valores de tiempo y recursos que de producirse harían escalar estas incidencias. Debemos tener en cuenta que como parte de un SGSI (u hoy en día cualquier infraestructura de seguridad) ya tenemos valores, métricas e indicadores, y si evaluamos correctamente nuestro Plan de escalada, contaremos con valores de "Umbrales" que serán los que activen el PCN, todo esto se puede incorporar a este PCN como un proceso más que responda a la siguiente lógica:



Este "Plan de escalada" forma parte de los procesos de la organización, si consideramos al PCN como un proceso más de gestión integrado dentro del SGSI, entonces pasa a formar parte del "Ciclo de vida de la seguridad", para ello lo debemos presentar como otro proceso más que considere al menos los siguientes pasos:



8.7.3. BS 25999.

Esta norma está compuesta por dos documentos:

- BS25999-1 (Código de prácticas) de diciembre 2006
- BS25999-2 (Especificación) de noviembre 2007

El código de buenas prácticas establece los principios y terminología de la Gestión de Continuidad de Negocio. Explica las bases para la implantación de la continuidad de

negocio. Describe también una serie de controles a tener en cuenta para el plan y su ciclo de vida al estilo PDCA y de forma muy similar a la gráfica que acabamos de presentar en el apartado anterior.

La especificación BS25999-2 nos describe con mayor grado de detalle todos los requisitos para:

- Establecer
- Implantar
- Operar
- Controlar
- Revisar
- Mantener
- Mejorar

Un Sistema completo de Gestión de Continuidad de Negocio

Hay aspectos que esta norma los impone como "Obligatorio", los cuáles son:

Poseer e identificar:

- Política de PCN.
- Alcance.
- Procedimientos.
- Controles.
- Terminología.
- Informe de BIAs (Business Impact Analysis).
- Informe de auditoria de riesgos.
- Plan de formación
- Detalles de las estrategias de BCM.
- Procedimientos para medir la efectividad de la planificación.
- Operaciones y control sobe los procesos de continuidad.
- Plan de continuidad de negocio y plan de gestión de incidencias.
- Información actualizada y detalles de la movilización de agentes/ organizaciones/recursos necesarios en caso de respuesta.
- Control de cambios en los procedimientos.
- Registros de riesgos, agenda de tests y resultados / registros de acciones de test.
- Matriz de incidencias.

- Estructura de respuesta.

Para implantar un Sistema de Continuidad del Negocio esta norma a su vez "requiere" la realización de una serie de actividades y el empleo de ciertos documentos. La primera actividad es el "BIA (Business Impact Análisis)", nuevamente nos vemos con algo que por provenir de diferentes normas parece ser nuevo, pero en verdad no es algo novedoso si hemos hecho a consciencia nuestro "Análisis de Riesgo", pues es prácticamente lo mismo llamado con diferente nombre. Lo que un BIA propone en definitiva es el llegar a determinar el Impacto que nos ocasionaría cualquier anomalía sobre nuestros activos, y como dice el refrán "Todos los caminos conducen a Roma" en principio ningún auditor nos dirá absolutamente nada si nuestro BIA hace total referencia al Análisis de Riesgo, pues de ello se trata.

Una vez evaluado este BIA, debemos pasar a la parte de "Actividades y Productos", las cuales son:

- Actividades y productos clave.
- Impactos a estas actividades.
- Investigar cuánto tiempo las actividades pueden verse impactadas.
- Establecer periodo máximo tolerable de disrupción.
- Identificar dependencias relevantes de proveedores y outsourcing.
- Categorizar las actividades en función de su prioridad e identificar actividades críticas.
- Estimar recursos para cada actividad crítica.
- Objetivos de tiempo de recuperación de actividades críticas con máximo tolerable.
- Revisión de BIAs en intervalos planificado.

La norma continúa con el tratamiento de Riesgos, proponiendo un conjunto de tareas: amenazas, vulnerabilidades, riesgos y controles. Se reiteran los conceptos que ya hemos tratado en el análisis de riesgo y en un SGSI:

Determinar:

- Cómo reducir de la probabilidad de disrupción.
- Cómo acortar el periodo de disrupción.
- Cómo limitar el impacto de la disrupción en los productos servicios clave.
- Estrategia de Continuidad de Negocio (respuesta efectiva, recuperación de actividades críticas, estructura de respuesta y relaciones externas).
- Estructura de respuesta en caso de incidente (personal clave, responsabilidades cómo gestionar el incidente, plan de gestión de incidentes, plan de activación, plan de operación, plan de coordinación y comunicación).
- Control de planes.

- Simulacros (plan y análisis de resultado)

Un aspecto sobre el que sí deseamos detenernos es la parte que desarrolla el seguimiento y medición un Sistema de Continuidad del Negocio, pues lo primero que esta norma describe es la "Revisión por la Dirección" la cual se permite en intervalos, pero de forma planificada y/o por ocurrencia de incidencias, pero lo trata como una actividad OBLIGATORIA, lo cual como sucedió con el SGSI (Revisión por la Dirección) vincula a los más altos niveles de la Organización con el PCN, lo cual insistimos es un punto a favor importantísimo para los responsables de seguridad informática. En este punto entra bastante en detalle y llega a establecer los aspectos que se deben tratar en esta revisión:

- Resultados de auditorias internas tanto a la Gestión del PCN, cómo a proveedores y subcontratistas clave.
- Feedback de partes interesadas.
- Nuevas técnicas, productos o procedimientos que mejoren el BCM – recomendaciones para la mejora.
- Nivel de riesgo residual y riesgo aceptable.
- Amenazas/ vulnerabilidades.
- Seguimiento de otras revisiones.
- Control de cambios.
- Resultados de los simulacros.
- Emergentes buenas prácticas y guías.
- Incidentes.
- Formación.

Como toda actividad de un SGSI, cada una de estas reuniones debe quedar registrada y adecuadamente descripta, por lo tanto es aconsejable contar con una serie de plantillas, una de ellas puede ser "Acta de revisión por la dirección del PCN", en la cual se deja un formato estándar de estos puntos que acabamos de mencionar, los cuáles deben ser tratados en cada reunión y luego firmados por los asistentes o responsables de ese comité.

Otro tema que la norma desarrolla y creemos interesante de destacar es el "Mantenimiento y mejora de un Sistema de Continuidad del Negocio", pues como dijimos al principio la rutina o abandono es uno de los más graves errores de esta actividad y evidentemente este estándar lo tiene muy claro pues establece con total claridad que:

- "La efectividad del programa solamente puede ser validada a través de revisiones y chequeos".

Estas revisiones son esenciales para el desarrollo del trabajo de equipo, mejora de la competencia, confianza y conocimiento.

Los últimos conceptos de esta norma están referidos a la realización de auditorias internas, debiendo contar con un plan de auditoria adecuado. La competencia técnica del equipo auditor debe poder ser demostrada y todo lo que se encuentre en cada una de ellas se expresará de la forma habitual a través de "No Conformidades, acciones correctivas y acciones preventivas", las cuales deberán ser tratadas y solucionadas en los plazos

respectivos que se establezcan en la "Revisión por la Dirección", pues al finalizar toda auditoría y recibir el informe de la misma, es uno de los hitos que imponen obligatoriamente una reunión de este tipo para determinar el curso de acción que se generará para su tratamiento.

8.7.4. Documento Ejemplo.

A continuación presentamos un esquema resumido de los puntos que pueden o deberían ser tratados en un documento del "Plan de Continuidad de Negocio".

1) BIA Análisis de impacto

Esta actividad puede presentarse luego del análisis a través de una "Matriz de Impacto" como la que figura a continuación:

TIPO DE IMPACTO	MAGNITUD DEL IMPACTO				
	CATASTRÓFICO	ALTO	MEDIO	BAJO	NULO
PÉRDIDA DE LA IMAGEN CORPORATIVA	El incidente aparece como titular en la prensa nacional.	El incidente aparece como titular en la prensa nacional.	El incidente aparece en los medios locales de forma destacada.	El incidente aparece en los medios locales de forma no destacada.	El incidente no aparece en la prensa.
PÉRDIDA DE OPERATIVIDAD	No se puede responder a la demanda de casi la totalidad de los agentes de la red.	No es posible satisfacer a más de la mitad de la demanda.	Los agentes importantes no pueden ser satisfechos de forma oportuna.	Sólo algunos de los agentes auxiliares no pueden ser satisfechos.	No hay efectos en la atención de los agentes.
DETERIORO CON ENTIDADES RELACIONADAS	Interrupción inmediata de la relación con las entidades relacionadas.	Muchos de los principales agentes han informado de interrumpir la relación.	Algunos de los principales agentes han informado de interrumpir la relación.	Uno o dos de los principales agentes han informado de interrumpir la relación.	No ha habido efecto en las relaciones con los agentes.
DETERIORO AMBIENTE	Todos los empleados entran	Los empleados de	Hay un manifiesto	Algún empleado	El ambiente

LABORAL	en conflicto con la organización y paralizan las actividades del negocio.	los departamentos críticos están en conflicto con la organización y han paralizado gran parte de las actividades del negocio.	descontento del personal y rumores de paralización de las actividades del negocio.	ha elevado quejas formales.	laboral se puede definir como normal.

2) ALCANCE

Se muestra a continuación el alcance de este documento, con especial hincapié sobre los procesos de negocio prioritarios en la organización (cuya continuidad se considera crítica para el negocio).

a. Procesos de Negocio

Se definen a continuación los siguientes niveles de criticidad para los procesos de negocio:

- **Nivel 0:** Son procesos críticos. Estos procesos por definición, no deben estar inactivos en ningún momento, por lo que en caso de desastre es necesario que continúen en marcha.
- **Nivel 1:** Procesos que pueden estar parados por un tiempo limitado, y que deben ser recuperados inmediatamente después de los procesos críticos.
- **Nivel 2:** Procesos y trabajos que pueden esperar a la recuperación completa de los sistemas y que no son prioritarios.

PROCESOS DE NEGOCIO				
DEFINICIÓN	Conjunto de tareas relacionadas lógicamente, llevadas a cabo para lograr un resultado de negocio definido (Contratación de personal, pago de nóminas, diseño de BBDD…).			
COD.	**NIVEL**	**NOMBRE**	**DESCRIPCIÓN**	**USUARIO**
PR–01	1	Pedidos.	Realización de pedidos: Material oficina, vales de comida.	Juan …..
PR–02	2	Documentos.	Realización de documentos: cartas, faxes...	
PR–03	2	Llamadas.	Recepción y emisión de llamadas telefónicas.	
PR–04	2	Visitas.	Recepción de visitas.	

PR–05	3	Personal.	….....	
PR–06	3	…..	….	

b. Aplicaciones Informáticas

Las aplicaciones informáticas listadas a continuación dan soporte a los procesos de negocio anteriormente descritos, y se agrupan de igual manera, en niveles de 0 a 2 con el mismo significado que en el caso de los procesos.

Aplicaciones Servidor:

- **Nivel 0:** Aplicaciones prioritarias vinculadas implícitamente con procesos de negocio críticos.
- **Nivel 1:** Aplicaciones necesarias que deben esperar a ser instaladas tras las de NIVEL 0.
- **Nivel 2:** Aplicaciones que pueden esperar a ser instaladas después de las de los niveles 0 y 1, ya que no son prioritarias y pueden estar sin funcionar un tiempo sin que afecte en gran medida a la operativa de la organización.

APLICACIONES SERVIDOR			
DEFINICIÓN	Son aplicaciones que dan soporte a los procesos de negocio descritos. Deben ser instaladas en caso de activación del PCN en el orden indicado.		
CODIGO	**NIVEL**	**NOMBRE**	**DESCRIPCIÓN**
AS–01	0	S.O.	Sistema Operativo Win SERVER.
AS–02	0	S.O.	Sistema Operativo Linux Debian.
AS–03	1	QMAIL.	Servidor Correo Linux.
AS–04	1	BD POS.	Gestor BDD Posgress.
AS–05	…..	…...	Configuración conectividad y Firewall.

c. Aplicaciones Cliente:

- **Nivel 0:** Aplicaciones cliente necesarias para el control u operación de las aplicaciones prioritarias.
- **Nivel 1:** Aplicaciones cliente necesarias consideradas como prioritarias, pero que no están vinculadas explícitamente a procesos críticos.
- **Nivel 2:** Aplicaciones que pueden esperar a ser instaladas después de las de los niveles 0 y 1, ya que no son prioritarias y pueden estar sin funcionar un tiempo sin que afecte a la operación de la organización.

APLICACIONES CLIENTE			
DEFINICIÓN	Son aplicaciones que dan soporte a los procesos de negocio descritos. Deben ser instaladas en caso de activación del PCN en el orden indicado.		
CODIGO	**NIVEL**	**NOMBRE**	**DESCRIPCIÓN**
AC–01	0	S.O	Win XP Proffessional SP2.
AC–02	1	S.O.	Linux Debian.
AC–03	1	OPEN OFFICE..	Paquete ofimático.
AC–04	….	……..	……...

3) Escenarios de Recuperación

Escenario A:

Una o varias máquinas importantes han sido afectadas en su totalidad. El CPD está en marcha, y su infraestructura básica está totalmente operativa. La infraestructura de red puede haberse visto afectada. No requiere la activación del PCN.

Escenario B:

Parte del CPD se declara inoperativo. Existe conectividad con el exterior, aunque presenta problemas de conectividad hacia el interior. Algunas máquinas que se encuentran en el CPD no funcionan y las instalaciones presentan algún problema de seguridad.

Escenario C:

Gran parte del CPD se declara inoperativo. No existe conectividad con el exterior, presenta serios problemas de conectividad hacia el interior. Varias máquinas que se encuentran en el CPD no funcionan y las instalaciones presentan serios problema de seguridad.

Escenario D:

Las oficinas están operativas, no existe riesgo para el personal, pero no se puede operar desde la red interna ni externa. Los equipos cliente funcionan sin conectividad, las líneas telefónicas aún se mantienen aunque pueden presentar ciertas anomalías, a los servidores se puede acceder físicamente pero no por red.

Escenario E:

Las oficinas se declaran no operativas a todos los efectos, la red interna y externa no funcionan, so se tiene acceso a ningún equipo, servicio, ni red. Las líneas de telefonía fija están fuera de servicio.

4) Recursos asociados al PCN

Equipo de Gestión y Recuperación

Decide si se producen los criterios para decretar una situación de emergencia.

Mantiene informada a la Dirección de la evolución de la recuperación ante un desastre.

Coordina las actividades para la pronta recuperación de las operaciones.

Solicita a la Dirección la autorización para los gastos monetarios derivados de la puesta en marcha del PCN y para la adquisición de recursos nuevos en caso de que fueran necesarios.

El equipo de Gestión y Recuperación está formado por:

- Responsable de los Planes de Contingencia.
- Responsables de los departamentos.
- Otros miembros que el Responsable de los Planes de Contingencia considere necesarios.

Recursos Materiales

Lo descrito a continuación incluye recursos materiales, recursos de infraestructura, lugares de reunión y toda aquella información crítica susceptible de ser requerida por el Equipo de Gestión y Recuperación frente a un desastre.

Equipamiento Informático:

- El equipamiento informático disponible para la recuperación serán todos los servidores, ordenadores de sobremesa y portátiles operativos, así como los periféricos que pudieran ser recuperados de las oficinas de la organización.
- Los equipos objetivos de recuperación inmediata y prioritaria en este PCN son ……que son los que dan soporte a los procesos críticos esenciales para el negocio.
- Los equipos cliente se recuperarán progresivamente a medida que sea posible.

CPD Alternativo:

⇒ Debido a la baja probabilidad de que quede inservible el CPD de la oficina y al alto coste económico de mantener un CPD

alternativo, la Dirección asume el riesgo de no tener un CPD alternativo, debiendo incrementar toda medida que permita una rápida instalación y replicación del mismo en no más de 5 (cinco) días.

Documentación Crítica

Se entiende por documentación crítica toda aquella información en papel o soporte electrónico necesaria para la continuidad del negocio en caso de desastre. Por ello, su almacenamiento y clasificación son prioritarios en todos los niveles.

Se debe tener en cuenta que existe la posibilidad remota de que no se puedan recuperar los datos desde las copias existentes de backup en la oficina de la organización. Por ello, la documentación citada a continuación, se hace imprescindible para recuperar las bases de datos de las aplicaciones críticas tras el desastre.

La información considerada como crítica es la siguiente:

- Información sobre clientes.
- Contratos y cualquier tipo de información válida sobre proveedores, bancos.
- Copias de software, manuales y demás documentación relacionadas con las maquinas del CPD.
- Listado de empleados, con especial atención a aquellos que están involucrados en la puesta en marcha de los procesos críticos, y todos aquellos que pueden ser llamados.
- Listado de direcciones y teléfonos de interés.
- Planos e información básica sobre el edificio ……
- ……
- ….

Mantenimiento y Pruebas del PCN

Para el correcto funcionamiento del plan, se requiere la existencia de un Grupo Permanente de Mantenimiento compuesto por:

Responsable de Mantenimiento del Plan:

- Coordina la actualización del PCN y lo mantiene al día en todas sus secciones y anexos.
- Promueve la importancia del plan y la necesidad de su continuo mantenimiento entre todas las personas en que se estime oportuno.

- Identifica y se pone en contacto con las personas que deben gestionar los grupos de trabajo en caso de desastre, informándolos sobre sus funciones.
- Promueve las reuniones con los responsables de Departamento, responsables de la puesta en marcha de los procedimientos alternativos a los procesos críticos de manera que en ellas se definan:
  - Las bases de datos y la documentación necesaria para la puesta en marcha de los procesos en caso de desastre.
  - En caso de almacenar información sensible en lugares no controlados directamente por el personal de la organización, especificar aquellos métodos de protección.
  - Realizar revisiones para comprobar la validez y vigencia del plan.

Responsables de Departamento:

- Personas que tienen bajo su responsabilidad la ejecución de procesos de negocio de niveles 0 y 1. Su tarea es describir y mantener actualizada la información necesaria para que los procesos de negocio críticos puedan ponerse en marcha con éxito.

Responsable de Mantenimiento:

- Actualización y mantenimiento de los listados anexos a este documento.
- Obtención y posterior almacenamiento de la documentación crítica que se considere esencial para la continuidad del negocio en caso de desastre.
- Actualización y mantenimiento de este documento, así como de hacerlo llegar a las personas contenidas en su Lista de Distribución.

Para que la aplicación del PCN tenga alta probabilidad de aplicarse con éxito en caso de una emergencia real, deben programarse simulacros de actuación a fin de que los equipos involucrados en estos planes estén familiarizados con las formas de actuación. Estas pruebas deben:

- Garantizar que los procesos de backup funcionan, y son efectivos en caso de desastre total.
- Decidir si la infraestructura de recuperación está disponible y funciona sin problemas.
- Decidir sobre la validez del procedimiento de recuperación planificado y desarrollado en este documento.
- Mostrar que los usuarios no tienen grandes problemas de acceso a los sistemas y pueden seguir trabajando tras el desastre.
- Quedar documentadas, solucionar cualquier problema, desperfecto o fallo que en el plan se detecte con las pruebas.

5) Procedimientos de respuesta y recuperación

Consideraciones Previas

Todas las aplicaciones informáticas tienen un responsable de aplicación de su mantenimiento tras la puesta en producción de las mismas.

Los procesos de negocio se relacionan con una cierta cantidad de documentación crítica de la que debe existir al menos una copia almacenada en un lugar externo a las oficinas de la organización.

Existe un listado de proveedores (Como ANEXO B) al cual se debe acceder en caso de necesidades tales como alquiler de nuevas líneas telefónicas, servicios de telecomunicaciones, software o hardware.

Se ha predefinido previamente un grado de criticidad para todos los procesos de negocio y para todas las aplicaciones informáticas.

El procedimiento global del PCN consta de las siguientes fases:

- Activación del PCN, identificando el tipo de desastre y el escenario.
- Formación del Equipo de Gestión y Recuperación.
- Convocatoria de los Responsables de Departamento y puesta en marcha de los procedimientos alternativos de los procesos críticos fundamentales para la organización (en caso de que sea necesario).
- Habilitación de la instalación del CPD de backup para recuperar la funcionalidad informática (en caso de que sea necesario).
- Habilitación de la recuperación del edificio donde se encuentra las oficinas de la organización y sus sistemas informáticos (en caso de que sea necesario).
- Verificación de que todo ha vuelto a la normalidad.

Activación del PCN

La activación del PCN es el efecto de la recepción de un aviso fiable que uno de los escenarios que lo lanzan se está cumpliendo. La recepción de este aviso puede llegar por diferentes formas:

- Aviso del personal interno (normalmente en horario de oficina).
- Aviso de empresas o personas ajenas a la organización fuera del horario de oficina. Este aviso debe ser verificado convenientemente y se le suele asociar directamente con un desastre.

Procedimientos Alternativos

Los siguientes procedimientos se aplican únicamente a los procesos críticos (Nivel 0) para su puesta en marcha. Se supone que se han dado las siguientes condiciones:

- Ausencia prolongada de fluido eléctrico.
- Ausencia de comunicaciones telefónicas.
- Indisponibilidad de los sistemas informáticos.
- Indisponibilidad de la oficina de la organización.

Procedimientos de Recuperación de Sistemas

Escenario A: ……..

Este escenario implica:

Escenario B: ……..

Este escenario implica:

Escenario C: ……

Este escenario implica:

Vuelta a la Normalidad

El Equipo de Gestión y Recuperación es el único capacitado para tomar la decisión de volver a poner en producción todos los sistemas de la organización.

Para su cumplimiento deberán seguir los siguientes pasos:…..

6) Lista de Distribución de este documento.

La distribución debe de hacerse de forma restringida, ya que este documento describe los procesos y los sistemas críticos para la organización. Deben disponer de copia en papel de este documento:

- Grupo de Mantenimiento y Actualización del PCN.
- Responsable del departamento de Sistemas.
- Componentes del Equipo de Gestión de Recuperación de Desastres.

- Responsables de los departamentos.
- Aquellas personas que el Responsable de los Planes de Contingencia indique.

Una copia de este documento debe estar almacenada en un lugar seguro fuera de las oficinas de la organización (aclarar el lugar…….), al igual que los documentos críticos y bases de datos que el Responsable de los Planes de Contingencia considere necesario.

ANEXO A: lista de distribución

CARGO	DATOS DE CONTACTO
Responsable de los Planes de Contingencia	**Nombre:** **Teléfono:** – **Email:**
Responsable del Mantenimiento del Plan.	**Nombre:** **Teléfono:** **Email:**
EQUIPO DE GESTIÓN Y RECUPERACIÓN	
Departamento de Sistemas	**Nombre:** **Teléfono:** **Email:**
Departamento……	**Nombre:** **Teléfono:** **Email:**
Departamento……..	**Nombre:** **Teléfono:** **Email:**
OTROS RESPONSABLES	
Director General	**Nombre:** **Teléfono:** **Email:**
……	**Nombre:** **Teléfono:** **Email:**
Gerente……	**Nombre:** **Teléfono:** **Email:**

ANEXO B: lista de proveedores y partners

EMPRESA	DATOS DE CONTACTO
……	**Nombre:** **Teléfono:** – **Email:**
……	**Nombre:** **Teléfono:** **Email:**

8.7.5. Proceder ante incidentes

A continuación presentamos un conjunto de tareas que pueden servir como guía para preparar, documentar y realizar como parte de un plan global de: preparación, seguimiento, respuestas y análisis forensic sobre incidentes. Se trata sólo de un breve resumen sobre las mismas, algunas de ellas se pueden agrupar dentro de otras, acorde a como se estructure esta actividad.

Nro	Tarea	Descripción
1	Pasos a seguir ante incidentes	Este documento puede constar de dos partes: - De carácter general debe cubrir los aspectos para cualquier personal que forme parte de la organización. - Particular: para el personal técnico (Detección, notificaciones, identificación, recuperación, adquisición de datos, monitorización y Logs, reconstrucción de sistemas y aseguramiento, imitación en sistemas similares.
2	Pasos a seguir en la recolección de información para análisis forensic	Quién, qué cuándo y cómo !!! - Creación de línea de tiempo. - Análisis de memoria, procesos, y disco. - Análisis de puertos y direcciones. - Análisis de archivos y directorios. - Análisis de ausencias o excesos de espacio en disco. - Análisis de red y sistemas de Logs. - Análisis de binarios por puertas traseras.
3	Aspectos que se deben analizar para la recolección de información para análisis forensic	Forma de documentar, parámetros, fecha/hora, , direcciones, nombres, rastros, aspectos legales (pruebas), no modificación ni alteración de información, copias, herramientas disponibles, metodología, hardware y software a emplear,

		personal interviniente,.
4	Preparación para incidentes	Metodología para instruir y actualizar al personal abocado al tratamiento de incidentes. - Políticas. - Procedimientos. - Planes.
5	Simulación de Incidentes	Planificación, implementación y explotación de ejercicios de simulación de incidentes en todas sus fases, niveles y ámbitos (trabajo, aula y gabinetes de crisis)
6	Preparación para análisis forensic	Metodología para instruir y actualizar al personal abocado al análisis forensic.
7	Políticas y procedimientos para análisis forensic	- Políticas. - Procedimientos. - Planes.
8	Cadena (árbol) de llamadas	Documento con el esquema de llamadas correspondiente con todos los canales primarios y alternativos para la ubicación de las personas involucradas.
9	Cadena (árbol) de escalada	Línea a seguir, nivel de criticidad y responsable de cada nivel para continuar escalando en grado de importancia de un determinado incidente.
10	Inventarios de hardware y software	- Sistemas y responsables - Aplicaciones. - Información de contactos.
11	Planos de Red	- El mayor grado de detalle posible. - Elementos activos y pasivos. - Direcciones. - Servicios. - Derechos y obligaciones. - Operatoria.
12	Formularios de reportes, planillas e informes	Se trata aquí de analizar los formatos más simples que aporten la información necesaria (ni mucha ni poca), en forma concisa y clara.
13	Métodos de comunicación	Reportes, teléfonos, hotline, mail. SMS, etc.
14	Formación de personal en Procedimientos Operativos Normales (PON)	Operadores, administradores, técnicos, gerentes, etc.

15	Herramientas disponibles	- Inventario de las mismas. - Secuencia de empleo. - Manuales de empleo. - Ubicación. - Responsables.
16	Procedimientos de Logs	Metodología, envíos, túneles, seguridad, centralización, normalización, resguardo, consulta o visualización.
17	Procedimientos de tiempo	Metodología, empleo de protocolo NTP, sincronización, monitorización.
18	Procedimiento de monitorización	Empleo de sniffers, zonas, activación, filtros, almacenamiento, capturas, formatos,
19	Procedimientos de resguardo de información general	Metodología, responsables, seguridad, redundancia, inventarios, planes de verificación periódica, políticas de destrucción.
20	Políticas de privacidad de la información	- Procedimientos respecto al carácter y uso de la información. - Clasificación de la misma. - Derechos y obligaciones. - Responsables. - Difusión.
21	Laboratorio de Forensic	Funciones, misiones, objetivos, herramientas,
22	Política de Seguridad	- Grado de exposición al que se desea llegar: - Cantidad de información que se desea exponer: - Metodología de trabajo en el sistema informático de la Organización: - Grado de acercamiento con otras entidades: - Importancia de la seguridad dentro de la Organización: - Presupuesto que se desea invertir para esta tarea: - Personal que se dedicará al tema: - Grado de compromiso del más alto nivel:
23	Medios de Comunicación	- Proceder con los medios de difusión. - Información que se puede y no puede transmitir. - Responsables de las comunicaciones. - Plan de escalada en las comunicaciones. - Aspectos legales del trato con personal externo a la Organización.
24	Plan de Seguridad	**2.1. Análisis de riesgo:** 2.1.1. Identificación de recursos: 2.1.2. Identificación de actividades:

2.2. Lineamiento del plan:

2.2.1. ¿Quién está autorizado a usar los recursos?
2.2.2. ¿Cuál es el uso correcto de recursos?
2.2.3. ¿Quién está autorizado a crear usuarios y conceder accesos?
2.2.4. ¿Quiénes pueden tener privilegios administrativos?
2.2.5. ¿Cuáles son las responsabilidades de los administradores del sistema?
2.2.6. ¿Qué hacer con la información sensible?
2.2.7. ¿Que sucede si el plan es violado?
2.2.8. Proceder ante incidentes:
2.2.9. Publicación del plan:

3. Análisis de detalle:

3.1. Identificación de problemas reales:

3.1.1. Puntos de acceso:
3.1.2. Configuración de sistemas:
3.1.3. Bug de software:

3.2. Medidas de protección:

3.2.1. Protección de recursos:
3.2.2. Seguridad física:
3.2.3. Reconocimiento de actividad no autorizada:
3.2.3.1. Monitoreo de los sistemas en uso:
3.2.3.2. Analizadores de protocolos:
3.2.4. Comunicación del plan de seguridad:
3.2.4.1. Educación de usuarios:
3.2.4.2. Educación de administradores:
3.2.5. Procedimientos de resguardo y recuperación:

3.3. Recursos para prevención de ataques:

3.3.1. Conexiones de red, módems, routers, proxys y Firewalls:
3.3.2. Confidencialidad:

3.3.2.1. Criptografía:
3.3.2.2. Privacidad en el correo electrónico:
3.3.3. Autenticación:

3.3.4. Integridad de la Información:
3.3.5. Fuentes de información:

4. Procedimientos normales:

4.1. Actividades agendadas:
4.2. Test de procedimientos:
4.3. Procedimientos para la administración de cuentas:
4.4. Procedimientos para la administración de contraseñas:

4.4.1. Selección:

		4.4.2. Cambios: **5. Procedimientos ante incidentes:** Como referencia se detallan a continuación los aspectos que se deben considerar en el plan: - Asegurar l a integridad de los sistemas críticos. - Mantener y restaurar datos. - Mantener y restaurar servicios. - Determinar cómo sucedió. - Detener escalamiento o futuros incidentes. - Detener la publicidad negativa. - Determinar quién lo hizo. - Penalizar a los atacantes. **5.1. plan contra incidentes:** **5.2. Determinación del problema (Evaluación):** **5.3. Alcance:** **5.4. Notificaciones:** 5.4.1. Información explícita: 5.4.2. Información verídica: 5.4.3. Elección del lenguaje: 5.4.4. Notificaciones a individuos: 5.4.5. Aspectos generales a tener en cuenta par las notificaciones: **5.5. Respuestas:** 5.5.1. Contención: 5.5.2. Erradicación: 5.5.3. Recuperación: 5.5.4. Seguimiento: **5.6. Registros:** **6. Procedimientos post incidentes** **6.1. introducción:** **6.2. Remover vulnerabilidades y depuración de sistemas:** **6.3. Lecciones aprendidas:** **6.4 Actualización de políticas y planes:**

8.7.6. Conceptos militares de Recuperación de desastres

En las operaciones militares el tema de recuperación ante "imponderables", es algo que se analiza desde siempre pues este tipo de problemas es cotidiano en el combate. La experiencia militar reunida a lo largo de miles de años es un punto de partida muy importante a la hora de evaluar este problema desde un punto de vista informático.

La presente metodología es la empleada por algunas fuerzas armadas en su adaptación de los sistemas informáticos desde su red administrativa a su red operacional. La rede administrativa de una fuerza militar es la que sustenta sus actividades cotidianas en tiempo

de paz, su red operacional es la que sustenta toda actividad una vez que se ha entrado en operaciones. Es natural deducir que la gran mayoría de la información y sistemas que se mantiene y opera en tiempo de paz, es necesaria también durante cualquier tipo de conflicto, por lo tanto, la mayoría de las fuerzas militares disponen hoy de lo que se denominan sistemas **"C^3I": Comando, Control, Comunicaciones e Informática.** Este tipo de sistemas, suele tener varios componentes esenciales que son vehículos en los cuáles se integran todos los recursos telemáticos para prestar estos servicios, y los podríamos definir como una especie de "Puente" de unión entre ambas redes (si bien se trata de verdaderos centros tecnológicos de la más avanzada sofisticación). Estos centros deben estar en capacidad de proporcionar toda la información necesaria para el combate, y continuar operando, ante cualquier circunstancia. El caso extremo es la destrucción de cualquiera de ellos, ante lo cual, la operación debe continuar.

Un aspecto ilustrativo y que siempre me ha despertado el interés (al relacionarlo con la seguridad informática) es que se trata de un "Blanco extremadamente fácil", pues por triangulación, se puede obtener inmediatamente su posición, pues os podréis imaginar que en el mismo confluyen y parten todo tipo de señales electromagnéticas, por esa razón es que en combate real, deben encontrarse en permanentes cambios de posición. Como ejemplo de esto, podemos citar que en la segunda guerra mundial, apareció como concepto de técnicas de transmisión "Forward" (que tratamos al inicio de este libro) una extremadamente rara, que es la que empleaban los submarinos, los cuales cuando debían transmitir un mensaje de radio, desplegaban su antena, que no era más que una sencilla boya unida por un cable al submarino, al llegar esta a superficie se podía comenzar transmitir, pero por las inmutables leyes de la naturaleza, una boya sube inexorablemente de forma vertical, por lo tanto la misma marcaba perfectamente la posición de este navío, el cual era detectado inmediatamente, por esta razón, los mensajes se transmitían textuales, tres o cuatro veces, inmediatamente se replegaba la antena y el submarino huía a toda velocidad de esa posición. Con esto se lograba, que si el mensaje inicial se decepcionaba con algún tipo de error, se podía reconstruir en base a los dos o tres restantes. Todo esto, es al sólo efecto ilustrativo, pero para que podamos ser conscientes que el concepto de "Recuperación de desastres" en este tipo de elementos es vital y concreto, por esa razón es que lo citamos como ejemplo, pues verdaderamente tiene mucho camino avanzado.

El mayor problema que se plantea en el caso militar es la interacción permanente que deben tener estos centros con la red administrativa en tiempos de paz, pues de ésta es de donde deben obtener la información básica para poder operar y mantenerla siempre actualizada, y de esta forma entrar en operaciones en el momento en que sea necesario respondiendo a la totalidad de las necesidades del combate y ante cualquier circunstancia.

El enfoque militar parte de una serie de pasos a cumplir desde el inicio de cualquier implementación del sistema, hasta la incorporación de todo nuevo elemento, los mismos se detallan a continuación (Respetaremos la puntuación que suele emplear la doctrina militar en estos casos):

1. Análisis de la situación:

a. Información General (sobre cada sistema)**.**

- Composición:
- Disposición:
- Localización:

b. Información particular de esta implementación o sistema.

- ¿Es vital?
- ¿Qué funcionalidad tiene?
- ¿Hace cuánto está en servicio?
- ¿Qué resultados ha logrado?
- ¿Cuáles son los próximos pasos a corto plazo?
- ¿Qué grado de peligrosidad se le asigna en su servicio?
- ¿Qué impacto pueden causar su caída?
- ¿Qué medidas se deben empezar a analizar?
- ¿Ha cometido errores o fallos?

c. Elementos de nivel superior que interaccionan con el mismo

Estructura similar al anterior

d. Elementos adyacentes que interaccionan con el mismo.

Estructura similar al anterior

e. Otros:

- Tipos de vínculos de conexión (Protocolos, anchos de banda, empresas prestadoras, etc.).
- Administración de los dispositivos de interconexión (Router, Switch, módem, etc.).
- Metodología de acceso
- Niveles de acceso.
- Permisos de acceso (Nombres, direcciones, puertos).
- Horarios de acceso.
- Monitorización de los vínculos.
- Medidas de seguridad combinadas.
- Medidas redundantes.
- Métodos de autenticación empleados.

- Dispositivos de seguridad entre las redes.
- Vínculos de salida al exterior de las otras redes.
- Relaciones entre los dominios administrados.
- Derechos y obligaciones de los administradores de las otras redes.

f. Terceros (Sistemas y/o elementos externos a la empresa):

- Descripción de cada empresa.
- Descripción de los administradores de red externos.
- Responsabilidades en cada vínculo.
- Dispositivos propios de seguridad.
- Medidas particulares en estos accesos.
- Transitividad de los accesos (Si se deja acceder a A a esta red y A deja acceder a B a la suya, ¿puede acceder B a esta red?).

2. Determinación de los requisitos mínimos operacionales:

Este paso es el más importante a seguir, pues un análisis exhaustivo del mismo es el que dará como resultado la relación óptima de COSTE/BENEFICIO, es decir que si se realiza con la máxima atención se lograrán los efectos deseados con la mínima inversión de recursos.

Se ha detectado que en general, las empresas tienen la tendencia a realizar la grave hipótesis simplificativa de reducir el problema de recovery disaster a los centros de cómputo, lo cual es una peligrosa visión del problema, pues un imponderable, puede producirse en muchos puntos/sistemas/canales/personas y procesos. Por lo tanto el planteamiento de los imponderables debe hacerse en todos los eslabones de esta cadena, pues de nada serviría duplicar completamente un centro de cómputos, si no se tuvo en cuenta que el director general de la empresa se quedó sin acceso al mismo, o el punto de conmutación de las conexiones falló, etc.

A continuación se presentan los aspectos a considerar:

a. Análisis de puntos débiles:

- Proveedores de información.
- Recolectores de información.
- Procesadores de Información.
- Comunicaciones.

- Puntos de control (Conmutación) de comunicaciones.

b. Niveles de Intervención:

- Cadena de comandos.
- Administradores.
- Usuarios.
- Terceros.

c. Interrogantes de elementos y/o recursos:

- ¿Quién?
- ¿Qué?
- ¿Cuándo?
- ¿Dónde?
- ¿Cómo?
- ¿Para qué?

3. Debilidades asumidas:

En este punto, se debe reconocer cuáles son los aspectos que se pueden asumir, y que si bien debilitan el funcionamiento, permiten obtener los resultados deseados. Se debe analizar independientemente de la magnitud que se piensa previamente adoptar y detallar un listado de los mismos con el mayor grado de detalle.

4. Determinación del nivel de eficiencia:

Sobre la base de los tres puntos anteriores, se determinará definitivamente el grado de eficiencia que se adoptará para este elemento/servicio. Se debe tener en cuenta, que para el funcionamiento integral del sistema, no todas las partes deberán ser óptimas, sino que cada una debe ser tratada independientemente.

Los niveles de cada una de ellas se los acotará a los tres siguientes:

- Mínimo Operacional.
- Eficiente.
- Óptimo.

5. Plan de acción.

El plan de acción es el verdadero **¿Cómo?**, responde al análisis de los puntos anteriores y permite documentar y determinar la cronología de acciones a seguir y las medidas a adoptar en cada una de ellas, tanto para implementarlas, como para mantenerlas actualizadas.

a. Concepto de la Operación:

El concepto de la operación definirá en grande los pasos a seguir y la secuencia de los mismos, se tendrá en cuenta fundamentalmente lo siguiente:

- Topología (Configuración).
- Accesos y comunicaciones
- Interconexión de zonas.
- Recursos.
- Acciones a tomar.
- Medidas y contra medidas a adoptar.
- Plan de mantenimiento y actualización.
- Plan de control y auditoría.
- Pruebas funcionales.

b. Actividades particulares a considerar:

Dentro de cada sistema/elemento a incorporar, surgirán una serie de módulos/elementos o subsistemas. Cada uno de ellos debe ser analizado en detalle y teniendo en cuenta las consideraciones particulares de cada uno de ellos. En este punto, se describen aspectos que afectan a cada componente en particular y no al resto. Se realizará un apartado por cada uno de ellos. Los aspectos más importantes de cada uno son:

- Perfil del elemento.
- Especificaciones técnicas.
- Topología.
- Accesos y comunicaciones.
- Recursos expuestos.
- Acciones a tomar.

c. Apoyos:

En este párrafo se desglosan todos los aspectos relacionados a colaboraciones ajenas al concepto de este plan.

d. Operaciones de seguridad:

Se detallan todas las medidas a adoptar para mantener el servicio o funcionamiento del sistema y también se analiza cualquier posible debilidad en cuanto a fugas de información o vulnerabilidades que se presentan con estas acciones.

e. Operaciones de Información:

Este párrafo define el conjunto de medidas a adoptar para:

- Catalogación.
- Almacenamiento.
- Tránsito.
- Recuperación.
- Destrucción.

De toda la información en juego durante estas acciones.

f. Operaciones de engaño.

Este aspecto que suele considerarse trivial, no lo es tanto, pues se trata aquí de contemplar un hecho cada vez más frecuente en los sistemas informáticos. Este detalle es la consecuencia de desastres producidos por acciones de hacking, factor muy importante a considerar. Como ya hemos tratado, el objetivo de la actividad de intrusos puede ser muy amplio, contemplando destrucción y/o caída de sistemas. El proceso de "convivencia" con este tipo de vandalismo, requiere en la mayoría de los casos, un conjunto de acciones a seguir, las cuales van directamente relacionadas con el plan de recuperación del sistema. Si estos pasos no son analizados, planificados y evaluados con anterioridad, es muy difícil poder mantener un determinado servicio desde el momento en que se producen estos delitos. Una de las acciones básicas es contemplar la posibilidad de que exista un servicio "Real" y uno "Ficticio" para la correcta operación del sistema, muchas de estas medidas son las que presentamos por medio de “Honey Pots”.

g. Contramedidas.

Ante la opción anterior, independientemente de las medidas adoptadas para el "desvío" de estas acciones, se debe considerar la posibilidad de reacción ante la acción de un intruso, esto es lo que se analiza dentro de este apartado.

h. otros.

x. instrucciones de coordinación

Contiene las instrucciones aplicables a dos o más aspectos particulares de los párrafos anteriores, es decir aspectos comunes del plan.

Contempla cualquier prescripción necesaria sobre:

- Objetivos tanto finales como intermedios.
- Ritmo de la acción.
- Líneas de coordinación (temporales y físicas).
- Esquemas de tiempo.
- Procedimientos operativos normales.
- Plan de comunicación e interacción entre responsables.

6. Logística:

Se consideran aquí los aspectos relacionados a apoyos logísticos que se necesitan o que se poseen.

- Abastecimiento
- Mantenimiento.
- Transporte.
- Trabajo.
- Obras.
- Servicios.

7. Mando y transmisiones:

a. Mando

Refleja las ubicaciones de los sistemas de mando y control. Determina al menos una ubicación futura para cada uno.

b. Transmisiones

Relaciona las instrucciones para el enlace a utilizar.

Referirse a un Anexo "Transmisiones", si se necesita.

EJERCICIOS DEL CAPÍTULO 8

1. Ejercicos con GPG:

Primero generaremos nuestro par de claves:

#gpg -gen-key

Luego generamos un certificado de revocación:

#gpg—output c878f0d3.asc—gen-revoke 0xc878f0d3

Para verificar todo lo que se ha creado podemos hacerlo consultando en el directorio /usuario/.gnupg/

Una actividad que suele ser muy útil es exportar las claves en binario y en ASCII (--armor)

#gpg --output ale.gpg -export acorletti@hotmail.com

#gpg –armor -output ale_ASCII.gpg -export acorletti@hotmail.com

Verificarlo consultando lo que se ha creado en ese directorio (“ls -l”)

Verificar (o añadir) en el archivo **gpg.conf** el servidor de claves:

Por ejemplo:

keyserver hkp://keys.gnupg.net

Ver nuestras claves: gpg -edit-key ale

(entraremos en modo “comando” “>” - el comando “help” nos muestra todas las opciones)

toggle (conmuta entre priv y pub)

Ahora nos dedicaremos a: buscar, refrescar, exportar e importar claves de un servidor de claves:

#gpg -keyserver keys.gnupg.net -send-key c878f0d3

#gpg -keyserver keys.gnupg.net -search-key ivancorletti@hotmail.com

#gpg -keyserver keys.gnupg.net -recv-key 7DBA961C

#gpg -keyserver keys.gnupg.net -refresh-key

Una vez importadas las claves, es necesario validarlas:

#gpg -list-keys

#gpg -edit-key ivancorletti@hotmail.com

- fpr (de “fingerprint”, y se verá su “Huella digital”)
- sign (firmar)

  ¿Está seguro? si

  Entre su frase de contraseña:
- check (para verificar su situación ahora que está firmada)

- quit (para salir) ¿Guarda los cambios? (s/N) s

Para cifrar documentos podemos hacerlo:

Con clave asimétrica:

#gpg -output pp2.gpg -encrypt pp1 (nos pedirá los IDs de los usuarios que podrán descifrar luego)

#Gpg –output -decrypt pp2.gpg

Con clave simétrica:

#gpg -output pp3.gpg -symmetric pp1

#gpg -decrypt pp3.gpg

Firma digital: (por defecto firma y cifra)

#gpg -output pp1.sig -sign pp1

#gpg -output pp4.desig -decrypt pp1.sig

Firma digital sin cifrar:

#gpg -clearsign pp1 (Generará pp1.asc)

para verlo: #vi pp1.asc

Para acompañar el documento con la firma:

#gpg -output pp5.sig_attach -detach-sig pp1

Para verificarlo:

#gpg -verify pp5.sig_attach pp1

Generar resúmenes

#sha256sum pp1 >>pp1.sha256 (Genera el resumen y lo guarda en pp1.sha256)

#sha256sum -c pp1.sha256 ("Check", Verifica su integridad)

EJERCICIOS PROPUESTOS CON GPG:

- Generar una clave.
- Generar un certificado de revocación
- Exportar las claves en binario y en ASCII
- Ver nuestras claves
- Buscar, Refrescar, Exportar e importar claves al servidor de claves: keys.gnupg.net (Hacerlo en equipos y reunir 3 o 4 claves en cada host)
- Verificar su huella digital y firmarlas (Verificar que estén firmadas).
- Generar un archivo de texto cualquiera en un directorio exclusivo para este trabajo.

- Emplear cifrado asimétrico, e intentar realizar varias acciones: con la clave de uno, de dos, de tres alumnos. Enviarlos por correo y verificar quién puede y quién no descifrarlo. Comparar los archivos originales con los cifrados.
- Realizar los mismos ejercicios con clave simétrica. ¿Se te ocurre cómo hacerle llegar esta clave al usuario final?
- Verificar todas las opciones de firma digital (con criptografía, sin criptografía, acompañando la firma) Verificar su firma y resultado si se altera algún bit
- Generar resúmenes sha-256 desde archivos de diferente tamaño, dirigirlo a un archivo cualquiera.
- Alterar algo y generar nuevamente el hash, alterar el hash y verificar el resumen

PARTE II

Seguridad por niveles

1. Nivel Físico.

En este nivel, como ya dijimos, son de especial importancia los aspectos mecánicos, eléctricos u ópticos y los procedimentales.

1.1. Aspectos mecánicos:

Aquí revista especial importancia para auditar el canal de comunicaciones que se emplee, este puede ser:

- *Propio o arrendado:* Un vínculo propio si pasa exclusivamente por caminos de acceso no público, incrementa la seguridad de interceptación. Por el contrario si es arrendado, se debe ser consciente que puede ser interceptado; para este caso existen estrategias de canal seguro, túneles o criptografía que incrementa la seguridad.
- *Cable de cobre*: Este medio presenta la característica que es difícil detectar su interceptación física "*Pinchado de línea*".
- *Fibra óptica:* La fibra óptica se la puede considerar imposible de interceptar, pues si bien existen divisores ópticos, la colocación de los mismos implica un corte del canal y una fácil detección por pérdida de potencia óptica.
- *Láser:* Este medio genera un haz de luz prácticamente lineal, apuntado a un receptor, el cual es el único punto en el que impacta la señal. Si bien es interceptable, en forma similar a la fibra óptica se detecta con facilidad, y a su vez por encontrarse visualmente unidos, con una inspección óptica se reconoce su trayectoria.
- *Infrarrojo:* Este se implementa de dos formas, de alcance directo y por reflexión. El primero se lo emplea en distancias extremadamente cortas, y el segundo se refleja en las paredes de los ambientes, llegando parte de esta señal al receptor, por lo tanto es altamente vulnerable si se encuentra dentro de los locales de alcance (que es muy reducido).
- *Radiofrecuencia*: Las distintas ondas de radio cubren una amplia gama de posibilidades, desde la HF hasta las microondas y hoy las LMDS (Local Multipoint Distributed Signal). En general cualquiera de ellas son interceptables y su análisis de detalle implica el tipo de señal (digital o analógica), el ancho de banda disponible, el tipo de modulación, y la frecuencia empleada.
- *Satélite*: Si bien se trata de radiofrecuencia, su implementación difiere en el hecho de poseer una antena reflectora llamada satélite a 36.000 km de altura en el caso de los geoestacionarios. Este recibe la señal proveniente de tierra si se encuentra dentro de su cono de aceptación (área de cobertura), le cambia de frecuencia y la reenvía dentro de su cono de aceptación. La conclusión cae de maduro, cualquiera que se encuentre dentro de este cono, está en capacidad de escuchar la señal.

Cada uno de ellos implica una característica diferente en su auditoría de seguridad. Para poder iniciar su auditoría **el punto de partida excluyente son los planos de la red**. En los mismos se deberá auditar los siguientes detalles:

- Identificación de los canales: Aquí debe estar claramente marcada su numeración, extremos, puestos de trabajo conectados y bocas vacantes.
- ¿Cuáles son los tramos críticos?: Se debe analizar las áreas de la Empresa donde físicamente residen las cuentas que tramitarán la información de mayor importancia. Sobre estos canales incrementar las medidas de seguridad, en lo posible emplear fibra óptica.
- Gabinetes de comunicaciones: Ubicación, llaves, seguridad de acceso al mismo, componentes que posee, identificación de las bocas.
- Caminos que siguen: Planos de los locales y perfectamente identificados los conductos que siguen, es eficiente su ubicación por colores (Zócalos, bajo pisos, falso techos, cable canal, etc.).
- Dispositivos de Hardware de red (teniendo en cuenta solo los aspectos físicos): Qué dispositivos existen, su ubicación, claves de acceso, configuración de los mismos, resguardo de configuraciones, permisos de accesos, habilitación o deshabilitación de puertos.
- Dispositivos mecánicos u ópticos de control de acceso: Hoy en día es común encontrarse con dispositivos de control de acceso por tarjetas, biométricos, dactilares, etc. A cualquiera de estos se le debe aplicar los mismos controles que al hardware de red.-
- Certificación de los medios: Mediciones realizadas acorde a lo establecido en la norma TSB–67 TIA/EIA (Telecommunications Industy Association/Electronics Industry Association) que especifica los parámetros de certificación que se deben realizar en los distintos medios de comunicaciones.
- Control de cambios: Toda modificación que frecuentemente se realiza en una red debe quedar documentada y controlada luego de su implementación. El abandono de esta documentación es el primer paso hacia una red insegura a nivel físico, pues en muy pocos años existirá un total descontrol del conectorizado de la red.
- Plan de Inspecciones periódicas: Es importante contar con un cronograma de trabajo que contemple la inspección (recorridas, controles, verificación remota de configuraciones, control de cambios, roturas, etc.) de los detalles anteriormente mencionados, para evitar justamente alteraciones intencionales o no.
- Inventarios de equipamiento: El control de inventarios es una buena medida. En particular haciendo hincapié en cambios y repotenciaciones, pues involucra dispositivos que pueden haber almacenado información.
- Control de actas de destrucción: Toda documentación de importancia o dispositivos de almacenamiento que dejan de prestar servicio o vigencia, debe ser destruido físicamente para imposibilitar un futuro acceso a esa información, dejando una constancia de esta operación, en lo posible controlada por más de una

persona.

- Seguridad física en la guarda de documentación y archivos: Se debe respetar un plan de resguardo de estos elementos acorde a distintos niveles de seguridad.
- Seguridad física de los locales: Todo local que posea elementos que permitan físicamente conectarse a la red debe poseer las medidas de seguridad de acceso correspondiente, y estar claramente identificado quien está autorizado a ingresar al mismo.
- Medidas de resguardo de información: La pérdida de datos es un error grave en un servidor, el responsable de una base de datos, no es el usuario que tiene derecho a no conocer los mecanismos de seguridad en el Backup, sino directamente el Administrador de ese servidor. Las medidas de Backup nunca deben ser únicas, se deben implementar todas las existentes y con más de un nivel de redundancia acorde a la importancia de la información a respaldar (cintas, discos extraíbles, Jazz, etc.).
- Coordinaciones con el personal de seguridad: Los responsables de la seguridad física de la empresa deben contar con una carpeta que regule claramente las medidas de seguridad a tener en cuenta para las instalaciones de red y como proceder ante cualquier tipo de anomalía.
- Se puede auditar también planes y medidas contra incendio, evacuación y contingencias: Todos estos se relacionan en forma directa con la seguridad, pues se debe tener en cuenta que la pérdida de información es una de las responsabilidades más importantes de la seguridad.
- Control de módem, hub y repeater: Los elementos de Hardware que operan estrictamente a nivel 1 son estos tres, pues sólo entienden de aspectos eléctricos u ópticos (Regeneran las señales), mecánicos (Hacen de interfaz entre conectores BNC, RJ-45, Ópticos, DB-25, DB-15, Winchester, etc.) y lógicos (Interpretan los niveles de tensión como unos o ceros). Por lo tanto la configuración de los mismos debe ser auditada aquí. Los aspectos fundamentales a controlar son: direcciones, claves de acceso, programación de puertos, protocolos autorizados, y un especial interés en los que poseen acceso por consola.
- Auditoría de otros componentes de acceso: En esta categoría se debe contemplar módem ADSL, DTU/CSU, PAD en X.25, Placas ISDN, ATM, centrales telefónicas, etc. Se debe prestar especial atención a los Host que tienen conectados módem, llevando un registro de estos, y monitoreándolos con frecuencia, pues aquí existe una peligrosa puerta trasera de la red. No se debe permitir conectar módem que no estén autorizados.
- Un Apartado muy especial debe ser considerado hoy para los dispositivos de **almacenamiento móvil**: Memorias de todo tipo, discos externos, mp3, móviles, ipod, ipad, etc… prestando especial atención a la metodología de conexión: USB, firewire, bluetooth.

1.2. Aspectos Lógicos:

- Análisis de la topología de la red: Este detalle impondrá una lógica de transmisión de la información.
- Estrategias de expansión: El crecimiento de una red es uno de los primeros parámetros de diseño, una red bien diseñada responderá a un crecimiento lógico adecuado, por el contrario, si se parte mal desde el inicio, llevará inexorablemente a un crecimiento irregular que ocasionará la pérdida del control de la misma.
- Asignación de prioridades y reservas para el acceso a la red: Esta medida se lleva a cabo en redes 802.4, 802.5 y **802.11** (mucho cuidado), y permite regular los accesos al canal, es una medida importante a modificar por alguien que desea incrementar su "poder en la red".
- Lógica empleada para VPN (Redes privadas Virtuales): Una capacidad que ofrecen hoy los dispositivos de red, es de configurar puertos formando grupos independientes como si fueran distintos Hub, switch o router. La lógica que se emplea en estos casos es de sumo interés pues en realidad se trata de "redes aisladas lógicamente", las cuales se integrarán o no en un dispositivo de nivel jerárquico superior. Si se encuentra este tipo de empleo, se debe replantear la distribución física de la red, pues a través de estos grupos, la topología lógica de esta red, **diferirá de lo que los planos indican!!!**
- Análisis de circuitos, canales o caminos lógicos: En las redes WAN orientadas a la conexión se programan generalmente en forma previa la conformación de estos medios. Se debe controlar especialmente que no se encuentre nada fuera de lo permitido.
- Puntos de acceso a la red: Auditar que esté perfectamente documentado y que cada una de las puertas de acceso a la red sea estrictamente necesaria pues lo ideal es que exista una sola. Especial interés hoy respecto a las tecnologías inalámbricas.

1.3. Aspectos eléctricos u ópticos:

- Potencia eléctrica u óptica: La irradiación de toda señal electromagnética implica el hecho de ser escuchado (en esto se basa la guerra electrónica). Cuanto menor sea la potencia, más se reduce el radio de propagación. Este detalle es especialmente significativo en antenas o fibras ópticas.
- Rango de frecuencias empleadas: Se debe especificar la totalidad de los canales que se emplean y su tipo (Símplex, semidúplex, dúplex, analógico, digital, PCM, E1, etc.).
- Planos de distribución de emisores y receptores: Se deberá aclarar su ubicación,

características técnicas, alcance, radio y medidas de protección.

- Ruido y distorsión en líneas: Este factor causa pérdida de información y facilita la posibilidad de ataques y detección de los mismos.

2. Nivel de enlace (se referirá a 802.3 o Ethernet, por ser la masa de las redes).

Este nivel comprende la conexión con el nodo inmediatamente adyacente, lo cual en una red punto a punto es sumamente claro, pero en una red LAN, es difícil de interpretar cual es el nodo adyacente. Por esta razón como mencionamos en la teoría IEEE los separa en 2 subniveles: LLC y MAC (LLC: Logical Link Control, MAC: Medium Access Control), en realidad como una de las características de una LAN es el empleo de un único canal por todos los Host, el nodo adyacente son todos los Host.

La importancia de este nivel, es que es el último que encapsula todos los anteriores, por lo tanto si se escucha y se sabe desencapsular **se tiene acceso a absolutamente toda la información que circula en una red**. Bajo este concepto se trata del que revista mayor importancia para el análisis de una red.

Las herramientas que operan a este nivel son las que hemos visto como ANALIZADORES DE PROTOCOLOS, y existen de varios tipos y marcas. Los que son Hardware diseñado específicamente para esta actividad como el Internet Advisor de Hewlett Packard o el Dominó de Vandell & Goltermann, poseen la gran ventaja de operar naturalmente en modo promiscuo, es decir que dejan pasar hacia el instrumental la totalidad de los bit que circulan por el medio de comunicaciones. Los desarrollados como herramientas de Software dependerán del tipo de acceso físico a la red que se posea, pues justamente la mayoría de estos dispositivos asumen tareas de comunicaciones para no sobrecargar con esto a la CPU, por lo tanto existe cierta información que no pasará al nivel superior. Aquí se desarrollará el análisis de las medidas a auditar en el enlace de datos para continuar estrictamente referido a un planteo de niveles.

Qué debemos auditar:

- **Control de direcciones de Hardware**: El objetivo de máxima en este nivel (Pocas veces realizado) es poseer el control de la totalidad de las direcciones de Hardware de la red. Esto implica poseer la lista completa del direccionamiento MAC o también llamado NIC (Network Interface Card), es decir de las tarjetas de red. Si se logra este objetivo, y aperiódicamente se audita la aparición de alguna no contemplada, esta red ofrecerá las mayores posibilidades de éxito en cuanto a la seguridad externa, pues por aquí pasan gran parte de las intrusiones, debido a que es sumamente complejo (si el resto de las medidas controlan los niveles superiores) falsificar una de estas desde el exterior (Se debe dejar claro que no es el mismo razonamiento si se tiene acceso internamente a la red).

- **Auditoría de configuración de Bridge o Switch**: Estos son los dispositivos que operan a nivel 2 (En realidad el concepto puro de Switch es el de un Bridge multipuerto), su trabajo consta de ir aprendiendo por qué puerto se hace presente cada dirección MAC, y a medida que va aprendiendo, conmuta el tráfico por la puerta adecuada, segmentando la red en distintos *"Dominios de colisión*". La totalidad de estos dispositivos es administrable en forma remota o por consola, las medidas que se pueden tomar en su configuración son variadas y de suma importancia en el tráfico de una red.

- **Análisis de tráfico**: En este nivel la transmisión puede ser Unicast (de uno a uno), Multicast (de uno a muchos) o Broadcast (de uno a todos). La performance (rendimiento) de una red se ve seriamente resentida con la presencia de Broadcast, de hecho esta es una de las medidas de mayor interés para optimizar redes y también es motivo de un conocido ataque a la disponibilidad llamado *"Bombardeo de Broadcast*". Otro tipo de medidas es el análisis de los multicast, pues son estos los mensajes que intercambian los Router, y es de sumo provecho para un *interesado en una red ajena* ser partícipe de estos grupos, pues en ellos encontrará servida toda la información de ruteo de la red.

- **Análisis de colisiones**: Una colisión se produce cuando un host transmite y otro en un intervalo de tiempo menor a 512 microsegundos (que es el tamaño mínimo de una trama Ethernet) si se encuentra a una distancia tal que la señal del primero no llegó, se le ocurre transmitir también. Ante este hecho, los dos host hacen silencio y esperan una cantidad aleatoria de "*tiempos de ranura*" (512 microsegundos), e intentan transmitir nuevamente. Si se tiene acceso físico a la red, un ataque de negación de servicio, es justamente generar colisiones, pues obliga a hacer silencio a todos los Host de ese segmento.

- **Detección de Sniffers o analizadores de protocolos**: Esta es una de las tareas más difíciles pues estos elementos solamente escuchan, solo se hacen presentes cuando emplean agentes remotos que colectan información de un determinado segmento o subred, y en intervalos de sondeo, la transmiten al colector de datos.

- **Evaluación de puntos de acceso WiFi**: Como hemos mencionado, esta tecnología sólo es segura si se configura adecuadamente, por lo tanto en esta aspecto es de especial interés verificar qué tipo de protocolos de autenticación se han configurado, los permisos de acceso a estos dispositivos, su potencia de emisión, la emisión de beacons, etc.

- **Evaluación de dispositivos bluetooth**: Aunque no es un tema aún explotado de forma frecuente, no debemos dejar de lado la existencia de este tipo de dispositivos y sobre todo que en muchas aplicaciones y hardware viene activado por defecto, con lo que estando a una distancia adecuada, es posible su explotación.

3. Nivel de red.

Este nivel es el responsable primario de los ruteos a través de la red. Si se trata de la familia TCP/IP, aquí se encontrará la mayor actividad, por lo tanto el centro de atención de la auditoría en este nivel, deberá estar puestos en los mensajes de ruta y direcciones:

- **Auditorías en Router: (**Este es el dispositivo por excelencia en este nivel).

  1) *Control de contraseñas:* Los router permiten la configuración de distintos tipos de contraseñas, para acceder al modo usuario es la primera que solicita si se accede vía Telnet, luego también para el ingreso a modo privilegiado, también se permite el acceso a una contraseña cifrada, la de acceso vía consola y por último por medio de interfaz gráfica por http.

  2) *Configuración del router:* Dentro de este aspecto se contemplan los detalles de configuración que muchas veces en forma innecesaria quedan habilitados y no se emplean (Broadcast Subnetting, local loop, puertos, rutas, etc.)

  3) *Resguardo de las configuraciones:* Un detalle de suma importancia es guardar la **startup-config** en forma consistente con la **running-config**, y esta a su vez en un servidor **t_ftp**, como así también en forma impresa.

  4) *Protocolos de ruteo:* El empleo de los protocolos de ruteo es crítico pues la mayor flexibilidad está dada por el uso de los dinámicos (RIP, IGRP, EIGRP, OSPF), pero se debe tener en cuenta que con esta medida se facilita información para ser aprovechada por intrusos, los cuales a su vez pueden emplearla para hacerse partícipe de las tablas de ruteo (En especial con RIP pues no se puede verificar el origen de los costos de las rutas, en OSPF, es más fácil pues se envía una tabla completa que pertenece a un router específico y a su vez a este se lo puede verificar con dos niveles de contraseña: normal y *Message Digest*). Las tablas de ruteo estáticas, por el contrario, incrementan sensiblemente las medidas de seguridad, pues toda ruta que no esté contemplada, no podrá ser alcanzada.

  5) *Listas de control de acceso:* Son la medida primaria de acceso a una red (Se tratarán en detalle en FIREWALL).

  6) *Listas de acceso extendidas:* Amplían las funciones de las anteriores, generalmente con parámetros de nivel de transporte (Se tratarán en detalle en FIREWALL).

  7) *Archivos .Log:* Permiten generar las alarmas necesarias.

  8) *Seguridad en el acceso por consola:* Se debe prestar especial atención pues por defecto viene habilitada sin restricciones, y si se tiene acceso físico al router, se obtiene el control total del mismo. Siempre hay que tener presente que un usuario experto, si tiene acceso físico puede iniciar la secuencia de recuperación de contraseña e iniciar el router con una contraseña nueva.

- **Auditorías de tráfico ICMP:**

1) Mejor ruta: Este se trata del Tipo Nro 5 de mensaje ICMP, y su mal empleo permite triangular la ruta de una red para obligarla a pasar siempre por un router sobre el cual se obtiene la información deseada.

2) Solicitud y respuesta de eco (Ping): Se lleva a cabo por medio del protocolo ICMP con una solicitud y respuesta de eco (Tipo 0 y 8, conocido como ping). Un conocido ataque es enviarlo con una longitud mayor a lo permitido por IP (65535 Byte). Al ser recibido, el host no sabe como tratarlo y se bloquea. Cabe aclarar que hoy la masa de los sistemas ya no lo permiten. También se puede negar el servicio, por medio de una inundación de estos.

3) Destino no alcanzable: Es el tipo 3 de ICMP, lo importante pasa por los códigos en que se subdivide, pues por medio de estos, se obtiene información que es de sumo interés. Al recibir respuestas de destino no alcanzable, desde ya no es lo mismo esta situación si se trata de prohibición de acceso, de puertos negados, de Servidores que administrativamente niegan acceso a sus aplicaciones, etc.

Lo importante de esta auditoría es que es muy claro lo que se debe observar y cualquier anomalía es bastante clara para detectar. Se debe analizar que tipo de mensajes se deben permitir y cuales no.

- **Auditoría ARP:** El ataque ARP es uno de los más difíciles de detectar pues se refiere a una asociación incorrecta de direcciones MAC e IP, por lo tanto se debe analizar todas las tramas que circulan por la red y comparar permanentemente las mismas con un patrón de referencia válido. Existen programas que realizan esta tarea, como ya hemos visto uno de ellos es ARPWATCH siendo de los más conocidos.

- **Auditoría de direccionamiento IP:** Como se mencionó con anterioridad, existen dos formas de asignación de direcciones IP (antiguamente existía también una asignación automática que hoy prácticamente no se emplea más):

  1) Estático: Se implementa en cada host manualmente, y se hace presente en la red siempre con la misma dirección IP.

  2) Dinámico: Se asigna a través del empleo del protocolo DHCP dentro del rango que se desee. Se debe tener en cuenta que al producirse las cuatro tramas de DHCP, se pueden configurar varios parámetros, uno de ellos también es la máscara de subred.

El Primer planteo de direccionamiento si bien es el más costoso en tiempo y control para el administrador, **es lejos el esquema más seguro y organizado**, pues a través de este se pueden identificar distribuciones lógicas por piso, sección, departamento, provincia, etc. Si se lleva un control adecuado, ningún usuario que no posea una dirección válida verá esta red. Si se roba alguna dirección, es muy probable que el conflicto con la ya existente sea detectado.

El segundo esquema es de muy fácil implementación, pero por tratarse de un protocolo pensado para ser respondido por el primer servidor que escuche la solicitud, es muy difícil organizar un esquema que identifique lógicamente una dirección y su ubicación física dentro de la red, a su vez también por este tipo de respuesta, al hacerse presente un host en la red, el servidor le **asignará siempre una dirección IP sea cliente o intruso** (y en el último caso la mayoría de los administradores no suelen alegrarse).

Subredes: Una buena distribución de subredes y rutas para alcanzar a estas es la mejor estrategia para limitar el alcance de una intrusión.

Una estrategia mixta para redes grandes puede ser también una muy buena solución, asignando rangos de direcciones estáticas y privadas por medio de subredes a cada zona controlada por un router, y dentro de ellas permitir asignaciones dinámicas, inclusive se puede determinar una o varias zonas críticas (también controladas por un router) en las cuáles sólo se asignen direcciones estáticas, negando toda salida de esa zona a cualquier datagrama cuya dirección origen no sea una de ellas.

- **Detección de ataques "*Tear Drop*":** Este ataque se lleva a cabo por medio del uso de la fragmentación y reensamble. Se envían series de paquetes que al intentar ser reensamblados, sus identificadores no coinciden con lo que los Header de TCP/IP creen que debería ser. Esto puede causar la caída del host atacado, o la rotura de sesiones anteriormente establecidas.

4. Nivel de transporte.

En este nivel dentro de la pila TCP/IP como se mencionó con anterioridad, existirán dos posibilidades, operar en modo orientado a la conexión para lo cual se emplea TCP o sin conexión cuyo protocolo es UDP, el responsable de decidir a qué protocolo le entregará su mensaje es el que se emplee en el nivel superior, para lo cual existe el concepto de *Puerto* que es el SAP (Service Acces Point) entre el nivel de transporte y el de aplicación. En este nivel los dos elementos importantes a auditar son el establecimiento de sesiones y los puertos, los cuales se pueden determinar con las siguientes actividades:

- **Auditorías de establecimientos y cierres de sesión:**

  Ataques LAND.

  Inundación de SYN.

- **Auditorías en UDP:** Este protocolo por no ser orientado a la conexión, no implementa ninguno de los bit de TCP, por lo tanto, es sumamente difícil regular su ingreso o egreso seguro en una red. Mientras que un Proxy, solo puede regular las sesiones TCP, una de las grandes diferencias con un FIREWALL es que el último puede "Recordar" las asociaciones entre los segmentos UDP y el datagrama correspondiente, de manera tal de poder filtrar toda asociación inconsistente. Este tipo de Firewall son los que permiten el *filtrado dinámico de paquetes*. Como medida precautoria CIERRE TODOS LOS PUERTOS UDP QUE NO EMPLEE.

- **Auditoría en Puertos UDP y TCP:** Dentro del encabezado de TCP o UDP se encuentran los campos Puerto Origen y Puerto Destino, los cuales son uno de los detalles más importantes a auditar dentro de una red pues a través de ellos, se puede ingresar a un Host y operar dentro de este. Por lo tanto se deberá considerar las medidas a adoptar acorde a los puertos detallados en el capítulo del nivel de transporte referido en lo referente al análisis de puertos.

- **Auditoría de puertos de Ataque Back Oriffice 2K y Netbus:**

  Se deberá prestar especial atención a este tipo de ataques. La metodología de operación de estas herramientas implica inexorablemente la infección de la máquina destino y luego desde esta misma iniciar las conexiones hacia el exterior, por lo tanto en una red bien asegurada es muy sencillo de identificar.

- **Auditoría de Troyanos:**

  Se deberá prestar especial atención a este tipo de actividades, lo cual como se acaba de mencionar en el punto anterior, implica procesos muy similares.

- **Aplicación de la teoría:** Sobre este nivel, hemos realizado bastante actividad práctica, sobre todo en el empleo de Firewalls, por lo tanto os invitamos a que llevéis a la práctica todos los ejercicios que hemos propuesto empleando de las herramientas adecuadas, estas son la mejor defensa en este nivel.

5. Nivel de Aplicación.

- **Auditoría de servidores de correo, Web, TFP y TFP, Proxy:**

  Limitar el acceso a áreas específicas de esos servidores.

  Especificar las listas o grupos de usuarios con sus permisos correspondientes. Prestar especial atención a la cuenta Anónimos y a toda aquella que presente nombres de fácil aprovechamiento. Requerir contraseñas seguras.

  Siempre controlar los archivos. Log.!!!!!

  Deshabilitar índices de directorios.

  Deshabilitar todos los servicios de red que no sean empleados por el servidor

- **Auditorías de accesos remotos:**

  En la actualidad es común el trabajo fuera de la Empresa, para lo cual es una buena medida permitir el acceso por medio de líneas telefónicas tanto fijas, móviles como ADSL. Al implementar esta medida, el primer concepto a tener en cuenta es CENTRALIZARLA, es decir implementar un pool de módem o un Access Server (Router con puertos asincrónicos) como única puerta de ingreso. La segunda actividad es AUDITARLA PERMANENTEMENTE. Todo sistema que posibilite el ingreso telefónico, posee algún tipo de registros, estos deben ser implementados en forma detallada y su seguimiento es una de las actividades de mayor interés. Una medida importante es incrementar las medidas de autenticación y autorización sobre estos accesos.

- **Auditorías en Firewall:**

  Un Firewall como ya vimos, es un sistema de defensa ubicado entre la red que se desea asegurar y el exterior, por lo tanto todo el tráfico de entrada o salida debe pasar obligatoriamente por esta barrera de seguridad que debe ser capaz de autorizar, denegar, y tomar nota de aquello que ocurre en la red.

  Aunque hay programas que se vendan bajo la denominación de Firewall, un Firewall NO es un programa. Un Firewall consiste en un conjunto de medidas HARDWARE y SOFTWARE destinadas a asegurar una instalación de red.

  Un Firewall recordemos que **actúa en los niveles 3 (red) a 7 (aplicación) de OSI**. Sus funciones son básicamente las siguientes:

  * Llevar *contabilidad* de las transacciones realizadas en la red.
  * *Filtrar accesos* no autorizados a máquinas (mediante filtrado de paquetes, o bien observando el contenido de las unidades de protocolo de Transporte, Sesión, Presentación, y aplicación).
  * *Alertar* en caso de ataques o comportamiento extraño de los sistemas de comunicación.

Lo más importante a considerar y auditar aquí es que todos nuestros Firewalls se mantengan ACTUALIZADOS Y MONITORIZADOS, pues sin estas dos medidas la seguridad que puede ofrecer estos dispositivos se pierde día a día.

- **Bombardeos de mail:**

Se puede llenar el espacio en disco de un servidor de correo, enviándole una cantidad suficiente de mails. Se debe tener en cuenta que hasta que el usuario buscado no se conecte, los mensajes permanecerán en el servidor. Si esto se produce, no se poseerá capacidad de almacenamiento para ningún otro mensaje entrante, por lo tanto se inhibirá el servicio de correo electrónico. Se puede también generar reportes si el tráfico de correo crece repentinamente.

SOLUCION: Auditar espacio en disco rígido enviando las alarmas correspondientes una vez alcanzado el porcentaje establecido. Dedicar grandes áreas de disco al almacenamiento de mensajes, y separar esta área del resto del sistema.

- **Bombardeos de SYSLOG y SNMP:**

Semejante al de mail, pero llenará el sistema de .Log y de administración de red.

Misma solución que el caso anterior.

- **FTP (Puerto TCP 20 y 21):**

Dos de los puertos sobre los que se debe prestar atención son los de Comando (21) y de datos (20) que están reservados para ftp. El acceso a una red a través de los mismos es bastante común. La principal ventaja que ofrecen es que se puede regular con bastante precisión su flujo de establecimiento de sesiones: Siempre es el cliente el que inicia el establecimiento de la sesión y en primer orden sobre el puerto 21 (comando), una vez establecido este triple Handshake, se inicia el establecimiento de sesión sobre el puerto 20 (datos). En este segundo proceso recordemos que pueden existir dos posibilidades: *activa* y *pasiva*.

Este proceder es tratado en este párrafo pues en base a la ubicación en la que se coloque el servidor ftp, se pueden (o se deben) tomar las medidas de filtrado referidas al pasaje **entrante o saliente** de los bit SYN y ACK cuando se trate de los puertos mencionados.

- **Servidores DNS:**

Recordemos prestar especial atención a la configuración de los mismos, en especial al tráfico TCO sobre el puerto 53.

- **Servidores de correo:**

Una de las principales herramientas que emplean los spammers para ocultar sus rastros son la infección de servidores de correo que tienen activada la opción de replay (o relé), la infección de un servidor de este tipo es muy difícil de localizar, pero no lo es así en cuanto al análisis de tráfico, pues se incrementa de forma muy voluminosa, por lo tanto es una buena práctica evaluar

el tráfico entrante y saliente (no su contenido) periódicamente.

- **Servidores de archivos:**

  Englobamos dentro de esta categoría a todo servidor que permita el almacenamiento de información, sea del tipo que fuere. Todo dispositivo que permita el acceso de escritura por un usuario es potencialmente débil, no sólo por el "Tipo" archivos que pueden ser subidos, sino por su uso y en particular hoy en día es de especial interés desde el punto de vista de la seguridad todo lo referido a propiedad intelectual, pues ya ha sucedido muchas veces que empresas se vean involucradas en problemas legales por detectarse en sus ordenadores información que tiene reservados derechos de propiedad intelectual, sin que la empresa tenga la menor idea de ello.

  Sobre este tema debemos tener en cuenta dos aspectos:

  - "Tipos de archivos": para evitar cualquier extensión que pueda infectar la máquina local o a otras de la red.
  - "Derechos de Propiedad intelectual": Es decir que no se guarde en ellos información que exija pagos de cánones o que esté violando derechos.

  Una muy buena medida sobre estos servidores es poder contar con herramientas de consistencia de archivos, tipo "Tripware" y como siempre, monitorizar permanentemente a los mismos.

- **Auditoría de aplicaciones de terceros**: Todo tipo de aplicación adquirida o arrendada por la organización debe ser motivo de auditorías periódicas para verificar su normal funcionamiento o solicitar las actualizaciones necesarias cuando se encuentren puntos débiles sobre las mismas.

6. Otras medidas.

- **Auditorías a usuarios:** Realice entrevistas con clientes de la red para verificar su responsabilidad en seguridad, cuán a menudo modifica sus contraseñas, que lógica emplea para los cambios, cómo interpreta las reglas de seguridad, etc. Emplee anonimato en las mismas, y no tome ninguna medida con usuarios. De ser posible, entreviste empleados que se alejen definitivamente de la Organización o que ya lo hayan hecho.

- **Resguardo de información:** Jamás es suficiente la totalidad de medidas tomadas para resguardar la información. No se detallará aquí como implementarlas pues ya lo hemos hecho en la teoría, pero lo que sí se deseamos recordar aquí es el control de integridad de todo lo resguardado, pues de nada sirve almacenar datos corruptos. Estas medidas se pueden implementar por comparación o simulando la recuperación de información (recordar PCN). *Se aconseja llevar a cabo esta auditoría muy frecuentemente*, **pues la Información es el bien más preciado de cualquier Organización**.

- **Seguridad en Internet, Extranet e Intranet:** Dentro del Plan de Seguridad, es conveniente poseer un apartado que contemple un resumen comparativo de las medidas de seguridad implementadas en estos tres límites de la red. La visión global y la comparación de estas tres interfaces, muchas veces indica la insuficiencia, redundancia o ausencia de las medidas adoptadas.

- **Listas de personal:** Se deberá coordinar y controlar con recursos humanos el pronto envío de las listas del personal que deja la Empresa o que cambia de ubicación o puesto, para *auditar el estado de sus cuentas y a su vez para revertir medidas, puertas traseras, contraseñas*, etc., en el caso de haber tenido acceso a esta.

- **Criptografía:** Sin entrar en detalle de las medidas a adoptar; en cuanto a la auditoría, se debe llevar un registro detallado de claves, canales, sistemas, dispositivos y personal que emplea esta tecnología y revisarlo **PERMANENTEMENTE!!!!.** Como experiencia se pone de manifiesto que muchos sistemas han sufrido daños realmente serios por confiar en su criptografía sin auditarla, sin tener en cuenta que una vez violado un sistema criptográfico, se tiene acceso a la información de mayor impacto de toda la Organización. Se hace especial hincapié en esta reflexión pues es el peor estrago que se pone de manifiesto al producirse una falla en el sistema que almacena o transporta lo más crítico a asegurar.

- **Inventarios:** La auditoría de inventarios es una de las tareas más tediosas de un Administrador, pues en virtud del acelerado avance tecnológico, este registro se modifica día a día. Lo único que se desea expresar sobre este punto es:

  SI NO SE SABE QUE SE TIENE, NO SE SABE QUE ASEGURAR.

7. Optimización de la red.

Para iniciar el estudio de optimización, es necesario aclarar primero que esta tarea se puede realizar de dos formas, la **primera** es incrementar el ancho de banda disponible por usuario (aumentando la velocidad de la red o reduciendo la cantidad de tráfico en la red generado por determinados servicios); la **segunda** es proveer a los usuarios de mejor tiempo de respuesta, lo cual se obtiene analizando en detalle el tráfico de la red y obteniendo la mejor performance de cada uno de los tipos de tramas o paquetes que circulan por ella. Una optimización adecuada será una solución de compromiso entre el tiempo de respuesta deseado y el ancho de banda disponible.

Otro factor a tener en cuenta es la predicción sobre el crecimiento de una red. Una red inexorablemente crece, a su vez diferentes servicios son adicionados o modificados lo cual provoca permanentes fluctuaciones en el rendimiento de la misma. Una de las principales tareas de un Administrador es tratar de predecir este crecimiento, analizando los efectos que producirá en el tráfico de información. Este trabajo cotidiano permite mantener estable la red durante estos cambios.

Hoy en día parece que no existe ancho de banda que de abasto con los requerimientos de los clientes, toda ampliación es poca, y desde el punto de vista de la seguridad, es un hecho concreto que atenta contra la "Disponibilidad" de los servicios que se ofrecen, por lo tanto hoy más que nunca es necesario incrementar todos los esfuerzos que estén a nuestro alcance para eliminar todo el tráfico de la red que es innecesario para el trabajo de la empresa. Para ello presentamos a continuación un conjunto de medidas que pueden ser consideradas.

7.1. Optimización del tráfico DHCP:

Si bien la implementación de DHCP no provoca un incremento considerable de tráfico, la adquisición de una dirección dinámica IP por parte de un cliente ocupa casi 300 milisegundos en una red vacía. Como ya se vio, una conversación DHCP ocurre al inicializar un cliente DHCP, se renegocia automáticamente en la mitad del tiempo de duración, al mover un cliente a otra subred, al cambiar el adaptador de red, al reiniciar un cliente y al ejecutar el comando IPCONFIG.

Para reducir la cantidad de tráfico es posible ajustar el tiempo de duración de las asignaciones IP, en Windows NT se ejecuta desde el Administrador DHCP, tiempo de duración. Por defecto es de tres días, y se puede incrementar siempre y cuando existan direcciones IP libres, si realmente sobran, se puede configurar 30 o 60 días, reduciendo considerablemente el diálogo de clientes DHCP. La alternativa de máxima es eliminar DHCP y trabajar con direcciones IP estáticas, para lo cual se debe llevar un registro estricto de las asignaciones para evitar direcciones duplicadas.

7.2. Optimización de tráfico WINS cliente – servidor:

Cada cliente NetBIOS que es registrado en un servidor WINS, implica: 2 tramas de registración al iniciar el cliente, 2 tramas de renegociación (Por defecto cada 3 días), 2 tramas para resolver el nombre al intentar acceder desde otra computadora y dos tramas de liberación

de nombre al detenerse un cliente. Se debe tener en cuenta que todo el tráfico WINS es dirigido, por lo cual el impacto que produce no es representativo en una red

7.2.1. Deshabilitar servicios innecesarios: La primera medida para minimizar el tráfico WINS es deshabilitar servicios de red innecesarios pues cada servicio que soporta NetBIOS se debe registrar en WINS.

7.2.2. Incrementar caché local: Después que el nombre NetBIOS fue resuelto, este es guardado en una caché interna de NetBIOS. Esta caché es chequeada antes de solicitar al servidor WINS, si el nombre se encuentra en la tabla, no se produce la consulta al servidor. Por defecto, esta entrada permanece en caché durante 10 minutos. Por lo tanto si se incrementa este valor, se reducen las cantidades de consultas al servidor WINS, y por lo tanto el tráfico en la red (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet \Services\NetBT\Parameters\CacheTimeout al más alto valor).

7.2.3. Implementar LMHOSTS: Un método estático de implementar resolución de nombres NetBIOS es la implementación de tablas en cada uno de los clientes que consten de una asociación de Nombres y direcciones IP, este método se logra por medio de una archivo de texto llamado LMHOST.EXE (existente también en Linux como lmhosts) que se copia en cada uno de los clientes. Como en esta tabla contiene la totalidad de los nombres de la red, no necesita consultar ningún servidor, anulando este tipo de tráfico. La contra que posee este tipo de tablas estáticas es que toda actualización deberá ser replicada en todos los clientes en forma “casi manual” por el administrador, causa por la cual se está dejando de lado.

7.2.4. Incrementar el tiempo de renegociación y permanencia: Desde el administrador de WINS en Windows NT se puede modificar el intervalo de renegociación, que por defecto es 3 días y el de permanencia que es 6 días. Si bien esto debe ser realizado con extremo cuidado pues las tablas pueden modificarse en el tipo y cantidad de servicios reales que están en vigencia, se debe tener en cuenta que por ejemplo un host que registró 6 servicios al iniciarse, deberá renegociar los seis cada 72 horas.

7.3. Optimización de tráfico de sesión de archivos.

La masa del tráfico de transferencia archivos sucede una vez que la sesión ha sido establecida. Sin embargo, el tráfico de establecimiento de sesión es repetido toda vez que un ordenador necesita establecer una sesión con otra. Existen dos métodos para limitar esta cantidad de tráfico:

7.3.1. Remover protocolos en exceso: Este es el mejor método, pues toda solicitud de conexión es enviada sobre todos los protocolos simultáneamente ocasionando tráfico innecesario.

7.3.2. Colocar servidor y clientes en mismos segmentos de subred.

Una medida de evitar compulsas por el uso del cable, es por medio de la segmentación a nivel de enlace, lo cual se puede realizar por medio de Switch o Bridge. Si una red se encuentra segmentada en subredes a nivel de enlace (Dirección

NIC o MAC) se debe tratar de agrupar los clientes que mayor actividad tienen sobre un servidor de aplicaciones en los mismos segmentos que el servidor. Con esta medida se obtiene que el tráfico a nivel 2 entre ese servidor y sus clientes, no salga de ese "Dominio de colisión", eliminando el tráfico de la sesión de archivos de ese segmento en todos los restantes.

7.4. Optimización del tráfico de validación.

Por defecto, los controladores de dominio en Windows, tienen configurado **maximizar el rendimiento para archivos compartidos** (Panel de control → red → propiedades → servidor). Esta configuración es óptima para servidores de archivos e impresión, pero no para controladores de dominio que necesitan validar usuarios. Lo óptimo es configurarlo para **Maximizar el rendimiento para aplicaciones de red**, esta opción puede triplicar el número de validaciones simultáneas desde 6/7 por segundo a 20.

7.5. Optimización del tráfico Browser de cliente:

Para permitir la correcta ubicación de los recursos dentro de una red, es que se implementa el servicio de Browser, el cual genera un tráfico considerable, la optimización del mismo, se puede llevar a cabo en tres aspectos:

7.5.1. Deshabilitar servicios:

La primera medida es analizar qué computadoras sobre la red, realmente prestan servicios a la misma. Luego determinar con qué periodicidad prestan mismos; si el resultado es mínimo, una buena medida es deshabilitarlos; si el resultado es nulo, no tiene sentido dejarlos habilitados. Esta medida reducirá los anuncios y reducirá el tamaño de las listas de Browser que serán mantenidas y transferidas en la red.

7.5.2: Controlar el número de browser:

El número de backup browser en una red es automáticamente determinado por el software de browser. Cuando otro backup browser es necesitado, un browser potencial es notificado por el master browser que debería constituirse como backup browser. Los potenciales browser son configurados como se detalla a continuación:

- Si un ordenador Windows no debiera ser un potencial browser, se debe colocar HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Parameters\MaintainServerList Value a No.
- Para Windows XP se debe colocar en: Panel de control → red → Compartir archivos e impresoras → examinador principal a "desactivado".

7.5.3: El sistema de browser es dependiente del protocolo, por lo tanto, si se están usando tres protocolos, todos los anuncios y elecciones serán repetidas tres veces, una por cada protocolo. Por lo tanto reduciendo el número de protocolos se optimizará sensiblemente el tráfico de browser.

7.6. Optimización de tráfico DNS:

La resolución de direcciones por medio de un servidor DNS, implica solamente dos tramas dirigidas, por lo tanto el esfuerzo es conveniente orientarlo hacia la reducción del tráfico recursivo entre servidores DNS que puede resultar si ante una consulta cliente, el servidor no puede resolverla. Esto se puede optimizar:

7.6.1. No configurando recursión. Esto acotará la lista de nombres disponibles.

7.6.2. Asegurando que el servidor DNS que contará con la mayoría de los nombres para un cliente en particular, sea su servidor DNS primario.

7.6.3. Incrementando el Tiempo de Vida de cada entrada en los servidores DNS. Cuando uno de estos servidores, solicita la resolución de un nombre a otro servidor DNS, este le enviará la dirección requerida, la cual será almacenada en caché por un tiempo determinado. Este tiempo por defecto es 60 minutos, para poder ampliarlo, desde Herramientas Administrativas →Administrador DNS → DNS menú → Propiedades → Propiedades de la zona → registros SOA →Mínimo TTL, colocar el valor deseado.

7.7. Optimización de tráfico Browsing de Intranet:

La más efectiva optimización del tráfico de intranet browsing es durante la creación de las páginas Web, pues la mayoría del tráfico Web es causado por el tamaño de los archivos que son copiados a través de la red. Se estima que el 48 % del tráfico de una red es producido por este tema, por lo tanto toda medida que se pueda tomar, es significativa, las cuales se detallan a continuación:

7.7.1. Páginas Web de pequeño tamaño: Como regla general en HTML es aconsejable limitar la longitud de las páginas, manteniendo claros indicativos para ir profundizando en otras páginas, para llegar al detalle de la información requerida, pero en sucesivas páginas de reducido tamaño.

7.7.2. Limitar el tamaño de los gráficos o aplicaciones. AVI: Como el punto anterior, cada uno de estos archivos, deberá ser bajado por un cliente, provocando un enorme volumen de tráfico.

7.7.3. Incrementar la memoria caché en disco de los clientes: Cuando un cliente navega en Intranet, las páginas serán bajadas sucesivamente al mismo, y ubicadas en un directorio llamado caché, a medida que el mismo se va llenando, nuevas páginas van reemplazando generalmente las de mayor tiempo de permanencia o las menos refrescadas (o consultadas nuevamente). Si se repite una consulta hacia una página ya sobre escrita en este espacio de disco, nuevamente deberá buscarla a través de la red generando tráfico, el cual se podría haber minimizado si el caché de disco hubiera sido mayor.

7.7.4. Considerar seguridad: Se debe tener en cuenta la estrategia de seguridad de las páginas Web, pues la autenticación de clientes Web, genera tráfico adicional por cada sesión que es establecida. Permitiendo las conexiones anónimas, este tráfico no existe.

7.8. Optimización del tráfico de sincronización de cuentas:

El tráfico de sincronización de cuentas de usuarios, puede ser regulado a través de parámetros (En Windows : HKKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ Netlogon\ Parameters). En un entorno Windows, el servicio de NetLogon sirve para sincronizar las bases de datos entre el PDC (Primary Domain Controller) y el o los BDC (Backup Domain Controller); las responsabilidades del servicio de Netlogon son las siguientes:

- Sincronizar las bases de datos de usuarios entre BDC y PDC.
- Proveer validación de cuentas de usuarios
- Proveer soporte a las relaciones de confianza entre dominios.

7.8.1. El parámetro mas frecuentemente modificado en el servicio de Netlogon, es el Gobierno de replicación, este parámetro controla el porcentaje de ancho de banda que el servicio Netlogon puede emplear mientras ejecuta la sincronización de cuentas de usuario. El valor por defecto es 100 %. Este parámetro puede causar serios daños en entornos WAN, donde varios usuarios emplean el mismo medio, y este es apropiado por un controlador de dominio, dejando fuera todo otro host. Se puede considerar, reducir este valor a 50, y es aconsejable no minimizarlo por debajo de 25 pues puede producirse el no completamiento de la sincronización.

7.8.2. Un parámetro que también influye es el pulso, el cual controla la periodicidad con que un PDC inspecciona su base de datos y envía mensajes de sincronización a los BDC que necesitan actualizar. El valor por defecto es 5 minutos y puede ser incrementado hasta 48 horas. Se debe tener cuidado con estos máximos, pues en este caso puede ocasionar que por pérdida de sincronismo, un controlador deba sincronizar la totalidad de la base de datos (no solo los cambios), provocando un tráfico mayor que si el valor fuera 5 minutos.

7.8.3. Pulso máximo: Este parámetro controla cuán a menudo el PDC enviará un mensaje de pulso a cada BDC, aunque su base de datos esté actualizada. El valor por defecto es cada 2 horas, y puede ser incrementado hasta 48 horas.

7.8.4. Tamaño de cambios de logon: Este parámetro controla el número de cambios en la base de datos de usuarios que puede soportar un servidor antes de generar una sincronización completa. En un entorno con gran cantidad de usuarios que cambian sus contraseñas con gran periodicidad, puede ocurrir que se llegue al límite de este valor, provocando que la entradas en este archivo, sean sobre escritas; esto generará una sincronización completa, ocasionando un exceso de tráfico. El valor por defecto es 64 KB que equivalen a unos 2000 cambios (de 32 Byte cada uno).

7.8.5. Concurrencia de pulso: Controla el número de BDC a los cuales el PDC les enviará simultáneamente un anuncio cuando un evento de sincronización ocurra. El valor por defecto es 10. Incrementar este valor, provocará un gran uso del ancho de banda; por otro lado, el disminuirlo, puede equivaler a largos períodos de sincronización.

7.9. Optimización del tráfico de relaciones de confianza:

Aunque las relaciones de confianza no producen un alto porcentaje de tráfico, existen dos áreas que pueden ayudar a reducir el tráfico:

7.9.1. Reducir el número de relaciones de confianza: Verificar que cada uno de los sentidos de las relaciones de confianza sean apropiados, de no serlos rómpalas. Como ya se expreso, el tráfico es mínimo, una vez que la relación ha sido establecida, la masa del tráfico es producido por importación y verificación de cuentas en las que se confía y pasajes de autenticación.

7.9.2. Usar cuentas de grupos:

* En el dominio confiado, adicionar los usuarios apropiados a un grupo global.

* En el dominio que confía, adicionar el grupo global confiado a un grupo local o a un recurso local.

Si se tomara como ejemplo, la verificación de SID de un grupo global implica la transferencia de 552 Byte, mientras que las mismas verificaciones de SID de 2 usuarios individuales ocasionan 636 Byte. Por lo tanto, si la diferencia ya se hace notable con 2 usuarios, estas verificaciones periódicas realmente hacen notables durante un período de tiempo en la línea de transmisión.

7.10. Optimización de tráfico browser entre servidores:

La masa del tráfico de browsing es generado en forma automática por los servidores, clasificados en Master y Backup browser. Existen tres métodos generales para reducir este tráfico:

7.10.1. Reducir los protocolos de red: El proceso de browsing actúa por separado en cada protocolo que se encuentre presente. Por lo tanto, si TCP/IP, NetBEUI e IPX/SPX se encuentran presentes, cada elección, anuncio individual y de grupo se presentará por cada protocolo independientemente, produciendo el triple del tráfico necesario.

7.10.2. Reducir entradas de browser: Deshabilitar los servicios y recursos compartidos en computadoras que no lo usan, esto reducirá el tamaño de las listas de browser en los servidores, y por lo tanto el pasaje de las mismas.

7.10.3. Optimización de parámetros: En servidores Windows existen dos parámetros que se pueden configurar para el control de la cantidad de tráfico generado por el browser que se encuentran en: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Browser\Parameters:

* Periodicidad del master: Determina cada cuánto tiempo un master browser contacta al master browser de dominio. Por defecto este valor es 720 segundos (12 minutos), con un mínimo de 300 segundos y un máximo de 4.294.967 segundos (casi 50 días). Este parámetro va dirigido al tráfico WAN, teniendo en cuenta que en cada subred, existirá un master browser, pero uno solo será el master browser del dominio.

* Periodicidad del backup: Especifica con que frecuencia, un backup browser contacta al master browser. El valor por defecto es 720 segundos, y se puede

llevar hasta 1800 segundos, reduciendo significativamente el tráfico. Este parámetro no afecta el tráfico WAN, pues los backup siempre se comunican con el master browser local, nunca con uno remoto.

7.11. Optimización de tráfico de replicación WINS:

Las replicaciones WINS pueden ser configuradas como PUSH o PULL. En el caso de Push, los servidores envían anuncios al resto de los servidores WINS que estén configurados con el número de cambios sucedidos en su base de datos. En el caso de pull, los servidores solicitan actualizaciones a intervalos de tiempo regulares o al ser notificados de cambios. Una relación en la cual cada servidor replica su base de datos con todos los otros, implica configurar push y pull en cada servidor WINS.

7.11.1. Configuración de push: La configuración de WINS al iniciar el servicio, genera una notificación, después que cierto número de actualizaciones han sido acumuladas. Por defecto, este valor es 20, y no existe un tiempo para iniciar las notificaciones y replicaciones push. Si se incrementa la cantidad de cambios acumulados, implica una menor frecuencia de actualizaciones, y lotes de mayor cantidad de registros en cada actualización. Una buena medida es operar con un porcentaje de la cantidad de registros en la base de datos WINS, con valores que oscilen en el 25 %.

7.11.2. Configuración de pull: Al configurar los servidores WINS como replicación pull, se puede configurar la periodicidad de solicitud de cambios. En entornos LAN, un valor adecuado, está en el orden de los 30 minutos para un servidor WINS local; y un valor aproximado de 6 horas, para servidores que operen a través de vínculos de 64 Kbps.

7.12. Optimización del tráfico de replicación de directorios:

La capacidad del servicio de replicación de directorios, permite a los servidores Windows duplicar árboles de directorios sobre múltiples ordenadores. Como puede apreciarse, este proceso puede llegar a generar un enorme volumen de tráfico. Por defecto, el servidor de exportación chequea cada 5 minutos por datos a ser replicados, pudiéndose modificar desde los registros de Windows. Un ejemplo de un solo directorio que contenga 16 archivos y 426 KByte de datos, implica 1.425 tramas y aproximadamente 42 segundos en replicarse.

7.12.1. Estructura de directorios: El mejor modo de reducir el tráfico de replicación de directorios es la creación de muchos grupos de carpetas de gran cantidad de pequeños archivos, en lugar de pocas carpetas de grandes archivos. El servicio de replicación de directorios chequea y luego copia archivos que han sido modificados en la estructura de directorios. Si un cambio es realizado en un determinado archivo, el mismo será replicado en lo servidores que así estén configurados, por lo tanto, menor será el tráfico cuanto más pequeño sea el volumen de información a transferir.

7.12.2. Usando el Administrador de servidores para controlar la replicación:

Una buena medida es poder determinar la replicación de directorios en horarios en los cuales el uso de la red sea mínimo, esto se puede realizar por medio del administrador de servidores en Windows → Propiedades → Replicación → Replicación de directorios → Exportación de directorios → Administrar.

7.12.3. Parámetros de registro: El control de las replicaciones, puede ser encontrado también en HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services \Replicatior\Parameters, cuyos dos valores más comunes a modificar son:

* El parámetro intervalo que originará la frecuencia con la que el servidor de exportación chequeará por actualizaciones en la estructura de directorios y envía notificaciones a los servidores de importación. El valor por defecto es 5 minutos y puede ser llevado hasta 60 minutos o más para reducir la frecuencia de replicación.

* El parámetro pulso actúa como un contador de control de cuán a menudo el servidor de importación contacta al de exportación. Este valor se multiplica <pulso> x <intervalo>, y determina el tiempo que esperará la computadora para contactarse con el servidor de exportación para una actualización, pues superado ese tiempo interpreta que hubo una falla. El valor por defecto es 2 minutos, lo que significa que si el intervalo fuera de 5 minutos, y la computadora de importación no escuchó al servidor de exportación durante 10 minutos, esta iniciará la comunicación con el segundo. El incremento del pulso, quizás esta no sea una medida de gran impacto, pero colabora con la optimización del tráfico.

ANEXO 1 (Aspectos a considerar para la certificación de una red).

En este anexo se describen las definiciones y aspectos básicos a considerar a la hora de contratar o realizar la certificación de una red LAN.

1. Conceptos previos relacionados a cableado.

1.1. DISTORSION

Es un fenómeno CAUSAL, no aleatorio, producido por la propia constitución de los circuitos eléctricos, que se establece a través del medio de comunicaciones a causa de las características *reactivas* de éstos. En términos prácticos, es una *deformación de la señal* respecto de la forma inicial que tenía.

Podemos definirla como la deformación que sufre la señal eléctrica a causa de elementos del propio circuito (resistencias, condensadores o bobinas) o externos a él.

Podemos clasificarla según las causas que la originan como:

- Distorsión por atenuación (la causa la impedancia)
- Distorsión por retardo de grupo (la velocidad en función de la frecuencia)
- Distorsión por efectos meteorológicos (por lluvia, polvo, etc.)

1.2. RUIDO

Es un fenómeno CASUAL, que ocurre en forma aleatoria, debido a fenómenos tales como tormentas, emisiones radiales, etc.

El ruido, son señales eléctricas no deseadas que alteran la transparencia de las señales transmitidas en el cableado de la LAN.

Se define como toda señal o interferencia aleatoria no deseada que ya sea en forma **endógena** o **exógena** se introduce en el canal de comunicaciones, sumándose a la señal útil y produciendo también una deformación de la señal, aunque con otro origen.

Las señales distorsionadas por el ruido en la LAN pueden provocar errores en la comunicación de datos.

Su principal característica es su aditividad, pues suma su intensidad a la de la señal.

Podemos definirlo también como todo fenómeno que, adicionado a la señal que se está transportando desde la fuente, afecta la calidad de la información recibida en el colector.

El ruido es generado por cualquier dispositivo o generador de tensión variable. Una variación de tensión, genera un campo electromagnético variable, el cual transmite ruido a los dispositivos vecinos en el mismo sentido que un radiotransmisor transmite a su terminal. Por ejemplo, las luces fluorescentes que usan 50 o 60 HZ AC, continuamente irradian una señal de 50 o 60 Hz que puede ser recibida por dispositivos vecinos como un ruido.

Los tipos de ruido se clasifican en:

- Gaussiano o blanco (movimiento aleatorio de electrones)
- Impulsivo (Relay – corta duración)
- Intermodulación (muchas señales senoidales en dispositivos no lineales)
- Diafonía o crosstalk (inducción mutua)
- Ruido en línea o simple (por los 50 Hz)

Los cables de una LAN actúan como antenas que pueden captar el ruido de los tubos fluorescentes, motores eléctricos, fuentes de energía eléctrica, fotocopiadoras, heladeras, ascensores, y otros dispositivos electrónicos. El cable coaxial es mucho menos susceptible al ruido que el cable de pares trenzados por la malla que lo recubre.

2. Parámetros de certificación en cableado.

Los tests de certificación que corrientemente se aplican a un par de cables cruzados, son los que miden los siguientes parámetros:

- Resistencia
- Longitud
- Velocidad de propagación
- Impedancia
- NEXT (Near-end Crosstalk)
- Atenuación
- ACR (Radio de atenuación por cruzamiento)
- TDR

2.1. Resistencia

Para el caso del UTP 5, la resistencia de cualquier conductor no podrá superar los 9,8 ohms cada 100 mts medidos a 20 ºC

Para el caso del STP a 25 ºC, no podrá exceder los 5,71 ohms en 100 mts.

Resistencia Desbalanceada: La resistencia entre dos conductores cualesquiera, no podrá exceder en más de un 5% para el caso del UTP y el 4% para el caso del STP.

2.2. Velocidad de propagación

La velocidad de propagación de una señal, es la velocidad relativa de la misma al atravesar el cable y la velocidad de la luz.

En el vacío, las señales eléctricas viajan a la velocidad de la luz. En el cable, lo hacen a menor velocidad, a un 60% a 80 % de la misma.

La fórmula que permite determinar la velocidad de propagación (NVP) es:

$$NVP = \frac{\text{Veloc de los pulsos a través del cable x 100 \%}}{\text{Veloc de la luz}}$$

Los valores de la NVP afectan los límites de longitud del cable para los sistemas Ethernet, porque la operación de las ethernet depende de la habilidad del sistema para detectar colisiones en un determinado espacio de tiempo. Si la NVP de un cable es muy lenta o el cable es muy largo, las señales se demoran y el sistema no puede detectar colisiones lo suficientemente rápido como para prevenir serios problemas en la red.

La medición de la longitud depende directamente del valor de la velocidad de propagación correspondiente al cable seleccionado. Para medir la longitud, primero se mide el tiempo que requiere un pulso para viajar toda el largo del cable y en función de tal tiempo, obtiene la medida de la longitud del cable.

2.3. Impedancia

La impedancia, es un tipo de resistencia que se presenta al flujo de corriente alterna. Las características de impedancia de un cable, es una propiedad compleja resultante de la combinación de efectos inductivos, capacitivos y resistivos del cable.

Estos valores están determinados por los valores de parámetros físicos tales como la medida de los conductores, distancia entre los mismos, y las propiedades del material dieléctrico.
Las propiedades de operación de la red, dependen de una característica constante de impedancia a través de los cables y conectores.

Los cambios en las características de la impedancia, llamados discontinuidades de impedancia o anomalías de impedancia, causan señales reflejadas, que pueden distorsionar las señales transmitidas por la LAN y consecuentemente producir fallas en la red.

La impedancia de un canal de comunicaciones se puede expresar a través de un número complejo Z, tal que resultará la expresión.

$$Z = R + j\,(XL - XC)$$

Donde R es la resistencia óhmica, la cual es directamente proporcional a un coeficiente propio de cada material, su resistividad, a la longitud e inversamente proporcional a su sección.

En rigor de verdad, en la impedancia actúan la resistencia, las reactancias capacitivas e inductivas, y su valor es función de su frecuencia.

La impedancia característica, de un cable UTP será de 100 ohms +- 15% medida desde 1Mhz hasta la máxima frecuencia de trabajo

Minimización de las discontinuidades de impedancia

Las características de impedancia son comúnmente modificadas por las conexiones del cable o por las terminaciones. La red puede trabajar con pequeñas discontinuidades porque las señales reflejadas resultantes son también pequeñas y se atenúan en el cable. Discontinuidades mayores pueden interferir la transmisión de datos. Tales discontinuidades son causadas por malos contactos, terminaciones de cableado deficientes, pérdidas en cables y conectores y por mal entrecruzamiento de los cables en el cable de pares trenzados.

Es posible minimizar tales problemas teniendo en cuenta las siguientes precauciones durante la instalación:

- Nunca se deben mezclar o combinar cables con diferente impedancia (A menos que se empleen circuitos de una impedancia especial)
- Siempre termine un cable coaxil con una resistencia de impedancia igual a las características del cable.
- Cuando desenlace los pares de cable trenzado para instalar conectores o hacer conexiones a distintos equipos, realice tales desenlaces lo más cortos posible (Para 4 Mbps, 1"; para 10 Mbps, ½" y para 100 Mbps, ¼")
- No realice empalmes en el cable.
- Manipulee el cable cuidadosamente durante la instalación. No lo pise ni lo atraviese para su fijación

2.4. NEXT (Near-End Crosstalk)

NEXT, es la distorsión de la señal durante la transmisión, causada por el acoplamiento con la señal transmitida. En cuanto a las pérdidas de esta naturaleza, se mide aplicando una señal de entrada balanceada a la entrada y se mide la modulación cruzada en la salida de dicho par. Para ello, se considera la diferencia en amplitud entre una señal de test y la señal de crosstalk, respecto de la misma terminación del cable. Esta diferencia es llamada NEXT y se expresa en dB.

En otras palabras podemos decir que NEXT, es una señal no deseada, producida por la transmisión de una señal por un par de cables a otro par vecino. Al igual que el ruido, puede causar problemas en las comunicaciones de la red.

De todas las características del cableado de una LAN, el cruzamiento, es el que más influye en la operación de la red.-

El NEXT es inversamente proporcional a la frecuencia, en consecuencia decrece a medida que la frecuencia se incrementa.

El peor valor de NEXT para una distancia mayor o igual a 100 mts para UTP 5 es de 64 dB a una frecuencia de 0,772 Mhz

Todas las señales transmitidas a través del cable son afectadas por la atenuación. Debido a ella, el cruzamiento que ocurre en el extremo del cable contribuye menos al NEXT que el producido en el comienzo del mismo. Para verificar las propiedades de performance del mismo, se debe medir el NEXT de ambos extremos.

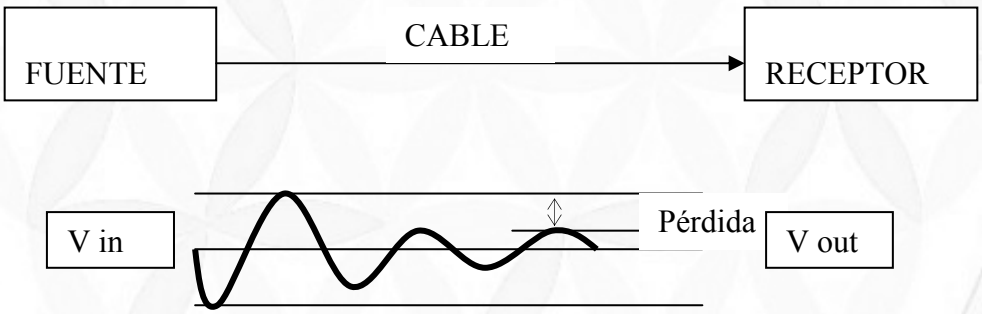
2.5. Atenuación

Es una pérdida de energía eléctrica en función de la resistencia del cable, que se aprecia en la disminución de la amplitud de la señal en función de la distancia recorrida por la misma.

Esta pérdida se expresa en decibeles (dB) y, a menor atenuación, corresponde mayor performance de calidad del medio. Así por ejemplo, cuando comparamos la performance de 2 cables a una frecuencia en particular, un cable con una atenuación de 10 dB es mejor que uno con una atenuación de 20 dB.

La atenuación de un cable es determinada por su construcción, su longitud y las frecuencias de las señales que por él se transmiten.

Se caracteriza por la disminución de la intensidad de la señal útil a medida que la misma va recorriendo el medio de comunicaciones sobre el que es transportada la señal. Aumenta en forma proporcional a la distancia y va produciendo como efecto una reducción en la amplitud de la señal.



Atenuación

Como mencionamos anteriormente, los cables tienen una atenuación que depende de la frecuencia de operación. Así por ejemplo el cable coaxial fino RG 58, que era utilizado en las redes Ethernet, tienen una atenuación de aproximadamente 4,8 dB en 100 mts a una frecuencia de 10 Mhz, la fibra óptica multipunto 62,5/125 tiene una atenuación de 1,6 dB en 1000 mts, 100 mts de cable UTP 5, a la misma frecuencia de 10 Mhz, tiene una atenuación de 6,5 dB.

2.6. ACR (Attenuation to Crosstalk Ratio)

El ACR, es la diferencia entre el NEXT (en dB) y la atenuación (en dB). El valor de ACR indica cómo es la amplitud de las señales recibidas por un transmisor lejano en comparación con la amplitud de cruzamiento producido por transmisiones cercanas.

Un valor elevado de ACR indica que las señales recibidas son mucho mayores que el cruzamiento. En términos de NEXT y valores de atenuación, un ACR elevado corresponde a un NEXT elevado y a baja atenuación.

2.7. TDR (Time Domain Reflectometry)

Las reflexiones, son causadas por las discontinuidades de impedancia que vimos anteriormente así como por falsos contactos, deficiencias de diseño del cable o impurezas del mismo.

Una falla del cable que tenga una impedancia superior a la impedancia característica del cable, refleja una señal que tiene la misma polaridad que la original. Si la falla no está completamente abierta, la amplitud de la señal reflejada será menor que la correspondiente a la original.
Si la impedancia de la falla es menor que la impedancia característica del cable, pero no es despreciable, la señal reflejada tendrá una polaridad contraria y menor amplitud que la señal original.

Debido a que las señales reflejadas pueden distorsionar las comunicaciones, los terminales no usados de los cables, deben terminar, previniendo tales reflexiones (en particular esta es una referencia para el cableado con coaxial), utilizando un dispositivo resistivo de impedancia igual a la impedancia característica del cable.

2.8. TDX (Time Domain Crosstalk)

Es un test que permite, si la medición que estamos realizando acusa una falla NEXT, localizar el punto singular donde se ubica el problema y la magnitud del NEXT

2.9. Reflectometría

La reflectometría es el método que se utiliza para detectar puntos singulares, aberturas, fallas en el cable, conectores, etc., aprovechando el crecimiento o decrecimiento de la impedancia.

Al acusar una falla de esta naturaleza, el método aprovecha la reflexión que se produce como consecuencia de la falla. Esta reflexión es de igual amplitud y polaridad que la de la señal original.

Midiendo el tiempo empleado por el pulso de la señal que se refleja, el test de reflectometría permite determinar la falla.

2.10. RL (Return Loss)

RL es la diferencia entre la potencia de la señal transmitida y la potencia de la señal reflejada, causadas por las variaciones de impedancia del cable.

Un valor elevado de RL significa que la impedancia es muy poco variable, lo cual implica una gran diferencia entre las potencias de las señales transmitidas y reflejadas.

Cables con un alto valor de RL, son más eficientes en la transmisión de señales en una LAN, porque significa que las señales reflejadas son muy pocas.

3. Ejemplo de mediciones en cableado con instrumental Fluke

SOLEJEMPLO Sumario de
Pruebas: FALLO
LUGAR: DIR DE COMPRAS ID. Cable: PISO 5
AYTIA

OPERADOR Juan García **Fecha/Hora: 11/12/2008 11:43:26**

NVP: 69.0% UMBRAL DE ANOMALIA DE FALLO: 15% Estánd. Pruebas: TIA Cat 5 Channel + ACR
FLUKE DSP-100 N/S: 7120385 Tipo de Cable: UTP 100 Ohm Cat 5
PASO LIBRE: -1.8 dB Versión de Estándares: 5.4
Versión de Software: 5.4

Mapa de Cableado PASA Result. TERM. RJ45: 1 2 3 4 5 6 7 8 B
| | | | | | | |
TERM. RJ45: 1 2 3 4 5 6 7 8

Par	1,2	3,6	4,5	7,8
Impedancia (ohmios), Límite 80-120	104	104	103	105
Longitud (m), Límite 100.0	8.5	8.3	8.1	8.3
Tiempo de Prop. (ns)	41	40	39	40
Diferencia Retardo (ns), Límite 50	2	1	0	1
Resistencia (ohmios)	1.7	1.7	1.7	1.6
Atenuación (dB)	2.1	1.8	2.0	2.2
Límite (dB)	24.1	24.4	24.5	24.4
Margen (dB)	22.0	22.6	22.5	22.2
Frecuencia (MHz)	97.4	99.6	100.0	99.4

Pares	1,2-3,6	1,2-4,5	1,2-7,8	3,6-4,5	3,6-7,8	4,5-7,8
NEXT (dB)	43.8	45.1	48.0	26.5*F	48.8	46.9
Límite (dB)	28.1	32.5	39.4	27.1	34.1	32.8
Margen (dB)	15.7	12.6	8.6	-.6	14.7	14.1
Frecuencia (MHz)	88.3	49.1	19.0	100.0	39.5	46.9

NEXT del Remoto (dB)	45.1	40.4	47.0	26.8 F	48.6	43.7
Límite (dB)	28.2	30.2	38.6	28.6	33.9	28.6
Margen (dB)	16.9	10.2	8.4	-1.8	14.7	15.1
Frecuencia (MHz)	86.8	67.0	21.4	82.8	40.5	82.6
ACR (dB)	61.2	44.5	68.2	58.2	67.2	50.3
Límite (dB)	32.7	17.2	53.5	52.2	39.9	25.0
Margen (dB)	28.5	27.3	14.7	6.0	27.3	25.3
Frecuencia (MHz)	14.3	45.4	1.7	2.0	7.6	27.0
ACR del Remoto (dB)	81.8	55.5	67.5	58.9	68.9	64.4
Límite (dB)	56.3	30.6	52.2	53.5	42.1	40.0
Margen (dB)	25.5	24.9	15.3	5.4	26.8	24.4
Frecuencia (MHz)	1.2	17.1	2.0	1.7	6.1	7.5

* El margen está dentro de los límites de exactitud del instrumento.

SOLAREJEMPLO Sumario de Pruebas:
FALLO
LUGAR: DIR MAT ID. Cable: CIT1
OPERADOR Juan García Fecha/Hora: 11/12/2008
12:36:21
NVP: 69.0% UMBRAL DE ANOMALIA DE FALLO: 15% Estánd. Pruebas: TIA Cat 5
Channel + ACR
FLUKE DSP-100 N/S: 7120385 Tipo de Cable: UTP 100 Ohm
Cat 5
PASO LIBRE: 0.6 dB Versión de Estándares: 5.4
Versión de Software: 5.4

AVISO Excesivo ruido detectado. Podrá degradarse la exactitud de la medición.

Mapa de Cableado FALLO Result. TERM. RJ45: 1 2 3 4 5 6 7 8 B
| | | | a | | |
TERM. RJ45: 1 2 3 4 5 6 7 8

Par	1,2	3,6	4,5	7,8
Impedancia (ohmios), Límite 80-120	119	118		112
Longitud (m), Límite 100.0	17.8	17.8	3.3	18.0
Tiempo de Prop. (ns)	86	86	16	87
Diferencia Retardo (ns), Límite 50	70 A	70 A	0	71 A
Resistencia (ohmios)	3.6	3.2	Abierto	4.7

Atenuación (dB)			3.8	3.8	99.9 F	3.8
Límite (dB)			24.5	24.5	3.0	24.5
Margen (dB)			20.7	20.7	-96.9	20.7
Frecuencia (MHz)			100.0	100.0	1.5	100.0

Pares	1,2-3,6	1,2-4,5	1,2-7,8	3,6-4,5	3,6-7,8	4,5-7,8
NEXT (dB)	42.7	38.5	31.8	27.9*	46.6	66.7
Límite (dB)	32.0	29.2	29.0	27.3	32.1	60.3
Margen (dB)	10.7	9.3	2.8	0.6	14.5	6.4
Frecuencia (MHz)	52.5	75.4	78.2	97.8	51.2	1.0
NEXT del Remoto (dB)	45.0	45.6	34.7	29.7	66.5	64.2
Límite (dB)	34.0	28.1	29.0	27.4	51.0	60.3
Margen (dB)	11.0	17.5	5.7	2.3	15.5	3.9
Frecuencia (MHz)	39.9	88.0	78.2	96.9	3.8	1.0
ACR (dB)	66.5	0.0 F	46.1	0.0 F	65.9	0.0 F
Límite (dB)	47.8	57.8	33.2	57.8	45.0	57.8
Margen (dB)	18.7	-57.8	12.9	-57.8	20.9	-57.8
Frecuencia (MHz)	3.3	1.0	13.7	1.0	4.5	1.0
ACR del Remoto (dB)	63.9	0.0 F	66.1	0.0 F	67.0	0.0 F
Límite (dB)	46.8	57.8	52.6	57.8	48.1	57.8
Margen (dB)	17.1	-57.8	13.5	-57.8	18.9	-57.8
Frecuencia (MHz)	3.7	1.0	1.9	1.0	3.2	1.0

* El margen está dentro de los límites de exactitud del instrumento.

SUNSET
LUGAR: PISO 5
OPERADOR Juan García
13:48:06
NVP: 69.0% UMBRAL DE ANOMALIA DE FALLO: 15%
Channel + ACR
FLUKE DSP-100 N/S: 7120385
Cat 5
PASO LIBRE: 5.1 dB

Sumario de Pruebas: FALLO
ID. Cable: INT
Fecha/Hora: 11/12/2008
Estánd. Pruebas: TIA Cat 5
Tipo de Cable: UTP 100 Ohm
Versión de Estándares: 5.4
Versión de Software: 5.4

Mapa de Cableado PASA

Result. TERM. RJ45: 1 2 3 4 5 6 7 8 B
| | | | | | | | |
TERM. RJ45: 1 2 3 4 5 6 7 8

Par		1,2	3,6	4,5	7,8	
Impedancia (ohmios), Límite 80-120		103	106	105	103	
Longitud (m), Límite 100.0		112.7 F	110.5 F	111.7 F	111.3 F	
Tiempo de Prop. (ns)		545	534	540	538	
Diferencia Retardo (ns), Límite 50		11	0	6	4	
Resistencia (ohmios)		19.4	19.2	19.2	18.9	
Atenuación (dB)		23.3	22.7	23.0	22.8	
Límite (dB)		24.5	24.5	24.5	24.5	
Margen (dB)		1.2	1.8	1.5	1.7	
Frecuencia (MHz)		100.0	100.0	100.0	100.0	
Pares	1,2-3,6	1,2-4,5	1,2-7,8	3,6-4,5	3,6-7,8	4,5-7,8
NEXT (dB)	43.0	38.3	48.7	58.3	55.6	60.8
Límite (dB)	31.7	31.1	40.3	51.5	43.2	54.8
Margen (dB)	11.3	7.2	8.4	6.8	12.4	6.0
Frecuencia (MHz)	54.5	59.0	16.7	3.5	11.3	2.2
NEXT del Remoto (dB)	52.3	48.1	47.6	34.2	52.1	43.0
Límite (dB)	43.4	41.3	38.8	29.1	43.1	37.6
Margen (dB)	8.9	6.8	8.8	5.1	9.0	5.4
Frecuencia (MHz)	11.0	14.7	20.6	76.5	11.4	24.3
ACR (dB)	26.2	21.2	39.8	54.4	48.6	57.7
Límite (dB)	14.4	13.1	30.9	47.3	35.3	51.3
Margen (dB)	11.8	8.1	8.9	7.1	13.3	6.4
Frecuencia (MHz)	54.5	59.1	16.7	3.5	11.3	2.2
ACR del Remoto (dB)	45.1	40.0	37.6	14.6	45.1	32.4
Límite (dB)	35.6	32.4	28.4	8.2	35.2	26.3
Margen (dB)	9.5	7.6	9.2	6.4	9.9	6.1
Frecuencia (MHz)	11.0	14.7	20.6	76.5	11.4	24.3

SOLAREJEMPLO — Sumario de Pruebas: FALLO
LUGAR: EDIF 1 PB — ID. Cable: A HUB 1RO
OPERADOR OPERADOR Juan García — Fecha/Hora: 11/12/2008 16:33:35
NVP: 69.0% UMBRAL DE ANOMALIA DE FALLO: 15% — Estánd. Pruebas: TIA Cat 5
Channel + ACR

FLUKE DSP-100 N/S: 7120385
Cat 5
PASO LIBRE: 14.7 dB

Tipo de Cable: UTP 100 Ohm

Versión de Estándares: 5.4
Versión de Software: 5.4

Mapa de Cableado FALLO

Result.	TERM. RJ45:	1	2	3	4	5	6	7	8	B
		x	x	x	|	|	x	|	|
	TERM. RJ45:	3	6	1	4	5	2	7	8	

Par			1,2	3,6	4,5	7,8
Impedancia (ohmios), Límite 80-120			104	104	103	104
Longitud (m), Límite 100.0			9.1	9.1	8.9	8.9
Tiempo de Prop. (ns)			44	44	43	43
Diferencia Retardo (ns), Límite 50			1	1	0	0
Resistencia (ohmios)			2.6	0.7	1.6	1.6
Atenuación (dB)			1.8	2.0	2.1	1.9
Límite (dB)			24.5	24.5	24.5	24.5
Margen (dB)			22.7	22.5	22.4	22.6
Frecuencia (MHz)			100.0	100.0	100.0	100.0
Pares	1,2-3,6	1,2-4,5	1,2-7,8	3,6-4,5	3,6-7,8	4,5-7,8
NEXT (dB)	52.1	44.1	46.2	49.7	50.5	48.4
Límite (dB)	32.3	27.1	31.5	28.0	29.0	29.5
Margen (dB)	19.8	17.0	14.7	21.7	21.5	18.9
Frecuencia (MHz)	50.4	100.0	55.8	89.1	78.0	72.9
NEXT del Remoto (dB)	57.7	43.8	45.1	49.5	56.6	47.7
Límite (dB)	36.9	27.7	28.9	27.5	31.3	30.4
Margen (dB)	20.8	16.1	16.2	22.0	25.3	17.3
Frecuencia (MHz)	27.0	93.4	79.2	95.4	57.1	65.1
ACR (dB)	61.8	72.6	68.4	79.1	89.8	82.7
Límite (dB)	32.7	38.4	47.8	52.2	57.8	57.8
Margen (dB)	29.1	34.2	20.6	26.9	32.0	24.9
Frecuencia (MHz)	14.3	8.8	3.3	2.0	1.0	1.0
ACR del Remoto (dB)	57.7	73.5	78.9	74.6	86.3	64.0
Límite (dB)	26.7	52.2	54.6	44.0	57.8	39.6
Margen (dB)	31.0	21.3	24.3	30.6	28.5	24.4
Frecuencia (MHz)	23.6	2.0	1.5	5.0	1.0	7.8

4. Fibra óptica.

4.1. Sistemas de transmisión de fibra óptica.

Para poder implementar la tecnología de transmisión por medio de luz, es necesario contar con todo un sistema diseñado para este uso, los componentes básicos del mismo son:

- Fuente óptica: Convierte la señal eléctrica en luz.
- El cable de fibra óptica que transporta la señal.
- El detector óptico que convierte la señal nuevamente a electrones.

Como fuentes ópticas se emplean comúnme el diodo LED o LD de modulación directa, mientras que como detector óptico se emplean el DP o el PIN – PD de alta sensibilidad y de respuesta veloz.

4.2. Características de la luz.

La luz se puede definir como el agente físico que ilumina objetos y los hace visibles, siendo emitida por cuerpos en combustión, ignición, incandescencia, etc. Desde el punto de vista físico, la luz es una radiación u onda electromagnética. El espectro electromagnético se extiende desde las ondas de radio hasta los rayos gamma. De todo este espectro, sólo una zona muy pequeña es detectable por el ojo humano, y es lo que se llama el espectro visible o luz visible.

Toda onda está caracterizada por dos parámetros fundamentales:

- La velocidad de propagación.
- La frecuencia.

La velocidad de propagación es la distancia recorrida por una señal en una unidad de tiempo. Toda onda electromagnética se desplaza en el vacío a 300.000 km/s. La frecuencia es el número de veces que la onda repite su período en un segundo; en el caso de la luz es del orden de varios cientos de billones de ciclos por segundo.

Otro parámetro a considerar es el de longitud de onda, que se refiere a la distancia que la señal viaja durante un período, es por esta razón que se mide en metros.

$$\lambda = C/F$$

- λ: Longitud de onda.

- C: Velocidad de propagación.
- f: Frecuencia.

La idea de longitud de onda o de frecuencia dentro del espectro visible, se asocia a la idea de un determinado color de una determinada luz. Una luz de un color puro se llama monocromática. Si está compuesta por todos los colores, se llama luz blanca.

4.3. Propagación de la luz.

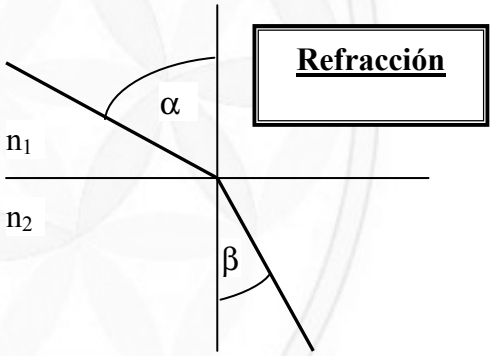
La luz se propaga en el vacío en forma rectilínea de acuerdo con lo que se denomina rayo o haz lumínico, en cualquier medio transparente cumple esta propiedad, siempre que la composición de ese material sea la misma en todo su recorrido. Todo medio físico opondrá resistencia al pasaje de una señal electromagnética, produciendo el efecto de disminuir su velocidad respecto al vacío. La relación entre la velocidad de la luz en el vacío y en un medio real se denomina índice de refracción.

4.4. Reflexión y refracción de la luz.

Al incidir una onda luminosa sobre una superficie plana divisoria de dos medios de índice de refracción diferente, su trayectoria se desviará acorde a la siguiente relación:

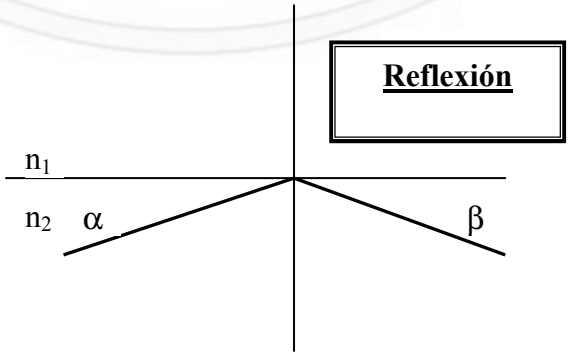
$$\text{Sen } \alpha / \text{sen } \beta = n_2 / n_1$$

$n_2 > n_1$



Como se puede apreciar, si se va incrementando el ángulo de incidencia desde n_2 a n_1, llegará un momento en el cual, el ángulo α llegará a ser de 90 grados, siendo siempre β menor a este valor (si: $n_2 > n_1$). Superado este umbral, el haz de luz deja de pasar a la superficie n_1, para producir el fenómeno denominado Reflexión total, en el cual la luz se propaga dentro del medio n_2, con un ángulo igual al de incidencia.

$n_2 > n_1$



Si el ángulo de incidencia del haz de luz se mantiene inferior al valor descrito, la luz se reflejará dentro de la superficie n_2. Al ángulo dentro del cual se produce la reflexión total se lo denomina Angulo de acepción o aceptación.

4.5. Fibra óptica (F.O.) (Descripción general).

La F.O. es un dispositivo de material dieléctrico (no conductor de c.e.) que es capaz de confinar y guiar la luz.

Las F.O. usadas en telecomunicaciones están formadas por dos cilindros concéntricos llamados núcleo y revestimiento con diferentes índices de refracción (n1 en el revestimiento y n2 en el núcleo), por medio de los cuales, si se ingresa un haz de luz dentro del cono de acepción, se producirá una y otra vez el fenómeno de reflexión total, transportando de esta forma la señal. Los diámetros que se suelen emplear son 125 µm para el revestimiento y desde los 9 a 62,5 µm para el núcleo acorde al modo (a tratar más adelante).

4.6. Propiedades de la F.O:

Basados en los diferentes parámetros del material, es que se puede clasificar las F.O. en cuatro grupos acorde a diferentes propiedades:

<table>
<tr><td rowspan="3">a.</td><td rowspan="3">Propiedades Ópticas</td><td rowspan="2">a.1.</td><td rowspan="2">Perfil de refracción</td><td>a.1.1.</td><td colspan="2">Monomodo</td></tr>
<tr><td rowspan="2">a.1.2.</td><td rowspan="2">Multimodo</td><td>- Índice gradual</td></tr>
<tr><td>a.2.</td><td colspan="4">Apertura Numérica</td><td>- Índice escalón</td></tr>
<tr><td rowspan="4">b.</td><td rowspan="4">Propiedades de transmisión</td><td rowspan="2">b.1.</td><td rowspan="2">Atenuación</td><td>b.1.1.</td><td colspan="2">Intrínseca</td></tr>
<tr><td>b.1.2.</td><td colspan="2">Extrínseca</td></tr>
<tr><td>b.2.</td><td colspan="4">Ancho de banda</td></tr>
<tr><td>b.3.</td><td colspan="4">Diámetro del Campo modal</td></tr>
<tr><td rowspan="4">c.</td><td rowspan="4">Propiedades geométricas</td><td>b.4.</td><td colspan="4">Long. de onda de corte</td></tr>
<tr><td>c.1.</td><td colspan="4">Diámetro del revestimiento</td></tr>
<tr><td>c.2.</td><td colspan="4">Diámetro del núcleo.</td></tr>
<tr><td>c.3.</td><td colspan="4">Concentricidad</td></tr>
<tr><td rowspan="4">d.</td><td rowspan="4">Propiedades físicas</td><td>c.4.</td><td colspan="4">No circularidad</td></tr>
<tr><td>d.1.</td><td colspan="4">Módulo de Young</td></tr>
<tr><td>d.2.</td><td colspan="4">Carga de rotura</td></tr>
<tr><td>d.3.</td><td colspan="4">Alargamiento en el punto de rotura</td></tr>
<tr><td>d.4.</td><td colspan="4">Coeficiente de dilatación lineal</td></tr>
</table>

a. Propiedades ópticas:

a.1. Perfil de refracción:

Se llama perfil de índice de refracción de una fibra óptica, a la variación que tiene el mismo conforme al desplazamiento respecto a la sección transversal de la fibra, es decir a lo largo de su diámetro. La señal luminosa se desplaza en el interior de la fibra en rayos o haces denominados modos; se puede decir que un modo es cada una de las distintas posibilidades de propagación de la energía en el interior de la fibra. El número de modos en que se propagan en la fibra se determina según:

- La longitud de onda óptica.
- La diferencia de índice de refracción entre núcleo y revestimiento.
- El perfil de índice de refracción.
- El diámetro del núcleo.

a.1.1. Monomodo (Single Mode Fiber): En realidad esta no se diferencia de la que se verá a continuación en cuanto a su variación, sino por el diámetro de su núcleo el cual es de 9 µm. Este tamaño la caracteriza por ser el núcleo más pequeño dentro de las fibras existentes, el cual guarda una relación muy cercana con la longitud de onda de la señal que transportará.

b.1.2. Multimodo (Multimode Fiber): La señal se transmite por distintos caminos dentro del núcleo, formando distintos modos.

- Índice gradual (Grade Index): El índice de refracción del núcleo varía en forma progresiva desde el centro a la periferia (en forma parabólica), siendo máximo en el centro de la fibra. El índice del revestimiento se mantiene constante y por supuesto de menor magnitud.

- Índice escalón (Step Index): Este posee un índice de refracción constante, tanto en el núcleo (n2) como en el revestimiento (n1), produciendo un salto de valor en la interfaz entre ambos, que es donde se produce la reflexión total. (Este sistema es también el empleado por las fibras monomodo, teniendo en cuenta que su diámetro es menor que en las multimodo).

a.2. Apertura numérica:

La Apertura numérica es un parámetro que da idea de la cantidad de luz que puede ser guiada por una F.O. Por lo tanto cuanto mayor es la apertura numérica de una fibra, mayor es la cantidad de luz que puede guiar, es decir más cantidad de luz es capaz de aceptar en su núcleo. En la tabla siguiente se representan valores típicos de fibras empleadas en telecomunicaciones:

Tipo de fibra	Apertura numérica

Multimodo índice escalón	0,3 - 0,4
Multimodo índice gradual	0,2
Monomodo	0,1

b. Propiedades de transmisión:

Las propiedades de transmisión son las que influyen el pasaje de la señal a través de la fibra.

b.1. Atenuación:

La atenuación es la disminución de la potencia que sufre toda señal a medida que avanza a través de un medio físico. La atenuación es dependiente del medio, la distancia y la longitud de donde de la señal. La unidad de medida es el "decibel" [dB] que es una magnitud relativa y logarítmica que responde a la siguiente fórmula:

$$At[dB] = 10 * Log (Ps / Pe)$$

Donde Pe es la potencia de entrada y Ps la de salida. Como es función de la distancia recorrida, se expresa en dB/km, donde valores típicos son 0,3 a0,4 dB/km en fibras monomodo.

Como se aprecia en el cuadro de clasificación las causas que influyen en la atenuación pueden ser Intrínsecas o Extrínsecas.

b.1.1. Intrínseca:

Son los factores propios de la F.O. dentro de los cuales existen dos fundamentales que se mencionan a continuación:

- ***Absorción* de la luz debido al material:** Los materiales empleados para la fabricación de la fibra en telecomunicaciones actualmente son dos(SiO_2 y GeO_2), estos presentan una absorción a la luz UV y una buena parte de la IR. Entre estas zonas queda una amplia gama donde el vidrio de Sílice es generalmente usado por ser más transparente. A pesar de esto, como es lógico imaginar, no se obtiene una pureza 100 % y existen materiales constituyentes de la fibra que presentan absorción de la luz en la zona antes mencionada; Los que presentan mayor anomalías son los iones OH^- generados durante el proceso de fabricación por la humedad ambiental, y que presentan picos de absorción en las longitudes de onda de: 950, 1.250, y 1.370 nm.
- ***Dispersión* de la luz debido al material:** Cuando la luz se propaga a través de un medio material no completamente homogéneo, a medida que sufre colisiones con inhomogeneidades, se produce el efecto conocido como dispersión Rayleigh, en el cual parte de ese haz de luz, sale esparcido en todas direcciones con igual potencia, la cual será función del tamaño de la colisión y de la longitud de onda. La cantidad de luz dispersada será menor cuanto mayor sea la longitud de onda. Este último concepto, es trascendente conceptualmente pues esta es la razón por la cual la tendencia es a emplear longitudes de onda cada vez mayor,

la cual no es de cualquier magnitud, sino como se verá más adelante responde a ciertos valores de "**ventana**", los cuales en la actualidad son de 850, 1300 y 1550 nm.

b.1.2. Extrínseca:

Los factores externos son aquellos que no guardan relación con el material o la composición de la fibra, sino con el ambiente donde esta opera.

El principal factor externo que afecta a las pérdidas de luz en una fibra lo constituyen las deformaciones mecánicas, dentro de las cuales las más importantes son las curvaturas. El manejo de las fibras, hace inevitable que sufran curvaturas; esto puede provocar que determinados rayos, no provoquen reflexión total, perdiendo luz por refracción que escapa al núcleo. Estas curvaturas se pueden diferenciar en dos tipos:

- Macrocurvaturas: Curvaturas con un radio de giro de 1 cm o más.
- Microcurvaturas: Radio de giro menor a 1 cm.

Como detalle significativo cabe señalar que una F.O. multimodo índice escalón sometida a una curvatura de **radio = 1 cm** puede perder la **mitad de la luz guiada**.

b.2. Ancho de banda (Δf):

El concepto de ancho de banda se lo puede definir desde varios puntos de vista:

- El primero de ellos es matemáticamente, a través de la Transformada discreta finita de Fourier, y representar por medio de esta el Espectro de frecuencias, en la gráfica de este, se nota nítidamente el área donde la función se hace cero por primera vez. Si se calcula el área allí comprendida, este será del orden del 90 % del área total de la función. Este porcentaje *es donde se encuentra concentrada la mayor parte de la energía de una señal*, y es quizás la definición más escolástica de ancho de banda.
- Otra definición se suele hacer respecto a los parámetros físicos de la señal, y se expresa domo el intervalo de frecuencias donde la señal sufre una atenuación dentro de parámetros normales; estos "Parámetros normales" se los suele considerar como de 3 dB encuadrado dentro de las llamadas frecuencias de corte Superior e Inferior. { $\Delta f = fcs - fci$ }.
- Por último se puede considerar el Δf como el parámetro que determina la cantidad de información que es capaz de transportar una señal.

La información viaja a través de la fibra en forma de pulsos luminosos muy estrechos y separados en el tiempo. Lo ideal, sería que la forma de estos pulsos no se modificara durante toda su trayectoria. Sin embargo esto no sucede, y la capacidad de transmitir información viene limitada por una distorsión de la señal que resulta en el ensanchamiento de los pulsos luminosos al transmitirse a lo largo de la fibra. Los factores que contribuyen a este ensanchamiento son dos:

- Dispersión modal: (Sólo en fibras Multimodo), se debe a que los diferentes rayos (modos), no recorren las mismas distancias y por lo tanto llegan desfasadas en el tiempo al final de la fibra.
- Dispersión cromática: Se debe a la variación del índice de refracción de la fibra según la longitud de onda de la luz que transmita.

b.3. Diámetro del campo modal:

Este es un parámetro de gran importancia en las fibras monomodo. A partir de este se pueden calcular las pérdidas por empalmes, por microcurvaturas y dispersión cromática de la fibra.

El diámetro del campo modal da idea de la extensión de la mancha de luz del modo fundamental a la salida de la fibra. Su valor aumenta proporcionalmente al aumento de la longitud de onda

b.4. Longitud de onda de corte:

La F.O. monomodo, no guía un único rayo para todas las longitudes de onda. Sólo a partir de una cierta longitud de onda, se comporta como monomodo. Para longitudes de onda por debajo de ese valor, la fibra transporta varios haces de luz y se comporta como multimodo. La longitud de onda en que se produce la separación entre el comportamiento mono y multimodo para una fibra se llama longitud de onda de corte. En el caso de las empleadas en telecomunicaciones, este valor se encuentra comprendido entre 1190 y 1320 nm.

c. Propiedades geométricas:

En este punto los parámetros que se tratan se refieren a la forma de la fibra.

c.1. Diámetro del revestimiento:

Es el diámetro exterior del revestimiento de la fibra, este dato cobra importancia ya que gran parte de los dispositivos de conexión y empalme lo emplean como referencia de alineación.

c.2. Diámetro del núcleo:

Es el diámetro exterior del núcleo. Este valor diferencia en una fibra el comportamiento monomodo o multimodo para un determinado valor de longitud de onda. La uniformidad de este parámetro es importante para conseguir buenos empalmes y conectorizados.

c.3. Concentricidad:

Este parámetro se refiere a la concentricidad del núcleo respecto al revestimiento, y mide la diferencia de distancia entre ambos centros. Para lograr mayor eficacia en los empalmes, es importante que el núcleo t el revestimiento sean concéntricos.

c.4. No circularidad:

Determina la deformación que sufre respecto a una circunferencia perfecta, tanto el núcleo como el revestimiento, en la tabla siguiente se describen los valores estandarizados:

Tipo de fibra	Diámetro del núcleo (µm)	Diámetro del revestimiento (µm)
Monomodo	10 ± 1	125 ± 2
Multimodo 50/125	50 ± 3	125 ± 2
Multimodo 62,5/125	62,5 ± 3	125 ± 2

d. Propiedades físicas:

Estas propiedades se refieren al comportamiento mecánico de la fibra.

d.1. Módulo de Young:

Es la fuerza por unidad de área que produce un alargamiento unidad en la fibra. Su valor es del orden de 7.000 kp/mm^2.

d.2. Carga de rotura:

Es la mínima fuerza por unidad de área capaz de romper la fibra. Su valor es del orden de 400 kp/mm^2 , es importante hacer la comparación con el alambre de cobre, cuya carga de rotura es 25 kp/mm^2.

d.3. Alargamiento en el punto de rotura:

El alargamiento en el punto de rotura de una fibra es del orden del 5%.

d.4. Coeficiente de dilatación lineal:

Indica el alargamiento relativo que sufre la fibra por cada grado de temperatura. El valor para la fibra es del orden de 0,5 * 10^{-6} /°C. Esto quiere decir que 1.000 m de F.O. sufrirán un alargamiento de 25mm al pasar de 20 °C a 70 °C.

4.7. Los cables de F.O.

Los cables de fibra óptica empleados en telecomunicaciones, son los de SiO_2 dopado con diferentes elementos. Al principio, su aplicación se limitó a los circuitos de larga distancia debido a su gran ancho de banda, actualmente esta situación está cambiando, llagando en muchos casos a abonados, y empleándolo en cableados verticales y horizontales de datos, como así también en CATV.

Se entiende por cable de fibra óptica el conjunto formado por las distintas partes que permiten que uno o varios hilos de F.O. puedan ser empleados en la instalación de un sistema de transmisión de información.

4.7.1. Elementos que constituyen un cable:

a. Elemento de tracción:

Es el elemento resistente del cable a los esfuerzos de tracción. Suele emplearse para esto alambre de acero, hilado sintético (Kevlar), fibras de vidrio, etc.

b. Fibras ópticas:

Las fibras con su protección secundaria (revestimiento) se disponen alrededor del elemento de tracción. En algunos casos suele rellenarse de algún material que impida la penetración de la humedad (gel).

c. Fajado:

Todo el conjunto se envuelve en cinta formando una estructura compacta.

d. Vaina externa:

Su función es brindarle al conjunto una protección mecánica adicional ofreciendo además una barrera contra la penetración de la humedad. Estas cubiertas son de tipo:

- Polietileno (PE).
- Policloruro de vinilo (PVC).
- Plásticos fluorados (FEP).

4.7.2. Tipos de cables:

Los distintos tipos de cables se pueden clasificar desde varios puntos de vista:

4.7.2.1. Por el tipo de instalación:

a. Aérea:

- Cables autoportantes: Son cables que se instalan en postes y que no necesitan de ningún elemento externo para su sustentación y protección frente a las tensiones provocadas por su propio peso, los vientos y el hielo.
- Cables adosados: Como su nombre lo indica, son cables que se adosan o "cosen" a un cable de acero u otro cable. Este segundo elemento ajeno al cable en sí es el que le proporciona todas las condiciones de aéreo. Existen dos familias: Cables adosados longitudinalmente y cables adosados en espiral.

b. Subterránea:

Dentro de este esquema existen tres grandes familias:

- Cables para instalación en ducto: Su cubierta exterior no debe quedar expuesta a la intemperie.
- Cables para enterramiento directo: Permiten ser depositados directamente en las zanjas. Se suele colocar sobre ellos y a una altura del orden de los 50 cm, cintas metálicas también enterradas con identificaciones de cables de F.O. para permitir la identificación ulterior de estos cableados con detectores de metales.
- Cables para instalaciones en galerías y túneles: Semejantes al primero, con cubiertas no propagadoras de llama, nula emisión de humo y cero halógenos.

c. Submarina:

Este tipo de cable, reúne las características básicas para anclarse en el fondo del mar. Se encuentran muy difundidos en la actualidad en virtud de reemplazar más eficientemente las transmisiones satelitales.

4.7.2.2. Por la disposición y forma de los hilos:

a. Fibras ajustadas:

Desarrolladas en Japón. Se coloca la protección directamente sobre la fibra. Se suelen utilizar para cables de interior donde las variaciones climáticas no son tan grandes como en los cables de planta exterior. En estos casos, los cables se los protege primariamente con una cubierta de 250 µm y una secundaria de 900 µm de diámetro extruída sobre la primera. Este tipo de protección transmite como es evidente, cualquier tipo de tensión a la fibra puesto que la segunda protección plástica está en íntimo contacto con la fibra.

b. Fibras holgadas:

En este tipo de fibras, se aplica en primer lugar una capa de acrilato con un diámetro total de 250 µm a la fibra desnuda y posteriormente se lo pinta para diferenciar los hilos entre sí. Una vez aplicada esta primera protección, el siguiente paso es construir alrededor de ella un tubo holgado, dentro de este tubo, se pueden situar una o varias fibras. El motivo de utilizar este tipo de construcción es que el material del que está compuesto el tubo y el diámetro del centro hueco del tubo, aíslan a las fibras de las tensiones exteriores producidas principalmente por las variaciones de temperatura. Este tipo de protección se usa principalmente en cables que se van a instalar en el exterior y generalmente en fibras monomodo.

c. Núcleo ranurado:

El núcleo ranurado se suele utilizar en Países de influencia Francesa. Consiste en introducir fibra desnuda o fibra en cintas en las ranuras helicoidales de un extremo central previamente ranurado. Este tipo de cable está pensado para su instalación en planta exterior y presenta la desventaja, frente al tubo holgado, de constituir un elemento muy rígido que dificulta su instalación. Además como el cableado de las fibras es helicoidal, la segregación de las fibras es complicada.

d. Cables de cinta de fibra:

En este cable las fibras son dispuestas formando cintas planas de 4,6 u 8 hilos. Primeramente a las fibras se les aplica una protección de acrilato generalmente de 250 µm, y se disponen linealmente con una separación mínima entre ellas. La cinta se construye situando el número de fibras deseado en forma adyacente (2, 4,6 u 8), y extruyendo una capa plástica que las mantiene unidas y les da una protección de tipo ajustado.

e. Cables de capas circulares:

Dentro de este grupo se distinguen los cables que poseen en su centro el elemento de tracción y rodeando a este con distribución circular se encuentran las distintas fibras. Si las fibras se encuentran formando un solo círculo se denominará

de una capa. Si luego de una capa primaria, se encuentra envolviendo a esta otra distribución circular de fibras, será de dos capas y así sucesivamente.

f. Cables de grupos:

Al igual que el anterior, en el centro se distingue el elemento de tracción, pero rodeando el mismo se arman grupos de fibras de capas.

4.8. Conectores.

Los conectores son los elementos que se emplean para la unión del cable óptico con los equipos de transmisión. La misión del conector es lograr la mayor eficiencia en el acople de los dispositivos. Por otro lado deben proteger los extremos de la fibra del daño que pueden sufrir durante el manejo, resistir las cargas de tensión esperadas en un cable y proteger las fibras de las condiciones ambientales adversas como el polvo y la humedad. Los principales conectores ópticos empleados hoy son:

4.8.1. FC: (Fibra a contacto) Es el conector más empleado en el ámbito de las comunicaciones, ofreciendo pérdidas del orden de 0,25 dB con fibras monomodo.

4.8.2. SC: (Sin contacto) Descendiente directo del FC, es el conector adecuado para aquellos sistemas de telecomunicaciones en los que exista una alta densidad de conexionado y se exijan altas prestaciones.

4.8.3. ST: Es el más empleado en instalaciones informáticas. Posee pérdidas del orden de 0,3 dB.

4.8.4. SMA, MT: Son otros tipos de conectores que se mencionan a título ilustrativo pero de poco uso en el ámbito informático.

Como último párrafo en lo referido a conectores se definen conceptos a tener en cuenta para su operación:

EVITAR:

- Tocar la cara pulida del conector.
- Dar golpes o permitir que caiga en superficies duras.
- Sacar el conector tirando de cable.
- Doblar el cable excesivamente en su unión con el cuerpo del conector.
- Forzar la entrad del conector.

PROCURAR:

- Mantener limpia la cara pulida del conector.
- Tratarlo con normalidad.
- Sacar el conector utilizando el cuerpo.
- No quebrar el cable en su unión con el conector.

4. 9. Empalmes:

En los casos en que la distancia entre el emisor y el receptor sea mayor que la longitud de fabricación del cable, es necesario realizar un empalme. Las pérdidas de un empalme, dependen fundamentalmente del alineamiento de los extremos de las fibras y de la correcta preparación de los extremos. La forma más habitual de realizar los empalmes es mediante la técnica de fusión. El proceso se basa en poner en contacto dos extremos preparados de la fibra y provocar la fusión del vidrio de manera tal que se produzca la continuidad mecánica y óptica. Los extremos de la fibra se calientan mediante un arco eléctrico o mediante microllamas. Existen también técnicas de empalme que utilizan dispositivos mecánicos para unir fibras aunque este sistema es considerablemente más bajo que el de fusión, para este tipo de empalmes existen diferentes sistemas: Spingroove, Norland, Elastromeric (GTE) y Rotativo (ATT). Las pérdidas por empalme, oscilan entre 0,05 y 0,2 dB.

Los empalmes son uno de los elementos más críticos dentro de toda red de fibra y por lo tanto deben ir debidamente protegidos. Se emplean cajas o protectores de empalme para esta actividad, existiendo una gran variedad de cajas y en el caso de los protectores consisten en un tubo de plástico que contiene un alambre de acero para proporcionarle rigidez y protección frente a las tensiones externas. El plástico es del tipo termocontraíble para conseguir un perfecto sellado.

4.10. Ventajas de la F.O:

a. Elevado ancho de banda, lo cual permite una gran capacidad de transmisión de información, que se traduce en un mayor rendimiento de los sistemas.

b. Inmune a interferencias electromagnéticas: Las señales se pueden transmitir a través de zonas eléctricamente ruidosas con muy bajo índice de error.

c. La diafonía no es un problema debido a la no inducción de campos eléctricos y magnéticos.

d. Puesto que las fibras no irradian energía electromagnética, la señal por ellas transmitida no puede ser captada por el exterior, además es técnicamente imposible extraer subrepticiamente información de una F.O. sin alterar notoriamente los parámetros de transmisión.

e. No plantea problemas de descargas eléctricas ni de incendios.

f. Por su reducido tamaño y peso, y relativamente alta resistencia mecánica, los problemas de almacenamiento, transporte y sobre todo de instalación se ven disminuidos.

g. Pueden fabricarse cables muy livianos, ya que el peso específico del vidrio es la cuarta parte del cobre, y se debe tener en cuenta que su diámetro es sensiblemente menor.

h. La materia prima es el Silicio que es uno de los recursos más abundantes de la superficie terrestre (25,8 %).

4.11. Desventajas de la F.O:

a. Si bien la materia prima es abundante, se requiere en comparación con la purificación metálica, mucha más energía para obtener vidrio de la pureza química necesaria.

b. En virtud de ser un producto delicado, implica durante su instalación un tratamiento particularmente cuidadoso.

c. Exige mayor precisión en conexionado y empalmes.

d. Tiempo de vida de los dispositivos activos, puntualmente fotoemisores y fotodetectores, menor a los dispositivos eléctricos.

4.12. Otros elementos ópticos.

En virtud del acelerado avance de la fibra óptica y la evolución de la tecnología, han aparecido nuevos elementos que amplían el esquema de empleo de la F.O. entre ellos se deben citar:

4.11.1. Divisores ópticos (Splitter):

Son los elementos encargados de dividir una señal óptica de entrada en n señales de salida sin afectar más que su potencia.

Su aplicación se encuentra en redes ópticas pasivas, donde un emisor ataca a varios receptores en buses ópticos para poder situar equipos activos en paralelo, en sistemas de supervisión, etc.

Actualmente existen dos tecnologías para construir divisores ópticos: La primera es la de fusión, donde una fibra se empalma a dos de salida a través de un empalme por fusión y es la más adecuada para construir divisores 1x2. La segunda tecnología es la planar, que consiste en litografiar sobre un substrato de vidrio muy especial, caminos ópticos que ramifican la entrada, esta es la tecnología más adecuada para divisores de alta capacidad: 1x4, 1x8, 1x16 y 1x32.

4.11.2. Multiplexores de longitud de onda:

Estos dispositivos también conocidos como WDM y WWDM permiten combinar/separar dos señales ópticas de diferente longitud de onda en una misma fibra. Estas dos señales pueden propagarse por a misma fibra en la misma dirección o en sentido contrario. Con estos dispositivos se puede duplicar la capacidad de una red, añadiendo solamente nuevos emisores/receptores que trabajen por ejemplo a 1.550 nm si ya se dispone de equipos a 1.300 nm.

4.11.3. Atenuadores ópticos:

Estos dispositivos se están utilizando para atenuar la potencia de salida de un emisor o la recibida en el otro extremo y se implementa en aquellas redes que se construyen parcialmente pero que tienen en mira una ampliación cercana. Esta ampliación implicará nuevos cables, empalmes, conectores, etc. En estos casos los equipos emisores, se dimensionan en la total capacidad de la red, por lo que en su primera fase, pueden estar dando demasiada potencia óptica y llegar a saturar a los receptores.

4.11.4: Conmutadores ópticos:

Los conmutadores ópticos se utilizan cuando se quiere dar una mayor seguridad a la red puesto que posibilitan la conexión de una entrada óptica a una de las n posibles salidas. Su uso permite aislar elementos defectuosos de la red, cuya puesta fuera de servicio puede afectar el funcionamiento del conjunto y re encaminar por caminos alternativos cuando se detecta una falla.

ANEXO 2 (Consideraciones a tener en cuenta para el diseño de un CPD).

1. Consideraciones iniciales.

Un CPD consiste en un edificio usado para mantener en él una gran cantidad de equipamiento electrónico. Suelen ser creados y mantenidos por grandes organizaciones con objeto de tener acceso a la información necesaria para sus operaciones.

Un proyecto de diseño y construcción de un Centro de Proceso de Datos se compone de los siguientes apartados:

- Obra civil (cerramientos): Estructura modular para la construcción del CPD, chapa de acero galvanizado revestido con aislante, según normativa EN-1047-2 y resistencia al fuego RF120 y estanqueidad al agua IP65. Apantallamientos Eléctricos. Definición y características del falso suelo (estructura y rampas de acceso); distancias; materiales. Definición y características del falso techo; Iluminación; limpieza de canalizaciones de líquidos. Puertas y accesos, disposición de salidas.
- Sistemas de climatización: dimensionamiento del aire en función de m3 y disipación de calor aproximada de los componentes del CPD. Canalizaciones y ubicación de componentes. Redundancia.
- Adecuación de elementos eléctricos: alumbrado.
- Alimentación UPS. Acometida eléctrica (doble). Alimentación auxiliar (de socorro o grupo electrógeno); sistema de activación. Alumbrado de emergencia. Alimentación corriente sucia. Todo el cableado suministrado cumplirá la normativa respecto a no contener componentes halógenos.
- Adecuación infraestructura del cableado de datos e infraestructura física de equipamiento (rack).
- Infraestructura de alimentación eléctrica: UPS en función de total KVA. Grupo electrógeno ó alimentación de socorro.
- Seguridad (video vigilancia, control de accesos, alarma y extinción de incendios, controles de humedad y humos).
- Auditoría: calidad de energía, termografía, seguridad (prevención de incendios). Auditorías medioambientales, limpiezas técnicas de CPD. Seguridad Operativa: falta de procedimientos, omisión de indicadores críticos, equipos en mala ubicación. Mantenimiento de CPD.

2. Servicios.

Un Centro de Proceso de Datos (CPD) es el conjunto de recursos físicos, lógicos, y humanos necesarios para la organización, realización y control de las actividades informáticas de una empresa.

A la hora de realizar el diseño de un Centro de Proceso de Datos, la idea inicial que se contempla es la del diseño de una sala de comunicaciones de grandes dimensiones donde se van a ubicar potentes servidores u ordenadores

Esta premisa inicial pone de manifiesto uno de los primeros condicionantes del diseño de CPD's, el carácter "crítico" de los datos que se manejan, puesto que la mayoría de las empresas dependen de la disponibilidad, seguridad y redundancia de la información que se almacena en sus servidores.

La no disponibilidad de esta información supone elevados costes para cualquier compañía, especialmente en las que el departamento financiero tiene que funcionar ininterrumpidamente, sin lentitud en el tráfico de datos y donde sus transacciones deben estar totalmente libres de errores.

Dada la demanda intensiva de estos datos, las prestaciones de la infraestructura de red son claves para un correcto funcionamiento y para evitar incurrir en costes generados por la no disponibilidad de los "datos".

Dado que la capacidad de gestionar la infraestructura física del centro de datos puede tener un impacto directo en el funcionamiento y rendimiento de nuestra red, a la hora de elegir soluciones de cableado, debemos apostar por soluciones avanzadas y contrastadas que nos permitan una libertad en el diseño para resolver así nuestras diversas y particulares necesidades.

3. Acondicionamiento de la Sala

Cerramiento perimetral, dotación de suelo técnico sobreelevado de calidad sala informática, falso techo de placas de fibra de vidrio.

Los Centros de Proceso y locales técnicos en general, deben diseñarse en un área específicamente concebida para ese uso, que cuente con las máximas medidas de seguridad, garantizando su funcionamiento, dentro de las normas que este tipo de locales exige.

El estudio de implantación ha de contemplar, entre otros, el tipo de cerramiento perimetral, los pasos para dotaciones y canalizaciones de servicios, riesgos de vecindad de locales limítrofes, compartimentación de suelo y techos, evacuación de personal en caso de emergencia, etc.

Los falsos suelos y falsos techos deben ser apropiados para el uso en salas informáticas, con alturas suficientes para el paso de conductos y en el caso de los suelos con capacidades de carga no inferiores a 20.000 N/m2, soportadas en estructuras de pedestales y largueros.

4. Climatización.

Equipos de climatización específicos para salas informáticas, control microprocesado de temperatura y humedad.

Para poder mantener el nivel de temperatura adecuado de los locales técnicos, así como el grado de humedad dentro de los límites medios de temperatura y humedad, se proponen dotaciones de equipos de climatización específicos para salas informáticas, del tipo servicio total, controlado por microprocesador y capaz de producir frío, calor y humectar o deshumectar de forma automática, dentro de unos márgenes de ± 1º C y ± 2% HR para valores de funcionamiento previstos de 21ºC y 60% HR.

Las unidades de climatización se calculan para un funcionamiento continuo 24 h/días y 365 días/año y su potencia frigorífica para una temperatura de bulbo seco interior de 24ºC será capaz de mantener las características de las salas para las variaciones de temperatura ambiente medias actuales y para el 120% de la carga total de los locales (carga eléctrica + aportaciones de los locales + iluminación + presencia no continua de personas en sala).

5. Instalación Eléctrica.

Del buen funcionamiento del suministro de energía, dependen todos los servicios de proceso y comunicaciones de la empresa, la instalación debemos dotarla de un cuadro específico para los Servicios de Información, comprobando la calidad de la tierra, y dimensionándolo para futuros crecimientos.

Los materiales (interruptores, magnetotérmicos, diferenciales, etc.) se instalarán con las últimas tecnologías para conexiones en caliente.

Las líneas desde el cuadro de distribución deben realizarse por canalizaciones de cable (normalmente bajo falso suelo), recomendando la utilización de una línea para cada equipo o grupo homogéneo de equipos.

Para eventuales cortes en el suministro, se recomienda la instalación de un equipo SAI y en determinadas ocasiones el respaldo de un grupo electrógeno de continuidad.

6. Sistemas ignífugos.

Los equipos y soportes de datos que se encuentran en locales sin medidas especiales de seguridad, son especialmente vulnerables ante riesgos de manipulación indebida, incendios o radiaciones, que pueden alterar y destruir el contenido de los mismos.

Las copias de back-up o los servidores de respaldo, también han de contar con protección ante eventuales riesgos que puedan afectar al servicio que deben proporcionar.

Los armarios ignífugos para datos, rack y equipos, proporcionan la más alta protección ante todo tipo de agentes externos como incendios, explosivos, acceso, gases, radiaciones y daños criminales.

La combinación de acero, células de hormigón y materiales especiales que absorben el calor, aseguran el mayor nivel de seguridad.

La protección certificada con la norma VDMA24991, asegura unos niveles de seguridad S120DIS.

Todos los armarios ignífugos están equipados con un tipo de cierre hermético de autosellado ante agentes agresivos externos (gas, fuego, agua, etc.), con el que basta impulsar la puerta a su posición cerrada, sin necesidad de cerrar con la llave.

7. Características físicas de un CPD.

Ante los factores que afectan a la seguridad física de un C.P.D., que a continuación detallamos, debemos de tener en cuenta una serie de características para el acondicionamiento de este.

Factores ambientales:

- Incendios.
- Inundaciones.
- Terremotos.
- Humedad.

Factores humanos:

- Robos.
- Actos vandálicos.
- Actos vandálicos contra el sistema de red.
- Fraude.
- Sabotaje.
- Terrorismo.

Actualmente los CPDs se han empezado a convertir en un conjunto de espacios con funciones y necesidades distintos:

- Núcleo de Procesamiento Principal
- Equipos de conmutación de Red
- Área de Impresión
- Área de Cintas/CD de back-up
- Área de Operadoras o exterior
- Área de Servidores
- Área de Aplicaciones

La separación en varias áreas presenta beneficios en términos de control de acceso, reducción del riesgo de fuego y control ambiental. Aunque los riesgos deberían ser mínimos, las consecuencias de un desastre aquí, pueden ser tan graves que merece la pena considerar otra línea de defensa — como dividir la sala principal en dos o más cuartos separados.

Si los cuartos se dividen con eficacia, es muy poco probable que se produzca un desastre que afecte a varios de los espacios. Para asegurar este punto, no debe haber ruta alguna entre los cuartos que permitan la propagación del fuego, del humo, del agua, de gases o de explosiones (las posibles rutas de cable deben ser selladas con material cortafuegos masillas, calafateados o espumas de silicona).

Los cuartos también necesitarán alimentación, HVAC (sistemas de calefacción, ventilación y aire acondicionado), protección contra incendios, seguridad y rutas de acceso independientes.

Es complicado estimar con seguridad los requisitos futuros de un CPD en entonos de edificios de oficinas. La prioridad del equipo de diseño debe ser permitir que el CPD se construya en base a lo requerido inicialmente y debería proporcionar, como mínimo, un espacio apropiado del CPD bastante grande. De forma orientativa, hasta un 10 % de la superficie útil del edificio se pueden requerir para este propósito.

Se requerirá:

- Una planta con altura de suelo a techo mínima de 3 m, preferiblemente más. Esto será suficiente para un piso con un falso suelo de 300 a 600 milímetros y proporcionará el suficiente espacio libre para los equipos y racks.

- Una ruta de acceso amplia para canalizaciones. La ruta debe ser grande y bastante fuerte para servir como toma de aire, material informático o para módulos de fuente de alimentación continua.

- Espacio para salas posibles de extensión.

Aunque los CPDs se sitúan normalmente en los sótanos, éste no es el único sitio ni de hecho es la mejor ubicación. Aunque el área del sótano no es normalmente el espacio más conveniente para oficinas y por lo tanto se usa más para servicios, hay que considerar el riesgo de inundaciones y de la posible capacidad de recuperación ante un desastre.

En muchos de los edificios de oficinas, el CPD es el mayor consumidor de alimentación y fuente de calor, por lo cual la provisión de sistemas de respaldo como grupos electrógenos, sistemas de alimentación ininterrumpida, SAIs o UPS, tiene gran importancia.

Los CPDs requieren un ambiente controlado en relación a la temperatura, la humedad y el polvo. La gran concentración de equipos de IT demanda sistemas de HVAC, para eliminar el exceso de calor y evitar niveles extremos de baja humedad, que provocan acumulación de estática; también sirven para limitar la presencia partículas aéreas.

Los niveles recomendados de temperatura y humedad son de 21 ° C y 50 % de humedad respectivamente y 95 % de eficiencia de filtrado de 5 micras con aire recirculado de tipo no evaporativo.

8. Detección – Extinción.

El CPD necesita un sistema propio de detección del fuego y de extinción. No se debe a que el CPD suponga en sí mismo una posible fuente de incendios, sino más bien al valor de la información almacenada y al considerable daño que supondría para el negocio una pérdida de la misma.

Los fuegos raramente comienzan en el CPD. Los CPDs resultan dañados más a menudo por los fuegos (o por el humo y gases) que comienzan en otras partes y se extienden a la sala de procesamiento de datos.

Los equipos pueden resultar seriamente dañados, por el humo y gases corrosivos (como el Cloruro de Hidrógeno producido en la combustión de los cables, salvo que se elija adecuadamente el cable). Estos equipos pueden también verse dañados por los materiales utilizados para la extinción del fuego incluyendo flurocarbono, agua y dióxido de carbono.

En muchos casos, el daño debido a los elementos de extinción del fuego es superior al producido por el fuego propiamente.

Dentro del CPD, los riesgos se reducen al mínimo guardando las impresoras, una fuente común de ignición, lejos de los otros equipos.

Todos los cables tendidos bajo el suelo deberían ser LSZH (Low Smoke Zero Halogen).

Los principios apropiados para la protección contra incendios son: reducir la probabilidad de que un fuego comience, reducir la probabilidad de que un fuego se disperse y reducir el daño mínimo que un fuego puede causar.

El riesgo de que un fuego comience se reduce al mínimo si el CPD está situado lejos de cuartos de la planta y de almacenes de materiales inflamables, y no construido sobre áreas de estacionamiento de coches. Las paredes del CPD deben tener un grado mínimo de resistencia al fuego de una hora (RF-60) aunque se recomienda un grado RF-120, y deben proporcionar barrera frente al humo. Todas las puertas de acceso deben tener una ventana con cierre propio.

Todos los materiales usados en la construcción de la sala de ordenadores deben ser incombustibles. Para controlar el daño por agua, todas las entradas del piso, de la pared y del techo deben estar selladas.

Los CPDs necesitan sistemas contra incendios así como preventivos. Mientras no haya un sistema ideal, los sistemas por aspersión que se activan por dos detectores son una buena elección. Si se instala este tipo de sistemas por aspersión, hay que evitar que en caso de incendio el agua caiga directamente sobre los equipos electrónicos y que los sistemas de circulación de energía y de aire en las áreas afectadas permanezcan abiertos. Deben proporcionarse extractores de gas y desagües.

Los extintores manuales contra el fuego deben ser de dióxido de carbono u otros gases con agentes de extinción. No debe haber componentes químicos de extinción por polvo seco en el área de ordenadores.

Exigencias de comportamiento al fuego.

Hay que tener en cuenta que las condiciones de reacción al fuego aplicables a elementos constructivos se justificarán:

- Mediante la clase que figura en cada caso, en primer lugar, conforme a la nueva clasificación europea.
- Mediante la clase que figura en segundo lugar entre paréntesis, conforme a la clasificación que establece la norma UNE-23727.
- Los productos deberán acreditar su clase de reacción al fuego conforme a la normativa 23727:1990 mediante un sistema de evaluación de la conformidad equivalente al correspondiente al del marcado CE que les sea aplicable.

9. Requisitos y necesidades.

Son varios los requisitos que debe cumplir un CPD:

a. Tipo de instalación.

Instalaciones de alto riesgo: Las instalaciones de alto riesgo tienen las siguientes características:

- Datos o programas que contienen información confidencial de interés nacional o que poseen un valor competitivo alto en el mercado.
- Pérdida potencial considerable para la institución y, en consecuencia, una amenaza potencial alta para su subsistencia.

Todas las instalaciones de riesgo alto presentan una o más de esas características. Por ello, resultará generalmente fácil identificarlas. En cualquier caso, es evidente que, en el caso que nos ocupa, un CPD será considerado sin lugar a dudas como instalación de alto riesgo.

b. Disponibilidad y monitorización “24x 7x 365”.

Un centro de datos diseñado apropiadamente proporcionara disponibilidad, accesibilidad y confianza 24 horas al día, 7 días a la semana, 365 días al año.

c. Fiabilidad Infalible (5 ‘nueves’).

Es decir, con un 99,999% de disponibilidad, lo que traduce en una única hora de no disponibilidad al año. Los centros de datos deben tener redes y equipos altamente robustos y comprobados.

d. Seguridad, Redundancia y Diversificación.

Almacenaje exterior de datos, tomas de alimentación eléctrica totalmente independientes y de servicios de telecomunicaciones para la misma configuración, equilibrio de cargas, SAIs o Sistemas de Alimentación Ininterrumpida), control de acceso, etc.

e. Control ambiental / Prevención de Incendios.

El control del ambiente trata de la calidad del aire, temperatura, humedad inundación, electricidad, control de fuego, y por supuesto, acceso físico.

f. Acceso Internet y conectividad WAN.

Los centros de datos deben ser capaces de hacer frente a las mejoras y avances en los equipos, estándares y anchos de banda requeridos, pero sin dejar de ser manejables y fiables. Las comunicaciones dentro y fuera del centro de datos se proveen por enlaces WAN, CAN/MAN y LAN en una variedad de configuraciones dependiendo de las necesidades particulares de cada centro.

g. Rápido despliegue y reconfiguración.

Otros aspectos tratan de las previsiones para hacer frente a situaciones, críticas, con el objetivo de superarlas y volver rápidamente a la normalidad en caso de catástrofe.

h. Gestión continua del Negocio.

El funcionamiento de muchas compañías que constantemente realizan miles de transacciones por minuto gira entorno a la información almacenada. Para garantizar la fiabilidad existen los sistemas inteligentes de control de asignaciones y monitorización.

i. Cableado Flexible, Robusto y de Altas Prestaciones.

La infraestructura física de los centros debe soportar sistemas de comunicación de alta velocidad y altas prestaciones capaces de atender al tráfico de SANs (Storage Area Networks), NAS (Network Attached Storage), granjas de servidores de archivos/aplicación/Web, servidores Blade y otros dispositivos de almacenaje (Fibre channel, SCSI o NAS) así como Sistemas de Automatización del Edificio, sistemas de voz, video y CCTV.

En resumen, la infraestructura es la base de toda la actividad del centro de datos y desempeña un papel vital en la misma.

En este borrador de estándar de Infraestructura de Telecomunicaciones para Centros de Datos se han recogido una serie de topologías y medios de transmisión admitidos en base al estándar de cableado TIA-568-B:

ANSI/TIA/EIA-568-B .2-1 Categoría 6 (recomendada).

ANSI/TIA/EIA-568-B.3-1 Se recomienda Fibra Multimodo optimizada para Láser 0M3.

ANSI/TIAIEIA-568-B .3 Fibra Monomodo.

Una de las premisas recogidas por estos borradores indica que, para poder gestionar de forma satisfactoria incluso el centro de datos más sencillo, se requiere que la infraestructura de cableado sea flexible y soporte aplicaciones futuras.

Así mismo los diseños más avanzados y complejos demandan una infraestructura de cableado flexible, preparada para el futuro y para conseguir la meta del servicio 'sin fallos" en emplazamientos tan críticos donde una lentitud excesiva o un error .pueden comprometer la rentabilidad de una compañía.

También el comité europeo está abordando esta cuestión en el borrador EN 50173-5:200x: Tecnología de la Información — Sistemas de Cableado Genéricos, Parte 5: Centros de Datos.

10. Control de acceso.

Los locales que albergan los Activos Tecnológicos de los Sistemas de Información, requieren altas medidas de seguridad, que eviten acciones, malintencionadas o no, que puedan poner en peligro el "corazón" de la empresa.

Los sistemas antiintrusión mediante sensores de presencia, alarmas por rotura de vidrio, alarmas de puerta abierta y el control del local a través de elementos de monitorización de los accesos y equipos sensibles, son una infraestructura básica en el diseño de la seguridad física de este tipo de centros.

Se deben considerar:

- Central de monitorización.
- Teclado de acceso u otra medida.
- Detectores de movimiento infrarrojo pasivo/microondas.
- Sirena Interior.
- Analizador de inundación.
- Etiquetas de "Local Protegido".
- Conexión a Central Receptora de Alarmas.

11. Cableado voz y datos.

Un Sistema de Cableado Estructurado (SCE) se define en el entorno de un CPD como el conjunto de elementos, incluyendo paneles de terminación, módulos, conectores, cable, y latiguillos,

instalados y configurados para proporcionar conectividad principalmente de datos desde los repartidores designados hasta las rosetas o puntos de planta que dan servicio al equipamiento ubicado en el CPD (Host, dispositivos de almacenamiento, etc.).

Las aplicaciones estándar soportadas deben incluir, entre otras, IEEE 802.3, 1OBASE-T, lOOBase-TX, y 100BASE-FX, 1000BASE-SX, 1000BASE-LX. Además, los enlaces o canales deben ser capaces de soportar las aplicaciones emergentes de alta velocidad como 10 Gigabit Ethernet, 1OGBASE-SR, l000Base-T, 1000 Base-TX y ATM a 52/155/622/1000 Mbps, Fiber Channel, etc., pensando principalmente en los enlaces entre servidores y backbone.

La solución de cableado deberá considerarse, en cuanto a prestaciones, como un sistema en su conjunto, en lugar de considerar individualmente las prestaciones de cada uno de sus componentes. Este es un método de medida más útil al tener en cuenta la combinación de los componentes requeridos para llevar la señal desde la roseta o punto en planta hasta el armario de interconexión, de esta manera se garantiza la calidad de la señal total.

Es preciso asegurar el cumplimiento de la Categoría 6/Clase E con total certidumbre. Los equipos de test tienen un rango de exactitud, recogido en los estándares, en el que pueden dar un "Falso Positivo" o "Falso Negativo". Ver requisitos, procedimientos de test y fórmulas en ANSI/TIA/EIA 568-B .2 o consultar con un fabricante de equipos de test.

Para evitar obtener mediciones en el rango de incertidumbre, que pueden resultar incorrectas en varios dBs, es preciso disponer de canales de cableado con prestaciones superiores a lo recogido en el estándar, cuyas mediciones estén fuera del mencionado rango de incertidumbre en un entorno tan crítico como es un CPD.

Nunca se deben admitir en la definición de prestaciones los valores típicos o medios, ya que no aseguran el correcto funcionamiento del sistema instalado.

NOTA IMPORTANTE: sobre distancias cortas en conexiones de categoría 6

No es demasiado conocido el hecho de que las normas de cableado imponen a la longitud del canal, no sólo un máximo de 90 m, sino también un mínimo de 15 m para evitar los efectos de la energía reflejada.

Habitualmente, este requisito se cumple dejando una coca en los enlaces menores de 15 m hasta alcanzar dicha distancia. Sin embargo, este procedimiento no siempre es fácil de realizar y, en algunos casos, como las conexiones en CPDs o baterías de servidores, es casi imposible.

Por tanto, se requiere que el sistema de cableado estructurado esté diseñado y fabricado para evitar esta restricción de distancia mínima, es decir, que garantice prestaciones de Categoría 6 en cualquier conexión por corta que sea.

Las empresas se están dando cuenta de la necesidad de sistemas de cableado con más ancho de banda, con troncales que funcionan a 1 Gb y que deben soportar 10 Gb en un horizonte no muy lejano dada la cantidad creciente de datos manejada. Por otra parte, las soluciones más limitadas no ofrecen la estabilidad necesaria debido a su falta de margen de maniobra o de rendimiento, lo que

puede causar tiempos de no disponibilidad o provocar lentitud de las operaciones, resultando en pérdidas de productividad.

Estos sistemas deben satisfacer o superar los valores de prestaciones del Canal definido por el Estándar ISO 11801 2 a Edición, para los casos de canal de 4 conexiones e incluso para canales más exigentes que se dan en este tipo de entonos como canales de 6 conexiones (100 metros de canal con 4 o 6 conexiones, con latiguillos y punto de consolidación). Este punto resulta esencial en entornos que requieren de una gran flexibilidad y robustez.

En referencia a los problemas de los canales cortos, los Centros de Datos se ven forzados a almacenar más cable debido a la regla de los 15 metros ya que los conectores basados en los estándares pueden tener serios problemas de transmisión en situaciones de canales cortos debido al comportamiento inherente del NEXT; las conexiones superiores a los 15 metros enmascaran esta problemática gracias a las pérdidas de Inserción (Atenuación).

Como referencia, los canales NEXT basados en 2 conexiones próximas están recogidos por los estándares pero los canales NEXT basados en 3 conexiones próximas o más NO están recogidos por los estándares. Esta consideración se traduce en la posibilidad de fallo de canal NEXT para sistemas con prestaciones muy ajustadas a las indicadas por el estándar.

12. Inundaciones.

Según norma:

Deben de existir detectores de inundación con alarmas en varios sitios instaladas en sitios visibles del edificio, ya que la falta de estos detectores supone la existencia de un riesgo muy elevado de pérdida de equipos en caso de producirse una inundación.

No siempre es posible evitar conducciones de agua dentro de las salas destinadas a ordenadores o centros técnicos o de telecomunicaciones, incluso la instalación de los sistemas específicos de estos locales, implican tener conductos de agua en su interior.

Las fugas de fluidos, si no se descubren a tiempo, pueden causar daños en los equipos o pérdidas de información. Existen en el mercado analizadores de inundación que deberían ser evaluados.

13. Ubicación del CPD.

Generalmente, la instalación física de un Centro de Proceso de Datos exige tener en cuenta los siguientes puntos:

Local físico. Espacio disponible, acceso de equipos y personal, instalaciones de suministro eléctrico, acondicionamiento térmico y elementos de seguridad disponibles.

Espacio y movilidad. Altura y anchura del local, posición de las columnas, posibilidades de movilidad de los equipos, suelo móvil o suelo técnico, etc. Iluminación.

Tratamiento acústico. Los equipos ruidosos como las impresoras con impacto, equipos de aire acondicionado o equipos sujetos a una gran vibración, deben estar en zonas donde tanto el ruido como la vibración se encuentren amortiguados.

Seguridad física del local. Se estudiará el sistema contra incendios, también se estudiará la protección contra inundaciones y otros peligros físicos que puedan afectar a la instalación.

Suministro eléctrico. El suministro eléctrico a un CPD, y en particular la alimentación de los equipos, debe hacerse con unas condiciones especiales, como la utilización de una línea independiente del resto de la instalación para evitar interferencias, con elementos de protección y seguridad específicos y siempre con sistemas de alimentación ininterrumpida SAI's (equipos electrógenos, instalación de baterías, etc.).

Factores inherentes a la localización.
Son aquellas condiciones del medio ambiente externo que rodean al local. Se dividen en:

Naturales. Se está expuesto a múltiples peligros cuya ocurrencia está fuera del control del hombre, como es el caso del frío, el calor, las lluvias, los terremotos y el peligro del terreno (como el hundimiento del piso).

Servicios. Líneas telefónicas, energía eléctrica, drenaje, facilidades de comunicación, antenas de comunicación y líneas para enlace radioeléctricas...

Seguridad. Se basa en que la zona sea tranquila, que no esté expuesta a riesgos de alto grado, que no sea un lugar desolado o desprotegido. También se debe prever que alrededor del edificio no existan fuentes que propicien incendios fácilmente. Se debe considerar también el peligro de inundación. Entre otros factores tenemos el vandalismo, el sabotaje y el terrorismo.

El CPD no debería estar contiguo a maquinaría pesada o almacenes con gas inflamable o nocivo. El espacio deberá estar protegido ante entornos peligrosos, especialmente inundaciones. Algunas ubicaciones presentan amenazas específicas:

- Ubicaciones cercanas a paredes exteriores, planta baja o salas de espera: pueden presentar problemas de vandalismo o sabotaje

- Sótanos: problemas de inundaciones debido a cañerías principales, sumideros o depósitos de agua.

- Última Planta: desastres aéreos, fuego

- Encima de estacionamientos de coches: fuego

Una buena ubicación son las plantas intermedias o ubicaciones centrales en entonos de campus.

El CPD deberá tener espacio suficiente para alojar todos los equipos de comunicaciones necesarios y espacio extra para poder realizar la mayoría de las ampliaciones sin interrumpir el funcionamiento normal. Debe evitarse la instalación de un CPD en áreas con fuentes de interferencia de radiofrecuencia, tales como transmisores de radio y estaciones de TV.

ANEXO 3 (Política de Seguridad).

La RFC 1244 se refiere a los distintos aspectos a tener en cuenta para la confección de la política de seguridad de una red. A través de este texto se tratará de llevar a la práctica los aspectos fundamentales de la misma e incluir la mecánica a seguir para la elaboración de las distintas actividades referidas a seguridad, no expresadas en la RFC.

Como introducción, fuera de lo que especifica la norma, se tratará de establecer una diferencia básica que se tiene en cuenta en Administración y Conducción. Al tratar todo tipo de problemas, se establece una gran diferencia entre el marco estratégico y el de planeamiento y ejecución. El marco estratégico es quién define las políticas a seguir en líneas generales, es el enfoque macro. Los elementos de Conducción y ejecución sí son los que efectivizan el detalle planificando y ejecutando las acciones.

Siguiendo este lineamiento es que a lo largo de este texto se tratará de diferenciar claramente la Política de seguridad (Estrategia) del Plan de seguridad (Ejecución).

Para iniciar esta actividad, es necesario entonces comprender que los responsables de la creación del plan y política de seguridad son los responsables de la toma de decisiones y el personal técnico que las llevarán a cabo. Una vez definidos todos sus pasos, pasarán también a ser responsables la totalidad de los usuarios de la Organización, por quienes pasan la masa de los puntos claves y deberán conocer sus derechos y obligaciones al respecto.

1. Política de seguridad:

El primer paso entonces es definir la estrategia que se desea para la seguridad de la Organización. Para esta actividad, el Directorio deberá tener en cuenta lo siguiente:

- Grado de exposición al que se desea llegar:
- Cantidad de información que se desea exponer:
- Metodología de trabajo en el sistema informático de la Organización:
- Grado de acercamiento con otras entidades:
- Importancia de la seguridad dentro de la Organización:
- Presupuesto que se desea invertir para esta tarea:
- Personal que se dedicará al tema:
- Grado de compromiso del más alto nivel:

Luego del análisis de cada uno de estos Ítem y con las pautas claras al respecto es cuando puede comenzar a elaborarse el Plan de seguridad, el cual deberá realimentar muchas de las decisiones tomadas en la Política, generando con esto un Feedback permanente, característico de todo proceso dinámico.

2. Plan de Seguridad:

2.1. Análisis de riesgo:

La primera actividad para la implementación del Plan es determinar que es lo que se necesita proteger y cómo hacerlo. Este es el proceso de analizar todos los riesgos y clasificarlos acorde a algún tipo de prioridad. Existen dos tipos de elementos que se deben identificar en este análisis:

2.1.1. Identificación de recursos:

Son los elementos físicos que se necesitan proteger, estos deben contener:

- Hardware: CPU, terminales, workstations, PC, discos, líneas de comunicaciones, Servidores, Hub, Switch, Router, etc.
- Software: Programas fuente y objeto, utilitarios, sistemas operativos, programas de comunicaciones, etc.
- Datos: Durante la ejecución, Almacenamiento en línea y fuera de línea, backups, registros de auditorías, bases de datos, información en tránsito.
- Personas: Usuarios, personal necesario para la ejecución de sistemas, programadores, etc.
- Documentación: De programas, de hardware, de sistemas, procedimientos.
- Auxiliares: papeles, formularios, medios magnéticos, CD, etc.

2.1.2. Identificación de actividades:

En estas se puede determinar qué potencial de pérdida puede existir.

- Accesos no autorizados: Autorización de empleo de cuentas de usuarios por otras personas, uso de recursos sin autorización.
- Desbloqueo de información: La modificación de permisos sobre recursos es el más común.
- Negación de servicio: Esta actividad se puede presentar de distintas formas y afectará a los distintos usuarios de manera diversa.

2.2. Lineamiento del plan:

Una vez analizados los riesgos es importante trazar los primeros lineamientos generales del plan, aquí se especificarán los problemas globales a considerar, en general estos son:

2.2.1. ¿Quién está autorizado a usar los recursos?

En este punto se inicia el análisis de los distintos niveles de acceso a recursos, dando el enfoque inicial a los futuros grupos de acceso a recursos.

2.2.2. ¿Cuál es el uso correcto de recursos?

Se trata aquí de especificar un guía de acceso a los diferentes tipos de usuarios, aclarando fehacientemente qué es lo correcto y lo incorrecto, definiendo cuáles son los límites de cada uno, deben quedar sumamente claras las responsabilidades de las acciones llevadas a cabo. Un detalle a tener en cuenta también es el alcance legal del copiado de Software, acorde a la política de licencias de la Organización

En redes con buena capacidad de administración, se puede fomentar la actividad de investigación de vulnerabilidades, esto quiere decir que usuarios autorizados pueden desarrollar actividades de "Hackers" para colaborar con la administración de seguridad, detectando fallas tempranamente. Si se desea llevar a cabo esta actividad, es imprescindible dedicar varios apartados del Plan para dejar claramente especificado que se debe y que no se debe hacer, hasta dónde se puede llegar y los procedimientos ante cada uno de los avances. En estos casos, una buena medida es aislar segmentos de red para estas tares, con la finalidad de poder testearlos e identificar los "propios de los ajenos"

2.2.3. ¿Quién está autorizado a crear usuarios y conceder accesos?

Si no se tiene control sobre quién autoriza los accesos, no se podrá controlar sobre quienes usan el sistema. Es una muy buena medida especificar procedimientos para la creación de cuentas y asegurarse que el personal que lo realiza conozca bien estas normas.

El garantizar el acceso a usuarios es una de las vulnerabilidades más grandes de un sistema. Un detalle importante a tener en cuenta es la metodología de selección de contraseñas (este tema se tratará más adelante)

Se plantea aquí uno de los puntos claves de seguridad:

¿Se centralizarán los accesos o existirán múltiples puntos?

Siempre cuanto más centralizado sean los mismos, más seguro será el sistema.

2.2.4. ¿Quiénes pueden tener privilegios administrativos?

Esta es una decisión de suma importancia, pues inevitablemente se deberá designar a un cierto grupo de personas para poseerlos.

2.2.5. ¿Cuáles son las responsabilidades de los administradores del sistema?

En particular se deben tener en cuenta los aspectos relacionados a la información propietaria de los distintos usuarios de la red como así también el análisis de tráfico o correo electrónico, el acceso a bases de datos, etc.

2.2.6. ¿Qué hacer con la información sensible?

Se planifican aquí las distintas estrategias de resguardo y recuperación de información en diferentes modos (Discos, tape, CD, etc.).

2.2.7. ¿Que sucede si el plan es violado?

Existen distintos tipos de violaciones al plan, cada una de las cuales deberá ser tratada de manera diferente, esta pueden ser por:

- Negligencia individual:
- Accidente:
- No haber sido correctamente informado de las medidas de seguridad:
- No entendimiento del plan:

Lo importante es la rápida reacción y la determinación de cómo y por qué se produjo. Se deberá determinar la respuesta a la violación.

2.2.8. Proceder ante incidentes:

Reiteramos que existen dos estrategias básicas a tener en cuenta ante un incidente de seguridad:

- Proteger y proceder: La premisa de esta es la preservación de los componentes del sistema, el gran problema es que si el intruso no pudo ser identificado, este podrá regresar por la misma puerta o por alguna otra.

  ¿Qué premisas se deben tener en cuenta para implementar esta estrategia?

  * Si los recursos no están bien protegidos.
  * Si existe un riesgo económico de magnitud al continuar la intrusión.
  * Si no existe la posibilidad de perseguir al intruso.
  * Si los usuarios no poseen conciencia de seguridad y sus recursos peligran.
  * Si los recursos no están claramente establecidos.

- Seguir y perseguir: Se permite al intruso continuar sus actividades hasta identificarlo y evidenciar las vulnerabilidades del sistema que fueron aprovechadas. Se requiere aquí conocimiento en el manejo de incidentes y herramientas adecuadas pues se está arriesgando demasiado. La gran ventaja de este proceder es que es la única forma eficiente de llegar a las causas del problema para que este no vuelva a repetirse.

¿Qué premisas se deben tener en cuenta para implementar esta estrategia?

* Si los recursos y sistemas están bien protegidos.
* Si se dispone de buenos backup.
* Si la frecuencia de ataques es considerable.
* Si el acceso de intrusos puede ser controlado.
* Si se posee la capacitación suficiente para enfrentar un ataque.
* Si existen contactos con otros organismos que puedan prestar apoyo ante ataques.
* Si existe soporte legal en la organización para responder ante estos casos.

2.2.9. Publicación del plan:

La última actividad que se debe considerara es cómo se difunde el plan a los usuarios del sistema, para esta actividad se deben tener en cuenta varios aspectos referidos al dinamismo de las actualizaciones, al carácter reservado del plan, al acuse recibo de su lectura, a las correcciones, al cumplimiento y control, etc.

3. Análisis de detalle:

Habiendo definido ya qué necesita ser protegido, qué es lo más importante y cuáles son sus prioridades, es el momento de diseñar **cómo** hacerlo. Esta actividad es la que se tratará en este punto.

3.1. Identificación de problemas reales:

Acorde a lo especificado en el análisis de riesgo, se comienzan ahora a determinar las vulnerabilidades reales del sistema.

3.1.1. Puntos de acceso:

Todo usuario de la organización para poder acceder a la misma deberá hacerlo a través de uno de una interfaz que conecte físicamente su estación de trabajo a la red. Se debe diferenciar aquí los accesos vía LAN, los cuales se realizarán en general a través de vínculos propios y "bajo cierto control físico" (Si se respeta la seguridad a nivel físico) por parte de la Organización; y por otro lado los accesos remotos a esta red que es desde donde puede ingresar en general un usuario ajeno a la Empresa.

Los puntos de acceso de la LAN deben quedar claramente especificados en los planos de red, incluidos en la documentación de red, dentro de esta carpeta se establecerán las

medidas de seguridad a los ductos, gabinetes de comunicaciones, locales de ubicación de hardware de comunicaciones, etc.

Los puntos de acceso WAN, pueden ser dial-up, punto a punto, multipunto, o a través de acceso a una red pública de datos. En este ítem se deberá detallar al máximo la totalidad de los accesos, sin dejar de considerar todos los equipos que poseen módem y lo emplean para salir a la red de telefonía pública, pues es aquí donde generalmente se abren puertas no tenidas en cuenta.

3.1.2. Configuración de sistemas:

Se deberá detallar aquí las distintas medidas adoptadas para la configuración de los sistemas, teniendo especialmente en cuenta aquellos detalles referidos a la transmisión de información y al acceso a recursos. Este punto es tenido en cuenta pues por defecto existe mucho software que viene por defecto con detalles de configuración que facilitan ciertas actividades para beneficio de los instaladores del mismo, como así también medidas que deben ser adoptadas para facilitar accesos o ruteos a determinados usuarios pero que pueden ser causa de vulnerabilidades. A continuación se detalla una lista de referencia:

- Tipos de servicios.
- Rutas en host.
- activación de tareas dinámicas (DHCP, WINS, RIP, OSPF, etc.).
- Cuentas invitado o Anonymous.
- Puertos abiertos.
- Protocolos de comunicaciones.
- Directorios con permisos de control total.
- Cuentas o contraseñas que no respetan los procedimientos normales.
- Relaciones de confianza.
- Fronteras.
- Puertas traseras.

3.1.3. Bug de software:

Todo Software posee bug, los cuáles provocan inconvenientes en los sistemas, muchos de estos son aprovechados para vulnerar medidas de seguridad. Al ser detectados por los fabricantes, van generando los parches necesarios a los mismos. La segunda causa de los ataques de seguridad (después de los usuarios internos) es provocada por estas falencias. Por lo tanto en este punto se debe especificar todas las actualizaciones de Software que fueron introducidas en el sistema, con el mayor grado de detalle posible. También se plantean los problemas descubiertos y aún no solucionados.

Aparece aquí una gran reflexión: MANTENERSE PERMANENTEMENTE ACTUALIZADO, es una de las herramientas más importantes que posee un Administrador de sistemas. Con sólo leer los diarios, aparecen cotidianamente evidencias de ataques

producidos por bug en sistemas que ya fueron resueltos pero que aún no fueron actualizados en esa Empresa, y como corresponde fue aprovechado por un intruso que seguramente sí está actualizado.

3.2. Medidas de protección:

3.2.1. Protección de recursos:

- Control sobre recursos: Se deben .definir qué tipos de recursos deben ser auditados y sobre estos qué detalles auditar. La regla básica es que si se desea auditar TODO, luego NADA se mira. Por lo tanto es sin duda más eficiente definir sólo lo fundamental (poco), y sobre esto sí incrementar el control.
- Estrategias múltiples de protección: Suele ser más seguro emplear varias medidas simples que pocas sofisticadas. Las combinaciones de medidas cruzadas son de común empleo en seguridad, se ponen en evidencia en las estrategias de resguardo de información o en el acceso a recursos con monitoreo y auditoría en simultáneo.

3.2.2. Seguridad física:

Si el acceso a las estaciones de trabajo, servidores, periféricos, dispositivos y canales de comunicaciones no es seguro, a partir de allí no se puede sustentar un plan de seguridad. Por lo tanto se debe aquí establecer la totalidad de las normas de seguridad en los accesos a cada uno de estos elementos.

3.2.3. Reconocimiento de actividad no autorizada:

Para esta actividad se pueden emplear distintas herramientas, muchas de estas ya vienen incorporadas con el software de los sistemas, y otras son adicionales. En este punto se deberá establecer el conjunto de ellas y regular su empleo.

3.2.3.1. Monitorización de los sistemas en uso:

Esta es la actividad de control sobre los distintos recursos, se deberá realizar en forma agendada y aleatoria, analizándola y luego guardando los registros.

La clave aquí son los registros que se hayan decidido establecer, **su revisión constante es la primera barrera de seguridad**, pues con ellos se determinará usuarios en horarios no frecuentes, reiteración de accesos negados, modificación de archivos y permisos, actividad no concordante, consulta o apertura de puertos de uso no común, archivos nuevos no conocidos, etc.

3.2.3.2. Analizadores de protocolos:

Estas herramientas permiten analizar el tráfico de un red, y por lo tanto desarmar todo su contenido. A través de estos se puede determinar direcciones fuente y destino tanto de hardware como de software, exceso de tráfico en la red, protocolos que se están empleando, tipo de información que circula, establecimiento y cierre de sesiones.

A través de este punto se dejará registrada la mecánica de trabajo de estas herramientas y los archivos de lo examinado, con las conclusiones obtenidas.

3.2.4. Comunicación del plan de seguridad:

Se debe definir una metodología de información permanente del plan de seguridad y sus actualizaciones y verificar su correcta interpretación en todos los niveles de la Organización.

3.2.4.1. Educación de usuarios:

Deben tener claro que es lo correcto y lo incorrecto en todos sus procederes, y a su vez cómo deben proteger sus propios recursos. Una actividad importante es el monitoreo de sus recursos, cuenta y contraseñas, pues un ataque común es tomar posesión de los recursos de un determinado usuario de la red, y hacer uso de sus privilegios. La persona más indicada para detectarlo es el mismo usuario, por notar cambios en sus propios archivos, performance del equipo, capacidad de disco, actividad en horarios diferentes, etc. Ante estas eventualidades, debe poder reconocerlas y tener perfectamente claro dónde informarlas.

3.2.4.2. Educación de administradores:

Dentro de una red no podrá existir en la práctica un sólo administrador, sino que esta actividad deberá ser implementada por distintas personas que desempeñarán tareas diferentes. Si bien poseerán muchos privilegios similares, no todos deben ostentarán los mismos permisos ni tendrán las mismas atribuciones. En base a los distintos grupos de administración que se definan, es aquí donde se debe instruir respecto al correcto uso de sus cuentas.

3.2.5. Procedimientos de resguardo y recuperación:

Nunca es suficiente el énfasis que se puede hacer sobre las medidas a adoptar para l resguardo de la información. Esta si bien puede ser contemplada dentro de otras actividades, es también una actividad de seguridad por excelencia, pues de esta depende la capacidad de restaurar cualquier información dañada o perdida.

En este apartada es donde se debe volcar todas las actividades que se llevan a cabo para el resguardo de la información y los registros de lo realizado, especificando la periodicidad (diario, semanal mensual), el tipo (Normal, copia, diferencial o incremental) y la información resguardada.

Existen muchos métodos para verificar la integridad de los backup, los cales deben realizarse pues guardar información corrupta, de nada sirve.

3.3. Recursos para prevención de ataques:

3.3.1. Conexiones de red, módems, routers, proxys y Firewalls:

Se deben detallar aquí todas las implementaciones de barreras físicas colocadas y sus reglas de control. Se analizará cada dispositivo en cada una de sus interfaces, confeccionando un cuadro con lo que está permitido y denegado en cada una de ellas. El mismo deberá coincidir con lo configurado en estos dispositivos, y es una de las metodologías de control cruzado, la comparación de este documento con la realidad.

3.3.2. Confidencialidad:

La confidencialidad es la acción de restringir el acceso a la información a ciertas categorías de usuarios. Se presentan tres puntos en los cuales la información puede perder esta cualidad:

- Cuando la información está almacenada sobre un host.
- Cuando la información está en tránsito.
- Cuando la información se encuentra almacenada en dispositivos de backup.

Por lo tanto es necesario centrar la atención en este tipo de información acorde a la clasificación que se la haya impuesto, y especificar aquí todos los detalles.

3.3.2.1. Criptografía:

Esta actividad consta en convertir información interpretable, a un formato bajo el cual no se la pueda interpretar. Existen distintas formas de realizarla, tanto por software como por hardware, y se debe prestar especial atención, justamente sobre la que se encuentra en tránsito que es dónde en general presenta más flancos.

Se deben especificar aquí las técnicas empleadas y en que momento se las emplea.

Un detalle común en casi todas las recomendaciones de seguridad es NO DEJAR ESCRITO LAS CLAVES. Este último punto es común para contraseñas de usuarios, recursos o claves públicas y privadas de criptografía.

3.3.2.2. Privacidad en el correo electrónico:

El correo electrónico tiene la característica de transferir información como texto puro. Por lo tanto es común en las distintas organizaciones, separar el correo interno del de Internet. Esta actividad es aconsejable realizarla a través de distintos servidores, los cuales se deben encontrar en zonas de distinta clasificación de seguridad. Si bien la integración o sincronización de los mismos es llevada a cabo, se debe tener muy especialmente en cuenta que el correo interno viajará por vínculos propios mientras que el de Internet dará

la vuelta al mundo. Este detalle hace que el tipo de información que se maneje en cada uno de ellos sea diferente.

En ambos casos, acorde al tipo de Organización, se puede implementar privacidad en la transferencia de correo electrónico. Existen varios productos para esta tarea e inclusive también una serie de RFC (1113, 1114 y 1115) que proponen un estándar para privacidad en correo electrónico.

3.3.3. Autenticación:

Aquí se trata de garantizar que "quien dice ser, realmente lo sea". El sistema primario es a través de la creación de la cuenta de usuario con su contraseña correspondiente. En un sistema seguro, en especial al tratar las cuentas de acceso es conveniente ampliar esta medida a través de algún mecanismo adicional de autenticación. Existen de varios tipos, a continuación se detallan algunas posibilidades:

- Kerberos: Fue desarrollado por el MIT y emplea una combinación de criptografía y comparación en una base de datos distribuida, incrementando las medidas de autenticación/
- Tarjetas Inteligentes: Estos dispositivos poseen una clave que va cambiando permanentemente acorde a una secuencia pseudoaleatoria que se encuentra sincronizada con el servidor de acceso, y al coincidir las mismas, autentica al usuario.

Si se emplea algún método adicional, se debe aclarar aquí, acorde a la metodología que se haya definido.

3.3.4. Integridad de la Información:

La Integridad de la Información se refiere al estado completo, correcto y sin cambios desde la última vez que haya sido verificada. Esta actividad se lleva a cabo mediante el control de accesos sobre la misma. La masa de los sistemas permiten llevar registros sobre el acceso a la información y realizar las comparaciones pertinentes.

Esta es la actividad que se debe detallar aquí, el análisis de estos registros y las conclusiones obtenidas.

3.3.5. Fuentes de información:

Como mantenerse actualizado es la medida más importante a tener en cuenta, en este apartado se mencionarán las distintas opciones que se pueden consultar y las relaciones que hayan sido establecidas con el grado de participación logrado, detallando toda actividad desarrollada. Las opciones que se presentan a continuación son algunas de estas:

- Listas de correo: Permiten suscribirse y participar de noticias o debates sobre temas en particular.
- Equipos de repuesta: Son equipos que asesoran y recaban información sobre distintos incidentes referidos a seguridad.

- Vendedores: El soporte técnico sobre los productos adquiridos es parte de la actividad comercial de los productores de software y hardware, por lo tanto se debe tener bien claro dónde recurrir en caso de incidentes en los cuales la causa es identificada con un producto.

4. Procedimientos normales:

En este apartado se tratará de definir las distintas actividades en forma normalizada:

4.1. Actividades agendadas:

En este punto se debe realizar un calendario de actividades, detallando las tareas a realizar diariamente, semanalmente y mensualmente. Cuál es el objetivo de las mismas y contra qué confrontarlas para obtener conclusiones.

4.2. Test de procedimientos:

Se trata aquí de verificar el correcto funcionamiento del plan de seguridad, esta es uno de los puntos más dinámicos pues cotidianamente aparecerán nuevos empleos, desde la restauración de los backup, la creación de cuentas, verificar accesos, consultar usuarios, o realizar auditorías completas. Lo importante de este paso es anotar todo lo nuevo que se implemente, pues seguramente será reusado con posterioridad. Si se detectaran fallas, esto generará modificaciones al plan que realimentarán todo el proceso. Al lanzar algún tipo de test es importante poder definirlo unívocamente, para evitar confusiones acerca de la actividad que se está realizando, pues puede ser aprovechada o solapada con alguna intrusión real.

4.3. Procedimientos para la administración de cuentas:

La creación de las cuentas de usuarios es una tarea que cuánto más estandarizada esté, más eficiente será la organización de este servicio y más clara será la identificación de cualquier anomalía. Sobre esta actividad es importante considerar los siguientes aspectos:

- ¿Quiénes están autorizados a crear o modificar cuentas?
- ¿Quiénes pueden tener cuentas en el sistema?
- ¿Cuánto tiempo durará la asignación de una cuenta, y cómo se renegocia?
- ¿Cómo serán removidas y cuándo caducan las cuentas obsoletas?
- ¿Las cuentas se crean centralizadamente o se puede distribuir su administración?

- ¿Quiénes pueden crear o modificar grupos?
- ¿Quiénes pueden formar parte de los distintos grupos?
- ¿Quiénes pueden ser usuarios remotos?
- ¿Quiénes incrementan el nivel de validación? (En caso de existir)
- ¿Se restringirá el acceso por equipo, usuario, horarios, etc..?
- ¿Se permite a más de un usuario usar el mismo equipo?
- ¿Cuál es la lógica de nombres de cuentas?

4.4. Procedimientos para la administración de contraseñas:

De manera similar a la administración de cuentas, el tema de las contraseñas se debe tomar con cuidado, pues tratar de romperlas o crackearlas es una de las primeras actividades que desea realizar un intruso. Un buen test es ejecutar programas de Crack y luego informarle al usuario cuánto tiempo tardó en descubrir su contraseña, para que este sea consciente de la importancia que revista. Sobre esta actividad es importante considerar los siguientes aspectos:

- ¿Los usuarios pueden dar su contraseña a otros usuarios?
- ¿Cómo se implementa la contraseña inicial?
- ¿Tendrán fecha de caducidad?
- ¿Qué cantidad mínima de dígitos se permitirá?
- ¿Se guardará historia de cambios?, ¿cuántas?
- ¿Los usuarios pueden cambiar sus contraseñas?

4.4.1. Selección:

Guía de detalles a tener en cuenta: para la selección de una contraseña:

- NUNCA emplear las contraseñas por defecto.
- NUNCA dejar por escrito listas de contraseñas.
- NO USAR nombres de usuarios como contraseñas (ni en inverso, mayúsculas, duplicados, etc.).
- NO USAR nombres, apellidos, etc.
- NO USAR nombres de esposa/o, hijos, parientes cercanos.
- NO USAR información de fácil obtención, como ser : número de documento, fechas, teléfono, patente de automóvil, etc.
- NO USAR Contraseñas de todas letras o todos números, mucho menos repetición de los dígitos.
- NO USAR palabras contenidas en diccionarios.

- NO USAR contraseñas menores a 8 dígitos.
- USAR mezclas de números y letras.
- USAR caracteres de puntuación, matemáticos, lógicos, etc.
- USAR contraseñas fáciles de recordar.
- USAR contraseñas que se puedan escribir rápidamente sin mirar el teclado.

4.4.2. Cambios:

Un detalle a aclarar aquí es la metodología de verificación del usuario que solicita un cambio de contraseña, pues es inclusive un reporte de varios CERT el hecho de solicitar esta actividad para obtener acceso por parte de intrusos. Por lo tanto para esta actividad se deberán extremar las medidas de control.

5. Procedimientos ante incidentes:

En general este es un apartado al cual se le dedica muy poca atención y el resultado es que cuando se produce un incidente, las decisiones son tomadas sobre la marcha, provocando muchas veces daños por falta de previsión. Es por esta razón que se tiene en cuenta esta actividad, y se plantea el desarrollo del plan contra incidentes, el cual eliminará muchas ambigüedades.

Este plan será el resultado de todas las tareas realizadas anteriormente, es por esta razón que no se puede definir con anterioridad, ni puede apartarse de todas las regulaciones que ya fueron establecidas dentro de la Política y el Plan de seguridad.

Como referencia se detallan a continuación los aspectos que se deben considerar en el plan:

- Asegurar l a integridad de los sistemas críticos.
- Mantener y restaurar datos.
- Mantener y restaurar servicios.
- Determinar cómo sucedió.
- Detener escalamiento o futuros incidentes.
- Detener la publicidad negativa.
- Determinar quién lo hizo.
- Penalizar a los atacantes.

5.1. Plan contra incidentes:

La primera medida del plan consiste en la determinación de prioridades, las cuales se detallan a continuación como referencia:

- Prioridad 1: Proteger vidas o seguridad de personas.
- Prioridad 2: Proteger datos clasificados.
- Prioridad 3: Proteger otros datos.
- Prioridad 4: Prevenir daños a los sistemas.
- Prioridad 5: Minimizar anomalías en los sistemas.

5.2. Determinación del problema (Evaluación):

¿Es esto real?

Este es el primer interrogante, pues a menudo se puede confundir una intrusión con virus, falla de un sistema o un test que se está ejecutando. Existen varios indicadores que se pueden tener en cuenta, como por ejemplo:

- Ruptura de sistemas.
- Nuevas cuentas de usuarios, o actividad en cuentas que hace tiempo no se empleaban.
- Nuevos archivos, en general con extraños nombres.
- Discrepancia en cuentas respecto a lo establecido en el plan.
- Cambios en la longitud de los archivos o datos (en clientes, se pone de manifiesto en general por el crecimiento de archivos ".exe" desconocidos).
- Intentos de escritura en sistemas.
- Modificación o borrado de datos.
- Negación de servicios.
- Bajo rendimiento de sistemas, host o red.
- Numerosos intentos de validación.
- Numerosos intentos de inicio de sesión en puertos no habilitados.
- Nombres ajenos al sistema.
- Direcciones IP o MAC ajenas al sistema.
- Modificación de rutas en dispositivos de comunicaciones.
- Alarmas.

5.3. Alcance:

Se detallan aquí un conjunto de criterios que permiten delimitar el problema:

- ¿El incidente está acotado a este sitio o es multi-sitio?

- ¿Cuántos host están afectados?
- ¿Existe información sensible involucrada?
- ¿Cuál es el punto de entrada del incidente?
- ¿Tomó participación la prensa?
- ¿Cuál es el daño potencial del incidente?
- ¿Cuál es el tiempo estimado para solucionar el incidente?
- ¿Qué recursos serán requeridos para controlar el incidente?

5.4. Notificaciones:

Al saber fehacientemente que un incidente se ha provocado, se debe comenzar a notificar a aquellos que deban tomar participación en el hacho. Para mantener el hecho bajo control es importante saber a quiénes es necesario hacerlo. A continuación se tratarán ciertos aspectos que se deben tener en cuenta.

5.4.1. Información explícita:

Toda notificación que se curse dentro o fuera del sitio deberá ser explícita, esto quiere decir que la misma deberá ser clara, concisa y completa. El tratar de enmascarar el hecho o decir parte de la verdad, sólo sirve para crear más confusión.

5.4.2. Información verídica:

Si el hecho ya está difundido, el tratar de brindar explicaciones que no son estrictamente ciertas, sólo empeorará paso a paso el problema.

5.4.3. Elección del lenguaje:

La elección del lenguaje puede tener un efecto muy importante en las notificaciones. Si se usa un lenguaje emocional o inflamatorio, crecerán las expectativas sobre el incidente. Otro detalle del lenguaje son las expresiones no técnicas que se empleen para dirigirse a la masa del personal, pues es más difícil explicar hechos en este lenguaje pero es dónde realmente se están esperando las notificaciones.

5.4.4. Notificaciones a individuos:

Este último aspecto debe quedar definido para dejar claramente sentado a quien se debe notificar y por qué medios. Se estila contar aquí con una cadena de comunicaciones.

- Personal técnico.
- Administradores.

- Relaciones públicas.
- Personal directivo.
- Equipos de respuesta (CERT).
- Personal legal.
- Vendedores.
- Service Provider.

5.4.5. Aspectos generales a tener en cuenta par las notificaciones:

- Mantener un nivel técnico bajo. Si un alto grado de detalle es difundido, puede facilitar las actividades de intrusión.
- No difundir especulaciones.
- Trabaje con personal legal, para determinar qué evidencias deberán o no ser difundidas.
- Trate de no ser forzado a divulgar información antes de estar listo a brindarla.
- No permita que las presiones por brindar información desvíen el control del incidente.

5.5. Respuestas:

Este es el punto central del tratamiento de incidentes, la respuesta caerá en alguna o varias de los procedimientos que se detallan a continuación:

5.5.1. Contención:

Se trata de limitar la extensión del ataque. Varias medidas pueden quedar aquí establecidas, como apagar ciertos servidores, desactivar servicios, desconectar segmentos de red, activar rutas de contención, etc.

5.5.2. Erradicación:

Una vez contenido el incidente, es momento de erradicar las causas que lo provocaron. Detectar programas troyanos, virus, limpiar backup, etc.

5.5.3. Recuperación:

La recuperación consta de retornar el sistema a su estado normal. Para esta actividad se deberá instalar los parches correspondientes, recuperar la información dañada, restituir los servicios negados, etc.

5.5.4. Seguimiento:

Este que es uno de los procedimientos más importantes, es también el más dejado de lado. Este el punto de partida para luego desarrollar las "Lecciones Aprendidas". Se lo suele llamar el "Análisis Post mortem", se deben analizar los siguientes aspectos:

- Exactamente qué sucedió.
- ¿En que horario y fecha?
- ¿Cómo respondió el personal involucrado al incidente?
- ¿Qué clase de información se necesitó rápidamente?
- ¿Cómo se obtuvo esa información?
- ¿Qué se debería hacer diferente la próxima vez?
- ¿Cómo fue la cronología de eventos?
- ¿Que impacto monetario se estima que causó (Software, archivos, recursos, hardware, horas de personal, soporte técnico, etc.)?

5.6. Registros:

Al determinar un incidente, es imprescindible detallar todos los eventos posibles, por lo tanto se deben registrar con el mayor grado de precisión todos los pasos y acciones tomadas por personal propio y por intrusos. Como mínimo se debe registrar:

- Todos los eventos.
- Todas las acciones tomadas y detectadas.
- Todas las conversaciones telefónicas y notificaciones.

La mejor manera de realizarlo es llevar un libro de registros.

6. Procedimientos post incidentes

6.1. Introducción:

Luego de superado el incidente, es aconsejable realizar también una serie de actividades para permitir justamente la realimentación del plan de seguridad:

- Determinación final de los cómo fueron afectados los recursos.

- Las lecciones aprendidas deberán replantear el plan de seguridad.
- Un nuevo análisis de riesgo debería ser realizado.
- Si dentro del plan está contemplado, se deberá lanzar una investigación y tomar las medidas legales pertinentes.

6.2. Remover vulnerabilidades y depuración de sistemas:

Esta es una tarea muchas veces dificultosa, desde ya que es necesario haber podido determinar cuál fue la brecha, y es frecuente tener que remover todos los accesos o funcionalidades para restituirlas a su estado original. Si no se poseía líneas de base en la configuración de los sistemas, se incrementará el nivel de dificultad. Debería existir un plan de limpieza de los sistemas.

6.3. Lecciones aprendidas:

Basado en el seguimiento y registros realizados, es prudente escribir un reporte que describa el incidente, métodos de descubrimiento, procedimientos de recuperación, procedimientos de monitoreo, y por último el sumario de las lecciones aprendidas.

6.4 Actualización de políticas y planes:

Se dejará aquí asentado, los cambios que provocó este incidente en la Política y Plan de seguridad. Es de especial interés tener en cuenta que si el incidente se produjo por una pobre Política o Plan, a menos que estos sean modificados, seguramente se repetirá.

ANEXO 4 (METODOLOGÍA Nessus – Snort)

(Nivel de Inmadurez de los NIDS {segunda parte})

Por: Alejandro Corletti (acorletti@darfe.es - acorletti@hotmail.com)
José Ignacio Bravo Vicente (jseigbv@yahoo.com)

1. Presentación:

El presente trabajo es la continuación natural del publicado anteriormente denominado ***"Nivel de Inmadurez de los NIDS",*** que se encuentra en varios sitios de Internet.

En el mismo se realizó la evaluación de algunos productos de detección de intrusiones y luego de una serie de mediciones y comparativas, se obtuvieron las siguientes conclusiones:

a. Disparidad en la detección de un mismo evento por distintos productos:
b. Ausencia de detección del no cumplimiento a lo establecido por las RFCs.
c. Faltas de desarrollos en el relevamiento del software y hardware de red.
d. Faltas de iniciativas sobre trabajo en reglas "Proactivas".
e. Estudiar muy en detalle qué IDS se ajusta mejor a la red de la empresa.

Luego de estos hechos se continuó avanzando sobre lo estudiado y se trató de idear una metodología de trabajo que permita mejorar estos aspectos, y poder optimizar la detección de intrusiones. Relacionado a cada conclusión se puede describir lo siguiente:

- Al punto a. (Disparidad en la detección de un mismo evento por distintos productos):

  Se optó por emplear dos tecnologías diferentes en cada zona y evaluar las respuestas de ambas.

- Al punto b. (Ausencia de detección del no cumplimiento a lo establecido por las RFCs):

  Se hizo evidente una notable mejoría en la respuesta a este punto a lo largo de las sucesivas actualizaciones, en particular con Snort y los protocolos más importantes de la familia TCP/IP, en concreto TCP, IP e ICMP.

- Al punto c. (Falta de desarrollos en el relevamiento del software y hardware de red.

  Este es uno de los puntos que más se analizó y dio como resultado gran parte de esta nueva metodología que se propone aquí.

- Al punto d. (Faltas de iniciativas sobre trabajo en reglas "Proactivas"):

Se comenzó a trabajar en este punto, desarrollando reglas que se ajusten a la propia red donde se instalen los sensores, de forma tal que permita evaluar el tráfico cotidiano y lo anómalo.

- Al punto e. (Estudiar muy en detalle qué IDS se ajusta mejor a la red de la empresa):

  Este es un punto que se mantiene bastante claro y que continua respondiendo a la capacidad del personal de red que se posea, a su inclinación a ciertas líneas de productos, al nivel de desarrollo en entornos GNU, a los recursos materiales, a la magnitud de la red, el soporte técnico, el grado de exposición e impacto de sus recursos, etc.

Los ítems de las conclusiones que pueden realmente aportar novedades en este artículo, van muy relacionados al avance que se produjo sobre los puntos "c", "d" y "e", que es donde se centra esta propuesta.

2. Introducción:

Como continuación de la evaluación anterior, se propuso el desafío de ingeniar una metodología de trabajo, que permita "Madurar", estas falencias detectadas, pues sí se creía importante implementar la tecnología NIDS, ya era evidente que se trataba de un eslabón fundamental en la cadena de la seguridad, realidad de la que hoy son conscientes la masa de los administradores serios de sistemas, y que sin duda es casi indispensable, pues se repite una vez más, que los NIDS no compiten con los Firewalls ni con los routers, sino que colaboran como un dispositivo más en el plan de seguridad.

Ante este desafío, surgieron una serie de ideas y trabajos, de los cuales algo quedó en el camino, y otros se fueron encadenando permitiendo llegar a la combinación de dos productos que se aprecia son de muy buena calidad (para no entrar en discusión si son los mejores o no en sus rubros), ambos GNU. Se trata de **Nessus** como detector de vulnerabilidades (o generador de ataques) y **Snort** como Netwok Intrusion detection System (NIDS), empleando este último con todo el conjunto de módulos adicionales que permiten desarrollar cualquier función o servicio igual o mejor que cualquier otro producto comercial.

3. Metodología:

Al hacerse evidente que ningún NIDS detectaba la totalidad de los ataques que se realizan hacia una red, se propuso la idea de determinar cuáles eran detectados y cuáles no. Para esta tarea se evaluaron las distintas herramientas que permiten detectar vulnerabilidades en sistemas. Se trabajó con varias, y nuevamente otra propuesta GNU (Nessus) quedó excelentemente posicionada, una vez

más no se entrará en la discusión si es la mejor o no, pero sí se afirma que es muy buena y sus reglas son las que se actualizan con más frecuencia en el mercado.

Al empezar a detectar vulnerabilidades con Nessus, en los sistemas propios desde Internet hacia las zonas desmilitarizadas (DMZ), se comenzó a tabular las vulnerabilidades con los eventos detectados por Snort. Aquí aparece el primer problema (ya detectado en el trabajo anterior), pues no es fácil relacionar un ataque de Nessus con un evento detectado por Snort, esto se debe a que ambos tienen diferente codificación y denominación del evento. Esta relación sólo se puede realizar cuando el mismo posee referencia hacia CVE o Bugtraq, hecho que aparece en un muy bajo porcentaje de reglas, tanto de Nessus como de Snort, y aún así existen ataques que responden a una misma CVE y eventos detectados por Snort en los que sucede lo mismo, por lo tanto, ni siquiera en estos casos, se trata de una relación unívoca entre un ataque y un evento detectado por el sensor.

Para comenzar a relacionar estos dos hechos se trabajó de la siguiente forma:

a. Se aisló en laboratorio una pequeña red.
b. Se realizó un scan con Nessus.
c. Se estudiaron las vulnerabilidades detectadas.
d. Se analizaron las reglas de Nessus, (las cuáles se realizan mediante el lenguaje NASL, propuesto por este software, con el que existe una regla por cada ataque).
e. Se llegaba hasta la regla que generaba cada uno de los ataques detectados en el Laboratorio.
f. Se generaba únicamente este ataque, teniendo en cuenta aquí que Nessus, la mayoría de los ataques, solo los lanza si encuentra ese puerto abierto, es decir, si se trata por ejemplo de un ataque para detectar una vulnerabilidad en un servidor de correo, primero intentará establecer una conexión con el puerto TCP 25, y si esta no se establece, entonces no lanza el resto del ataque (que es lo que interesa capturar en este análisis), sólo lanzará la totalidad del ataque programado por su regla correspondiente (xxxx.nasl) si se cumple esta primera condición. Esta táctica la emplea para mejorar su rendimiento, pues no tendría sentido seguir generando tráfico sobre un puerto inexistente y por lo tanto hacia un servicio que no se está prestando. Por lo expresado entonces, se tuvo que instalar cada uno de los servicios que se deseaba evaluar.
g. Se capturaba la totalidad del ataque con un analizador de protocolos.
h. Se evaluaba si Snort lo detectaba o no.

Hasta aquí fue la tarea de tabulación e individualización de cada una de las vulnerabilidades presentes en este laboratorio, dejando en una hoja de cálculo qué ID de Nessus se correspondía con cuál ID de Snort (junto con otros datos adicionales). Al finalizar la misma, se puso de manifiesto la posibilidad de seguir adelante comenzando a crear las reglas de Snort que permitan detectar aquellas que este NIDS "no veía", y así siguió este trabajo.

Un comentario adicional se debe realizar aquí, pues cualquiera puede plantearse el tema de las vulnerabilidades existentes en una red, bajo la idea que si existe una vulnerabilidad, entonces hay que solucionarla, dejando ese servicio, host, puerto o sistema asegurado respecto a este evento. En

realidad esto no es tan fácil de realizar pues en muchos casos simplemente NO SE PUEDE, pues hay muchas causas que no permiten hacerlo, por ejemplo:

- Parches no existentes.
- Sistemas que no lo permiten.
- Aplicaciones propietarias que al modificar puertos o protocolos dejan de funcionar.
- Software enlatado que no se puede tocar.
- Servicios que fueron siendo modificados a lo largo de los años, y se hace muy peligroso de parchear.
- Sistemas que al ser bastionados dejan de funcionar, y no se está muy seguro de por qué.
- Servicios que no pueden dejar de prestarse.
- Accesos que deben ser abiertos sí o sí.
- Políticas empresariales, que no permiten asegurar esa vulnerabilidad.
- Dependencias de sistemas ajenos al organismo de seguridad.
- Etc., etc., etc., ...

Como conclusión a este comentario adicional entonces, se puede decir (y así nos ha sucedido en muchos casos), que si existe una vulnerabilidad detectada, la secuencia lógica es:

a. Ser consciente que se puede detectar o no.
b. Si es posible, solucionarla (No siempre se podrá).
c. Si no se puede solucionar, asegurar que **sí o sí** será detectada por los sensores.
d. Al ser detectada en la red, entonces **se trata de una alerta crítica** (Pues se sabe fehacientemente que se es vulnerable).
e. Generar una alarma en línea.

Basado en el manual de Snort se continuó realizando las reglas que permitieran detectar los ataques, tomando como referencia la regla de Nessus correspondiente (xxxx.nasl) y lo capturado con el analizador de protocolos. Cada nueva regla creada, se alojaba dentro de las local.rules de Snort, que están pensadas justamente para esta actividad y son tenidas en cuenta cada vez que se actualiza el conjunto de rules de Snort, pues las local.rules no son tocadas.

Se fueron generando nuevas reglas, teniendo como detalle adicional el incluir {nessus} dentro del mensaje de cada regla, para identificar posteriormente que se trataba de una detección procedente de un evento generado por este software.

El resultado final fue que se contaba con un NIDS que detectaba la totalidad de las vulnerabilidades presentes en este laboratorio. Este punto es de suma importancia, pues si bien con este aporte no se está totalmente seguro del 100 % de detección, pues siempre existen y existirán ataques que no son contemplados por Nessus, se pudo verificar que cubre un altísimo porcentaje de anomalías, y a su vez, también se hizo evidente la velocidad de actualización de plugins por parte de Nessus en

cuanto aparece una nueva vulnerabilidad y/o exploit en Internet, cosa que no sucedía con otros productos. Este aspecto proporcionaba un avance significativo a lo estudiado con anterioridad y que dio origen al artículo *"Nivel de Inmadurez de los NIDS"*, pues ahora se puede trabajar con sensores que se ajusten a la red en particular y garanticen una alta confiabilidad en la detección de eventos.

Al llegar aquí en el laboratorio, apareció una nueva forma de trabajar con NIDS.

¿Por qué no realizar la misma tarea en la red en producción?, mezclando la tecnología Nessus – Snort, para incrementar el nivel de seguridad del sistema. Es decir, en este punto se contaba con una metodología de trabajo que permitía:

- Detectar vulnerabilidades reales en la propia red.
- Verificar si nuestros sensores las reconocían en los patrones de tráfico.
- Generar las reglas necesarias.
- De hacerse presente un ataque de este tipo en la red, se trata de un caso crítico, pues ya se sabe que se es vulnerable en ese punto.

4. Mediciones:

Se presenta a continuación ejemplos de este trabajo:

a. La tabla que se presenta a continuación, es un resumen de la que se emplea para codificar (Asociar) eventos entre Nessus y Snort. En la misma se contemplan una serie de columnas que son las que proporcionan la información suficiente para aplicar esta metodología, que en definitiva el resultado final es obtener los Id de Nessus y de Snort en una misma línea, que son los datos que después permitirán la detección de un ataque que se sabe que en la red existe como vulnerabilidad.

   La descripción de los campos es:

   - Id Plugin (Nessus): Número que identifica unívocamente a ese ataque.
   - Name (Nessus): Nombre que aparece en la pantalla del cliente de Nessus.
   - Risk (Nessus): Riesgo que le asigna Nessus (no necesariamente en la red que se está analizando tendrá el mismo valor o impacto).
   - Nasl (Nessus): Regla específica que lanza ese ataque.
   - Summary (Nessus): Descripción breve del ataque.

- Family (Nessus): Familia dentro de la cual esta asignado este ataque en la pantalla de cliente Nessus. Este dato se emplea para luego buscar ese ataque específico en la consola y poder filtrar para generar únicamente este y no otro.
- Snort name (Snort): Nombre con que se visualizará este evento en formato Log.
- Id_snort (Snort): Número que identifica unívocamente ese evento. El mismo, puede existir ya en las rules (Número menor a 1.000.000) o se trata de una regla generada para esta red, la cual está incluida en el conjunto de las local.rules (número mayor que 1.000.000).

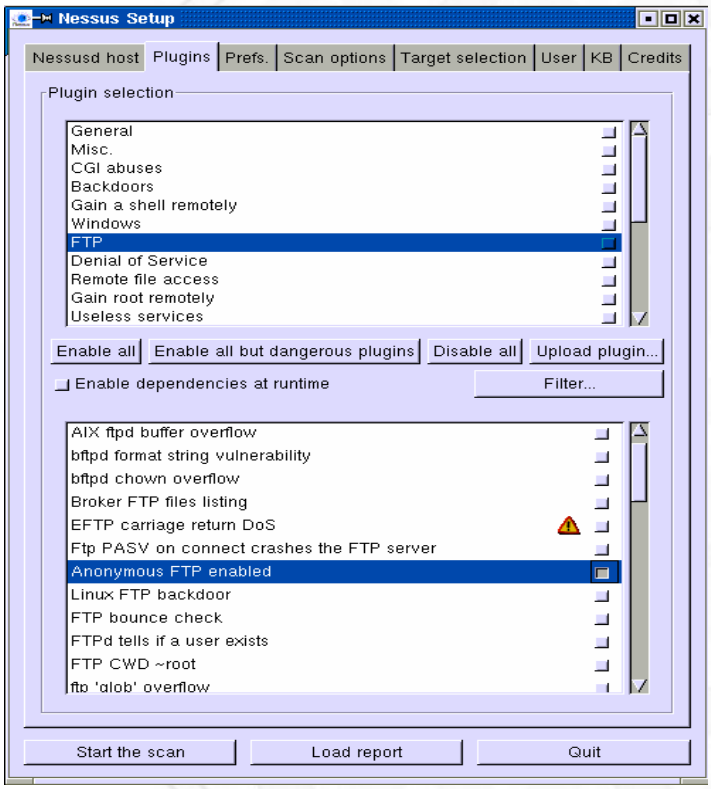
NESSUS						***SNORT***		***AMBOS***
id plugin	**Name**	**risk**	**nasl**	**summary**	**family**	**Snort name**	**Id_snort**	**referencias**
10012	Alibaba 2.0 buffer overflow	High	alibaba_overflow.nasl	Alibaba buffer overflow	Gain root remotely		1001019	CAN-2000-0626
10019	Ascend Kill	High	ascend_kill.nasl	Crashes an ascend router	Denial of Service	DOS Ascend Route	281	
10022	Axent Raptors DoS	High	axent_raptor_dos.nasl	Crashes an axent raptor	Denial of Service		1001021	CVE-1999-0905
10077	Microsoft Frontpage exploits	High	frontpage.nasl:	Checks for the presence of Microsoft Frontpage extensions	CGI abuses	WEB-FRONTPAGE _vti_rpc access	937	/www.securityfocus.com/bid/2144
10079	Anonymous FTP enabled	Low	ftp_anonymous.nasl	Checks if the remote ftp server accepts anonymous logins	FTP	Anonymous FTP enabled {nessus}	1001001	CAN-1999-0497
10080	Linux FTP backdoor	High	ftp_backdoor.nasl	Checks for the NULL ftpd backdoor	FTP		1001015	CAN-1999-0452
10088	Writeable FTP root	Serious	ftp_root.nasl	Attempts to write on the remote root dir	FTP		1001002 - 1001003	CAN-1999-0527
10096	rsh with null username	High	rsh_null.nasl	attempts to log in using rsh	Gain a shell remotely		1001022	CVE-1999-0180
10107	HTTP Server type and version	Low	http_version.nasl	HTTP Server type and version	General	WEB-IIS script access	1287	
10119	NT IIS Malformed HTTP Request Header DoS Vulnerability	High	iis_malformed_request.nasl	Performs a denial of service against IIS	Denial of Service		1001023	CVE-1999-0867
10150	Using NetBIOS to retrieve information from a Windows host	Medium	netbios_name_get.nasl	Using NetBIOS to retrieve information from a Windows host	Windows		1001016	
10160	Nortel Contivity DoS	Serious	nortel_cgiproc_dos.nasl	crashes the remote host	Denial of Service	WEB-CGI Nortel Contivity cgiporc DOS attempt	1763-1764	CAN-2000-0064
10161	rlogin -froot	High	rlogin_froot.nasl	Checks for rlogin -froot	Gain root remotely	RSERVICES rsh froot	604-609	CAN-1999-0113
10179	pimp	Serious	pimp.nasl	Crashes the remote host via IGMP overlap	Denial of Service		1001024	
10188	printenv	Medium	suse_cgi_bin_sdb.nasl:	Checks for the presence of /cgi-bin/printenv	CGI abuses	ATTACK RESPONSES 403 Forbidden - WEB-IIS scripts access	1201 - 1287	

b. Para ejemplificar, se toma una línea en concreto, en este caso la quinta:

10079	Anonymous FTP enabled	Low	ftp_anonymous.nasl	Checks if the remote ftp server accepts anonymous logins	FTP		1001001	

En principio se puede apreciar aquí que este evento no es detectado por Snort, pues se trata de una regla propia d esta red, por eso tiene asignado el ID 1001001.

c. Para llegar a esta codificación, primero se aisló desde Nessus, únicamente ese ataque, como se aprecia a continuación:



d. Se lanzó el ataque, el cual en este caso no es detectado por Snort (Hasta que no se cree la regla correspondiente). Se presenta a continuación la captura del mismo realizado con un analizador de protocolos:

```
1 11.566632 BILLIO5A59F1 0030050830C6 TCP ....S., len:    0, seq:4282991099-4282991099,
ack 10.64.130.195 10.64.130.14

2 11.566632 0030050830C6 BILLIO5A59F1 TCP .A..S., len:    0, seq:1559011737-1559011737,
ack 10.64.130.14 10.64.130.195

3 11.566632 BILLIO5A59F1 0030050830C6 TCP .A...., len:    0, seq:4282991100-4282991100,
ack 10.64.130.195 10.64.130.14
```

En estas 3 primeras tramas se está estableciendo la sesión TCP hacia el puerto 21 (FTP)

```
4 11.566632 0030050830C6 BILLIO5A59F1 FTP Resp. to Port 2582, '220 w2000rst Microsoft FTP
S 10.64.130.14 10.64.130.195
```

Se hace presente el servidor FTP `220 w2000rst Microsoft FTP`

```
5 11.566632 BILLIO5A59F1 0030050830C6 TCP .A...., len:    0, seq:4282991100-4282991100,
ack 10.64.130.195 10.64.130.14
```

Se envía el ACK correspondirente

```
6 11.576647 BILLIO5A59F1 0030050830C6 FTP Req. from Port 2582, 'USER anonymous'
10.64.130.195 10.64.130.14
+ Frame: Base frame properties
+ ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+ IP: ID = 0x458A; Proto = TCP; Len: 68
+ TCP: .AP..., len:   16, seq:4282991100-4282991116, ack:1559011789, win: 5840, src: 2582
dst:   21 (FTP)
+ FTP: Req. from Port 2582, 'USER anonymous'

00000:  00 30 05 08 30 C6 00 10 60 5A 59 F1 08 00 45 00   .0..0Æ..`ZYñ..E.
00010:  00 44 45 8A 40 00 40 06 DB D8 0A 40 82 C3 0A 40   .DEŠ@.@.ÛØ.@,Ã.@
00020:  82 0E 0A 16 00 15 FF 49 41 FC 5C EC A1 CD 80 18   ,.....ÿIAü\ì¡Í□.
00030:  16 D0 70 FF 00 00 01 01 08 0A 03 53 04 7D 00 43   .Ðpÿ.......S.}.C
00040:  FB 69 55 53 45 52 20 61 6E 6F 6E 79 6D 6F 75 73   ûiUSER anonymous
```

Aquí Nessus prueba si permite la validación como USER anónimo: **`USER anonymous`**

```
7 11.576647 0030050830C6 BILLIO5A59F1 FTP Resp. to Port 2582, '331 Anonymous access
allowed 10.64.130.14 10.64.130.195
```

Aquí se hace evidente que permite este acceso: `331 Anonymous access allowed`

```
8 11.576647 BILLIO5A59F1 0030050830C6 FTP Req. from Port 2582, 'PASS nessus@nessus.org'
10.64.130.195 10.64.130.14
+ Frame: Base frame properties
+ ETHERNET: ETYPE = 0x0800 : Protocol = IP:  DOD Internet Protocol
+ IP: ID = 0x458B; Proto = TCP; Len: 76
+ TCP: .AP..., len:   24, seq:4282991116-4282991140, ack:1559011861, win: 5840, src: 2582
dst:   21 (FTP)
+ FTP: Req. from Port 2582, 'PASS nessus@nessus.org'

00000:  00 30 05 08 30 C6 00 10 60 5A 59 F1 08 00 45 00   .0..0Æ..`ZYñ..E.
00010:  00 4C 45 8B 40 00 40 06 DB CF 0A 40 82 C3 0A 40   .LE‹@.@.ÛÏ.@,Ã.@
00020:  82 0E 0A 16 00 15 FF 49 42 0C 5C EC A2 15 80 18   ,.....ÿIB.\ì¢.□.
00030:  16 D0 E3 22 00 00 01 01 08 0A 03 53 04 7D 00 43   .Ðã".......S.}.C
00040:  FB 69 50 41 53 53 20 6E 65 73 73 75 73 40 6E 65   ûiPASS nessus@ne
```

Aquí Nessus prueba si permite cualquier contraseña: **`PASS`** nessus@nessus.org

```
9 11.606690 0030050830C6 BILLIO5A59F1 FTP Resp. to Port 2582, '230 Anonymous user logged
in 10.64.130.14 10.64.130.195
```

Aquí se hace evidente que permite este acceso: `Anonymous user logged in`

```
10 11.626719 BILLIO5A59F1 0030050830C6 TCP .A...F, len:    0, seq:4282991140-4282991140,
ack 10.64.130.195 10.64.130.14

11 11.626719 0030050830C6 BILLIO5A59F1 TCP .A...., len:    0, seq:1559011892-1559011892,
ack 10.64.130.14 10.64.130.195
```

En estas 2 últimas tramas se está cerrando la sesión TCP hacia el puerto 21 (FTP)

e. Si se desea se puede analizar la regla de Nessus que genera este ataque, la cual es: ftp_anonymous.nasl, que se presenta a contiinuación:

```
#
# This script was written by Renaud Deraison <deraison@cvs.nessus.org>
#
#
# See the Nessus Scripts License for details
#

if(description)
{
 script_id(10079);
 script_version ("$Revision: 1.23 $");
 script_cve_id("CAN-1999-0497");
 script_name(english:"Anonymous FTP enabled",
         francais:"FTP anonyme activé",
         portugues:"FTP anônimo habilitado");

 script_description(english:"The FTP service allows anonymous logins. If you do not
 want to share data with anyone you do not know, then you should deactivate
 the anonymous account, since it can only cause troubles.
 Under most Unix system, doing :
    echo ftp >> /etc/ftpusers
 will correct this.

 Risk factor : Low",
                 francais:"Le serveur FTP accepte les connections anonymes. Si vous
ne souhaitez pas partager des données avec des inconnus, alors vous devriez
désactiver le compte anonyme, car il ne peut que vous apporter des problèmes.
Sur la plupart des Unix, un simple :
    echo ftp >> /etc/ftpusers
corrigera ce problème.

Facteur de risque : Faible",
                         portugues:"O servidor FTP está permitindo login anônimo.
Se você não quer compartilhar dados com pessoas que você não conheça então você
deve
```

desativar a conta anonymous (ftp), já que ela pode lhe trazer apenas problemas.
Na maioria dos sistemas UNIX, fazendo:
    echo ftp >> /etc/ftpusers
irá corrigir o problema.

Fator de risco : Baixo");

 script_summary(english:"Checks if the remote ftp server accepts anonymous logins",
                francais:"Détermine si le serveur ftp distant accepte les logins anonymes",
                portugues:"Verifica se o servidor FTP remoto aceita login como
anonymous");

 script_category(ACT_GATHER_INFO);
 script_family(english:"FTP");
 script_family(francais:"FTP");
 script_family(portugues:"FTP");
 script_copyright(english:"This script is Copyright (C) 1999 Renaud Deraison",
                  francais:"Ce script est Copyright (C) 1999 Renaud Deraison",
                  portugues:"Este script é Copyright (C) 1999 Renaud Deraison");
 script_dependencie("find_service.nes", "logins.nasl", "smtp_settings.nasl");
 script_require_ports("Services/ftp", 21);
 exit(0);
}

#
The script code starts here :
#

port = get_kb_item("Services/ftp");
if(!port)port = 21;

state = get_port_state(port);
if(!state)exit(0);
soc = open_sock_tcp(port);
if(soc)
{
 domain = get_kb_item("Settings/third_party_domain");
 r = ftp_log_in(socket:soc, user:"anonymous", pass:string("nessus@", domain));
 if(r)
 {
  security_warning(port);
  set_kb_item(name:"ftp/anonymous", value:TRUE);
  user_password = get_kb_item("ftp/password");
  if(!user_password)
  {
   set_kb_item(name:"ftp/login", value:"anonymous");
   set_kb_item(name:"ftp/password", value:string("nessus@", domain));

```
 }
 }
 close(soc);
}
```

f. Y por último sólo quedaría crear la regla correspondiente en Snort, la cual podría ser la siguiente:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"Anonymous FTP enabled
{nessus}"; flags:AP; content:"USER anonymous"; nocase; depth: 16;
reference:CVE, CAN-1999-0497; classtype:attempted-user; sid:1001001; rev:1;)
```

g. Este es un caso interesante de análisis pues como se puede apreciar en la ftp_anonymous.nasl, se trata de un ataque estandarizado por CVE como candidata: script_cve_id("CAN-1999-0497");

5. Propuesta:

Luego de todos los avances realizados en estos años gracias al estudio de estas herramientas, el trabajo fue decantando en forma natural hacia esta arquitectura donde todos sus componentes son GNU. Al llegar a este estado de la cuestión y por tratarse justamente de una metodología abierta, se consideró la posibilidad de ponerlo a disposición de quien quisiera emplearlo y/o colaborar a su vez con el mismo. En este punto hubo un total acuerdo pues la evolución de estas arquitecturas de dominio público, demuestran que la suma de personas que desinteresadamente desarrollan estos proyectos aportan mucho mayor beneficio que el egoísmo personal o empresarial en cerrarse para guardar el secreto de una investigación y lucrar con ella.

Con esta postura, es que se publica el presente artículo en Internet, en el cual se invita a todos aquellos que estén interesados en implementar arquitecturas confiables de NIDS, bajo software gratuito, ajustando los mismos a su red en particular y colaborando en la confección de nuevas reglas y codificaciones entre Nessus y Snort. Para aquellos interesados, la propuesta es la siguiente:

SITUACION:

a. Se trata de una metodología de trabajo que puede realizar un aporte muy importante en la tecnología NIDS, si se logra sumar personas que lo mantengan vivo, sin egoísmos y transmitiendo todas sus experiencias.

b. Ya se tomó contacto con Snort.org y otros organismos que colaboran con el entorno Linux.

c. Aún no se ha decidido dónde residirá definitivamente, por lo cual, en los mismos sitios en los cuales se encuentra publicado el artículo, se va a informar más adelante en qué páginas Web se continuará con la investigación. Sobre este punto aún no se desea tomar una decisión hasta evaluar todas las opciones.

d. Se propone: Crear un proyecto de normalización de trabajo con las herramientas Nessus-Snort, bajo el cual, se pueda identificar cada ataque con la detección del mismo unívocamente. Sería muy interesante la participación de mitre.org, nessus.org y Snort.org en el mismo

e. **Desarrollar este trabajo principalmente en un entorno Hispano** (demostrando la capacidad que se posee en estos lugares del mundo, poco manifiesta en Internet, y en ningún sitio que se refiera a Nessus o Snort). Este apartado no excluye a otras lenguas ni traducciones de todo aquel que desee sumarse en otro idioma.

f. Se plantea:
   - Familiarizarse con estos productos.
   - Instalarlos en cada entorno participante (en laboratorio o producción).
   - Analizar las propias vulnerabilidades.
   - Comenzar a emplear analizadores de protocolos o sniffers.
   - Evaluar lo detectado por los NIDS.
   - Tratar de identificar y codificar, ID de ataques con ID de Snort.
   - Aprender a crear reglas para Nessus (Con el lenguaje NASL, documentado en Nessus.org)
   - Aprender a crear reglas con Snort (Acorde al manual, los documentos y How To de Snort.org).
   - Colaborar con la investigación y aportar nuevas ideas, códigos, reglas, actualizaciones, etc.
   - Tener siempre presente la idea de estandarizar procedimientos, reglas y definiciones.

g. Todo lo desarrollado estará a disposición una vez decidido el alojamiento.

MARCO DE TRABAJO:

a. Comenzar a realizar parte de lo planteado en el apartado anterior.

b. Una vez decidido el alojamiento de este trabajo, se insertarán documentos similares a los presentados en las mediciones. Los mismos estarán divididos en:
   - Plantilla de codificación Nessus- Snort.
   - Reglas aportadas para Snort.
   - Reglas aportadas para Nessus.
   - Cualquier otro módulo que amplíe el trabajo de estas herramientas.

c. Tratar de normalizar los eventos hacia un estándar, pareciera ser el más adecuado el propuesto por CVE (www.mitre.org), con quienes se tomará contacto a su debido tiempo.

d. Toda esta actividad la implementará un responsable para evitar alteraciones no debidas, deseadas o no.

e. Generar un foro de discusión sobre este proyecto, donde se pueda dialogar sobre el tema.

f. El ámbito queda abierto para todo aquel que desee participar y esté dispuesto a cumplir con lo establecido en el software GNU.

g. Los plazos y etapas se presentarán al estar disponible el proyecto.

LISTADO DE ABREVIATURAS

3DES (triple DES, para criptografía)
ADSL (Asymmetric Digital Subscriber Line)
ADSs (Anomaly Detection Systems)
AfriNIC (Africa Numbers Internet Community)
AH (Authentication Header)
AID (Association Identity)
AP (Access Ponit)
APNIC (Asia Pacific Network Information Centre)
ARIN (American Registry for Internet Numbers)
ARQ (Allowed to ReQuest)
AS (Áreas de Seguridad)
AuC (Authentication Center)
BDC (Backup Domain Controller)
BER (Bit Error Rate)
BG (Border Gateway)
BIA (Business Impact Analysis)
BIND (Berkeley Internet Name Domain)
BPF (Berkeley Packet Filter)
BTS (Basic TransceiverStation)
BSD (Berkeley Software Distribution)
BSC (Basic Station Controler)
BSS (Basic Station System)
C^3I (Comando, Control, Comunicaciones e Informática)
CA (Autoridad de Certificación)
CBC (cifrado encadenado de bloques)
CG (Charging gateway)
CF (Contention-Free)
CGI (Common Gateway Interface)
CPD (Centro de Procesamiento de Datos)
CRC (Control de Redundancia Cíclica)

CRL (Certificate Revocation List)
CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)
CSMA/CD (Carrier Sence Multiple Access/Colition Detect)
CTS (Clear To Send)
DCF (Función de coordinación distribuida)
DES (Data Encryption Standard)
DIDS (Distributed Intrusion Detection System)
DMZ (zona desmilitarizada)
DoD (Departamento de Defensa de EEUU)
DRP (Disaster Recovery Plan)
DS (Distribution System)
DSS (Digital Standard Signature).
DSSS (Direct Sequence Spread Spectrum)
DTP (Data Transfer Process)
ECC (Elliptic Curve Cryptosystem)
ECDH (Elliptic Curve Diffie-Hellman Key Agreement)
ECDSA (Elliptic Curve Digital Signature Algorithm)
ECES (Elliptic Curve Encryption Scheme)
ECMQV (Elliptic Curve Menezes-Qu-Vanstone Key Agreement)
ECNRA (Elliptic Curve Nyberg-Rueppel Signature Scheme with Appendix)
EIFS (Extended IFS)
EIR (Equipment Identity Register)
ESP (Encapsulation Security Payload)
ESS (Extended Service Set)
ESSID (Extended SSID)
ETD (Equipo Terminal de Datos)
EUI (Extended Unique Identifier)
FAC (Final Assembly Code)
FCS (Frame Control Sequence)
FEC (Forward Error Control)
FHSS (Frequency Hopping Spread Spectrum)
FNMT (Fábrica Nacional de Monedas y Timbres)
FTP (File Transfer Protocol)

FW (Firewall)

GEA (GPRS Encryption Algorithm)

GGSN (Gateway GPRS Support Node)

GNSO (Generic Names Supporting Organization)

GnuPG o simplemente GPG (GNU Privacy Guard)

GPL (General Public License)

GPRS (General Packet Radio System)

GRE (General Routing Encapsulation)

GRX (GPRS Roaming Exchange)

GSM (Global System for Mobile Communications u originariamente del portugués: Groupe Spécial Mobile)

GTP (GPRS Tunneling Protocol)

HDLC (High Level Data Link Connection)

HDSL (High bit-rate DSL)

HEC (Header Error Check)

HIDS (Host Intrusion Detection System)

HLR (Home Location Register)

HMAC-SHA1 (Hash Message Autentication-Secure Hash Standard Versión 1)

HTTP (HiperText Transfer Protocol)

IANA (Internet Assigned Numbers Authority)

IBSS (Independent BSS)

ICANN (Internet Corporation for Assigned Names and Numbers)

IDP (Intrusion Detection Prevention)

IDS (Intrusion Detection System)

IEC (Comisión Internacional de Electrotécnia)

IKE (Internet Key Exchange)

IPS (Intrusión Prevention System)

IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)

IETF (Ineternet Engineering Task Force)

IFS (Interframe Space)

IIS (Internet Information Server)

IMAP 4 (Internet Message Access Protocol Versión 4)

IMEI (International Mobile Equipment Identity)

IMSI (International Mobile Suscriber Information)

IPNG (IP Next Generation)

IPv6 (IP versión 6) o IP Next Generation (IPNG)

IR (InfraRed)

ISAKMP (Internet Security Association and Key Management Protocol)

ISAPI (Internet Server Application Programming Interface)

ISDL (ISDN DSL)

ISECOM (Institute for Security and Open Methodologies)

ISM (Industrial, Scientific and Medical Band)

ISMS (Information Security Management System)

ISO (International Standard Organization)

ISP (Internet Service Provider: Proveedores de Servicio de Internet)

IV (Vector de Inicialización)

JTC 1 (Join Technical Committee Nº1).

JVM (máquina virtual Java)

L2TP (Layer 2 Tunneling Protocol)

LACNIC (Latin American and Caribean Internet Address Registry)

LANE (LAN Emulation)

LAP-M (Link Access Procedure - Modem)

LDAP (Lightweight Directory Access Protocol)

LIN (Lawful Interception Node)

LIR (Registro local de Internet)

LLC (Logical Link Control)

LMDS (Local Multipoint Distributed Signal)

LML (Licencias de Música Libre)

LOPD (Ley Orgánica de Protección de datos)

MAC (Medium access control)

MAC (código de autenticación de mensaje)

MCC (Mobile Country Code)

MD5 (Messge Digest Verión 5)

ME (Mobile Equipment)

MIB (Managment Information Base)

MIME (Multimedia Internet Mail Extensions)

MNC (Mobile Network Code)
MPLS (MultiProtocol Label Switching)
MS (Mobile Station)
MSIN (Mobile Suscriber Identity Number)
MTA (Agente de transferencia de mensajes: message transfer agent)
NAI (Network Associates Inc.)
NAPT (Network Address Port Translation)
NAS (Network Access Server)
NASL (Nessus Attack Scripting Lenguaje)
NAT (Network Address Translation)
NAV (Network Allocation Vector)
NDIS (Network drivers Interface Standard)
NetBIOS (Servicio Básico de Entradas y Salidas de red)
NIC (Network Information Center)
NIC (Network Interface Card)
NIDS (Network Intrusion Detection System)
NIR (Registro Nacional de Internet)
NMS (Network Managment Station)
NNTP: (Network News Transfer Protocol)
NTP (Network Time Protocol)
NVT (Network Virtual Terminal)
SKEME (Secure Key Exchange Mechanism for Internet)
ODI (Open Data Interface)
OpenVAS (Open Vulnerability Assessment System)
OSA (Open System Authentication)
OSI (Open System interconnection)
OSSTMM (Open Source Security Testing Methodology Manual)
OSVDB (Open Source Vulnerability DataBase)
OUI (Organizationally Unique Identifier)
PC (Point Coordinator)
PC (Personal Computer)
PCC (Proof-Carring Code)
PCF (Función de coordinación puntual)

PCN (Plan de Continuidad de Negocio)

PDC (controlador principal de dominio)

PDCA (Plan – Do – Check – Act)

PDP (Packet Data Protocol - Primary Domain Controller)

PDU (Protocol Data Unit)

PEM (Privacy Enhanced Mail)

PGP (Pretty Good Privacy)

PI (Protocol Interpreter)

PIN (Personal Identification Number)

PKCS (Public-Key Cryptography Standards)

PKI (Infraestructura de clave pública)

PKIX (para infraestructura de clave pública)

PLMN (Public Land Mobile Network)

PLW (Physical Length Word)

PMD (Physical Medium Depend)

PON (Procedimientos Operativos Normales)

POP (Post Office Protocol)

POS (Procedimiento Operativo de Seguridad)

PRN (Plan de Recuperación de Desastres)

PSF (Physical Signaling Rate)

QAM (Modulación en cuadratura de Fases con más de un nivel de amplitud).

RAC (Registration Authority Commitee)

RADIUS (Remote Authentication Dial-In User Server)

RADSL (Rate-Adaptive DSL)

RAND (Random Number)

RAVs (Risk Análisis Values)

RFC (Request For Commentaries)

RIPE (Réseaux IP Européens - en Francés: "IP para redes Europeas")

RIR (Registro Regional de Internet)

ROI (Retorno a la Inversión)

RSA (Rivest, Shamir and Aldeman).

RTS (Request To Send)

SA (Security Asociation)

SAD (Security Associaton Database)

SAM (Security Accounts Manager)

SAP (Service Acces Point)

SCE (Sistema de Cableado Estructurado)

SDSL (Single-Line DSL)

SET (Secure Electronic Translation)

SFD (Delimitador de inicio de trama)

SGSI (Sistema de Gestión de la Seguridad de la Información)

SGSN (Serving GPRS Support Node)

SHA (Secure Hash Standard)

SHA-1 (Standard Hash Algorithm Versión 1)

SID (identificadores de seguridad)

SIFS (Short IFS)

SIM Card (Subscriber Identity Module - módulo de identificación del suscriptor)

SM (Industrial, Scientific and Medical band)

SMB (Server Message Block: Servidor de Bloques de Mensajes)

SMTP (Simple Mail Transfer Protocol)

SNMP (Single Network Monitor Protocol)

SNR (Serial Number)

SPD (Security Policy Database)

SRES (Signed Responsed)

SSH (Secure SHell - intérprete de comandos seguro)

SSID (Service Set identifiers)

SSL (Secure Socket Layer)

TAC (Type Aproval Code)

TAP (Top Level Domain)

TCP (Transport Protocol Protocol)

TDI (Transport Driver Interface)

TIA/EIA (Telecommunications Industy Association/Electronics Industry Association)

TIM (Traffic Indication Map)

TLD (Top Level Domain)

TLS (Transport Layer Security)

TPV (Terminal de Punto de Venta)

TSF (Función Sincronización de Tiempo)

T_FTP (Trivial FTP)

UA (Agente de usuario: user agent)

UDP (User Datagram Protocol),

UIT-T (Unión Internacional de Telecomunicaciones - Sector Normalización de las Telecomunicaciones)

UKAS (United Kingdom Acreditation Service)

UMTS (Universal Mobile Telecommunications System)

URI (Uniform Resource Identifier - Identificador Uniforme de Recurso)

URL (Uniform Resource Locator)

USM (User-Based Security Model)

UTP (Unshield Twisted Pair)

VACM (View-Based Access Control Model: *Modelo de Control de Accesos basado en Vistas)*

VLR (Visitor Locator Register)

VPLS (Virtual Private LAN Segments)

VPN (Virtual Private Network o Red Privada Virtual)

WECA (Wireless Ethernet Compatibility Alliance)

WEP (Wired Equivalent Privacy)

WiMAX (Worldwide Interoperability for Microwave Access: Interoperabilidad mundial para acceso por microondas)

WINS (Windows Internet Name Service)