# Tema 5: Seguridad en Redes

Luis Enrique Corredera de Colsa – Flag Solutions S.L.
Fernando García Fernández – Flag Solutions S.L.

## 5.1 Tabla de contenidos

## 5.2 Introducción

En este capítulo aprenderemos las bases, la teoría y la práctica de los ataques más populares en redes de ordenadores, así como protegernos de ellos mediante cambios en las configuraciones, el uso de cortafuegos, sistemas de detección de intrusos y sistemas de cebo para entretener a los atacantes. En el apartado de defensa veremos la última tecnología, aún experimental para la detección de intrusos, basada en agentes inteligentes.

Veremos también la inseguridad que puede suponer el despliegue de redes inalámbricas sin la debida planificación, y qué mecanismos debemos usar para asegurar estas redes.

Entre los ataques que se explican tenemos las escuchas, las falsificaciones de diversos tipos, los secuestros de sesión y los ataques de denegación de servicio, tanto centralizados como distribuidos.

## 5.3 Ataques

### 5.3.1 Sniffing

#### *5.3.1.1 Introducción*

Cuando nos encontramos recabando información sobre sistemas remotos en una red, antes de aplicar técnicas de ataque, es muy recomendable conseguir toda la información posible acerca de la víctima. Una de las muchas técnicas que podemos usar para obtener dicha información es el *sniffing*, que descubriremos y analizaremos a lo largo de este tema.

*Sniff* es un vocablo que proviene de la lengua inglesa y podemos traducir al castellano como "husmear". Y en la práctica, al aplicar técnicas de sniffing, lo que haremos es precisamente eso: husmear. Escucharemos el tráfico de la red, lo guardaremos y posteriormente lo analizaremos para obtener información sobre nuestro objetivo.

En función del tiempo que empleemos en nuestras sesiones de sniffing, conseguiremos recolectar contraseñas de usuarios, correos electrónicos (tal vez confidenciales), conversaciones a través de servicios de mensajería (MSN, ICQ, Jabber, Yahoo,...), chats, etc.

Cabe señalar que hay básicamente dos formas de hacer sniffing:

- **Sniffing por software**: es la forma más habitual. Usa programas informáticos para capturar el tráfico en la red.
- **Sniffing por hardware**: es una forma más cara de hacerlo. Tendríamos que "pinchar" en un segmento de red un dispositivo que fuera capaz de almacenar el tráfico que escucha.

En este capítulo nos dedicaremos al sniffing por software, ya que es la opción que se encuentra más al alcance de cualquier mano.

#### *5.3.1.2 Condiciones de uso y limitaciones*

Una de las condiciones para poder practicar el sniffing es la cercanía o el acceso al medio de transmisión. Vayámonos unos años atrás en nuestras vidas, y recordemos los tiempos del colegio: cuando un compañero 'Juan' quiere mandar un mensaje a otra compañera 'Alicia', escrito en un papel arrugado, en ocasiones era necesario el uso de intermediarios para hacer llegar el papel a su destino. A los compañeros que participaban en el transporte nada les impedía leer el contenido del papel. Sin embargo los compañeros que no intervenían en el transporte no tenían posibilidad de leer el mensaje, ya que éste no pasaba por sus manos.

Volviendo a la vida en el momento actual, con el *sniffing* ocurre lo mismo que con el paso de mensajes de Alicia y Juan: podremos husmear en las comunicaciones que transcurran por un medio físico que podamos escuchar.

¿Podremos entonces hacer snfiffing de un paquete de datos que transcurre entre New York y Tokio? Seguramente no. Tendría que darse una situación bastante compleja para poder hacerlo.

Entonces, ¿para qué nos sirve el sniffing? Supongamos que nos encontramos en la red de nuestra empresa, y supongamos que los ordenadores se conectan a un hub ethernet. Todo el tráfico de la red viajará por nuestro cable, y tendremos oportunidad de husmear las comunicaciones.

Otro típico caso típico de uso del sniffer son las máquinas remotas comprometidas, donde nos aportarán información interesante sobre los servicios de la red.

El sniffing es una técnica compatible con otras, como diversos spoofing, obteniendo de la combinación unos muy buenos resultados.

No obstante el uso de sniffers no es exclusivo de individuos con malas intenciones. Aplicaciones como tcpdump (o windump) son muy usadas por los administradores de redes.

*5.3.1.3 Arquitectura típica de un sniffer*

La explicación de la utilidad de un sniffer nos abre un bonito mundo de posibilidades tanto para husmear con fines didácticos como para auditar situaciones extrañas en nuestras redes. Sin embargo, si pretendemos exprimir las capacidades de un sniffer es necesario conocer como funcionan por dentro y como ampliar las capacidades de extensión que muchos de ellos proporcionan.

Como el estudio de la implementación de un sniffer se escapa de los objetivos de este documento, veremos sucintamente la arquitectura de un sniffer genérico, que presentaría las siguientes capas:

- **El hardware:** La mayoría de productos funcionan sobre adaptadores de red estándar, aunque algunos requieren hardware especial. Con analizadores hardware especializados, es posible comprobar fallos como errores de CRC, problemas de voltaje, errores de negociación, etc.

- **Driver de captura:** Esta es la parte más importante. Captura el tráfico de red del cable y lo almacena en un buffer. Para minimizar el consumo de recursos en muchas aplicaciones se permite especificar filtros para guardar en el buffer solamente los paquetes que son de nuestro interés.
- Para poder llevar a cabo la captura indiscriminada de los paquetes de datos es imperativo que la interfaz de red soporte la operación en modo promiscuo. Prácticamente todas los adaptadores ethernet lo soportan pero, por ejemplo, muchos dispositivos Token Ring o 802.11x no lo permiten.

- **Buffer:** Una vez los paquetes son capturados de la red, estos se almacenan en un buffer. Existen un par de modos de captura: hasta que el buffer se llene, o usar un buffer rotatorio donde los nuevos datos sobrescriben los más antiguos. Algunos productos como BlackICE Sentir IDS, de Internet Security Systems pueden mantener un buffer rotativo de captura en disco capaz de operar a 100 mbps. Esto permite tener cientos de GB de buffer en lugar de estar limitado por la cantidad de memoria del equipo.

- **Análisis en tiempo real:** Esta característica realiza algunos análisis a nivel de bits de los paquetes que atraviesan el cable. Esto permite encontrar fallos de eficiencia en la red mientras continua capturando. Algunos vendedores han extendido estas funcionalidades en algunos de sus productos para pasar a ser IDS, aunque aún queda mucho trabajo por hacer en las herramientas de detección de intrusiones.

- **Decodificación:** esta característica transforma los datos binarios a un formato entendible para su posterior análisis. En muchos casos resulta interesante filtrar un contenido específico definido por el usuario.

- **Editar paquetes (transmitir)**: algunos productos (los menos) contienen características de editar los paquetes en la propia red y enviarlos de nuevo a ella. Es decir pueden generar paquetes personalizados usados con fines muy definidos. Algunos intrusos usarán estas herramientas para realizar ataques man-in-the-middle y secuestro de sesiones mediante las cuales intervienen un tráfico y suplantan la identidad original del individuo (puede ser temible).

#### *5.3.1.4 Servicios vulnerables*

A priori son objetivo de este tipo de ataque todos los servicios que no usen cifrado en las comunicaciones, o bien no se lleven a cabo a través de un canal cifrado punto a punto. En el momento en que una parte de la comunicación transcurra por un tramo no cifrado, el servicio es vulnerable.
Entre los servicios que pueden ser atacados con mayor impacto están:
- Telnet
- FTP
- POP3
- SMTP
- …

Aunque en muchos casos los administradores de red confían en el uso de switches para contrarrestar los ataques con sniffers, esta medida es muy pobre. Como hemos visto en el tema anterior, podemos engañar fácilmente a los routers con envenenamiento de ARP.

#### *5.3.1.5 Sniffers comerciales*

Los programas para realizar sniffing se han distribuido de dos formas diferenciadas: comerciales y gratuitos o libres (en algunas literaturas se refieren a ellos como 'underground').
Típicamente los sniffers comerciales han sido usados para mantener redes, y los sniffers "underground" para asaltar a los ordenadores de una red. Hoy día, gracias a la gran evolución de las comunidades de desarrolladores de software libre las diferencias entre los sniffers comerciales y no comerciales están muy limadas, llegando en muchos casos a superar los no comerciales a los comerciales.

Entre los usos típicos de un sniffer incluyen los siguientes:
- Captura automática de contraseñas enviadas en claro y nombres de usuario de la red. Esta capacidad es utilizada en muchas ocasiones por intrusos para atacar sistemas a posteriori.
- Conversión del tráfico de red en un formato entendible por los humanos.

Análisis de fallos para descubrir problemas en la red, tales como: ¿por qué el ordenador A no puede establecer una comunicación con el ordenador B?

Medición del tráfico, mediante el cual es posible descubrir cuellos de botella en algún lugar de la red.
Detección de intrusos. Aunque para ello existen programas específicos llamados IDS (Intrusión Detection System, Sistema de Detección de intrusos), estos son prácticamente sniffers con funcionalidades específicas.
Creación de registros de red, de modo que un intruso no pueda detectar que está siendo auditado.

Sentados los precedentes, podemos pasar a analizar algunos sniffers. Aunque el título de esta sección presenta "sniffers comerciales", no podemos hacer el avestruz e ignorar las maravillas que tenemos a nuestro alcance gracias al software libre, por lo que nos pararemos a ver otros sniffers no comerciales, pero muy efectivos.

##### 5.3.1.5.1 Tcpdump

Tcpdump es un sniffer que captura el tráfico de la red y después permite realizar análisis sobre los datos obtenidos. La página web oficial es http://www.tcpdump.org, de donde podremos obtener la última versión, y una amplia documentación sobre su funcionamiento

Tcpdump está basado en la librería libcap, que proporciona rutinas para hacer las lecturas de la red y también sirve de base para otros muchos programas de captura. Esta librería también puede ser encontrada en el sitio web citado arriba.

Hay versión de libpcap y tcpdump para usuarios de MS. Windows, que podremos encontrar bajo la denominación "windump".

Dejémonos de palabrería, y vayamos al grano. Después de instalar nuestro querido tcpdump en una máquina en una red no conmutada, podremos ejecutarlo como sigue:

```
zeus:~# tcpdump -i eth0
tcpdump: listening on eth0
12:31:45.638338 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: S 2402413213:2402413213(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)
12:31:45.641857 192.168.0.103.1030 > dns.terra.es.domain:  58187+ PTR? 179.106.46.207.in-addr.arpa. (45) (DF)
12:31:45.888916 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: S 840086698:840086698(0) ack 2402413214 win 17520 <mss 1460,nop,nop,sackOK>
12:31:45.889700 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: . ack 1 win 64240 (DF)
12:31:45.889889 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 1:21(20) ack 1 win 64240 (DF)
12:31:46.147843 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 21 win 17500
12:31:46.148518 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 1:21(20) ack 21 win 17500
12:31:46.148995 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 21:110(89) ack 21 win 64220 (DF)
12:31:46.295681 dns.terra.es.domain > 192.168.0.103.1030:  58187 1/5/5 (308) (DF)
12:31:46.297129 192.168.0.103.1030 > dns.terra.es.domain:  58188+ PTR? 103.0.168.192.in-addr.arpa. (44) (DF)
12:31:46.399963 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 110 win 17411
12:31:46.401133 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 21:186(165) ack 110 win 17411
12:31:46.401704 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: P 110:150(40) ack 186 win 64055 (DF)
12:31:46.653799 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 150 win 17371
12:31:46.655741 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: P 186:342(156) ack 150 win 17371
12:31:46.657797 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: F 150:150(0) ack 342 win 63899 (DF)
12:31:46.689630 dns.terra.es.domain > 192.168.0.103.1030:  58188 NXDomain 0/1/0 (121) (DF)
12:31:46.691822 192.168.0.103.1030 > dns.terra.es.domain:  58189+ PTR? 3.113.235.195.in-addr.arpa. (44) (DF)
12:31:46.907939 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: . ack 151 win 17371
12:31:46.924282 baym-cs179.msgr.hotmail.com.1863 > 192.168.0.103.4812: F 342:342(0) ack 151 win 17371
12:31:46.924642 192.168.0.103.4812 > baym-cs179.msgr.hotmail.com.1863: . ack 343 win 63899 (DF)
12:31:48.607022 10.3.15.1.bootpc > 255.255.255.255.bootps:  hlen:16 xid:0x7062181d flags:0x8000 [|bootp]
12:31:50.631991
23 packets received by filter
0 packets dropped by kernel
```

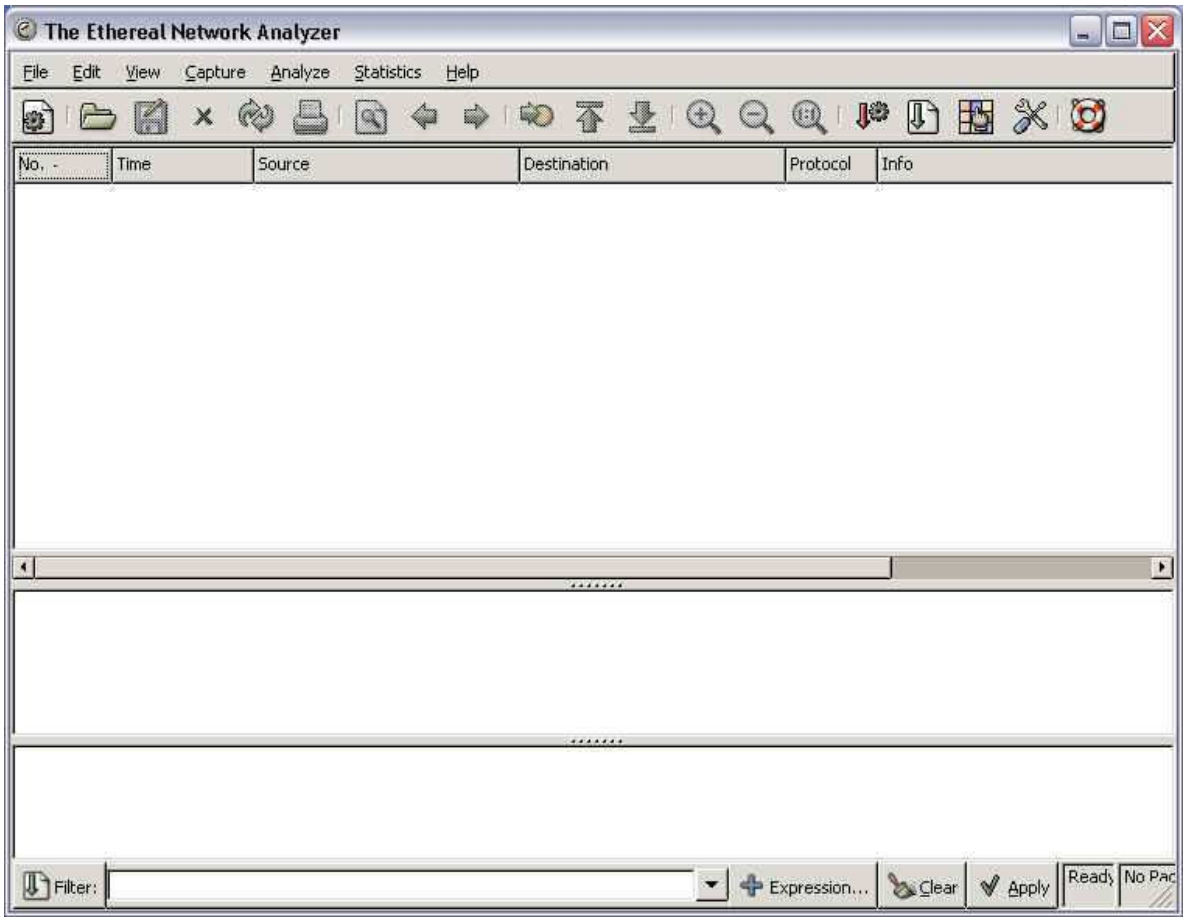
Con –i eth0 le estamos indicando al sniffer que analice el tráfico de la interfaz eth0.
Las opciones más útiles que podemos usar en tcpdump son:

| Opciones | Descripción |
|---|---|
| -n | Indica a tcpdump que no haga las traducciones de los nombres de host y los servicios y presente los valores numéricos correspondientes. |
| -s len | Establece la longitud de datos que tcpdump debe capturar, donde “len” representa dicha longitud. El valor por defecto son 68 bytes, que es suficiente para protocolos IP, TCP o UDP, aunque a veces insuficiente para otros protocolos de más alto nivel, como NFS, y algunos otros protocolos de red. |
| -v | Modo “verbose”. Presenta más información en la salida. |
| -vv | Modo más “verbose” que –v |
| -vvv | Muestra aún más información |
| -x | Imprime el contenido del paquete en hexadecimal. |
| -X | Imprime el contenido del paquete en ASCII. El tamaño del contenido viene limitado por –s. |
| -w file | Almacena la información capturada en un fichero. |
| -r file | Lee el contenido de un fichero generado con tcpdump –w file. |

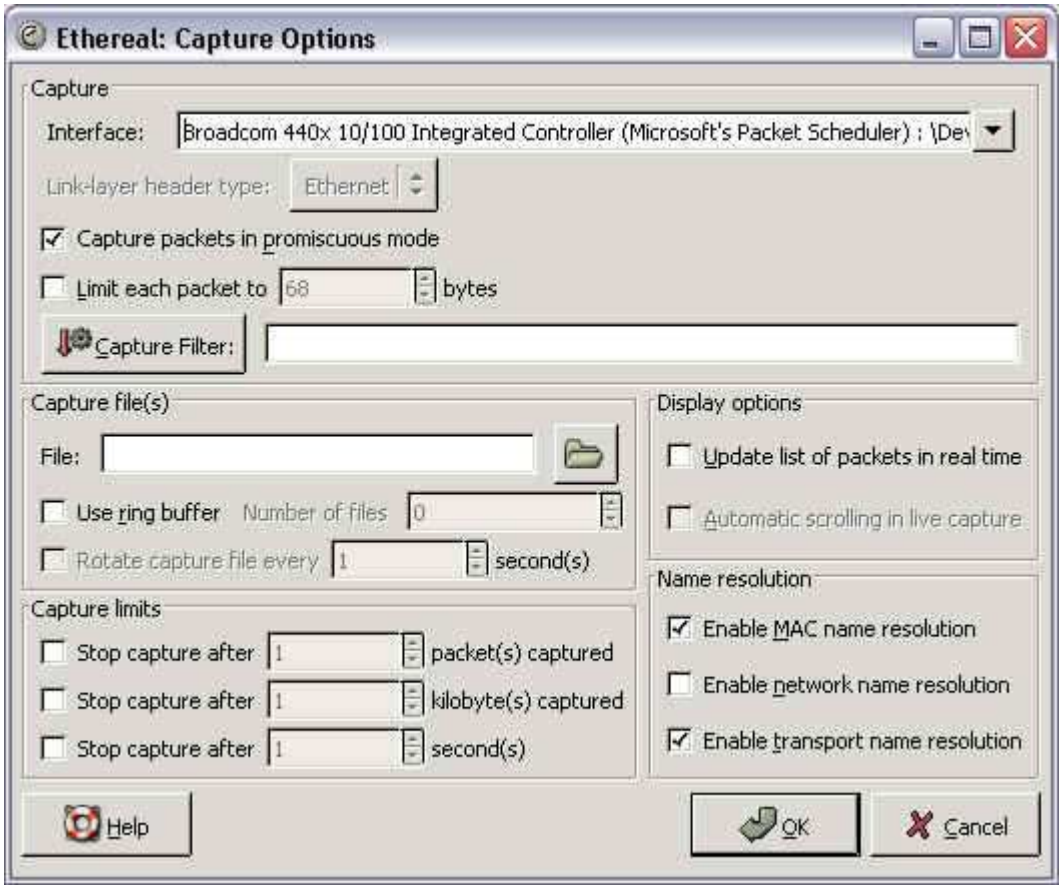
En cualquier caso, no vamos a pararnos a comentar todas las opciones de tcpdump, ya que para ello hay un completo manual que se puede consultar (man tcpdump), incluyendo las capacidades de filtrado.
Simplemente, un ejemplo más sobre tcpdump, en el que se captura el tráfico de la red dirigido a servidores telnet o ftp:
# tcpdump tcp and \(port 23 or port 21 \)

5.3.1.5.2 Ethereal

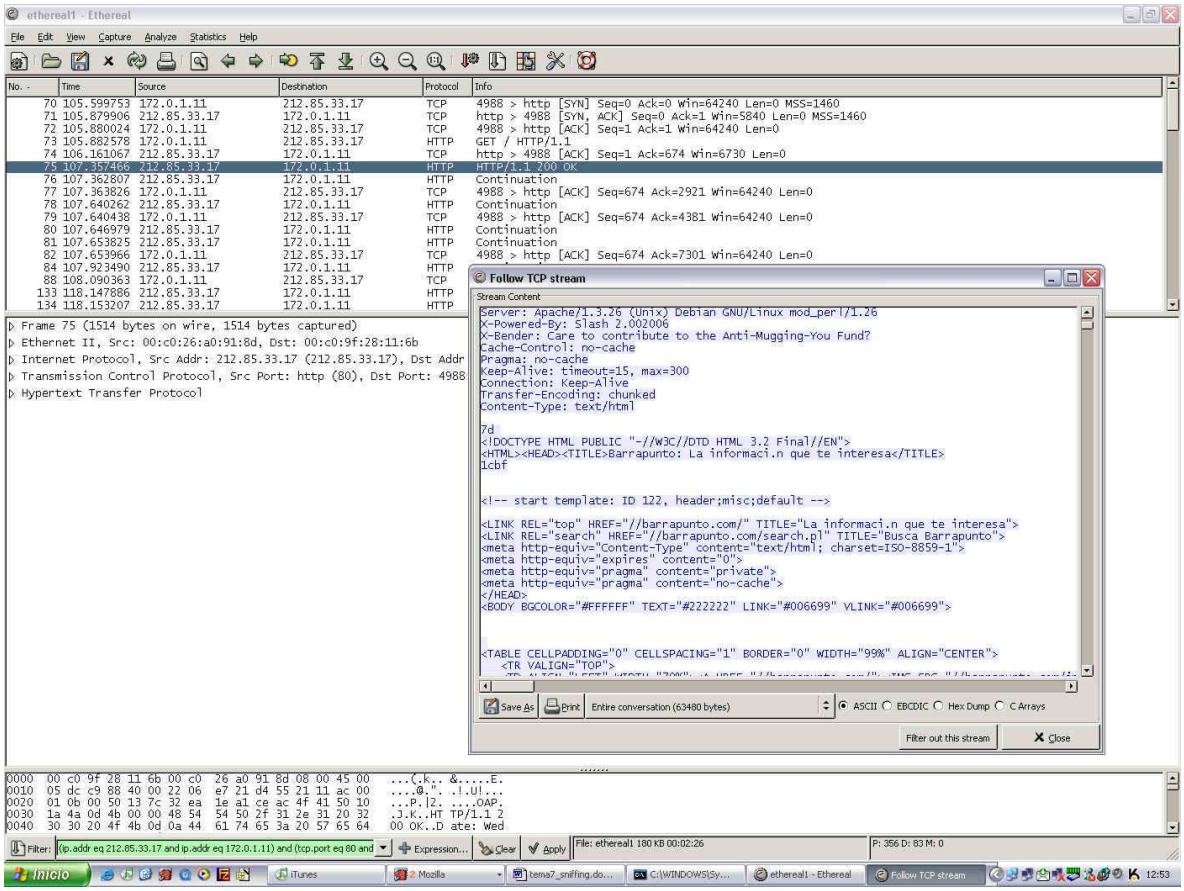
Este sniffer cuenta con una interfaz gráfica desarrollada en GTK, lo que le añade un punto de sencillez de cara al manejo por el usuario. También tiene versión para Microsoft Windows.
A día de hoy, es el sniffer más completo de cara al análisis de protocolos, que soporta más de 300. De forma análoga al tcpdump, soporta filtros (con el mismo formato que tcpdump).
Podemos obtener más información sobre ethereal en http://www.ethereal.com
El uso de ethereal es muy sencillo:
Arrancamos la aplicación:


The Ethereal Network Analyzer
File Edit View Capture Analyze Statistics Help
No. - Time Source Destination Protocol Info
Filter: Expression... Clear Apply Ready No Pac

Seleccionamos capture → start:




Seleccionamos el interfaz sobre el que capturar los paquetes, los filtros a aplicar, los ficheros de captura, configuración del buffer, etc.. y al darle a OK, comenzará la captura.
Cuando la detengamos, tendremos una pantalla con todos los datos capturados:




Entre las muchas opciones para trabajar con los paquetes, tenemos “follow tcp stream”, que permite analizar los paquetes a nivel de aplicación (por ejemplo, en la captura de pantalla, hemos seguido un stream de http).

5.3.1.5.3 Dsniff

Entre los sniffers de que podemos disponer, sin duda posiblemente el mejor es Dsniff. Dsniff está formado por un conjunto de herramientas que a continuación exponemos:

- Dsniff: sirve para sacar todas las contraseñas no encriptadas que viajan por la red.
- Arpspoof: se usa para hacer arpspoofing.
- Dnsspoof: se usa para realizar spoofing de DNS.
- FileSnarf: se dedica a capturar tráfico de NFS.
- Macof: se usa para inundar la red con direcciones MAC.
- Mailsnarf: captura los paquetes relativos al correo electrónico para poder leerlos.
- Msgsnarf: captura los mensajes de programas de mensajería instantánea.
- Sshmitm: sirve para hacer ataques man-in-the-middle en conexiones ssh.
- Tcpkill: acaba las trasferencias tcp/ip.
- Tcpnice: limita las conexiones tcp.
- Urlsnarf: captura las conexiones por medio de http.
- Webmitm: para hacer ataques MITM con servidores http / https.
- Webspy: sirve para enviar los url capturados a un navegador.

Otras herramientas interesantes son **Hunt** y **Ettercap,** que usan técnicas de envenenamiento ARP para practicar sniffers en redes conmutadas.

5.3.1.5.4 Detención de sniffers

Cuando nos enfrentamos a la detección de sniffers en la red, tenemos dos situaciones perfectamente diferenciadas:

Consulta directa de las interfaces de red.
NO es posible consultar directamente las interfaces de red.

5.3.1.5.5 Consulta directa de las interfaces de red.

En el primer caso lo que tendremos que hacer es mirar el estado de las diferentes interfaces de redes que tengamos en dicho equipo. La forma más habitual e utilizar el comando *ifconfig* (paquete net-tools), aunque podemos usar otros como *ifstatus* o *cpm* (check for network interfaces in promiscuous mode).

Estando el sniffer en ejecución, podemos ver en la primera línea la palabra "**PROMISC**", que nos revela el estado de la tarjeta de red:

Normalmente cuando la interface pasa a modo promiscuo, queda reflejado en el fichero de logs, tal y como podemos ver aquí.

```
# cat /var/log/messages
.
Nov 20 08:51:20 maquineta
        /kernel: fxp0: promiscuous mode enabled
```

Aunque es la forma más sencilla y directa de detectar un sniffer, tampoco es infalible, puesto que aun estando en marcha el sniffer puede que no aparezca la interfaz como *promiscuos* sobretodo si han crackeado la maquina y le han metido un LKM del estilo del **RhideS v1.0** (rhides.c en 7a69#12):

"Is usualy to install a sniffer when you hack some system, but if you do it, the net device is established to promisc mode and if the admin is inteligent must to discover the sniffer. Using RhideS you can to hide some promisc mode interface easily. Inserting the module you can specify magic words."

Otras posibles medidas para detectar el sniffer son:

Controlar y detectar los **logs** que genera el sniffer.
Controlar las **conexiones al exterior**, por ejemplo, el envío sospechoso de e-mail a cuentas extrañas.
Utilizar la herramienta **lsof** (LiSt Open Files), de forma que tengamos monitorizados los programas que acceden al dispositivo de red.

5.3.1.5.6 No es posible la consulta directa de las interfaces de red

En caso de que no podamos acceder y consultar el estado de las interfaces de red, puesto que el sniffer no esta en nuestra máquina sino que se encuentra en alguna otra máquina de la red. Lo que tendremos que hacer, es utilizar algún defecto en la implementación concreta del protocolo TCP/IP por algún programa/comando (tal y como hace el programa **neped** respecto a el *arp*) o ingeniárnoslas para averiguar de alguna forma si tenemos algún sniffer corriendo en la red:

"Una de las posibles técnicas, consiste en enviar paquetes a una máquina inexistente y cuya dirección no está dada de alta en el servidor de nombres. Sabremos que tenemos un sniffer en nuestra red si posteriormente detectamos cualquier intento de acceso a la máquina ficticia".

**Antisniff**, del que tenemos incluso el código fuentes en la version **Unix**, es una de las mejores herramientas de detección de sniffer de forma remota, aunque quizás este un poco obsoleto, sobretodo porque no contempla la nueva generación de sniffers.

**Sentinel** es otra interesante herramienta, cuyo objetivo principal es la detección remota de sniffers. Utiliza las librerias **libcap** y **libnet** y tenemos el código fuente disponible.

The sentinel project is an implementation of effective remote promiscuous detection techniques. For portability purposes, the sentinel application uses the libpcap and libnet libraries.

Por último comentar la existencia de una curiosa herramienta: **AntiAntiSniffer Sniffer**, cuyo objetivo es detectar la ejecución en la red del **Antisniff**, evitando ser detectado por el mismo.

### 5.3.2 Spoofing

#### *5.3.2.1 Introducción*

El vocablo inglés "spoof" significa algo así como "parodia", "truco", "engaño". Así spoofing hace referencia a cualquier forma de hacerse pasar por quien no se es (para acceder a nuestro sistema por ejemplo). Imaginemos que llama a nuestra casa un señor vestido con un mono de Repsol, que nos presenta el carné de instalador de Repsol y nos dice que ha de verificar nuestra instalación. Si no es realmente empleado y le dejamos entrar entonces podríamos decir que nos han engañado. De igual forma, cuando somos victimas de alguna técnica de spoofing, también estaremos sufriendo un engaño.

En función de los niveles en los que trabaje, podemos clasificarlos en:

- Nivel de Enlace: Envenenamiento.
- Nivel de Red.
- Nivel de Aplicación: SMTP, WebSpoofing

#### *5.3.2.2 Nivel de enlace*

##### 5.3.2.2.1 Introducción

La segmentación de redes mediante el uso de conmutadores (o switches) parecía la solución perfecta para evitar los temibles sniffers, que se tratarán en el tema siguiente. Pero no es oro todo lo que reluce y es posible aprovechar una inseguridad en el protocolo ARP para espiar en la red. Vamos a explicar en que consiste una de las técnicas utilizadas para poder espiar en una red segmentada mediante conmutadores : *ARP SPOOFING*.

El ataque denominado ARP Spoofing hace referencia a la construcción de tramas de solicitud y respuesta ARP falseadas, de forma que en una red local se puede forzar a una determinada máquina a que envíe los paquetes a una máquina atacante en lugar de hacerlo a su destino legítimo. La idea es sencilla, y los efectos del ataque pueden ser muy negativos: desde denegaciones de servicio hasta interceptación de datos, incluyendo algunos Man in the Middle contra ciertos protocolos cifrados.

##### 5.3.2.2.2 Redes ethernet

La ethernet fue concebida en torno a una idea principal: todas las máquinas de una misma red local comparten el mismo medio (el cable). Todas las máquinas son capaces de “ver” todo el tráfico de la red. Debido a esto, las tarjetas ethernet incorporan un filtro que ignora todo el tráfico que no está destinado a él. Esto se consigue ignorando aquellos paquetes cuya dirección MAC (*Media Access Control*) no coincide con la suya. Un sniffer elimina este filtro de la tarjeta de red y la coloca en modo promiscuo. De esta forma la tarjeta es capaz de “ver” todo el tráfico que pasa por la red. Sólo es cuestión de colocar los filtros adecuados y comenzar a capturar los paquetes que más nos interesen (usuario y clave de conexiones telnet, POP3, web...)

El empleo de conmutadores soluciona este problema. Mediante la segmentación de la red el único tráfico que seremos capaces de ver será el nuestro, ya que el conmutador se encarga de enrutar hacia nuestro segmento solo aquellos paquetes destinado a nuestra dirección MAC.

##### 5.3.2.2.3 ¿Qué es una dirección MAC?

Todos los ordenadores de una misma red comparten el mismo medio, por lo que debe de existir un identificador único para cada equipo, o mejor dicho para cada tarjeta de red. Esto no sucede en una conexión telefónica mediante módem, ya que se supone que cualquier dato que se envía está destinado al equipo que se encuentra al otro lado de la línea. Pero cuando se envían datos en una red local, hay que especificar claramente a quien van dirigidos. Esto se consigue mediante la dirección MAC, un número compuesto por 12 dígitos hexadecimales que identifica de forma única a cada dispositivo ethernet. La dirección MAC se compone de 48 bits. Los 24 primeros bits identifican al fabricante del hardware, y los 24 bits restantes corresponden al número de serie asignado por el fabricante, lo que garantiza que dos tarjetas no puedan tener la misma dirección MAC. Direcciones MAC duplicadas causarían problemas en la red.

¿Cómo conocer las direcciones MAC de los equipos de nuestra red?. La configuración de la tarjeta de nuestro equipo la obtendremos con el comando ifconfig (Unix/Linux) o ipconfig (Win32). La salida de este comando se asemejará al siguiente:

```
eth0 Link encap:Ethernet HWaddr 00:C0:4F:68:BA:50
```

```
inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:31658 errors:0 dropped:0 overruns:0 frame:0
TX packets:20940 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:100
Interrupt:19 Base address:0xdc00
```

Si queremos conocer las direcciones de otros equipos de la red, haremos uso de la caché arp de nuestro equipo, mediante el comando arp -a. Este comando nos mostrará la relación IP/MAC de los equipos que la caché tiene almacenados en ese momento. Por ejemplo, si queremos obtener la dirección ethernet, de una máquina, primero le haremos un ping. De esta forma almacenaremos la dirección MAC en nuestra caché y mediante arp -a podremos obtener su dirección MAC. Cada vez que deseamos comunicarnos con un equipo de la red, debemos de conocer su dirección MAC. Para ello enviaremos un arp -request a la dirección de broadcast (FF:FF:FF:FF:FF:FF), solicitando la MAC de la dirección del equipo con el que queremos contactar. Este responderá con un arp reply informándonos de su MAC. Esta quedará almacenada en la caché arp, durante algunos minutos, para futuras comunicaciones. De esta forma no tendremos que volver a solicitar la dirección MAC. Aquí es donde comienza el problema.

5.3.2.2.4 Mac spoofing

Esta técnica es quizás la más básica a éste nivel. Consiste simplemente en cambiarse la dirección MAC asignada al dispositivo de red. Esto puede permitir la suplantación de personalidad cuando la autentificación se realice por la MAC, como pueda ser asignación de IP's. Teniendo en cuenta que ésta es una técnica muy simple de implementar mediante ifconfig o edición del registro, no debería de usarse como método de autentificación ni confiabilidad.

Debe tenerse en cuenta que si el atacante y la victima se encuentran ambas conectadas, las MAC coincidirán y con el método explicado antes de obtenerlas, se vería dos máquinas con idéntica MAC, además de producir problemas de encaminamiento a éste nivel.

La solución más factible es evitar este tipo de relaciones de confianza, y si la red es conmutada, determinar que MACs y que IP's se conectan a cada boca. Esto no impide que un atacante use esa boca, pero ya es problema de seguridad física el acceso a dicha boca. Resulta más rápido robar el ordenador o copiar la información que andar usando mac spoofing.

5.3.2.2.5 Arp spoofing

Este método no pone la interfaz de red en modo promiscuo. Esto no es necesario porque los paquetes son para nosotros y el switch enrutará los paquetes hacia nosotros. Vamos a ver como es esto posible.

El método consiste en "envenenar" la caché arp de las dos máquinas que queremos espiar. Una vez que las cachés estén envenenadas, los dos hosts comenzarán la comunicación, pero los paquetes serán para nosotros, los sniffearemos y los enrutaremos de nuevo al host apropiado. De esta forma la comunicación es transparente para los dos hosts. La única forma de descubrir que existe "a man in the middle" en nuestra conexión sería ver la caché arp de nuestra máquina y comprobar si existen dos maquinas con la misma dirección MAC. El esquema de la comunicación es sencillo:

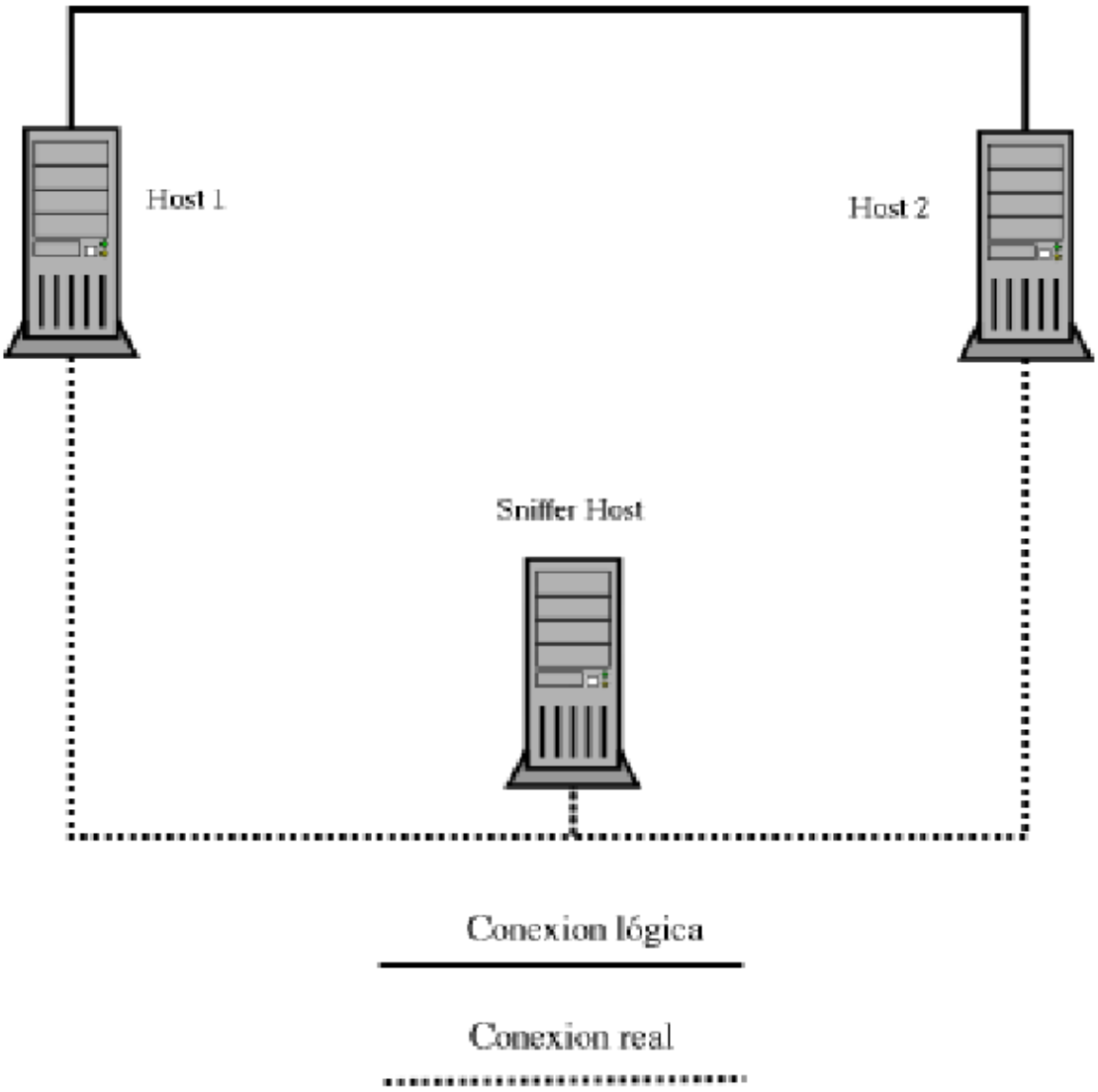
Desde nuestra máquina enviaremos paquetes de tipo ARP_REPLY falsos a las dos host que queremos sniffear. En estos reply's debemos de decirle al host 1 que la dirección ethernet del segundo host es la nuestra, quedando esta información almacenada en su caché arp. Este equipo enviará ahora los paquetes al host 2 pero con nuestra dirección MAC. Los paquetes ya son nuestros. El switch se encargará de hacernos llegar los datos. Imaginemos la siguiente situación: enviamos un flujo constante de ARP_REPLY (para evitar que la caché arp de las maquinas se refresque con la información verdadera) al host 1 y host 2 con los siguientes datos:

HOST 1 : ARP_REPLY informando que 192.168.0.2 tiene dirección MAC 03:03:03:03:03:03
HOST 2 : ARP_REPLY informando que 192.168.0.1 tiene dirección MAC 03:03:03:03:03:03

De esta forma estamos "envenenando" las caché arp. A partir de ahora lo paquetes que se envíen entre ambas nos llegarán a nosotros, pero para que ambos hosts no noten nada extraño, deberemos de hacer llegar los paquetes a su destino final. Para ello deberemos de tratar los paquetes que recibamos en función del host de origen:

Paquetes procedentes de HOST1 → reenviar a 02:02:02:02:02:02
Paquetes procedentes de HOST2 → reenviar a 01:01:01:01:01:01

De esta forma la comunicación entre ambos no se ve interrumpida, y podemos "ver" todo el tráfico entre ellos. Solo tendremos que utilizar un sniffer para poder capturar y filtrar el tráfico entre ambos, ya sea login/passwd de telnet, ftp, POP3,..., o incluso la sesión completa. Eso ya depende de la habilidad y el interés de cada cual.




Existen varios programas para "juguetear" con el ARP_ SPOOFING: Arptool, Arp_ Fun, ettercap. Este último esta muy completo, ya que permite varios tipos de sniffeo: Por IP, MAC y Arp_ Spoofing. Pudiendo ejecutarse bien en modo comando, o mediante un entorno de ventanas. En este entorno se nos mostrará al inicio un listado de los hosts encontrados en la LAN. Para realizar esta búsqueda, el programa envía un ARP_REQUEST de las IP teniendo en cuenta la IP del host donde se está ejecutando y la máscara de red. Obteniendo a continuación los ARP_REPLY's podremos componer la lista de los hosts presentes en la red. Hay que tener mucho cuidado con la máscara de red que usemos, porque si es de clase B (255.255.0.0) el programa realizará 255*255=65025 ARP_REQUEST, lo cual le llevará su tiempo ya que el retardo entre cada petición es de 1 milisegundo.

Hasta aquí hemos visto la forma en la que se pueden utilizar las vulnerabilidades del protocolo ARP para poder espiar en nuestra red. Pero las posibilidades son múltiples: ataques DoS (Denegación de servicio), si "envenenamos" la caché arp de una máquina haciéndonos pasar por el equipo pasarela de la red, toda comunicación con el exterior pasará por nosotros. Si desechamos los paquetes procedentes de este host y no los reenviamos a la pasarela, el host no podrá comunicarse con el exterior. Algunos switches pueden ser manipulados mediante paquetes ARP para que en vez de actuar en modo puente (*bridging*) lo hagan en modo repetición. Es decir, que en vez de enviar los paquetes por la boca o puerto adecuado del switch, los enviará por

todos, a todas las máquinas les llegarán todo los paquetes de la red. Esto se consigue inundando la tabla de direcciones con gran cantidad de direcciones MAC falsas. El switch al recibir un paquete cuya dirección MAC de destino no tenga en su caché, lo enviará a todos los equipos, esperando la respuesta del equipo para poder almacenar su MAC en la caché. Pero como estamos "bombardeándola" con direcciones MAC falsas, esto no ocurrirá.

La solución, como antes, pasa por asignar una MAC y una IP a la boca del conmutador, lo cual impide que se manden paquetes a una MAC o IP falseada por una boca diferente a la asignada.

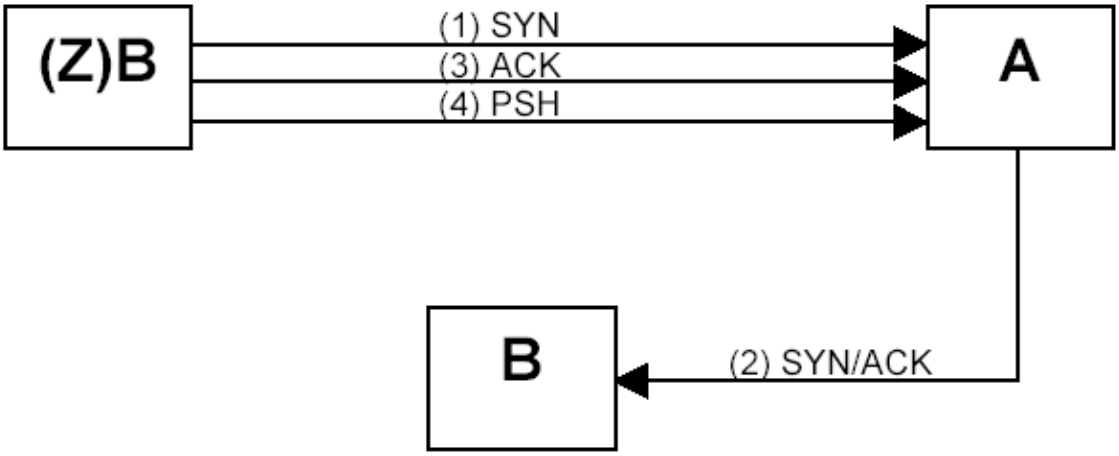
#### *5.3.2.3 Nivel de red*

##### 5.3.2.3.1 IP spoofing

El método más común de spoofing es el IP Spoofing, que consiste en suplantar una IP. El atacante logra identificarse con una IP que no es la suya, con lo que a ojos del atacado, el agresor es una tercera persona, que nada tiene que ver en el asunto, en vez de ser el atacante real. Por IP spoofing se conoce a la creación de tramas TCP/IP utilizando una dirección IP falseada; la idea de este ataque - al menos la idea - es muy sencilla: desde su equipo, un pirata simula la identidad de otra máquina de la red para conseguir acceso a recursos de un tercer sistema que ha establecido algún tipo de confianza basada en el nombre o la dirección IP del host suplantado. Y como los anillos de confianza basados en estas características tan fácilmente falsificables son aún demasiado abundantes (no tenemos más que pensar en los comandos r-, los accesos NFS, o la protección de servicios de red mediante TCP Wrapper), el spoofing sigue siendo en la actualidad un ataque no trivial, pero factible contra cualquier tipo de organización.

Como hemos visto, en el spoofing entran en juego tres máquinas: un atacante, un atacado, y un sistema suplantado que tiene cierta relación con el atacado; para que el pirata pueda conseguir su objetivo necesita por un lado establecer una comunicación falseada con su objetivo, y por otro evitar que el equipo suplantado interfiera en el ataque. Probablemente esto último no le sea muy difícil de conseguir: a pesar de que existen múltiples formas de dejar fuera de juego al sistema suplantado - al menos a los ojos del atacado - que no son triviales (modificar rutas de red, ubicar un filtrado de paquetes entre ambos sistemas...), lo más fácil en la mayoría de ocasiones es simplemente lanzar una negación de servicio contra el sistema en cuestión. Aunque en el punto siguiente hablaremos con más detalle de estos ataques, no suele ser difícil `tumbar', o al menos bloquear parcialmente, un sistema medio; si a pesar de todo el atacante no lo consigue, simplemente puede esperar a que desconecten de la red a la máquina a la que desea suplantar (por ejemplo, por cuestiones de puro mantenimiento).

El otro punto importante del ataque, la comunicación falseada entre dos equipos, no es tan inmediato como el anterior y es donde reside la principal dificultad del IP spoofing. En un escenario típico del ataque, un pirata envía una trama SYN a su objetivo indicando como dirección origen la de esa tercera máquina que está fuera de servicio y que mantiene algún tipo de relación de confianza con la atacada. El host objetivo responde con un SYN+ACK a la tercera máquina, que simplemente lo ignorará por estar fuera de servicio (si no lo hiciera, la conexión se resetearía y el ataque no sería posible), y el atacante enviará ahora una trama ACK a su objetivo, también con la dirección origen de la tercera máquina. Para que la conexión llegue a establecerse, esta última trama deberá enviarse con el número de secuencia adecuado; el pirata ha de predecir correctamente este número: si no lo hace, la trama será descartada y si lo consigue la conexión se establecerá y podrá comenzar a enviar datos a su objetivo, generalmente para tratar de insertar una puerta trasera que permita una conexión normal entre las dos máquinas.




El host atacante falsifica su dirección IP para que sea la del trusted host (el cual todavía debería estar sufriendo los efectos del ataque DoS) y envía su petición de conexión al puerto 513 del host objetivo (1).
En (2), el host objetivo responde a la petición de conexión "spoofeada" con un SYN/ACK, que recorrerá su camino hasta el trusted host (el cual, si pudiera procesar este segmento entrante, lo consideraría un error, e inmediatamente envía un RST al host objetivo) Si todo va de acuerdo con lo previsto, el SYN/ACK será ignorado por el trusted host. Después de (1), el atacante puede descansar un poco para darle tiempo al host objetivo para enviar el SYN/ACK (el atacante no puede ver este segmento).

Entonces, en (3) el atacante envía un ACK al host objetivo conteniendo el número secuencial previsto (más uno, porque estamos ACKeándolo). Si el atacante acierta en su predicción, el host objetivo aceptará el ACK.
(4) El host objetivo establece la conexión y la transferencia de datos puede comenzar.

Podemos comprobar que el spoofing no es inmediato; de entrada, el atacante ha de hacerse una idea de cómo son generados e incrementados los números de secuencia TCP, y una vez que lo sepa ha de conseguir `engañar' a su objetivo utilizando estos números para establecer la comunicación; cuanto más robusta sea esta generación por parte del objetivo, más difícil lo tendrá el pirata para realizar el ataque con éxito. Además, es necesario recordar que el spoofing es un ataque ciego: el atacante no ve en ningún momento las respuestas que emite su objetivo, ya que estas van dirigidas a la máquina que previamente ha sido deshabilitada, por lo que debe presuponer qué está sucediendo en cada momento y responder de forma adecuada en base a esas suposiciones. Sería imposible tratar con el detenimiento que merecen todos los detalles relativos al spoofing por lo que para obtener información adicional es necesario dirigirse a excelentes artículos que figuran al final.
Para evitar ataques de spoofing exitosos contra nuestros sistemas podemos tomar diferentes medidas preventivas; en primer lugar, parece evidente que una gran ayuda es reforzar la secuencia de predicción de números de secuencia TCP. Otra medida sencilla es eliminar las relaciones de confianza basadas en la dirección IP o el nombre de las máquinas, sustituyéndolas por relaciones basadas en claves criptográficas; el cifrado y el filtrado de las conexiones que pueden aceptar nuestras máquinas también son unas medidas de seguridad importantes de cara a evitar el spoofing.
Sin embargo, el IP spoofing no es una técnica cuyo éxito dependa única y exclusivamente de quién emite el paquete. Es decir, nosotros podemos mandar un paquete con una dirección de origen falseada, pero ese paquete no llegar nunca a su destino.
En redes grandes (como Internet) existen muchos dispositivos de red que pueden realizar filtrados al tráfico que gestionan. Uno de esos filtros es precisamente la comprobación de la IP de origen. Los dispositivos a los que me refiero son los de siempre:

a. cortafuegos
b. routers
c. conmutadores
d. servidores de acceso

Evidentemente, el IP spoofing funciona en una LAN (sobre todo en las que sean tipo bus pasivo). Sin embargo, en Internet no podemos suponer que, a priori, el IP spoofing puede funcionar. Los ISP aplican (o deberían aplicar) filtros en los servidores de acceso y en sus routers para 'paliar' este problema. En el peor de los casos, si no lo hace el ISP, lo hará el carrier[1], así que estamos en una situación parecida.

[1] Carrier: empresa que se encarga del mantenimiento de las líneas de datos.

Veamos un ejemplo. Un Servidor de acceso puede incorporar un filtro que garantice que los paquetes enviados tienen la IP de origen que asignó el RADIUS a cada uno de sus puertos. En el peor de los casos, el filtro puede ser por pool de direcciones asignadas al servidor de acceso. En este caso, verificaría que la IP de origen estuviera dentro de los rangos asignados a la tarjeta correspondiente (normalmente menos de una clase C).

Lo mismo puede hacerse en un router, pero con menos precisión. Un router filtra necesariamente por grupos de direcciones. El administrador conoce qué rangos direcciones de origen deberían llegar por cada una de las interfaces del router, y debería aplicar los filtros correspondientes.

Por tanto, para que un ataque DoS + IP spoofing tenga éxito desde una enlace PPP dial-up convencional, tanto el carrier como el ISP tendrían que ser muy descuidados.

Si tenemos acceso interactivo a un host remoto (una shell Unix por ejemplo), podríamos hacer spoofing en su segmento de red y atacar los sistemas situados en ese entorno, pero ese es un escenario distinto: las comunicaciones no se inician desde nuestra conexión dial-up, sino en el host remoto.

Eso significa que si alguien quiere nukear, normalmente tendrá que hacerlo 'a cara descubierta' (con el riesgo de ser detectado) o a través de una shell Unix en cualquier punto de Internet. También implica que el uso de ICMP para forzar desconexiones normalmente sólo es posible si podemos situarnos en la misma LAN y *no* atravesando Internet.

En redes internas o cerradas, tal vez pudiera funcionar, si asumen un modelo de confianza elevado. En cualquier caso, no podemos asumir, a priori, que un ataque basado en IP Spoofing pueda tener éxito desde Internet: sería una cuestión de prueba-error. ;)

#### *5.3.2.4 DNS Spoofing*

La explicación del protocolo siempre puede consultarse en las RFCs (***http://www.rfc-editor.org***) que tratan todos los aspectos de DNS: 1031, 1032, 1033, 1034, 1035.

##### 5.3.2.4.1 Conceptos Básicos

La función de un servidor DNS (Nameserver - NS ) es la de resolver IPs a nombres y viceversa. ¿Como funciona la resolución? Pues bien, pongamos un ejemplo:

Nuestro NS es ns.midominio.es y el nombre que nos han dicho que resolvamos es "pc1.laboratorio.empresa.com". Lo que hará nuestro nameserver es destripar la dirección en partes. Primero consultará a uno de los ROOT-nameservers de Internet quien sirve el dominio "com". El NS tiene una lista de estos ROOT-NS con sus IPs asociadas, luego no tiene que consultar ya que IPs tienen (sino no conseguiría resolver nunca nada). Bueno, pues consulta el dominio "com". Entonces obtiene la dirección del NS que sirve al dominio "com". A continuación pregunta a ese NS quien sirve el dominio "empresa". Obtiene la dirección de otro NS, a éste le pregunta quien sirve el dominio "laboratorio", obtiene la dirección de otro NS y le pregunta que IP tiene la maquina "pc1". Así, ya ha obtenido la IP de " pc1.laboratorio.empresa.com", pongamos, por ejemplo, que es 192.168.15.34.

Aquí ha terminado el trabajo de nuestro NS, que además de resolver la dirección, la guarda en su caché para futuras consultas. Esta caché es un registro de las últimas direcciones que resolvió el NS, por si se le consulta sobre éstas no tener que volver a resolverlas.

Nota: Éste es el funcionamiento de un NS que soporta la función de recursividad (la mayoría). Esto significa que si el NS no sirve la dirección que se le pide que resuelva, enviará la petición a otro NS para que la resuelva, hasta obtener la resolución o un error de dirección irresoluble. Si el NS no tiene esta función de recursividad, simplemente, si sirve el dominio de la dirección que le pedimos que resuelva la resolverá, y sino nos dirá que no puede resolverla, que nos busquemos la vida.

##### 5.3.2.4.2 Vulnerabilidades de servidores DNS recursivos

## Datagramas DNS

El protocolo DNS se comunica por UDP, un protocolo de transporte sin conexión, es decir, que lanza el paquete a la red, y ya no se preocupa mas de él. Esto imposibilita que se lleve un control del flujo de la conexión,

corrección de errores, etc. Éste es un punto muy importante, pues UDP no puede, al contrario que TCP, mantener un flujo de comunicación entre dos hosts, con los ya conocidos números de secuencia.

Así pues si queremos espiar o meternos en medio de una conexión UDP no tendremos que predecir estos números. Puede consultarse la estructura de los datagramas DNS en la RFC 1035 si se quieren más datos más concretos sobre como funciona.

Dado que el protocolo de transporte no identifica los paquetes por conexión, debe ser el protocolo DNS el que asigne a los paquetes una "identificación" para poder saber que paquete responde a, por ejemplo, una pregunta anterior. A esta identificación se le llama QueryID y la asigna el NS que inicia la comunicación (el que pregunta). A partir de ese momento todos los paquetes referentes a la resolución de esa pregunta irán identificados con ese QueryID.

## Inyección de datos falsos en la caché de un NS

A continuación viene la parte interesante, como engañar a un NS para que resuelva un nombre dado a una IP que queramos nosotros o al revés.

### Método A: Suplantación de un NS

Para este ataque debemos disponer de un NS primario al que pueda consultar cualquier ordenador sobre un dominio.

Digamos que nosotros somos m1.hh.org (1.1.1.1) y el NS que controlamos es ns.xs.com y queremos spoofear como XXX.lechuck.org al NS ns.victim.com.

Se trata de hacernos pasar por el NS de lechuck.org y hacer creer a ns.victim.com que ha resuelto bien a XXX.lechuck.org. Para crear este "paquete" con información falsa, el NS de victim.com debe preguntarnos sobre él, y en ese momento será cuando nosotros contestemos. Pero como no somos el NS de lechuck.org no podemos saber cual será el QueryID de ns.victim.com. Por tanto tenemos un problema, ¿cómo sabemos el QueryID que tendrá el paquete que preguntará sobre XXX.lechuck.org?

Si hacemos una consulta a ns.victim.com sobre una dirección que sirva el ns.xs.com (por ejemplo, consultamos www.xs.com), ns.victim.com se pondrá en contacto con ns.xs.com y le preguntará la dirección de www.xs.com, nuestro NS le responderá, y tan contentos ya que ya sabremos con que QueryID ha preguntado. El QueryID del protocolo DNS no es un número aleatorio o seudo-aleatorio como el de algunas implementaciones de TCP, sino que es secuencial, esto es, que para cada pregunta, aumenta en uno el QueryID. Esto es así porque el QueryID no se diseñó como un mecanismo de seguridad ante posibles spoofs, sino como una manera de controlar que respuesta corresponde a que pregunta y al revés.

Pues bien, una vez tenemos el QueryID del NS víctima creamos un datagrama DNS con la información falsa que queremos transmitir (XXX.lechuck.org <-> 1.1.1.1 ), como si la enviara el NS de lechuck.org, y con destino a ns.victim.com. Hacemos una consulta al ns.victim.com preguntando por alguna dirección del dominio lechuck.org y inmediatamente enviamos el paquete de respuesta falso. En lugar de enviar un paquete es conveniente enviar unos cuantos, con QueryIDs consecutivos, dado que en el tiempo en que hemos tardado entre que obtenemos el queryID y solicitamos la consulta por lechuck.org el NS victima puede haber recibido otras peticiones de resolución, y por tanto el QueryID puede haber aumentado ligeramente.

Así, si ns.victim.com recibe antes nuestra respuesta que la respuesta del NS de lechuck.org (en caso de que exista) o un error indicando que no hay tal dominio (en caso de que no exista) ya tendremos en la caché del NS víctima la información falsa que queríamos inyectarle.

### Método B: Inyección de datos en nuestra respuesta

Para este ataque debemos disponer igual que antes de un NS primario.

Cuando un NS hace una consulta a otro no sabe que número de respuestas puede obtener a su query, ni le importa, pues todo lo que sea información lo recibirá con mucho gusto.

Pues bien, si al hacer una consulta a un NS, además (o en lugar) de la información que solicita se le devuelve la información falsa que nosotros queramos, el NS que preguntó la aceptará y ya la tendremos en la caché del DNS que nos preguntó :)

Así, aplicado al caso anterior, lo único que debemos hacer es preparar nuestro NS (ns.xs.com) para que responda con la información falsa (XXX.lechuck.org <->1.1.1.1) a las preguntas de un NS (ns.victim.com) y hacer que el NS víctima pregunte a nuestro NS.

Este método es mucho más sencillo y efectivo que el anterior, aunque requiere más recursos para que funcione, pues debemos modificar el programa servidor de nombres o bien hacer un programa que funcione como tal, y que sirva nuestros datos falsos.

## Impacto

Aparte del uso que todos podáis estar pensando para este tipo de ataques a DNS (fanfarronear en IRC) hay multitud de usos mucho más serios, en que la seguridad de muchos sistemas puede quedar comprometida:

- NFS (Network File Sharing) en sistemas que exporten para nombres de hosts
- Servicios r* (rlogin, rsh, etc.) haciéndonos pasar por máquinas confiables.
- TCP Wrappers que se basen en nombres de hosts para cortar el paso.

5.3.2.4.3 Conclusión

El protocolo DNS es sensible a ataques de este tipo, y probablemente existen muchas más vulnerabilidades que las que aquí se han descrito.
El primer ataque seria evitable si se usase un protocolo de transporte como TCP, con números de secuencia "aleatorios" o bien usando QueryIDs también más o menos aleatorios.
La segunda vulnerabilidad es ya inherente al protocolo DNS, y necesitaría de una profunda revisión de la especificación de DNS para controlar que las respuestas se correspondan con lo que hemos preguntado.

### *5.3.2.5 ICMP Spoofing*

El Protocolo de Control de Mensajes en Internet (ICMP) se usa para manejar errores e intercambiar mensajes de control. ICMP puede usarse para determinar si una máquina responde en Internet. Para hacer esto, se envía una petición ICMP de eco a una máquina. Si la máquina recibe este paquete, devuelve un paquete ICMP de respuesta de eco. Una implementación común de este proceso es el comando “ping” que es incluido con muchos sistemas operativos y paquetes de software de red. ICMP se usa para controlar el estado e información de errores incluyendo la notificación de congestión en la red y otros problemas de transporte de red. ICMP también puede ser una valiosa herramienta para el diagnóstico de problemas en la red.
Las herramientas basadas en el protocolo ICMP, como puedan ser ping o traceroute, mandan paquetes de prueba a una máquina y calculan los paquetes perdidos en función de los paquetes de respuesta recibidos en un periodo de tiempo. Pero esto tiene dos problemas principalmente:
*Loss asymmetry*. La tasa de paquetes perdidos en la dirección de bajada de una máquina es frecuentemente diferente de la tasa en la dirección de subida. Esto hace que sin otra información adicional sea imposible determinar si el paquete se perdió o si lo que se perdió fue la respuesta. Como consecuencia, la verdadera tasa de pérdida puede representarse por:

$$1 - ((1 - loss_{fwd}) \cdot (1 - loss_{rev}))$$

donde $loss_{fwd}$ es la tasa de pérdida en la subida y $loss_{rev}$ es la tasa de pérdida en la bajada. Aunque parezca que esto no tiene importancia, hay que tener en cuenta que algunos protocolos consideran de forma diferente las pérdidas de paquetes en una u otra dirección. Por ejemplo TCP tolera mejor la pérdida de paquetes de asentimiento que los de datos. De igual forma en los protocolos de streaming la perdida de paquetes en la dirección opuesta a los datos no perjudica en nada al sistema.

Los dos componentes principales del ataque por denegación de servicios *smurf* son el uso de paquetes ICMP de petición de eco falsificados y el direccionamiento de paquetes a direcciones IP de broadcast.

En las redes IP, un paquete puede dirigirse a una máquina individual o puede transmitirse a una red entera. Cuando un paquete se envía a la dirección IP de broadcast de una red local desde una máquina de la propia red el paquete es transmitido a todas las máquinas de esa red. Cuando un paquete se envía a una dirección IP de

broadcast fuera de la red local, el paquete es transmitido a todas las máquinas de la red objetivo (siempre y cuando los routers por los que transita el paquete estén configurados para permitir el paso a este tráfico).

Las direcciones IP de broadcast normalmente son direcciones con la parte del host con todos sus bits a 1. Por ejemplo, la IP de broadcast de la red 10.0.0.0 es 10.255.255.255. Si se tiene dividida una red de clase A en 256 subredes, la dirección IP de broadcast para la subred 10.50.X.X sería 10.50.255.255. Las direcciones de red con todos los bits a cero en la parte del host, como por ejemplo, 10.50.0.0, también pueden producir una respuesta de broadcast.

En el ataque *smurf*, los atacantes utilizan paquetes ICMP de petición de eco dirigidos a IP de broadcast de máquinas remotas para generar ataques por denegación de servicios. Hay tres partes en estos ataques: el atacante, el intermediario y la víctima (Nótese que el intermediario también puede ser una víctima).

El intermediario recibe un paquete ICMP de petición de eco dirigido a la IP de broadcast de su red. Si el intermediario no filtra el tráfico ICMP dirigido a direcciones de broadcast, muchas máquinas en la red recibirán este paquete y devolverán un paquete ICMP de respuesta de eco. Cuando todas las máquinas en una red responden a esta petición de eco, el resultado puede ser la congestión severa de la red.

Cuando los atacantes crean estos paquetes, no usan la dirección de IP de su propia máquina como dirección de origen. Crean paquetes falsificados que contienen como dirección de origen la dirección IP de la víctima. El resultado es que cuando todas las máquinas del sitio intermediario responden a las demandas de eco, estas respuestas se envían a la máquina de la víctima. La red de la víctima estará sujeta a una congestión que puede potencialmente dejarla inutilizable. Aunque no se ha etiquetado al intermediario como una víctima, podría sufrir estos mismos ataques.

Se han desarrollado herramientas automatizadas que permiten enviar estos ataques al mismo tiempo a intermediarios múltiples, causando que todos los intermediarios dirijan sus respuestas a la misma víctima. También se han desarrollado herramientas para buscar routers de la red que no filtran el tráfico a direcciones IP de broadcast y redes donde múltiples máquinas responden. Estas redes pueden usarse como intermediarias en los ataques.

La solución al problema pasa por evitar que paquetes con direcciones IP falsificadas circulen por la red y evitar también la circulación de paquetes que tienen como dirección destino una dirección IP de broadcast. Vea la sección **Prevención del IP Spoofing**.

Como solución añadida sería recomendable para el intermediario en el ataque configurar su sistema operativo para impedir que la máquina responda a los paquetes ICMP enviados a direcciones IP de broadcast.

Si un intruso viola la seguridad de una máquina de su red, puede intentar lanzar un ataque *smurf* contra su red desde dentro. En este caso, el intruso usaría la máquina comprometida para enviar el paquete ICMP de petición de eco a la dirección IP de broadcast de la red local. Este tráfico no viaja a través de los routers para alcanzar las máquinas en la red local, con lo cual, el filtrado que ha hecho en sus routers no es suficiente para prevenir esto.

*5.3.2.6 Web Spoofing*

El Web Spoofing permite a un atacante la creación de una "shadow copy" DE TODAS LAS PÁGINAS DE LA WEB. Los accesos a este sitio están dirigidos a través de la maquina del atacante, permitiéndole monitorear todas las actividades que realiza la victima, desde los datos que se pueda escribir en un simple formulario, hasta sus passwords, su numero de tarjeta, etc...

El método consiste en que el atacante crea un falso (pero convincente) mundo alrededor de la victima, y la victima hace algo que le podría ser apropiado. El falso web se parece al verdadero, tiene las mismas paginas, links... En definitiva, el atacante es quien controla el falso web, así pues, todo el trafico entre el navegador de la victima y el verdadero web pasa a través del programa filtro que programó el atacante. Desafortunadamente, las actividades que parecen ser razonables en el mundo imaginario suelen ser desastrosas en el mundo real.

Las personas que usan Internet a menudo toman decisiones relevantes basadas en las señales del contexto que perciben. Por ejemplo, se podría decidir el teclear los datos bancarios porque se cree que se esta visitando el sitio del banco. Esta creencia se podría producir porque la pagina tiene un parecido importante, sale su url en la barra de navegación, y por alguna que otra razón mas.

Como el atacante tiene el control de la conexión, puede observar, e incluso modificar cualquier dato que vaya entre la victima y el verdadero web, tiene muchas posibilidades de salirse con la suya. Esto incluye Vigilancia y Manipulación.

5.3.2.6.1 Vigilancia

El atacante puede mirar el trafico de una manera pasiva grabando las paginas que visita la victima, y su contenido, como por ejemplo todos los datos que aparezcan en los formularios cuando la respuesta es enviada de vuelta por el servidor. Como la mayoría del comercio electrónico se hace a través de formularios, significa que el atacante puede observar cualquier numero de cuenta o passwords que la victima introduce.

5.3.2.6.2 Manipulación

El atacante también es libre de modificar cualquiera de los datos que se están transmitiendo entre el servidor y la victima en cualquier dirección. Por ejemplo, si la victima esta comprando un producto online, el atacante puede cambiar el numero, la cantidad, la dirección del remitente ... también le podría engañar a la victima enviándole información errónea, por parte del servidor para causar antagonismo entre ellos. Un ejemplo grafico es el siguiente:

**a) ¿Como ataca?**

La clave es que el atacante se sitúe en medio de la conexión entre la victima y el servidor.
Lo primero que se hace es grabar todo el website dentro del servidor del atacante para que así se apunte al servidor de la victima, en vez de la verdadera. Otro método sería instalar un software que actúe como filtro. Por ejemplo, si la URL del atacante es http://www.atacante.org y la del servidor verdadero es http://www.servidor.com, quedaría:

*http://www.atacante.org/http://www.servidor.com*
El navegador de la victima reclama una página de www.atacante.org

1) www.atacante.org se la reclama a www.servidor.com.
2) www.servidor.com se la entrega a www.atacante.org
3) www.atacante.org la rescriba o modifica
4) www.atacante.org le entrega la versión de la pagina que ha hecho al navegador de la victima.

**b) ¿Qué pasa con los Formularios?**

Si la victima llena un formulario de una pagina web falsa, el atacante también puede leer los datos, ya que van encerrados dentro de los protocolos web básicos. Es decir, que si cualquier URL puede ser spoofeada, los formularios también.

**c) Las conexiones seguras no ayudan**

Una propiedad angustiosa de este ataque es que también funciona cuando la navegador de la victima solicita una pagina vía conexión segura. Si la victima accede a un web "seguro" (usando Secure Sockets Layer, SSL) en un web falso, todo sigue ocurriendo con normalidad, la pagina será entregada, y el indicador de conexión segura, se encenderá (generalmente suele ser un candado).

El navegador de la victima dice que hay una conexión segura, porque tiene una, pero desgraciadamente esa conexión es con www.atacante.org, y no con el sitio que piensa la victima. El indicador de conexión segura solo le da a la victima una falsa sensación de seguridad.

## Empezando el Ataque

El atacante debe, de alguna manera colocar un cebo a la victima, para que visite la web falsa del atacante. Hay varias maneras para hacer esto, poner un link en cualquier pagina que visite la victima, engañar a los motores de búsqueda, o incluso, si se sabe su dirección de mail, enviarle uno para que visite la pagina, ...

## Completando la ilusión

Este ataque es bastante efectivo, pero no perfecto. Todavía hay detalles que pueden hacer sospechar a la victima que el ataque ya esta en marcha. En última instancia, el atacante puede llegar a eliminar cualquier rastro del ataque.

**a) La línea de estado**

Es una simple línea de texto abajo del navegador que informa de varios mensajes, como a que servidor trata de localizar, si se conecta, o el tiempo que falta todavia para recibir la totalidad de la pagina.

Este ataque deja dos tipos de evidencias en la barra de estado. La primera, cuando se pasa el mouse por encima de un enlace, informa la URL a la que apunta. Así pues, la victima podría darse cuenta de que la URL se ha modificado. La segunda es que por un breve instante de tiempo, se informa cual es la dirección del servidor que esta intentando visitar. La victima podría darse cuenta que el servidor es www.atacante.org, y no el servidor verdadero.

El atacante puede tapar estas huellas añadiendo codigo JavaScript en cada pagina reescrita para ocultar el texto en la línea de estado o hacer que cuando haya un enlace a http://www.atacante.org/http://www.servidor.com, en la línea de estado salga http://www.servidor.com, que se haría de la manera siguiente:

```
<a href="http://www.atacante.org/http://www.servidor.com"
OnMouseOver="window.status='http://www.servidor.com';return true;">
http://www.servidor.com</a>
```

Este detalle hace más convincente el ataque.

**b) La línea de navegación**

Es la encargada de informar qué URL se esta visitando. Así pues, el ataque causa que las paginas reescritas en www.atacante.org salgan en la línea de navegación. Este detalle puede hacer sospechar a la victima que el ataque está en marcha.

Esta huella puede ser ocultada ayudándose de un poco de JavaScript, de esta manera:

```
    function AbreVentana()
    {
        open("http://www.atacante.org/http://www.servidor.com/",
        "DisplayWindow","toolbar=yes,directories=no,menubar=no,
        status=yes");
}
```

También se puede hacer un programa que reemplace a la línea de navegación verdadera, que parezca que sea la correcta, colocándola en el mismo sitio, ... Si esta bien hecho se puede hacer que escriba la URL que espera ver la victima, incluso que se puedan producir entradas por teclado, para que la victima no se de cuenta.

**c) Ver documento fuente**

Los navegadores mas populares ofrecen la posibilidad de examinar el código fuente html de la pagina actual. Un usuario podría buscar URLs reescritas, mirando su código fuente, para darse cuenta del ataque.

Este ataque también puede prevenir esto ayudándose de un programa en JavaScript que oculte la barra de menús, o que haga una barra idéntica, con la salvedad que si la victima mira el código fuente, en vez de enseñar el que esta viendo, apunte a la dirección verdadera.

**d) Ver información del documento**
Esta huella se puede eliminar siguiendo las indicaciones arriba mencionadas.

5.3.2.6.3 Remedios

Web Spoofing es un ataque peligroso, y difícilmente detectable, que hoy por hoy se puede llevar a cabo en Internet. Afortunadamente hay algunas medidas preventivas que se pueden practicar:

**a) Soluciones a corto plazo**

1. Desactivar la opción de JavaScript en el navegador.
2. Asegurarse en todo momento que la barra de navegación está activa.
3. ESTA ES LA MÁS IMPORTANTE: Poner atención a las URL que se enseñan en la barra de estado, asegurándote que siempre apuntan al sitio que quieras entrar.

Hoy en día tanto JavaScript, como Active-X, como Java tienden a facilitar las técnicas de spoofing, así que desde aquí recomendamos al lector que las desactive de su navegador, al menos en los momentos que vaya a transferir información critica como login, password, números de tarjeta de crédito o cuenta bancaria, ...

**b) Soluciones a largo plazo**
Todavía no se ha descubierto ningún método para evitar este ataque.

*5.3.2.7 SMTP spoofing*

En un nivel superior, concretamente a nivel de aplicación, en el protocolo SMTP (puerto TCP 25) es posible falsear la dirección fuente de un correo o e-mail, enviando por tanto mensajes en nombre de otra persona. Es así porque el protocolo no lleva a cabo ningún mecanismo de autenticación cuando se realiza la conexión TCP al puerto asociado.

Muchos de los últimos virus que circulan por la red llevan incorporado un servidor de correo que usa las listas de contacto y direcciones de email de los usuarios infectados para enviarse con una identidad falseada en un intento de aumentar la confiabilidad de las futuras víctimas, además de ocultar el usuario que realmente está infectado. Por ejemplo, Netsky.B para autoenviarse emplea su propio motor SMTP, falsificando el e-mail del remitente con direcciones obtenidas del sistema del usuario rastreando entre archivos de diversas extensiones, como .html, .html, .php, .eml, .msg, txt, .wap o comprimido en un .zip). De esta forma la dirección no corresponderá al usuario realmente infectado desde cuyo sistema se está enviando el gusano.

También se usa esta técnica como forma de engañar a usuarios para que entren en sitios web falsificados con la intención de robar datos confidenciales. Banesto, Banco Popular, BBVA y más son ejemplos de correos engañosos enviados a sus clientes.

5.3.2.7.1 Remedio

La forma de evitar esto es implantar mecanismos de autentificación que permitan verificar que el remitente es quien dice ser y además tiene permiso para enviar el mensaje, como puedan ser autentificación en SMTP, autentificación POP antes de SMTP... o algún tipo de credenciales o intercambio de claves.

Además hay que desconfiar de mensajes no esperados con ficheros en formatos peligrosos, si bien ésta es más una regla de protección de virus y web spoofing que de SMTP spoofing.

### 5.3.3 Hijacking

TCP/IP permite la inserción de paquetes en las corrientes de datos, de modo que podemos llegar a forzar la ejecución de comandos en un host remoto.

Este tipo de ataque necesita del uso de medios compartidos, y de un poquito de suerte para llevarse a cabo de forma satisfactoria.

Hay herramientas de secuestro, como Hunt que permiten espiar las conexiones para posteriormente inyectar comandos.

### 5.3.4 Ataques a SSH

SSH se caracteriza por usar criptografía de clave pública y criptografía de clave privada para llevar a cabo las comunicaciones.

El cliente y el servidor intercambian las claves públicas, y negocian un algoritmo y una clave para el cifrado simétrico, que será el que se use para asegurar las comunicaciones posteriores.

Visto de esta manera, el protocolo SSH parece esperanzador y seguro. Al margen del historial de seguridad que han presentado algunas implementaciones, hay una situación que deberíamos considerar muy seriamente, y es la posibilidad de que existiera un hombre intermedio en la negociación de claves (Man in the middle).

De esta forma, el cliente hablaría con el MITM, y el MITM hablaría con el servidor, pudiendo interceptar todas las comunicaciones, inyectar comandos en el servidor y cualquier cosa que se pudiera ocurrir.

Por ello es una buena práctica desconfiar siempre que al conectar a un servidor se nos avise un cambio de clave, ya que realmente podemos encontrarnos bajo un ataque de hombre intermedio.

### 5.3.5 DOS

Los ataques de negación de servicio son un tipo de ataque que, como su nombre indican, intentan (y en muchos casos consiguen) provocar la caída de servicio de una organización.

Los ataques más incómodos en la denegación de servicio son los basados en consumo de ancho de banda. Fácilmente un atacante con una línea T1 es capaz de saturar líneas de 64 kbps. Sin embargo la situación contraria también es factible. Un usuario con una línea de 64 kbps podría saturar una T3 sin excesivos problemas. Ello es posible mediante el uso de otros hosts, con otras líneas, para amplificar el ataque. Las balas más usadas para los ataques DOS son los paquetes ICMP, que aunque resulta muy útil para los diagnósticos, también es muy peligroso.

Una contramedida contra los ataques por inundación de ancho de banda basado en ICMP es acordar con nuestro ISP que corten los paquetes de datos que llegan a nuestra red. En caso de ataque,

#### *5.3.5.1 Smurf*

El smurf es un ataque de tipo net flood, que consiste en el envío de paquetes ICMP a las direcciones de broadcast de las redes. Con ello se consigue que todas las máquinas en la red respondan a la máquina que originó la petición.

En función del tamaño de la red sobre la que operemos, este ataque podría constituir una denegación de servicio a la máquina que lo realiza. El factor de escala de este ataque es N, siendo N el número de máquinas existentes en la red.

Si este concepto se enriquece con IP-Spoofing, podríamos falsificar la dirección de origen de la petición por la de la máquina objetivo del ataque, y todas las máquinas de la red enviarán paquetes a la máquina objetivo.

#### *5.3.5.2 Fraggle*

Fraggle básicamente es un ataque análogo a smurf, con la diferencia de que en vez de usar ICMP emplea paquetes UDP. Usando el puerto 7 (echo) si el puerto está abierto, reemitirá el paquete a la dirección de origen (spoofeada, claro está). Si está cerrado, emitirá un paquete ICMP inalcanzable, que también supone tráfico.

#### *5.3.5.3 SYN Flood*

Antes de smurf y fraggle el ataque más devastador que había es el ataque de inundación SYN. Su funcionamiento es sencillo: el envío de un paquete SYN a un puerto abierto de un servidor, en el protocolo de establecimiento de conexión en 3 pasos, provoca que el servidor que escucha envíe un paquete SYN/ACK al origen. Generalmente, el momento que más recursos consume en un servidor es el establecimiento de las

conexiones, por lo que con relativamente pocas conexiones SYN, con la IP de origen falsificada (con la dirección IP de un equipo que no existe), puede provocar el paro del servidor.

Si el equipo de origen existe, éste enviará un paquete RST al servidor y se aborta el proceso de conexión, con lo que el ataque no tendrá éxito.

Si el equipo no existe, el servidor quedará esperando, con la conexión encolada en la cola de conexiones. Por ello, con un ancho de banda pequeño, enviando paquetes SYN a un puerto cada 10 segundos podemos caer una máquina en ese puerto. El tiempo de inactividad puede ser tanto de tiempos cortos, unos 75 segundos, hasta muchos minutos, en función de las implementaciones de las pilas TCP/IP.

Como prevención a los ataques SYN Flood podemos:

Aumentar el tamaño de la cola de conexión: esta solución no es óptima porque hará uso de recursos adicionales a los planificados, y puede afectar negativamente al rendimiento.

Disminuir el periodo del tiempo de establecimiento de conexión: al igual que el aumento del tamaño de la cola de conexiones, ayuda a paliar los efectos del ataque, pero no a evitarlos.

Activar las protecciones de los sistemas operativos: el núcleo de Linux incorpora la opción de usar desafíos criptográficos SYN-Cookie que permitan el acceso a los usuarios legítimos a través de un ataque.

Usar IDS de red: un IDS puede detectar ataques SYN Flood y enviar paquetes RST a la máquina atacada.

#### 5.3.6 DDOS

Los ataques DDOS son ataques de denegación de servicio distribuidos. Cuando se ejecuta un DDOS, no hay un atacante fijo, sino que el ataque proviene de muchas fuentes, involuntarias en muchos casos.

Este tipo de ataques se automatiza, buscando hosts vulnerables, con configuraciones débiles que son usados como zombies para perpetrar el ataque.

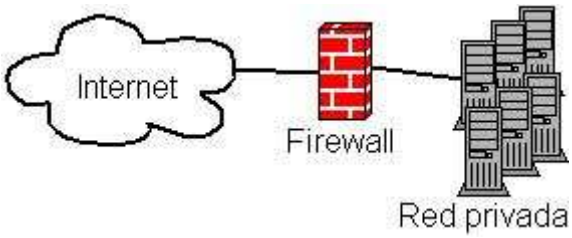
Las herramientas más significativas para este tipo de ataques son TNF, Trinoo, Stacheldraht, TFN2k y WinTrinoo. Hay herramientas posteriores, como Shaft y mStreams, pero están basadas en las anteriores.

## 5.4 Protección

### 5.4.1 Firewalls

#### *5.4.1.1 Introducción*

Un firewall es un dispositivo que proporciona un punto de defensa que controla y vigila los tráficos de entrada y salida que circulan entre dos redes (generalmente una red privada e internet).




Las características generales que debe presentar cualquier firewall son:

- Control a nivel de capa de red y de transporte.
- Reglas en base a los protocolos usados, puertos de origen y destino, direcciones IP de origen y destino y control de estado de las conexiones.

- Los firewalls no inspeccionan el tráfico a nivel de aplicación.
- Deben ser muy rápidos y transparentes a los usuarios.

Actualmente a los firewalls se les ha incorporado el soporte de nuevas funcionalidades como son NAT, VPN, filtro de contenidos (a nivel de aplicación), arquitecturas de alta disponibilidad (clusters de firewalls), integración con productos de terceros (antivirus, listas para filtrado de urls, ...).

#### *5.4.1.2 Implementación de firewalls*

La implementación de los dispositivos firewall generalmente presenta dos aspectos: soluciones compactas, o soluciones por software.

Una solución compacta es "una caja" que adquirimos a un fabricante (Checkpoint, Cisco, BlueCoat, ...) y configuramos las reglas de filtrado que queramos aplicar. Normalmente usan un sistema operativo propietario, revisado y asegurado por el fabricante.

Las soluciones por software implican el uso de un sistema operativo de propósito general, al que se le instala y configura un software específico que hará las labores de filtrado. Un ejemplo clásico es un ordenador, con dos interfaces de red, GNU-Linux e IPTables.

Las características a tener en cuenta a la hora de implementar un firewall son:

- Sistema operativo asegurado.
- Throughput
- Sesiones simultaneas.
- Número y tipo de adaptadores de red que soporta.
- Interfaces de gestión y administración.
- Soluciones de alta disponibilidad.
- Integración con productos de terceros fabricantes.

A la hora de implantar el firewall debemos:

- Denegar todos los servicios excepto los que sean estrictamente necesarios.
- Análisis periódico de los ficheros de bitácora.
- Mantener al día los sistemas en cuanto a nivel de actualizaciones y parches de seguridad.
- Realizar auditorías periódicas para evaluar el nivel de seguridad.

##### 5.4.1.2.1 Firewalls a nivel de red. Escenarios.

El primer elemento de seguridad que podemos encontrar en una red son los firewalls, que normalmente guardan la conectividad entre nuestra red corporativa e internet.

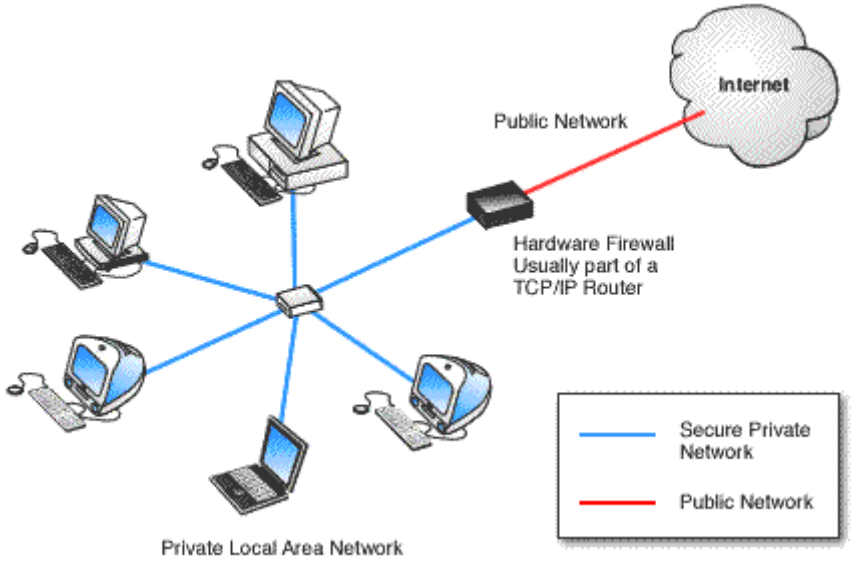
El principal objetivo que cumplen los firewalls es escuchar el tráfico que pasa entre las redes que flanquea, y llevar a cabo acciones con los paquetes de datos.

También permiten el uso de políticas de listas de control de accesos, con las que crear reglas que automáticamente refuerzan las políticas de seguridad usando direcciones IP, puertos, protocolos y estado de las conexiones para sobre las que aplicar las acciones.

Dentro de nuestra red, existen varias formas de situar los firewalls, consiguiendo de esta manera cumplir objetivos diferentes para la seguridad, negación y concesión de acceso.

## Arquitectura básica

La arquitectura básica de implementación de un firewall es aquella en la que se comunican dos redes diferentes y se configuran políticas de seguridad para la comunicación entre ambas. El ejemplo más claro que podemos encontrar sobre esta arquitectura es la red local que se conecta a internet, con un firewall que actúa como puente entre las dos redes, y solo permite que entre hacia nuestra organización el tráfico que hemos establecido en las reglas de filtrado del firewall.




Normalmente al firewall en esta topología se le suele llamar firewall perimetral, ya que determina el perímetro de acceso a la red.

Esta misma solución puede ser usada con más de una red de área local y la conexión a internet, disponiendo de diferentes reglas para cada red que se conecte.

## Arquitectura DMZ

Una forma de aumentar la seguridad en la red, si tenemos la necesidad de prestar servicios a usuarios externos a la misma, es la creación de una zona con acceso permitido desde fuera. Normalmente a estas zonas se les suele llamar zonas desmilitarizadas (DMZ).

Un claro ejemplo de este tipo de arquitecturas son las redes que disponen de servidores web, de correo, etc. accesibles desde fuera de la organización.

El hecho de permitir tráfico entrante en la red de la organización supone un factor de riesgo que trataremos de mitigar. Para ello hay diferentes arquitecturas DMZ para implementar en las organizaciones. Este es un tema muy amplio, por lo que recomiendo echar un vistazo al siguiente artículo de la publicación Linux Journal:

http://www.linuxjournal.com/article.php?sid=4415

A continuación se muestra en la figura dos arquitecturas típicas de DMZ:

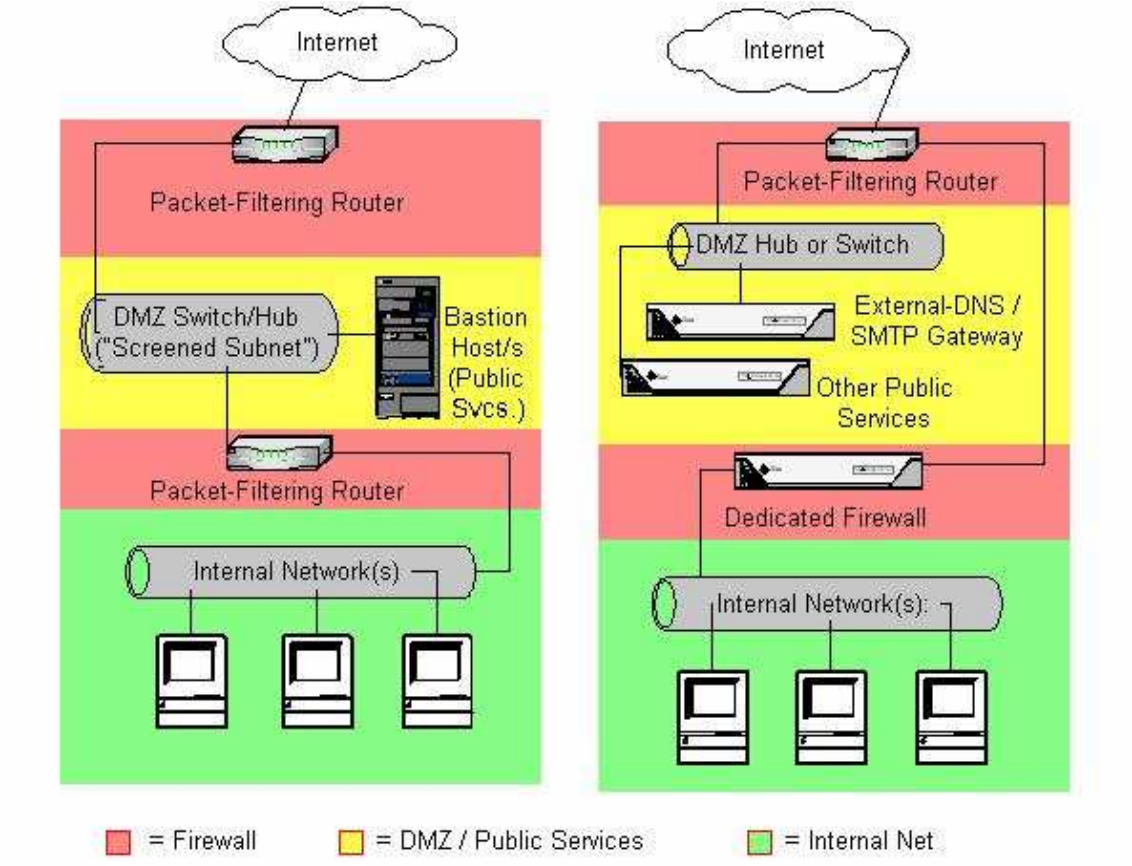



**Figure 3: "Screened Subnet" and "Flappin' in the Breeze" DMZ Architectures**

5.4.1.2.2 Soluciones software

Vamos a presentar dos tipos de soluciones firewall a nivel de software: firewalls personales e iptables (linux).

## Firewalls personales

Los firewalls personales son programas de aplicación que cumplen la función de avisar, permitir o denegar conexiones desde otros ordenadores al nuestro, o desde el nuestro hacia otros, conforme a reglas de filtrado que se pueden establecer fácilmente.

Estos programas están orientados a usuarios y sus funciones suelen ser, aunque limitadas, muy útiles: si establecemos avisos cuando un programa quiere conectarse a internet, podremos ver si algún software desconocido está intentando enviar información (como puede ser algún troyano), recibimos avisos de intentos de acceso a nuestro equipo, etc.

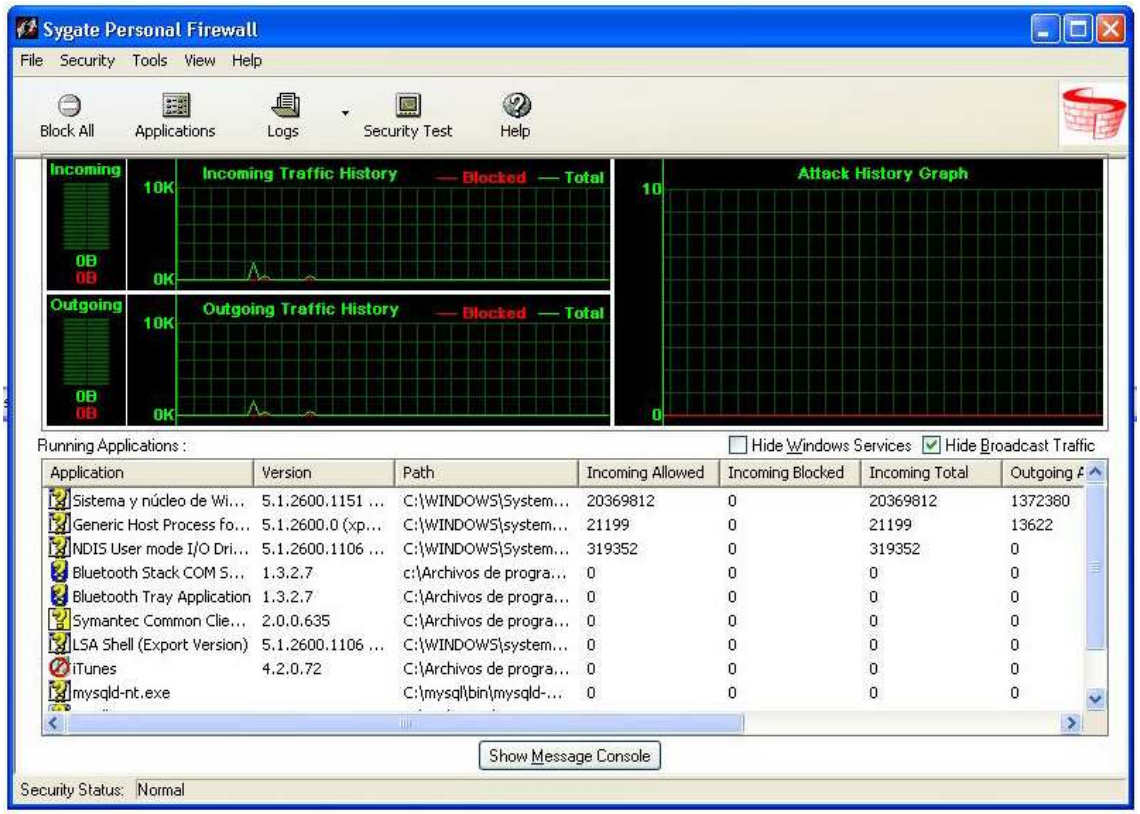
Hay programas que están muy extendidos, como son:

- ZoneAlarm (http://www.zonelabs.com)
- BlackIceDefender (http://www.iss.com)

- Kerio Personal Firewall (http://www.kerio.com).
- Sygate Personal Firewall (http://www.sygate.com)
- Otros… (http://www.google.com/)

Como ilustración, adjunto unas capturas de los firewalls personales de Kerio y Sygate.


**Kerio Personal Firewall**




**Sygate Personal Firewall**

## IPTables

IPTables es un sistema de filtrado de paquetes que forma parte del Kernel de Linux desde las versiones 2.4. También es conocido como NetFilter.

Precisamente estas son las tres reglas basicas de todo Firewall: **INPUT**, **OUTPUT** y **FORWARD**. Aunque con iptables podremos hacer muchas mas cosas como: NAT (Network Address Translation), Masquering, Control de ancho banda, Control segun la MAC, evitar ataques DoS, Control Estado (esta es una de las principales novedades respecto a ipchains), etc...

Todo esto y mucho mas, lo podemos consultar en la abundante ayuda y documentacion relativa al iptables en Internet. En la sección de bibliografía incluyo abundantes enlaces interesantes sobre el tema.

Para no dejar mal sabor de boca, haremos una pequeña introducción al uso de IPTABLES:

La sintaxis básica es:

```
Usage: iptables -[AD] chain rule-specification [options]
       iptables -[RI] chain rulenum rule-specification [options]
       iptables -D chain rulenum [options]
       iptables -[LFZ] [chain] [options]
       iptables -[NX] chain
       iptables -E old-chain-name new-chain-name
       iptables -P chain target [options]
       iptables -h (print this help information)
```

Ejemplo: supongamos que queremos denegar todas las conexiones que vengan al puerto 23 tcp de nuestra máquina. Deberíamos establecer la siguiente regla:

```
iptables –A INPUT –p tcp --dport 23 –j DROP
```

Y si queremos denegar por defecto cualquier tráfico de entrada que no encaje con nuestras reglas, deberemos poner:

```
Iptables –P INPUT DROP
```

Para aceptar las peticiones al puerto 80 tcp:

```
iptables –A INPUT –p tcp --dport 80 –j ACCEPT
```

Para más detalles, consultar la bibliografía sobre IPTABLES, al final del capítulo.

## Definir políticas de filtrado

Lógicamente, antes de ponernos a configurar nuestro firewall, tenemos que tener muy claro que tráfico hemos de permitir el paso y a cual no.

Una sana práctica para este paso es establecer las políticas de entrada a DROP, y a partir de la denegación absoluta, abrir solamente el tráfico que nos interese, en los puertos que nos interese, desde las direcciones que nos interese. No permitir todo aquello que no sea estrictamente necesario disminuye la superficie del posible ataque.

### 5.4.2 IDS

#### *5.4.2.1 Introducción a los Sistemas de Detección de Intrusos*

##### 5.4.2.1.1 ¿Qué es un Sistema de Detección de Intrusos?

Un Sistema de Detección de Intrusos o IDS (Intrusion Detection System) es una herramienta de seguridad encargada de monitorizar los eventos que ocurren en un sistema informático en busca de intentos de intrusión.

Definimos intento de intrusión como cualquier intento de comprometer la confidencialidad, integridad, disponibilidad o evitar los mecanismos de seguridad de una computadora o red.

Las intrusiones se pueden producir de varias formas: atacantes que acceden a los sistemas desde Internet, usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están autorizados y usuarios autorizados que hacen un mal uso de los privilegios que se les han asignado.

##### 5.4.2.1.2 ¿Por qué utilizar un IDS?

La detección de intrusiones permite a las organizaciones proteger sus sistemas de las amenazas que aparecen al incrementar la conectividad en red y la dependencia que tenemos hacia los sistemas de información.

Los IDSs han ganado aceptación como una pieza fundamental en la infraestructura de seguridad de la organización. Hay varias razones para adquirir y usar un IDS.

**Prevenir problemas al disuadir a individuos hostiles**.
Al incrementar la posibilidad de descubrir y castigar a los atacantes, el comportamiento de algunos cambiará de forma que muchos ataques no llegarán a producirse. Esto también puede jugar en nuestra contra, puesto que la presencia de un sistema de seguridad sofisticado puede hacer crecer la curiosidad del atacante.

**Detectar ataques y otras violaciones de la seguridad que no son prevenidas por otras medidas de protección.**

Los atacantes, usando técnicas ampliamente conocidas, pueden conseguir accesos no autorizados a muchos sistemas, especialmente a aquellos conectados a redes públicas. Esto a menudo ocurre cuando vulnerabilidades conocidas no son corregidas.

Aunque los vendedores y administradores procuran dar a conocer y corregir estas vulnerabilidades, hay situaciones en las que esto no es posible:

- En algunos sistemas heredados, los sistemas operativos no pueden ser parcheados o actualizados.
- Incluso en los sistemas en los que podemos aplicar parches, los administradores a veces no
- tienen el suficiente tiempo y recursos para seguir e instalar las últimas actualizaciones necesarias.
- Esto es un problema común, sobre todo en entornos que incluyen un gran número de
- hosts con sistemas operativos y hardware variado.
- Los usuarios y administradores pueden equivocarse al configurar sus sistemas.

Un sistema de detección de intrusos puede ser una excelente herramienta de protección de sistemas.

Un IDS puede detectar cuando un atacante ha intentado penetrar en un sistema explotando un fallo no corregido. De esta forma, podríamos avisar al administrador para que llevara a cabo un backup del sistema inmediatamente, evitando así que se pierda información valiosa.

**Detectar preámbulos de ataques (normalmente pruebas de red y otras actividades).**
Cuando un individuo ataca un sistema, lo hace típicamente en fases predecibles. En la primera fase, el atacante hace pruebas y examina el sistema o red en busca de un punto de entrada óptimo. En sistemas o redes que no disponen de un IDS, el atacante es libre de examinar el sistema con un riesgo mínimo de ser detectado. Esto le facilita la búsqueda de un punto débil en nuestra red.

La misma red con un IDS monitorizando sus operaciones le presenta una mayor dificultad. Aunque el atacante puede examinar la red, el IDS observará estas pruebas, las identificará como sospechosas, podrá activamente bloquear el acceso del atacante al sistema objetivo y avisará al personal de seguridad de lo ocurrido para que tome las acciones pertinentes.

**Documentar el riesgo de la organización.**
Cuando se hace un plan para la gestión de seguridad de la red o se desea redactar la política de seguridad de la organización, es necesario conocer cual es el riesgo de la organización a posibles amenazas, la probabilidad de ser atacada o si incluso ya está siendo atacada.

Un IDS nos puede ayudar a conocer la amenaza existente fuera y dentro de la organización,
ayudándonos a tomar decisiones acerca de los recurso de seguridad que deberemos emplear en nuestra red y del grado de cautela que deberemos adoptar al redactar la política de seguridad.

**Proveer información útil sobre las intrusiones que se están produciendo.**
Incluso cuando los IDSs no son capaces de bloquear ataques, pueden recoger información relevante sobre éstos. Esta información puede, bajo ciertas circunstancias, ser utilizada como prueba en actuaciones legales. También se puede usar esta información para corregir fallos en la configuración de seguridad de los equipos o en la política de seguridad de la organización.

### *5.4.2.2 Clasificación de los IDSs*

Existen varias formas de clasificar los IDSs:

#### 5.4.2.2.1 Clasificación según fuentes de información

Existen varias fuentes de las que un IDS puede recoger eventos. Algunos IDSs analizan paquetes de red, capturados del backbone de la red o de segmentos LAN, mientras que otros IDSs analizan eventos generados por los sistemas operativos o software de aplicación en busca de señales de intrusión.

## IDSs basados en red (NIDS)

La mayor parte de los sistemas de detección de intrusos están basados en red. Estos IDSs detectan ataques capturando y analizando paquetes de la red. Escuchando en un segmento, un NIDS puede monitorizar el tráfico que afecta a múltiples hosts que están conectados a ese segmento de red, protegiendo así a estos hosts.

Los IDSs basados en red a menudo están formados por un conjunto de sensores localizados en varios puntos de la red. Estos sensores monitorizan el tráfico realizando análisis local e informando de los ataques que se producen a la consola de gestión. Como los sensores están limitados a ejecutar el software de detección, pueden ser mas fácilmente asegurados ante ataques.

Muchos de estos sensores son diseñados para ejecutarse en modo oculto, de tal forma que sea más difícil para un atacante determinar su presencia y localización.

Ventajas:

- Un IDS bien localizado puede monitorizar una red grande, siempre y cuando tenga la capacidad suficiente para analizar todo el tráfico.
- Los NIDSs tienen un impacto pequeño en la red, siendo normalmente dispositivos pasivos que no interfieren en las operaciones habituales de ésta.
- Se pueden configurar para que sean muy seguros ante ataques haciéndolos invisibles al resto de la red.

Desventajas:

- Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho trafico y pueden fallar en reconocer ataques lanzados durante periodos de tráfico alto. Algunos vendedores están intentando resolver este problema implementando IDSs completamente en hardware, lo cual los hace mucho más rápidos.
- Los IDSs basados en red no analizan la información cifrada. Este problema se incrementa cuando la organización utiliza cifrado en el propio nivel de red (IPSec) entre hosts, pero se puede resolver con una política de seguridad más relajada (por ejemplo, IPSec en modo túnel).
- Los IDSs basados en red no saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado. Esto significa que después de que un NIDS detecte un ataque, los administradores deben manualmente investigar cada host atacado para determinar si el intento de penetración tuvo éxito o no.
- Algunos NIDS tienen problemas al tratar con ataques basados en red que viajan en paquetes fragmentados. Estos paquetes hacen que el IDS no detecte dicho ataque o que sea inestable e incluso pueda llegar a caer.

Quizá el mayor inconveniente de los NIDS es que su implementación de la pila de protocolos de red puede diferir a la pila de los sistemas a los que protege. Muchos sistemas servidores y de escritorio actuales no cumplen en ciertos aspectos los estándares TCP/IP, pudiendo descartar paquetes que el NIDS ha aceptado. Esta inconsistencia de información entre el NIDS y el sistema protegido es la base de la ocultación de ataques que veremos más adelante.

## IDSs basados en host (HIDS)

Los HIDS fueron el primer tipo de IDSs desarrollados e implementados. Operan sobre la información recogida desde dentro de una computadora, como pueda ser los ficheros de auditoría del sistema operativo. Esto permite que el IDS analice las actividades que se producen con una gran precisión, determinando exactamente qué procesos y usuarios están involucrados en un ataque particular dentro del sistema operativo.

A diferencia de los NIDSs, los HIDSs pueden ver el resultado de un intento de ataque, al igual que pueden acceder directamente y monitorizar los ficheros de datos y procesos del sistema atacado.

Ventajas:

- Los IDSs basados en host, al tener la capacidad de monitorizar eventos locales a un host, pueden detectar ataques que no pueden ser vistos por un IDS basado en red.

- Pueden a menudo operar en un entorno en el cual el tráfico de red viaja cifrado, ya que la fuente de información es analizada antes de que los datos sean cifrados en el host origen y/o después de que los datos sea descifrados en el host destino.

Desventajas:

- Los IDSs basados en hosts son más costosos de administrar, ya que deben ser gestionados y configurados en cada host monitorizado. Mientras que con los NIDS teníamos un IDS por múltiples sistemas monitorizados, con los HIDS tenemos un IDS por sistema monitorizado.
- Si la estación de análisis se encuentra dentro del host monitorizado, el IDS puede ser deshabilitado si un ataque logra tener éxito sobre la máquina.
- No son adecuados para detectar ataques a toda una red (por ejemplo, escaneos de puertos) puesto que el IDS solo ve aquellos paquetes de red enviados a él.
- Pueden ser deshabilitados por ciertos ataques de DoS.
- Usan recursos del host que están monitorizando, influyendo en el rendimiento del sistema monitorizado.

5.4.2.2.2 Tipo de análisis

Hay dos acercamientos al análisis de eventos para la detección de ataques: detección de abusos y detección de anomalías. La detección de abusos es la técnica usada por la mayoría de sistemas comerciales. La detección de anomalías, en la que el análisis busca patrones anormales de actividad, ha sido y continua siendo objeto de investigación. La detección de anomalías es usada de forma limitada por un pequeño número de IDSs.

## Detección de abusos o firmas

Los detectores de abusos analizan la actividad del sistema buscando eventos que coincidan con un patrón predefinido o firma que describe un ataque conocido.

Ventajas:

- Los detectores de firmas son muy efectivos en la detección de ataques sin que generen un número elevado de falsas alarmas.
- Pueden rápidamente y de forma precisa diagnosticar el uso de una herramienta o técnica de ataque específico. Esto puede ayudar a los encargados de la seguridad a priorizar medidas correctivas.
- Pueden permitir a los administradores de seguridad, sin importar su nivel o su experiencia en este campo, el seguir la pista de los problemas de seguridad de sus sistemas.

Desventajas:

- Solo detectan aquellos ataques que conocen, por lo que deben ser constantemente actualizados con firmas de nuevos ataques.
- Muchos detectores de abusos son diseñados para usar firmas muy ajustadas que les privan de detectar variantes de ataques comunes.

## Detección de anomalías

La detección de anomalías se centra en identificar comportamientos inusuales en un host o una red.
Funcionan asumiendo que los ataques son diferentes a la actividad normal. Los detectores de anomalías construyen perfiles representando el comportamiento normal de los usuarios, hosts o conexiones de red. Estos perfiles son construidos de datos históricos recogidos durante el periodo normal de operación. Los detectores recogen los datos de los eventos y usan una variedad de medidas para determinar cuando la actividad monitorizada se desvía de la actividad normal. Las medidas y técnicas usadas en la detección de anomalías incluyen:

- Detección de un umbral sobre ciertos atributos del comportamiento del usuario. Tales atributos de comportamiento pueden incluir el numero de ficheros accedidos por un usuario en un periodo de tiempo dado, el numero de intentos fallidos para entrar en el sistema, la cantidad de CPU utilizada por un proceso, etc. Este nivel puede ser estático o heurístico.
- Medidas estadísticas, que pueden ser paramétricas, donde la distribución de los atributos perfilados se asume que encaja con un determinado patrón, o no paramétricas, donde la distribución de los atributos perfilados es aprendida de un conjunto de valores históricos, observados a lo largo del tiempo.
- Otras técnicas incluyen redes neuronales, algoritmos genéticos y modelos de sistema inmune.

Solo las dos primeras se utilizan en los IDSs actuales, el resto son parte de proyectos de investigación.

Ventajas:

- Los IDSs basados en detección de anomalías detectan comportamientos inusuales. De esta forma tienen la capacidad de detectar ataques para los cuales no tienen un conocimiento específico.
- Los detectores de anomalías pueden producir información que puede ser utilizada para definir firmas en la detección de abusos.

Desventajas:

- La detección de anomalías produce un gran numero de falsas alarmas debido a los comportamientos no predecibles de usuarios y redes.
- Requieren conjuntos de entrenamiento muy grandes para caracterizar los patrones de comportamiento normal.

5.4.2.2.3 Respuesta

Una vez se ha producido un análisis de los eventos y hemos detectado un ataque, el IDS reacciona.

Las repuestas las podemos agrupar en dos tipos: pasivas y activas. Las pasivas envían informes a personas, que se encargarán de tomar acciones al respecto, si procede. Las activas lanzan automáticamente respuestas a dichos ataques.

## Respuestas pasivas

En este tipo de respuestas se notifica al responsable de seguridad de la organización, al usuario del sistema atacado o a algún CERT de lo sucedido. También es posible avisar al administrador del sitio desde el cual se produjo el ataque avisándole de lo ocurrido, pero es posible que el atacante monitorice el correo electrónico de esa organización o que haya usado una IP falsa para su ataque.

## Respuestas activas

Las respuestas activas son acciones automáticas que se toman cuando ciertos tipos de intrusiones son detectados. Podemos establecer dos categorías distintas:

- Recogida de información adicional: consiste en incrementar el nivel de sensibilidad de los sensores para obtener más pistas del posible ataque (por ejemplo, capturando todos los paquetes que vienen de la fuente que originó el ataque durante un cierto tiempo o para un máximo número de paquetes).
- Cambio del entorno: otra respuesta activa puede ser la de parar el ataque; por ejemplo, en el caso de una conexión TCP se puede cerrar la sesión establecida inyectando segmentos TCP RST al atacante y a la víctima o filtrar en el router de acceso o en el firewall la dirección IP del intruso o el puerto atacado para evitar futuros ataques.

*5.4.2.3 Herramientas y complementos*

5.4.2.3.1 Sistemas de valoración y análisis de vulnerabilidades

Las herramientas de análisis de vulnerabilidades determinan si una red o host es vulnerable a ataques conocidos. La valoración de vulnerabilidades representa un caso especial del proceso de la detección de intrusiones. Los sistemas que realizan valoración de vulnerabilidades funcionan en modo ’batch’ y buscan servicios y configuraciones con vulnerabilidades conocidas en nuestra red.

5.4.2.3.2 Controladores de la integridad de los ficheros

Los ’File Integrity Checkers’ son otra clase de herramientas de seguridad que complementan a los IDSs. Utilizan resúmenes de mensajes (*message digest*) u otras técnicas criptográficas para hacer un compendio del contenido de ficheros y objetos críticos en el sistema y detectar cambios, de tal forma que para cualquier cambio del

contenido del fichero el compendio sea totalmente distinto y que sea casi imposible modificar el fichero de forma que el compendio sea igual al del fichero original.

El uso de estos controladores es importante, puesto que los atacantes a menudo alteran los sistemas de ficheros una vez que tienen acceso completo a la máquina, dejando puertas traseras que más tarde facilitan su entrada al sistema, esta vez "sin hacer tanto ruido".

El producto freeware Tripwire (`www.tripwiresecurity.com`) es quizá el ejemplo más conocido de este tipo de herramientas.

#### *5.4.2.4 Agentes Autónomos para la Detección de Intrusiones*

El Grupo de Agentes Autónomos para la Detección de Intrusiones (Autonomous Agents for Intrusión Detección Group, [21]) está formado por un número de estudiantes y profesores del CERIAS en la Universidad de Purdue, (Gene Spafford como director), y están estudiando métodos distribuidos para la detección de intrusiones.

##### 5.4.2.4.1 Enfoque del grupo

Enfocan el problema de la detección de intrusiones desde un ángulo diferente: en lugar de un diseño monolítico del sistema de detección de intrusos, proponen una arquitectura distribuida que utiliza pequeñas entidades, conocidas como agentes, para detectar comportamientos anómalos o maliciosos.

Su diseño aventaja a otros enfoques en términos de escalabilidad, eficiencia, tolerancia a fallos y flexibilidad.

Estudian este enfoque desarrollando sistemas que lo implementen y miden su rendimiento y capacidades de detección. De esta forma, esperan ser capaces de conocer las capacidades y limitaciones del enfoque basado en agentes cuando es aplicado a sistemas reales.

##### 5.4.2.4.2 La arquitectura AAFID

Fue propuesta en 1994 por Crosbie y Spafford. La idea consistía en usar agentes autónomos para realizar detección de intrusiones, y sugirieron que los agentes podrían evolucionar automáticamente usando programación genética de tal forma que el sistema de detección de intrusiones automáticamente se ajuste y evolucione conforme al comportamiento del usuario.

La idea de usar programación genética no fue nunca implementada. Sin embargo, la idea de utilizar agentes para la detección de intrusiones fue evolucionando, y entre 1995 y 1996 la arquitectura AAFID fue desarrollada en el laboratorio COAST. La arquitectura inicial tuvo una estructura jerárquica que permanece en la actualidad y fue usada para implementar el primer prototipo del sistema.

Desde 1997 hasta hoy, la arquitectura AAFID evolucionó con la incorporación de filtros y la separación entre el interfaz de usuario y el monitor. La nueva arquitectura ha sido utilizada par el crear un último prototipo que se estudia en la actualidad. En la figura 1 podemos ver los cuatro componentes principales de la arquitectura: agentes, filtros, transceivers y monitores.

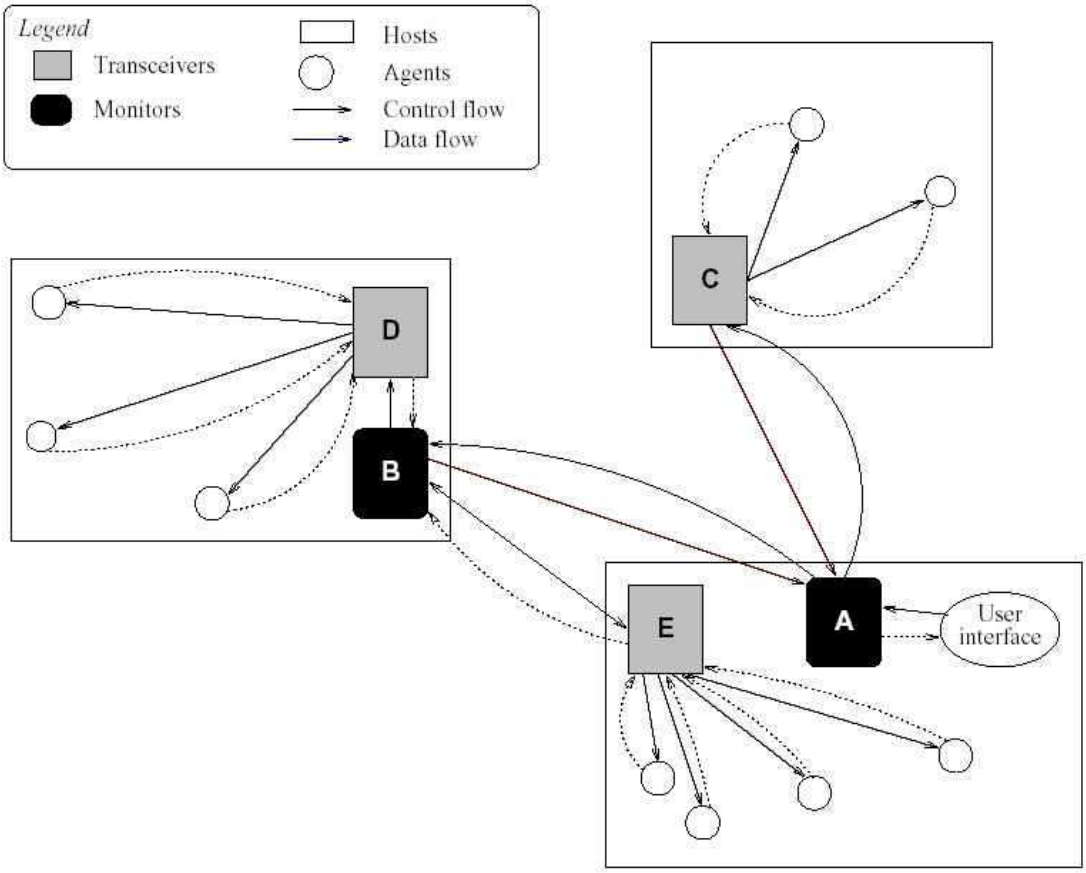



**Figura 1: Arquitectura de sistema AAFID.**

Nos referiremos a cada uno de estos componentes como "entidades AAFID" o simplemente "entidades", y a la totalidad del sistema de detección de intrusiones como "sistema AAFID".

Un sistema AAFID puede estar distribuido sobre cualquier número de hosts en una red. Un **agente** es una entidad independiente que monitoriza ciertos aspectos de un host y los notifica al **transceiver** apropiado.

Por ejemplo, un agente podría estar buscando un largo número de conexiones telnet a un host protegido, y considerar su ocurrencia como sospechosa. El agente generaría una alerta que sería enviada al transceiver apropiado. El agente no tiene la autoridad de lanzar directamente una alarma, sino que son los transceivers los que, combinando notificaciones de distintos agentes, conocen el estado actual de la red y son capaces de saber cuando existe realmente peligro.

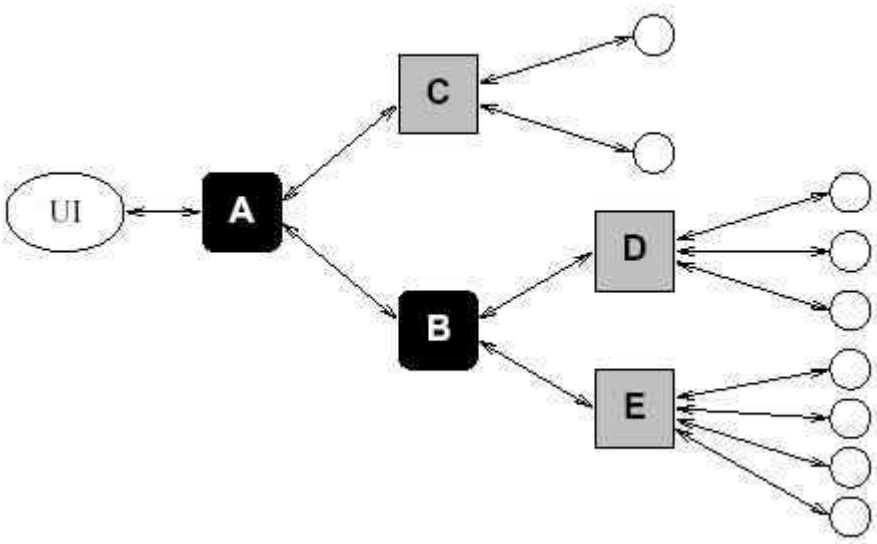



**Figura 2: Organización lógica de un AAFID.**

La figura 2 muestra la organización lógica del mismo AAFID mostrando la comunicación jerárquica de los componentes. Las flechas bidireccionales representan flujo de datos y control entre las entidades.

Los filtros se encargan de la selección de datos y de la abstracción de éstos hacia los agentes. Cada host puede contener cualquier número de agentes que monitoricen eventos interesantes que ocurran en él. Los agentes pueden usar filtros para obtener datos de una forma independiente al sistema, lo cual permite la portabilidad de agentes a distintas plataformas simplemente adoptando el filtro adecuado.

En la arquitectura AAFID original, cada agente era responsable de obtener los datos que necesitaba.
Cuando el primer prototipo fue implementado, este acercamiento mostró los siguientes problemas:

- En un único sistema, puede haber mas de un agente que necesite datos de la misma fuente. Teniendo a cada agente leyendo los datos del mismo origen se duplicaba el trabajo de lectura en los ficheros.
- Puede haber agentes que puedan proporcionar una función útil bajo diferentes versiones de UNIX, o incluso bajo diferentes arquitecturas (como Windows NT). Sin embargo, los datos requeridos por el agente pueden ser localizados en diferentes lugares en cada sistema y pueden ser almacenados en diferentes formatos. Esto significa tener que escribir un agente diferente para cada sistema, que sepa donde encontrar el dato y como leerlo.

Ambos problemas fueron resueltos con la incorporación de los filtros, una capa software que independiza a los agentes del tipo de sistema en el que están corriendo y gestiona los recursos de éste de manera óptima.

Los transceivers son el interfaz de comunicación externa de cada host. Tienen dos papeles: control y procesamiento de datos. Para que un hosts sea monitorizado por un sistema AAFID, debe haber un transceiver corriendo en ese host.

Los monitores son las entidades de más alto nivel en la arquitectura AAFID. Reciben información de todos los transceivers que ellos controlan y pueden hacer correlaciones a alto nivel y detectar eventos que ocurren en diferentes hosts. Los monitores tienen la capacidad de detectar eventos que pueden pasar desapercibidos por los transceivers.

5.4.2.4.3 Implementación de AAFID

El primer prototipo fue desarrollado entre 1995 y 1996, basado en la primera especificación de la arquitectura AAFID. Este prototipo fue implementado por una combinación de programas escritos en C, Bourne shell, AWKy Perl. Su principal objetivo era el de tener algo tangible de todo lo desarrollado hasta el momento y enfrentarse a las primeras decisiones de diseño. La primera implementación fue solamente probada internamente en el laboratorio de COAST.

La segunda implementación incorpora los cambios más recientes en la arquitectura, como son los filtros. Esta implementación es conocida como AAFID2 y fue lanzada en septiembre de 1998. La primera release de AFFID2, que incluía el sistema base y algunos agentes, fue probada bajo Solaris.
La segunda release pública fue lanzada en septiembre de 1999. Fueron añadidos nuevos mecanismos de procesamiento de eventos y fue probada en Linux y Solaris. El prototipo AFFID2v2 está implementado completamente en Perl5, lo cual lo hace fácil de instalar, ejecutar y portar a diferentes plataformas. Sólo ha sido probado en entornos UNIX, pero está en proceso de ser portado a Windows NT.

El objetivo del diseño de AAFID2v2 es que sea sencillo de experimentar y extremadamente flexible.

Fue desarrollado usando características de programación orientada a objetos de Perl5, lo cual permite que su código sea fácilmente reutilizable. AAFID2v2 también incluye una herramienta para la generación de código para desarrollar nuevos agentes.

### 5.4.3 Honeypots

http://www.securityfocus.com/infocus/1659
Los honeypots constituyen una tecnología relativamente nueva y, sin lugar a dudas, muy interesante. Nos permiten observar a nuestros enemigos para poder tomar la iniciativa ante un ataque.

No podemos acuñar una definición exacta sobre qué es un honeypot, ya que para algunos autores lo consideran como una herramienta para cebar a un atacante en un punto que realmente no es peligroso para el funcionamiento de la empresa. Para otros autores un honeypot es una máquina que ha sido destinada para ser atacada y así aprender de esta experiencia, mientras que otros autores opinan que son máquinas para la detección de intrusos (aunque no IDS).

Realmente ninguno de los términos expuestos es falso, sino que todos tienen su punto de razón. Un honeypot es una herramienta de seguridad que permite ser atacada, y que responde mentiras a las pruebas de detección de los atacantes, a los ataques, o incluso la información comprometida en esa máquina.

Podemos dividir los honeypots en dos tipos diferentes:

- De producción: se usan para asegurar las redes de las organizaciones.
- De recopilación: se pueden usar para recoger información sobre ataques (siendo buenas herramientas de aviso temprano de ataques), conocer el grado de sofisticación del atacante, e incluso aportar pruebas con fines legales.

#### *5.4.3.1 Honeypots de producción*

Para aplicar utilidad a nuestro honeypot, vamos a cosiderar el modelo de seguridad propuesto por Bruce Schneier, que consiste en tres capas: prevención, detección y respuesta.

Para la prevención, podemos usar honeypots que reduzcan la velocidad, o incluso dentengan ataques automatizados (por ejemplo gusanos) en TCP/IP. Un honeypot con esta finalidad es “Labrea Tarpit”. Para los atacantes humanos (no automatizados) se pueden usar como herramientas psicológicas que hagan cesar el ataque bien mediante técnicas de decepción o de desincentivación, que induzcan al atacante a abandonar el juego.

Los honeypots de producción también se pueden usar como sistemas para la detección de intrusiones. Al contrario que los IDS, los honeypots no generan falsos positivos. Cuando un honeypot “mete ruido” podemos estar seguros de que algo malo está ocurriendo en nuestra red. Además, los IDS generalmente no reaccionan bien contra ataques desconocidos. En cambio un honeypot avisará del ataque, aunque sea desconocido.

Finalmente, un honeypot recoge mucha información acerca del atacante, que puede servirnos para interceptar su ataque, o como pruebas en el ámbito legal.

Vamos a ver, como ilustración de uso de un honeypot HoneyD. En el momento de la elaboración de este documento, HoneyD es software libre, con código fuente disponible, gratuito y licenciado bajo licencia BSD. Funciona bajo entornos Unix-Like, y se está planeando portar el código a win32.

Durante su ejecución, Honeyd vigila el uso de las direcciones IP que no están en uso por las máquinas de nuestra red; en una clase C de direcciones, rara vez tendremos en uso 254 direcciones. Si detectamos tráfico que proviene de las direcciones IP no usadas por las máquinas de la red, podemos tener la completa seguridad de que algo malo pasa en nuestra red: un atacante, un escaneo, o incluso un gusano que intenta propagarse.

Bien, pues honeyd monitoriza el uso de todas las direcciones IP simultáneamente, y cuando es conectado desde alguna de ellas, asume que fuera un usuario e interactúa con él. Este enfoque para detección de intrusos es muy conveniente, ya que cuando el honeypot genera una alerta, podemos tener la completa seguridad de que hemos sido atacados. Las alertas no dependen de complejos algoritmos y comprobación de firmas de ataques, por lo que se elimina la detección de un gran número de falsos positivos. Dada la simpleza del funcionamiento de honeyd, su configuración es muy simple. Además, no solo no genera falsos positivos, sino que es capaz de alertar tanto de clásicos ataques, muy conocidos, como de vulnerabilidades de tipo "0-day".

Para poner en marcha honeyd no necesitamos especificar los puertos ni protocolos en los que queremos detectar las actividades "sospechosas", sino que él mismo se encarga de escuchar el tráfico TCP y UDP que llegue a la máquina. Así mismo, podemos crear scripts que simulen el comportamiento de determinados servicios bajo determinadas arquitecturas, como pudiera ser un servidor telnet en un router cisco. Los servicios emulados pueden ser creados de forma muy sencilla, usando casi cualquier lenguaje de script (perl, python, bash, ...).

Toda la actividad generada en los servicios emulados es registrada por honeyd (inicio de conexión, intentos de autenticación, fin de conexión, script que atendió el servicio emulado, etc).

Otra característica interesante es la emulación de sistemas operativos a nivel de kernel. Con ello conseguimos agregar mucho más realismo a nuestro honeypot, ya que conseguiremos que un fingerprint de nuestro honeypot resulte en el sistema operativo que deseamos emular para el atacante.

Para el uso de honeyd necesitamos también el demonio de del protocolo de resolución de direcciones en espacio de usuario arpd. Cuando arpd detecta que existe tráfico de una dirección IP no usada, falsifica la MAC de la víctima para que sea el honeypot el destinatario del paquete, y así comenzar a interactuar con el atacante. Al ser el ARP Spoofing una maniobra de nivel 2, también funcionará en entornos conmutados.

### 5.4.4 Asegurar servicios

Abordar un apartado que pretenda hablar de cómo asegurar todos los servicios que puede prestar un ordenador, o una red de servicios es un absurdo, ya que requeriría una dedicación y una extensión fuera del ámbito de este documento.

Sin embargo, vamos a revisar unos consejos y normas de actuación genérica a la hora de poner en marcha un servidor, o una red de servicios:

1. Eliminar todos los servidores instalados que no estén prestando servicios. Es un punto elemental. Si por ejemplo, nuestro servidor no presta servicios de ftp, debemos eliminar el software que lo presta. La dedicación a la seguridad en los servicios que se ejecutan ya nos ocupará lo suficiente como para tener que preocuparnos de servicios que no nos aportan ninguna función y, sin embargo, son potenciales puntos de entrada a nuestro sistema.

2. Mantenerse al día en las actualizaciones de seguridad. Todo software es susceptible de tener errores o descuidos, y los servidores no son una excepción. Debemos estar suscritos a listas de distribución de alertas de seguridad, leerlas diariamente y actuar en caso de que nuestros servidores aparezcan como potenciales víctimas.

3. Configurar que los servidores se ejecuten bajo cuentas de usuario sin privilegios, para que en caso de ser comprometido, el atacante no pueda ganar el control sobre la máquina.

4. Revisar periódicamente los archivos de bitácora de los sistemas para determinar posibles ataques, y poder actuar con previsión.

### 5.4.5 Seguridad en Wireless

#### *5.4.5.1 Introducción*

Desde hace muchos años el ser humano se ha procurado incesantemente idear artefactos que le permitieran el control remoto de los diferentes elementos con los que se relaciona en la vida cotidiana. Desde los juguetes que se controlan de forma remota, mediante transmisores y receptores de radio, mandos a distancia para televisores y vídeos, hasta los teléfonos móviles.

Este afán de manejar dispositivos de forma "telequinética" ha derivado, junto con la implantación del uso de las redes de comunicación de datos, en la necesidad de disponer de medios de transmisión de datos que no acarrearan la necesidad del uso de cables.

Muy pronto, la entrada de la tecnología de comunicación basada en redes inalámbricas ha creado nuevas expectativas de futuro para el desarrollo de sistemas de comunicación, pero también ha entrañado nuevos riesgos para la seguridad de la información y de los dispositivos que la manejan.

Como toda tecnología nueva que aún está en evolución, presenta riesgos debidos al optimismo inicial y la adopción de la tecnología sin observar los riesgos inherentes a su uso. Cuando hablamos de redes inalámbricas, va implícito el uso de un medio de transmisión observable, que es el aire.

#### *5.4.5.2 Protocolos para redes inalámbricas*

En la actualidad hay básicamente 4 estándares para el despliegue de redes de datos inalámbricas para nuestros propósitos: IrDa, HomeRF, BlueThooth, y Wi-Fi.

- BlueTooth no permite la transmisión de grandes cantidades de datos entre ordenadores de manera continua.

- IrDa (Infrared Data Association) necesita una línea de visión directa entre los componentes de la red.

Tanto Wi-Fi como HomeRF están basados en las especificaciones 802.11 (Ethernet Inalámbrica) de IEEE. En este trabajo nuestro objetivo será analizar las fallas de seguridad que acechan a Wi-Fi, ya que ésta es la tecnología que más se ha extendido en el último año, creciendo su uso a un ritmo realmente desorbitante.

#### *5.4.5.3 802.11x*

En las redes Wi-Fi podemos encontrar dos topologías: redes con puntos de acceso (APs), y redes ad-hoc.
Las redes con puntos de acceso (también llamadas redes con infraestructura) tienen como partes fundamentales los puntos de acceso. Un punto de acceso es un dispositivo dotado de una interfaz inalámbrica que emite señales periódicamente (beacons) indicando que presta el servicio de infraestructura para una red inalámbrica. La comunicación solo es posible cuando las estaciones que desean hacer uso de los servicios de acceso inalámbrico superan un reto de autentificación, y son asociadas al punto de acceso.

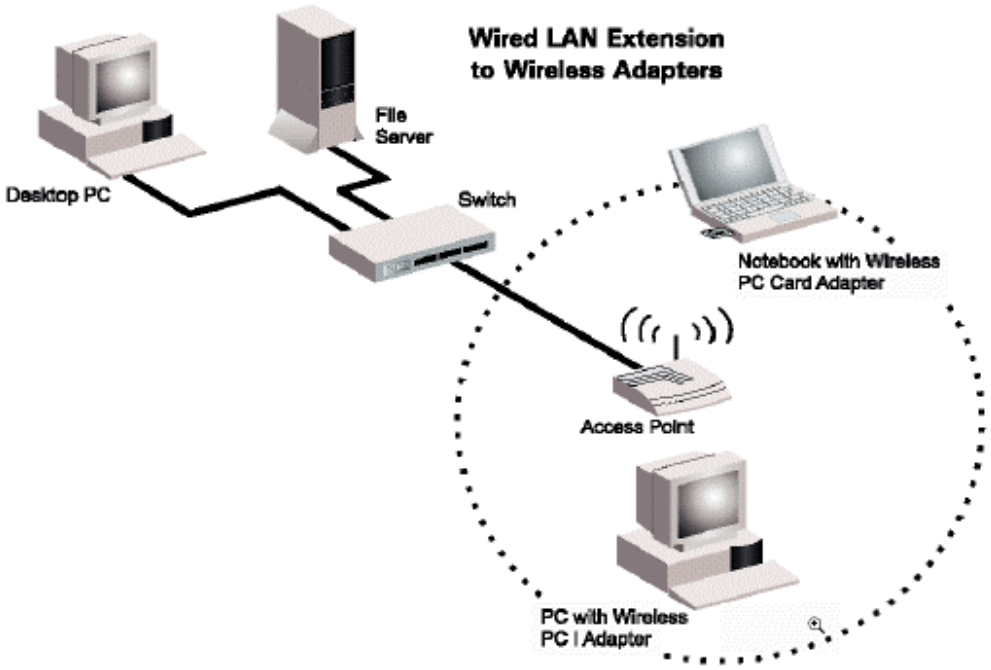



**Ilustración 1: Red inalámbrica con punto de acceso.**

Las redes ad-hoc prescinden del uso de puntos de acceso, siendo las propias estaciones que conforman la red las que se encargan de retransmitir los paquetes a través de su interfaz inalámbrica, sirviendo de paso para el tráfico entre estaciones distantes.

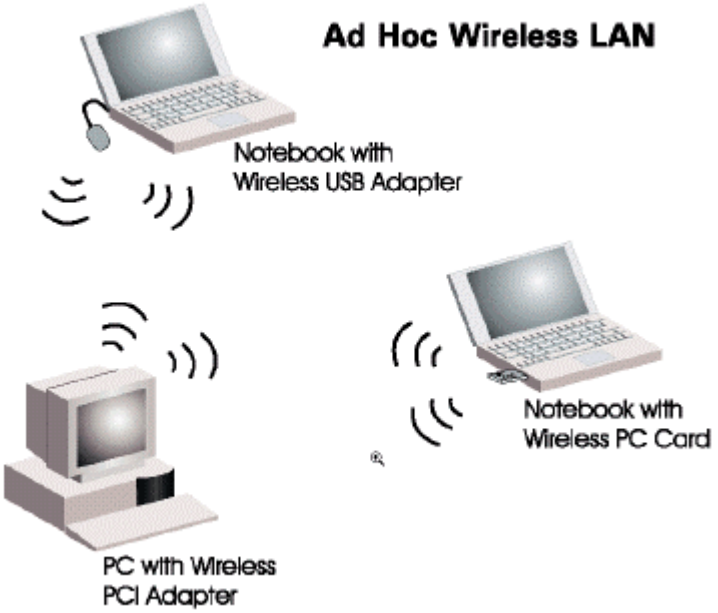



**Ilustración 2: Red  inalámbrica ad-hoc.**

Las ilustraciones presentadas en este apartado han sido obtenidas de [1].

Es necesario reseñar que dentro de las especificaciones de 802.11 hay varios comités, que se encargan de diferentes estándares siendo 802.11a, 802.11b y 802.11g los más aceptados comercialmente. Cada uno de ellos tiene sus ventajas y sus limitaciones, por lo que en un diseño de una red inalámbrica deberá sopesarse el uso que se haga de las mismas de cara a obtener el mayor rendimiento y fiabilidad.

#### *5.4.5.4 Amenazas a la seguridad de las redes inalámbricas*

##### 5.4.5.4.1 Detección de redes inalámbricas

La detección de redes inalámbricas consiste, como su nombre indica, descubrir la existencia de redes inalámbricas en un área determinada, valiéndose de dispositivos inalámbricos y programas informáticos que permitan hacer escuchas, barriendo los canales en busca de transmisiones.

Son ejemplos de programas que pueden ser usados para la detección de redes inalámbricas Network Stumbler, Kismet, Arisnort, y una gran colección de ejemplares.

Un factor que puede ayudar a evitar la detección de redes inalámbricas es desactivar en los puntos de acceso la emisión de "beacons" (mensajes que anuncian la existencia de una red con un determinado SSID). De esta forma, podremos evitar la detección activa de redes inalámbricas, y las características de las mismas.

Por otra parte, para detectar redes inalámbricas que tienen desactivada la emisión de beacons se recurre a técnicas de escucha pasiva, que es el hecho de recoger todos paquetes que circulan por la red y analizarlos con la finalidad de encontrar el identificador de la red, y luego pedir unirse a ella para extraer características de la misma. Kismet es una herramienta perfectamente adecuada para llevar a cabo este tipo de prácticas.

## Wardriving y Warchalking

Una de las prácticas habituales para la detección de redes inalámbricas es el uso de un vehículo para pasear mientras un sistema informático ejecuta desde el vehículo uno de los programas para la detección de redes. A esta práctica se le denomina wardriving. Es habitual apuntar la situación donde se ha encontrado la red inalámbrica, su ssid, el canal, si usa encriptación y si el acceso es abierto.

El warchalking es análogo al wardriving con la peculiaridad de apuntar en el suelo, normalmente con tiza o pintura los datos encontrados en el proceso de descubrimiento.

Cuando se hace referencia a un sistema informático nos referimos a un ordenador portátil, o a una PDA. También es habitual el uso de antenas omnidireccionales de baja ganancia en los vehículos.

Una práctica con resultados sorprendentes es salir de la ciudad y, desde un punto con visibilidad de la misma, usar antenas direccionales de alta ganancia para la detección de redes. El problema es que no podemos determinar fácilmente el origen de la señal desde la lejanía, y detectaremos muchas señales de puntos con los que no hay línea visual directa.

##### 5.4.5.4.2 Fallas inherentes al medio de transmisión

Las redes inalámbricas, como he comentado anteriormente, tienen un factor que determina totalmente su seguridad inicial, y es que el medio de transmisión que usan es el aire.

Si pensamos en nuestras redes de datos cableadas podemos observar desde el primer momento un factor reductor de riesgos de escucha no deseada: la transmisión se lleva a cabo a través de un cable. Es por ello que dispositivos ajenos a nuestra red no pueden espiar nuestro tráfico.

Sin embargo, el aire es de todos. Cualquier transmisión que ocurra por el aire es susceptible de ser al menos escuchada, y en muchos casos la inserción de datos en las transmisiones. Esta vulnerabilidad afecta a la privacidad, confidencialidad y a la autenticidad del origen de la información.

Con esto como antecedente, nos encontramos ante el problema de aportar la seguridad suficiente para la red, de cara a evitar el acceso no autorizado, la evitar las escuchas, la suplantación de personalidades en la red, o los ataques de denegación de servicio.

Podemos localizar como puntos vulnerables en una red inalámbrica los siguientes:
Puntos de acceso.
- Antenas.
- Interfaces inalámbricas.
- Conectores de cable.
- Servidores hardware.
- Errores de software.

#### 5.4.5.4.3 Seguridad a bajo nivel

Entrando un poco a bajo nivel, 802.11 usa tecnologías de expansión del espectro para la transmisión en el medio físico. Una de las técnicas de expansión de espectro que usan es expansión de espectro por salto de frecuencia (FHSS, Frecuency Hop Spread Spectrum), que salta entre 75 frecuencias distintas en base a una secuencia de códigos aleatorios que adoptan el emisor y el receptor. Hay 22 pautas diferentes de salto, que son seleccionadas por el emisor. El receptor puede detectar una pauta de salto y sincronizar con el emisor. Es una técnica de bajo nivel que, reseteando las secuencias cada cierto tiempo, pretende evitar las escuchas del tráfico de la red.

Otra forma de transmitir es usando la expansión de espectro por secuencia directa (DSSS, Direct Sequence spectrum Spread), que divide cada bit a transmitir en porciones de señal que se envían a frecuencias diferentes y el receptor ensambla las porciones para decodificar la señal original.

#### 5.4.5.4.4 Seguridad en el nivel de acceso al medio

Dentro de la pila de protocolos, a nivel 2 las redes ethernet usan CSMA/CD para no provocar colisiones en el medio físico. De forma análoga, 802.11 usa CSMA/CA (Carrier Sensitive Médium Access / Collision Avoid).

## Ataques de denegación de servicio

La idea de un problema de denegación de servicio recientemente publicado en AusCERT se basa en que todos los dispositivos inalámbricos de una red determinada operan en el mismo canal, por lo que solo uno de ellos es capaz de emitir a la vez, o se producirá el efecto de colisión análogo en ethernet, y la información no será válida.

Por ello, cada dispositivo escucha el canal en el que opera, y no transmite hasta que no encuentra a nadie emitiendo en el mismo. Esto no quita que dos dispositivos escuche el canal libre, y decidan emitir al mismo tiempo. En este caso, habrá un solapamiento de la señal y ambos mensajes quedarán inservibles. Por ello, cada estación que haya emitido al mismo tiempo, detendrá su transmisión y esperará un tiempo aleatorio para volver a emitir, si el canal sigue libre. Para minimizar las colisiones, los dispositivos implementan un mecanismo llamado CCA (Clear Channel Assessment), que consiste en ir comprobando la señal que se recibe y así decidir cuando el canal está libre, y puede ser usado.

El ataque de denegación de servicios consiste en falsear el mecanismo de forma que las estaciones siempre detecten el canal como ocupado, y por consiguiente tengan que esperar infinitamente.

La característica que hace especialmente incómodo este ataque es que cualquier dispositivo inalámbrico es capaz de provocar esta denegación de servicio en su radio de cobertura, sin necesidad de hacer modificaciones en el hardware.

Al ser éste un fallo inherente al diseño de 802.11, el problema no es parcheable en los dispositivos. Afortunadamente no todos los dispositivos 802.11 son vulnerables al ataque: solo son 802.11b y 802.11g en un modo de transmisión de 22 Mbps o inferior. 802.11a y 802.11g no son vulnerables, en el modo de transmisión de 54Mbps.

Como impacto de este ataque, basta imaginar una empresa cuya red está desplegada de forma completa mediante 802.11b. Romper la productividad de esta empresa sería un juego de niños y podría hacerse de forma muy sencilla desde dentro de la organización, desde fuera de la misma, por ejemplo en la calle, desde un tejado cercano, o con una antena direccional desde varios kilómetros de distancia. Las cifras de pérdidas que puede ocasionar un ataque de este tipo son cuantiosas, ya que es capaz de detener de forma completa todas las comunicaciones de la empresa.

### Filtrado de direcciones MAC

Una forma de establecer un control de acceso de usuarios a las redes inalámbricas es restringir las direcciones MAC que pueden emparejarse con nuestros dispositivos, estableciendo una ACL a tal efecto.

Las direcciones MAC de las interfaces de red inalámbricas 802.11, al igual que en ethernet tienen una dirección física, que es asignada por el fabricante y que es única para cada dispositivo.

De esta forma, se pretende asegurar que solo acceden a la red inalámbrica.

El problema de este método de validación de equipos que se añaden a la red es patente. Variar la dirección MAC de las interfaces de red no solo es posible, sino que además es muy sencillo. Basta la ejecución de un simple comando bajo Linux, o la edición del registro bajo Windows para cambiar la MAC de un dispositivo por una MAC que hayamos escuchado en el medio y entrar directamente en la red que estaba hipotéticamente cerrada.

Respecto de la seguridad supone un problema importante, ya que cualquiera podría insertar tráfico en nuestra red, acceder a los recursos, o incluso falsificar tráfico.

5.4.5.4.5 El cifrado WEP

WEP es el acrónimo de privacidad equivalente al cableado (Wired Equivalent Privacy). Es un protocolo de cifrado que forma parte del estandar 802.11 y pretende emular el control de acceso a las transmisiones de forma análoga a lo que es el cableado para una red ethernet.

La única forma de defender de escuchas las transmisiones de las redes inalámbricas es el uso de encriptación para los contenidos que se transmiten, y aquí es donde WEP toma sentido. Por otra parte WEP también evita que usuarios no autorizados hagan uso de la red mediante autentificación fuerte.

## Funcionamiento de WEP

WEP está diseñado para evitar que nadie escuche o modifique ninguna proporción del flujo de datos. Para ello usa una secuencia RC4 de 40 bit para encriptación de datos y un CRC de 32 bit para verificarlo.

WEP usa una clave secreta compartida por el usuario inalámbrico y el punto de acceso, de modo que todos los datos transmitidos y recibidos por ambos puedan ser encriptados usando esa misma clave compartida. El 802.11 permite el uso de una clave secreta única.

WEP usa un vector de inicialización (VI) de 24 bits aleatorio y que cambia con cada trama transmitida.

Este vector de inicialización es usado para concatenar con la clave secreta formando el SEED, que se usa para generar números pseudoaleatorios del tamaño del payload (cuerpo + CRC) más el vector de comprobación de integridad (ÏCV).

El ICV es una suma de comprobación que se recalcula en el destino para comprobar la validez de los datos enviados.

En el momento de envío de la trama, WEP combina la clave con el payload/ICV mediante operaciones XOR a nivel de bit, y añade el vector de inicialización sin cifrar, y los primeros bytes del cuerpo de la trama, sin cifrar también.

El receptor usa el IV de la trama para descifrar el payload.

Aparentemente el cifrado wep puede parecer seguro, pero ostenta varias debilidades que provocan el parecer en una falsa sensación de seguridad:

- La longitud del vector de inicialización es muy corta.
- Usa llaves de cifrado estáticas.
- Usando solo 24 bits, WEP repetirá el vector de inicialización en paquetes diferentes, pudiendo ser repetido en transmisiones continuas.
- Con suficientes tramas se puede conocer la clave secreta compartida por criptoanálisis diferencial.

Con ello podemos deducir que el cifrado WEP no es la solución definitiva para la protección de la información en redes inalámbricas, pero no debemos olvidar que poca seguridad es mejor que ninguna, WEP es una barrera de entrada que evitará el éxito de los intrusos con pocos recursos. Es apropiado para entornos domésticos y educativos.

También es recomendable cambiar la clave secreta de WEP de forma periódica y frecuente, de cara a evitar que un atacante consiga toda la información que pueda necesitar para llevar a cabo un ataque a la red.

#### *5.4.5.5 WPA*

WPA surge con la finalidad de añadir integridad y protección a las redes inalámbricas, conformando un sistema de “migración” de las redes actuales hasta que 802.11i sea ratificado y se pueda adoptar WPA2.

Entre sus características relativas a seguridad se encuentran las siguientes:

- Fuerza la autenticación: requiere la autenticación 802.1x, que en 802.11 era opcional. En entornos en los que no exista un servidor Radius, WPA admite una clave previamente compartida.
- Requiere el protocolo de integridad de clave temporal (TKIP), expuesto más abajo.
- Michael: el método Michael proporciona características que mejoran la integridad de los mensajes añadiendo un MIC de 8 bytes que se coloca entre los datos del marco IEE 802.11 y el ICV de 4 bytes, y se cifra junto con los otros datos. También es una ayuda contra los ataques de reproducción, ya que introduce un contador de trama.
- Admite AES como sustituto de WEP, de cara a mejorar la fortaleza del cifrado, aunque es opcional para los fabricantes.

#### *5.4.5.6 Buenas prácticas en el diseño de redes inalámbricas*

Se podrían hacer varias recomendaciones para diseñar una red inalámbrica e impedir lo máximo posible el ataque de cualquier intruso.

Como primera medida, se debe separar la red de la organización en un dominio público y otro privado. Los usuarios que proceden del dominio público (los usuarios de
la red inalámbrica) pueden ser tratados como cualquier usuario de Internet (externo a la organización). Así mismo, instalar cortafuegos y mecanismos de autentificación entre la red inalámbrica y la red clásica, situando los puntos de acceso delante del cortafuegos y utilizando VPN a nivel de cortafuegos para la encriptación del tráfico en la red inalámbrica.

Los clientes de la red inalámbrica deben acceder a la red utilizando SSH, VPN o IPSec y mecanismos de autorización, autenticación y encriptación del tráfico (SSL). Lo ideal sería aplicar un nivel de seguridad distinto según que usuario accede a una determinada aplicación.

La utilización de VPNs nos impediría la movilidad de las estaciones cliente entre puntos de acceso, ya que estos últimos necesitarían intercambiar información sobre los usuarios conectados a ellos sin reiniciar la conexión o la aplicación en curso, cosa no soportada cuando utilizamos VPN.

Como contradicción, es recomendable no utilizar excesivas normas de seguridad por que podría reducir la rapidez y la utilidad de la red inalámbrica. La conectividad entre estaciones cliente y PA es FCFS, es decir, la primera estación cliente que accede es la primera en ser servida, además el ancho de banda es compartido, motivo por el cual nos tenemos que asegurar un número adecuado de puntos de acceso para atender a los usuarios.

También se podrían adoptar medidas extraordinarias para impedir la intrusión, como utilizar receivers (Signal Leakage Detection System) situados a lo largo del perímetro del edificio para detectar señales anómalas hacia el edificio además de utilizar estaciones de monitorización pasivas para detectar direcciones MAC no registradas o clonadas y el aumento de tramas de reautentificación.

Por último también podrían ser adoptadas medidas físicas en la construcción del edificio o en la utilización de ciertos materiales atenuantes en el perímetro exterior del edificio, debilitando lo máximo posible las señales emitidas hacia el exterior. Algunas de estas recomendaciones podrían ser, aún a riesgo de resultar extremadas:

- Utilizar cobertura metálica en las paredes exteriores.
- Vidrio aislante térmico (atenúa las señales de radiofrecuencia).
- Persianas venecianas de metal, en vez de plásticas.
- Poner dispositivos WLAN lejos de las paredes exteriores.
- Revestir los closets (rosetas) de la red con un revestimiento de aluminio.
- Utilizar pintura metálica.
- Limitar el poder de una señal cambiando la atenuación del transmisor.

#### *5.4.5.7 Cambios futuros: 802.11i*

Tomando conciencia de las debilidades que acarrea el estándar 802.11 en su protocolo WEP, se constituyó el comité 802.11i para tratar de mejorar los aspectos de seguridad en las redes inalámbricas.

A estas alturas muchos fabricantes han desarrollado tecnologías propietarias para mejorar la seguridad en las capas superiores del modelo OSI, que no estarán especificadas en 802.11i por no ser un objetivo del estándar tratar las capas superiores.

##### 5.4.5.7.1 802.1x

El estándar 802.1x es un estándar de control de acceso a la red basado en puertos. Restringe el acceso a la red hasta que el usuario se ha validado. El sistema presentaría los siguientes elementos:

- Estaciones cliente.
- Punto de acceso.
- Servidor de autenticación.

En un sistema con 802.1x activado, se generarán 2 claves: la clave de sesión (pairwise key) y la clave de grupo (groupwise key). Las llaves de grupo son compartidas por todas las estaciones conectadas a un mismo punto de acceso, y son usadas para el tráfico multicast. Las llaves de sesión son únicas para cada asociación entre cliente y punto de acceso, y establecerán un puerto privado entre ambos.

Respecto de la seguridad, y comparado con WEP, 802.1x aporta las siguientes mejoras:

- Presenta un modelo de seguridad con administración centralizada.

- La llave de cifrado principal es única para cada sesión, por lo que el tráfico de la misma es reducido (no se repite en otros clientes, como en WEP).
- Existe una generación dinámica de claves por parte del servidor de autenticación, sin necesidad de ser administrado manualmente.
- Se aplica una autentificación fuerte en la capa superior.

5.4.5.7.2 TKIP (Temporal Key Integrity Protocol)

Este protocolo pretende resolver las deficiencies del algoritmo WEP u mantener la compatibilidad con el hardware usado hasta el momento mediante una actualización de firmware.

El protocolo TKIP contiene los siguientes elementos:
- Un código de integración de mensajes, que cifra el checksum incluyendo las direcciones físicas de origen y destino, y los datos en texto plano de la trama 802.11. Con ello hacemos frente a los ataques de falsificación.
- Medidas para reducir la probabilidad de que un atacante pueda usar una clave determinada, o deducirla de la transmisión.
- Uso de un vector de inicialización de 48 bits, llamado TSC (TKIP Secuence Counter) para protegerse de ataques por repetición, descartando los paquetes recibidos fuera de orden.

5.4.5.7.3 CCMP (Counter Mode with CBC-Mac Protocol)

Es un protocolo complementario a TKIP y presenta un nuevo método de cifrado basado en AES, con el algoritmo CBC-MAC.

CCMP usará un vector de inicialización de 48 bits denominado Número de Paquete, usado a lo largo del proceso de cifrado, junto con la información para inicializar el cifrado AES para calcular el MIC y el cifrado de la trama.

Mientras que TKIP será opcional en 802.11i, CCMP va a ser de uso obligado.

## 5.5 Resumen

En este capítulo hemos podido aprender las diferentes formas en que podemos “interpretar” las normas para provocar situaciones que en muchos casos no han sido previstas por los administradores de redes y sistemas.

Hemos visto como hacer escuchas en la red, inyectar información en las comunicaciones, falsificar la procedencia de las comunicaciones, secuestrar las sesiones, falsificar los registros de DNS para llevar a cabo suplantaciones en servicios y como “mantener entretenidos” servicios que no queremos que estén funcionando de forma correcta . También hemos visto como evitar que nuestra organización se encuentre en este tipo de situaciones, y en caso de encontrarlas, como reaccionar ante ellas.

También hemos aprendido a usar los IDS, los firewall y los honeypots para proteger nuestras redes de servicios. El siguiente punto es la experimentación: preparar entornos AISLADOS para poder probar las técnicas que hemos aprendido y los métodos de protección.

No debemos llevar a cabo ningún tipo de prueba sobre redes de producción, ya que podríamos ocasionar pérdidas económicas a los propietarios, ni probar sobre servicios que podamos encontrar en internet, ya que podría tener repercusiones penales sobre las personas que las efectúen.

Debemos usar el cuchillo que hemos estado afilando para ayudarnos a cortar la comida, no para apuñalar a la gente.

## 5.6 Bibliografía

**Sundar, García-Fernández, Isacoff, Spafford, Zamboni**: “An architecture for Intrusión Detection using Autonomous Agents”. 1998. COAST Laboratory. Purdue University.

**Spafford, Zamboni**: “A framework and prototype for a distributed intrusion detection system”. COAST Laboratory. Purdue University.

[1] http://www.ixbt.com/comm/wrls-smc-80211b.shtml
[2] http://www.drizzle.com/~aboba/IEEE/
[3] http://standards.ieee.org/getieee802/802.11.html
[4] http://neworder.box.sk