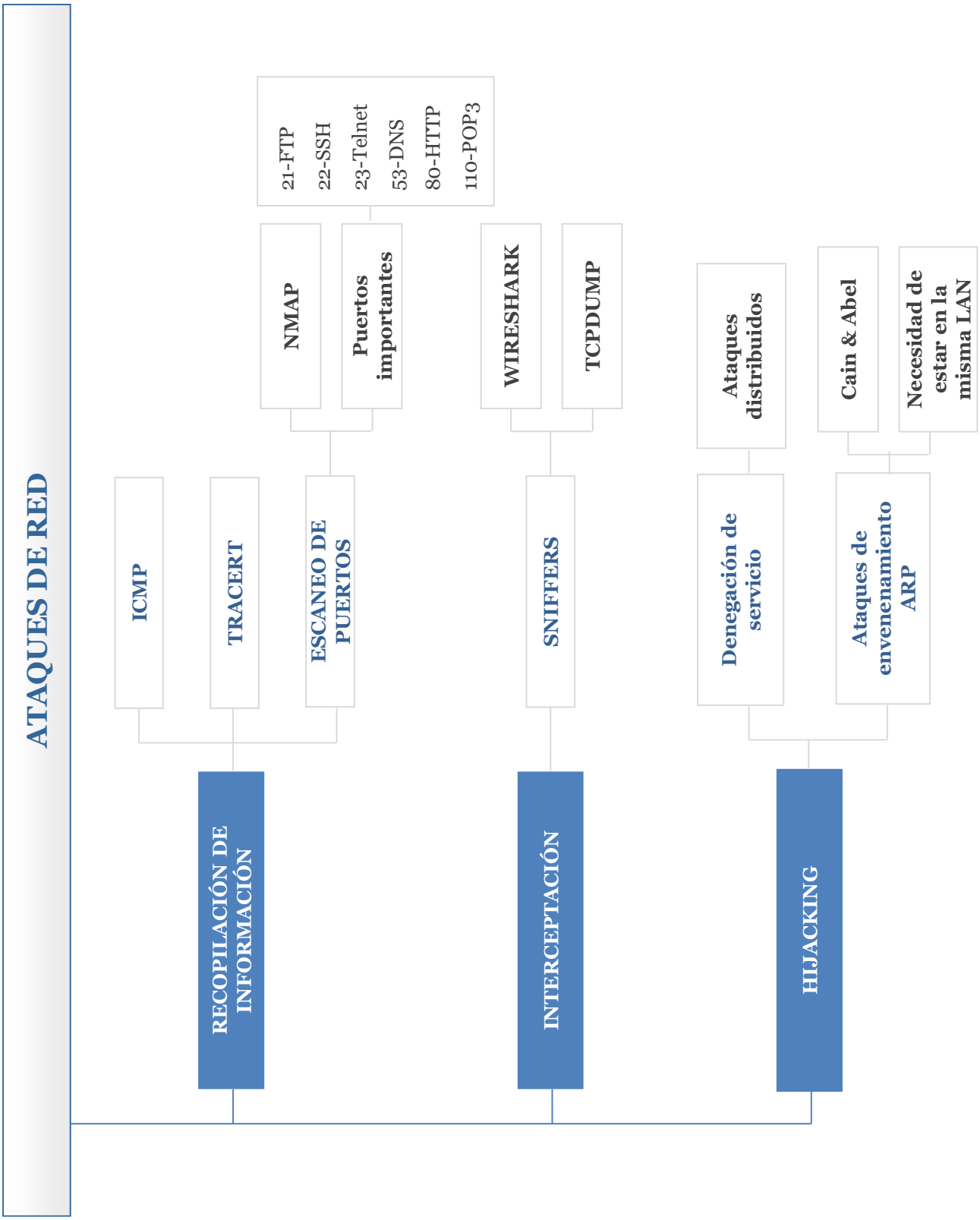
Ataques en redes

TEMA 5

Esquema


ATAQUES DE RED
RECOPILACIÓN DE INFORMACIÓN
ICMP
TRACERT
ESCANEO DE PUERTOS
NMAP
Puertos importantes
21-FTP
22-SSH
23-Telnet
53-DNS
80-HTTP
110-POP3
INTERCEPTACIÓN
SNIFFERS
WIRESHARK
TCPDUMP
HIJACKING
Denegación de servicio
Ataques distribuidos
Ataques de envenenamiento ARP
Cain & Abel
Necesidad de estar en la misma LAN

Ideas clave

5.1. ¿Cómo estudiar este tema?

El estudio de este tema se realiza a través de los contenidos desarrollados en las **Ideas clave** expuestas a continuación.

El objetivo de este tema es enseñar al alumno los principales ataques que tienen como objetivo una red, no una máquina concreta. ¿Cómo atacar ua red? Lo mas sencillo es hacer ver al alumno un ejemplo sencillo: se pueden inyectar paquetes en una red para provocar denegaciones del servicio sin ir a ninguna máquina en concreto. Por ello, aprender a monitorizar el tráfico de toda una red con sniffers capaz de detectar comportamientos anómalos s ehace esencial.

Otra cuestión fundamental es impedir o al menos detectar suplantaciones de identidad o spoofing, como ARP Spoofing, DNS Spoofing, DHCP SPoofing, etc.

En este tema realizaremos una **primera aproximación a la seguridad de las redes** de la mejor manera posible: **«aprendiendo a atacarlas»**. Comenzaremos estudiando una taxonomía completa de los tipos de ataques, sus características y principales contramedidas.

5.2. Amenazas y ataques de una red

Veamos la importancia de este tema con un caso real: has estudiado la asignatura de *Redes*, sabemos que en el establecimiento de la conexión entre dos máquinas a través del protocolo TCP está basado en el intercambio de tres segmentos. Los dos primeros llevan activos el *flag* SYN que, además de indicar este incio de conexión, las máquinas que intervienen en el proceso, se comunican sus capacidades, como tamaño del búfer, número de byte de inicio y tamaño del segmento de datos.

¿Y si fuéramos capaces de enviar *muchas* solicitudes de establecimiento de conexión contra una misma máquina?... la estaríamos obligando a que nos reservará una pequeña parte de memoria para nuestros envíos.

Si esas peticiones se meten en un bucle de miles de solicitudes de establecimiento de conexión, inutilizamos la máquina de destino por una denegación del servicio: hemos agotado sus recursos. Esto es un pequeño ejemplo de un ataque en red, en este caso concreto, se trata del llamado Ataque del SYN.

Obviamente todo proceso de implantación de contramedidas de seguridad debe comenzar por comprender claramente **cuáles son las amenazas reales** ante las que debemos defendernos.

En este tema analizaremos las **amenazas y vulnerabilidades** más importantes de las actuales redes de comunicaciones, junto con los **ataques más frecuentes y posibles contramedidas** que pueden aplicarse.

Buenos y malos: amenazas y contramedidas

Prácticamente todos los tipos de amenazas existentes actualmente pueden clasificarse en alguna de las siguientes grandes categorías:

- **Recopilación de información** (*harvesting*): el intruso suele comenzar su ataque buscando información acerca de la topología de la red, número y tipo de dispositivos presentes y su configuración. Con ello busca vulnerabilidades que pueda explotar y puntos de entrada vulnerables.
- **Interceptación de tráfico** (*sniffing*): el atacante captura el tráfico de forma pasiva, sin modificarlo para no ser detectado, en busca de contraseñas y cualquier otro tipo de información sensible que circula por la red.
- **Falsificación** (*spoofing*): el intruso quiere, en este caso, suplantar una identidad real, haciéndose pasar por una persona o equipo. Suele utilizarse para enmascararse el origen real de un ataque o para engañar a un sistema de control de acceso basado en la dirección IP de origen.
- **Secuestro de sesión** (*hijacking*): el atacante intercepta los datos, como en el sniffing, pero en esta ocasión puede también manipularlos, de forma que los hace pasar por legítimos. De esta forma, consigue hacer creer a las partes que están comunicándose con el equipo correcto, cuando en realidad se trata del equipo del atacante. A este tipo de ataques se les conoce también con el nombre de **hombre-en-el-medio** (*man-in-the-middle o MITM*).

- **Denegación de servicio** (*denial of service, DoS*): en este caso el objetivo es evitar que los usuarios legítimos puedan acceder a un servicio de la red, habitualmente inundándola con tráfico basura que consume todo su ancho de banda y recursos. Cuando este tipo de ataque se lleva a cabo de forma coordinada por muchos equipos simultáneamente (cientos o incluso miles de ellos) se llama ataque de **denegación de servicio distribuido** (*DDoS*, por sus siglas en inglés, *Distributed Denial of Service*).

Normalmente, cuando un atacante ha decidido su objetivo suele comenzar su actividad con la enumeración o exploración del sistema. Veamos a continuación este aspecto con mayor detalle.

5.3. Enumeración

El **objetivo** de esta primera fase es obvio: **obtener el máximo de información posible sobre el sistema**, tales como el número de equipos que lo componen, si son accesibles desde el exterior, servicios o aplicaciones que corren en cada uno, etc.

Con esta información el atacante podrá decidir qué equipos son más vulnerables y en cuáles merece la pena invertir más esfuerzo. Con esta razón, dado que **todos los sistemas expuestos en Internet son potenciales objetivos**, esta es una cuestión en la que un buen administrador de sistemas debería prestar especial atención.

ICMP

La prueba inicial más sencilla pero que siempre se lleva a cabo, es utilizar el **protocolo ICMP** para mandar un mensaje esperando recibir una respuesta (un eco) del sistema objetivo. Esta prueba puede realizarse simplemente con el comando *ping* (existente en *Windows* y *UNIX*), indicando la máquina destino. Observa la **Figura 1** para encontrar un ejemplo.

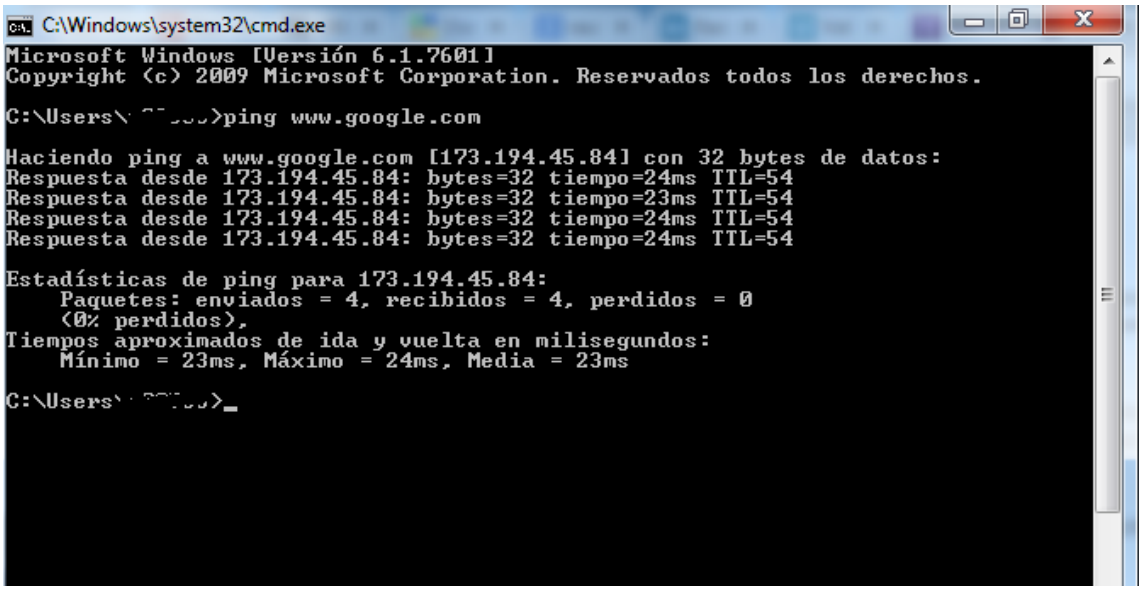


Figura 1. Petición ICMP (ping) a los servidores de Google

El tiempo de vida (la cadena TTL que se observa en la captura) es el acrónimo de **tiempo de vida** (*Time To Live*, en inglés), y sirve para conocer el número de saltos que ha dado el paquete por su viaje a través de las redes. Los paquetes se envían con un valor de TTL determinado, normalmente igual a 64, y en cada nodo que atraviesa se decrementa en una unidad. Su sentido es evitar que un paquete esté circulando eternamente por Internet, ya que en cuanto se alcanza el valor 0 se descartan. Por tanto, para saber por cuántos enrutadores pasó el eco, no hay más que restar 64 (a veces 128) del valor devuelto en la columna TTL.

Existen herramientas más potentes y completas para esta misma función (hay que recordar que ping nació como una herramienta simple de diagnóstico). Podemos citar a ***nping***, que se ejecuta también desde la línea de comandos, permite modificar el contenido de los paquetes y realizar múltiples peticiones simultáneas.

Accede a la herramienta a través del aula virtual o desde la siguiente dirección web:
http://www.nmap.org/nping

Tracert

Uno de los inconvenientes de la herramienta anterior es que podemos conocer el número de saltos, pero no cuáles son estos exactamente. Para ello podemos utilizar la herramienta ***tracert***, que envía varios paquetes con TTL desde 1 al número de saltos para que éstos vayan expirando en tránsito y así conocer el camino.

Este programa normalmente funciona enviando a la máquina destino un paquete UDP a un puerto que no esté a la escucha, por defecto el 33434, con el TTL a 1. Cuando el paquete llega al primer enrutador, este decrementa el TTL y al ser ya 0, descarta el paquete y notifica por medio de un paquete ICMP al equipo que lo envió que el tiempo de vida ha expirado. Así, *tracert* ya sabe la dirección del primer salto. A continuación, envía otro paquete, ahora con TTL a 2, luego a 3 y así sucesivamente. Finalmente, la máquina destino le devolverá un paquete ICMP informándole de que el puerto está cerrado, lo que indica al programa trazador que ha llegado al destino.

En la **Figura 2** puedes observar de nuevo un ejemplo de ejecución de este comando.

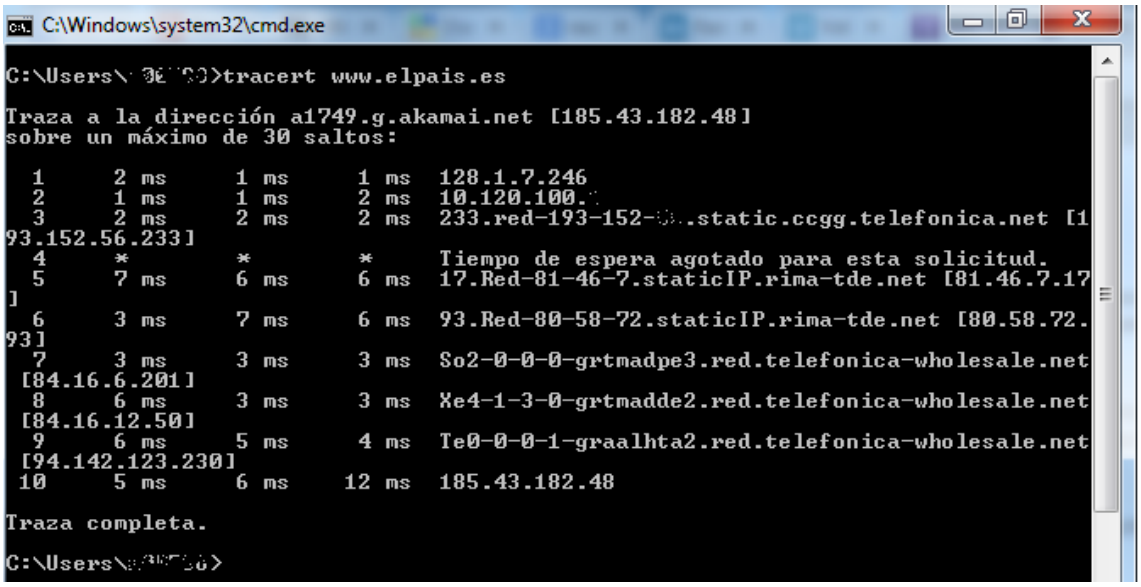


Figura 2. Ejemplo de ejecución de tracert a los sistemas del periódico EL PAIS

En la práctica, en sistemas *Windows* se suele utilizar más otra herramienta llamada ***pathping***, más potente que *tracert*, ya que combina lo mejor de esta y de *ping*, proporcionando más información.

Escaneo de puertos

El siguiente de los pasos que muy probablemente daría un atacante, una vez identificado el objetivo con las herramientas anteriores, es realizar un **escaneo de puertos**. Esta técnica pretende **conocer qué puertos TCP o UDP** están «escuchando» en la máquina objetivo, **para saber así los servicios asociados y poder atacarlos**.

No es necesario que conozcas los números de puertos de todos los servicios existentes (¡hay 65535!), pero sí de los más importantes. Los imprescindibles puedes encontrarlos en la
Tabla 1.

Número de puerto	Servicio asociado
21	FTP
22	SSH
23	Telnet
53	DNS (UDP)
80	HTTP
110	POP3
139	NetBIOS
443	HTTPS

Tabla 1. Lista de puertos TCP correspondientes a los servicios más comunes

Para llevar a cabo un escaneo de puertos se utilizan varias técnicas, pero la más básica es **enviar un paquete TCP con el bit SYN activado**, simulando querer iniciar una conexión. Recordarás que el sistema destino deberá contestar con los bits SYN y ACK en caso de que el puerto esté abierto y RST en caso contrario. En el caso de UDP, la respuesta se determina como puerto cerrado cuando el objetivo responde con un paquete *ICMP port unreachable* o se considera abierto si no se recibe este paquete.

La herramienta más conocida sin duda para realizar esta operación es ***Nmap***, en sus distintas versiones para *UNIX* y *Windows*, con y sin interfaz gráfico. Conviene que estés familiarizado con el uso de esta herramienta, pues se considera la «navaja suiza» mínima de cualquier administrador de sistemas o consultor de seguridad. Estudia sus opciones y aprende las más básicas.

Observa, por ejemplo, la **Figura 3.** En ella podrás ver el siguiente comando y su salida:

```
$ nmap -sS -F -O -sV host-de-destino
```

Estudia (a través de la ayuda del programa, disponible con la opción –h) estas opciones y entiende para qué sirven antes de memorizarlas.

Figura 3. Resultado de un escáner real de un equipo en Internet

Una última nota de atención antes de acabar. Recuerda que esta técnica suele ser el paso inicial de un atacante, por lo que **no realices escaneos de puertos a sistemas en los que no tengas autorización**. En ocasiones, dependiendo de la criticidad del equipo, existen detectores de este tipo de escaneos (que estudiaremos en próximos temas) que pueden estar monitorizándote.

5.4. Interceptación de tráfico: *sniffers*

El siguiente de los grandes bloques de amenazas en red es la **interceptación de tráfico**, o ***sniffing*** en inglés. Como su nombre indica, consiste en que un ataque intercepta el tráfico (que no va dirigido a él) que fluye en una red, pudiendo obtener así acceso a cualquier información sensible (que no haya sido previamente cifrada, por supuesto).

Este ataque es posible en gran medida debido al diseño de las **redes Ethernet**. Como sabes, estas redes de área local funcionan utilizando el **método del *broadcast***, en el que el tráfico se anuncia a todos los participantes en la red. Además, no existe a nivel de red autenticación de las peticiones, lo que abre también la puerta a otro tipo de ataques, que analizaremos en el **epígrafe 5.6**.

Por esta razón, con un simple programa que ponga la tarjeta de red en modo promiscuo (es decir, que recibe todo el tráfico de la red, no sólo el que va dirigido a dicha tarjeta) tendremos acceso inmediatamente a los datos que estén en nuestro mismo *dominio de difusión*. Por supuesto, aunque estas herramientas pueden (y suelen) utilizarse para el ataque, muchas veces se utilizan también como métodos de diagnóstico. Particularmente para:

- **Analizar en tiempo real el tráfico de una red**, para encontrar la causa de algún problema (pérdida de conectividad con Internet, por ejemplo).
- **Realizar un «perfilado» y obtener estadísticas del tipo y volumen del tráfico.**
- **Aprender la estructura de protocolos de red desconocidos, o no documentados.**

Existen multitud de programas que llevan a cabo esta tarea, aunque destaca claramente la **herramienta *Wireshark***, disponible en entornos *UNIX* y *Windows*. En la **Figura 4** puedes encontrar una captura del tráfico habitual en una LAN (en este caso, una navegación Web).

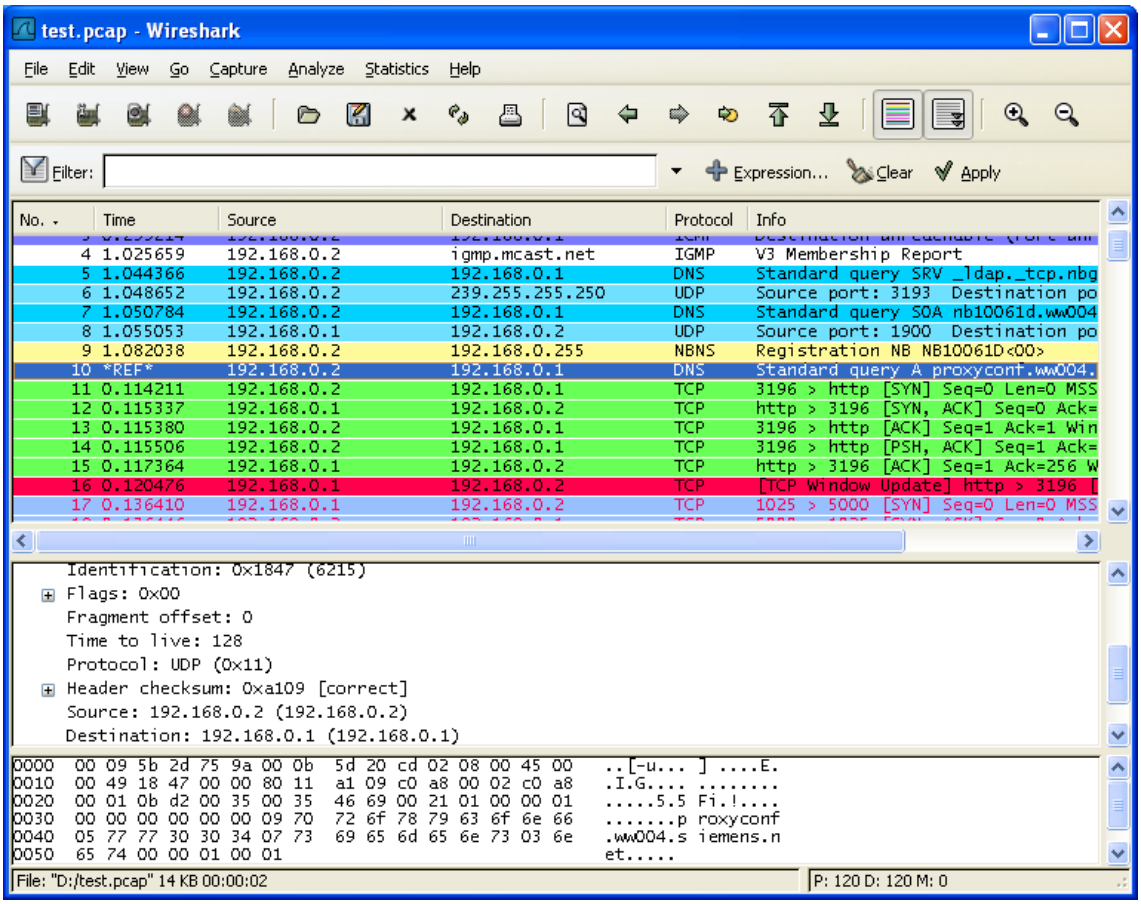


Figura 4. Captura de ejemplo de Wireshark

5.5. Ataques de denegación de servicio

Con el paso del tiempo y considerando la evolución que ha sufrido Internet, el enfoque sobre posibles ataques ya no es tanto el intentar acceder a un sistema remoto, sino el imposibilitar su acceso. De cara a una empresa, quizás sea más costoso el que su sistema permanezca inaccesible a que accedan al mismo. Sin considerar que cada día resulta más complejo asaltar sistemas considerados de interés. Es por ello que los *hackers* adoptan una nueva estrategia de ataque: provocar la denegación de servicio o imposibilidad de prestar el servicio del sistema atacado.

La denegación de servicio, por tanto, pretende **impedir que los usuarios legítimos de un sistema puedan acceder a él y, por consiguiente, a los servicios que proporciona**. En las circunstancias actuales de globalización, el daño económico y de imagen que sufre una empresa por un ataque de este tipo probablemente sea mucho mayor que el derivado de una simple intrusión.

Y esto, desde luego, no debe considerarse un ataque de último recurso o juegos de *script-kidies*: las pérdidas económicas de las que hablamos son elevadísimas, como vamos a ver a continuación.

Orígenes históricos

Los ataques de denegación de servicio distribuidos (*DDoS* en inglés) se han hecho muy populares últimamente entre la comunidad *hacker* en Internet, aunque sus orígenes son bastante anteriores. En la semana del 7 de Febrero de 2000 se produjo su presentación en sociedad. Ese día *Yahoo!* fue atacado y durante 3 horas, su portal estuvo inaccesible. Al día siguiente, les tocó el turno a *Amazon*, *Oz.net*, *eBay*, *Etrade* y *Buy.com*, aunque estos ataques no fueron los primeros dirigidos a sitios comerciales: un año antes, *eBay* estuvo fuera de servicio durante 22 horas por un ataque *DoS*, que supuso que la compañía tuviese que reconstruir y reordenar todo el fichero de clientes.

Después vinieron ataques a *American Online*, *New York Times*, servidores de IRC (el 17 de febrero) y uno que consiguió «tumbar» los ordenadores del FBI el 25 de febrero.

Los analistas estiman que durante las tres horas que *Yahoo!* estuvo caído, sufrió unas pérdidas de alrededor de 500.000 dólares. *Amazon.com*, por su parte, tuvo unas pérdidas aproximadas de 600.000 dólares en las casi 10 horas que estuvo fuera de combate. Durante los ataques, Buy.com cayó de un 100% de accesibilidad a un exiguo 9.4%. Por otro lado, apenas un 5% de los usuarios podía conectar con el sitio de *CNN.com* mientras sufría los ataques. Aunque la peor parte la llevaron *Zdnet.com* y *Etrade.com* que fueron virtualmente inaccesibles.

Sin duda, **el tipo de ataque de denegación de servicio más conocido es la llamada inundación SYN del protocolo TCP** (*SYN flooding*). Básicamente, consiste en que el cliente manda una petición de conexión al servidor, anunciando su intención de establecer una conexión. El servidor responde con una confirmación (*ACKnowledge*), aceptando el establecimiento de la conexión y reservando un slot en la cola de peticiones pendientes de finalizar. El ataque está servido: el cliente no tiene más que no responder a esta confirmación y repetir la petición de conexión. El resultado final es el desbordamiento de la cola de peticiones pendientes, y la consiguiente denegación de servicio a los clientes legítimos que intenten conectarse al servidor.

Un primer vistazo al problema

Un problema obvio en el escenario de un ataque SYN es que todas las comunicaciones preliminares se llevan a cabo **antes** de cualquier forma de autenticación y, por tanto, el servidor no tiene forma de distinguir una petición legítima de una que no lo es. Desgraciadamente, no hay mucho que podamos hacer para evitar esto, pues tratar de poner los mecanismos de autenticación como primer paso de la comunicación, nos llevaría a otro ataque de denegación de servicio en sí mismo, pues el servidor gastaría todo su tiempo verificando firmas digitales, por ejemplo.

Una causa menos obvia de ataque de denegación de servicio tiene que ver con la **falta de control sobre los recursos consumidos por cada cliente**. En general, se considera que existen **tres factores claves para protegerse contra este tipo de ataques**:

- **Control:** sobre todos los recursos consumidos por los clientes.
- **Detección:** cuando los recursos consumidos por algún cliente sobrepasan un límite preestablecido.
- **Contención**: pudiendo reclamar los recursos consumidos tras detectar un ataque dedicando unos recursos mínimos por parte del servidor en esta tarea.

Esto tiene que ver con cómo fue originalmente diseñado Internet: con muy poca preocupación sobre cómo los recursos consumidos son contabilizados y controlados. De hecho, la contabilización de recursos fue la prioridad global más baja en el diseño inicial (al contrario, por ejemplo, de la red telefónica, donde era una prioridad crítica).

Otro factor importante que ayuda a que este tipo de ataques tengan éxito y sean efectivos, es el malo e ineficiente proceso de los paquetes que llegan a los servidores bajo fuertes condiciones de carga. Muchos sistemas operativos modernos utilizan subsistemas de red que utilizan interrupciones, que han demostrado carecer de eficacia y estabilidad bajo condiciones de congestión en la red.

El problema viene del hecho de que estos subsistemas dan mayor prioridad a los paquetes entrantes de la red, sin importar de a qué aplicación corresponden, si esta aplicación está ejecutando o no o si tiene una prioridad más baja que la aplicación que está ejecutándose actualmente.

Por tanto, podría darse una **situación de *livelock***, donde el servidor gasta gran parte de sus recursos en aceptar paquetes de la red, para después descartarlos porque no hay tiempo de CPU que dar a las aplicaciones.

Es decir, con los actuales mecanismos de gestión de la red, los paquetes son descartados sólo después de que los recursos del servidor ya han sido desperdiciados, que ayuda claramente a que se produzcan ataques de denegación de servicio.

Análisis de las posibles contramedidas

Actualmente no existe una solución clara y completa a este tipo de ataques. Sin embargo, hay multitud de soluciones parciales que pueden paliar bastante el problema. Podemos clasificarlas así:

1. Filtros en routers y firewalls
- Firewall funcionando como un real - Firewall funcionando como un gateway semi-transparente - Filtros de entrada y salida en routers - Desactivar la amplificación de peticiones de broadcast
2. Mejoras en los sistemas operativos
- Fuerza bruta - Eliminación aleatoria de peticiones
3. Mejoras en los protocolos
- SYN cookies - Protocolos sin estado - Protocolos de puzzles de clientes

Veamos estos puntos en mayor detalle.

1. Filtros en routers y firewalls

Esta solución trata de mitigar el problema utilizando una solución sencilla y fácil de implementar: **utilizar filtros de entrada y salida en routers y firewalls**. Vamos a ver que pueden ser configurados de varias maneras, cada una de las cuales con sus ventajas e inconvenientes.

Firewall funcionando como un intermediario (proxy)

En esta configuración, el firewall responde en nombre del equipo al que protege. Una conexión con el equipo final se establece sólo tras haber hecho lo propio con el firewall.

Durante un ataque, el firewall responde al SYN enviado por el atacante; como el ACK nunca llega a éste, el firewall termina la conexión con un paquete RST, de manera que el equipo nunca recibe el paquete. Para conexiones legítimas, el firewall crea una nueva conexión con el equipo en nombre del cliente, y continúa actuando como un proxy para transformar los números de secuencia de los paquetes que fluyen entre el cliente y el servidor.

Ventajas: el servidor está completamente blindado de ataques DoS, y nunca recibe paquetes SYN falsos.

Inconvenientes: se introducen nuevos retrasos para los clientes legítimos.

Firewall funcionando como un gateway semi-transparente

En este caso, el firewall pasa los paquetes SYN al servidor. Cuando éste responde con un paquete SYN+ACK, el firewall lo pasa al cliente y manda un paquete ACK (pre-confirmación) al servidor, de manera que éste cree que la conexión ya se ha establecido. Si el firewall no recibe un ACK legítimo del cliente en un periodo determinado de tiempo, se manda un paquete RST al servidor para acabar con la conexión. En el caso de un cliente legítimo, el ACK duplicado que llega al servidor es descartado por el protocolo TCP, y los paquetes, en adelante, fluyen sin intervención del firewall.

Ventajas: no se introducen retrasos para los clientes legítimos.

Inconvenientes: el periodo de espera (time out) debe ser establecido con cuidado, de manera que no se deniegue el acceso a conexiones legítimas con tiempos largos de latencia.

Filtros de entrada en routers

Es muy común que la dirección IP de los paquetes utilizados en un ataque DoS esté falsificada, de manera que sea difícil saber de dónde partió el ataque. Esto podría evitarse si los routers comprobasen que los paquetes de entrada que reciben tienen la dirección IP que deberían tener, utilizando la máscara de subred correspondiente.

Ventajas: realmente evitaría los ataques DoS con direcciones de origen falsas.

Inconvenientes: si las direcciones de origen son legítimas, esta técnica es inútil. Además, podría causar muchos problemas a los servicios de IP dinámica.

Filtros de salida en routers

En este caso, se trata de evitar que los paquetes de salida abandonen nuestra red con direcciones de origen falsas, y nuestra infraestructura se utilice en el ataque DoS.

Ventajas: es útil si el router está muy cerca de la víctima; en ese caso, es efectivo.

Inconvenientes: es difícil de aplicar por los proveedores de servicio (ISP), porque a menudo necesitan enviar tráfico legítimo que no forma parte de su espacio de direcciones.

Desactivar la amplificación de peticiones de broadcast

Una red puede actuar como un amplificador para inundar otras redes con ataques de denegación de servicio.

Ventajas: junto con los filtros de salida, esta técnica puede prevenir que se use nuestra red como origen de ataques «smurf».

Inconvenientes: la amplificación de broadcast es una herramienta de diagnóstico bastante útil. Sin ella, el servidor WINS de una red no recibirá el broadcast, lo que causará que algunos servicios de resolución de nombres de Windows no funcionen correctamente.

2. Mejoras en los sistemas operativos

Esta aproximación trata de abordar el problema, mejorando los inconvenientes de los sistemas operativos que los hacen vulnerables a estos ataques: normalmente, colas de datos extenuadas o un excesivo tiempo de CPU consumido.

Fuerza bruta

A veces algunas soluciones, o más bien contramedidas, pasan por la fuerza bruta; es decir, aumentar los recursos de memoria o CPU de un servidor para que sea capaz de hacer frente a un ataque de denegación de servicio. Obviamente, siempre se podrá crear un ataque tan potente que sea capaz de sobrepasar cualquier capacidad, por grande que ésta sea. Ya hemos visto ejemplos de ataques recientes a sitios con tanta capacidad de cómputo o ancho de banda como *Facebook*.

Ventajas: estas mejoras son relativamente fáciles de implementar, y los ataques DoS tienen menos posibles de resultar exitosos porque necesitarían un ratio de paquetes/segundo que superaría los anchos de banda más usuales.

Inconvenientes: el tiempo de respuesta del servidor podría ser más lento debido al gran tamaño de la cola de peticiones, donde tiene que buscar.

Eliminación aleatoria de peticiones

Alan Cox, uno de los gurús de Linux, propuso un cambio a la pila TCP de Linux para evitar ataques SYN, eliminando aleatoriamente slots de la cola de peticiones cuando esta esté llena. El algoritmo puede elegir un slot al azar, seleccionar el más viejo o una combinación de ambos.

Ventajas: funciona bien tanto con trafico bajo como alto manteniendo una tasa de perdida de rendimiento menor del 10%, incluso a tasas muy altas de ataque SYN.

Inconvenientes: un atacante puede, ocasionalmente, provocar que se le deniegue la conexión a un cliente legítimo.

3. Mejoras en los protocolos

Dado que los ataques SYN son posibles gracias a una debilidad en el protocolo TCP, parece lógico pensar que la solución más adecuada se obtendría al cambiar el protocolo para hacerlo más resistente a este tipo de ataques.

SYN Cookies

Este tipo de defensa funciona de la siguiente manera: cuando un cliente manda el paquete SYN, el servidor calcula un *hash* del número de secuencia del paquete, los puertos, una clave secreta del servidor y un contador que cambia cada minuto. El servidor manda el resultado al cliente, sin reservar ningún espacio en memoria para esta petición.

Cuando el cliente responde con un paquete ACK, el servidor recalcula la misma función hash y tira aquellos paquetes que no verifiquen dicha función. Si el resultado es el esperado, se reserva espacio para esta petición y la conexión se establece normalmente.

Ventajas: la cola de peticiones nunca se agota por ataques SYN, pues, en realidad, ni siquiera existe como tal.

Inconvenientes: en caso de que el paquete no verifique la autenticación sobre él, el servidor no contesta con un paquete SYN+ACK, lo que rompe claramente con la estructura del protocolo TCP.

Protocolos de puzzles de tiempo

Para prevenir el *spam* en los servidores de correo, los investigadores Dwork y Naor propusieron un sistema según el cual el remitente de un correo debía calcular con anterioridad una función 'moderadamente difícil' de computar o una *delay function*; el coste de computar estas funciones es insignificante para los usuarios normales, pero muy alta para los que pretenden realizar *spam*.

Juels y Brainard ampliaron la idea de manera que, si un servidor sospecha que está sufriendo un ataque DoS, se mandan al cliente pequeños retos criptográficos que deben resolver. Sólo si lo hacen correctamente la conexión se establece finalmente.

Ventajas: los retos pueden tener distintos niveles de dificultad, de manera que ésta puede ir creciendo en consonancia con la severidad del ataque.

Inconvenientes: requieren un software en el lado del cliente capaz de resolver el puzzle.

5.6. Ataques de envenenamiento ARP

La siguiente gran categoría de ataques de red que vamos a estudiar son los que se realizan **dentro de una red local**, concretamente **contra el protocolo ARP**. Este protocolo ARP (*Address Resolution Protocol*) sirve, básicamente, para traducir direcciones IP en direcciones físicas (habitualmente, Ethernet).

Cuando una máquina quiere comunicarse con otra de la que únicamente conoce su dirección IP, actúa de la siguiente manera:

1. Lanza una petición a la red en forma de broadcast con la pregunta: **«¿Quién tiene la dirección IP x.x.x.x?»** Esto constituye una ***petición ARP***.
2. La máquina cuya dirección IP coincide con la pedida responde: **«Yo tengo esa IP y mi dirección física (MAC) es x:x:x:x:x:x.»**. Esto se llama ***respuesta ARP***.

Todos los dispositivos de red cuentan con lo que se conoce como una ***caché ARP***: una memoria con caducidad en la que se definen todos los pares de direcciones IP-direcciones MAC conocidos. Este caché sirve para no repetir peticiones ARP para dispositivos con los que ya ha habido una comunicación previa.

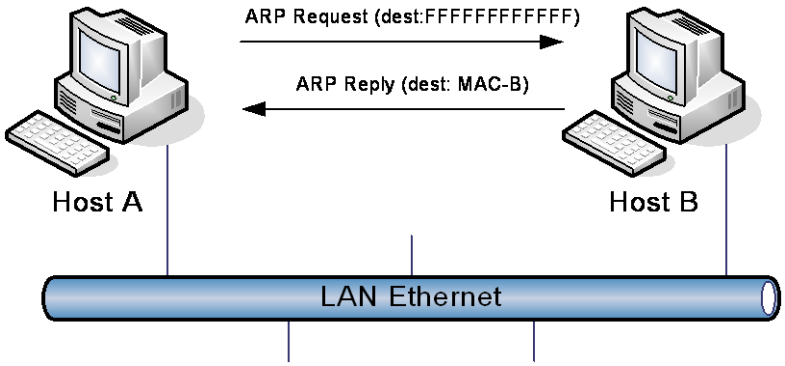


Figura 5. Petición ARP

Como puede apreciarse en el esquema del punto anterior, las respuestas ARP no están autenticadas. Esto quiere decir que una máquina puede responder en nombre de otra legítima, «enmascarándola». En esto, básicamente, consiste el ataque de envenenamiento ARP: ***modificar el contenido del caché ARP de los dispositivos de red*** de manera que un atacante puede hacerse con el control del tráfico de una red.

Este **«control»** puede tomar una de las siguientes formas:

- **Denegación de servicio:** un atacante puede asociar una dirección IP legítima a una dirección MAC falsa. Por ejemplo, el atacante podría enviar una petición ARP asociando la dirección IP del router de la red con una dirección MAC que no exista. Las máquinas enviarán sus paquetes al gateway por defecto, como habitualmente, pero los paquetes se perderán. De esta forma, el atacante puede dejar sin acceso a Internet a toda la red.
- **Man-in-the-middle:** esta es la versión más conocida del ataque, aunque no la única, como erróneamente se cree. Consiste en modificar las cachés ARP de distintos dispositivos (switches, routers o máquinas) para interceptar el tráfico entre dos de estos dispositivos. Lo más peligroso de esta técnica es que resulta prácticamente indetectable.
- **Inundación MAC:** este ataque va dirigido únicamente contra los switches presentes en una red. Cuando ciertos switches se ven sobrecargados por el volumen del tráfico que gestionan caen a menudo a modo «hub». En este modo, el switch está demasiado ocupado como para realizar las comprobaciones de seguridad en los distintos puertos y se limita a difundir todo el tráfico de cada ordenador de la red. En ese momento, puede accederse al tráfico de cualquier máquina.

Evidentemente, este ataque es el menos utilizado por lo «ruidoso» y fácilmente detectable que resulta. Además, no funciona con todos los switches y el volumen de datos necesario para provocar la inundación MAC puede ser demasiado elevado para conseguirlo.

Requisitos del ataque

Para que un ataque de este tipo pueda llevarse a cabo, deben cumplirse las siguientes **condiciones**:

- **El atacante debe estar en el mismo dominio de difusión que las máquinas que pretende atacar**, es decir, en la misma LAN; no funcionará contra otras subredes o VLAN's.
- **El atacante debe ser capaz de re-enrutar los paquetes a su destino original** o las máquinas atacadas no serán capaz de comunicarse y se percatarán del ataque.
- Para cumplir el punto anterior, **el atacante debe conocer los pares MAC-IP que le interesan antes de lanzar el ataque**.

Otras características de este ataque son:

- **Este ataque ralentizará el tráfico de las máquinas atacadas**, pues todo ese tráfico es gestionado por una única máquina (la del atacante).
- **Esta técnica no inserta una nueva entrada ARP en las cachés**, sino que modifica las existentes. Por tanto, no se puede atacar a una máquina que todavía no haya generado tráfico (aunque este supuesto es realmente poco probable).

Herramientas

Si las consecuencias del ataque en el ámbito local son bastante serias, estas se agravan por la cantidad de herramientas existentes para realizarlo y sobre todo, por la facilidad de su manejo.

La herramienta más conocida es, sin duda, ***Caín*** (disponible en: http://www.oxid.it) disponible para entornos *Windows*. No hacen falta conocimientos profundos en redes; de hecho, ni siquiera entender con claridad en qué consiste el ataque. Puedes ver su aspecto en la **Figura 6**.

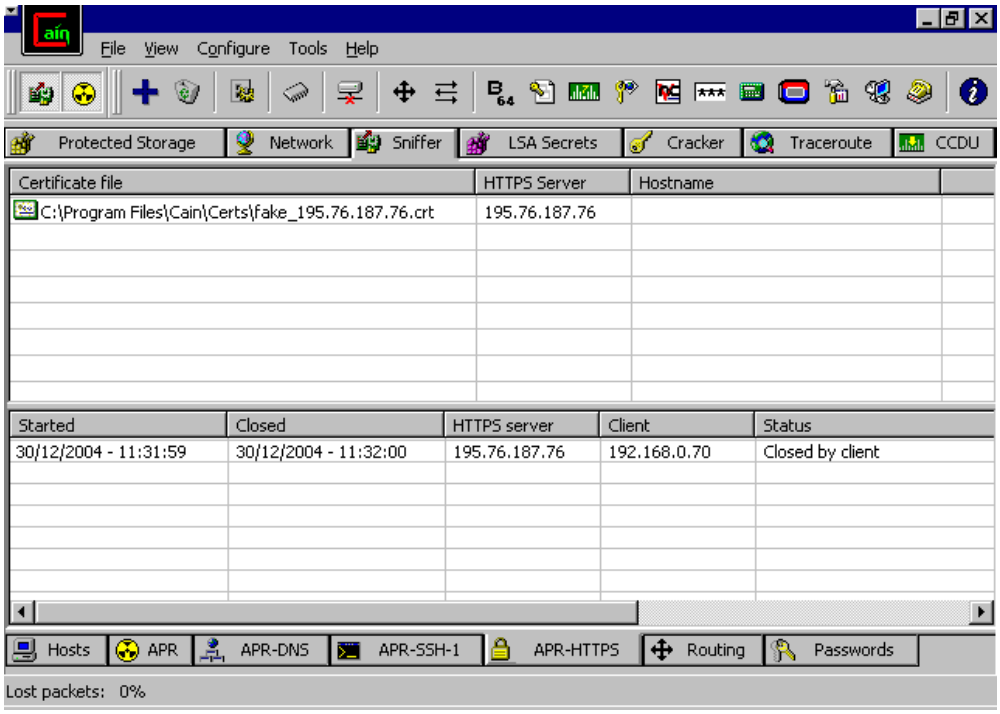


Figura 6. Herramienta de ataque Caín&Abel

Para entornos *UNIX* la herramienta más utilizada es ***Ettercap***, bastante más complicada en su instalación y manejo, aunque de funcionalidad superior a ***Caín***.

Escenarios de ataque

Como hemos visto, la única condición para que el ataque tenga éxito es que se tenga acceso al tráfico de la persona que se desea atacar. Evidentemente, esto incluye cualquier LAN o, en general, cualquier red que esté en el mismo dominio de difusión.

- **Redes corporativas.** Según un estudio muy conocido, el 70% de los ataques que sufre una compañía provienen, directa o indirectamente, de trabajadores de la misma. Por tanto, las redes corporativas pueden considerarse el escenario más común para este ataque, dada la confianza existente en estas redes y la viabilidad técnica de llevarlo a cabo.
- **Accesos inalámbricos.** De un tiempo a esta parte ha surgido otro escenario que es especialmente peligroso: aquellos clientes que se conectan a Internet a través de un enlace inalámbrico.

Estas son **algunas de las posibilidades**:

- Cada vez más IPS's ofertan accesos que incluyen un router o módem inalámbrico que, como es públicamente conocido, sufren graves problemas de seguridad que los hacen altamente inseguros. Con un equipo básico (una antena omnidireccional y una tarjeta inalámbrica de coste aproximado de 150€) un atacante puede acceder a redes inalámbricas desde una distancia de hasta centenares de metros en condiciones ideales. Un cliente, por ejemplo, podría sufrir el ataque en su propia casa sin poder detectar la fuente.
- Las zonas wi-fi de diversas compañías (sobre todo *Telefónica*) son cada vez más comunes. Suelen encontrarse en aeropuertos, estaciones de tren, hoteles, etc… y, por definición, tienen baja o nula seguridad.

Contramedidas y recomendaciones

Técnicamente hablando, debido al diseño del protocolo ARP **es prácticamente imposible desarrollar una contramedida efectiva**. Una adecuada segmentación de red y herramientas específicas de monitorización (como ARPWatch) pueden ayudar a mitigar la situación en entornos especialmente críticos, pero resultan inútiles en redes muy grandes o con servicios como DHCP.

Caso de estudio: consecuencias en banca electrónica

Uno de los mayores peligros de este ataque es cuando se utiliza contra el protocolo SSL. Éste se utiliza como mecanismo de autenticación e integridad de los datos enviados y recibidos de los servidores Web y, entre otros muchísimos, en aquellos que alojan las aplicaciones de banca electrónica.

Utilizando **envenenamiento ARP** es factible poder tener acceso al tráfico *en claro* de un cliente conectándose a un servicio de banca electrónica aún cuando éste viaje cifrado. Para conseguirlo, se utiliza un ataque de envenenamiento ARP combinado con uno de suplantación:

1. **El atacante lleva a cabo un ataque de envenenamiento ARP sobre el cliente que quiere atacar** (debe tener acceso a su tráfico, es decir, debe estar en su misma red local).

2. Acto seguido, **realiza un ataque de suplantación presentando un certificado falso para poder descifrar el tráfico SSL**. Este certificado es generado por Caín y es idéntico al suplantado excepto, claro está, porque la firma de la CA correspondiente sobre él no se verifica.
3. **Desde ese momento, toda comunicación SSL del cliente puede ser leída y modificada por el atacante**.

Este ataque, peligroso de por sí, se ve agravado por el hecho de que los navegadores Web no son, en general, muy claros a la hora de explicar las razones por las que el certificado del servidor no se verifica. Este es, por ejemplo, el mensaje mostrado en el paso 2 del ataque por un navegador *Internet Explorer*:

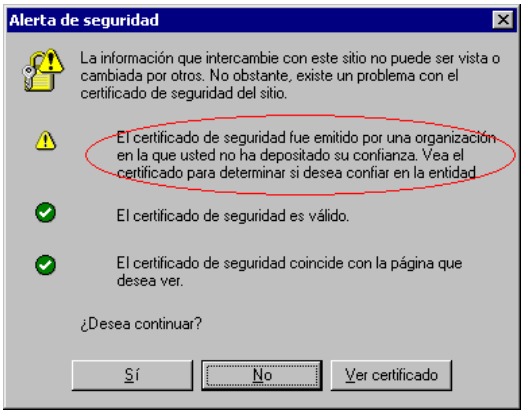


Figura 7. Mensaje de Internet Explorer tras el ataque de suplantación

Y esta es la copia que hace *Caín* del certificado. Como puede observarse, la copia es perfecta para una persona sin los conocimientos adecuados:

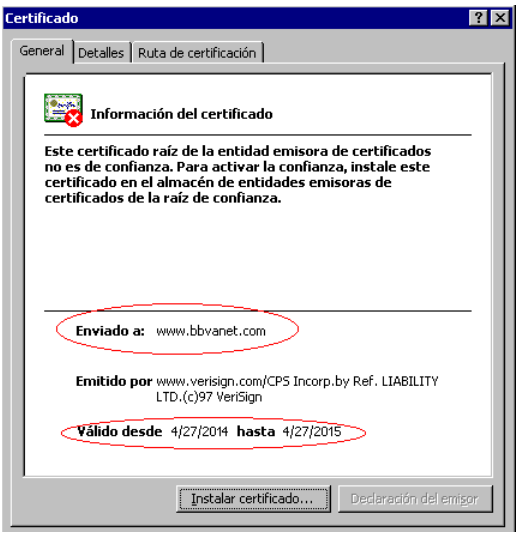


Figura 8. Copia de un certificado legítimo llevada a cabo por Caín

Como se comprueba en la **Figura 7**, solo una de las tres comprobaciones de seguridad no se verifica, lo que puede inducir al usuario a aceptar el certificado. Las comprobaciones que sí resultan correctas son:

- **El certificado de seguridad es válido:** esto significa únicamente que el certificado todavía no ha caducado (se refiere a su validez en el tiempo). Y esta validez temporal se puede establecer arbitrariamente en la copia del certificado, como hemos visto, con lo que esta comprobación no es garantía de seguridad. Sin embargo, un usuario puede entender que el certificado se verifica y es correcto. Obviamente la redacción de este mensaje es muy mejorable, pues induce claramente a error.
- **El certificado de seguridad coincide con la página que desea ver:** esta comprobación hace referencia a la coincidencia o no del campo «Emitido a» y la URL de la página que se está visitando. Tampoco es garantía de seguridad, por la misma razón que en el punto anterior.

La comprobación que no resulta correcta, la primera que se muestra, es la más importante. Significa que el sistema operativo no reconoce la CA que ha firmado el certificado que se está presentando. Evidentemente, esto es así, pues lo contrario significaría que se ha encontrado la forma de romper el principio básico en el que se basan todas las infraestructuras PKI actuales.

Un problema con este mensaje, sin embargo, es que los usuarios lo ven demasiado a menudo, por causas completamente distintas a las que estamos viendo, y eso provoca que dejen de considerarlo peligroso. Por ejemplo, un sitio Web cuyo certificado de servidor estuviera firmado por una CA que no estuviera incluido en el almacén de confianza de Windows provocaría este mensaje.

Otra razón muy habitual es que los certificados de un servidor están vinculados a una URL concreta, por lo que si ésta cambia o es incorrecta volverá a aparecer este mensaje. Este problama ocurre en sitios muy conocidos o que, al menos, deberían ser conscientes de este problema. Observa, por ejemplo, la **Figura 9** y comprueba como el sitio de denuncias de la Policía española tiene un certificado firmado por la CA raíz de la Dirección General de la Policía (a la derecha), que no está incluida por defecto en el almacén de certificados de confianza del sistema operativo (Windows en este caso).

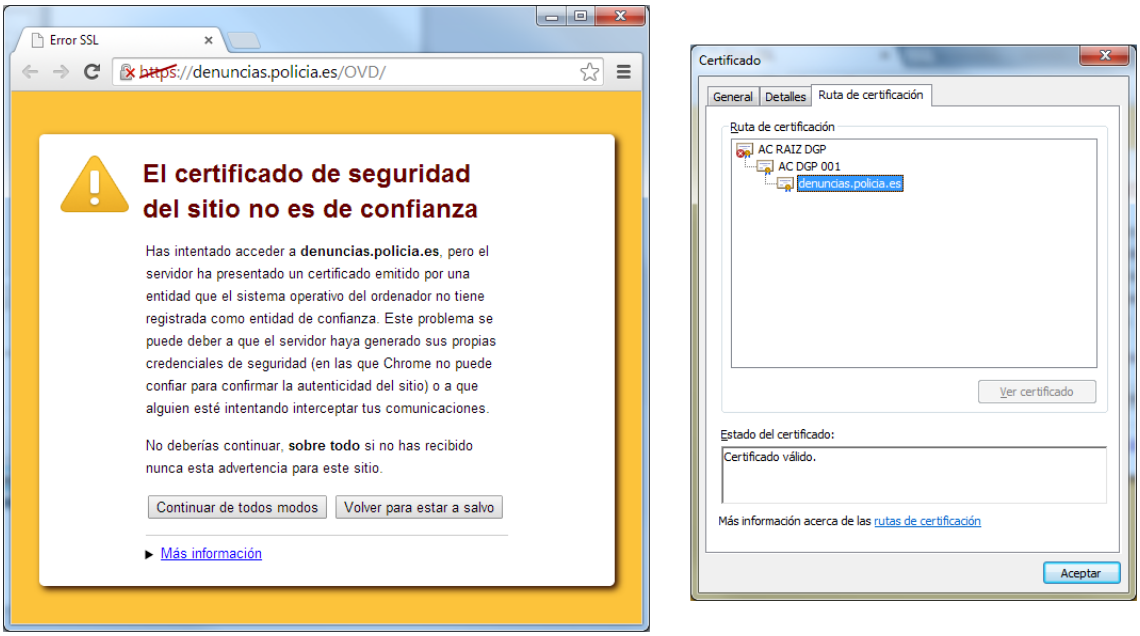


Figura 9. Fallo de comprobación de un certificado: la CA que lo firma no está incluida en el almacén de certificados de confianza de Windows

En consecuencia, **no es posible distinguir este caso de un intento de ataque**, circunstancia que podría ser aprovechada por un atacante.

En resumen, podemos decir que los usuarios se han acostumbrado a ignorar estos mensajes, unas veces justificadamente debido a fallos de configuración en los servidores y otras, por lo poco adecuados que son los mensajes de error.

Finalmente, es importante comprender que **este no es un ataque dirigido contra SSL** y que, por supuesto, no significa que éste haya sido comprometido. En realidad, se trata más de un ataque de ingeniería social, que aprovecha la falta de conocimiento de la mayoría de los usuarios y, sobre todo, la forma muy mejorable en la que los navegadores Web informan de problemas en los certificados de una conexión SSL.

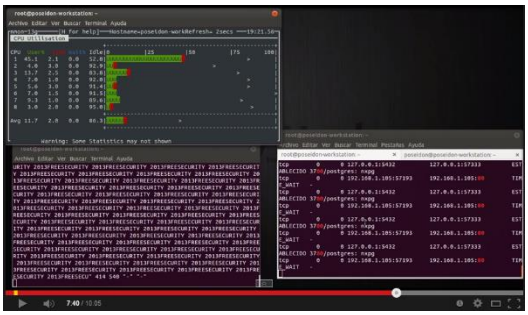
La experiencia nos dice que, ante esta situación, una gran mayoría de los usuarios continuarían con la transacción, quedando desde ese punto, expuestos al ataque.

Lo + recomendado

No dejes de ver...

Denegación de servicio distribuido (DDoS)

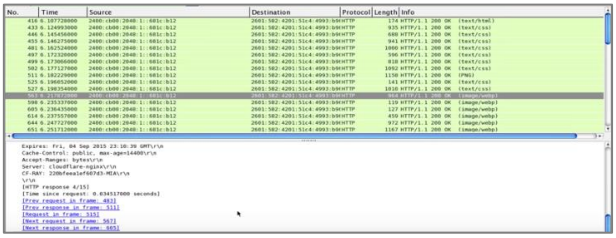
En este vídeo puede observarse cómo se llevan a cabo los ataques de denegación distribuidos con herramientas reales, como hping3, LOIC o slowloris, en este caso contra un servidor web *Apache*.



Accede al vídeo a través del aula virtual o desde la siguiente dirección web:
https://www.youtube.com/watch?v=7sZB_VzdkOg

Aprendizaje de sniffers con wireshark

En este vídeo podemos aprender cómo usar el sniffer por excelencia: wireshark.



Accede al vídeo a través del aula virtual o desde la siguiente dirección web:
https://www.youtube.com/watch?v=TkCSr30UojM

Ataques a redes Wifi WPA y WPA2

Existe una estupenda herramienta gratuita, denominada *Wifislax*, que se utiliza habitualmente para realizar auditorías y buscar vulnerabilidades en redes wireless. En este video se explica paso a paso su funcionamiento, mostrando sus principales características, de una manera amena y fácil de seguir.

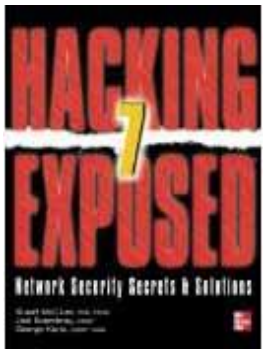
Accede al vídeo a través del aula virtual o desde la siguiente dirección web:
https://www.youtube.com/watch?v=1rVnq3YSbZo

+ Información

A fondo

Hacking Exposed 7

Stuart McClure, S. (2012). *Hacking Exposed 7*. McGraw-Hill. ISBN: 0071780289

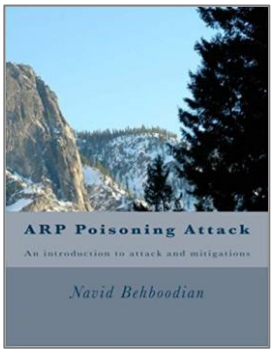


Este libro es sin duda, uno de los mejores y más completos de los escritos sobre hacking en redes. Se trata ya de la séptima edición, por lo que los autores lo han ido refinando y actualizando a lo largo de los años. En esta última edición se incluyen un amplio rango de ataques, incluyendo el ataque de redes inalámbricas, el ataque de cierto tipo de hardware o protocolos de VozIP.

ARP Poisoning Attack: An introduction to attack and mitigations

Behboodian, S. (2012). *ARP Poisoning Attack: An introduction to attack and mitigations*. CreateSpace Independent Publishing Platform. ISBN: 978-1468068511.

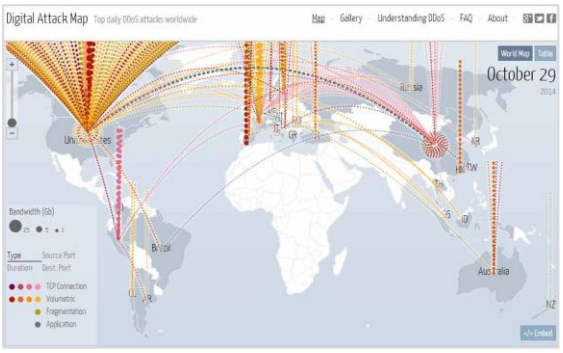
Este libro ilustra una descripción general del ataque de envenenamiento ARP y las soluciones actuales para detectar y proteger contra las direcciones de suplantación de identidad de los atacantes.



Enlaces relacionados

Mapa de ataques

En la página Web indicada más abajo, podrás encontrar una estupenda iniciativa de Google: un mapa digital que visualiza en tiempo real los ataques DDoS que se están produciendo en el mundo. Puedes seleccionar, también, cualquier fecha para ver qué ocurrió en el pasado, países más atacados y más originarios de ataques, detalles técnicos sobre los ataques, etc.



Accede a la web a través del aula virtual o desde la siguiente dirección:
http://www.digitalattackmap.com/

Visualización del ataque ARP

En la página Web que encontrarás abajo, que aloja la conocida herramienta Cain&Abel, se encuentra una conocida animación que explica perfectamente cómo funciona el ataque de envenenamiento de ARP. De forma gráfica y sencilla muestra cada uno de los pasos del ataque, las condiciones para que funcione y sus posibles consecuencias.

Accede a la web a través del aula virtual o desde la siguiente dirección:
http://www.oxid.it/downloads/apr-intro.swf

Recursos externos

Nmap

Escáner de puertos multiplataforma.

Accede a la página desde el aula virtual o a través de la siguiente dirección web:

http://nmap.org/

Accede al manual a través del aula virtual o desde la siguiente dirección web:

http://www.csirtcv.gva.es/es/descargas/gu%C3%ADa-avanzada-de-nmap.html

Caín&Abel

Herramienta de ataques de red para plataformas *Windows*.

Accede a la web a través del aula virtual o desde la siguiente dirección web:

http://www.oxid.it

Accede al manual de uso a través de aula virtual o desde la siguiente dirección web:

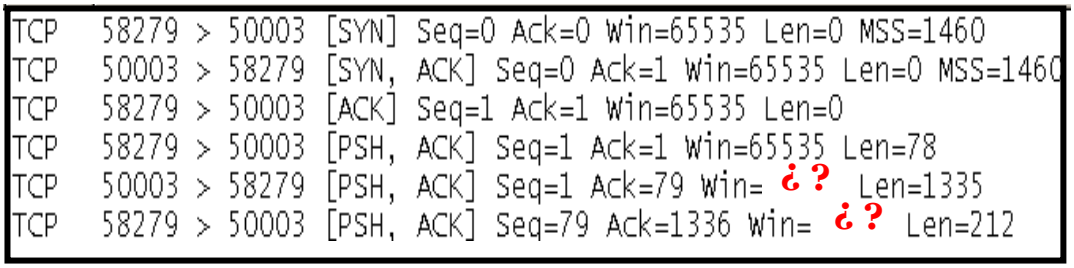
http://www.oxid.it/downloads/apr-intro.swf

Test

1. La primera fase que todo atacante suele llevar a cabo cuando se plantea llevar a cabo un intento de intrusión es:

A. Recopilación de información
B. Interceptación de tráfico
C. Búsqueda de vulnerabilidades
D. Ataques de denegación de servicio

2. De la información proporcionada por la captura siguiente, determina el valor del campo ventana que faltan en los dos últimos segmentos:



A. 65457 y 64200 respectivamente
B. 64200 y 65457 respectivamente
C. 65535 en ambos casos
D. 62400 y 65535 respectivamente

3. Los puertos 21, 53 y 80 corresponden a los servicios:

A. DNS, FTP y HTTP
B. FTP, DNS y HTTP
C. HTTP, DNS y FTP
D. FTP, HTTP y DNS

4. Un *sniffer* es una herramienta que sirve para:

A. Realizar un escaneo de puertos
B. Realizar un ataque de denegación de servicio
C. Capturar tráfico dirigido a otras máquinas
D. Captura tráfico dirigido a la propia máquina

5. *Nmap* es una herramienta de:

A. Descubrimiento y gestión de vulnerabilidades
B. Ataque del protocolo ARP
C. Trazado de rutas a un *host* remoto
D. Escaneo de puertos

6. Un ataque de denegación de servicio muy sencillo puede llevarse a cabo:

A. Enviando un paquete con el bit ACK activado
B. Enviando un paquete con el bit RST activado
C. Enviando un paquete con los bits SYN y ACK activados
D. Enviando un paquete con el bit SYN activado y no respondiendo al ACK del servidor

7. El protocolo ARP sirve para:

A. Traducir las direcciones de nivel de red en direcciones de nivel físico
B. Traducir las direcciones de nivel de aplicación en direcciones Ethernet
C. Traducir direcciones MAC en direcciones IP
D. Traducir direcciones TCP/IP en direcciones Ethernet

8. Para que un ataque de envenenamiento ARP pueda llevarse a cabo con éxito, es necesario que:

A. El atacante esté en el mismo dominio de difusión que el objetivo
B. El atacante esté en el mismo dominio de colisión que el objetivo
C. No haya ningún elemento de red (hub, switch, etc.) en el camino entre atacante y objetivo
D. El objetivo esté conectado a una VLAN diferente a la del atacante

9. Una de las herramientas más conocidas para llevar a cabo un ataque de envenanamiento ARP es:

A. Nmap
B. Hping
C. Cain&Abel
D. Nessus

10. Una de las consecuencias más peligrosas del envenenamiento ARP es cuando se ataca el protocolo SSL, debido a que:

A. Los navegadores Web no detectan este tipo de ataques y no informan al respecto

B. Los usuarios suelen ignorar los mensajes de advertencia mostrados por los navegadores Web

C. Crear un vínculo criptográfico entre una clave pública y una identidad

D. Certificar que una persona o máquina (en general, una entidad) es realmente quien dice ser, a través de una Autoridad de Certificación, o CA