Vulnerabilidades en redes

PID_00255332

Jordi Herrera Joancomartí
Guillermo Navarro Arribas

Universitat Oberta de Catalunya

Índice

Introducción

En el presente módulo se pretende dar una visión global de la complejidad y diversidad de las vulnerabilidades en red. Para ello, se hará un repaso de algunas vulnerabilidades presentes en distintos niveles de redes, desde redes locales hasta protocolos de encaminamiento de Internet.

Sin embargo, este módulo no pretende presentar un análisis exhaustivo de todas la posibles vulnerabilidades de red que existen. Un objetivo de esta envergadura implicaría un contenido mucho más extenso y de mayor detalle y complejidad. Se irán comentado algunas vulnerabilidades importantes de protocolos de red, ya sea por su importancia desde el punto de vista histórico, didáctico, o actual. El módulo se centra principalmente en redes TCP/IP, y en concreto en la versión IPv4, que es la más extendida y utilizada hoy en día.

Por último, el módulo también presenta los escáneres de vulnerabilidades, herramientas básicas para mejorar la seguridad de sistemas informáticos. Los escáneres de vulnerabilidades permiten detectar vulnerabilidades que pueden derivar en problemas de seguridad.

Objetivos

Los objetivos que el estudiante debe haber conseguido después de estudiar los contenidos de este módulo son los siguientes:

1. Entender la complejidad y diversidad de las vulnerabilidades de red.
2. Identificar adónde afectan las vulnerabilidades de red más relevantes.
3. Conocer algunas de las principales vulnerabilidades de TCP/IP y tecnologías asociadas.
4. Conocer los escáneres de vulnerabilidades como herramienta para su detección.

1. Conceptos básicos

En este módulo repasaremos algunas vulnerabilidades de las redes informáticas. Para ello, nos centraremos en dar una visión global del tipo de vulnerabilidades que nos podemos encontrar en la redes telemáticas. Antes de entrar en detalle, repasamos algunos conceptos básicos.

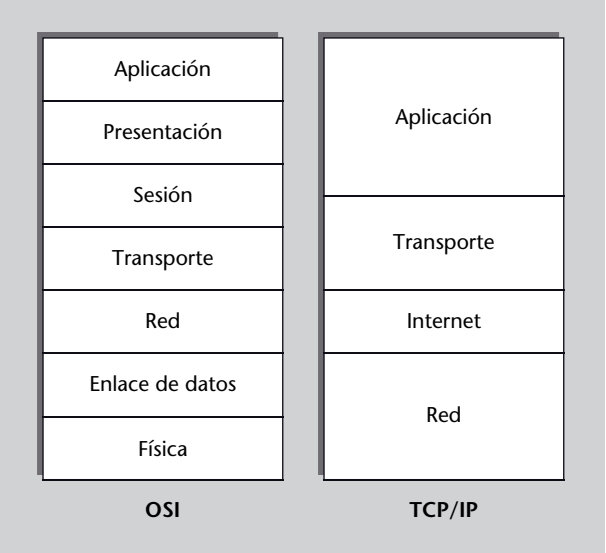
Las redes informáticas se organizan en una pila o *stack* de protocolos. El modelo OSI de interconexión de redes define 7 capas que van desde el medio físico de transmisión de señal, hasta las aplicaciones de alto nivel que hacen uso de la red. Esta separación de protocolos por capas permite definir y aislar claramente la funcionalidad de cada protocolo y aporta un diseño muy modular.

Sin embargo, actualmente y desde un punto de vista más práctico, se ha impuesto el modelo TCP/IP, que es el que define la *suite* de protocolos que dieron luz a Internet tal como se conoce en la actualidad. En la figura 1 vemos las principales capas de los modelos OSI y TCP/IP, y su correspondencia aproximada.

Vigencia del modelo OSI

El modelo OSI sigue siendo utilizado como referencia y ha sido implementado por algunos protocolos, pero la popularidad de Internet ha hecho que sea el modelo TCP/IP el más utilizado actualmente.

Figura 1. Relación de capas en el modelo OSI y TCP/IP



En TCP/IP no existe una división por capas tan clara como en el modelo OSI y muchas veces la frontera entre una capa y otra es algo difusa.

El presente módulo se centra en el modelo TCP/IP. A continuación detallamos la principal funcionalidad de cada capa:

- **Red**: también llamada *link layer* o *network access layer*, engloba las conexiones de red local.

- **Internet**: la capa de interconexión de red (en inglés *internet*) permite el envío de datos entre redes locales. Para ello, proporciona un sistema de direccionamiento global (direcciones IP) y el encaminamiento de paquetes de datos.

- **Transporte**: se encarga de la transferencia de datos extremo a extremo con independencia de la red local. Puede incluir funcionalidad para el control de errores, segmentación, control de flujo, control de congestión, y el direccionamiento a la capa de aplicación mediante el uso de *puertos*.

- **Aplicación**: incluye protocolos de alto nivel utilizados directamente por las aplicaciones, como por ejemplo HTTP (*hypertext trasfer protocol*), FTP (*file transfer protocol*) o SMTP (*simple mail trasport protocol*).

Para revisar las vulnerabilidades de red nos centraremos en tres grandes bloques: protocolos locales, interconexión de redes y protocolos extremo a extremo. Estos bloques se corresponden vagamente con las capas TCP/IP de red, internet y transporte, respectivamente. La correspondencia no es exacta, ya que nos centramos, por motivos de simplicidad, en la funcionalidad propia de cada bloque con cierta independencia de si los protocolos tratados corresponden estrictamente a una u otra capa.

Con el objetivo de simplificar la exposición se ha optado por dejar de lado algunos temas importantes. El primero es la capa de aplicación; la seguridad de las aplicaciones de red y sus vulnerabilidades serán tratadas en el módulo "Ataques a aplicaciones web", y otras aplicaciones concretas se cubrirán en asignaturas concretas de seguridad en redes. De la misma manera, aunque algunas de las vulnerabilidades que veremos son comunes a muchos tipos de red, no se verán problemas específicos de redes inalámbricas, como IEEE 802.11 (wifi), Bluetooth, ZigBee, etc. Dichas redes serán también tratadas en asignaturas específicas.

Asimismo, es importante remarcar que nos centramos principalmente en IPv4, ya que es la versión más extendida y estudiada, y nos permite presentar de una manera global la problemática de la seguridad en redes cableadas. Es importante recordar que se está produciendo la larga migración de IPv4 a IPv6. En esta última versión algunos de los problemas de seguridad que veremos a continuación no están presentes debido al propio diseño de los protocolos. El estudio de la seguridad en IPv6 se verá en otras asignaturas específicas de redes.

Finalmente, el módulo repasa los escáneres de vulnerabilidades como herramientas básicas para mejorar la seguridad de redes y sistemas informáticos en general. Veremos qué tipos de escáneres existen, así como su uso y aplicabilidad.

2. Protocolos locales

Existen muchos protocolos de red de área local o LAN (*local area network*). Sin duda, las redes locales cableadas más utilizadas son las basadas en Ethernet. La familia de tecnologías Ethernet permite hoy en día la creación de redes locales relativamente extensas y capaces de alcanzar gran velocidad de transmisión.

Ethernet utiliza direcciones físicas o MAC (*media access control*), de 48 bits únicas globalmente y asignadas por el fabricante. Utiliza paquetes denominados *frames* de 1.518 bytes con un espacio para 1.500 bytes de datos. Cuando una red Ethernet se conecta a otra red (o a Internet) mediante TCP/IP es necesario "traducir" direcciones físicas a direcciones IP. Esta traducción se lleva a cabo mediante un protocolo de bajo nivel denominado ARP (*address resolution protocol*).

Aunque Ethernet ha evolucionado mucho desde sus orígenes, sigue presentando problemas de seguridad y existen vulnerabilidades importantes en este tipo de redes. A continuación veremos algunas de las más representativas.

2.1. *Sniffers* de Ethernet

Uno de los principales problemas o vulnerabilidad que presentaba inicialmente Ethernet era la facilidad de esnifar tráfico local. Ethernet utilizaba una topología de bus donde todos los paquetes se enviaban al bus y solo el *host* con la dirección destino del paquete recogía dicho paquete.

En las tarjetas de red Ethernet existe un modo de funcionamiento promiscuo (*promiscuous mode*) que permite recoger todos los paquetes que pasan por el bus. Este modo de funcionamiento está pensado para monitorizar la red en la detección de problemas y permite el uso de la tarjeta como puente (*bridge*) para la virtualización de hardware. Pero se puede hacer un uso malicioso de este modo para esnifar todo el tráfico local de la red. Típicos *sniffers* de red como *tcpdump*, *ettercap* o *Wireshark* pueden esnifar paquetes Ethernet desde una tarjeta en modo promiscuo.

La mayoría de los sistemas operativos requieren privilegios administrativos (de superusuario o *root*) para poder operar una tarjeta en modo promiscuo.

La tecnología Ethernet ha evolucionado mucho. De la clásica topología de bus se pasó a topologías de estrella utilizando unos dispositivos de red denominados concentradores* que simulaban la funcionalidad de un bus, y por tanto presentan los mismos problemas que el bus. Actualmente, el concentrador suele ser reemplazado por un conmutador**. A diferencia de un concentrador, un conmutador Ethernet no simula el funcionamiento del bus, sino que tiene la capacidad de aprender la topología de la red. Es decir, basándose en las direcciones MAC de los paquetes que reciben/envían los *hosts*, sabe dónde está cada uno y solo envía los paquetes destinados a dicho *host* por su cable correspondiente. De esta manera, no solo se consigue minimizar el tráfico de toda la red, sino que además imposibilita que un *host* en modo promiscuo pueda recibir todos los paquetes que circulan por la red.

*En inglés *hub*.

**En inglés *switch*

Existen vulnerabilidades inherentes a los actuales conmutadores Ethernet que permiten convertir un conmutador en un concentrador y consecuentemente esnifar todo el tráfico de la red desde un *host*. Esta vulnerabilidad se puede explotar con un ataque de *MAC flooding*.

2.1.1. MAC flooding

Un conmutador Ethernet mantiene una tabla llamada CAM (*content addressable memory*) donde establece un vínculo entre direcciones MAC y puertos físicos del propio conmutador. Esta tabla le permite al conmutador enviar los paquetes únicamente a su destinatario por el puerto físico correspondiente. El conmutador establece la tabla CAM observando el tráfico generado y destinado a cada *host* conectado a este.

El problema es que la tabla CAM de un conmutador tiene una memoria limitada y, por tanto, un atacante puede saturar dicha tabla con el propósito de dejarla inutilizada. Para ello, bombardea el conmutador con paquetes Ethernet con direcciones MAC de orígenes diferentes, lo que provoca que el conmutador las añada a la tabla CAM hasta que esta se agota. En ese momento, dado que el conmutador no puede añadir más entradas, pasa a un modo de funcionamiento conocido como *failopen*, en el que empieza a actuar como un *hub*. En este modo el conmutador envía los paquetes en *broadcast* a todos los *hosts* de la red.

Ataque MAC flooding

MAC flooding es un ataque muy conocido y estudiado. Existen numerosas herramientas que permiten estudiar el comportamiento de la red al realizar el ataque, como la utilidad *macof* del paquete *dsniff* (http://monkey.org/~dugsong/dsniff/).

Actualmente se puede intentar mitigar esta vulnerabilidad utilizando sistemas de monitorización de red. Muchos fabricantes de conmutadores permiten limitar el número máximo de direcciones MAC para cada puerto físico (técnica conocida como *port security*). También existen mecanismos para requerir la autenticación con servidores de autenticación y autorización.

2.2. Modificación de direcciones MAC

La dirección física de Ethernet es asignada por el fabricante y tradicionalmente estaba inequívocamente asociada a la tarjeta de red. Es decir, dicha dirección es única globalmente y no se puede modificar. Esto hizo que se desarrollasen mecanismos de control de acceso basados en direcciones MAC.

Por ejemplo, una red puede permitir el acceso solo a unas direcciones MAC concretas. Otro ejemplo muy común hoy en día son las redes (generalmente inalámbricas) que permiten una conexión gratuita a la red de, por ejemplo, 15 minutos al día; en ellas, el control sobre el tiempo de conexión de cada usuario suele hacerse mediante la dirección MAC.

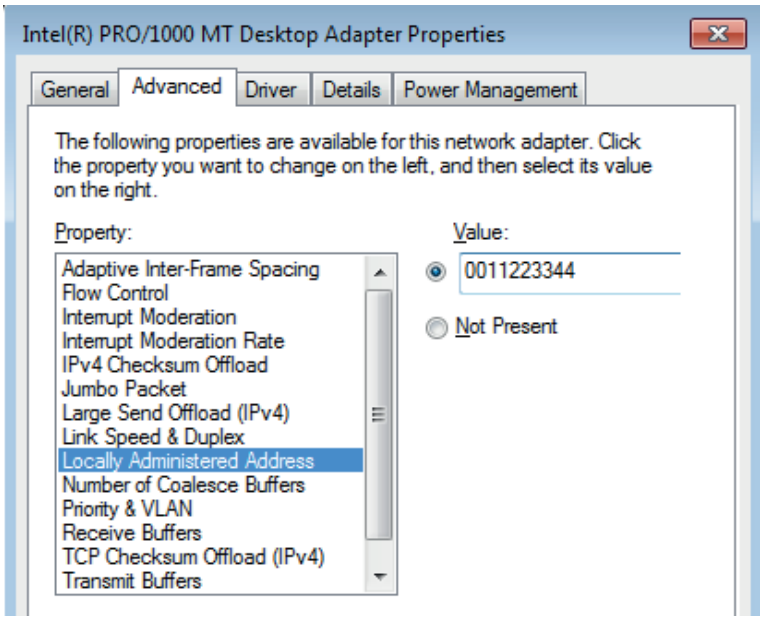
En la actualidad una dirección física Ethernet puede ser fácilmente modificada. Esta funcionalidad es actualmente tan común que muchos sistemas operativos incluyen la opción de modificar la dirección MAC con las herramientas propias de gestión de red, sin necesidad de programas externos.

Por ejemplo, utilizando el comando *ifconfig* o las utilidades *iproute2* de Linux podemos modificar la dirección física de la interfaz de red *eth0*:

```
# ifconfig eth0 hw ether 00:11:22:33:44:55
# ip link set dev eth0 address 00:11:22:33:44:55
```

En Windows se puede modificar desde las propiedades del adaptador red (podéis ver la figura 2).

Figura 2. Modificación de la dirección MAC en Windows 7



La actual facilidad para cambiar direcciones físicas supone una vulnerabilidad importante en sistemas de control de acceso o autenticación de red basados en dichas direcciones.

2.3. Vulnerabilidades en el protocolo ARP

Como hemos comentado, el protocolo *address resolution protocol* (ARP) permite traducir direcciones MAC en direcciones IP. En redes locales que soportan broadcast como Ethernet, el funcionamiento de ARP es el siguiente. Supongamos que el *host A* conoce la dirección IP de *B* (IP_B) pero desconoce su dirección física F_B:

Observación

En realidad, ARP se diseñó para traducir direcciones físicas a direcciones de protocolo superior de interconexión (no necesariamente IP); aquí hacemos referencia a IP porque es el más usado.

1) *A* envía en broadcast una petición ARP preguntando quién tiene la dirección IP_B.

2) *B* contesta con una respuesta ARP a *A* diciendo que él tiene la dirección IP_B (es decir, que IP_B le corresponde a la dirección física F_B).

Para evitar estar haciendo peticiones ARP continuamente, cada *host* mantiene una tabla con las correspondencias entre dirección física y dirección IP llamada *cache ARP*. Las entradas tienen una caducidad de aproximadamente 20 minutos. Un punto importante es que cualquier petición ARP es utilizada por el resto de los *hosts* para actualizar la entrada correspondiente al emisor.

Este protocolo actualmente presenta vulnerabilidades difíciles de solucionar. La mayoría de los problemas de seguridad de ARP se basan en el envío de mensajes ARP falsos para "envenenar" las cachés ARP, es decir, para introducir información falsa. En inglés, esta técnica recibe el nombre de *ARP poisoning*. ARP poisoning permite varios ataques diferentes, como:

ARP poisoning también se conoce con los nombres: *ARP spoofing, ARP flooding* o *ARP poison routing.*

- **Denegación de servicio**: se puede conseguir que un *host* no reciba ningún paquete al difundir una dirección física inexistente asociada a su dirección IP real. Si la víctima de la denegación de servicio es el encaminador o puerta de enlace de la red local, se consigue aislar la red local del exterior.

- **Man in the middle (MITM)**: un atacante se puede hacer pasar por otro *host* (víctima) y recibir así todos los paquetes destinados a dicha víctima.

ARP poisoning puede tener usos legítimos, por ejemplo para direccionar accesos de red a un portal de autenticación (típicamente usado en hoteles, redes universitarias o redes de acceso público), o en sistemas de redundancia para que un servidor pueda tomar el lugar de otro en caso de que este sufra algún percance.

Actualmente solo existen soluciones limitadas para mitigar estas vulnerabilidades. Algunas soluciones actuales son incluir entradas fijas en la caché ARP, generalmente las correspondientes a *hosts* críticos como encaminadores, el uso de *port security* en los conmutadores, o la monitorización de la red en busca de comportamiento inusual.

Ved también

La técnica del port security se estudia en el subapartado 2.1.1.

3. Interconexión de redes

Dentro de la interconexión de redes nos centramos en el protocolo IP y los servicios asociados. Dada la complejidad de Internet y todas las tecnologías asociadas a la interconexión de redes, el número de vulnerabilidades potenciales es muy grande. En este apartado enumeramos algunas de ellas a modo ilustrativo y con fines didácticos.

3.1. Vulnerabilidades en IP

Vemos a continuación algunas de las vulnerabilidades del protocolo IP especialmente relevantes partir de los siguientes ataques:

- **IP spoofing**: consiste en generar paquetes IP con la dirección de origen falsa. Esta vulnerabilidad se suele explotar con el objetivo de hacer ataques de denegación de servicio o suplantar a un *host* concreto.

Ved también

Sobre los ataques de denegación de servicio podéis ver el subapartado 4.1.1.

- **Packet-of-death**: IP ha sufrido algunas vulnerabilidades en la implementación. El envío de paquetes IP deliberadamente erróneos puede causar problemas importantes en algunas implementaciones. Un ejemplo es el uso de la misma dirección IP como origen y destino (*land attack*).

- **Vulnerabilidades en la fragmentación**: IP puede realizar fragmentación de paquetes para adaptarse a los tamaños de paquetes de redes locales. El envío de fragmentos erróneos donde se solapan los campos de datos ha dado problemas en algunas implementaciones. Un caso conocido es el *teardrop attack*, que explotaba una vulnerabilidad en la implementación de SMBv2 de Windows Vista.

Teardrop attack

Teardrop attack es un ataque que explota la vulnerabilidad CVE-1999-0015. Se puede consultar su publicación en: http://www.microsoft.com/technet/security/advisory/975497.mspx.
Más información sobre vulnerabilidades en la fragmentación IP se puede encontrar en el RFC 1858 (sección de referencias).

- **IP source routing**: IP incluye un par de opciones que permiten especificar la ruta (parcial) de retorno que debe seguir el paquete de respuesta. Esto permite que un atacante utilizando IP spoofing con una dirección de origen de un *host* de confianza de la víctima pueda recibir el paquete de respuesta. Actualmente estas opciones no se suelen utilizar, y muchos dispositivos de red bloquean el paso de paquetes con estas opciones.

3.2. Vulnerabilidades en ICMP

El protocolo ICMP (*Internet control message protocol*) es un protocolo de control y notificación de errores del protocolo IP que ha presentado algunas vulnera-

bilidades importantes, generalmente asociadas a ataques de denegación de servicio. A continuación, detallamos algunos ataques que explotan vulnerabilidades de ICMP:

- **Ping flooding**: ataque clásico de denegación de servicio utilizando mensajes *echo request* (*ping*) de ICMP.

- **Ping of death**: ataque de interés histórico que afectó a casi todas las implementaciones de ICMP hasta finales de los noventa. Consiste en enviar un paquete ICMP de mayor tamaño que el máximo permitido por IP fragmentado. Esto provocaba un *buffer overflow* en el *host* de destino.

- **Smurf attack**: ataque de denegación de servicio en el que el atacante envía mensajes de *ping* en *broadcast* con la dirección de origen de la víctima. Esto provoca que todos los *hosts* que reciben el paquete envíen la respuesta del *ping* a la víctima. En este caso la vulnerabilidad se encuentra en el uso de mensajes ICMP en *broadcast*. Actualmente, los *hosts* no contestan a *pings* enviados en *broadcast*. Asimismo, se modificó el estándar para requerir que por defecto los encaminadores bloqueen mensajes enviados en *broadcast*.

Tratamiento de mensajes de *broadcast*

En general, el uso de mensajes de broadcast suele implicar vulnerabilidades importantes en la interconexión de redes. Con el tiempo se ha ido limitando mucho su uso. Por ejemplo, podéis ver:
D. Senie (1999). *Changing the Default for Directed Broadcasts in Routers*. RFC 2644. IETF, The Internet Society.

3.3. Vulnerabilidades en DNS

DNS (*domain name system*) permite la resolución de nombres de dominio mediante servidores organizados jerárquicamente a partir de 13 servidores raíz (9 de ellos distribuidos geográficamente utilizando *anycast*). DNS presenta muchas ventajas. Es un sistema distribuido, eficiente en la resolución de nombres y tolerante a fallos. Existen sin embargo algunas vulnerabilidades en DNS que pueden dar lugar a ataques importantes:

- **DNS spoofing**: consiste en dar información errónea de manera deliberada sobre la correspondencia de dirección IP a nombre de dominio. El objetivo puede ser por ejemplo asociar una IP "falsa" a un nombre de dominio conocido (con lo que se puede redirigir el tráfico a dicho dominio). Existen distintos modos de realizar estos ataques. El más sencillo es poner un servidor DNS que emita respuesta falsas o que pueda suplantar un servidor conocido (por ejemplo, mediante IP spofing), también se puede interceptar la petición de DNS y responder antes de lo que lo haría el servidor legítimo. Es importante tener en cuenta que para que una respuesta sea aceptada como legítima debe cumplir los siguientes puntos:

Ved también

El IP spofing se estudia en el subapartado 3.1.

– Volver a la misma dirección IP que emitió la petición.
– Volver por el mismo puerto desde donde se envió la petición.
– Que la respuesta corresponda a la petición.
– Que el número de transacción coincida con la petición. Este número es teóricamente aleatorio y permite vincular respuesta a petición.

En muchos casos la facilidad para predecir el número de transacción ha sido una vulnerabilidad importante, similar a la predicción de números de secuencia en TCP. Un atacante puede falsear una respuesta DNS sin necesidad de ver la petición para saber el número de transacción. El hecho de que DNS funcione sobre UDP también facilita este tipo de ataques. Es importante remarcar también que, dada la organización jerárquica de DNS, estos ataques se pueden hacer directamente sobre el cliente o a algún servidor intermedio.

Ved también

La predicción de números de secuencia se estudia en el subapartado 4.1.2 de este módulo.
Las vulnerabilidades en UDP se estudian en el subapartado 4.2.

- **DNS cache poisoning**: para mejorar la eficiencia de DNS cada servidor mantiene una caché con las últimas resoluciones hechas (respuestas de DNS) para poder contestar a futuras peticiones de manera rápida. Al igual que en el caso de ARP, se puede forzar la entrada de relaciones de nombre de dominio a dirección IP falsa en dicha caché. Muchos ataques de DNS spofing buscan precisamente "envenenar" la caché de servidores intermedios (por ejemplo, el DNS de un ISP) con el objetivo de que todos sus clientes queden afectados.

- **DNS amplification attacks**: los ataques de amplificación de DNS son unos ataques de denegación de servicio que explotan el hecho de que las peticiones de DNS se resuelven recursivamente y que una petición de tamaño pequeño (60 bytes) puede llegar a generar respuestas más grandes ($\leq$ 512 bytes). De manera similar a los ataques *smurf*, se envían muchas peticiones con la dirección IP de origen de la víctima que recibirá todas las respuestas. La denegación de servicio se agrava por el gran tamaño que pueden alcanzar dichas respuestas (amplificación). En octubre del 2002 se realizó un gran ataque de amplificación en el que las víctimas eran los servidores raíz de DNS que consiguió comprometer a alguno de ellos. El hecho de que no todos los servidores fuesen comprometidos es visto como una ventaja de la redundancia de DNS. Un ataque similar fue repetido en septiembre del 2007 y consiguió afectar a dos servidores raíz.

Enlace de interés

Los servidores raíz de DNS son conocidos en inglés como *root name servers*. Información sobre estos servidores raíz se puede consultar en: http://www.root-servers.org.

Ved también

Los ataques *smurf* se estudian en el subapartado 3.2 de este módulo.

DNS fue diseñado sin tener en cuenta la seguridad, y sus vulnerabilidades han sido importantes e incluso detalladas en el RFC 3833. Actualmente existe DNSSEC (*domain name system security extensions*), un conjunto de especificaciones de la IETF que busca solucionar los problemas de seguridad de DNS. DNSSEC proporciona principalmente autenticación e integridad.

Lectura recomendada

D. Atkins; R. Austein (2004). *Threat Analysis of the Domain Name System (DNS)*. RFC 3833.
Disponible en: http://www.ietf.org/rfc/rfc3833.txt

3.4. Vulnerabilidades en OSPF y BGP

IP utiliza varios protocolos de encaminamiento. Por una parte, están los protocolos de encaminamiento internos a un sistema autónomo, conocidos como IGP (*interior gateway protocol*), y los que se utilizan para el encaminamiento entre sistemas autónomos, EGP (*exterior gateway protocol*).

Posiblemente el IGP más utilizado en la actualidad sea el OSPF (*open shortest path first*), un protocolo de encaminamiento adaptativo basado en *link-state* que se utiliza como protocolo de encaminamiento interior en sistemas autónomos. Existen varias vulnerabilidades en OSPF que permiten a un atacante introducir información de encaminamiento falsa en el sistema autónomo. Esto facilita distintos ataques, como la denegación de servicio o la "desconexión" de una red local (en inglés se suele denominar *blackhole*), desviación de tráfico, etc.

Actualmente OSPF permite incorporar distintos mecanismos de autenticación que pueden mitigar algunas de estas vulnerabilidades pero, en general, es difícil defenderse de ataques internos en OSPF.

Por otra parte, el protocolo EGP que se utiliza hoy en día en Internet es BGP (*border gateway protocol*). BGP ha sufrido varias vulnerabilidades que se han intentado resolver a lo largo del tiempo. Las últimas versiones del protocolo incorporan mecanismos para autenticar los encaminadores que anuncian rutas BGP.

El principal problema que presenta actualmente BGP es la credibilidad depositada en los encaminadores de confianza. Por lo general, un encaminador BGP está configurado para recibir y emitir anuncios de rutas únicamente de encaminadores de confianza, lo que facilita la posibilidad de ataques internos.

Pakistan Telecom

Un caso famoso fue protagonizado por Pakistan Telecom (el principal ISP de Pakistán), que en febrero del 2008 empezó a anunciar que los rangos de direcciones IP correspondientes a YouTube se encontraban dentro del sistema autónomo de Pakistan Telecom. El origen de este anuncio se encuentra en una orden gubernamental que obligaba a todos los ISP de Pakistán a bloquear el acceso a unos vídeos de YouTube. Para ello, Pakistan Telecom pone el rango de direcciones IP de YouTube como interno al propio sistema autónomo. El problema es que no evitaron que estas rutas falsas se anunciasen por BGP a ISP de confianza. Estos encaminadores a su vez redistribuyeron la rutas a sus encaminadores de BGP de confianza, y así hasta llegar a abarcar gran parte de Internet. Esto provocó que todo el tráfico destinado a YouTube se redireccionase a Pakistan Telecom y las direcciones IP de YouTube fuesen inaccesibles. Una vez detectado el problema se pudo aislar al encaminador BGP que emitía estas rutas falsas, de manera que los encaminadores BGP lo podían excluir de sus listas de encaminadores de confianza.

Enlace de interés

Se puede consultar el informe elaborado por RIPE sobre el incidente de Pakistan Telecom en: http://www.ripe.net/internet-coordination/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study

La importancia de Internet hoy en día y el importante papel que desempeñan en su funcionamiento los protocolos de encaminamiento (especialmente BGP), provoca que cualquier vulnerabilidad, y consecuentemente ataque a estos protocolos, tenga consecuencias muy importantes. Por ejemplo, el caso de

Pakistan Telecom duró tan solo dos horas, pero sus consecuencias pudieron ser importantes.

Actualmente existen vulnerabilidades en BGP difíciles de solucionar, por lo que se opta por monitorizar continuamente el funcionamiento de BGP para poder detectar problemas y actuar rápido.

4. Protocolos extremo a extremo

Dentro de los protocolos extremo a extremo utilizados por TCP/IP cabe destacar por su extenso uso: TCP (*trasmission control protocol*), y UDP (*user datagram protocol*).

Estos protocolos introducen el uso de puertos que permiten direccionar datos de la capa inferior IP a aplicaciones concretas. Esto es considerado por algunos autores como una vulnerabilidad, ya que posibilita el uso de escáneres de puertos para obtener información sobre qué servicios esta ofreciendo un *host*, incluso información adicional, como el sistema operativo, la versión, etc. Sin embargo, otros autores no lo consideran una vulnerabilidad, ya que no consideran que la información obtenida por estos sistemas sea crítica desde el punto de vista de la seguridad.

4.1. Vulnerabilidades de TCP

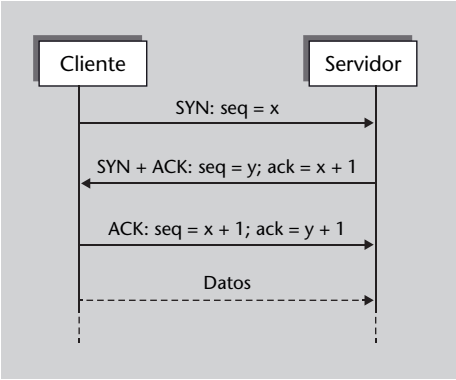
TCP proporciona un servicio genérico de transmisión fiable de datos extremo a extremo. Se encarga de controlar errores en la transmisión, el orden de los paquetes, la detección de duplicados, el control de velocidad de transmisión, etc.

TCP es un protocolo orientado a conexión. Se establece una conexión entre los dos extremos que se mantiene durante la transmisión de datos. Esta conexión se crea mediante el denominado *3-way handshake*.

Como vemos en la figura 3, la conexión se establece con tres mensajes: un mensaje tipo SYN (de sincronización), al que el servidor contesta con un SYN y ACK (sincronización y reconocimiento) y finalmente el cliente envía un ACK (reconocimiento). Una vez establecida la conexión, el cliente y el servidor se pueden enviar datos que serán reconocidos cada cierto tiempo (mediante mensajes ACK) por el receptor.

Como vemos, una de las tareas del establecimiento de conexión es fijar los números de secuencia de cliente y servidor (en la figura 3, *seq* y *ack*). Estos números se utilizan para identificar los bytes de datos enviados y permiten realizar una gestión del flujo de datos: control del orden de envío de segmentos, pérdida, duplicados, etc. El número de *ack* confirma la correcta recepción de todos los bytes con número de secuencia inferior.

Figura 3. Establecimiento de conexión en TCP



A continuación veremos algunos ataques y vulnerabilidades de TCP.

4.1.1. SYN flooding

En el establecimiento de sesión, cuando el cliente envía el mensaje SYN, el servidor contesta (SYN+ACK) y se queda esperando el ACK del cliente. ¿Qué sucede si el cliente no envía ese ACK?

Como es de esperar, el servidor espera un tiempo prudencial y si no recibe el ACK, da la conexión por perdida. Esta funcionalidad presenta una vulnerabilidad importante de TCP que puede ser explotada para realizar ataques de denegación de servicio. La idea es bombardear un servidor con peticiones de conexión y no hacer el último ACK. De esta manera, el servidor se queda con conexiones medio establecidas que consumen, durante un tiempo determinado, recursos del servidor (memoria principalmente). Si hay suficientes intentos simultáneos, se puede llegar a saturar el servidor y agotar sus recursos.

Publicación de la vulnerabilidad

Se puede consultar la publicación en 1995 y 1996 de la vulnerabilidad relativa a SYN flooding e IP spoofing en: http://www.cert.org/advisories/CA-1996-21.html

Este ataque se conoce como SYN flooding, y fue muy importante cuando se descubrió. Existe una variante basada en IP spoofing, en la que el primer mensaje SYN del cliente lleva una dirección IP de origen falsa, por lo que le resulta imposible al servidor enviar el SYN+ACK.

Hoy en día esta vulnerabilidad no suele suponer un riesgo importante, ya que existen varios mecanismos que previenen frente a ataques de SYN flooding.

Prevención de SYN flooding

Existe un RFC sobre mecanismos de prevención de SYN flooding:
W. Eddy (2007). *TCP SYN Flooding Attacks and Common Mitigations*. RFC 4987. IETF Internet Society. http://tools.ietf.org/rfc/rfc4987.txt.

4.1.2. Predicción de números de secuencia

La facilidad a la hora de predecir los números de secuencia de una conexión TCP resultó ser una vulnerabilidad de seguridad importante. Para verlo, expli-

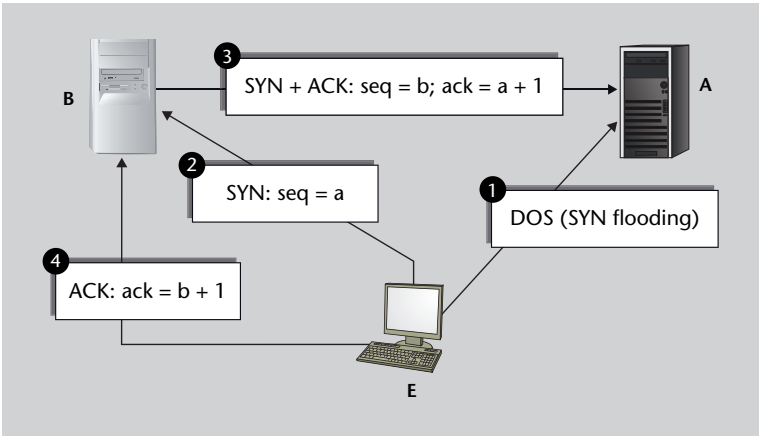
caremos brevemente uno de los ataques informáticos más conocidos y documentados en la historia de la seguridad informática.

El día de Navidad de 1994, un *hacker* llamado Kevin Mitnick realizó un ataque al ordenador de Tsutomu Shimomura situado en la Universidad de California, en San Diego. El ataque pretendía obtener el código fuente de un modelo de teléfono móvil (el que tenía Mitnick) que estaba almacenado en el ordenador de Shimomura. Mitnick pretendía modificar el software del teléfono y así intentar evitar los sistemas de seguimiento y localización de este.

Sin entrar en muchos detalles, el ataque consistió en los siguientes pasos que se detallan a continuación (podéis ver también la figura 4):

1) El ataque se inicia desde un servidor externo *E*, al que el atacante ha podido acceder con anterioridad.

Figura 4. Ataque de Mitnick



2) Desde la máquina externa *E* se recopila información del objetivo y se descubre que entre dos servidores *A* y *B* existe una relación de confianza. Esta permitía realizar conexiones de uno a otro a partir de su dirección IP. Es decir, el servidor *A* acepta peticiones de conexión del servidor *B*.

3) Desde *E* se realiza un ataque de *SYN-flooding* a *A* para evitar que dicho servidor pueda responder a cualquier mensaje. De esta manera, se quiere *silenciar* el servidor *A*, con el objetivo de que el atacante se pueda hacer pasar por dicho servidor, e iniciar así una conexión a *B*. Para suplantar al servidor *A*, el atacante realiza un ataque de *IP spoofing* que le permita suplantar la dirección IP de *A*. En este punto el atacante puede enviar mensajes a *B* haciéndose pasar por *A* pero no podrá ver la respuesta que genera, ya que no se encuentra en la misma red local.

4) Para poder establecer una conexión con *B* el atacante necesita predecir cómo va a contestar *A* al intento de conexión de *B* (dado que no puede ver dicha contestación). Esto es, predecir el número de secuencia TCP de los mensajes que genera *B*. La conexión TCP consta de 3 tres pasos, como se detalla en la figura 3, en la que los números de secuencia *seq* y *ack* deben coincidir.

Lectura complementaria

Para saber más sobre la historia de Mitnick y Shimomura, se pueden consultar los siguientes libros (cada uno presenta un punto de vista diferente):
T. Shimomura; J. Markoff (1996). *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw-By the Man Who Did It.* Hyperion Books.
J. Littman (1997). *The Fugitive Game: Online with Kevin Mitnick*. Little, Brown and Company publishers.
J. Goodell (1996). *The Cyberthief and the Samurai: The True Story of Kevin Mitnick-And the Man Who Hunted Him Down*. Dell publishers.

Ved también

El ataque *SYN-flooding* se estudia en el subpartado 4.1.1.
Para saber más sobre el ataque de *IP spoofing* podéis ver el subapartado 3.1 de este módulo.

5) Una vez se puede predecir el número de secuencia, el atacante puede realizar una conexión al servidor *B* para forzar a que *B* pase a aceptar conexiones de cualquier dirección IP.

6) El atacante puede ahora acceder a *B* desde cualquier sitio.

Una vez Mitnick obtuvo acceso al servidor de Shimomura pudo copiar todo el código fuente que buscaba. Este ataque dio lugar a uno de los sucesos más sonados de la seguridad informática. Finalmente, el FBI, con la ayuda de Shimomura, capturó a Mitnick, quien acabó pasando 5 años en prisión.

La posibilidad de predicción de números de secuencia resultó ser una vulnerabilidad importante de TCP; hoy en día las implementaciones de TCP ponen mucho interés en generar los números de secuencia de la manera más aleatoria posible para evitar este tipo de problemas.

4.2. Vulnerabilidades en UDP

User datagram protocol (UDP) es un protocolo de transmisión extremo a extremo que ofrece la funcionalidad mínima. No proporciona ninguno de los mecanismos de control de flujo de TCP.

La mayoría de las vulnerabilidades de UDP son propias de errores de implementaciones concretas y no del protocolo. De esta manera, nos encontramos con un tipo de ataque conocido como **UDP Bomb**, que explota vulnerabilidades presentes en implementaciones que fallan al recibir un datagrama UDP erróneo o mal construido. Un error típico es especificar en la cabecera del datagrama un tamaño que no se corresponde con el tamaño real del datagrama. Esto puede producir un *buffer overflow* en la implementación del protocolo.

Existen otros ataques, como el **fraggle attack**, que explotan el uso (y encaminamiento) de direcciones de *broadcast*. Este es un ataque idéntico al ataque *smurf* de ICMP pero con paquetes UDP, en este caso dirigidos a los servicios UDP *echo* y *chargen* (puertos UDP 7 y 19 respectivamente).

Ved también

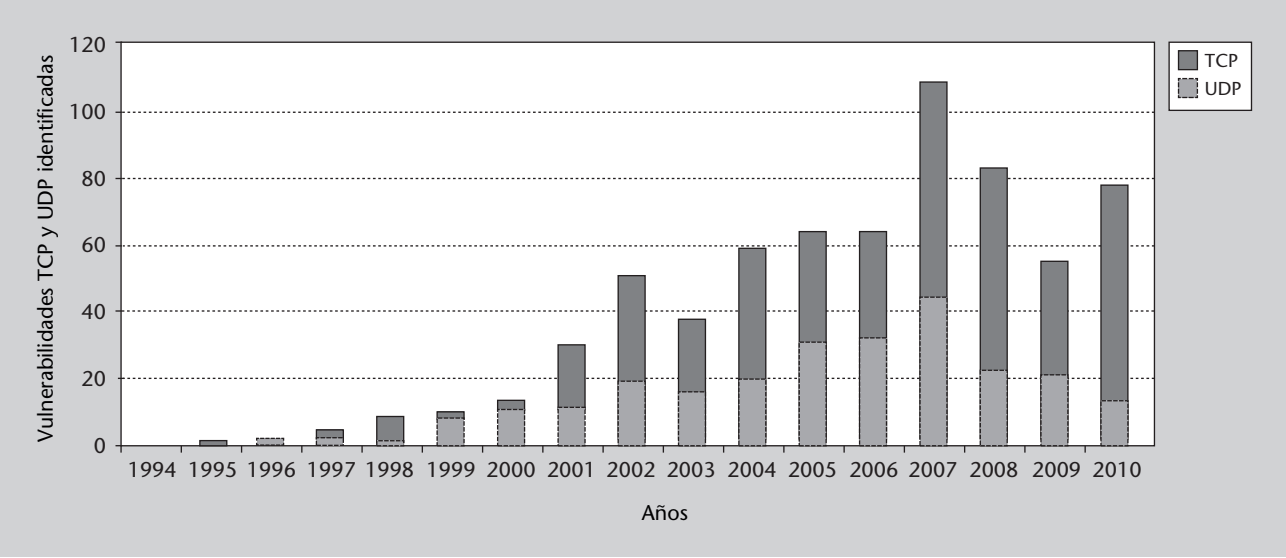
El ataque *smurf* se estudia en el subapartado 3.2 de este módulo.

El hecho de utilizar UDP en ocasiones facilita la explotación de otras vulnerabilidades debido a la falta de control en comparación con TCP. Por ejemplo, realizar un ataque de MITM usando IP spoofing es mucho más sencillo sobre UDP que sobre TCP, ya que no es necesario hacer el secuestro de sesión descrito en el subapartado 4.1.2.

Aquí vemos un caso claro de la relación entre la complejidad de un sistema y la presencia de vulnerabilidades. Un sistema complejo suele presentar más vulnerabilidades que uno más sencillo. De esta manera, el número de vulnerabilidades que afectan a TCP es mucho mayor que las asociadas a UDP.

Como ejemplo, en la figura 5 se muestra el número de vulnerabilidades reportadas que hacen referencia a UDP y a TCP. Es importante remarcar que esto no significa que UDP sea más seguro que TCP, simplemente que presenta menos vulnerabilidades. Es más, generalmente se considera UDP un protocolo de transporte más inseguro que TCP debido a que no realiza ningún control sobre el flujo de datos.

Figura 5. Vulnerabilidades relativas a TCP y UDP



Fuente: National Vulnerability Database (NVD)

5. Escáneres de vulnerabilidades

Entre las herramientas y los mecanismos que nos permiten mejorar la seguridad de los sistemas informáticos se encuentran las que están enfocadas a la detección de anomalías que pueden derivar en problemas para la seguridad del sistema. En este sentido, podemos realizar una distinción entre las herramientas que permiten detectar una vulnerabilidad y aquellas que permiten detectar un ataque. Si bien la distinción entre vulnerabilidad y ataque a menudo puede quedar diluida, en cuanto a su detección se considera que las herramientas que permiten la detección de vulnerabilidades quedan englobadas dentro de lo que se conoce como escáneres de vulnerabilidades, mientras que aquellas herramientas que se utilizan para la detección de los ataques se incluyen dentro de la familia de sistemas de detección de intrusos. Sin embargo, también puede suceder que un escáner de vulnerabilidades detecte un ataque, ya que, por ejemplo, un ataque puede poner al descubierto o generar una nueva vulnerabilidad.

En este apartado nos centraremos solo en los escáneres de vulnerabilidades que permiten detectar las vulnerabilidades de un sistema.

5.1. Características generales de los escáneres

Los **escáneres de vulnerabilidades** son un conjunto de herramientas que nos permiten detectar las vulnerabilidades de un sistema, ya sea por medio de simulaciones de ataques, ya sea porque se detecta una configuración que implica una deficiencia de seguridad.

El funcionamiento general de un escáner de vulnerabilidades se podría dividir en tres etapas:

1) Durante la primera etapa se realiza una extracción de muestras del conjunto de atributos del sistema para poder almacenarlas posteriormente en un contenedor de datos seguro.

2) En la segunda etapa, estos resultados son organizados y comparados con unas bases de datos de reglas y firmas que permiten identificar configuraciones inseguras.

3) Finalmente, se generará un informe con las diferencias entre ambos conjuntos de datos.

La principal ventaja de los escáneres de vulnerabilidades es que permiten la detección y solución de la vulnerabilidad antes de que esta pueda ser explotada para realizar un ataque. No obstante, hay que tener en cuenta que la mayoría de las vulnerabilidades detectadas por un escáner no pueden ser reparadas por el propio escáner. Este se limita a proporcionar una serie de información e informes que, en la mayoría de los casos, requieren una intervención manual del administrador.

Como vemos, el mecanismo de trabajo de los escáneres de vulnerabilidades se asemeja al de los antivirus, por su dependencia a una base de datos en la que se incluyen las reglas que tendrá en cuenta el escáner. Por este motivo, los escáneres de vulnerabilidades comparten ciertas características con los antivirus. Por ejemplo, una de las limitaciones básicas de los escáneres de vulnerabilidades es que únicamente permiten identificar las vulnerabilidades que están ya tipificadas en su base de datos. Esto implica que, en general, solamente se puedan detectar vulnerabilidades en software estándar, de modo que, por ejemplo, aplicaciones web personalizadas no pueden ser escaneadas con escáneres de vulnerabilidades de propósito general y precisan analizadores específicos de aplicaciones web. Por otro lado, dada la dependencia de la base de datos de vulnerabilidades con la que trabaja el escáner, la frecuencia de actualización de esta es un punto muy importante que se debe tener en cuenta para la selección de un escáner de vulnerabilidades, dado que será preciso una actualización constante de la base de datos de referencia para que el escáner pueda detectar las últimas vulnerabilidades publicadas.

Bases de datos de vulnerabilidades

Es importante distinguir entre las bases de datos de los escáneres de vulnerabilidades y las bases de datos que mantienen los CERT. En estas últimas, la información que se incluye es una información más descriptiva, mientras que las bases de datos de los escáneres de vulnerabilidades incorporan, para cada vulnerabilidad, un conjunto de tests que deben permitir su detección.

Dada la importancia de la base de datos de vulnerabilidades en los escáneres, existe un porcedimiento para determinar la compatibilidad del producto respecto al estandar CVE (*common vulnerabilities and exposures*) que permite etiquetar las vulnerabilidades. Este procedimiento, establecido por *The Mitre Corporation*, especifíca los requisitos que un escáner de vulnerabilidades debe poseer para que sea compatible con CVE. En concreto, para que un producto sea compatible con CVE debe cumplir:

- **Búsqueda por CVE**: el producto certificado debe permitir la búsqueda de vulnerabilidades en su base de datos utilizando el identificador CVE.

- **Salida CVE**: la información de la vulnerabilidad que ofrece el producto debe incluir el identificador CVE.

- **Identificación**: el producto debe proporcionar información suficiente de cómo identifica la vulnerabilidad de su base de datos con la versión específica de CVE y, a su vez, debe intentar que esta identificación sea tan precisa como sea posible.

- **Documentación**: La documentación estándar del producto debe incluir una descripción de CVE, la compatibilidad CVE, y los detalles de cómo

sus clientes pueden utilizar la funcionalidad relacionada con CVE de su producto o servicio.

El cumplimiento de estas condiciones por parte de un escáner es relevante puesto que permite a los usuarios complementar las informaciones que el escáner proporciona en sus informes utilizando fuentes externas, como los CERT, dado que la identifiación de la vulnerabilidad será únivoca. Por otro lado, la identificación unívoca de las vulnerabilidades mediante su código CVE también permite comprobar el grado de actualización de la base de datos del escáner.

Software para el escaneo de vulnerabilidades

Existen diferentes empresas que comercializan software para el escaneo de vulnerabilidades. Las principales diferencias entre los distintos productos se encuentran en sus características, como, por ejemplo, la ratio de falsos positivos que genera la detección, la variedad de los posibles sistemas operativos que permiten escanear, los tipos de dispositivos que pueden escanear (servidores, encaminadores, impresoras de red, etc.), el número diferente de aplicaciones que pueden escanear (bases de datos, servidores de aplicaciones PHP, Java, .NET, etc.), la frecuencia de actualización de la bases de datos o la calidad de la información que reportan para que los administradores puedan arreglar o eliminar la vulnerabilidad encontrada.

5.2. Clasificación de los escáneres

Los escáneres de vulnerabilidades admiten diferentes tipos de clasificación. La clasificación más común que se solía asociar con los escáneres de vulnerabilidad es la distinción que se refiere a la localización del propio escáner. De este modo, los escáneres se pueden clasificar en aquellos basados en máquina (*host-based scanners*) y los escáneres basados en red (*network-based scanners*). Los primeros son escáneres situados en los propios dispositivos que se pretende escanear, mientras que los segundos se sitúan en servidores de la red y permiten realizar análisis de otras máquinas.

La sofisticación de los escáneres basados en red, así como la aparición de nuevas técnicas para el escaneo de vulnerabilidades, permiten una nueva clasificación más precisa. De este modo, podemos clasificar los escáneres en función de sus habilidades de escaneo:

- Escaneo interno y activo de un dispositivo
- Escaneo externo y activo de un dispositivo
- Escaneo externo y pasivo de un dispositivo

Como veremos en las descripciones de cada uno de los tipos, todos estos escáneres no son excluyentes (en el sentido de que la utilización de un tipo de escáner no invalida a los demás), ya que hay vulnerabilidades que se pueden detectar con un tipo de escáner pero no con otro. Por tanto, un buen administrador de sistemas utilizará cada uno de ellos para detectar diferentes tipos de vulnerabilidad.

5.2.1. Escaneo interno y activo de un dispositivo

El escaneo interno y activo de un dispositivo se refiere a la posibilidad de ejecutar el propio escáner dentro de la máquina que se pretende escanear.

Esta característica permite un escaneo de datos de bajo nivel, como pueden ser servicios específicos de la máquina, detalles de su configuración, el propio sistema de ficheros, así como información específica del software y sistema operativo que utiliza. Permite analizar si las cuentas creadas en la máquina escaneada tienen contraseñas por defecto, o incluso si no tienen contraseñas. También permite verificar si el sistema ya ha sido atacado, analizando la existencia de ficheros sospechosos o programas en ejecución con privilegios inadecuados.

Los motores de análisis de vulnerabilidades de este tipo están muy relacionados con el sistema operativo que evalúan, lo que provoca que su mantenimiento sea un tanto costoso y complica su administración en entornos heterogéneos.

Este tipo de escaneos se pueden realizar mediante escáneres basados en máquina y también utilizando escáneres basados en red con credenciales. Los primeros fueron los primeros en utilizarse para la evaluación de vulnerabilidades. Se basan en la obtención de la información mediante consultas al sistema o a través de la revisión de distintos atributos de este.

COPS

Uno de los primeros escáneres de vulnerabilidades en sistemas Unix fue COPS, una herramienta que se encargaba de analizar el sistema en busca de problemas de configuración típicos, como por ejemplo permisos erróneos de ficheros, directorios y servicios, etc.

Ejemplo

Un simple guión de sistema como el que se muestra en la figura 6 se encargaría de avisar mediante correo electrónico al administrador del sistema en caso de encontrar entradas anómalas en el fichero de contraseñas del sistema:

Figura 6. Ejemplo de escáner basado en máquina

```
#!/usr/bin/perl
$count==0;
open(MAIL, "| /usr/lib/sendmail mikal");
print MAIL "To: Administration\n";
print MAIL "Subject: Password Report\n";
open(PASSWORDS, "cat /etc/passwd |");

while(<PASSWORDS>) {
      $linenumber=$.;
      @fields=split(/:/, $_);
      if($fields[1] eq "") {
            $count++;
            print MAIL "\n***WARNING***\n";
            print MAIL "Line $linenumber has a blank password.\n";
            print MAIL "Here's the record: @fields\n";
      }
}

close(PASSWORDS);
if($count < 1) print MAIL "No blank password found\n";
print MAIL ".\n";
close(MAIL);
```

Alternativamente, también se puede realizar un escaneo interno y activo mediante un escáner de red, lo que se conoce como escáner basado en red con credenciales. De esta manera, el escáner, accediendo al sistema con las credenciales, normalmente a través de conexiones SSH, puede ejecutar los mismos controles que tradicionalmente se realizaban salvo con los escáneres basados en máquina.

Ved también

En el subapartado 5.2.2 veremos una descripción más detallada de los escáneres basados en red.

5.2.2. Escaneo externo y activo de un dispositivo

El escaneo externo y activo de un dispositivo es un tipo de escaneo que se realiza mediante las herramientas conocidas como escáneres basados en red. En este caso, dicho escaneo se puede categorizar como un escaneo sin credenciales, en el sentido de que el actor que escanea un dispositivo no tiene acceso a él. En esta situación, la información obtenida del proceso de escaneo es una información semejante a la que puede ver un atacante (obviamente, que no haya tenido acceso al dispositivo).

Este tipo de escáneres de vulnerabilidades se instala en una máquina que será la encargada de escanear distintos dispositivos de la red. A través de la red el escáner obtiene la información necesaria, mediante las conexiones que establece con el objetivo que hay que analizar. Esta característica facilita la instalación de los escáneres basados en red, dado que al instalarse en máquinas distintas a las que escanea no es preciso instalar ningún software concreto en los dispositivos que se pretenden escanear.

Este tipo de escáneres permite la detección de cortafuegos mal configurados, servidores web vulnerables, riesgos asociados a software utilizado en los servidores, así como los riesgos asociados a una mala administración tanto de los servidores como de la red. Cabe destacar que, a diferencia de los escaneos internos, el escaneo externo no permite detectar ciertas vulnerabilidades porque no tienen acceso al propio dispositivo (puesto que no posee las credenciales necesarias).

Como ya hemos comentado, a diferencia de los escáneres internos, que se sitúan en la misma máquina que se pretende escanear, una consideración especial que hay que tener en cuenta en los escáneres externos es su ubicación en la red, dado que la ubicación donde emplazaremos el escáner será determinante en su funcionamiento. Por ejemplo, si se sitúa el escáner detrás de un cortafuegos esto implica que los resultados del escaneo se vean filtrados por las propias reglas de éste. Si realizamos un escaneo de la red interna desde fuera del cortafuegos se estará analizando solamente las vulnerabilidades que se pueden explotar desde fuera de la red, pero no se tendrá información de las

posibles vulnerabilidades que se encuentran dentro de ella, una vez que un posible atacante haya conseguido burlar la seguridad del cortafuegos. Por este motivo, es muy importante tener en cuenta la topología de la red.

Es muy importante tener en cuenta la topología de la red para el posicionamiento del escáner, de modo que el escaneo de los diferentes escáneres situados en distintos puntos de la red nos permita tener una idea clara de las vulnerabilidades de nuestro sistema dependiendo de desde dónde se accede.

Dentro de los escáneres basados en red podemos encontrar distintos tipos, como por ejemplo escáneres de propósito general, escáneres de puertos, escáneres de servidores web o escáneres de aplicaciones web.

Nessus

Nessus es un escáner de vulnerabilidades activo de propósito general que puede trabajar tanto con credenciales (escaneo interno) como sin ellas (escaneo externo). Su gran popularidad se debe a que hasta su versión 3 se distribuía bajo licencia GPL (*general public license*) de GNU, pero actualmente su distribución es comercial mediante la compañía TENABLE Network Security. Sin embargo, sigue siendo el escáner de vulnerabilidades más utilizado.

Como vemos, una característica del escáner externo y activo es la utilización de la red para la realización del escaneo. Es muy importante tener en cuenta este punto, puesto que el tráfico que puede generar el escaneo exhaustivo de distintos dispositivos puede provocar un aumento sustancial en el volumen de tráfico en la red que provoque su saturación, pudiendo llegar a provocar una denegación de servicio de la red.

Otro aspeto importante que cabe tener en cuenta en la utilización de escáneres externos es la protección de la información obtenida del escaneo. Cuando se ejecuta un escaneo externo y activo se está generando un conjunto de información referente al dispositivo o dispositivos que se están escaneando, que puede ser interesante para un atacante, puesto que puede identificar posibles vulnerabilidades sin la necesidad de ejecutar análisis que puedan resultar sospechosos. Por este motivo, la información generada por el escaneo que circule por la red debe intentar protegerse, en la medida de lo posible, utilizando ténicas de cifrado.

Desde el punto de vista del funcionamiento, dos de las técnicas más utilizadas para la evaluación de vulnerabilidades basadas en red son las siguientes:

- **Prueba por explotación**. Esta técnica consiste en lanzar ataques reales contra el objetivo. Estos ataques están programados normalmente mediante guiones de comandos. En lugar de aprovechar la vulnerabilidad para acceder al sistema, se devuelve un indicador que muestra si se ha tenido éxito o no. Obviamente, este tipo de técnica es bastante agresiva, sobre todo cuando se prueban ataques de denegación de servicio.

- **Métodos de inferencia**. El sistema no explota vulnerabilidades, sino que busca indicios que indiquen posibilidades de ataque, tratando de detectar posibles deficiencias de seguridad en el objetivo. Este método es me-

Técnicas de inferencia

Ejemplos de técnicas de inferencia pueden ser la comprobación de versión de sistema para determinar si existe una vulnerabilidad, la comprobación del estado de determinados puertos para descubrir cuáles están abiertos, la comprobación de conformidad de protocolo mediante solicitudes de estado, etc.

nos agresivo que el anterior, aunque los resultados obtenidos son menos exactos.

5.2.3. Escaneo externo y pasivo de un dispositivo

El escaneo externo y pasivo de dispositivos es una técnica que combina las capacidades de escucha de los *sniffers* con las capacidades de análisis de los escáneres de vulnerabilidades activos para detectar vulnerabilidades en los sistemas.

Un escáner pasivo de vulnerabilidades se coloca en la red en una posición en la que se puede controlar el tráfico que viene de varios segmentos, de manera similar a lo que se haría con un sistema de detección de intrusos. El escáner pasivo escucha el tráfico en tiempo real y lo analiza mediante la comparación con un conjunto de reglas, como un escáner de vulnerabilidades activo, de modo que si se incumplen las reglas establecidas, se alerta al administrador de la red. Estas características permiten detectar vulnerabilidades de manera más continua que los escáneres activos.

Si bien un escáner de vulnerabilidades pasivo puede parecer lo mismo que un sistema de detección de intrusos, es importante destacar que las tareas de análisis de tráfico que realizan ambos son distintas.

Por ejemplo, si suponemos las miles de conexiones que se pueden realizar a un servidor web, un sistema de detección de intrusos deberá analizar todas y cada una de ellas para identificar un ataque, mientras que el análisis que lleva a cabo un escáner pasivo se puede llevar a cabo únicamente con el análisis (tan exhaustivo como se requiera) de una única conexión que tiene como destino el servidor que se pretende escanear.

Una de las principales ventajas de los escáneres pasivos es la poca incidencia que tienen sobre los sistemas que analizan. Dado que se trata de un análisis de la información que viaja por la red, los escáneres pasivos son muy poco intrusivos y no afectan al rendimiento del sistema que se está escaneando, hecho que puede ocurrir con los escáneres activos. Esta característica les permite ser utilizados en sistemas críticos en los que no se puede permitir una disminución del rendimiento o la eventual parada del sistema, que podría llegar a provocar un escaneo activo.

Más allá de esta ventaja, los escáneres de vulnerabilidades pasivos presentan otras características interesantes que mejoran algunos aspectos de los escáneres activos, si bien no se debe ver un escaneo pasivo como un sustituto de un escaneo activo, sino más bien un complemento. Como veremos, un escaneo pasivo puede proveer información para una mejor eficiencia de un escaneo activo.

Una de las principales ventajas de los escáneres pasivos es su capacidad de análisis continuo, característica que no presentan los escaneos activos (tanto internos como externos). Tal y como hemos comentado anteriormente, el modo de análisis del escaneo activo le confiere una visualización instantánea del estado del sistema que se analiza, que se asemeja a una fotografía de la situación de los sistemas en el instante preciso que se realiza el escaneo. Esto implica que una modificación del sistema analizado con posterioridad al escaneo activo (por ejemplo, por una actualización o la instalación de nuevo software) puede dar lugar a una vulnerabilidad sin que el proceso de escaneo lo detecte. Por el contrario, los escáneres pasivos analizan constantemente el tráfico alertando de posibles vulnerabilidades cuando estas se detectan.

Esta idea de continuidad en el análisis nos lleva hasta una diferencia temporal entre un escáner activo y uno pasivo. Los escáneres pasivos precisan un tiempo para la realización del análisis. Por ejemplo, hasta que el usuario *A* no se comunica con el servidor *B*, el escáner no puede analizar si el puerto por el cual el servidor *B* se comunica tiene algún servicio con una vulnerabilidad. Sin embargo, en un escaneo activo, el propio escáner es quien inicia la comunicación con *B* y por tanto, en cualquier momento, puede determinar si existe la vulnerabilidad en dicho puerto.

En este sentido, parecería que un escáner activo aventaja al pasivo dadas estas reflexiones. Sin embargo, es importante destacar que un servidor *B* no siempre puede responder cuando el escáner activo le interrogue. Además, en esta situación, se asume que el escáner activo tiene identificadas las máquinas que debe escanear. Esta suposición, si bien puede parecer adecuada, en ocasiones no es real, puesto que los administradores desconocen la existencia de máquinas o servicios que requieren su escaneo.

Ejemplo

Un usuario podría iniciar un servidor FTP en una máquina que no estuviera autorizada a ello y esto podría pasar inadvertido al administrador. Un escáner activo no podría detectar una posible vulnerabilidad en la versión de servidor FTP iniciada puesto que el escáner no analizaría dicha máquina, dado que supuestamente no tendría que estar albergando un servidor FTP. Sin embargo, el análisis de tráfico que realiza un escáner pasivo podría detectar la utilización del servicio de FTP en dicha máquina y analizar la posible existencia de vulnerabilidades.

Otra habilidad que proporciona el escaneo pasivo es la optimización del proceso de escaneo.

Ejemplo

Siguiendo con el caso anterior, un escáner activo podría intentar identificar al servidor FTP no autorizado simplemente realizando escaneos exhaustivos de todos los dispositivos del sistema (escaneando todas las direcciones IP de la red, o todos los puertos de los todos dispositivos, etc.). Sin embargo, esta tarea es sumamente ineficiente, por no decir imposible, en el caso de una organización con direcciones IPv6. En cambio, un escáner pasivo analizará solo los dispositivos y los puertos por donde circule tráfico. De hecho, el escáner pasivo puede complementar al escáner activo proporcionando la información necesaria de dónde realizar el escaneo activo.

Por último, otra de las ventajas de un escaneo pasivo es la posibilidad de análisis de vulnerabilidades del cliente en un entorno cliente-servidor. Dado que los escáneres pasivos analizan el tráfico de la red, estos pueden detectar vulnerabilidades en la parte de la comunicación del cliente. Esta es otra ventaja de los escáneres pasivos, dado que los escáneres activos se focalizan en el análisis de vulnerabilidades de la parte del servidor, descuidando la parte del cliente.

Si bien hemos visto distintas ventajas de los escáneres pasivos, existen también algunas limitaciones en su uso. Una de las principales desventajas de los escáneres pasivos es la dificultad de fijar su correcto emplazamiento. Al igual que los sistemas de detección de intrusos, la determinación del emplazamiento de los analizadores de red es de vital importancia para la efectividad del escáner, puesto que el tráfico que circule por el segmento de red donde se sitúa el analizador será el que proporcionará la información para el análisis.

Otra de las limitaciones que presentan los escáneres pasivos son la dependencia que tienen de los datos que analizan. Si el escáner se limita a analizar las cabeceras de los paquetes que circulan por la red para determinar, por ejemplo, el tipo de sistema operativo que se encuentra en una máquina, es posible que un atacante pueda manipular la información de los paquetes para que el escáner pasivo proporcione información incorrecta o genere tanta información que el escáner no pueda analizar. Es decir, el nivel de análisis que el escáner realiza de los datos que obtiene determinará la calidad de las alertas que genere.

Resumen

En el presente módulo hemos analizado las vulnerabilidades que podemos en contrar en el nivel de red, centrándonos espcíficamente sobre IPv4. Hemos centrado su clasificación en función de si estas afectan a protocolos locales, a interconexión de redes o a protocolos extremo a extremo.

Como hemos podido ver, en el terreno local, destacan como problemas que generan vulnerabilidades el hecho de que la información que circula por la red puede ser esnifada por cualquier usuario, o los mecanismos y protocolos existentes para la asignación de direcciones físicas a direcciones IP. En lo que se refiere a la interconexión de redes, hemos visto algunas de las vulnerabilidades que presentan los protocolos más utilizados en este segmento, como son el protocolo IP, el ICMP, el sistema de DNS y los protocolos de encaminamiento OSPF y BGP. Por otro lado, hemos visto algunas de las vulnerabilidades que afectan a protocolos de extremo a extremo, como lo son TCP y UDP.

Por último, hemos visto cómo los escáneres de vulnerabilidades pueden ayudar a detectar las vulnerabilidades de un sistema. Para ello, existen distintas técnicas que dependen del modo de funcionamiento del escáner. De este modo, los escaneos activos permiten obtener una imagen fija de las posibles vulnerabilidades del sistema en un instante de tiempo concreto, mientras que los escáneres pasivos analizan de manera constante el sistema para detectar posibles vulnerabilidades que quedarán al descubierto por el propio uso de dispositivos, protocolos o programas que las contengan.

Actividades

1. En los primeros apartados del módulo hemos repasado algunas vulnerabilidades importantes de TCP/IP centradas principalmente en IPv4. La introducción de IPv6 provoca que algunas de estas vulnerabilidades desaparezcan y que surjan nuevas vulnerabilidades. Buscad información por Internet sobre vulnerabilidades propias de IPv6 y detallad brevemente en qué consisten.

Ejercicios de autoevaluación

1. La posiblidad de enviar mensajes en *broadcast* puede suponer una vulnerabilidad en algunos protocolos de red. Nombrad al menos 3 vulnerabilidades o ataques diferentes que utilizan el envío de mensajes en *broadcast* e indicad a qué protocolo de red afectan.

2. De las siguientes afirmaciones indicad las que son falsas.
a) La facilidad para predecir los números de secuencia de TCP se considera una vulnerabilidad de seguridad.
b) UDP es un protocolo más seguro que TCP porque tiene menos vulnerabilidades conocidas.
c) Si se permite el envío de mensajes en *broadcast* se pueden hacer ataques de denegación de servicio en UDP.
d) Todas las anteriores son falsas.

3. En la figura 7 se muestra una captura de paquetes hecha con el *sniffer* Wireshark. Cada línea corresponde a un paquete capturado en una red local Ethernet. Las columnas son por orden de izquierda a derecha: número de paquete, tiempo en segundos, dirección de origen, dirección de destino, protocolo de más alto nivel incluido en el paquete, tamaño del paquete e información de su contenido. En este caso se trata de paquetes ARP; la información *Who has A? Tell B* nos dice que el contenido del paquete ARP está preguntando quién tiene la dirección IP *A* y que la respuesta tiene que ser enviada a la IP *B* (supuestamente quien ha hecho la petición). Comentad qué se ve en la figura. ¿Se está haciendo algún ataque?, ¿qué vulnerabilidades se explotan?

Figura 7. Ejemplo de captura de paquetes con Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.159? Tell 24.166.172.1
2	0.098594	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.172.141? Tell 24.166.172.1
3	0.110617	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.161? Tell 24.166.172.1
4	0.211791	Cisco_af:f4:54	Broadcast	ARP	60	Who has 65.28.78.76? Tell 65.28.78.1
5	0.216744	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.163? Tell 24.166.172.1
6	0.307909	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.175.123? Tell 24.166.172.1
7	0.330433	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.165? Tell 24.166.172.1
8	0.408556	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.175.82? Tell 24.166.172.1
9	0.455104	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.220.131? Tell 69.76.216.1
10	0.486666	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.168? Tell 24.166.172.1
11	0.504694	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.221.27? Tell 69.76.216.1
12	0.510684	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.174.184? Tell 24.166.172.1
13	0.540733	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.169? Tell 24.166.172.1
14	0.587308	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.174.181? Tell 24.166.172.1
15	0.662937	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.216? Tell 69.76.216.1
16	0.690450	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.172? Tell 24.166.172.1
17	0.692934	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.217? Tell 69.76.216.1
18	0.771600	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.217.186? Tell 69.76.216.1
19	0.792105	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.174.221? Tell 24.166.172.1
20	0.801633	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.218.94? Tell 69.76.216.1
21	0.806611	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.174.207? Tell 24.166.172.1
22	0.856709	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.222? Tell 69.76.216.1
23	0.884248	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.223? Tell 69.76.216.1
24	0.896756	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.176? Tell 24.166.172.1
25	0.931326	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.220.86? Tell 69.76.216.1
26	0.932294	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.224? Tell 69.76.216.1
27	1.063549	Cisco_af:f4:54	Broadcast	ARP	60	Who has 65.28.78.114? Tell 65.28.78.1
28	1.065493	Cisco_af:f4:54	Broadcast	ARP	60	Who has 65.26.92.195? Tell 65.26.92.1
29	1.104112	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.230? Tell 69.76.216.1
30	1.105552	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.172.6? Tell 24.166.172.1
31	1.131107	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.216.28? Tell 69.76.216.1
32	1.133591	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.174.177? Tell 24.166.172.1
33	1.133679	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.231? Tell 69.76.216.1
34	1.152139	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.173.181? Tell 24.166.172.1
35	1.182181	Cisco_af:f4:54	Broadcast	ARP	60	Who has 24.166.172.232? Tell 24.166.172.1
36	1.184173	Cisco_af:f4:54	Broadcast	ARP	60	Who has 69.76.223.232? Tell 69.76.216.1

Fuente: Wireshark Wiki: http://wiki.wireshark.org/SampleCaptures

4. Buscad información sobre las vulnerabilidades de red CVE-1999-0128 y CVE-1999-0513 e identificadlas con las que aparecen en este módulo.

Solucionario

Ejercicios de autoevaluación

1. Ping flooding (ICMP), smurf attack (ICMP), fraggle attack (UDP).

2. *b* y *d*.

3. En la captura se ven muchas peticiones de ARP con origen en la máquina con dirección MAC Cisco_af:f4:54 (00:07:0d:af:f4:54) y destino *broadcast*. En todas las peticiones se pide resolver direcciones IP diferentes y únicamente se incluyen dos dirección IP para recibir la respuesta. Aquí vemos varios casos interesantes.

Se está realizando un ataque generando muchos paquetes (más de 20 por segundo), pidiendo la resolución de diferentes direcciones IP. Esto puede provocar *ARP poisoning o flooding* en los *host* que reciben estas peticiones. Dado que la dirección IP que se pide varía siempre parece que el objetivo es simplemente provocar un inundamineto de la caché ARP. Por otra parte, vemos que la dirección IP origen de los paquetes (que no tiene por qué corresponderse con la dirección MAC) son dos direcciones concretas. Esto puede indicar un intento de denegación de servicio a los *host* que tienen dichas IP, ya que serán los que reciban todas las respuestas (algo similar a los ataques de fragle y smurff vistos en el módulo).

4. Tanto la vulnerabilidad CVE-1999-0128 como CVE-1999-0513 son vulnerabilidades que afectan al protocolo ICMP. Ambas se han descrito en el apartado 3 correspondiente a la interconexión de redes. Concretamente, la primera corresponde a la vulnerabilidad de Ping of death, mientras que la segunda identifica un Smurf attack.

Glosario

ARP (*address resolution protocol*) *m* Protocolo que permite resolver direcciones de protocolos de interconexión de red a direcciones de red.

BGP (*border gateway protocol*) *m* Protocolo de encaminamiento exterior utilizado en Internet.

CAM (*content addressable memory*) *m* Espacio de memoria de un encaminador de red en el que se establece un vínculo entre las direcciones MAC y los propios puertos del conmutador.

ethernet *f* Familia de tecnologías para redes de área local.

DNS (*domain name system*) *m* Sistema de nombres jerárquico y distribuido que permite asociar nombres de dominio a direcciones IP.

dirección IP *f* Dirección utilizada por el protocolo IP.

dirección MAC *f* Dirección física de red.

ICMP (*Internet control message protocol*) *m* Protocolo de control, principalmente para envío de mensajes de error, de TCP/IP.

IP (*Internet protocol*) *m* Protocolo para la interconexión de redes.

MITM (*man in the middle*) *m* El ataque de hombre a medio camino consiste en que el atacante se interpone en la comunicación entre el emisor y el receptor legítimos pudiendo analizar la información que estos se intercambian.

OSPF (*open shortest path first*) *m* Protocolo de encaminamiento utilizado internamente por muchos sistemas autónomos.

port security *m* Permite limitar el número de direcciones MAC asociadas a un puerto físico de un conmutador.

sniffer *m* Dispositivo o programa que permite obtener el tráfico que circula por un canal de comunicación, normalmente una red TCP/IP.

TCP (*transmission control protocol*) *m* Protocolo de transporte (extremo a extremo) de TCP/IP.

UDP (*user datagram protocol*) *m* Protocolo de transporte (extremo a extremo) de TCP/IP.

Bibliografía

Comer, D. (2006). *Internetworking With TCP/IP. Vol. 1: Principles Protocols, and Architecture*. 5.ª edición. Prentice Hall: New Jersey.

McClure, S.; Scambray, J.; Kurtz, G. (2009). *Hacking exposed 6: network security secrets & solutions*. McGraw-Hill.

Plummer, D. C. (1982). *An Ethernet Address Resolution Protocol – or – Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*. RFC 826. IETF, The Internet society [en línea] <http://www.ietf.org/rfc/rfc826.txt>.

Senie, D. (1999). *Changing the Default for Directed Broadcasts in Routers*. RFC 2644. IETF, The Internet society [en línea] <http://www.ietf.org/rfc/rfc2644.txt>.

Ziemba, G.; Reed, D.; Traina, P. (1995). *Security Considerations for IP Fragment Filtering*. RFC 1858. IETF The Internet Society [en línea] <http://www.ietf.org/rfc/rfc1858.txt>.