Software malicioso

PID_00245596

Erik de Luis Gargallo

Universitat Oberta de Catalunya

Índice

Introducción

APWG

APWG es una coalición internacional que pretende unificar una respuesta global a los delitos informáticos entre todos los sectores de la industria, el Gobierno, las fuerzas del orden y las ONG.

De acuerdo con el Informe de Tendencias en la Actividad de AntiPhising Working Group (APWG), durante el primer semestre de 2016 fueron detectadas 873.488 campañas de correo electrónico relacionadas con acciones de *phishing*. El *phishing* consiste en el envío masivo de mensajes que, aparentando provenir de fuentes fiables, intentan conseguir que el usuario proporcione datos confidenciales.

En los tres primeros meses de 2016, se registró el mayor número de casos de *phishing* hasta la fecha, 557.964. Las campañas de *phishing* estaban destinadas principalmente a empresas del sector servicios (43 % de los casos) y del sector financiero (16 % de los casos), con un promedio mensual de 418 ataques.

Las cifras hablan por sí solas: indudablemente, nos encontramos en un momento de crecimiento del *malware*, entendido como cualquier programa cuyo objetivo es causar daños en las redes y los sistemas informáticos. La palabra *malware* proviene de la composición de las palabras inglesas *malicious* y software, es decir, programa malicioso. En la actualidad resulta muy frecuente oír hablar de las famosas campañas de *ransomware*: software malicioso cuyo objetivo es lucrar al atacante por medio de rescates económicos.

Durante años, mucho software malicioso no ha sido tenido en cuenta porque tanto los usuarios convencionales como los administradores de sistemas de información han creído que los antivirus estándar se encargaban de ello.

Hoy, nunca más lejos de la realidad, es necesario no solo disponer de otras herramientas para hacer frente al potente *malware* actual, sino también conocer cómo actuar en caso de resultar infectados por este. Los objetivos del *malware* son muy variados, aunque entre ellos destacan la obtención de información, las estafas en línea o directamente dañar los sistemas, por poner tres ejemplos.

En este módulo estableceremos una clasificación básica de los distintos tipos de *malware* que nos podemos encontrar, así como de las clases de ataques que, ya sean causa de estos o no, podemos recibir como usuarios de un sistema informático.

Objetivos

Los materiales didácticos de este módulo proporcionan los contenidos y herramientas imprescindibles para lograr los objetivos siguientes:

1. Conocer las clases de software malicioso que hay en la actualidad.

2. Identificar los principales tipos de ataques presentes en Internet.

1. Clasificación del software malicioso

Tal y como se ha descrito anteriormente, el código malicioso o *malware* hace referencia a programas o software instalado en un ordenador o sistema informático (lo que incluye dispositivos electrónicos varios), con la intención de comprometer la confidencialidad, la integridad o la disponibilidad del sistema, las aplicaciones y los datos.

El hecho de que el software malicioso cambie con mucha frecuencia (se suele utilizar el término «mutar»), así como el diverso comportamiento que este tiene, hace difícil su clasificación. Cada día surgen nuevas muestras de *malware* susceptibles de mutar o transformarse adquiriendo nuevas funcionalidades.

Las vías de infección por *malware* son muy variadas: descarga de software no oficial, las redes de compartición de ficheros tipo P2P o el simple intercambio de documentos infectados, normalmente adjuntos a correos electrónicos o que están en lápices de memoria tipo USB.

A continuación se establecerá una clasificación básica sobre los tipos de *malware* más comunes que podemos encontrarnos.

1.1. *Backdoors* y troyanos

Se entiende por *backdoor* (puerta trasera) el software malicioso capaz de saltarse los controles de acceso, identificación y autenticación de usuarios, permitiendo accesos no controlados a un entorno.

Se le llama puerta trasera por analogía a lo que sería una supuesta entrada trasera o puerta de servicio en una empresa. En la entrada principal habría mucha seguridad (guardias, arco de control, cámaras,...), pero no así en la puerta de servicio. Cualquier atacante que conociese la puerta de servicio podría usarla para acceder a las instalaciones saltándose todos los controles de seguridad. Los *backdoors* aprovechan errores en el sistema para crear puertas traseras que permitan entrar en el sistema saltándose los controles de seguridad.

También recibe el mismo nombre el conjunto de instrucciones no documentadas dentro de un programa o sistema operativo (recordemos que los programas informáticos están formados por instrucciones), que permiten acceder o

tomar el control del equipo saltándose los controles de seguridad. Así, por lo tanto, los *backdoorkits* son *malware* cuyo objetivo es la explotación de dichas puertas traseras.

Mediante un *backdoor* podemos llevar a cabo una incursión ilegal en servidores o estaciones de trabajo; se puede llegar a tomar el control de los sistemas comprometidos, con los consiguientes daños que ello implica.

Remote Administration Tool

En inglés, *backdoor* también recibe el nombre de *remote administration tool* (RAT), aunque en principio los RAT son programas lícitos que permiten a los administradores de sistemas acceder a los ordenadores de los trabajadores de la empresa para solucionar problemas o instalar aplicaciones (también puede usarse para reparar ordenadores domésticos a distancia). Los RAT no deberían nunca ocultarse al usuario, pero podemos encontrar alguno que sí se oculta, en cuyo caso puede ser usado como *backdoor*.

Este tipo de *malware* tiene dos componentes principales:

- el programa Servidor, que se instala en el ordenador de la víctima (el servidor sirve los datos) y
- el programa Cliente, que actúa en la computadora del atacante (el cliente recibe los datos), que dispone de un programa que le permite ejecutar diferentes funciones.

Dentro de estas funciones podemos encontrar la eliminación de ficheros o la destrucción de la información del disco duro, la captura y el reenvío de datos confidenciales a un usuario no autorizado o la apertura de puertos de comunicaciones, lo que permitirá que un posible intruso controle el sistema de forma remota.

Es importante destacar que, mediante este *malware*, un atacante podrá ejecutar las mismas acciones que el usuario infectado; es decir, tendrá los mismos privilegios en el sistema. Por lo tanto, cuantos más privilegios tenga el usuario, más acciones podrá llevar a cabo.

Los troyanos o caballos de Troya, en recuerdo al antiguo caballo de Troya de la mitología griega, no pueden penetrar en los sistemas por sí mismos como hacen los virus, sino que se encuentran en el interior de otro software aparentemente inofensivo, que finge realizar una función útil para quien lo ejecuta, pero que en realidad ejecuta una acción generalmente dañina y que el usuario desconoce.

A diferencia de los virus, un troyano no se reproduce infectando otros ficheros ni se propaga haciendo copias de sí mismo como hacen los gusanos. Su finalidad, como en el caso de los *backdoors*, es la de recabar información de la máquina anfitriona sin que el usuario se dé cuenta y sin que conceda su consentimiento: causa desperfectos en el sistema, envía ilícitamente de forma masiva correos electrónicos o destruye datos con fines recaudatorios (troyano Cryptolocker).

Cryptolocker

Cryptolocker es un software malicioso que cifra un equipo o ciertos archivos almacenados en él para, posteriormente, pedir un rescate a cambio de la liberación de estos. Las técnicas utilizadas para instalarse en la máquina de la víctima suelen consistir en el uso de ingeniería social o la explotación de vulnerabilidades del software instalado.

Entre las funciones o acciones más conocidas, podemos destacar:

- Tomar el control del ordenador atacado.
- Utilizar el sistema infectado para atacar otros sistemas.
- Robar la información contenida en él. En este caso, recibe especial importancia la información sensible, como por ejemplo los datos bancarios (troyano Zeus).
- Realizar cambios en el sistema informático.

Zeus

Zeus es un troyano que infecta ordenadores con sistema operativo Windows con el objetivo de robar las credenciales bancarias u obtener otro tipo de información relevante. También permite al atacante ejecutar programas en el ordenador infectado. Una vez que un ordenador es infectado con este troyano, pasa a formar parte de una *botnet* o red zombi de ordenadores, con lo que puede ser utilizado para cometer acciones criminales o maliciosas.

Como norma general, los antivirus, a medida que se van descubriendo nuevas versiones o tipos de troyanos, almacenan sus firmas. Aun así, es posible que este *malware* sea capaz de modificar el comportamiento del antivirus y no ser detectado.

Firma

Una firma es una cadena de caracteres (números, letras y símbolos) que representa de forma inequívoca a un software malicioso.

1.2. Virus y gusanos

Los virus son programas que se reproducen copiándose dentro de otros archivos, de manera que cuando el usuario abre el archivo infectado ejecuta sin querer el virus, que aprovecha para infectar nuevos archivos. Por esta capacidad de copiarse a sí mismos se dice que son autorreplicables.

Los virus pueden ser desde molestos a destructivos. Por lo tanto, un virus es considerado *malware* primero por la infección de archivos que lleva a cabo, y segundo por la ejecución de acciones molestas o dañinas. Se suele considerar que los virus tienen dos objetivos bien definidos: producir daños y reproducirse. La infección tiene lugar cuando se ejecuta el código que contiene el virus.

Una de las principales vías de infección por virus ha resultado ser las cadenas de mensajes de correo electrónico que reenviaban presentaciones en PowerPoint. Dicho esto, podemos adelantar que uno de los mejores métodos para mitigar el riesgo es la concienciación del usuario. Se ha de mejorar el concepto de cultura de la seguridad.

La clasificación de los virus no siempre ha resultado sencilla, aunque, desde un punto de vista de afectación, los podríamos clasificar en:

- **dañinos**, impiden, por ejemplo, el arranque normal de un sistema,

- **invisibles**, en el sentido de que el usuario desconoce su existencia en el sistema aunque funcionen en segundo plano, y

- de ***script***, capaces de ejecutarse utilizando otro software normalmente relacionado con la ofimática.

Cierto que es que hace unos años, los virus representaban la principal amenaza para los equipos informáticos. Algunos de ellos han resultado tener renombre internacional, como, por ejemplo, Jerusalem o Michelangelo. Hoy en día, han quedado atrás en lo que se refiere al número de amenazas, en favor de otro tipo de *malware* como los troyanos, que ha cobrado mayor importancia.

Virus Michelangelo

El día 6 de marzo, el virus Michelangelo inutiliza el disco duro de los ordenadores infectados. El virus Michelangelo se transmite con disquetes infectados, y tuvo mucha relevancia en 1991, puesto que antes del 6 de marzo muchos ordenadores habían resultado infectados.

En la década de los noventa, junto con los virus aparecen los gusanos (*worms* en inglés). Este término, acuñado en 1975 en la obra de ciencia ficción de John Brunner *The Shockwave Rider*, hace referencia a programas capaces de «viajar» por sí mismos por medio de redes de computadores para realizar cualquier actividad una vez alcanzada una máquina destino.

Los gusanos son programas que se replican a sí mismos, de sistema a sistema, sin utilizar otros archivos para hacerlo. En esto se diferencian de los virus, que necesitan extenderse mediante un archivo infectado. Los gusanos utilizan los recursos de red para distribuirse.

Como *malware* que son, se distribuyen por medio de correos electrónicos, mensajería instantánea, redes de compartición de archivos (P2P), dispositivos de almacenamiento masivo, etc.

Dado que la finalidad principal del gusano es distribuirse, su principal función al penetrar en un equipo es obtener las direcciones de red que este pueda albergar para comenzar a enviar copias de sí mismo.

Mayoritariamente, los gusanos utilizan los correos electrónicos como medio de propagación, en muchos casos utilizando las agendas de direcciones del cliente de correo electrónico, por ejemplo.

Es tal el potencial de propagación de los gusanos que los considerados gusanos de red de rápida distribución han sido capaces de distribuirse en continentes en cuestión de minutos. Tal nivel de propagación viene dado por su sencillez y finalidad: primero, no necesitan infectar otros ficheros para reproducirse, ya que se propagan realizando copias de sí mismos, y segundo, tienen el fin de colapsar las redes en las que se infiltran.

El 2 de noviembre de 1988, Robert T. Morris saltó a la fama cuando uno de sus programas se convirtió en «el Gusano» con mayúsculas, en *The Worm* de Internet. Este *malware* aprovechaba vulnerabilidades en programas muy utilizados en el entorno Unix de la época y motivó que, en pocas horas, miles de equipos conectados a Internet dejaran de funcionar.

Algunos de los gusanos o *worms* más conocidos son: I Love You, Navidad, Pretty Park, Happy99, ExploreZip, Conficker, Stuxnet, Blaster o Sasser.

Gusano Stuxnet

En enero de 2010, el gusano Stuxnet tomó el control de mil máquinas que participaban en la producción de materiales nucleares en Irán y les envío instrucciones para que se autodestruyeran.

Se trata de un *malware* que aprovecha una vulnerabilidad de Windows para instalarse en el ordenador; una vez en él, realiza un ataque dirigido con el objetivo de recopilar información.

Aunque en 2011 se dijo que el gusano fue creado en un laboratorio por Estados Unidos e Israel para sabotear el programa nuclear de Irán, esta información nunca llegó a ser confirmada por los países en cuestión.

1.3. *Keyloggers y rootkits*

La palabra *keylogger* proviene de los términos ingleses *key*, que significa «tecla», y *logger*, que se traduce por «grabador». Los *keyloggers* son programas espía que toman el control de los equipos, para espiar y robar información registrando las pulsaciones del teclado y/o del ratón, para robar información como las contraseñas y los nombres de usuario.

Un *keylogger* es un tipo de software que tiene la capacidad de registrar en un *log* cada pulsación de teclado.

log

Un *log* es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema. En el caso de los *keyloggers*, la información que se añade son las distintas teclas pulsadas en el teclado del sistema infectado

Un *keylogger* puede registrar cualquier tipo de información escrita mediante la utilización del teclado como, por ejemplo, los mensajes instantáneos o los correos electrónicos. El archivo de *log* creado por el *keylogger* puede ser enviado a un destinatario concreto, utilizando una característica añadida al *malware* para transmitir los datos registrados en el ordenador objetivo, situando los datos disponibles en una ubicación remota.

Los *keyloggers* se clasifican en:

- ***Hardware keyloggers***: son aquellos dispositivos físicos que se encargan de detectar las distintas pulsaciones en el teclado.

- ***Software keyloggers***: son dispositivos tipo software, que, como en el caso anterior, registran las pulsaciones del teclado y/o del ratón.

Lógicamente, una de las maneras más sencillas y efectivas de evitar la instalación de dispositivos *keylogger* en su versión hardware es la prohibición del acceso físico a los sistemas. Evidentemente, realizar inspecciones visuales es el mejor método para descubrirlos.

En su versión software, cabe decir que las compañías de antivirus añaden en su base de firmas las de los *keyloggers*, detectando así la aparición de uno de ellos en caso de resultar infectados y advirtiendo al usuario de ello.

Haxdoor

Algunos de los *keyloggers* más conocidos son Revealer Keylogger y Haxdoor. Este último se utilizó para realizar un ataque al banco sueco Nordea (compañía bancaria resultado de múltiples uniones y adquisiciones de bancos suecos, finlandeses, daneses y noruegos), por medio de un troyano que encapsulaba un *keylogger*, que le hizo perder 1,5 millones de dólares, en agosto de 2006.

Los clientes de Nordea empezaron a recibir correos electrónicos que simulaban ser del banco, con ofertas para instalar un producto *antispam* supuestamente adjunto al mensaje. En el momento en que el usuario trataba de abrir el archivo y descargarlo en su ordenador, este se infectaba con el conocido troyano Haxdoor, que se activaba cuando las víctimas se registraban en el servicio en línea de Nordea. El troyano lanzaba entonces una notificación de error solicitando al usuario reingresar la información provista en el momento de registrarse. Luego, un *keylogger* que venía incorporado en el troyano grababa todos los datos ingresados por los clientes del banco, y acto seguido procedía a enviar toda la información recogida al atacante. De este modo, los atacantes accedían a las cuentas de los clientes y transferían los fondos que había en ellas.

Producto *antispam*

Un producto *antispam* evita que el correo basura (*spam*) inunde la bandeja de entrada de un cliente de correo electrónico con mensajes de correo no solicitados, sobre todo desplazando dicho correo no solicitado a una carpeta de correo destinada a tal efecto.

Los *rootkits*, más que un programa malicioso, son un conjunto de programas cuyo objetivo es permitir a un atacante o *hacker* pasar inadvertido mientras lleva a cabo su cometido, que consistirá principalmente en obtener un acceso no autorizado al sistema ocultando objetos como procesos, archivos o entradas del registro del sistema operativo.

Para alcanzar dicho propósito, este *malware* ejecutará uno de los dos métodos siguientes: o bien reemplazará bibliotecas del sistema que pretende infectar, o bien instalará un módulo en el *kernel* del sistema.

Kernel

El *kernel* de un sistema software constituye una parte fundamental del sistema operativo. Es el encargado de que el software y el hardware de un ordenador trabajen juntos.

Posteriormente, se instalará el *rootkit* que le permitirá una elevación de privilegios como usuario del sistema, hasta conseguir acceso como administrador.

Resumiendo, un *rootkit* es una colección de herramientas utilizadas para ocultar una intrusión y obtener acceso con privilegios de administrador a un sistema. Los *rootkits* pueden incorporar otras funciones ocultas que facilitan, entre otras cosas, el control remoto del equipo comprometido.

Uno de los *rootkits* más conocidos, en este caso para la plataforma Linux, es el *rootkit* «t0rn».

***rootkit* t0rn**

El *rootkit* t0rn está compuesto por un conjunto de herramientas diseñadas para hacer más difícil la detección y la extirpación de un intruso informático, e incorpora puertas traseras para proporcionar un subsiguiente acceso fácil al sistema por parte del atacante.

Este *rootkit* añade, además, un sistema estándar de borrado de *logs*, llamado «t0rnsb», un *sniffer* llamado «t0rns» y un modificador de *logs* denominado «t0rnp».

1.4. *Spyware* y *adware*

Se entiende por programa espía o *spyware* el software malicioso que recolecta información de un usuario sin que este haya dado su consentimiento.

En la mayoría de los casos, la obtención de información se lleva a cabo sin que el usuario sea consciente de la recogida. Normalmente recolectan información sobre el software que utiliza el usuario o sus acciones en el sistema, aunque se han dado casos en los que la recolección ha dejado paso a una redirección del tráfico, mostrando al usuario una página web que no se corresponde con la que este había solicitado, sino con la que el software malicioso ha querido mostrar.

Principalmente, el *spyware* recoge datos acerca de hábitos de uso de Internet del usuario y los envía a empresas de publicidad, las cuales pueden usar la información para enviarle anuncios publicitarios o vender su perfil a terceros. El *spyware* puede ser instalado en el sistema por numerosas vías, en muchos casos sin que medie consentimiento expreso del usuario, así como con su conocimiento o falta de este respecto a la recopilación y/o uso de los datos ya mencionados.

Para llevar a cabo su cometido pueden monitorizar el uso del teclado, analizar archivos del disco de forma arbitraria, espiar a otras aplicaciones como procesadores de texto, leer e interpretar las *cookies*, o monitorizar diversos aspectos del comportamiento del usuario, entre otros.

Cookies

Una *cookie* es un archivo de pequeño tamaño que conserva información acerca del uso de un sitio web, bien de forma temporal para una sesión, bien de forma persistente para identificar a un usuario en visitas sucesivas.

Las *cookies* persistentes se pueden utilizar como *spyware* para rastrear la navegación web del usuario.

Por otro lado, los *adware* son aquellos programas que muestran publicidad al usuario mediante múltiples ventanas, utilizando cualquier tipo de medio, como, por ejemplo, ventanas emergentes, *banners*, cambios en la página de inicio o de búsqueda del navegador, etc., sin que el usuario tenga ningún control sobre estas.

El *adware* se instala en el equipo al acceder a webs de contenido sexual, software pirata o publicidad.

También se suele encontrar acompañando a otro software que normalmente es gratuito; la publicidad aparece en la interfaz de uso. En algunos casos, dicho software también recolecta y envía datos del usuario de carácter personal.

Las recomendaciones para evitar este software malicioso, de forma preventiva, son que nunca se instale ni se acepte ningún tipo de *plugin* o complemento cuando se navegue por páginas web de dudosa reputación o que no nos transmiten la suficiente confianza.

Plugin

Un *plugin* es un programa que añade nuevas funciones a un determinado programa ya existente. En el caso que nos ocupa, un *plugin* suele añadir funcionalidades al navegador web.

2. Tipos de ataques

Es tal el número de las tipologías de intrusos o atacantes de sistemas informáticos, *hackers*, *crackers*, *sniffers*, *phreakers*, *spammers*, etc., que se convierte en una ardua tarea realizar una clasificación o, simplemente, completar la selección de los principales ataques informáticos existentes en la actualidad.

Como en la vida real, son varios los motivos por los que un individuo o grupo de individuos llevan a cabo actos ilícitos contra sistemas de información, ya sea un ordenador personal, el sistema informático de una empresa u organización, o incluso el de un estamento gubernamental.

A título informativo, a continuación se enumeran y se describen las principales fases de un ataque informático:

- **Reconocimiento**. Aunque no se puede considerar propiamente un ataque, dado que no provoca daño alguno, el atacante busca lucrarse de información como pueden ser las versiones de software instalado en el ordenador, el sistema operativo o el diseño de la red a la que está conectado, todo ello mediante métodos de escaneo de redes y/o sistemas.

- **Escaneo**. En esta fase se lleva a cabo la detección de las vulnerabilidades que le presenten la oportunidad de acceder al sistema. Estas pueden surgir de software instalado pero desactualizado; es decir, en el cual no se han instalado los parches adecuados, o bien servicios o puertos abiertos innecesarios o mal configurados ofrecidos por el sistema.

- **Acceso**. Es propiamente en esta fase cuando el atacante intenta lograr el acceso no autorizado al sistema. Para ello, utilizará la explotación de alguna de las vulnerabilidades localizadas en el punto anterior. Obviamente, dependiendo de la gravedad de la vulnerabilidad, el acceso tendrá más o menos repercusión en el sentido de que el atacante recibirá uno u otro privilegio en el sistema atacado.

- **Mantenimiento**. Una vez que el atacante accede al sistema, buscará poder mantenerse en él, ya que un simple reinicio del sistema por parte del usuario legítimo o simplemente por el propio dispositivo infectado podría hacer perder el privilegio conseguido por el atacante. Para llevar a cabo la permanencia en el dispositivo, se suele introducir otro software malicioso o modificar archivos del sistema para garantizar la persistencia del ataque.

- **Eliminación**. Aunque no siempre se lleva a cabo, en muchos casos el *hacker* o atacante decide borrar aquellas pruebas que puedan relacionarle con el acto ilícito. A tal efecto, el atacante suele tomar nota de las acciones

llevadas a cabo durante la explotación para después poder hacer reversible el acto, evidentemente con la intención de evitar posibles acciones legales contra él.

En los siguientes subapartados, veremos cuáles son los principales ataques informáticos a los que podemos tener que hacer frente en la actualidad. Como se ha dicho anteriormente, no pretende ser una compleja explicación de terminología informática de difícil comprensión, sino un breve apunte de los ataques más comunes, para poder entender la terminología básica del mundo de la seguridad de los sistemas informáticos interconectados, denominada «ciberseguridad».

2.1. *Spoofing*

El *spoofing* consiste, por definición, en la generación de tráfico utilizando un origen falseado. Eso quiere decir que, ante todo, se parte del falseado o suplantado de alguno de los actores participantes en la comunicación en red, dando así lugar a los distintos tipos de *spoofing* existentes.

Los tipos de *spoofing* más comunes son el IP *spoofing*, el ARP *spoofing* y el DNS *spoofing*, aunque en este subapartado veremos también el web *spoofing* y el *mailspoofing*.

Aunque queda fuera de estudio de este módulo, es importante conocer que existe un modelo en informática, denominado TCP/IP, formado por distintos protocolos, el cual rige, por norma general, las comunicaciones en las redes. Bien, pues, uno de los protocolos usados es el protocolo IP o *Internet Protocol*, que asigna una dirección IP, que es un número que identifica de manera lógica y jerárquicamente a un dispositivo dentro de una red que utilice dicho protocolo. Podemos decir entonces que cualquier dispositivo conectado a la red dispone de una dirección IP que lo identifica inequívocamente.

Dicho esto, el IP *spoofing* consiste en ocultar el origen del ataque, suplantando o falseando la dirección IP.

Así, un intruso ataca cambiando la dirección IP del paquete que contiene información que se trasmite por la red, haciendo parecer que este se ha originado en otro lugar. De esta manera, si modificamos un paquete poniendo en él una dirección IP de uno de los ordenadores autorizado a acceder a un sistema, podremos hacer llegar a dicho sistema, desde un dispositivo en principio no autorizado, aquellos paquetes que deseemos enviar.

A título de ejemplo, mediante el IP *spoofing* se podrían ejecutar acciones en un sistema informático, en nombre de un usuario que mantiene una sesión activa y lícita en él, suplantando su IP, como, por ejemplo, transferencias bancarias o compras en línea.

Para comprender el ARP *spoofing*, debemos introducir brevemente el protocolo ARP. Es un protocolo de comunicación que se encarga de resolver y mantener la tabla de equivalencias entre las direcciones MAC y las direcciones IP de las máquinas que se comunican en una red. Cabe recordar que, en el entorno *ethernet*, para poder enviar los paquetes de datos es necesario conocer la dirección MAC del destinatario con independencia de su IP.

MAC

La dirección MAC (en inglés *Media Access Control*) es una dirección que identifica una interfaz de red de hardware de manera única. La dirección MAC se compone de 48 bits.

Dado que todas las máquinas de una misma red local comparten el mismo medio y que, por lo tanto, son capaces de ver todo el tráfico originado en la red, las tarjetas *ethernet* incorporan un filtro que ignora todo el tráfico que no está destinado a ellas mismas. Esto se consigue ignorando aquellos paquetes cuya dirección MAC no coincide con la suya.

Ethernet

Ethernet es una familia de tecnologías basadas en el marco de las redes de ordenadores para redes de área local (LAN).

Dicho esto, no es difícil imaginar el objetivo del ARP *spoofing*. La autentificación ARP está basada, al igual que el IP *spoofing*, en la dirección fuente, es decir, la dirección de donde provienen los paquetes.

El proceso de ARP *spoofing* consiste en manipular la tabla ARP de las dos máquinas a las que se desean suplantar.

De esta manera, una vez modificadas las tablas ARP de ambas máquinas, estas empezarán a comunicarse sin darse cuenta de que toda la información es previamente obtenida y analizada por una máquina intermedia, que, una vez la obtiene, la reenvía a la máquina destino de la comunicación para evitar ser detectada. De esta forma, la comunicación es transparente para los dos dispositivos. Obviamente, la detección es muy sencilla: solo debemos analizar nuestra tabla ARP (se suele utilizar el comando `arp -a`) y observar que no tengamos dos máquinas con la misma dirección MAC.

El DNS *spoofing* va un poco más allá. A diferencia del caso anterior, en este caso utilizaremos un dispositivo de red normalmente ubicado fuera de nuestras redes: el DNS.

Sin entrar en muchos detalles, diremos que el DNS (Domain Name Server) es el corazón del Internet actual, en el sentido de que es el encargado de traducir los nombres de dominios de las webs a la IP del ordenador donde está alojada la página que estamos buscando por Internet. Es decir, imaginemos que queremos visitar la página de la UOC: www.uoc.edu. Nosotros escribimos dicha página en el navegador y este nos lleva hasta a ella. Pero, en medio de este proceso, ha habido un elemento de red, el servidor DNS, que ha traducido

www.uoc.edu (nombre de dominio) a la dirección 213.73.40.242 (IP), dado que como humanos que somos nos resulta mucho más sencillo recordar el nombre de dominio antes que la IP.

Los ataques de falsificación de DNS, o DNS *spoofing*, provocan un direccionamiento erróneo en los equipos afectados, debido a una traducción equivocada de los nombres de dominio a direcciones IP, lo que facilita la redirección de los usuarios de los sistemas afectados hacia páginas web falsas.

Para poder llevar a cabo dicho ataque, un atacante direcciona nuestra petición web (el nombre de dominio que hemos tecleado) a un servidor DNS ilegítimo que nos lleve a una página web que no se corresponde con la solicitada. Para que no lo detectemos seremos enviados a un sitio web erróneo que simulará ser el sitio web real. A partir de ahí, en esa falsa web introduciremos nuestros datos de usuario, que obviamente serán almacenados y analizados por el atacante. ¿Podemos imaginar las consecuencias si dicha web es la de nuestra entidad bancaria?

Con una finalidad similar a la del caso anterior, el *webspoofing* consiste en que el atacante crea un sitio web (falso) similar al que en la víctima desea entrar.

Los accesos a este sitio están dirigidos por el atacante (quizás utilizando el DNS *spoofing* anteriormente mencionado), permitiéndole monitorizar todas las acciones de la víctima: datos, contraseñas, números de tarjeta de crédito, etc. Como podemos ver, el DNS *spoofing* y el web *spoofing* se complementan a la perfección para culminar uno de los ataques compuestos más peligrosos de Internet.

Pero no solo el uso de un DNS *spoofing* nos puede llevar a ejecutar una web falsa. Hay otros métodos mucho más sencillos que permiten la infección de nuestra máquina, a partir de la cual llevar a cabo actividades ilícitas. Uno de los más simples, pero a la vez más peligrosos, es utilizar un virus para que desvíe la conexión de una víctima y sus datos mediante una página falsa con aspecto de verdadera.

El SMTP *spoofing* puede ocurrir de distintas maneras, pero todas tienen un resultado similar: un usuario recibe un correo electrónico que parece haberse originado por parte de un emisor lícito cuando en realidad fue enviado desde un emisor falso.

SMTP

El protocolo SMTP (*Simple Mail Transfer Protocol*) es el protocolo estándar de Internet para el intercambio de correo electrónico.

El SMTP *spoofing* es a menudo un intento de engañar al usuario para la divulgación de información confidencial como los nombres de usuario y sus contraseñas.

Son varios los ejemplos de *e-mails* falsos que podrían ser utilizados para engañar a un usuario. Veamos un par:

- Un correo electrónico que dice ser de un administrador del sistema que solicita a los usuarios que cambien sus contraseñas en una cadena especificada y amenazando con suspender su cuenta si no lo hace.

- Un correo electrónico que afirma provenir de una persona con autoridad que solicita a los usuarios que les envíe una copia de un archivo de contraseñas u otra información sensible.

A partir de ese punto, y de la misma manera que ocurría en los casos anteriormente estudiados, el atacante ya dispone de las claves de acceso necesarias para poder llevar a cabo las actividades ilícitas que desee.

2.2. *Sniffing*

Conceptualmente, el *sniffing* es algo muy sencillo de entender: es la captura del tráfico que circula por una red.

Los *sniffers*, también llamados *packet sniffer*, son programas que permiten capturar los paquetes que viajan por una red y pueden ser usados para propósitos legítimos e ilegítimos. Por ejemplo, un administrador de red puede usarlos para monitorear el flujo de tráfico en la red para asegurarse de que la red funciona adecuadamente. Pero, por otro lado, los *sniffers* también pueden ser utilizados por usuarios maliciosos para obtener información personal valiosa.

Para llevar a cabo sus finalidades, los *sniffers*, suelen emplear un equipo conectado a la red con su tarjeta de red configurada en «modo promiscuo», para poder procesar todo el tráfico que recibe aunque vaya dirigido a otros equipos.

Como vimos en el caso del ARP *spoofing*, hace falta modificar las tablas ARP, que contienen las direcciones MAC de las máquinas que mantienen una comunicación, para poder obtener los paquetes de datos que estas se mandan. Bien, pues, las direcciones MAC que modificar vienen impuestas de fábrica en cada tarjeta de red de nuestros dispositivos.

Conservar la confidencialidad de nuestra información es uno de los puntos clave para mantener la seguridad de nuestros datos. Por eso, y haciendo frente a posibles ataques en red como el *sniffing*, el cual permitiría la lectura de

nuestros datos como si se tratara de un libro abierto, se recomienda el uso de protocolos de cifrado que cifren todos los datos confidenciales que se envían por medio de Internet o de nuestra red (por ejemplo, los correos electrónicos).

Por tal motivo, en la actualidad son pocas las páginas web que utilizan el protocolo HTTP (Hyper Text Transfer Protocol Secure). Lo emplean los navegadores para comunicarse con los servidores web, permitiéndonos ver las páginas web. El problema de este protocolo es que los datos circulan en «texto plano», es decir, sin cifrar.

Por ese motivo, nació el protocolo HTTPS (Hyper Text Transfer Protocol Secure). En sí mismo, el HTTPS no es más que el HTTP sobre SSL/TLS (Secure Sockets Layer/Transmission Layer Security); son dos protocolos para enviar paquetes cifrados por Internet.

De esta manera, y dado que se cifran todos los datos de HTTP, no solo la página web, sino también dirección de una página en Internet, los parámetros enviados o las *cookies*, aunque un usuario estuviera escuchando nuestras comunicaciones no entendería los datos que de ella estuviera capturando y viendo. Esos datos resultan totalmente ilegibles.

2.3. *Phishing*

El *phishing* consiste básicamente en el robo de información a un usuario suplantando o falseando una entidad legítima en la que este a priori confía.

Un ejemplo de *phishing* podría consistir en que un atacante replique la página web de una entidad bancaria con la que normalmente trabajamos. En el momento en que accedemos a la web falsa (que creemos que es la verdadera) con nuestro usuario y nuestra clave, el atacante dispondrá a su antojo de nuestros datos de acceso, con lo que solo debe acceder a la verdadera web de la entidad bancaria y operar con nuestras cuentas libremente.

En estos casos tan sencillos, el atacante solo debe preocuparse de dos cosas. Primero, encontrar un dominio web muy parecido al que quiere falsear; por ejemplo, si desea suplantar www.elbanco.com, puede substituir la «l» del nombre por un «1», quedando de la siguiente manera www.e1banco.com, y dificultándose enormemente ver el cambio en la URL.

Y, en segundo lugar, realizar un envío masivo de correos electrónicos basura o *spam* esperando que coincida que alguno de los receptores de los mensajes crean que «su banco», por medio de ese correo electrónico les solicita acceder mediante un enlace incorporado en dicho mensaje, para realizar cualquier

operación (da igual el motivo; es una excusa para poner el usuario y la contraseña). Por supuesto que ese correo electrónico buscará parecer igual a la correspondencia legal enviada por la entidad bancaria original.

Los primeros casos de *phishing* conocidos solían ser bastantes burdos en el sentido de que las páginas web utilizadas para falsear las webs originales estaban bastante mal elaboradas: se veían errores ortográficos por doquier. Solían elaborarlas ciberdelincuentes extranjeros que utilizaban traductores en línea cuyas traducciones dejaban mucho que desear. A menudo, daban mensajes de error muy sospechosos una vez introducidos el usuario y la contraseña.

Pero en la actualidad su desarrollo ha evolucionado enormemente. La ciberdelincuencia ha mejorado sus traducciones, ya que hay ciberdelincuentes en todos los países, a la vez que los traductores en línea automáticos también han mejorado sus prestaciones. Además, se ha depurado la interacción con el usuario para no levantar sospechas. Así, una vez que hemos introducido nuestro usuario y nuestra contraseña, la página falsa suele reenviar la petición a la página verdadera. Resulta un proceso transparente para el usuario engañado y no levanta ningún tipo de sospecha.

Por tal motivo, no deben sorprendernos noticias como que los casos de *phishing* en Estados Unidos aumentan un 400 % durante la declaración de impuestos, o que Paypal o Ebay son algunas de las compañías que más casos de *phishing* acumulan año tras año.

Aunque queda fuera del contenido de este módulo, en la actualidad se suele hablar también del *pharming*. A título informativo, tener presente que el *pharming* se trata de una táctica fraudulenta que consiste en cambiar los contenidos del DNS, ya sea por medio de la configuración del protocolo TCP/IP o del archivo correspondiente en cada sistema operativo, para redirigir los navegadores a páginas falsas.

Es importante recordar que, como en la mayoría de los ataques vistos, las personas suelen ser el eslabón más débil de la cadena, susceptibles a ataques como el *phishing*, por lo que la concienciación en el ámbito de la seguridad resulta fundamental.

2.4. Inyección SQL

Antes de entrar propiamente en materia, es importante entender a qué nos referimos cuando hablamos de SQL.

Una base de datos es un conjunto de ficheros que contiene datos y los programas que gestionan la estructura y la forma en la que estos se almacenan, así como la forma en la que deben relacionarse entre sí. Algunos ejemplos de sistemas de bases de datos son: Access, Oracle, SQL, etc.

El SQL o Structured Query Language es, por lo tanto, un lenguaje de programación estándar, destinado a la gestión, administración y comunicación de bases de datos, utilizado para interactuar con bases de datos relacionales. La unidad típica de ejecución de SQL es la consulta o *query*, que es un conjunto de instrucciones que permiten modificar la estructura de la base de datos. Es decir, mediante una consulta o *query*, interactuamos con una base de datos para consultar, modificar, añadir o borrar información en ella.

El ataque por inyección de código SQL consiste en un ataque de introducción de consultas en el que un atacante puede ejecutar sentencias SQL maliciosas en un servidor de base de datos.

Se produce porque no se filtra de forma adecuada la información enviada por el usuario. Desde una vulnerabilidad de inyección SQL se puede llegar a afectar a cualquier aplicación que haga uso de una base de datos basada en este lenguaje de programación. Esta vulnerabilidad es una de las más antiguas, más frecuente y más peligrosa de entre las distintas vulnerabilidades que se suelen ejecutar sobre aplicaciones web.

Al aprovechar una vulnerabilidad de inyección SQL, y con las circunstancias adecuadas, un atacante puede recuperar el contenido de una base de datos completa. La inyección SQL también se puede utilizar para añadir, modificar y eliminar registros en una base de datos, hecho que afecta a la integridad de estos.

Como consecuencia de estos ataques y, dependiendo de los privilegios del usuario de base de datos bajo el cual se ejecutan las consultas, se podría acceder no solo a las tablas relacionadas con la operación de la aplicación del servidor web, sino también a las tablas de otras bases de datos alojadas en el mismo servidor web.

Dado que el SQL es un lenguaje de programación, la manera de evitar este tipo de ataques consiste en programar adecuadamente la base de datos y en depurar el código para evitar su correspondiente explotación o brecha de seguridad.

2.5. *Cross-site scripting* (XSS)

Un ataque de *Cross-site scripting* (XSS) consiste básicamente en la ejecución de código *script* o un pequeño programa arbitrario en un navegador, en el contexto de seguridad de la conexión a un determinado servidor web.

Script

El término *script* hace referencia a todos aquellos ficheros o secciones de código escritas en algún lenguaje de programación, como Visual Basic Script (VBScript), JavaScript, etc.

Los ataques XSS ocurren cuando un atacante utiliza una aplicación web para enviar código malicioso, generalmente en forma de un *script* del lado del navegador, a un usuario final. Esta inyección de código HTML o Javascript donde no debería haberlo hace que el atacante obtenga algún provecho, como, por ejemplo, el robo de *cookies*, el reenvío a una página falsa, etc.

Javascript

El Javascript es un lenguaje de programación que aporta características dinámicas (datos variables en función del tiempo y el modo de acceso, interactividad con el usuario, personalización, etc.) a las páginas web, escritas en lenguaje HTML.

Un atacante puede usar XSS para enviar un *script* malicioso a un usuario desprevenido. El navegador del usuario final no tiene manera de saber que sus acciones no son fiables, así que se ejecutará la secuencia de comandos porque cree que la escritura vino de una fuente de confianza. Así, el script malicioso puede acceder a las claves de sesión u otra información sensible retenida por el navegador y utilizada en ese sitio.

Dado que, en este tipo de vulnerabilidades, los atacantes explotan la confianza que un usuario tiene en un sitio en particular, en primer lugar es fundamental contar con una solución de seguridad instalada y actualizada. Esto es importante, ya que, ante la ejecución de alguna aplicación maliciosa sin nuestro consentimiento, tal como *malware* o *exploits*, automáticamente será bloqueada. Además, si se trata de una redirección a algún sitio de *phishing*, se cuenta con la protección del antivirus y el bloqueo proactivo por parte de los navegadores.

En segundo lugar, es importante observar la dirección URL a la que accedemos. Como se ha comentado, uno de los métodos de engaño consiste en la redirección a páginas web falsas o del interés del atacante, por lo que, observando la barra de direcciones del navegador, veremos si estamos moviéndonos por el dominio correcto o no.

2.6. DoS y DDos

Un DoS o *Denial of Service* es un ataque contra un recurso con el objetivo de degradar total o parcialmente los servicios prestados por este a sus usuarios legítimos.

Dicho de otra manera, consiste en inutilizar un dispositivo de red, normalmente esencial, con la intención de anular o reducir la funcionalidad del servicio ofrecido por dicho dispositivo. Mediante estos tipos de ataque, la persona que lo realiza puede llegar a obtener acceso a un sistema, ya que el ataque dispara otros tipos de *bugs* en un sistema.

Bug

Un *bug* es un término que se emplea para indicar un fallo o error en un programa informático.

Imaginemos, por ejemplo, una página web de una organización a la que realizamos miles de peticiones de conexión a la vez. Si el servidor de la página o un dispositivo de red intermedio no están preparados para hacer frente a la gestión de un número de peticiones tan elevado, lo más probable es que dicho servidor quede fuera de servicio.

Algunas de las oportunidades que ofrece el ataque DoS son:

- Una ejecución que eleve el consumo de los recursos de las máquinas afectadas: procesador, memoria, disco duro, etc., provocando una caída en su rendimiento. Para ejecutarlo es necesario el establecimiento de múltiples conexiones simultáneas o ataques lanzados contra los puertos de configuración de los encaminadores, entre otros.

- Colapsar redes de ordenadores mediante la generación de grandes cantidades de tráfico, generalmente desde múltiples equipos.

Ping de la muerte

A título de ejemplo, hablaremos de uno de los casos conocidos de DoS más importantes: «el ping de la muerte». El comando `ping` (Packet Internet Groper) se utiliza para comprobar si hay errores de red. El funcionamiento del mecanismo es muy simple y trabaja mediante el envío de series de información a una dirección IP, máquina o servidor. Así, la máquina fuente envía una solicitud de eco, es decir, una petición de contestación, a una máquina destino.

La máquina destino recibe la solicitud y envía una respuesta a la máquina fuente y otra información de interés. El comando de ejecución es `ping` *`dirección_IP_equipo_a_evaluar`*.

Bien, pues, mediante el comando `ping -l 65510` *`dirección_IP_equipo_víctima`*, el atacante envía un paquete IP de un tamaño superior a los 65.535 bytes, con el fin de colapsar el sistema atacado. Obviamente, en la actualidad, la mayoría de los sistemas están protegidos contra un ataque tan sencillo de realizar, pero llegó a afectar a la mayoría de los sistemas operativos, como Unix, Linux, Mac, Windows, a las impresoras y a los encaminadores.

Por otro lado, un DDoS o *Distributed Denial of Service* es una denegación de servicios distribuida, es decir, un ataque de denegación de servicios (DoS) realizado al mismo tiempo desde varios ordenadores.

Un DDoS está causado por un ataque masivo y simultáneo a un único objetivo real. En la actualidad es uno de los ataques más extendidos. Resulta lógico pensar que si, a cambio de una sola máquina atacante, disponemos de cientos o quizás miles de ellas, las probabilidades de éxito aumentan considerablemente.

Para llevarlo a cabo, en primer lugar debemos comprometer un número de máquinas lo suficientemente elevado para poder ejecutar el ataque sin problemas. Dado que no todo el mundo está dispuesto a facilitar su máquina para tales fines, resulta obvio que el atacante intentará infectar el mayor número de máquinas que le sea posible, resultándole así un «ejército» de máquinas *zombies*. Estos equipos *zombies* serán equipos infectados por algún virus o troyano que permitirá el control remoto por parte del usuario atacante.

Zombie

Un *zombie* es un ordenador controlado mediante la utilización de *bots*. Un *bot* es el software malicioso utilizado por el atacante para el control del ordenador infectado. Así, a la red de ordenadores infectados, se la denomina *botnet*.

Por este motivo, no solo es importante la protección ante un DDoS, sino también la seguridad de nuestros ordenadores como usuarios conectados a Internet, ya que sin darnos cuenta podemos convertirnos en cómplices colaboradores de un ataque masivo contra cualquier organización en la red.

Casos de DDoS

Son numerosos los casos de DoS o DDoS ejecutados anualmente en el mundo. Así podemos encontrar noticias como «Varios ciberataques masivos inutilizan las webs de grandes compañías», publicada en el periódico español *El País* el día 22 de octubre de 2016, donde se comentaba lo siguiente:

«Una oleada de ciberataques masivos contra el proveedor de Internet Dyn interrumpió este viernes el servicio de páginas web de grandes compañías y medios de comunicación internacionales, como Twitter, Spotify, Amazon, Netflix o The New York Times. El ataque, planeado en varias fases, duró casi once horas.

»El Departamento de Seguridad Nacional de Estados Unidos ya avisó la semana pasada de que los *hackers* estaban utilizando un nuevo enfoque "muy poderoso" para lanzar estas campañas. Este nuevo sistema consiste en infectar *routers*, impresoras, televisiones inteligentes y todo tipo de objetos conectados con un *malware* que los convierte en una especie de "ejército robot" que pueden lanzar los llamados ataques DDoS.

»Este tipo de ataques de denegación de servicio satura con datos inservibles a los servidores —en este caso, los de Dyn—, de manera que impide a los usuarios reales acceder a las páginas por la sobrecarga del ancho de banda provocada por la acción de los *hackers*. El servidor no puede atender la cantidad enorme de solicitudes».

Resumen

En la primera parte de este módulo se han mostrado los principales tipos de software malicioso que podemos encontrarnos. Se han estudiado los *backdoors*, los troyanos, los virus, los gusanos, los *keyloggers*, los *rootkits*, el *spyware* y el *adware*. Hemos visto que en algunos casos este *malware* trabaja de forma conjunta, incrementando así su capacidad maliciosa.

En la segunda parte del módulo se han enumerado y descrito las principales fases de un ataque informático, que son: reconocimiento, escaneo, acceso, mantenimiento y eliminación.

También hemos visto los distintos ataques que podemos encontrarnos. Los principales ataques estudiados han sido el *spoofing*, el *sniffing*, el *phishing* (haciendo referencia al *pharming*), la inyección SQL, el *Cross-Site Scripting* (XSS) y el *DoS* y *DDoS*.

Se han puesto ejemplos de algunos de ellos para mostrar la actual permanencia y constancia en el día a día de estos ataques, cuya actividad no baja, sino que se incrementa a medida que las organizaciones criminales se dan cuenta de los beneficios económicos que suponen sus actos ilícitos.

Actividades

1. En la actualidad, una de las mayores amenazadas en cuanto a los *malware* a la que está expuesta una organización, Gobierno o empresa, es la denominada «amenaza persistente avanzada», cuyas siglas en inglés son APT (*advanced persistent threat*). ¿En qué consiste este tipo de *malware*? ¿Por qué se caracteriza principalmente? ¿Conoces algún caso de APT que se hiciera público?

2. En el subapartado 2.3 hemos definido el *phishing* como el robo de información a un usuario suplantando o falseando una entidad legítima en la que este *a priori* confía. Dicho esto: ¿qué es el *spearphishing*? ¿Qué parecido tiene con las APT del primer enunciado? ¿Conoces algún caso de *spear phishing* que se hiciera público?

Bibliografía

Abliz, M. (2011). *Internet denial of service attacks and defense mechanisms*. Department of Computer Science, University of Pittsburgh.

Arbor Networks, Inc. (2016). *Worldwide Infrastructure Security Report*. Arbor Networks.

Aycock, J. (2005). *Computer Viruses and Malware*. Springer.

Cole, E.; Krutz, R.; Conley, J. (2005). *Network Security Bible*. John Wiley & Sons.

Richardson, R. (2010). *2010/2011 CSI Computer Crime and Security Survey*. Computer Security Institute.

Enlaces de Internet

https://www.symantec.com/connect/articles/introduction-spyware-keyloggers

https://ecrimeresearch.org/

https://apwg.org/report-phishing/

http://virusattack.blogspot.com.es/2007/08/qu-es-un-keylogger-leccin-9.html

http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf

https://www.nist.gov/publications