MALWARE E INGENIERÍA SOCIAL

Estas notas de apoyo tienen como objetivo suplementar las transparencias y facilitar la comprensión de los contenidos vistos en clase. Son, por tanto, un complemento y no un sustituto de estas.

1 ATACANTES

Antes de adentrarse en profundidad, conviene analizar el perfil del atacante para entender su contexto, motivaciones o capacidad real de causar daño.

Originalmente y en los medios, se ha hecho uso de la palabra **hacker** para definir a un atacante informático. Incluso se subdividido en categorías en función de sus motivaciones: *black hat hacker* (de sombrero negro) para alguien maligno, *white hat hacker* (de sombrero blanco) para alguien que intenta defender sistemas o encontrar fallos para corregirlos e, incluso, *grey hat hacker* para un *hacker* que, en función de sus intereses, actúa a veces como white hat y otras como black hat. Realmente, la palabra *hacker* tiene un significado mucho más amplio y no necesariamente negativo en inglés, por lo que se considera preferible la palabra *atacante* independientemente de sus intenciones.

Las atacantes pueden ir desde *script kiddies* (literalmente nenas/es de script) hasta espías o mercenarios bajo contrato con un objetivo mucho más preciado y específico. También se debe considerar que, en muchos casos, los atacantes provienen del interior del perímetro: empleados de la propia empresa o empresas subcontratadas o colaboradoras que tienen alguna motivación para dañar a la empresa.

En función del nivel de conocimientos, podemos distinguir atacantes de distintos niveles:

- **Élite**, son capaces de utilizar métodos muy avanzados e, incluso, desarrollar ataques propios, además de poseer la capacidad de utilizar recursos muy extensos (capacidad de computación, almacenamiento, ancho de banda, número de nodos, etc.). Estos podrían ser un estado, una multinacional, un atacante muy experimentado con un apoyo económico significativo, etc.
- **Moderado**, si bien tienen conocimiento suficiente para modificar ataques existentes y pueden tener acceso a recursos no triviales (más allá del alcance de un usuario normal), no llegan al nivel anterior.
- **Bajo**, utilizan métodos sencillos, no tienen conocimientos suficientes para modificar ataques existentes ni son capaces de acceder ni emplear recursos más allá de los de un usuario normal.
- **Ninguno**, se corresponde con un usuario básico medio sin conocimientos ni experiencia concreta en seguridad.

En la transparencia 4, se puede ver la distribución (aunque puede variar en función del año) de ataques en función del nivel de conocimientos del atacante. Tiene sentido que el grueso de los atacantes sea de nivel moderado o bajo, puesto que los atacantes de nivel élite van a ser mucho más selectivos a la hora de elegir objetivos y realizar ataques mientras que los atacantes sin conocimientos nunca van a suponer una gran parte de los ataques.

2 MALWARE

El **malware** (o *malicious software*, software malicioso) es un tipo de software que tiene objetivos malignos de algún tipo. Se puede clasificar bien por el mecanismo de propagación que emplee, por el mecanismo de ocultación e, incluso, si su objetivo principal consiste en algún tipo de beneficio económico.

Cabe destacar que la mayoría del malware no tiene una clasificación única y utiliza un enfoque mixto empleando diferentes características de varias de las clases de malware que tratamos a continuación.

2.1 PROPAGACIÓN

En función de la propagación, el malware se puede clasificar en virus o gusanos.

Si bien la comparación con los virus biológicos es bastante lejana, un **virus informático** es un programa cuyo mecanismo de propagación consiste en insertar su código dentro del código de otros programas. De esta forma, el código del virus se ejecuta también al ejecutar uno de estos programas *"infectados"* por el virus, de ahí el nombre.

La principal característica de un virus es que requiere de la acción de un usuario para saltar de máquina: un usuario puede copiar el programa infectado a otro ordenador y ejecutarlo, o descargarlo de la red, etc. Pero es importante destacar que el virus sólo está activo un breve lapso de tiempo cuando se ejecuta el programa infectado y luego cede el control al programa original, por lo que no está activo todo el tiempo ni es capaz de actuar de forma autónoma. *Sólo se ejecuta cuando el usuario o cierto evento activa el programa infectado.*

Los virus informáticos pueden emplear diferentes técnicas de inserción para intentar pasar desapercibidos frente al análisis de antivirus o software antimalware, algunas de ellas se pueden ver en la figura de la transparencia 8.

La técnica más básica, de apéndice, consiste en añadir el código del virus al final del ejecutable e incorporar una instrucción de salto al principio para que nada más iniciarse la ejecución se salte al código del virus y luego de nuevo al código del programa original. Evidentemente, esta técnica crea una estructura muy reconocible por los antivirus (un salto al final del ejecutable que no suele ser frecuente en un programa no infectado).

Para evitar que sea tan reconocible, la técnica de queso suizo (por los agujeritos) consiste realizar una secuencia de saltos más larga (con distintas variaciones de dirección, etc.) antes de llegar al código del virus. Si bien esto lo hace más difícil de detectar, un antivirus avanzado puede analizar el código para seguir todos los saltos y descubrir una estructura bastante similar a la de apéndice.

Por último, se muestra la técnica de fraccionado, en la que el código del virus se distribuye a lo largo de todo el código del ejecutable de forma más o menos homogénea y se llama de forma desordenada, simulando una especie de estructura de llamada y retorno a procedimientos o funciones. Esto lo hace mucho más difícil de detectar.

Además de utilizar diferentes esquemas de inserción, se pueden utilizar otras técnicas que dificulten la detección del virus: compresión de código, código que varía (muta) con cada inserción (infección), cifrado del código con una clave aleatoria, etc.

También se pueden distinguir diferentes tipos de virus en función de cómo infectan (transparencia 9). El más común sería el virus de programa que se inserta en ejecutables convencionales y se activa al ejecutar el programa; puede sorprender que hay muchas más extensiones que *.exe*, *.com* o *.dll* que pueden

contener código ejecutable y, por tanto, un virus informático. Si bien hay herramientas automatizadas, desarrollar un virus de programa desde cero requiere un conocimiento muy elevado del sistema operativo, de la arquitectura del procesador y otros aspectos.

Otro tipo de virus muy común es el de macro; estos explotan la capacidad de ciertos documentos (office, etc.) que permiten la ejecución de algún tipo de lenguaje de programación incorporado en el documento y que se activa al abrirlo en la aplicación correspondiente. Son muy comunes como adjuntos de email y se suelen corresponder con un nivel de atacante más básico.

Los virus de arranque se insertan en el sector de arranque del disco duro o dispositivo de almacenamiento lo que aporta dos ventajas claras: se ejecuta siempre al arrancar el ordenador (y puede quedar residente en memoria) y además se arranca antes del sistema operativo lo que permite afectar o evitar algunos aspectos del sistema operativo que puedan permitir la detección del virus. Este tipo de virus implica un conocimiento muy avanzado y se desarrollan casi siempre en ensamblador dado el espacio limitado del que se dispone. Funcionan parecido a un sistema de arranque de sistema (conocido como *Bootstrap,* LiLo o Grub son ejemplos populares) pero instalando el virus en memoria antes de ceder el control al arranque legítimo del sistema.

Un virus asociado consiste en suplantar un comando o herramienta legítima del sistema operativo en lugar de infectar a un ejecutable. Se engaña al usuario para que ejecute el virus y luego se cede el control al programa correcto. Es un buen punto de entrada para un virus de programa convencional. Como curiosidad, cabe destacar que, en Windows, si hay dos ejecutables en el *path* con el mismo nombre, pero extensiones distintas, se ejecuta primero la extensión *.com* antes que la *.exe*; esto permite suplantar a un ejecutable *.exe* de forma relativamente sencilla. Otro ejemplo sería en sistemas tipo Unix, que si hay dos ejecutables con el mismo nombre en el *path*, se ejecuta primero aquel que esté en el directorio actual; esta es la razón por la que, en muchos sistemas, hay que escribir *"./ejecutable"* y no simplemente *"ejecutable"* para ejecutar algo del directorio actual, evitando así una suplantación de un comando del sistema por un ejecutable maligno que se haya puesto en el directorio actual.

Por el contrario, un **gusano** es un programa que explota una vulnerabilidad para entrar en una máquina a través de la red y, una vez que se ha copiado a dicha máquina, busca nuevos objetivos vulnerables a los que atacar. De esta forma, los gusanos son programas autónomos, están activos todo el tiempo y no requieren de la acción de un usuario para propagarse. Esto les confiere la capacidad de recibir órdenes remotas y ser controlados a distancia o por eventos de red.

Uno de los primeros gusanos y más populares fue el de Robert Tappan Morris que creó un gusano en 1988 con el objetivo de parchear una vulnerabilidad de ciertos sistemas, pero cometió un fallo en el código que evitaba que el gusano infectara múltiples veces la misma máquina y al final lograba una denegación de servicio involuntaria agotando los recursos de la máquina. A pesar de ser de los primeros, utilizaba técnicas bastante inteligentes: mandaba el código fuente y se compilaba en destino, lo que lo hacía relativamente multiplataforma. No salió demasiado mal parado teniendo en cuenta el gran impacto (involuntario) de su malware, fue un caso muy mediático en su época.

En la transparencia 11 se resume en una tabla las principales diferencias entre virus y gusanos. Cabe resaltar de nuevo que hay malware que presenta características de ambos, pero es importante saber las diferencias entre ellos. Las acciones maliciosas que pueden llevar a cabo son similares (transparencia 7), con la salvedad de que un gusano, además, permite recibir órdenes remotas y actuar como un atacante en sí mismo (bajo control o autónomo).

2.2 OCULTACIÓN

Dentro del malware cuyo objetivo es la ocultación, vamos a distinguir troyanos, rootkits, bombas lógicas y puertas traseras. Al igual que en la propagación, la mayoría del malware también emplea diversas técnicas y un enfoque mixto respecto a la ocultación.

Un **troyano** o caballo de Troya es un programa que tiene una actividad oculta no aparente. El nombre proviene del relato del caballo de Troya empleado por los aqueos. Puede consistir en una aplicación que realiza algo útil como un calendario, una calculadora o un informe climatológico que, además de realizar su función principal, realiza otra negativa de forma oculta[1]: enviar datos privados, modificar la configuración de ciertas aplicaciones, instalar otro software sin conocimiento del usuario, etc. Lamentablemente, en los medios y la literatura es muy común llamar troyano a malware que no lo es; lo que crea cierta confusión al respecto.

Un **rootkit** es un malware que tiene como principal objetivo ocultar la actividad de otro malware frente a los usuarios, sistema operativo y antivirus/antimalware. Es fácil imaginar que, con estas propiedades, es muy preciado como complemento para otro malware. Funcionan modificando el sistema operativo para que ignore ciertos ficheros o procesos. Su desarrollo requiere un nivel muy avanzado y pueden funcionar modificando partes del sistema operativo (más sencillos de implementar, pero menos potentes) o funcionando por debajo del sistema operativo. Esta última alternativa los hace muy difíciles de detectar y de eliminar, así como muy potentes; pueden funcionar de forma similar a un virus de arranque (y por lo tanto interceptando la comunicación entre el sistema y el hardware) o suplantando el firmware de ciertos dispositivos o componentes. Un caso muy popular fue el de los CD de Sony, donde llegaron a utilizar rootkits como sistema (draconiano) antipiratería.

Una **bomba lógica**, es un programa o parte de un programa que permanece inactivo hasta que se satisface una condición preestablecida y realiza una actividad maliciosa. Por este motivo, son muy difíciles de detectar antes de su activación. No deben confundirse con ciertas bromas o secretos no malignos que se incluyen en programas o juegos a modo de guiño a los usuarios (huevos de pascua o *easter eggs*); uno de los más populares es del Excel 95, que incluía un juego tipo *Doom* al realizar ciertas acciones concretas. Las bombas lógicas son un recurso frecuente para los atacantes internos, en las transparencias 16 y 17 se describen tres casos concretos[2] en los que se utilizó una bomba lógica para realizar un ataque.

Una **puerta trasera**, es un programa o una parte del código de un programa que permite el acceso ignorando las restricciones normales de seguridad. Puede ser intencionada o no y a veces cierto malware tiene como acción el habilitar una puerta trasera para ataques posteriores. Un caso bastante curioso es la charla que impartió Ken Thompson (uno de los creadores de Unix y de Go) cuando le entregaron el premio Turing, en la que explicaba que, tal vez, habría introducido una puerta trasera en el compilador original de C y, por ende, en todos los programas compilados con dicho compilador (lo que probablemente incluye todos los compiladores de C existentes de forma indirecta, etc.).

2.3 BENEFICIO

Una **botnet** consiste en un gran número de máquinas infectadas con un malware que permita el control remoto por parte de un atacante. El nombre viene de red de *bots* (robots, también llamados *zombis*). El número de máquinas que puede llegar a conformar una botnet es muy elevado; hay que tener en cuenta

[1] Resulta relativamente curioso como muchas de las redes sociales populares encajarían en nuestra definición de troyano, si bien la actividad negativa (minería y rentabilización de los datos de los usuarios) no es algo en sí oculto, sino que elegimos ignorar voluntariamente en cierta forma.

[2] A partir de los cuales, sacamos la conclusión tangencial de que salía más a cuenta una bomba lógica en el sector de defensa que en otro tipo de empresas.

que se suele emplear una estrategia exponencial: cuando una máquina queda bajo control de la botnet, esa máquina puede emplearse para encontrar nuevos objetivos que serán añadidos también a la botnet, resultando en una expansión muy rápida. Se han contabilizado redes con millones de bots.

El mecanismo de control puede ser variado. Originalmente, muchas botnet utilizaban IRC (un protocolo de chat textual muy utilizado en los inicios de internet) ya que es un protocolo relativamente simple: los bots se conectaban a un canal privado y el atacante mandaba comandos mediante ese chat. Como IRC no es tan popular hoy en día, resulta relativamente fácil de filtrar o detectar; por ello, en la actualidad se suele emplear HTTP: los bots acceden a una web determinada (generalmente alojada en dominios preconocidos o generados con un cierto patrón preestablecido) y la interpretan para establecer los comandos a ejecutar. Otra opción es consultar los posts de ciertos usuarios en redes públicas como Twitter o Instagram e interpretarlos para extraer los comandos de control. Al contrario que IRC, estos métodos permiten asincronía entre la publicación de la orden del atacante y la ejecución por parte de los bots, dificultando significativamente la trazabilidad de dichos ataques.

Las botnet son una plataforma ideal para realizar ataques ya que otorgan un cierto grado de ocultación para el atacante, suelen estar activas durante un largo período de tiempo y, aunque algunas máquinas puedan dejar de funcionar (se apagan o se elimina el malware), siempre está activo un porcentaje significativo de las máquinas de la red. Los ataques posibles son innumerables, entre otros:

- Para realizar *spam* era necesario encontrar una vulnerabilidad en algún servidor de correo que permitiera enviar muchos mensajes con distintos emisores, etc. o sortear otro tipo de limitaciones. Con una botnet, basta con que cada bot envíe uno o varios emails para lograr una masa crítica enorme.
- La propia botnet puede servir para *distribuir malware*, buscando objetivos potenciales y atacándolos de forma distribuida, lo que dificulta mucho su detección y filtrado.
- También se puede emplear para *manipular* encuestas o juegos. Por ejemplo, indicando a la botnet que vote por una cierta opción y convertirla en la más popular o que realice visitas a cierta página para que los ingresos por publicidad sean mayores.
- El ataque más común realizado por una botnet es la *denegación de servicio*. Si bien requería explotar alguna vulnerabilidad de los protocolos que permitiera amplificación de tráfico (que el atacante consuma poco tráfico pero que el atacado reciba una cantidad de datos intratable), con una botnet se simplifica enormemente ya que basta dirigir cada máquina de la red a la misma página o al mismo servicio para provocar una sobrecarga de peticiones que aparentan ser legítimas y, además, proceden de diferentes orígenes (*denegación de servicio distribuida*) lo que dificulta enormemente distinguir el tráfico correcto.

En muchos casos, no es necesario tener los conocimientos para desarrollar una botnet propia, se pueden alquilar o comprar.

El **spyware**, es software que espía a los usuarios obteniendo información privada sin su permiso. Uno de los objetivos más comunes es recabar el historial de visitas para luego realizar publicidad dirigida. No obstante, se pueden emplear otras técnicas que involucren seguimiento por imágenes a través de webcam, de la actividad del usuario (procesos, ficheros…), etc. Cabe destacar que el spyware es un problema cada vez más importante al incrementarse el uso de dispositivos con cámara, micrófono y localización.

El **adware** es un concepto similar al spyware. Aquí el objetivo es realizar publicidad no deseada al usuario y, como conocer los gustos del usuario permite realizar publicidad más efectiva, se suele combinar con spyware.

Tradicionalmente, un **keylogger** es un dispositivo hardware que permite almacenar todas las pulsaciones de teclado y por tanto descubrir las contraseñas, números de tarjeta y otra información privada que se introduzca mediante teclado. Este concepto se puede extender también a software y otros dispositivos que no sean el teclado (por ejemplo, los movimientos de ratón o capturas de pantalla, etc.). Cabe destacar que los receptores inalámbricos de teclado y ratón, si no utilizan un protocolo de comunicación seguro, pueden permitir capturar la información de forma remota sin tener que instalar un dispositivo o software en la máquina local.

También es interesante como muchos servicios de banca han empezado a obligar a teclear el *PIN*, no permitiendo copiar y pegar desde el portapapeles. El motivo es que éste suele ser compartido entre todas las aplicaciones por lo que, si se copia algo secreto, todos los demás procesos pueden leer el contenido del portapapeles. También existe el enfoque inverso: no permitir teclear y obligar a seleccionar los dígitos en pantalla con el ratón, etc.; pero nada impide a alguien capaz de colocar un keylogger capturar también los movimientos de ratón o la pantalla. Ambos enfoques tienen ventajas e inconvenientes.

El **ransomware** es un concepto que ha adquirido cierto auge en la actualidad. El ransomware antiguo utilizaba técnicas básicas de extorsión como bloquear el equipo si no se satisfacía cierta demanda, pero es recientemente cuando se ha empezado a utilizar criptografía y, sobre todo, criptografía de clave pública para hacer casi insalvable uno de estos ataques (Moti Yung y Adam Young introdujeron el concepto de "criptovirología" en 1996). La idea es que se cifra el disco (o ficheros) con una clave aleatoria y esta se guarda cifrada con la clave pública del atacante, por lo que no se puede obtener sin su cooperación. Los casos recientes más populares han sido *cryptolocker* y *wannacry* y, como los beneficios obtenidos han sido significativos, se esperan muchos más ataques de este tipo en el futuro. La mejor defensa ante el ransomware es disponer de una estrategia de *backup* sólida.

3 INGENIERÍA SOCIAL

La ingeniería social es un nombre técnico para indicar el aprovecharse del engaño o de las debilidades del comportamiento humano para realizar ataques de ciberseguridad.

La **suplantación de identidad** consiste en hacerse pasar por otra persona o usuario. Puede ser mediante un enfoque activo, usurpando un rol (técnico, gestor, etc.) que permita obtener información o manipular ajustes o sistemas de seguridad; o un enfoque relativamente más pasivo, en el que se suplanta a alguien de cierta capacidad especial y se espera a que se solicite información, indicando información incorrecta que permita o facilite el ataque.

El **phishing** consiste en simular una fuente legítima (mediante web o email, etc.) con el objetivo de que el usuario se confíe y proporcione voluntariamente información privada. El nivel de detalle con el que se copian los recursos legítimos es, muchas veces, indistinguible para el usuario medio. Hay variaciones de phishing en función del método o el objetivo: *pharming* que implica la redirección del tráfico a cierta web, *spear phishing* en el que se es muy selectivo con el objetivo, *whaling* en el que sólo se buscan objetivos muy lucrativos, *vishing* cuando se realiza *phishing* mediante vídeo, etc.

El **spam** consiste en enviar publicidad no solicitada en masa (generalmente mediante email) con el objetivo principal de conseguir ventas, aunque sea en un porcentaje pequeño de los casos. Y ahí estriba el problema, ya que el beneficio económico puede sorprender si se analiza (como se indica en la transparencia 29) aunque el porcentaje de eficacia sea mínimo. Cabe destacar que, en muchos casos, no es el propio fabricante o vendedor quien realiza el spam, sino que son agentes comerciales que van a comisión en las ventas los que intentan maximizar su beneficio mediante técnicas de spam. Al igual que con el phishing hay toda suerte de nombres para distintos tipos, como el Spim (spam sobre mensajería instantánea). También se ha utilizado el sistema de avisos de *bluetooth* para hacer spam a los viandantes y, en algunos casos, los sistemas recientes de *beacons* se pueden subvertir en fuentes de spam.

El **hoax** consiste en engañar mediante mensajes o información falsa y lograr, de esta forma, una seguridad reducida para realizar el ataque o engañar para habilitar otro tipo de ingeniería social. Este tipo de actividad maliciosa suele ir en combinación con otras de las técnicas descritas previamente.

4 AMPLIACIÓN

En el canal *computerphile* de YouTube tienen un vídeo acerca de WannaCry. Esta canal tiene otros muchos vídeos acerca de seguridad (y otros temas de informática) y se suelen poder activar los subtítulos en español.

Existen muchos materiales en O'Reilly Safari (acceso gratuito con la cuenta de la UA) para ampliar, entre otros:

- Libro "Malicious Cryptography: Exposing Cryptovirology" (Moti Yung et al.)
- Libro "Practical Malware Analysis"
- Topic "Malware"