Universidad Nacional Autónoma de México

Facultad de Ciencias

Tarea Semana 3: Metodología para pruebas de penetración

Miguel Concha Vázquez
416062401

Tarea presentada en cumplimiento con la asignatura de Análisis de Software Malicioso impartida por el profesor
Jonathan Banfi Vázquez
16 de febrero de 2019

Índice

1. Fases y herramientas de la metodología para pruebas de penetración

1.1. Introducción

Día con día se emprenden ataques informáticos que atentan contra la seguridad de las empresas, haciéndose con activos como su información o la de sus empleados. Para poder vulnerarlas, los criminales primero tienen que analizar a fondo sus puntos débiles para poder explotarlos adecuadamente. Debido a que las técnicas utilizadas por los maleantes cambian a grandes velocidades en este mundo tan tecnológico, los protocolos ya no pueden detener en seco a los tan variados vectores de ataque como sucedía unos cuantos años atrás. Por esta razón se desarrollaron algunos estándares y metodologías que permiten a profesionales del área de seguridad llevar a cabo pruebas de penetración a las empresas y compañías (*pentesting*) cuando son contratados para esto[1], haciéndoles ver en última instancia cuáles son los fallos en sus políticas y cómo podrían tratar de enmendar sus fragilidades al simular ciberataques reales.

A grandes rasgos, en la metodología para las pruebas de penetración podrían enumerarse cinco fases, mismas que son parte de un ciclo continuo que permite una constante mejora en la seguridad informática:

- Reconocimiento: Se obtiene información preliminar de la víctima de forma directa (reconocimiento activo) o indirecta (pasivo) a través de un intermediario para planear el ataque de forma exitosa.
- Escaneo: Se profundiza en el estudio del objetivo, pero desde un punto de vista más técnico que podría involucrar un escaneo de los puertos o escaneos de vulnerabilidades de los equipos de las víctimas.
- Obtención de Acceso: En este punto se lleva a cabo el ataque por medio de su explotación, con tal de tomar control del equipo y extraer información o bien para poder lanzar ataques.
- Mantenimiento del acceso: Se trata de mantener el control o la entrada al equipo penetrado para poder conseguir más información o dar pie a nuevas vulnerabilidades.

[1] En este contexto, las víctimas de los ataques serán aquellos que contratar a los *pentesters*.

- Ocultado de huellas: Se esconden las evidencias del ataque para que sea más difícil para un forense hallar pistas de su ocurrencia y así evitar ser detectados. También tiene que ver con redactar reportes que detallen puntualmente la penetración llevada a cabo.

Sin embargo, el estándar para pruebas de penetración (*PTES, Penetration Testing Execution Standard*) que funge como la norma adoptada por los líderes de la comunidad de la seguridad informática identifica y define claramente un total de siete fases que se revisarán en las siguientes subsecciones, ahondando de igual forma en las herramientas en las que se suele auxiliar cada una de ellas.

1.2. Fases del estándar *PTES*

1.2.1. Interacciones de contacto

Oficialmente, el nombre de la fase es *pre-engagement interactions*. Es sumamente importante definir correctamente el **alcance** de una prueba de penetración antes de efectuarla, además de aclarar los objetivos concretos y términos junto con el cliente, el dónde, cuándo y el porqué. Durante esta fase se recomienda entonces aclarar varios puntos de la prueba misma para que así no se vayan a rebasar barreras que podrían perjudicar realmente a la empresa. Asimismo puede incluir los procedimientos necesarios para aislar a los equipos que serán atacados para evitar potencialmente daños mayores y aclarar con el cliente cómo y a quiénes deberán ir dirigidos los reportes finales para poder saber qué tantos tecnicismos incluir o el propósito de cada documento.

También es fundamental hacer notar el aspecto legal de la prueba a quien contrata al *pentester* para entender que ciertos ataques y uso de *software* en algunas jurisdicciones sería ilegal y de esta forma poder buscar alternativas que se apeguen a la ley del lugar en donde se efectuará la prueba de penetración. Los clientes deben ser a su vez claros con cuáles procedimientos sería mejor no meterse para no poner en jaque políticas o normas internas de la organización[2]. Por su puesto, también se tienen que negociar en este punto y ponerse de acuerdo en torno a los recursos e infraestructura que deberá ser provista, el costo de todo el proceso y el tiempo que se deberá destinar

[2]Podría ser el caso que el cliente tenga contratado el servicio de una empresa de seguridad inoformática (*MSSP, Managed Security Service Provider*) y entonces sería crucial analizar los contratos para no violarlos y posiblemente avisarles igualmente de la prueba de penetración. También es importante notificar a los equipos de repuesta.

para la prueba par evitar entrometerse en las operaciones empresariales.

En esta fase el *pentester* también debe preparar todos las herramientas y el *software* necesario para dar cabida a la prueba misma, lo que dependerá en gran parte de los factores, alcance y reglas discutidas previamente. Por lo general es preferible no tener que instalar programas en los ambientes de prueba, sino que el experto cargue con sus propias herramientas.

Algunas de las herramientas más comunes que deberán reunirse en esta fase son:

- Virtualizador: Será importante contar con *software* como `VMware`, `VirtualBox` o `Parallels` con tal de virtualizar el equipo de cómputo y contar con avarios sistemas operativos dentro de una misma estación de trabajo.
- Sistemas operativos: Escoger adecuadamente el sistema operativo que se usará en la prueba de penetración puede serun factor crítico para su éxito. Por lo general será conveniente tener acceso a una distribución de *Linux* por la amplia gama de herramientas que tiene[3], o bien un *Windows XP/7* que tiene herramientas de penetración disponibles para ser descargadas y son comerciales.
- Herramientas de radio frecuencia: Esto engloba a los escáners de frecuencias, contadores de frecuencias, analizadores de espectros, antenas externas, *USB GPS* y los adaptadores *USB 802.11* para *Wi-Fi*. Estos últimos permiten una conexión sencilla de un adaptador de red al sistema para la prueba de penetración, pero en general las herramientas mencionadas sirven para poder examinar y estudiar composiciones espectrales de ondas eléctricas y ópticas, además de analizar señales y propagaciones y así determinar si los transmisores cumplen o no con los estándares.
- *Software* complementario: Se utilizan programas tanto de código abierto como comerciales para ayudar en la prueba. Tres de los más comunes serían:
  - `Maltego`: La herramienta más usada para minar datos de individuos o empresas, además de ofrecer una versión gratuita.
  - `Nessus`: Se trata de un escáner de vulnerabilidades que se utiliza para hallarlas y documentarlas al ser usado dentro de una red dada.

[3] La mayor[ia de las pruebas de penetración se llevan a cabo con este SO.

- `Nexpose`: Es de los que desarrollan `Metasploit` y también hace un escaneo de vulnerabilidades como en el caso previo.

1.2.2. Recogida de información

También podría traducirse como *reunido de información* y su nombre en inglés es *intelligence gathering*, pero muchas veces también es conocida como **OSINT** (*Open Source Intelligence*). Es el punto en que comienzan a tomarse medidas para acercase a la víctima desde diferentes perspectivas, y todo para poder conocer cómo funciona internamente la organización objetivo y entender cómo podría ser atacada efectivamente:

- Activa: Requiere de una mayor planeación porque deja rastros que podrían ser advertidos por las víctimas. En este escenario, se hacen mapeos de la infraestructura de la red, se enumeran puertos y servicios activos que podrían ser vulnerables, se buscan directorios no enlistados y se podrían escanear también aplicaciones web.

- Semi-pasiva: El perfil de las víctimas se va formando a partir de métodos que podrían pasar inadvertidos para un incauto pues podrían simular tráfico y comportamiento común en la red. Básicamente incluye hacer consultas a servidores para buscar información relevante y analizar los metadatos de archivos disponibles.

- Pasiva: Aquí definitivamente queda descartada la opción de enviar tráfico de red a la víctima y el *pentester* tiene que limitarse a juntar información de documentos y archivos almacenados que pueden ya ser viejos o tener información incorrecta puesto que puede venir de terceros. A grandes rasgos se buscan nombres de empleados, números telefónicos y direcciones de correo electrónico, enlaces a otros sitios asociados a las empresas, ramas y ubicaciones de las compañías, blogs de empleados descontentos y la información de la política de seguridad interna.

A su vez, algunos métodos clásicos para reunir información incluyen:

- Ingeniería social: Como se investigó en la primera tarea, consiste en engatusar a las víctimas o personas cercanas a ellas para conseguir información valiosa. Dependiendo de cómo se lleve a cabo, podría ser considerada como pasiva o activa, pero puede auxiliarse a su vez de otras herramientas como sitios web (*Pipl, PeekYou, Spokeo*) para tener acceso a direcciones de correo y números telefónicos, del *vishing*

para obtener información privilegiada mediante engaños por teléfono, del *thrashing* para buscar en documentos olvidados el divulgado de información sensible que dé pistas acerca de la estructura interna de la empresa, o de ténicas como *eavesdropping* y *shoulder surfing* para ver y escuchar información importante al mezclarse en lugares públicos. Aquí se suelen usar las siguientes herramientas:

  - `Touchgraph`: Un servicio que permite modelar las interacciones sociales de distintos individuos.
  - `Maltego`: Ya mencionada también en la fase previa, puede contribuir a organizar lo averiguado de una forma más lógica para crear perfiles de las personas.
  - `Perfil de Hoovers`: Es un servicio que conjunta muchos datos de organizaciones, compañías y servicios desde un punto de vista más simplista del negocio en cuestión.

- Uso de `Google`: Podría pensarse como una práctica que entraría en el rubro de la ingeniería social, pero hoy en día es una práctica tan usada que debemos comenzar a identificarla como algo separado. En este sentido, el *pentester* usa información de búsqueda provista por el motor que es conocida como *Google Hacks* o como *Google Docks*. A grandes rasgos consiste en saber y utilizar correctamente varios operadores de búsqueda en conjunto al hacer consultas a *Google* que permitan conocer vulnerabilidades y configuraciones erróneas de los sitios web, así como proporcionar pistas de la infraestructura de la red. Con ésto podrían por ejemplo descubrirse páginas que no son normalmente mostradas por al algoritmo de búsqueda normal. Una herramienta clásica que es usada para este propósito es `Google Hacking Database`, misma que lista varias consultas de búsqueda (*dorks*) para encontrar información de una amplia gama de sitios web.

- Análisis del `DNS`: En caso que los servidores de nombres (*Domain Name System Servers*) encargados de resolver las consultas de nombres de dominios a direcciones lógicas IP estén mal configurados, se puede aprovechar la información que filtran para conocer más de los servidores. Estos datos podrían incluir por ejemplo varias direcciones IP, la fecha de creación del sitio, el dueño del dominio, otros servidores de nombres, etcétera. Se puede conseguir desde alguna herramienta en Internet como `DNS Stuff`, `Domain Tools` o `DNS Watch` o bien con las utilerías provistas por sistemas operativos como *Linux* y *Windows*:

- *Nslookup*: Utilizada para obtener información de al hacer consultas al servidor `DNS`, ya sean consultas tradicionales (*forward*) o inversas (*reverse, dirección IP a nombre de dominio*).
- `ipconfig`: En *Windows* muestra información de los registros del `DNS` como su tipo y cuáles sitios web han sido visitados por el equipo desde que se creó el *caché* por última vez.
- Comando `host`, `dig`, búsquedas con `WhoIs`: Todos enfocados a conocer más información de un dominio, o en el último caso, un servicio de búsqueda en una base de datos que podría arrojar direcciones de correo y números telefónicos.

Algunas otras herramientas generales usadas en esta fase tan importnte con gran regularidad son:

- `Netcraft`: Herramienta en línea que brinda información de las tecnologías usadas en el lado del servidor y el cliente.
- `MetaGoofil`: Permite extraer metadatos de una gran variedad de archivos y tratarlos como `HTML` o `JSON`, por ejemplo.
- `Threatagent`: Un sitio que necesita de registro, pero en donde unos *drones* ayudan a extraer la información solicitada de un sitio web y te la entregan en forma de reporte.

1.2.3. Modelado de amenazas

Se estudian a profundidad dos elementos primordiales: los activos y los atacantes (como comunidades y como agentes). Su objetivo es proveer claridad en cuanto a qué activos son más importantes para el cliente y cuáles podrían ser las comunidades de atacantes que representen el mayor riesgo al formar un perfil de estos. También la permite al *pentester* enfocarse de mejor manera en emular un ataque informático real con las herramientas, técnicas, capacidades y accesos que pueda tener el atacante de acuerdo al perfil creado.

El modelo de amenazas es creado debe ser creado —siempre que sea posible— en coordinación con la empresa que contrata el *pentester*. De igual forma, debe estar completamente bien documentado y entregado como parte de los documentos finales de la prueba. Las subfases identificadas son:

1. Reunir documentos relevantes.

2. Identificar y categorizar a los axtivos como primarios y secundarios.

3. Identificar y categorizar las amenazas y a las comunidades de amenaza.

4. Hacer una correlación entre las comunidades de amenaza y los activos identificados.

Algunas herramientas que se pueden usar en esta fase son:

- `Microsoft Threat Modeling Tool, MTM`: Es un *software* del 2016 que permite encontrar amenazas en la fase del diseño de aplicaciones. Se basa en la metodología *STRIDE* y puede ser utilizado en este contexto para crear perfiles de los atacantes con base en las amenazas a nivel empresarial.

- `VAST Modeling`: Es una plataforma que automatiza el modelado de amenzadas bajo la metodología `VAST` (*Visual, Agile and Simple Threat*). Trabaja con diagramas de flujos de procesos y su escalabilidad permite que sea aplicada a empresas de todo tamaño.

- `Trike`: Es un *framework* de auditoría que trata al modelo desde un punto de vista de manejo de riesgos y una perpectiva de defensa. Se define al sistema entendiendo a sus actores, activos, acciones y reglas y se crean matrices en donde las columnas representan activos y los renglones a los actores.

- `OCTAVE` (*Operationally Critical Threat, Asset, and Vulnerability Evaluation*): Fue creado con la ciberseguridad en mente por una `CERT` en 2003 y sus aspectos primordiales son los riegos operacionales, las prácticas de seguridad y la tecnología. Opera en tres fases: Creación de los perfiles de riesgo basados en activos, identificación de vulnerabilidades en la infraestructura y desarrollo de estrategias y planes de seguridad.

1.2.4. Análisis de vulnerabilidades

Tiene que ver con el proceso de escaneo y enumeración (extracción de nombres de usuarios, recursos de red y servicios de un sistema), y proviene del inglés *vulnerability analysis*. En esta fase se descubren las fallas en los sistemas y aplicaciones que podrían ser aprovechadas por atacantes reales y van desde un diseño inseguro en las aplicaciones hasta configuraciones indebidas de los equipos. Hay varias técnicas de escaneo y enumeración, por ejemplo:

- Extracción de nombres de usuario a partir de direcciones de correo electrónico.
- Uso de contraseñas de fábrica para tener acceso fácil a los datos.
- Ataques de fuerza bruta al servicio de directorios de Microsoft para enumerar a varios usuarios válidos.
- Adivinado de cadenas de consulta para el protocolo `SNMP` (*Simple Network Management Protocol*) para extraer nombres de usuarios.
- Obtenido de información de la topología de la red por medio de consultas al `DNS`.
- Enumeración de puertos de los sistemas. El escaneo de puertos es sin lugar a dudas de las técnicas más usadas para descubrir vulnerabilidades en los sistemas, pues se pueden dar una idea de qué servicios están en uso, qué usuarios son dueños de dichos servicios o bien si algunos servicios web requieren o no de autenticación. Dentro de esta área también hay técnicas que se siguen como los escaneos *ARP*, de *Vanilla TCP*, *TCP SYN*, *TCP FIN*, *TCP Reverse Ident Scan*, entre varios otros que pudieran permitir identificar puertos efímeros abiertos. Se pueden usar los *TCP Wrappers* para tratar de limitar la información que un atacante ve cuando hace un escaneo de los puertos o bien *PortSentry* para imponer un límite al número de consultas inválidas a los puertos. Por su parte, el atacante o el *pentester* podría auxiliarse de las siguientes herramientas:

  - `Nmap`: Es un escáner de puertos de código abierto para auditoría de redes que envía paquetes de IP para poder saber qué puertos están escuchando, qué servicios y versiones ofrecen los servidores e inclusive qué sistemas operativos y esquemas de *firewall* implementan.
  - `Angry IP Scanner`: Está pensado para un fácil y rápido uso de escaneo de puertos y direcciones lógicas IP que pueda ser usado en diversas plataformas.
  - `UnicornScan`: En este caso es un motor que encuentra correlaciones que fue diseñado por la comunidad de expertos en seguridad informática con el objetivo de que fuera escalable, exacto, flexible y eficiente. Permite ver las salidas de diferentes formas (incluso como bases de datos relacionales) y hace escaneos de `TCP` y `UDP` con banderas que se le pueden pasar.

    - `AutoScan`: Su ventaja es que no requiere de una configuración previa para poder ser usado. Permite hacer modelado de la topología de la red, escaneos de hilos, direcciones IP, escaneo de puertos, etcétera.

- *Fingerprinting* de la pila `TCP/IP`, que consiste en descubrir las *huellas* que dejan los sistemas operativos y otro tipo de *software* en los paquetes de red analizando valores del tiempo de vida de paquetes IP (`IP TTL Values`), tamaños de la vantana de varios protocolos, opciones de `TCP`, entre otros. Existen dos modalidades para esta estrategia: activa y pasiva. En la primera se envían datos al sistema para ver cómo responde, basándose en la premisa de que cada sistema operativo implementa la pila de `TCP` de forma distinta; es detectable. En el segundo tipo, se examina el tráfico de red con ayuda de un analizador de protocolos (*sniffer*, como el caso de `Wireshark`) en aras de determinar el sistema opeativo; es indetectable, pero menos exacto. Algunas herramientas muy usadas serían:

    - `Nmap`: Ya se mencionó para el escaneo de puertos, pero puede servir también para determinar el sistema operativo de un equipo.
    - `Network Miner`: Es una herramienta de análisis forense para equipos con *Windows* que reúne datos de los equipos de una red y no del tráfico que viaja por esta. Entre lo que puede aportarnos es información para detectar sistemas operativos.
    - `POf`: Se usa para identificar equipos remotos de forma pasiva, qué tan lejos está de nosotros y cuánto tiempo lleva funcionando. Permite identificar filtros de paquetes, dando pie a identificar los SOs.

- Análisis de amenazas, que es el proceso en el que se descubren y analizan las amenazas que podrían comprometer a un sistema informático. Entre las herramientas que automatizan este proceso se ecnuentran:

    - `eEYE Retina`: Escaner de vulnerabilidades que correlaciones y valida lo que logren encontrar otras herramientas como `Nmap` y `Nessus`.
    - `Qualys`: Es un servicio basado en la nube que proporciona visibilidad inmediata y global a aquellos puntos en los que el sistema podría ser vulnerable.

- `IMPACT`: En un conjunto de herramientas (*toolset*) para pruebas de penetración y explotación que hace eficiente el descubrimiento de vulnerabilidades.

1.2.5. Explotación

Quizás la fase de *exploiting* debería ser mejor traducida como el *obtendio y elevado del acceso*. Todas las labores previas de planeación, reconocimiento y análisis se conjuntan llegados a este punto, que es cuando el *pentester* logra hacerse con el control del equipo objetivo, que también podría ser una aplicación web o una red de computadoras. Puede ser llevado a cabo al acceder a una dirección `URL` que no estaba expuesta a simple vista, pero más en general incluye:

- La ejecución de *exploits* cuando se tiene la certeza de que el sistema tiene ciertas vulnerabilidades que permitirán explotar los riegos. Estos *exploits* son piezas de *software* que —diseñados por expertos e investigadores del área, de acceso público o bien vendidos de forma ilegal en el mercado negro— toman ventaja de las vulnerabilidades de un sistema para causar comportamiento anómalo. Se pueden tratar de *exploits* remotos o *exploits* locales dependiendo desde dónde pueden ser puestos en marcha. Ejemplos de vulnerabilidades que pueden ser explotadas con este tipo de *software* incluyen:
  - `Dirty COW` (*copy-on-write*) que permite ir escalando privilegios de forma remota o local en equipos con *Linux* debido a condiciones de carrera.
  - Vulnerabilidad de deserialización de `Java` en servidores *WebLogic*: Los objetos de `Java` pueden ser guardados en un archivo binario y más tarde recuperados, pero muchas veces no se pregunta por el origen del código antes de deserializarlo, permitiendo que el atacante ejecute así código arbitrario.
- Escalado de privilegios, que sirve para poder conseguir más credenciales administrativas y poder ejecutar más programas una vez ganado el acceso a la computadora de la víctima como un usuario normal. Los privilegios se podrían ganar horizontalmente (obteniendo recursos y datos de otros usuarios del mismo nivel) o bien verticalmente (obteniendo permisos pensados para los administradores del sistema).
- El pivoteo (*pivoting*), que consiste en atacar y poder controlar otras computadoras en la red al usar el equipo originalmente comprometido

como un punto que actúe de intermediario. Sirve por ejemplo cuando se tiene control de un equipo detrás de un *firewall* y se quiere controlar lo que hay dentro de la red.

En general, algunas herramientas usadas en esta fase son:

- `Metasploit`: Se trata de una plataforma creada en 2004 de código abierto para poder desarrollar, probar y usar *exploits*. Viene con cientos de *exploits* que pueden ser integrados unos con otros, aunque también existe una versión de paga.
- `w3af`: Es un *framework* popular para encontrar y explotar las vulnerabilidades ser aplicaciones web, además de ser fácil de usar y contar con muchos *plugins*.
- `Core Impact`: Es muy caro (al menos $30,000 dls.), pero es por muchos considerado como la mejor herramienta de ejecución de *exploits*. Cuneta con una enorme base de datos de *exploits* que está en constante actualización, además de permitir la creación de túneles cifrados para ejecutar posteriormente otros *exploits*.
- `sqlamp`: Automatiza el proceso de descubrimiento de vulnerabilidades para hacer inyecciones `SQL`, pudiendo así controlar servidores de bases de datos. También permite hacer análisis de huellas a la base de datos e incluso poder ganar control del sistema de archivos del servidor.
- `Canvas`: Desarrollado por *Immunity Sec*, cuenta con 370 *exploits* y es más barato que `Core Impact`. Incluye el código fuente completo y a veces también se le añaden *exploits* de día cero.

1.2.6. Post-explotación

Después de concluir el ataque, se tienen que llevar a cabo dos acciones importantes:

- Mantenimiento del acceso: Una vez comprometido el sistema actual, conviene mantener el control del mismo la mayor cantidad de tiempo posible y comenzar a habilitar todas las opciones que podrían conducir a atacar, escanear y explotar otros sistemas, o bien a seguir explotando el mismo sistema de forma cautelosa, para evitar ser descubierto. Los mecanismos para lograrlo, incluyen:

- Puertas traseras y *troyanos*: Sirven para ganar acceso a sistemas ya comprometidos fáctilmanete. En el caso de los *troyanos* es necesario instalar el *malware* de forma local y luego pueden llegar a tener permisos de acceso administrativos. Pueden servir también para obtener información sensible como lo son las contraseñas. Por el otro lado, las puertas traseras pueden subir archivos de interés de forma remota y trabajan en puertos bien conocidos para poder mezclar su actividad en el tráfico de red normal.
- Canales encubiertos: Se utilizan para mandar la información extraída dentro de una red por canales secretos de comunicación de tipo `VoIP`, `DNS`, `ICMP` y `HTTP`; la información suele ir cifrada.
- *Rootkits*: Es *software* malicioso especializado en esconderse y hacerse indistinguible dentro de un sistema comprometido para mantener el acceso al mismo. Son muy difíciles de identificar, incluso pudiendo seguir en el equipo cuando se ha reinstalado por completo el sistema operativo (*rootkits* de `BIOS`) y diseminando su código como si fuera parte propia del kernel del sistema operativo. Son generalmente instalados por algún *trojano*, pero ya por su parte se hacen pasar por código benigno que en realidad está escalando privilegios y espiando a los usuarios.
- Exfiltración de datos: Constituye la transferencia no permitida de datos de una computadora o sistema a otro medio, ya de de forma manual o automática y además puede darse a través de medios electrónicos o direcramente a medios físicos. Se busca obtener información personal (*PII, Personal Identifiable Information*) e información de salud (*PHI, Personal Health Information*), además de propiedad intelectual e información financiera. Las formas de lograr estos cometidos son muy variadas, pudiendo hacerlo con ayuda de protocolos de red como `FTP` (*File Transfer Protocol*) o bien al esconder la información en imágenes y otros archivos. También suelen preferirse los canales de comunicación más usados para mezclar la actividad con la actividad normal de la red.

Algunas herramientas que podrían ser usadas en este contexto son:

- `Trojan-Spy` y *Keyloggers*: Son troyanos que espían las acciones de los usuarios, haciendo por ejemplo capturas de pantalla y obteniendo listas de aplicaciones en ejecución. La información es luego transmitida a los criminales o a los *pentesters* en nuestro caso.

- `BlackPOS Malware`: Es un programa maliciosos que también espía y está diseñado para ser instalado en sistemas de puntos de venta, permitiendo obtener datos de tarjetas de crédito y débito.
- `SuckIT`, `Adore`, `TOrn` y `ARK`: Ejemplos de *rootkits* que pueden ser usados para crear puertas traseras y pasan muy indetectados en el sistema.

- Borrado de huellas: Se pretenden borrar todos los rastros que el atacante o *pentester* pudo haber dejado en las fases previas como parte mismo del proceso. con esto, se toman medidas en contra del equipo de respuestas y los forenses. Al equipo de respuestas se le pretende confundir al camuflar las fuentes de actividades maliciosas, esconder actividades en servidores ocupados y al ir abriendo puertas traseras de forma secreta, por ejemplo.

  Por su parte, a los forenses que tratan de ayudar en investigaciones legales posteriores al ir hallando evidencias legales que pudieran ser usadas en un juicio, se les trata de dar la vuelta al borrar, cambiar, esconder, manipular y destruir toda la información digital que dé pistas del ataque como pueden ser la alteración de los *logs* (registros, *log tampering*), de los *timestamps* para confundir cuándo fue realmente accesado y modificado por última vez un archivo o bien simplemente cambiando las extensiones de los archivos para que su búsqueda automatizada se complique más.

  Algunas herramientas comunes incluyen:

  - `SRM`: Es un programa de línea de comandos similar a `rm`, pero en una versión segura que lo que hace es sobreescribir los datos en los archivos destino antes de desenlazarlos. Así, es más complicado recuperarlos desde la línea de comandos.
  - `WIPE`: Su fin es desmagnetizar la superficie del disco duro para que sea prácticamente imposible recuperar la información que se almacenaba en este.
  - `Overwrite`: Se trata de uan herramienta de *Linux* que pretende complicar el proceso de recuperación de datos para un forense. Como su nombre lo indica, sobreescribe archivos con otros patrones que son en parte deterministas y lo demás aleatorio.
  - `DBAN`: Es el acrónimo para *Dark's Boot and Nuke* y se encarga de borrar por completo los discos duros de la mayoría de las

computadoras que puede detectar, lo que lo hace conveniente para escenarios en los que se tiene que borrar el disco por alguna emergencia.

- `Diskzapper Dangerous`: Fue pensado para las computadoras en las que no es fácil o conveniente conectar un monitor o un teclado. El punto es que los discos comienzan a borrarse automáticamente una vez que la máquina termina su proceso de encendido.

1.2.7. Reportado

En la fase de *reporting*, se enumeran y listan todos los hallazgos que pudo entrever el *pentester* luego de emular el ataque de forma que sea entendible, aceptable y útil para la organización. Estos reportes incluyen los defectos que permiten que los atacantes violen las políticas de seguridad y tengan algún impacto. Por supuesto, el tipo de reporte dependerá de para quién vaya dirigido. En general se identifican dos tipos de reportes:

- Reporte a nivel ejecutivo: Incluye aspectos como los impactos a nivel empresarial, modelos de madurez y delineado de estrategias, además de un apéndice con los tecnicismos empleados, que no deben ser muchos.
- Reporte técnico: Va dirigido a departamentos de seguridad dentro de la organización, así que incluye descripciones más detalladas de los descubrimientos técnicos, con capturas, pruebas de concepto, metodologías seguidas y todo lo necesario para que ellos puedan replicar los resultados.

Algunas herramientas que se pueden usar para auxiliarse en el generado del papeleo son:

- `Dradis`: Una plataforma de código abierto para crear reportes de forma colaborativa por expertos de seguridad. Es una herramienta que ya viene por defecto instalada en *Kali Linux*. Es posible incluir en ella fácilmente capturas de `Nesuss`, `Burp`, `Nikto`, `Owasp CAP`, etc.
- `Magic Tree`: Se trata de un sistema de manejo de datos y herramienta para la creación de reportes muy parecida a la anterior que fue diseñada para un fácil uso. También permite la ejecución de comandos externos e igualmente viene preinstalada en *Kali Linux*. A diferencia de `Dradis` que tiene una arquitectura cliente-sevidor, esta es una aplicación de escritorio, pero el código no es abierto.

- `Metagoofil`: Es una herramienta pensada para extraer metadatos de documentos públicos de la empresa y dar mucha información relevante de los documentos escaneados. Puede servir entonces para presentar fácilmente este información en los reportes finales, aunque también podría ser usada en etapas de reconocimiento previas de la metodología para las pruebas de penetración. Como en el caso de las dos anteriores, también está preinstalada en *Kali Linux*.

2. Tarea Moral Ataques (Casos reales)

2.1. Robo de identidad: mes y medio de pesadilla

Y en efecto fue toda una pesadilla para la pobre protagonista que dejó por un par de días su credencial de elector con los guardias privados de un edificio al asistir a una junta. Se describe cómo le robaron su identidad gracias a ese INE y en poco tiempo ya habían hecho abierto nuevas líneas de crédito en sus bancos, tramitado más servicios y la habían endeudado por miles de pesos. Lo de menos a no era tramitar una nueva identificación oficial, sino presentar denuncias y ampliar declaraciones ante distintas instancias gubernamentales que más que ayudarla parecía que le ponían el pie a propósito, casi ayudando a los delincuentes. Las trabas burocráticas fueron decenas y ella tuvo que luchar contra mar y tierra, tratando de evitar que con solo hacerse de su identificación los maleantes pudieran desconectar a todo el sistema mexicano montado sobre leyes sinsentido, normas y descuidos de bancos o las impensables acciones emprendidas por empresas que toman incluso las huellas dactilares equivocadas.

No se reporta nunca el nombre de la mujer a la que le pasó todo esto y en ese sentido podría ser el vivo reflejo de cualquiera de nosotros. Como moraleja, debemos resguardar bien nuestros datos que pueden ser explotados y servir de información valiosa para granujas como los que le hicieron pasar por tan mal sabor de boca. EL robo de identidad ocurre día con día en nuestro país y de acuerdo con datos oficiales de CONDUSEF, México ocupa el octavo lugar a nivel mundial con más casos, además de aumentar año con año. Dese mi perspectiva, las firmas personales son tomadas a menos en muchas ocasiones al hacer pequeñas compras y la gente muchas veces parece darle poca importancia a los efectos que puede tener el perder una carpeta con documentos valiosos o su cartera si es en una ventana de tiempo relativamente breve. No debería ser así, pues los malhechores saben cómo engañar al sistema fácilmente con los pocos recursos que se requieren y darle

la vuelta a cientos de leyes que parecen no servir en muchos casos. Ojalá se creen a raíz de estos tristes episodios medidas cada vez más concretas y contundentes que permitan identificar correctamente a los mexicanos y que los organismos públicos sean de más ayuda.

2.2. La extorsión de Marcela

La grabación consiste en un intento de extorsión a un adolescente de diecisiete años bajo la amenaza de tener secuestrada a su madre. El maleante se hizo pasar por el líder de la organización criminal del Cártel del Golfo (Los Zetas), Miguel Treviño Morales, que sería en realidad aprendido tan solo un mes después de que se subiera el video con el audio en un operativo de la Marina en Nuevo León. Al hacerse pasar por este capo de la droga, el señor pretende asustar al muchacho y conseguir algo de dinero, pero de entrada resulta desconcertante (incluso para un incauto) que un criminal de esa calaña ande lucrando ilegalmente por teléfono por algo más de quinientos pesos.

Luego de que la hermana del muchacho, Marcela de veinticinco años se una a la llamada, el tono cambia completamente y parece más una comedia en donde la chica comienza a burlarse del extorsionador. Ella se percata de inmediato que la historia que ha montado es una inmensa mentira y el malhechor comienza a inmiscuirse en más y más irregularidades en su fantasía, llegando inclusive a decir que está a doscientos metros de la casa de los muchachos y podría quemarla en cualquier momento y que a la par se encuentra en Tepic con la madre y otros hombres armados.

Considero que es un buen retrato de la realidad que se vivía hasta hace tiempo en donde la gente desprevenida caía ante semejantes historias y los bandidos se podían salir con la suya. Con el paso del tiempo y el acceso cada vez mayor a la información, cada vez es más fácil reconocer cuando estas personas de mala fé han montado un circo y colgarles de inmediato. Yo he platicado en varias ocasiones con mis abuelos, familiares y amigos: ha todos nos han tratado de extorsionar en algún momento, pero es muy fácil darse cuenta de las mentiras al hacerles unas cuantas preguntas a los bribones. En consiguiente, cada vez es más difícil que estos criminales lucren con llamadas de esta naturaleza (a pesar de lo que afirma al final el extorsionador), exigiéndoles más tiempo de preparación.

Creo que el tema tratado y el resultado de la llamada mal planeada del

extorsionador se relaciona con dos fases muy importantes de la metodología revisada en esta tarea: el *pre-engagement* (previo al ataque, contacto) y la fase de *intelligence gathering* (acumulación de inteligencia), pues una pobre indagación de las víctimas hará que el atacante o bien el que realiza la prueba de penetración para una empresa o compañía no sepa ni por dónde meterse o será descubierto de manera casi inmediata. Es importante de todas maneras seguir al pendiente de las modas y mañas de los extorsionadores telefónicos en un país tan inseguro como el nuestro y ser conscientes de que a su vez podrían evolucionar sus acciones para ser cada vez más cautelosos y reunir información por distintos medios, inclusive como lo harían los ingenieros sociales al ir engatusando a sus víctimas.

2.3. El jefe de *Hooters* se infiltra en su propia empresa

El video sigue de cerca a Coby G. Brooks, presidente y jefe de operacions (CEO) de Hooters, quien asumió cargos importantes a una corta edad luego de que su padre se lo concediera. Al principio platica que no fue fácil contraer tal responsabilidad, sobre todo tomando en cuenta el carácter del padre, quien había adquirido los derechos de todas las franquicias de lo que en un principio había surgido en 1983 en Clearwater, Florida, como el proyecto de tan solo seis personas que buscaban crear un espacio para que la gente pudiera relajarse, comer y ver partidos de americano. Actualmente la empresa genera cerca de mil millones de dólares anuales y tiene locales en más de veinticinco países, pero a Coby le preocupaba el hecho de que las ventas fueran cada vez más bajas y estaba buscando expandir su base de clientes.

Para lograrlo, acudió al programa de televisión y asumió una nueva identidad como Scotty Acher. Se hizo pasar como una persona en busca de trabajo que provenía de la industria de la construcción, seguido por cámaras que estaban documentando sus logros. Su objetivo era meterse de incógnito y trabajar directamente en las primeras líneas para entender las fallas del negocio y tratar de enmendarlas. Luego de hablar al respecto con el equipo ejecutivo, puso su plan en marcha y estuvo trabajando en diferentes locales de Texas.

En el primer trabajo estuvo haciendo labores de cocina —lo que no había hecho en casi veinte años—, mezclando alitas, moviendo contenedores de basura, sirviendo y cargando jarras, limpiando bandejas y, en general, siendo tratado como *carne de cañón*. En realidad a Brooks se le hicieron muy com-

plicadas estas labores y no obtuvo el puesto en esta sede de Dallas, pero le sirvió para entender que en el negocio se le estaba dando mucho énfasis a las meseras y poco a lo ocurre tras bambalinas en la preparación de los alimentos y proceso de limpieza.

Después se puso en contacto con otra sede de *Hooters*, en donde tuvo la oportunidad de participar en una promoción de alitas gratis en la calle. Ahí pudo escuchar en carne propia comentarios de varias personas en cuanto a cómo percibían el negocio y su filosofía. Muchas de ellas simplemente no iban porque consideraban que degrandan, explotan y cosifican a las mujeres, desde el propio logo de la compañía. Para él fue interesante escuchar cómo ve a *Hooters* otro sector de mercado distinto.

Luego se dirigió al que me pareció el peor lugar de todos. Estaba pidiendo el puesto de encargado, pero el encargado actual claramente no tenía idea de cómo tratar a las mujeres, comportándose como un verdadero misógino. Las ponía en fila para *inspeccionarlas* y las hacía participar en juegos degradantes. En el lugar prácticamente no había reglas y el encargado, Jimbo, se había pasado de la raya por donde se le viera. En contraparte, luego fue a otro *Hooters* en donde entendió la forma en que la encargada Marcee era mucho más comprensiva con sus subordinadas, pues había pasado antes ella por ahí y trabajaba muy duro para sacar adelante a sus hijas. Me recordó a la película del 2018, *Support the Girls*, que trata justamente de una encargada de un pequeño local de comida rápida al estilo de *Hooters* y es un poco una crítica malograda de estas cadenas.

Por último, Brooks fue a *Naturally Fresh* en donde, entre varias cosas, se hace la salsa para las alitas vendidas en *Hooters*. Tuvo la oportunidad de oir a los trabajadores, quienes se sentían abandonados por Brooks Jr., muchos de los cuales ni siquiera sabían nada de él. Eclipsado bajo la sombra de su padre, entendió que debía frecuentar más la planta para hacerlos sentir como parte de la familia y que vieran que seguían en buenas manos.

Al final del episodio Brooks les revela a las personas con las que estuvo conviviendo esa semana quién es él en realidad. A partir de lo que pudo entender decidió hacer algunas donaciones al ejército, darla vacaciones a Marcee y reprender a Jimbo —quien después renunciaría— por el mal trato que daba a las meseras. También decidió involucrar a un par de chicas más en una nueva campaña publicitaria para mejorar la imagen de *Hooters*, pero se me hizo extraño que nunca mencionara nada de aumentar el

suelo a los trabajadores de *Naturally Fresh* que tanto lo habían manifestado.

Desde mi punto de vista, este episodio es un claro ejemplo de cómo debemos indagar en lo más profundo para poder encontrar vulnerabilidades en todo sistema y poder disminuir el efecto de los riesgos en la medida de lo posible. Brooks se hizo pasar por un empleado de la misma manera en que un *pentester* se hace pasar por un atacante para que al final todo pueda mejorar.

3. Conclusiones

Hoy en día es imposible tener un sistema o infraestructura completamente segura y perfecta. Sin lugar a dudas este hecho puede convertirse en un mal sueño para las empresas y compañías que manejan grandes volúmenes de información sensible y activos, pero para su salvaguarda se han ideado metodologías claras que permiten simular ataques informáticos gracias a los conocimientos de un experto en el área: el *pentester*. Cada fase de esta metodología cíclica es igual de importante y no puede ser olvidada o dejada de lado: de su correcta conjunción dependerá en gran parte la identificación de las vulnerabilidades y riesgos informáticos que logren al final plasmarse en reportes para que puedan ser enmendados los errores.

Haciendo un símil con el video del *CEO* de *Hooters*, considero que hay que inmuscuirse y ensuciarse las manos, yendo a los más profundo del funcionamiento de una empresa para poder descubrir sus puntos débiles; esto es justamente lo que se hace en una prueba de penetración. No podemos proceder sin cautela como hizo la persona de la llamada telefónica de extorsión a Marcela: debemos planear cada detalle para que su ejecución sea lo mejor posible y no seamos detectados.

Finalmente, pudimos ver y entender una amplia variedad de herramientas que pueden ser usadas en cada una de las distintas fases, así que es fundamental que el *pentester* se mantenga siempre informado y a la vanguardia en sus conocimientos, además de estarse certificando continuamente para poder expresar su valía como se ha expresado varias veces en clase.

4. Referencias

Referencias

[1] El Financiero, Salvador Camarena. Mes y medio de pesadilla. http://www.elfinanciero.com.mx/opinion/salvador-camarena/mes-y-medio-de-pesadilla. Consultado el día: 16 de febrero de 2019.

[2] La extorsión de Marcela. https://www.youtube.com/watch?v=At5hSNoDEgk. Consultado el día: 16 de febrero de 2019.

[3] El Jefe se infiltra en su propia empresa, Programa 2 (Completo) en Español. https://www.youtube.com/watch?v=qaDe9Oqj8vg. Consultado el día: 16 de febrero de 2019.

[4] Condusef. https://www.condusef.gob.mx/Revista/index.php/usuario-inteligente/consejos-de-seguridad/563-robo-de-identidad. Consultado el día: 16 de febrero de 2019.

[5] OWASP. Penetration testing methodologies. https://www.owasp.org/index.php/Penetration_testing_methodologies. Consultado el día: 16 de febrero de 2019.

[6] Cybrary, Ryan. Summarizing The Five Phases of Penetration Testing. https://www.cybrary.it/2015/05/summarizing-the-five-phases-of-penetration-testing/. Consultado el día: 16 de febrero de 2019.

[7] Infosec Institute, Irfan Shakeel. Penetration Testing Methodologies and Standards. https://resources.infosecinstitute.com/penetration-testing-methodologies-and-standards/. Consultado el día: 16 de febrero de 2019.

[8] Infosec Institute, Venkat Reddy. Process: Pre-Engagement. https://resources.infosecinstitute.com/process-pre-engagement/. Consultado el día: 16 de febrero de 2019.

[9] Infosec Institute, Dimitar Kostadinov. Penetration Testing: Intelligence Gathering https://resources.infosecinstitute.com/penetration-testing-intelligence-gathering/. Consultado el día: 16 de febrero de 2019.

[10] Infosec Institute, Irfan Shakeel. Process: Scanning and Enumeration. https://resources.infosecinstitute.com/process-scanning-and-enumeration/. Consultado el día: 16 de febrero de 2019.

[11] Infosec Institute, Srinivas. Process: Gaining and Elevating Access. https://resources.infosecinstitute.com/process-scanning-and-enumeration/. Consultado el día: 16 de febrero de 2019.

[12] Infosec Institute, Dimitar Kostadinov. Penetration Testing: Maintaining Access. https://resources.infosecinstitute.com/penetration-testing-maintaining-access/. Consultado el día: 16 de febrero de 2019.

[13] Infosec Institute, Dimitar Kostadinov. Penetration Testing: Covering Tracks. https://resources.infosecinstitute.com/penetration-testing-covering-tracks/. Consultado el día: 16 de febrero de 2019.

[14] Pentest Standard, Vulnerability Analysis. http://www.pentest-standard.org/index.php/Vulnerability_Analysis. Consultado el día: 16 de febrero de 2019.

[15] SecTools, Network Security Tools. https://sectools.org/tag/sploits/. Consultado el día: 16 de febrero de 2019.

[16] Pentest Standard, Threat Modeling. http://www.pentest-standard.org/index.php/Threat_Modeling#High_level_threat_modeling_process. Consultado el día: 16 de febrero de 2019.

[17] Cernegie Mellon University, Software Engineering Institute, Nataliya Shvchenko. https://insights.sei.cmu.edu/sei_blog/2018/12/threat-modeling-12-available-methods.html. Consultado el día: 16 de febrero de 2019.

[18] Infosec Institute, Satyam Singh. https://resources.infosecinstitute.com/kali-reporting-tools/#gref. Consultado el día: 16 de febrero de 2019.

[19] Security Wizardry, Biltraser. https://www.securitywizardry.com/index.php/products/forensic-solutions/anti-forensic-tools.html. Consultado el día: 16 de febrero de 2019.