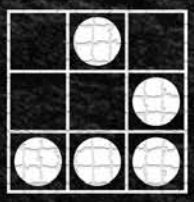
- Grado en ingeniería informática -
Trabajo final de grado: Seguridad informática

SEGURIDAD EN REDES Y SISTEMAS

Técnicas y conceptos sobre hacking y pentesting



Autor: Cristian Jiménez Jiménez
Consultora: Cristina Pérez Solà

Fecha: 5/6/2016

Dedicatoria y agradecimientos

Dedicar este trabajo a mi familia y amigos, en especial a mi madre, a mi pareja y a quien ya no está, por sus infinitas paciencias al aguantarme innumerables horas pegado al ordenador, y por darme el soporte moral necesario para seguir con mis estudios y mi pasión.

Agradecer también el soporte de Cristina Pérez en el proyecto, revisando y corrigiendo los conceptos que se describen a continuación. También a todos los profesores que me han ayudado a lo largo de todo el grado.

Gracias.

FICHA DEL TRABAJO

Título del trabajo:	*Seguridad en redes y sistemas: Técnicas y conceptos sobre hacking y pentesting.*
Nombre del autor:	*Cristian Jiménez Jiménez*
Nombre de la consultora:	*Cristina Pérez Solà*
Fecha de entrega:	*05/06/1016*
Área del Trabajo Final:	*Seguridad Informática*
Titulación:	*Grado en Ingeniería Informática*

Resumen del trabajo :

El siguiente documento trata sobre seguridad informática. Se detallarán técnicas y conceptos sobre seguridad en redes y sistemas, pentesting y auditorías de seguridad. Está destinado a lectores con un nivel de conocimientos informáticos de nivel medio, dado que se obvian muchas explicaciones básicas sobre protocolos, redes y configuraciones de sistemas. El documento se centrará en unas secciones principales (excluyendo las conclusiones, los glosarios y los anexos); introducción al hacking, gathering, análisis de vulnerabilidades, ingeniería social, malware, explotación de sistemas, explotación de redes, ataques a aplicaciones web y webservers y análisis forense.

El objetivo es tener una visión global de todo el mundo de la seguridad TIC detallando y diferenciando los modos de ver de los atacantes y los defensores de los sistemas. Para ello se ha dividido en diferentes secciones, correspondientes a los modos de actuar de los hackers malintencionados. A cada apartado, se ha añadido un apartado de herramientas utilizadas y recomendaciones para prevenir los ataques. Después se ha añadido un apartado de pruebas de concepto, correspondientes a cada uno de los apartados, para visualizar empíricamente que la teoría se puede llevar a la práctica si se tienen los conocimientos necesarios.

Abstract :

The following document is about computer security. Techniques and network security concepts and systems, pentesting and safety practices will be described. It is intended for readers with a middle level of computers, because many basic explanations about protocols, network and system configurations are obviated. The work will focus on a few main sections (excluding the conclusions, glossaries and annexes); introduction to hacking, gathering, analysis of vulnerabilities, social engineering, malware, exploit systems, networks attacks, web applications and webservers attacks and forensics analysis.

The objective is to have a global view of the computer security, detailing and differentiating ways to see systems attack; the defenders and the attackers. For it has been divided into different sections, corresponding to the modes of action of malicious hackers. For each section, has been added a section of recommendations and tools used to prevent and perform attacks. After all sections, has been added some proofs of concepts, corresponding to every sections, to show empirically that the theory can be put into practice if we have the sufficient knowledge.

Palabras clave (entre 4 y 8):

Seguridad informática, Hacker, Pentesting, Auditoría, Redes, Sistemas,

Índice

CAPÍTULO IV

CAPÍTULO V

CAPÍTULO VI

CAPÍTULO VII

CAPÍTULO VIII

CAPÍTULO IX

CAPÍTULO X

CAPÍTULO XI

CAPÍTULO I

1. Planificación general

1.1 Contexto y justificación del trabajo

Hoy en día, las nuevas tecnologías están al alcance de todo el mundo. Entregamos parte de nuestras vidas a un mundo informatizado, en forma de datos los cuales pueden ser robados y manipulados para el beneficio ajeno. Es por ello que las redes y sistemas informáticos deben estar asegurados y protegidos ante la amenaza del robo de la privacidad de los datos. El tema se acentúa cuando incorporamos al tablero de juego información sensible de las empresas cuyo valor puede ser incalculable y puede determinar el éxito o fracaso de ésta en contra de los beneficios de la competencia.

De dicho cambio de paradigma tecnológico, aparece la necesidad de realizar y mantener la seguridad informática mediante auditorías (o pentesting) e informes. Dichas auditorías son un conjunto de métodos, técnicas y estrategias necesarias para poner a prueba la robustez de un sistema o red con el fin de mejorar y arreglar posibles defectos que pueda tener .Pero la tarea no es simple, los sistemas y las redes son diversas y las artimañas prácticamente infinitas. Si añadimos la tremenda rapidez con la que la tecnología avanza, se intuye que los conocimientos necesarios para dichas auditorías son elevados. La ética de la persona que realice las pruebas y pueda acceder a los datos determinará el tipo de profesional o criminal que es, dada la responsabilidad que conlleva manejarlos. Comúnmente llamados hackers en términos generales, aparece disyuntivamente la figura del hacker ético como catalogación de la filosofía y moral de qué se va hacer con los datos y el objetivo del pentesting. Es por ello que surge la necesidad de estar permanentemente actualizado y la motivación extra de demostrar las técnicas actuales que se utilizan [2, 4].

Este documento está enfocado a la realización de pruebas en entornos controlados como máquinas virtuales o webs, sobre los principales puntos existentes de la seguridad informática, englobado en la filosofía y el método de los hackers éticos y malintencionados. Se darán por supuesto algunos conceptos por cuestiones de extensión del documento, mostrando pruebas de concepto de las herramientas junto con explicaciones empíricas.

1.2 Objetivos del trabajo

El trabajo tiene cuatro objetivos clave:

1. Comprender la figura del profesional de la seguridad informática, sobre todo las luces y sombras de la figura del hacker y hacker ético, junto con la labor del pentesting.
2. Tener una visión global avanzada de todos los puntos clave a tener en cuenta a la hora de realizar una auditoría de seguridad informática.
3. Demostrar empíricamente que los procesos teóricos y peligros sobre seguridad se pueden aplicar a casos concretos y con herramientas concretas.

1.3 Enfoque y método seguido

El método seguido será realizar una investigación concreta de cada campo (en libros, foros de internet especializados y webs) seguido de una descripción de los aspectos técnicos y de proceso más avanzados que puedan surgir. Se obviaran muchas explicaciones dado el carácter práctico y compacto que se quiere dar al documento.

Después se darán algunos nombres de las posibles herramientas y estrategias a seguir. Después de tomar una decisión sobre qué, con qué y cómo hacer la prueba, se realizará y se plasmará en el documento. Se tendrá configurada una red virtual con unos equipos a modo de laboratorio donde se harán las pruebas para evitar riesgos. Finalmente se darán algunos puntos clave del apartado (consejos, como mitigar esos ataques, etc.).

Resumido queda (por cada apartado):

1. Investigación y redacción de puntos clave, estrategia y herramientas.
2. Realización y plasmación empírica.
3. Consejos y mitigaciones.

1.4 Plan de trabajo

1.4.1 Estructura de la memoria y plan de trabajo

Diagrama de Grant:

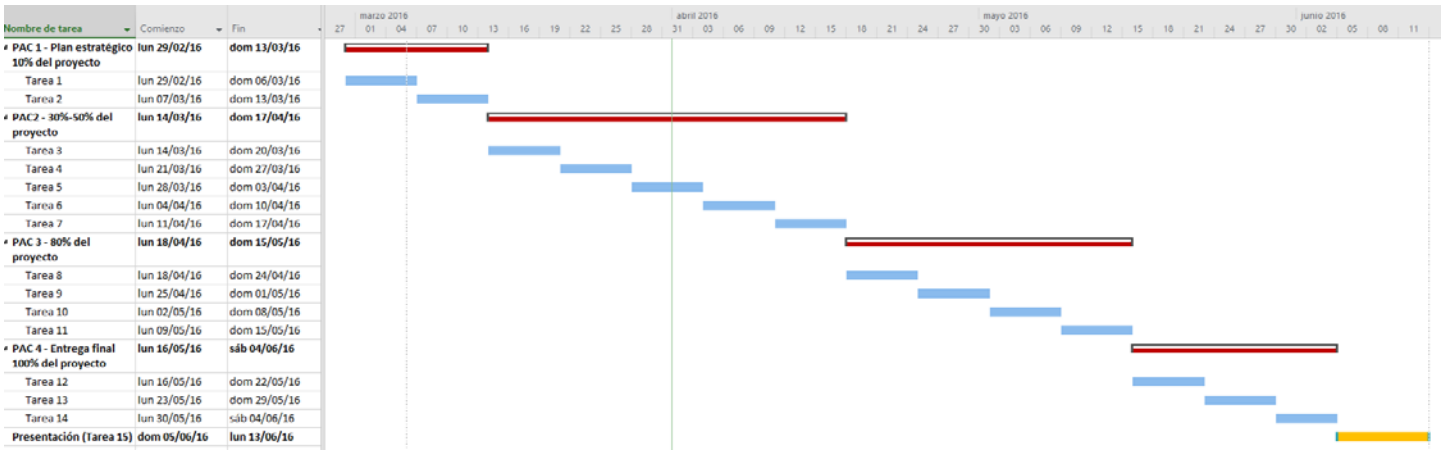


Figura 1.1: Diagrama de Grant

Plan de trabajo: tabla

Fecha	Días	Tarea	% de la memoria
Del 29/2/16 al 6/3/16	7 días	Tarea 1	PAC 1 (10%)
Del 7/3/16 al 13/3/16	7 días	Tarea 2	
Del 14/3/16 al 20/3/16	7 días	Tarea 3	PAC 2 (40%-50%)
Del 21/3/16 al 27/3/16	7 días	Tarea 4	
Del 28/3/16 al 3/4/16	7 días	Tarea 5	
Del 4/4/16 al 10/4/16	7 días	Tarea 6	
Del 11/4/16 al 17/4/16	7 días	Tarea 7	
Del 18/4/16 al 24/4/16	7 días	Tarea 8	PAC 3 (80%)
Del 25/4/16 al 1/5/16	7 días	Tarea 9	
Del 2/5/16 al 8/5/16	7 días	Tarea 10	
Del 9/5/16 al 15/5/16	7 días	Tarea 11	
Del 16/5/16 al 22/5/16	7 días	Tarea 12	PAC 4 --ENTREGA FINAL-- (100%)
Del 23/5/16 al 29/5/16	7 días	Tarea 13	
Del 30/5/15 al 4/6/16	6 días	Tarea 14	
Del 5/6/16 al 13/6/16	9 días	Tarea 15	PRESENTACIÓN

Figura 1.2: Plan de trabajo

1.4.2 Descripción de tareas

Aclaración importante: de la tarea 2 a la tarea 9 (ambas incluidas), hay que almacenar documentos, organizados por carpetas, para la elaboración del producto a partir de la tarea 11 (ver tabla2). Los términos susceptibles de incluirse en el glosario, se marcarán en rojo y en negrita para su posterior edición y producción en la tarea 12. Las referencias se marcaran en verde por el mismo motivo. También se generará un fichero aparte donde se almacenarán todos los documentos y fuentes que se vayan consultando.

Tarea 1

Proponer el tema a la consultora e intercambiar mensajes para concretar la puesta en marcha del proyecto. Realizar el plan de trabajo (tabla de tareas, diagrama de grant, índice, reparto de páginas, etc.). Redactar la parte de introducción (punto 1) propuesto por la asignatura (motivos, descripción del trabajo, etc.). Entregar la PAC1.

Tarea 2

Redactar el punto 2: Hacking y auditorías: aclarando conceptos. Ir preparando la red virtual (descarga de imágenes de S.O., configuraciones, etc.).

Tarea 3

1. Realizar el punto 3 y 5: Gathering y análisis de vulnerabilidades
    - Determinar los puntos y estrategias principales de recogida de información existentes.
    - Demostrar cómo se lleva a cabo la recogida de una empresa.
    - Consejos y estrategias para mitigar la fuga de información.
2. Terminar la red virtual.

Tarea 4

3. Realizar el punto 4: Ingeniería social
    - Determinar las principales estrategias y métodos usados por la ingeniería social.
    - Utilizar la técnica de phishing para demostrar un robo de credenciales.
    - Consejos para evitar los casos de robo de información.

Tarea 5

1. Realizar el punto 6: Malware
    - Determinar los principales tipos de malware existentes y como se crean.
    - Crear un malware e infectar un sistema de la red virtual para demostrar que efectos produce (probablemente será un troyano o un ransomware).
    - Como ocultar el archivo malicioso para que los antivirus no lo detecten.
    - Consejos y estrategias para evitar el malware.

Tarea 6

1. Realizar el punto 7: Sistemas y plataformas móviles.
    - Determinar los principales defectos de los sistemas móviles y estáticos.
    - Describir algunas herramientas y estrategia para la explotación de los sistemas.
    - Dos opciones: Utilizar metasploit en la máquina virtual y/o utilizar una rat en un dispositivo móvil para demostrar un ataque efectivo.
    - Realización de una prueba de ingeniería inversa sobre un archivo concreto real y una prueba sobre análisis forense en un sistema de la máquina virtual.
    - Consejos sobre políticas de seguridad.

Tarea 7

1. Realizar el punto 8: Redes.
    - Determinar las principales estrategias de ataques a redes IPv4 e IPv6
    - Realizar una intrusión en una red real (máquina virtual) y una craqueo de una red wifi WPA2.
    - Consejos sobre configuraciones idóneas y configuración de firewall.
2. Entregar PAC2

Tarea 8

1. Realizar el punto 9: Web aplications y servidores
    - Determinar los principales vectores de ataque y estrategias hacia las aplicaciones y servidores web.
    - Realizar una ataque real a una aplicación web (se prevé un laboratorio de pruebas online). Denegación de servicio, sql injection y como evadir IDS, firewalls y honeypots.
    - Consejos y buenas prácticas.

Tarea 9

1. Realizar el punto 10: Forense
    - Explicar las estrategias y herramientas forenses

Tarea 10

Semana de margen por posibles retrasos en la redacción de los puntos anteriores. Repaso general de la memoria. Si se va bien de tiempo y cabe en la memoria, redactar el punto 10.

Tarea 11

Semana de margen por posibles retrasos en la redacción de los puntos anteriores. Repaso general de la memoria. Si se va bien de tiempo seguir con el punto 10. Entregar PAC3.

Tarea 12

Redactar el glosario y la bibliografía y las conclusiones. Empezar a organizar el producto.

Tarea 13

Acabar de redactar/construir el producto y empezar a finiquitar el trabajo.

Tarea 14

Dar los últimos toques al trabajo final. Entregar PAC4. Si se va bien de tiempo, empezar la presentación virtual.

Tarea 15

Preparar y entregar la presentación virtual (máximo 20 transparencias).

1.4.3 Requisitos necesarios:

Virtualbox (máquina virtual base) sistemas operativos diversos como Windows 8.1, Windows 7, Linux mint 17.3, Kali Linux 2.0, Windows 10. Herramientas open source incluidas en kali Linux.

Dos pcs, uno base donde se alojara el laboratorio virtual (i7, 8Gb RAM con Linux Mint 17.3, Kali Linux 2.0 y Windows 7) y otros deos pc suplementarios para realizar pruebas de red (A4, 4gb RAM con Linux Mint 17.3 y Windows 8.1).

1.4.4 Tabla de características internas de cada punto

Apartado	Máquina virtual	Ejemplo práctico
2. Hacking y auditorías: aclarando conceptos	-	-
3.Gathering	No	Sí
4.Ingenieria social	No	Sí
5.Análisis de vulnerabilidades y enumeración	No	Sí
6.Malware	Sí	Sí
7. Sistemas y plataformas móviles	Sí	Sí
8.Redes	No	Sí
9.Web aplications y servidores	Sí	Si
10. Forense	No	No
		Total=86
Otras partes (portada, índice, anexo, bibliografía, glosario) 12 págs.		

Tabla 1.3: Características internas de cada punto

Estructura básica de cada punto (aproximación):

- 1 o 2 páginas de conceptos y teoría (solo la necesaria, no exhaustiva ya que se da por sabida)
- Un apartado de como mitigar los ataques descritos.
- Resto de páginas: pruebas de concepto descritas en la tabla anterior (ejemplos descriptivos o empíricos en la máquina virtual u online)

1.5 Estado del arte

Es bastante complejo medir y determinar el estado de arte en el que se encuentra el hacking y la labor del pentester. Bastante se ha escrito sobre el hacking y sobre el hacking ético. Desde su primera mención en *Hackers:Heroes of the Computer Revolution* (Steven Levy ,(1984)), un ensayo a partir del cual se han dado muchas definiciones y conceptos de los que pocos o nadie tiene la última palabra. Esto ha dado lugar a libros sobre técnicas hacking, de cómo hacer test de intrusión y cómo prevenirse de ataques a sistemas y redes.

La evolución de la tecnología ha sido (y es) excesivamente acelerada como para que exista algún conjunto de métodos y planes capitales. Es por ello que es necesaria una revisión constante de cada método en cuanto un 0-day[1] es revelado.

Actualmente existen varios certificados oficiales, respaldados y actualizados internacionalmente que ofrecen y enseñan los conocimientos necesarios generales. Es el caso de CEH (Certified Etichal Hacking) impartido por la escuela EC-Council que obliga a sus alumnos a revalidar su título cada 3 años. Sobre el pentesting se podrían enumerar bastantes libros, por ejemplo *Pentesting con kali* (*González,P.P., Sánchez, G.G., Soriano,C.JM. (2013)*), es uno de mis preferidos, donde se hace referencia a un sistema operativo linux muy utilizado para realizar auditorías, por las herramientas que integra de serie y el cual se utilizará en la elaboración de este trabajo.

En conjunto es una disciplina bastante desconocida y algunas veces desestructurada que engloba bastante confusión. Por lo tanto se intentará analizar e investigar las técnicas más actuales y las fuentes más innovadoras con el objetivo de ser lo más vanguardista posible.

[1]0Day: Vulnerabilidad explotada y utilizada por un pequeño grupo de gente o un individuo, que se da a conocer al resto del mundo.

1.6 Breve sumario de productos obtenidos

Los productos que se prevén serán:

- Scripts y sentencias de códigos utilizados para la realización de los ataques/pruebas.
- Documentos obtenidos derivados de las pruebas.

1.7 Breve descripción de los otros capítulos de la memoria

En la memoria se han considerados todos los aspectos técnicos de seguridad, dejando los aspectos prácticos para el producto. De igual modo, el esquema general será:

2. Introducción al hacking: se pretenderá explicar los conceptos fundamentales, la estructura del pentesting y las diferentes visiones según el auditor o el empresarial (normas, recolección, explotación, informes, revisión, auditoría interna, externa, web y forense, etc.).

3. Gatering: es la parte de la recolección de la información útil previa al ataque. Se explicarán métodos y sistemas para dicha recolección.

4. Ingeniería social: se describirán las técnicas más comunes utilizadas para la extracción de información mediante ingeniería social. Se utilizará el phishing para demostrar las técnicas.

5. Análisis de vulnerabilidades: se explicarán las técnicas y herramientas óptimas para analizar las vulnerabilidades de los sistemas.

6. Malware: se explicará todo el tipo de malware (troyanos, backdoors, rootkits, adware, virus, worms, botnet, ransomware, spyware, crypter.) incluyendo un apartado propio de la criptografía llamado esteganografía. Se creará un malware a medida y se infectará un sistema entre otras cosas.

7. Seguridad en sistemas y plataformas móviles: se quiere dar un repaso a las vulnerabilidades de los sistemas y se utilizará metasploit para aprovechar algún exploit de los sistemas de las máquinas virtuales.

8. Redes se explicarán las técnicas más utilizadas en redes inalámbricas y fijas (sniffing, hijacking, spoofing, etc). Se hará una ataque a una red real y un craqueo de una red wifi segura WPA2

9. Servidores y aplicaciones web. Apartado que tratará sobre el hackeo de webservers y aplicaciones web. La prueba será sobre denegación de servicios, sql injection y XSS.

10. Forense: se explicará que es la informática forense y algunes de las técnicas más usadas al ahora de analizar los datos.

CAPÍTULO II

2.0 Hacking y auditorías: aclarando conceptos.

Antes de comenzar a realizar pruebas de concepto y de entrar en materia en profundidad, se deben definir los conceptos que contextualizan el resto del documento y el mundo del hacking. Es por ello que seguidamente se incluyen un conjunto de puntos en los cuales se aclararán las reglas y las fichas del juego.

2.1 Personajes del juego.

Antes de nada, se van a explicar los diferentes tipos de figuras que nos podremos encontrar en el arte de la seguridad informática [1].

Mucho se ha escrito sobre cómo definir el termino hacker. El término aparece las primeras veces de los programadores del instituto tecnológico de Massachusetts (MIT), los cuales utilizaban arreglos y trucos para facilitarse las labores (los llamados hacks). También se dice que proviene del término anglosajón 'hack' que significa hachar señalando al típico golpe con el que se arreglaban los primeros equipos electrónicos.

Los cierto y comúnmente aceptado es que un hacker es una apasionado de la información y de la libertad del conocimiento, del saber el porqué y el cómo de las cosas. Son individuos inteligentes con amplios conocimientos y habilidades en informática que en muchos casos toman esta rama como hobby aprendiendo autodidactamente. Es por ello que el término se puede extrapolar a otros ámbitos y a otras artes que no son propiamente informáticas[2, 6].

Ahora bien, popularmente, se asocia el término a un pirata informático necesariamente malicioso y aunque, si bien las intenciones pueden ser negativas, existe un amplio abanico de matices. Hasta aquí lo que se sabe comúnmente y a partir de aquí se explicarán todos los tipos de sub-categorías que engloban al término hacker.

Dependiendo de las intenciones del hacker, se engloban en tres categorías [1]. Unos son los black hats, que se especializan en habilidades ofensivas, intrusivas, destructivas y maliciosas, priorizando el interés personal y el beneficio económico sin ningún tipo de escrúpulo. A éstos se les conoce más coloquialmente como crackers o de forma general, cyberdelincuentes. En el otro extremo tenemos a los white hats, individuos cuyas intenciones benévolas y defensivas son las de mejorar la seguridad y la privacidad de los sistemas y redes. En el mundo empresarial se les conoce como analistas de seguridad o coloquialmente como hackers éticos. En el término medio tenemos a los grey hats, individuos que no son especialmente maliciosos pero tampoco estrictamente éticos, que curiosean generalmente y se mueven en los dos lados de la ley. Pero las categorías no acaban aquí. Existen los suicidal hackers, individuos que intentan hacer caer una infraestructura crítica por una causa (no necesariamente buena) sin preocuparse de las consecuencias civiles o penales. Cuando el individuo prioriza ideales políticos se cataloga como hacktivista cuyos intereses a priori van en beneficio de la población (¿alguien dijo 'anonymous'?). Cuando aparecen figuras que quieren imponer su propia ley basada en la religión y en los radicalismos políticos, hablamos de cyberterroristas. A menudo estos dos últimos términos son utilizados por los políticos por su propio interés, como el caso de Edwar Snowden, catalogado de cyberterrorista y de traidor a la patria, y de hacktivista por haber revelado los casos de espionaje masivo de EEUU [17]. Tenemos también a los phreakers que son hackers especializados en los sistemas telefónicos .

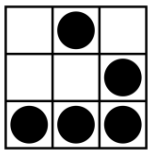


Figura 2.1: Glider

Pero aún hay algunas figuras más, menos conocidas, como los script kiddies, personas que no son consideradas hackers con conocimientos avanzados (más bien al revés) pero que aprovechan las herramientas y scripts de éstos sin comprenderlas para comprometer sistemas y redes. Una figura un tanto especial

es la del state sponsored, que son individuos contratados por un gobierno para hacer test de penetración[2] a infraestructuras de otros gobiernos para obtener información secreta. Anotar por último y como curiosidad, que a los hackers novatos se les apoda newbies y a los que presumen de tener conocimientos avanzados sin tenerlos se les cataloga como lammers.

En 2003, Eric S. Raymond, reputado hacker autor de libros como *¿Cómo llegar a ser hacker?* [2], propuso la idea de unión comunitaria y reconocible de toda la escena que se estaba formando. Para ello propuso el símbolo del planeador (Glider) del juego de la vida[3] (ver figura 1) como estandarte. Actualmente está comunmente reconocida como el símbolo de los hackers.

2.2 Fases y escenarios del hacking.

Una vez nos hemos posicionados respecto a las posibles figuras que pueden aparecer en el área de la seguridad informática, es conveniente definir las fases de intrusión que se pueden llevar a cabo.

Existe todo un conjunto de metodologías, protocolos y actitudes que determinan los ataques y los análisis [7].

Primero de todo, hay que diferenciar tres grandes categorías dentro del hacking a la seguridad empresarial. Por un lado tenemos las auditorías de seguridad, que son un conjunto de protocolos y métodos que chequea una organización para concluir si se están llevando a cabo los estándares de procedimientos y políticas de seguridad. Por otro lado, tenemos las evaluaciones de vulnerabilidades, que se focalizan en descubrir las vulnerabilidades en los sistemas de información pero no determinan si estas vulnerabilidades pueden ser explotadas ni de cuantos daños puede causar a la empresa en el caso de que ocurriera. Por último, tenemos el test de penetración (de aquí la palabra pentesting o pentest) que es un conjunto metódico y escrupuloso que engloba las dos otras categorías; auditorias de seguridad y evaluaciones de vulnerabilidades. También intenta demostrar si dichas flaquezas del sistema pueden ser explotadas y el alcance de daños que pueda tener en la empresa, junto con una propuesta de mitigación y/o solución técnica. En este documento nos situaremos siempre desde el punto de vista del pentester y del cracker para poder realizar las pruebas.

Hay que decir que estas pruebas se realizan siempre en entornos controlados que no ponen en peligro la integridad de la información ni de la infraestructura auditada. Además, no solo se necesitan conocimientos de informática avanzados para poder llevarlos a cabo, hacen falta actitudes como el sentido común, la perspicacia y la experiencia en conjunto con la metodología y orden adecuados.

2.2.1 Fases del hacking

En general, las fases de un test de intrusión, desde el punto de vista de un *white hat* son las siguientes [13]:

1. Definición de las normas y límites. En esta primera, el auditor debe ponerse de acuerdo con la empresa auditada para establecer los parámetros y límites del pentest. Estas normas legalizarán muchas de las prácticas que el auditor llevará a cabo y que de otra forma serían ilegales, como por ejemplo acceder a información personal o realizar ingeniería inversa o social al sistema y sus usuarios. Es muy importante detallar hasta donde se está dispuesto a llegar y

[2] Explicación detallada en el apartado 2.3 de éste mismo documento.
[3] Juego matemático de Martin Gardner equivalente a una máquina de Turing basado en dos reglas muy sencillas.

a qué objetivos se quiere llegar con la finalidad de salvaguardar la privacidad y no violar la ley en ningún caso. Son *las reglas del juego*.

2. Recolección de la información (conocido como **gathering**). Una vez establecida la primera fase, empieza labor de recogida de información. En esta fase, el auditor deberá reunir toda la información posible acerca de los sistemas y las redes, el software y el hardware, metadatos, configuraciones, versiones de sistemas operativos, información en internet, redes sociales, etc. En este punto se realizan técnicas de fingerprinting, footprinting, google hacking, etc. Es tremendamente importante ser escrupuloso en esta fase dado que determinará el éxito o fracaso del test de intrusión, pudiendo dar falsos positivos o conclusiones inválidas. Cuanto más se sepa en este punto, mejor se podrá realizar una estrategia adecuada para la empresa. Existen casos en que la empresa facilita información, pero eso lo explicaremos más adelante.

3. Análisis de las vulnerabilidades. Una vez se sabe todo lo posible de los sistemas y redes, se hace un análisis exhaustivo en busca de posibles vulnerabilidades o defectos susceptibles de ser explotados. Aquí entran en juego herramientas creadas para dichas tareas (que explicaremos en los siguientes puntos) junto con la experiencia y astucia del auditor. Es evidente que pueden existir infinidad de combinaciones válidas para llegar a vulnerar un sistema, pero dependerá de la combinación de dichos factores tener más o menos éxito.

4. Explotación de las vulnerabilidades. Esta es la fase más intrusiva y donde queda patente la realidad de las vulnerabilidades del sistema. Mediante herramientas como metasploit, se utilizan exploits para aprovechar las debilidades encontradas. El auditor deberá tener sumo cuidado al lanzar dichos exploits dado que es fácil perder el control de éstos y sobrepasar los límites descritos en la primera fase. También es común darse cuenta de que en la fase 2 y 3 quedaron puntos sin aclarar o sin descubrir. También, por ejemplo mediante técnicas de pivoteo[4], será común hacerse con el control de buena parte de la red, por lo tanto nunca se deberán perder de vista los documentos de la fase 1.

5. Generador de los informes. Por último, el pentester deberá realizar unos documentos que describan las conclusiones que todo el trabajo ha abocado. En cada fase anterior excepto la primera, se deberán ir documentando las acciones llevadas a cabo con tal de no dejarse nada en el tintero. Aquí se deberán incluir las medidas de mitigación que el auditor considere oportunas para intentar detener los posibles ataques descontrolados. También se incluirán los posibles efectos y alcances de los ataques en el caso de que se produjeran. Dado que los informes serán detallados y los receptores de éstos no deben ni tienen por qué tener conocimientos informáticos avanzados, estos se dividirán en dos: los informes técnicos y los informes ejecutivos. Los primeros contendrán los detalles técnicos y medidas precisas de mitigación de las vulnerabilidades encontradas, así como los resultados de las explotaciones de estas. En los segundos, el nivel de detalles será muy inferior, dando a entender claramente las deficiencias que ocurren así como las recomendaciones generales que el auditor propone.

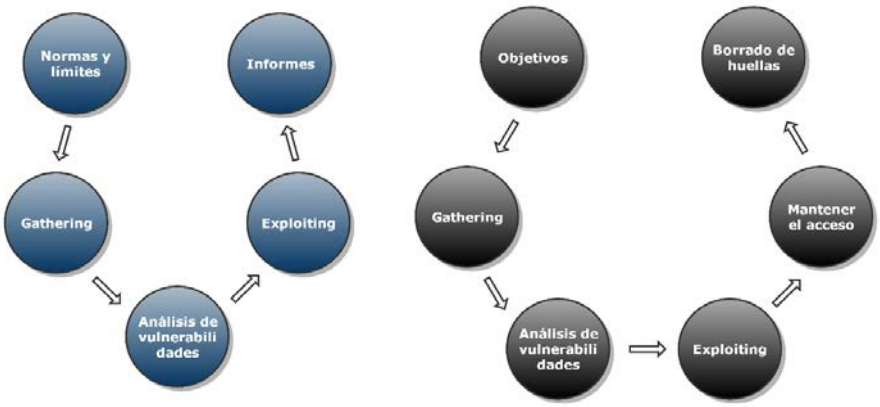


Figura 2.2: A la izquierda, proceso visto desde la perspectiva de un hacker ético, a la derecha, visto desde una black hat.

[4] Pivoteo: acción de pivotar, desde un dispositivo del que se ha obtenido acceso, a otro dispositivo de su misma red, infectándolo también y obteniendo acceso a él.

Nos podemos preguntar ahora, ¿un black o grey hat no realizará estos pasos no? Pues la verdad es que los realizará exceptuando la fase 1 y 5 (ver figura 2).

La fase de definición de las normas y límites pasará a ser la fase de objetivos. Aquí el hacker definirá que quiere obtener y hasta dónde quiere llegar. Por ejemplo, si lo que quiere es hacerse con una base de datos de un servicio web para luego venderla a terceros, no le interesará destrozar todo lo que encuentre, sino ser cuidadoso para obtener toda la información posible sin corromper la información.

Por otro lado, la fase de generación de informes, se elimina, y se divide en dos, permanencia del acceso y el borrado de huellas. En la primera el intruso deberá permanecer en el sistema/red todo el tiempo que le sea posible sin ser detectado. En esta fase instalará backdoors y creará usuarios con altos privilegios, entre otras acciones, para poder perpetuar su fin con éxito. En la segunda fase, el atacante se supone que ya ha realizado las acciones o extraído la información que le interesaba al sistema, con lo que lo único que le resta es eliminar toda pista de que ha estado allí. Esto incluye borrado de logs en servers y pcs, eliminación de usuarios ocultos, entre otras técnicas. En esta última fase es donde los crackers suelen cometer más fallos y donde se define la calidad del atacante.

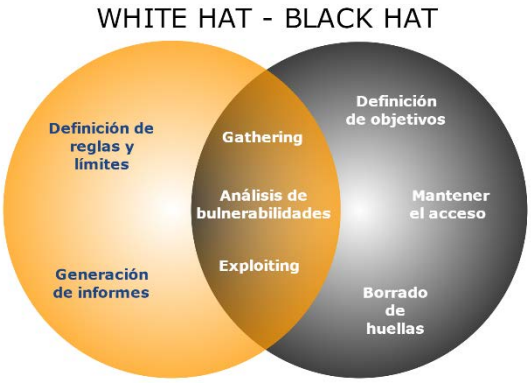


Figura 2.3: Relación de fases entre black hats y white hats

En este documento, debido a las limitaciones de tiempo y espacio, nos limitaremos principalmente a las fases 2, 3 y 4 de los dos puntos de vista ya que son comunes (ver figura 3). No obstante es posible hacer algunas menciones más a las dos últimas fases del punto de vista del cracker.

2.2.2 Modalidades de hacking

Dependiendo de la información de la que se dispone previamente a la fase del pentest, se definirá una de las tres modalidades siguientes [3]. Es importante esta clasificación dado que la información previa es directamente proporcional a la dificultad y al tiempo de inversión del proyecto, por lo que el coste del proyecto variará sustancialmente.

White box hacking: también denominado hacking transparente, es la modalidad en la que el auditor dispone de una amplia y extensa variedad de información proporcionada directamente por la empresa a auditar. Es la modalidad que menos recursos en costes acarreará y normalmente solo se aplica a pruebas de intrusión internas.

Black box hacking: solo se aplica a pruebas de intrusión externas, es decir, ataques desde fuera de la red. Además no se dispone de ningún dato proporcionado por la empresa que pueda suponer una cierta ventaja. Realmente es la modalidad más cercana al cracker y la más rigurosa de caras a un ataque real.

Gray box hacking: esta es la modalidad intermedia en la que el auditor dispone de algunas directrices básicas de los sistemas y de la red pero sin demasiados detalles. Por ejemplo estructura principal de la red (firewall, direcciones IP, etc.). Normalmente estas modalidades se aplican a auditorías internas dado que suele ocurrir que el auditor parte de la base de un dispositivo conectado a la red simulando un trabajador normal real.

2.3 Vectores de ataque: ofensiva.

Sabiendo ya los escenarios, los protagonistas y las posibles visiones que puede tener un pentester o cracker, toca saber los frentes por donde los ataques pueden abrir una brecha. A continuación se detalla una visión global de los principales vectores de ataques a sistemas susceptibles de ser auditados.

Principalmente tenemos **tres categorías de amenazas** [4] que abren todos los frentes siguientes (figura 3).

Amenazas en la red	Amenazas en los sistemas	Amenazas en las aplicaciones
Ataques a firewalls e IDS	Ataques de malware	**Errores en las configuraciones**
Ataques DoS y DDoS	Footprinting	**Buffer overflow**
Envenenamientos DNS y ARP	Ataques a passwords	**SQL injection**
Hijacking y MITM	Ataques físicos	**Ataques criptográficos**
Snooping, Sniffing y eavesdropping	Backdoors y ejecución de código arbitrario	**Validación de datos de entradas incorrectos**
Facilitación de gathering	Ataques DoS	**Control y gestión de errores y excepciones incorrectos**
Keys y passwords comprometidos	Accesos desautorizados y escalada de privilegios	**Ataques sobre autorizaciones y autentificaciones**
Ataques de ingeniería social (transversal)		

Figura 2.4: categorías de amenazas

En los próximos apartados se darán más detalles y PoC[5] sobre cada uno de estas amenazas.

2.4 Políticas de seguridad: defensa.

Por último y para redondear la visión global del hacking que necesitamos, se detallarán las principales categorías que engloban la defensa de las redes y sistemas. Normalmente existen cuatro tipos de políticas de seguridad dependiendo del grado de restricción al que se someten los trabajadores y sus sistemas [5].

Política promiscua: es una política en que los sistemas no tienen restricciones de acceso a los recursos del sistema. Una política nada recomendable.

Política permisiva: aquí existe un grado bajo de control limitando los casos más graves y evidentes de amenazas. Servicios potencialmente peligrosos son bloqueados y los sistemas deben ser actualizados con regularidad para que sean efectivos. Es una política de seguridad estándar que podríamos tener en nuestros ordenadores personales mediante nuestra configuración por defecto del sistema operativo.

Política prudente: es un grado de política de seguridad alto donde los recursos están limitados exclusivamente a las personas más imprescindibles. Todos los servicios están bloqueados excepto los imprescindibles que son accesible mediante login. Es la política de seguridad que deberían tener las empresas que trabajan con TI.

[5] PoC: acrónimo de Prove of Concept (Prueba de Concepto)

Política extrema: En este grado extremo de seguridad, la conexión a internet es nula o limitada a casos muy estrictos y cuidadosamente controlados. Puede darse en almacenes de datos donde se guardan backups masivos de grandes empresas.

Dicho esto, no nos podemos ni debemos olvidar de las políticas de seguridad física. De nada sirve establecer una política de seguridad prudente si el acceso a los sistemas es fácil físicamente. Peligros comunes como exposición al calor/frío, incendios, polvo, inundaciones, vandalismo y terrorismo, etc. pueden provocar accesos desautorizados, robo de información, sabotaje y daños irreversibles. Aquí vuelve a aparecer el peligro de la ingeniería social que no es exclusiva de los sistemas y redes.

Por todo ello se establecen una serie de recomendaciones sobre controles de seguridad física que establecerán en mayor parte las políticas de seguridad física. Conceptos como seguridad en puertas y ventanas, cámaras CCTV, alarmas, botones de pánico, muros, guardas de seguridad, controles biométricos y dactilares, etc. son la base de la seguridad física. También forman parte un área de recepción, un área exclusiva para el servidor que no incluyan puertos externos como USB, lectores de CD/DVD, etc., desactivación por defecto de aparatos que no se utilizan frecuentemente, personal de mantenimiento informático habitual, registro de entrada y salida del edificio, control de humedad y temperatura, etc. Nos podemos encontrar también con las áreas separadas y de difícil acceso. Por ejemplo, recientemente Microsoft ha estado haciendo pruebas sobre la introducción de contenedores en el océano que contienen servidores y backups para establecer nuevas políticas de seguridad físicas.

CAPÍTULO III

3.0 Gathering: recopilando información

A continuación se detallará el proceso de Gathering, que englobaremos junto con el análisis de vulnerabilidades. La recopilación de información es un proceso base desde el cual el atacante partirá para hacer todo el resto del trabajo. Es importante resaltar que una metodología deficiente en este punto dará lugar, más tarde, a falsos positivos que producirán un resultado poco eficaz.

La mayor parte de la recogida de información se lleva a cabo mediante técnicas de analizado nombradas footprinting, que básicamente consisten en acumular el máximo posible de información (por irrelevante que parezca) [7]. Después del reconocimiento de footprinting, el gather sabrá la política de seguridad de la organización, detalles como direcciones ip, DNS, SSH, el esquema de red, entre otras muchas cosas, que dará lugar a poder identificar vulnerabilidades adecuadas en cada caso. En la figura 3.1 se explícita una visión del tipo de información que nos podemos encontrar por categorías.

Información de sistemas	Información de redes	Información de la organización
Nombres y pass. de usuarios/grupos	DNS, IDN, IP's, etc.	Detalles de empleados
Tablas de erutación	Bloques de red	La web de la organización
Información SNMP	Webs privadas/publicas	Comentarios en código fuente
Banners del sistema	Servicios activos	Directorios de compañía
Tipo de acceso remoto a los sistemas	ACM y ACL's	Políticas de seguridad
Nombres de sistemas	Puntos VPN	Direcciones físicas
Información de banners	Números telefónicos	Detalles de localización
Versiones de sistemas operativos	Sistemas de autentificación	Links server-organización
Metadatos en general	Protocolos de red	Noticias y prensa
Software y sus versiones	Versiones de firmwares	Trasfondo de la organización

Figura 3.1 - Tipología de datos obtenidos en el proceso de Gathering.

El footprinting, en esta fase puede llevarse a cabo interna o externamente dependiendo de si el análisis se hace desde fuera o dentro de la red. A su vez, el footprinting externo (que es el que llevaremos a cabo) se puede categorizar en dos subcategorías, el pasivo y el activo. El active footprinting es más intrusivo dado que analiza directamente elementos de la infraestructura de la organización (cabeceras, puertos, servicios, etc.). Por otro lado, el passive footprinting se dedica a analizar información que ya está a disposición 'de cualquiera' que sepa donde ver y buscar (motores de búsqueda, foros, registros públicos, etc.). El footprinting interno lo analizaremos más adelante en este mismo documento. También una buena parte de la información se puede obtener a partir de la ingeniería social, sección que se detallará en el punto 4 de este documento [1].

3.1 Metodología y herramientas

Dentro de la categoría de active footprinting, existen diversas áreas que analizar. La mayoría de ellas se describen a continuación [13, 3].

Fingerprinting web

A la hora de recolectar datos referentes a una empresa, es importante analizar su web en busca de información que nos pueda resultar útil. Principalmente nos interesan averiguar que softwares y sistemas operativos que utilizan y sus versiones, servidor web, detalles CMS[6](plugins por ejemplo), subdirectorios, nombres de archivos, paths, nombres de bases de datos, plataforma de scripting, nombres de empleados, direcciones de email, etc. Para encontrar esta información, analizaremos las cabeceras de los paquetes, las cookies, el código fuente HTML.

Existen una infinidad de herramientas que nos proporcionarán unas u otras informaciones. Como ejemplo práctico, podemos utilizar sparta (que es una interfaz gráfica del versátil nmap) incluido en Kali 2.0 para realizar un escaneo de los servidores [1,7].

Para el análisis de las cabeceras HTML bastará con analizar el tráfico de peticiones de paquetes. Wireshark es una herramienta estupenda para esta tarea.

Si queremos analizar a fondo el código fuente de la web, basta con acceder a ella desde cualquier navegador e introducir 'inspeccionar elemento'. Podemos encontrar comentarios, lenguaje utilizado, etc. Desde el mismo navegador podemos abrir las cookies (si utiliza) que nos dirán datos referentes al softwares y plataformas de scripting utilizados.

Con herramientas como webdataextractor (conocidas como webspider) nos rastreará mails, nombres de usuarios, etc. Si hacemos un volcado de la web entera para trabajar con ella offline (técnica nombrada como *mirroring web*) accederemos a todo el material tipo fotos, videos, flash, etc. y podremos posteriormente analizarlo en busca de metadatos utilizando, por ejemplo surfOffline para el volcado de la web, y Foca para el análisis de metadatos [1,7].

Referente a los CMS, una herramienta idónea es WhatWeb, la cual nos analizará frameworks, librerías javascript, blogs, servidores web, etc. Para buscar información derivada de los plugins, podemos utilizar blindElephant que deduce a través de los plugins, versiones y vulnerabilidades de frameworks. Ver anexo I para más detalles sobre herramientas [1].

Se podría escribir un libro concreto detallando las cualidades de las herramientas y de este y del resto de subprocesos, pero eso excede los límites de este documento. Dependerá del investigador y del tiempo del que disponga encontrar más o menos información, aunque también depende de las políticas de seguridad de la empresa.

Escaneo SYN o Half-Open

Busca identificar servicios asociados al protocolo TCP y se basa en el envió de SYN (sin ACK) esperando el SYN+ACK. Como la comunicación está incompleta, si se recibe el SYN+ACK se determina el puerto como abierto, si se recibe un RST el puerto está cerrado y si no se recibe nada, el puerto se considera filtrado. Las conexiones incompletas quedan en cola (de los routers o firewalls por ejemplo) y se eliminan si no se completan, por lo que son ideales para no ser detectados.

Resoluciones DNS

A partir de los servidores DNS y de sus registros podemos obtener una buena cantidad de información sensible, como ips, nombres de usuarios, localizaciones, nombres reales del servidor, nombres de hosts (tanto IPv4 como IPv6), etc. Por ejemplo, con el registro LOC podemos saber la localización del dominio, el A o AAAA traduce los nombres de hosts a sus

[6] Content management system: gestores de contenido tipo framework utilizados para administrar webs, por ejemplo Wordpress o Joomla.

direcciones ips, el NS define la asociación que existe entre un nombre de dominio y los servidores de nombres del dominio citado. Es decir, es cuestión de saber qué dice cada registro y de saber cómo pedirle la información. Con herramientas como dnsmap, maltego, dnsenum o el mismo sparta podremos obtener dicha información [7]. Incluso existen herramientas online que hacen el trabajo [18]. Al final tendremos los verdaderos nombres de dominio, direcciones ips del host, servicios activos, localizaciones, registros de intercambio de correo, y un largo etc. que nos ayudarán a hacer un primer mapeado de la red objetivo.

Ya que se menciona Maltego, hay que decir que esta herramienta abarca mucho más que resoluciones DNS (similar a lo que le sucede a FOCA o Recon-ng) con lo que a partir de un simple dominio web, puede obtener gráficamente información de servidores, equipos, cuentas de correo, documentos de metadatos, información de redes sociales y un largo etc. Es decir, Maltego, FOCA y Recon-ng son herramientas transversales que nos ayudarán en diferentes áreas a la vez. Ver anexo I para más detalles sobre herramientas.

SMTP/Emails

De las cabeceras de los emails se puede extraer una sustanciosa cantidad de información. Pero ocurre que no sabemos a priori cual es el nombre real del servidor de correo, por lo que tendremos que utilizar las herramientas de consultas dns anteriores como dnsenum y nmap para averiguarlo. Una vez lo sepamos, podemos saber el esquema de cabecera de email y así obtener datos. La única cuestión nuevamente es saber que significa cada dato del paquete del header. Por ejemplo este contendrá el email del destinatario y del emisor, el sistema de autentificación usado (por ejemplo RSA-sha256), fechas de envío y recepción, ip del emisor y receptor, etc. Para realizar estas pruebas podemos utilizar la herramienta SWEAKS incluida en kali 2.0. Otras herramientas online también permiten hacer uso de esta técnica, como en [19] donde podremos además saber la geolocalización de una ip. Ver anexo I para más detalles sobre herramientas.

Redes

Es necesario recopilar toda la información posible de las redes antes de realizar los ataques. Es por ello que intentar analizar desde fuera la estructura de la red en la medida de lo posible nos dará ventaja para poder tomar mejores decisiones.

El protocolo ICMP utilizado para las comunicaciones de red, puede ser explotado por herramientas como Traceroute para detectar los saltos que hacen los paquetes y así averiguar si existen varios routers, cuantos hay o si va directo al host. Repitiendo las búsquedas con traceroute, se puede llegar a dibujar un diagrama bastante aproximado de la red. Ver anexo I para más detalles sobre herramientas

Banner grabbing, VoIP e IDS/IPS

Otra de las áreas que más nos puede proporcionar información es a través de los banners que ofrecen algún tipo de servicio. Por ejemplo con ncat y el comando adecuado se pueden obtener datos como servicios con su versiones y su base (por ejemplo SSH 2.0 basado en OpenSSH 5.1), versión de los servidores (por ejemplo Google front end v1.0), etc [1].

Para el análisis de las tecnologías VoIP tenemos la suite SIPVicious o la herramienta ACE VoIP la cual imita el comportamiento de un dispositivo de comunicación IP (como un teléfono IP), para descargar entradas de nombre, extensiones, etc. El resultado pueden ser un conjunto de datos sobre teléfonos IP de la organización objetivo, direcciones TFTP y MAC [1].

Sobre IDS/IPS tenemos Fragroute/Fragrouter incluida en Kali 2.0 que nos pueden ayudar a evadir técnicas de detección de fingerprinting pasivo de los sistemas operativos, o la característica ‘state full’ de algunos firewalls. También en esta área tenemos WAFW00F que permite detectar firewalls web [7].

Por otro lado, dentro de la categoría de passive footprinting tenemos los siguientes campos.

Hacking con buscadores.

Comenzaremos con los más utilizados, Google y Bing. Existen técnicas bastante experimentadas para extraer información muy particular de los resultados a priori genéricos de Google y Bing. Detallando el uso de **parámetros (dorks) en la búsqueda**, podemos acotar los resultados. A continuación se muestra en la figura 3.2 un resumen de los parámetros principales a utilizar.

cache:	Páginas web guardadas en la caché de Google
realted:	Páginas web relacionadas con la web especificada
link:	Links a la dirección especificada
site:	Resultados exclusivos del dominio indicado
info:	Busca información secundaria sobre la frase introducida
intitle/allintitle:	Devuelve documentos con el título especifico o las palabras exactas y completas
inurl/allinurl	Del mismo modo que el anterior pero referente a las urls
filetype/ext:	Devuelve archivos con la extensión indicada
ip:	Busca dentro de la ip indicada (sólo bing)
contains:	Busca páginas con links a archivos con determinada extensión
intext:	Busca texto dentro de la web completa.
define:	Busca resultados donde se responde a lo indicado

Figura 3.2: Parámetros más utilizados en google/bing hacking.

Por supuesto debemos recordar que tenemos también las búsquedas avanzadas de los buscadores donde podremos extraer información de clientes, socios, afiliaciones, etc. de la compañía a auditar. También existen bases de datos online donde se recopila información utilizando el google hacking/ bing hacking. Es el caso de www.exploit-db.com y www.hackersforcharity.org.

Con herramientas como Netcraft, podremos determinar los SSOO utilizados y webs relacionadas con la empresa, restringidas al público. También recordar el uso de google maps, google earth, wikimapia, yahoo maps, bing maps, etc. para geolocalizar la empresa.

Pero por supuesto, hay vida más allá de los buscadores habituales. Es el caso de Shodan, Robtex y Yandex. Shodan es un buscador de dispositivos y aparatos conectados a internet. Se utilizan varios filtros y podremos encontrar routers, servidores o cosas más exóticas como semáforos, cámaras de seguridad, gasolineras o sistemas de calefacción. También admite el uso de parámetros al igual que Google o bing, por ejemplo: city, country, geo, hostname, net, os, port etc. (el nombre del parámetro ya es suficientemente descriptivo).

Otro buscador poderoso es Robtex. Éste puede realizar de manera pasiva, gathering considerado activo, ya que en vez de realizar queries directamente a los dominios y servidores DNS, lo hace mediante una consulta simple y así evitamos dejar rastro en dichos servidores. También existe Yandex que es un buscador ruso, el cual nos puede proporcionar unos resultados alternativos a los comunes.

También recordar la técnica de crear alertas en los buscadores para monitorear las actualizaciones de webs o usuarios en redes. Por ejemplo en Google Alerts o Twitter Alerts. Ver anexo I para más detalles sobre herramientas.

Protocolo WHOIS

El protocolo Whois está basado en el protocolo TCP y realiza consultas a una base de datos intentando obtener nombres de propietarios de dominios, fechas de caducidad de estos, datos de contacto del propietario, cuando se creó el dominio, ip públicas, etc. Actualmente se apoya en la base de datos ARIN y realmente es una herramienta muy potente y muy utilizada para la

obtención de datos. Simplemente con realizar la búsqueda en su web, obtendremos los resultados visibles del protocolo. Ver anexo I para más detalles sobre herramientas.

Redes Sociales

Por último, hay que mencionar la inmensa fuente de información que se puede obtener de las diversas redes sociales. Se tiende a proteger cuentas como Facebook y twitter dado que son las más usadas dejando vulnerable información en otras como linkedin, google+ o youtube [20]. Los usuarios mantienen sus perfiles activos y actualizados (u olvidados con datos sensibles) donde conectan con amigos, familiares, lugares de vacaciones, trabajos, sitios a los que suelen ir, gustos, videos personales, currículos, lugares donde ha trabajado, habilidades, comentarios, emails, lugar de residencia, etc. y todo esto es susceptible de ser analizado en beneficio ajeno. Por otro lado, las compañías como tales también tienen sus perfiles donde realizan encuestas, promocionan productos, dan soporte a clientes, ofrecen ofertas de trabajo, etc. y esto puede dar lugar a que un black/white hat pueda deducir estrategias de negocio, perfiles de empleados y productos, plataformas y tecnologías utilizadas, etc. Mediante perfiles de usuario falsos se podrán aplicar técnicas de ingeniería social que explotarán todos estos recursos. Este último punto se explicará más detalladamente en el punto 4 de éste mismo documento.

3.2 Visión resumen

Para tener una visión general del proceso dentro del pentesting, se ha creado un gráfico que visualiza y ordena en una sola pantalla todos los pasos que se deben llevar a cabo (ver figura 3.3).

3.3 Contramedidas

Para evitar todas estas amenazas y fugas de información, se deben tomar unas medidas preventivas que las minimizarán [1]. Aun así, el factor humano es muy importante y nunca se podrá mitigar al 100% estas clases de fugas.

- Endurecer las políticas de seguridad y perfiles de usuario, para que los empleados no puedan acceder a toda la información y ésta no pueda ser revelada.
- Dividir los servidores DNS a internos y externos, para que proteger los internos de las transferencias de información DNS.
- Eliminar los servicios que no se usen y privatizar servicios en las bases de datos de Whois.
- Establecer una metodología para el borrado de metadatos estricta.
- Encriptar los datos y los passwords, así como realizar copias de seguridad periódicamente, en cuanto a información sensible se refiere.
- Educar y enseñar a los empleados a utilizar seudónimos en los grupos, foros y blogs, y a ser conscientes de la importancia de la seguridad de sus perfiles privados en redes sociales.
- Controlar la información que se emite en las webs y los servidores de la empresa, utilizando las técnicas de footprinting periódicamente para eliminar las fugas de información.
- Seguir la normativa sobre las recomendaciones estándares ISO 27000 [21].
- Utilizar servicios de registro anónimo de dominios para evitar el hacking en buscadores y la explotación del protocolo whois

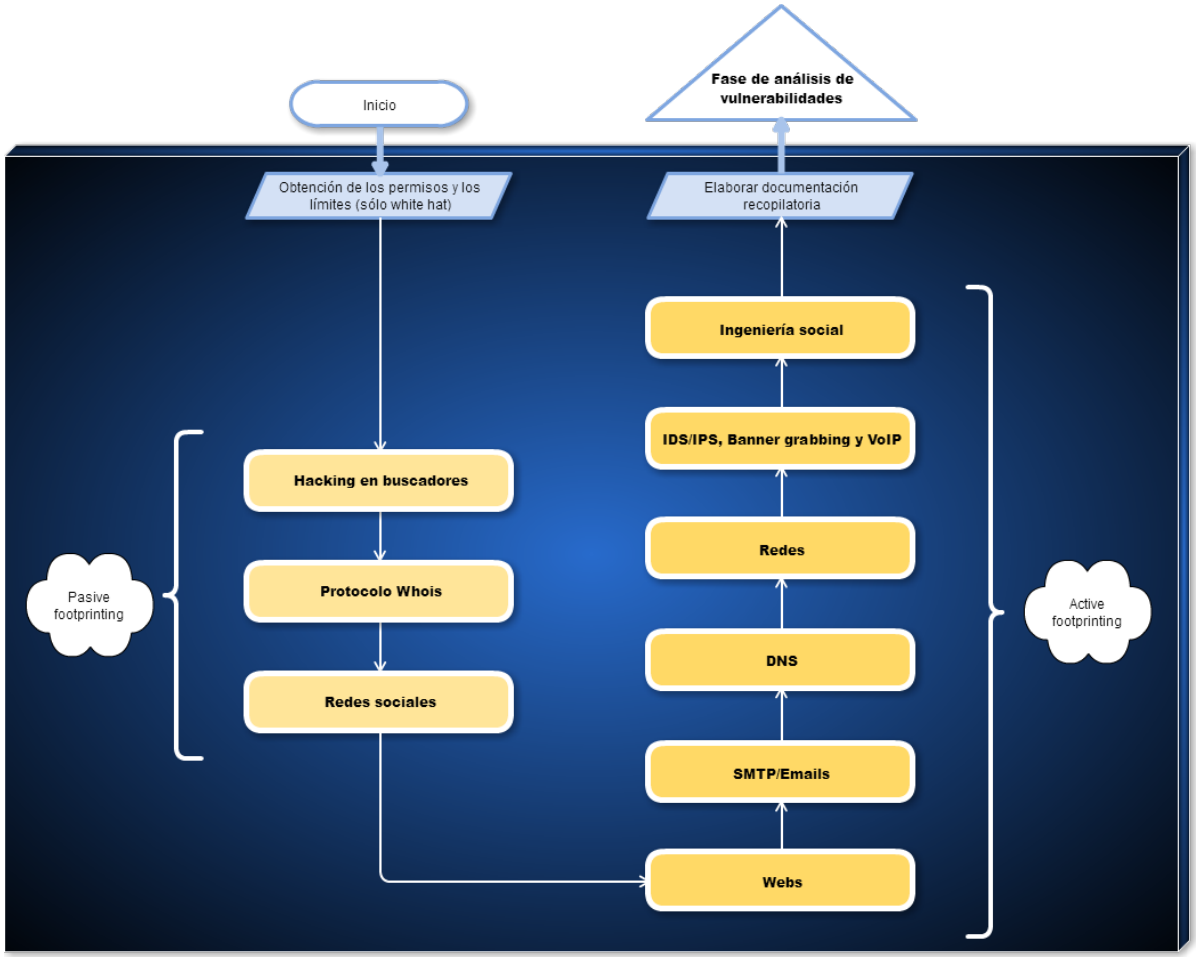


Figura 3.3 – Propuesta de camino a seguir en la elaboración del gathering.

CAPÍTULO IV

4. Ingeniería social

Una parte importante del gathering es la ingeniería social. Ya hemos visto que en los límites del haking ético no se explotarán la mayoría de las técnicas relacionadas con esta disciplina pero es importante identificar los vectores susceptibles de ataques por hackers maliciosos. Se basa en utilizar/engañar/manipular al considerado eslabón más débil de la cadena de información, la persona que trabaja en la organización. Aquí se hablará exclusivamente de la ingeniería social relacionada con la seguridad informática analizando las técnicas que llevan a cabo los black hat para explotar las debilidades o ingenuidades de aquellos que tiene acceso a la información legítimamente [1].

4.1 Técnicas

Para categorizar las aplicaciones y ataques, podemos identificar 3 tipos básicos de categorías en los cuales se basan los ataques: basados en personas, en sus dispositivos o en aplicaciones [1].

La primera categoría mencionada como ataques basados en personas es probablemente la más amplia y pícara, y consiste directamente en engañar o manipular personalmente (por ejemplo vía telefónica o en persona) mediante una serie de técnicas. Algunas de las técnicas siguientes no necesariamente deben ser catalogadas como de ingeniería social por si solas, pero se describen para mostrar que pueden ser aprovechadas con fines propios de ingeniería social, como robo de información, espionaje, etc. [1].

Rendering

Esta técnica consiste en hacerse pasar por alguien que pueda acceder a algún tipo de información de la empresa, ya sea vía telefónica o personalmente. Estos papeles representativos pueden ser los de un usuario convencional, un usuario que normalmente tiene privilegios o un empleado del departamento técnico. Haciendo creer al trabajador que se está hablando con el que dice ser, se puede extraer información sensible de todo tipo (nombres de usuarios y contraseñas, tipologías de red, software utilizado, etc.) de una forma relativamente fácil. Dicho esto existen una infinidad de escenarios posibles y dependerá de cada objetivo y de la picaresca del que realiza la acción obtener más o menos información.

Ingeniería social en redes sociales

Hoy en día las redes sociales contienen mucha más información de las que muchos propietarios de cada perfil son conscientes. Si estos perfiles no tienen las medidas de seguridad adecuadamente configuradas, o existen brechas de seguridad en los portales web, un atacante puede reunir gran cantidad de información solo realizando unas simples búsquedas. Además, estos detalles pueden llegar a ser de lo más íntimo como lugar de vacaciones, relación sentimental, círculo de amistades, lugar de trabajo, lugar de residencia, gustos musicales, de cine y de tv, ideología política y religiosa, comentarios en páginas y fotos y un largo etc. que todos conocemos. Si combinamos este abanico de información con el siguiente punto (robo de identidad), la extracción de información puede ser completamente comprometedora y determinante en el proceso de gathering.

Robo de identidad

El robo de identidad consiste en recolectar información de una víctima (redes sociales como likedin, Facebook, datos del propietario, etc.) para luego crear perfiles falsos con datos reales del objetivo, que den lugar por ejemplo a conversaciones con terceros, haciéndose pasar por ellos. Pero estos ataques no acaban aquí. Existen numerosos casos en los que el atacante ha obtenido tarjetas de crédito, dirección habitual de vivienda, etc. que han dado lugar a deudas

procedentes de compras por internet o préstamos. A la hora de hacer reclamaciones, es muy difícil probar que las acciones no las ha hecho la auténtica persona, dado que todos los datos son reales. Dicho de otro modo, puede ocurrir que un robo de identidad se perpetúe en el tiempo indefinidamente dado que, no cambiamos de lugar de residencia habitualmente, ni de familia o empresa.

Eavesdropping

Consiste en espiar deliberadamente las conversaciones o comunicaciones de un objetivo con el objetivo de extraer información sin que éste se entere. Normalmente también se intenta interceptar comunicaciones derivadas como videos, audios y todo tipo de escritos que puedan resultar útiles. Mediante la instalación de backdoors o técnicas de sniffing de redes como 'man in the middle' se puede llegar a esta acometida.

Dumpster diving

Esta técnica consiste en inspeccionar la basura y papeleras de los trabajadores en busca de documentos que no hayan sido adecuadamente eliminados. Si se tiene acceso a estos desechos, se pueden obtener datos financieros, facturas, extractos bancarios, notas en postits, impresiones desechadas, facturas telefónicas o cualquier otra cosa.

Ingeniería social reversa

En esta ocasión se trata de realizar las acciones de manera pasiva, al contrario que todas las otras técnicas. Dicho de otra manera, el atacante pone una trampa al objetivo y espera a que éste pique el anzuelo. Por ejemplo, un atacante deja su falsa tarjeta de técnico en reparación de ordenadores en el buzón de una empresa y espera a que ésta le llame. En ese momento el atacante podrá obtener gran cantidad de información sensible. No parece muy efectiva pero si se incrementan los objetivos masivamente las probabilidades crecen.

Piggybacking

Consiste en manipular a un responsable de seguridad física o de paso (por ejemplo a una recepcionista) para que deje pasar al atacante a algún área restringida, alegando escusas como olvido de tarjeta identificativa, ser nuevo en la empresa, etc.

Tailgating

Consiste en obtener el mismo objetivo que el piggybacking pero en este caso, falsificando algún tipo de documento identificativo que permita el acceso.

Shoulder surfing

Esta técnica más clásica, consiste en espiar al objetivo en momentos de descuido para obtener información, por ejemplo pasando cerca de su zona de trabajo, dejar un dispositivo de grabación escondido, etc.

La segunda categoría mencionada como ataques basados en dispositivos principalmente los computadores utilizados por los trabajadores. El objetivo es utilizar herramientas propias de los ordenadores para obtener información útil. En general se utilizan cuatro vectores de ataque: la infección con malware del dispositivo, el envío de emails maliciosos, el uso de direcciones web con segundas intenciones como banners y pop-ups y el chat directo. El primer punto se detallará en el punto 6 de este mismo documento ampliamente. El tema de los banners y pop-ups son pequeños fragmentos de código en javascript, HTML o flash incrustados en las webs los cuales la mayoría de veces intentan que la víctima haga clic en los enlaces para luego redireccionarlo a otra web donde se pretenderá guardar los datos que se le pidan. También puede ocurrir que pregunte al objetivo que acepte descargarse un pequeño programa aparentemente inofensivo que una vez instalado en el sistema operativo, se dedicará a recopilar y enviar información alojada en el computador, o simplemente abrir anuncios publicitarios en tus navegadores indiscriminadamente, de webs que buscan el robo de información. Conseguir realizar un chat directo con algún empleado, combinado con el robo de

identidad anteriormente descrito en redes sociales, puede llegar a revelar información por razones obvias.

En cuanto a los emails, existen de varios tipos. Los emails calificados como spam, buscan lo mismo que los banners y pop-ups; redirección a otra web publicitaria o con interés de gathering masivo para proyectos de big data por ejemplo. Estos emails pueden ser dirigidos a un grupo de usuarios objetivos (por ejemplo, el departamento contable de una empresa) o un directivo, haciéndolos más creíbles. En muchos casos se ofrece algún producto específico o muy atractivo para el receptor con el fin de tenderle una trampa.

Phishing

Es la técnica por excelencia de robo de información mediante emails y links, y consiste en realizar una copia de una página web real y que reclame credenciales (por ejemplo, una página de un banco o webs de redes sociales). Una vez copiada la estructura básica de la web original, se aloja en un servidor y se envía un email dirigido a la persona víctima. El email debe ser también una copia de los que la empresa cebo envía a sus clientes para que el ataque sea creíble. Una vez hecho esto, la víctima recibe un email con un link a la web falsificada pidiéndole que ingrese los datos oportunos (tarjeta de crédito, pin, nombres de usuarios, passwords, dirección, etc.) que son guardados por la web y por lo tanto, quedan en posesión del atacante. Es uno de los ataques más efectivos y suelen tener gran repercusión, dado que muchas veces se envían a grandes colectivos de usuarios objetivos, aumentando las probabilidades de obtener información. Un caso reciente es el de *ashleymadison* o el de robo de credenciales de Apple de cuentas de famosas. En ambos casos quedaron expuestos datos, conversaciones, fotos y videos de sus usuarios, que fueron publicados Tor [20].

La tercera categoría mencionada es la de ataques a aplicaciones, mayormente móviles y alojadas en nuestros smartphones o tabletas. El objetivo es el mismo que en los anteriores casos pero cambiando de medio. El modus operandi que los banners y malware tiene en las víctimas, se traslada ahora a aplicaciones de las stores de Android y Apple por ejemplo. El atacante crea una aplicación maliciosa (normalmente es una aplicación ya existente, reensamblada con el malware) y la sube a la tienda pública (por ejemplo AppleStore) con el objetivo de que la víctima o conjunto de víctimas se descarguen la aplicación y empiece el proceso de recolección de información del dispositivo. Después dicha información interna se envía al atacante, y/o se reclama la información directa a la víctima para el mismo fin. Se podría pensar que las aplicaciones que se suben a las tiendas deberían estar controladas íntegramente por la empresa responsable del dispositivo/store pero esto no es completamente así, dada la inabarcable cantidad de aplicaciones que se suben/bajan al día a nivel mundial (aunque sí se hace en gran parte). Además existen otras stores alternativas (legales e ilegales) muy populares donde se ofrecen software, en ocasiones pirata (como BlackMArt o Aptoide), donde proliferan a sus anchas estas aplicaciones maliciosas.

Por otro lado, también hay que mencionar los ataques dirigidos a víctimas vía SMS. En estos mensajes se adjunta un link advirtiendo algo (similar al phishing) y lo que nos provoca es una

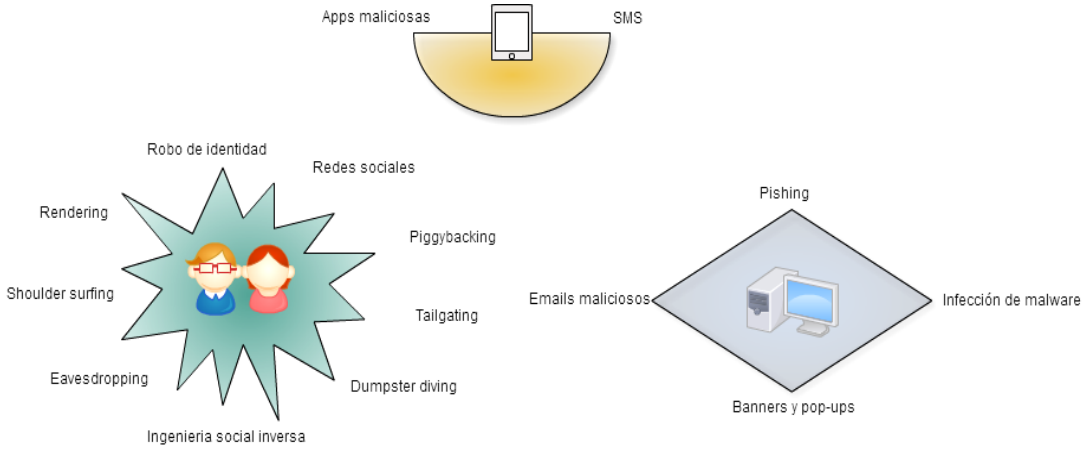


Figura 4.1 – Frentes abiertos en ingeniería social

infección de malware en el dispositivo que recopilará y enviará toda la información para la que esté programado. Esto se explicará más detalladamente en el punto 7 de este documento sobre exploiting. En la figura 4.1 se muestra un resumen de los frentes abiertos en ingeniería social. Recalcar una vez más que muchas de estas técnicas son consideradas como robo de información y están tipificadas como delitos penales en la mayoría de países.

4.2 Herramientas

Dada la parte creativa y efímera de esta parte del gathering, no existen un gran número de herramientas apropiadas para beneficiar la ingeniería social. A parte de software típico como el de grabador de video y voz, merece la pena comentar algunos:

Metasploit: esta herramienta todo terreno puede ayudar a crear malware para infectar un dispositivo móvil mediante un SMS o un mensaje instantáneo (por ejemplo WhatsApp). Se detallarán sus extensas cualidades en el apartado 7 sobre exploiting.

Social enginyering toolkit (SET): es una herramienta incluida en kali basada en Python y creada para la confección de ataques de ingeniería social. Mediante su menú podemos seleccionar diversas y útiles funciones en la que se encuentra variedades de ingeniería social (ataques SMS, ataques con códigos QR, usando powershell y un largo etc.).

Photoshop: Es muy posible que se requiera falsificar algún documento o alguna fotografía para el robo de identidad o el piggybacking, por lo que una herramienta sofisticada para la edición de fotos es de gran utilidad.

Netcraft y PhishTank: estas dos herramientas no son para propiciar la ingeniería social, sino todo lo contrario, ayudan a detectar el phishing y los pop-ups para las posibles intrusiones.

4.3 Contramedidas

Dada la naturaleza humana de los ataques de ingeniería social, ninguna empresa está a salvo de estos ataques. Dicho esto, algunas de las contramedidas recomendadas para minimizar los riesgos en lo posible son:

- Utilizar adblocks para bloquear ventanas pop-ups y banners maliciosos
- Cambiar periódicamente las contraseñas y utilizar si es posible el doblecheck, es decir, vincular a un segundo teléfono o dispositivo una segunda verificación de inicio de sesión.
- Establecer políticas sobre contraseñas fuertes y poco intuitivas que dificulten los ataques.
- Identificar a los trabajadores con placas identificativas, uniformes, etc. para que sea más difícil la entrada de un intruso. También cachear a los visitantes y controlar las estancias con cámaras de seguridad complementa esta recomendación
- Cambiar las políticas de seguridad física y endurecerlas. También aumentar las capas antimalware en los dispositivos y sistemas.
- Nunca difundir información personal en redes sociales. Si esto no es posible, al menos restringir las políticas de los empleados a difundir la información. Si se sospecha de un perfil falso, revisar detalladamente dicho perfil en busca de incoherencias (por ejemplo, fecha de registro muy cercana a la comunicación).
- Enseñar a los empleados a detectar los emails de phishing visualmente; diferencias del tono de lenguaje o faltas gramaticales, archivos adjuntos sospechosos, direcciones de email desconocidas o urgencia de las demandas de procedimientos de información son algunas de las pautas que deben saber detectar a simple vista.
- Guías y jerarquías de la información pueden ayudar a distinguir qué tipo de información es relevante y cual susceptible de ser robada (aunque a priori cualquier tipo de información es útil).

CAPÍTULO V

5.0 Análisis de vulnerabilidades

A continuación se van a explicar los diferentes conceptos que debemos tener en cuenta a la hora de empezar esta nueva fase de análisis de vulnerabilidades [6, 7]. Ya tenemos toda la información posible recogida del proceso de gathering junto con la ingeniería social que se haya podido explotar (dependiendo de los límites legales y morales del pentester). En ocasiones también, este proceso se entremezcla con la fase de recogida de información, incluso muchas de las herramientas son las mismas.

Una vulnerabilidad informática es una debilidad en un software o hardware (un producto en general) que permite a un individuo malintencionado ejercer acciones sobre estos sin consentimiento. Para ello compromete la integridad, disponibilidad y/o confidencialidad.

Hay algunos aspectos que conviene matizar [7, 3]. Una debilidad no es una vulnerabilidad en sí, si en esencia el proceso es así. En caso contrario, si es un resultado o proceso deficiente el cual ha sido ocasionado accidentalmente, si sería considerado vulnerabilidad. Por ejemplo, si establecemos un cifrado de 16 bits no sucedería una vulnerabilidad, sino que sería una debilidad de diseño. Si es adecuado o no para el propósito es otro tema, dado que el cifrado ya está programado para ser de 16 bits (suponiendo que funciona correctamente). Por otro lado, la implantación de un cifrado RSA que en algún punto pierde dígitos de la clave pública sí sería una vulnerabilidad.

Una vulnerabilidad de producto aparece cuando algún proceso específico de este no lo lleva a cabo con la seguridad para la que está preparado. Por ejemplo que un programa se comunique vía FTP en texto plano no es una vulnerabilidad si el producto especifica que así debe ser. Si se comunica vía SSL en texto plano sí que será una vulnerabilidad dado que el protocolo SSL nunca debe viajar en texto plano.

En cuanto a la integridad, no es una vulnerabilidad si alguien con permiso legal puede interferir en los datos/archivos, y sí que sería en caso contrario, alguien sin permiso, interfiere en ellos. Los mismo pasa con la disponibilidad. Si alguien ajeno manda peticiones legítimas y los servidores pueden con ellas, no es considerada vulnerabilidad. Sin embargo, en el momento de que el servidor no puede absorber tal cantidad de peticiones, pasa a considerarse vulnerabilidad. Por último, en cuanto confidencialidad, si alguien puede acceder a información que no debería se considera vulnerabilidad, pero si el proceso implica la difusión de información (aunque no sea conveniente) no se considera vulnerabilidad. Por ejemplo, revelar el lugar de tus vacaciones en una red social no es una vulnerabilidad en sí, aunque se pueda aprovechar para un ataque de ingeniería social.

5.1 Estructura de un análisis de vulnerabilidades.

Siguiendo el estándar PTES[7] [13] tenemos cinco categorías en las cuales se engloban los procesos que debe seguir un pentester para poder llevar a cabo adecuadamente el proceso. En el caso de un black hat, es probable que no siga estrictamente este estándar, sino que siga su intuición hasta que consiga el objetivo concreto.

Testeo activo

Los procesos de testeo activo se basan en buscar y aprovechar debilidades de componentes y configuraciones que requieran contacto directo con el objetivo, por ejemplo con una pila TCP o una interfaz web. Si el testeo es de forma automática se utilizan herramientas como escáneres genéricos, escáneres de aplicaciones web, escáneres de vulnerabilidades de red o de redes

[7]Siglas de Penetration testing Execution Estándard, conjunto de estándares para la realización de auditorías informáticas.

VoIP. Por otro lado, se debe complementar con un testeo de **forma manual** para eliminar falsos positivos y para analizar los resultados válidos de las herramientas.

A parte de esto, hay que tener en cuenta que nos encontraremos con IDS, IPS y WAF que nos limitaran las acciones, por lo que deberemos ofuscar los análisis alternando objetivos, cambiar los nodos de salidas, etc. En el siguiente apartado se detallarán algunas técnicas de escaneo principales. Los modos de evasión de IDS, IPS, etc. junto con evasión de Honey pots y otros se detalla en el apartado 7 de éste documento.

Testeo pasivo

Esta fase consiste en buscar vulnerabilidades de manera pasiva, es decir, analizar periódicamente objetivos en busca de cambios de comportamientos, información confidencial intermitente, etc.

Validación

En este apartado se intenta crear una correlación entre los resultados para que sean aprovechables y entendibles. Para realizarlo, se crean dos categorías. La primera es la correlación específica, agrupa los resultados en función de la debilidad encontrada, las cuales están definidas en unos listados identificados por una ID cada una. Estos ID a su vez se categorizan por nombre de los equipos, IP's, direcciones MAC, etc. La otra categoría es la correlación categórica (valga la redundancia) las cuales engloba las vulnerabilidades dependiendo de otras estructuras de categorías, como por ejemplo marcos estándar de cumplimientos (NIST, HIPPA, OWASP, etc.).

Una vez hecho esto, se realizan prueba manuales específicas para cada protocolo encontrado, por ejemplo, para servicios VPN, DNS; WEB, SMTP, Citrix, etc.

Por último, se documentan los vectores de ataque específicos. Aquí es muy importante tener en cuenta los límites establecidos en la fase previa a la auditoría, dado que es fácil definir un vector de ataque que provoque daños en el sistema. Para todo ello, se realizan árboles de ataque que se detallan y actualizan durante toda la auditoría siguiente, y que serán uno de los pilares del informe final realizado por el pentester. En el caso de un hacker malintencionado, también deberá mantener estos árboles actualizados si quiere explotar todos los objetivos. Tener un laboratorio a modo de réplica, nos ayudará a realizar pruebas con los exploits para saber el alcance del ataque y el control de éste, para evitar daños involuntarios.

Investigación

Una vez identificadas las vulnerabilidades se abre la tercera fase, que es la de investigación, que puede ser pública o privada. En la investigación pública se concreta la gravedad del problema. Esto puede variar si se trata de una administración pública o en unos repositorios de paquetes de open source, por lo que es importante determinar el alcance. Existen bases de datos de exploits y módulos frameworks de disponibilidad pública, que están categorizados y comentados en función de sus vectores de ataque, consecuencias, IDs, etc [5]. También existen frameworks especializados en exploits como el caso de Metasploit, que nos ayudarán a identificar más debilidades, junto con características adicionales útiles. Todo ello se detallará en el apartado 7, cuando se hable de explotación de sistemas. También podemos encontrar en la web, manuales con configuraciones por defecto (como contraseñas y usuarios en routers) de diversos dispositivos y software, que algunos administradores no cambian y acaban siendo vulnerabilidades por defecto de configuración. A parte, también existen guías de fortificación y errores de configuración 'típicos' de acceso público y específicos (por ejemplo para un determinado firewall) que ayudan en la labor.

Por otro lado, existe la investigación privada, que se basa en tres pilares. El primero es configurar una réplica de entorno. Para ello se crea un entorno virtual simulado (con ayuda de una VM). De este modo se puede investigar maneras de explotación de las vulnerabilidades encontradas. Con esto se podrán probar diversas configuraciones y sistemas operativos, que es otro de los pilares. Por último, el tercer pilar es el fuzzing. Este proceso consiste en

introducir datos válidos y aleatorios esperando resultados anormales. Para ello se configura un depurador y se inyectan datos a diversas zonas de la memoria con el fin de analizar los resultados cada vez que aparezca una anomalía.

Señalar también el uso de pruebas alternativas para identificar posibles vías o vectores de ataque. Por ejemplo, podría ser entradas de caracteres no válidos o excesivamente largos en un login de una aplicación web esperando fallas.

Para acabar, hay que resaltar la ingeniería inversa, que es otro gran arte que está a medio camino entre el análisis de vulnerabilidades y el exploiting, y entre la legalidad y la ilegalidad. Algunos veteranos de la seguridad informática comentan que es el verdadero hacking. Esta especialidad consiste en descompilar el código de un software y analizarlo para encontrar 0days o nuevas debilidades; lo que se haga con estos descubrimientos dependerá del pentester/cracker. Estos procesos se detallarán en el apartado 7 de este documento.

5.2 Técnicas de escaneo

En el apartado de testeo activo se ha hablado sobre la posibilidad de ser detectado por IDS, IPS, etc. por esto, existen unas técnicas específicas que no se pueden dejar de comentar para realizar un análisis [6].

Escano FULL o Connect-Scan

Este escaneo es como el anterior pero completando la comunicación. Es beneficiosos porque genera menos falsos positivos pero muy probablemente quedará registro en un log.

Escaneo UDP

Estas pruebas son sobre protocolos UDP. Se envía un paquete UDP al host objetivo. Si recibe un ICMP port-unreacheable el puerto se declara como cerrado, si se recibe otro tipo de ICMP (código 1, 2, 9,10 o 13 tipo 3) se decalra como filtrado y si se retorna un segmento UDP, el puerto está abierto. También dejará huella en el registro dependiendo del resultado.

NuLL-Scan, Fin-Scan y XMAS-Scan

Aquí se modifican los flags de un paquete TCP. Dependiendo de las que se dejen activas el análisis se codifica como Null-Scan, Fin-scan o XMas-Scan (o Xmas-tree). Dependiendo de los resultados, se pueden determinar los puertos abiertos. Este método genera muchos falsos positivos dado que algunos fabricantes siguen otras normas RFC por lo que hay analizar los resultados cuidadosamente.

Escaneo ACK

Con esta prueba, se busca si existe un firewall en la red o no. Se activa el flag ACK y se envía al puerto abierto del destino, si se recibe un RST implica que el puerto no está filtrado. Después se catalogan los puertos con respuesta ICMP – *algún error*, como filtrados.

Técnicas de anonimato

Unas técnicas de anonimato a la hora de realizar los escáneres es la utilización de servidores proxys. Estas conexiones se implantan como intermediarios entre las el auditor y el objetivo, siendo estas las que realizan las peticiones de escaneo. De este modo la información que pueda quedar en los logs no es la de los equipos y redes del auditor, sino la de los servidores proxys. Normalmente estos servidores están en otros países donde las leyes son menos restrictivas con el fin de ofuscar las acciones. También se pueden utilizar redes alternativas diseñadas para prevalecer el anonimato de los navegadores, como por ejemplo la red TOR, que mediante un sistema de nodos descentralizados y urls cifradas, resulta extremadamente difícil analizar los datos que circulan por ella. De todas maneras se sabe que, en ocasiones alguna institución gubernamental como la NSA norteamericana, ha instalado nodos controlados

para analizar el tráfico de usuarios de esta red. Dada su característica anónima, es caldo de cultivo para todo tipo de usuarios con intenciones maliciosas e ilegales como venta de droga, armas, etc [20].

5.3 Herramientas de escáner

Nmap+NSE (Nmap + nmap scripting Engine): De nuevo el todopoderoso Nmap se ve reforzado con un motor de ejecución de scripts con los cuales realiza descubrimientos de red, detección de vulnerabilidades, detección de versiones más sofisticadas, detecta backdoors e incluso explota dichas vulnerabilidades. Es una de las herramientas más potentes en muchos campos [1].

OpenVAS (Vulnerability Assessment System): como su nombre indica, es un analizador de vulnerabilidades de código abierto, basado en Nessus. Está incluido en Kali 2.0 [7]

Nessus: Es otro peso pesado en el software de escáner de vulnerabilidades. Incluye perfiles agresivos que puede combinar scritps experimentales, probar servicios basados en SSL, etc. Se puede combinar también con Nikto para potenciar sus resultados [5].

Burp Suite: esta herramienta, es otra de las más completas y ya ha sido utiliza en el proceso anterior de gathering. Dispone de muchos recursos como los Spiders, y ofrece un 'todo en uno' para tener con resultados compactos [6].

Hay varios softwares interesantes adjuntos en el Anexo.

Existen 5 tipos generales de vulnerabilidades [14]. En la figura 5.1 podemos ver un cuadro de algunos de los posibles tipos de vulnerabilidades que nos podemos encontrar al realizar los escáneres.

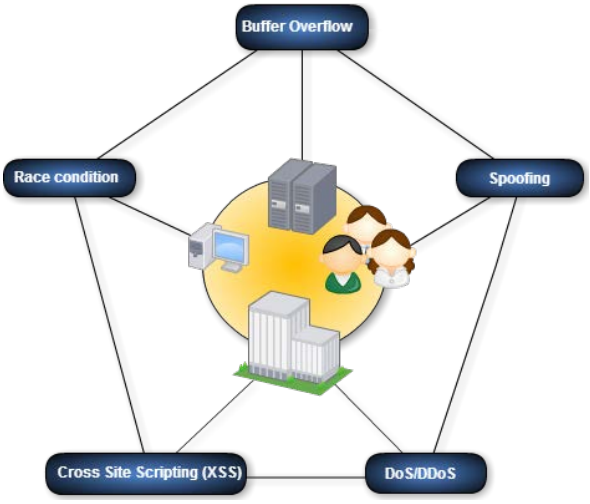


Figura 5.1 – Esquema de los tipos de vulnerabilidades más encontradas.

Las casusas típicas de estas vulnerabilidades también se pueden catalogar en varios tipos:

Diseño: implican debilidades de diseño de protocolos de red y políticas de seguridad deficientes o nulas.

Implementación: Incluyen existencia de backdoors, errores de programación y descuido de los fabricantes.

Uso: Esta causa implican malas configuraciones, desconocimiento/ falta de sensibilización con los temas de seguridad informática, poca disponibilidad de las herramientas de seguridad y limitaciones gubernamentales de tecnologías efectivas en pro a la seguridad de las TIC.

0Day: Son aquellas que se conoce su existencia pero aún no se ha establecido ninguna corrección por parte de la empresa o institución fabricante.

5.4 Contramedidas

A continuación se detallan algunas de las principales recomendaciones y buenas prácticas. Las medidas son parecidas a las de hardening[8] [10], [12], [11] de redes, servidores y aplicaciones web en general:

- Realizar un hardening de los sistemas operativos y las aplicaciones de los equipos que vayamos a conectar a la red.
- Deshabilitar los servicios y puertos que no se vayan a utilizar. Por ejemplo si solo es consulta de información via HTTPs, solo necesita los puertos 80 y 443 abiertos. Se trata de minimizar y acotarlo recursos.
- Habilitar las actualizaciones automáticas de los sistemas operativos.
- Todos sabemos que hay sistemas operativos más seguros que otros, por ejemplo Windows server y Apache, por lo que es recomendable utilizar los menos susceptibles a ataques y análisis.
- Tener al día los contratos con suministradores de software/hardware para acudir a ellos en caso de aparición de un 0Day.
- Diseñar la red con DMZ, IDS, firewalls, Honeypotsetc. para evitar el escaneo indiscriminado…
- …y aplicar y configurarlos correctamente, es decir, aplicar las reglas a los firewalls indicadas para rechazar escaneos (desde fuera hacia adentro y al revés).
- Realizar la tarea de un pentester en la propia red para intentar descubrir fallas de seguridad.

5.5 Enumeración

La enumeración es una fase incluida en la fase de análisis de vulnerabilidades que consiste en determinar la mayor parte de información de las subredes y sus usuarios para completar todo el proceso que se lleva a cabo hasta el momento. Normalmente se hace desde la misma red interna en el caso de que el contrato de auditoría de seguridad lo permita. En el caso de un hacker malicioso este proceso lo llevará a cabo mediante explotación de algunas de las vulnerabilidades que ha encontrado en la fase anterior. Por ejemplo podemos encontrar datos usuarios y grupos, nombres de equipos y dispositivos, recursos compartidos y aplicaciones o grupos de dominios [6].

5.6 Vectores de ataque y herramientas de enumeración

Usuarios y grupos

Normalmente, para enumerar a usuarios y grupos se utiliza CIFS/SMB[9] bajo SNMP sobre active directory en plataformas Windows. En el caso de Linux, se utiliza NIS[10] o SMB/NetBIOS sobre LDAP[11]. Esto significa que se intercambian una serie de mensajes y comandos

[8]Proceso de fortificación de una red/servidor. Se intenta limitar las acciones para minimizar las amenazas.
[9]CIFS/SMB: Common Internet File System y Server Message Blocks respectivamente.
[10]NIS: Network Information Service.
[11]LDAP: Leghtweight Directory Access Protocol, base de código abierto en la que se fundó active directory.

específicos para estos métodos que incluyen mensajes de manipulación de nombre y archivos, mensajes para establecer la conexión, mensajes de impresoras, etc. que son susceptibles de enumeración. Una herramienta útil para NetBIOS es SuperScan o Hyena. Para enumeración LDAP un software útil es Softerra LDAP Administrator. Para SNMP tenemos opUtils o Engineer's Toolset

Nombres de equipos y dispositivos internos/externos

Para la obtención de nombres de quipos y dispositivos podremos utilizar **Nmap**. Es muy habitual encontrar nombres excéntricos de los equipos como personajes famosos de televisión/cine, lugares del mundo, etc. Existe un protocolo especial para los dispositivos internos llamado uPnP[12] que permite encontrar configuraciones automáticas a dispositivos externos recientemente conectados (por ejemplo por USB). Pero solo eso, sino que existen redes como las de P2P que también utilizan estos protocolos. Por lo tanto, es posible obtener información sobre equipos que tengan este protocolo habilitado creando peticiones específicas. Nuevamente Nmap hace esta tarea.

Recursos compartidos

Encontrar recursos compartidos puede otorgar un gran valor dado que pueden contener datos sensibles como tarjetas de crédito, archivos de configuración, etc. Con las herramientas anteriormente descritas podremos encontrar este tipo de información, o con otras nombradas en la fase de gathering sobre enumeración y descubrimiento DNS.

Aplicaciones

Por último, para encontrar aplicaciones podemos utilizar también Nmap con diferentes instrucciones. Otros softwares que pueden servir son Telnet, netcat o Nessus. Mediante puertos encontrados, estas aplicaciones pueden conectarse a éstos, y realizar escaneos de credenciales y obtener todas las aplicaciones instaladas. Estos ataques son nombrados como tipo caja blanca.

5.7 Contramedidas enumeración

Estas son algunas de las contramedidas que debemos tener en cuenta para evitar la enumeración:

- Deshabilitar el servicio SNMP o el SNMP agent, y si no esto no es posible, cambiar los nombres por defecto y actualizar el protocolo a SNMP3 que encripta los mensajes y passwords.
- Crear restricciones de acceso a conexiones anónimas.
- Crear accesos seguros IPSec, sesiones nulas y sesiones compartidas.
- Deshabilitar las transferencias de zona DNS y asegurarse de que la información sensible como IPs internas no están publicadas en ellas.
- Habilitar el registro de los servicios DNS y utilizar estándares de red para los administradores de estos.
- Los servidores SMTP deben estar configurados para ignorar los emails de destinatarios desconocidos y no incluir detalles sobre otros emails, mails del servidor, o de los hosts locales en las respuestas.
- Usar SSL para evitar enumeración de LDAP y utilizar nombres de usuarios diferentes para las direcciones de email y las de usuarios de equipo, etc.
- Deshabilitar el protocolo SMB en servidores DNS y deshabilitar los puertos TCP 139 y 445 asociados al protocolo.

[12] uPnP: Unerversal Plug and Play

CAPÍTULO VI

6. Malware

A estas alturas, seguro que todos sabemos qué es un malware, pero la verdad es que existen muchas clases de ellos. Como antesala de la fase de explotación de sistemas y redes, un atacante creará malware basándose en las vulnerabilidades encontradas y en toda la información que, mediante la fase de gathering, ha conseguido.

Por definición, un malware viene de la palabra *Malicious Software* y es un programa con intenciones maliciosas que normalmente se instala en nuestro pc aprovechando alguna vulnerabilidad o debilidad del sistema [1, 2].

6.1 Tipos de malware

Conviene destacar algunas categorías de malware menos conocidas para tener una perspectiva de todo lo que pueden llegar a hacer. Dependiendo de su comportamiento y de sus intenciones podemos crear 14 tipos [1, 2, 9, 22].

Virus: un virus concretamente es un código malicioso que busca infectar otros archivos ejecutables incrustando el código del virus en éste. De éste modo, cada vez que se ejecute el archivo infectado, pasará a ser una nueva fuente de infección.

Gusanos: los gusanos a diferencia de los virus, no necesitan estar alojados dentro de un archivo portador, sino que se reproducen por sí solos a través de la red (emails, redes P2P, etc.).

Troyano: este malware es un pequeño programa que se suele alojar en otro archivo huésped con el fin de pasar inadvertido. En el momento que se ejecuta el archivo portador, éste se instala y pasa a realizar acciones transparentes al usuario. Los troyanos suelen ser muy modificables y pueden hacer una amplia variedad de acciones como permitir el acceso a un atacante a la máquina, recoger todo lo que se entra por teclado, realizar capturas de pantalla/webcam, y un largo etc.

Keylogger: es un pequeño programa que se encarga de guardar todo lo que se entra por teclado, incluyendo contraseñas, emails, usuarios, etc. Se suelen incluir en los troyanos para ampliar la utilidad es estos. También se busca que pasen desapercibidos para que, en un momento dado, el atacante puede recibir la información. Un usuario infectado con un keylogger, expondrá todas las contraseñas, nombres de usuario, conversaciones, etc.dado que quedarán registradas.

Backdoors: son unas pequeñas aplicaciones que se instalan y buscan pasar desapercibidas con el fin de establecer un canal de conexión entre la máquina víctima y la del atacante. Se pueden utilizar para obtener acceso al sistema infectado en el momento en que se quiera o para crear botnets.

Botnets: este código malicioso es controlado por un atacante y forma redes con otros ordenadores infectados con el mismo código. Dado que permanece el malware oculto en el sistema (se dice que es un equipo bot o zombie), el atacante utilizara la red de equipos infectados a su antojo para realizar ataques descentralizados y planificados, por ejemplo D/DoS.

Hijackers: este malware se encarga de secuestrar las acciones de nuestros navegadores de internet con el fin de redireccionar las conexiones hacia servidores asociados o interesados por el atacante. Normalmente se les suele incluir en los adware.

Adware: estos programas ofrecen funciones lícitas (por ejemplo escanear el sistema en busca de malware) pero contienen unos fines secundarios lucrativos. Éstos consisten en permitir un

foco de propaganda activo por el cual se financia el software. El usuario suele aceptar en los términos de uso e instalación el consentimiento a que el programa haga estas actividades intrusivas y molestas, por lo que en la mayoría de casos es totalmente legal.

Spyware: este software es similar al adware pero sus actividades no se limitan a enviarnos publicidad, sino a recopilar toda la información posible sobre el usuario (hábitos de navegación, IP's, webs visitadas, etc.). Esta información es enviada a empresas cuyos objetivos es hacer campañas de spam. También recalcar que al igual que el adware, suelen hacer realmente alguna función legítima como software (normalmente de dudosa calidad), aunque su verdadera intención sea otra.

Rogue: estos son programas que aparentan ser lo que no son, por ejemplo un falso antivirus que siempre nos detecta virus, o un falso optimizador de sistema que de nuevo siempre nos detecta optimizaciones críticas. El objetivo es el de vender el software falso al usuario antes de que se descubra la estafa.

Riskware: este software no es ningún programa malintencionado (puede ser cualquier programa) pero está desactualizado. Se ha incluido aquí por ésta razón, ya que un programa desactualizado y del que se conoce una vulnerabilidad, es una puerta abierta a infecciones de malware y explotaciones.

PUP: son las siglas de Potentially Unwanted Programs y como su nombre indica, son programas que se instalan sin el consentimiento explícito de usuario. Las intenciones son las típicas, recaudar información del usuario, utilizar los recursos del ordenador para beneficiar las intenciones del atacante, etc.

Rootkit: este software es un conjunto de herramientas que se alojan en los archivos más internos y vitales del sistema para que en caso de intento de eliminación dejen el sistema inutilizable. Las acciones que hacen van desde introducirse en el sistema, a realizar ataques de fuerza bruta, pasando por establecer comunicaciones con el atacante y un largo etc. Son una navaja suiza del malware, un todo en uno.

Ransomware: son códigos maliciosos que encriptan los datos del sistema dejándolos incomprensibles si no se dispone de la clave. Después se chantajea al usuario para que pague una cantidad concreta a cambio de la clave, y así, pueda recuperar sus datos. En los últimos años, los ransomware han proliferado y se han especializado siendo una amenaza muy grave. Algunos ejemplos conocidos de este tipo de malware son Cryptolocker, que apareció en 2013. El funcionamiento consistía en crear un par de llaves de 2048 bits tipo RSA con las que encriptaba archivos de una extensión concreta. Luego se pedía el pago de 1 bitcoin en un plazo de tres días, el cual se incrementaba hasta 10 bitcoins (unos 2300$ en aquel momento). Otros ejemplos más recientes pero con los mimos objetivos son CryptoWall 3.0 (2015), TeslaCrypt (2015), Petya (2016). En la mayoría de ocasiones se conoce el ransomware pero no se conoce el código fuente ampliamente. A medida que se van descubriendo tipos de ransomware, se van publicando en blogs y webs de seguridad (ver anexo I).

6.2 Focos de infección

Los principales vectores de infección de malware, son procedentes de internet o de dispositivos físicos [20].

Los primeros engloban la categoría de emails con técnicas de phishing o spam, redes sociales mediante banners o links en chats fraudulentos, servidores web fraudulentos que contengan software maliciosos, incluso webs legítimas infectadas sin el conocimiento de sus propietarios. También provienen de programas que lucen ser gratuitos o de descargas de redes P2P, donde los archivos no están controlados.

En la categoría de dispositivos físicos se incluyen todos los medios extraíbles en un sistema como DVD's/CD's, USB, que contengan virus que se replican automáticamente al conectarse o iniciarse.

Mencionar también que cuando se habla de sistemas, se incluyen los dispositivos móviles, que por supuesto no están exentos de amenazas. Mediante SMS o descarga de software malicioso los medios portátiles quedan igualmente infectados. Además suelen ser un foco menos controlado por los usuarios y más suculentos para los atacantes dada la gran cantidad de información personal que suelen contener.

6.3 Herramientas

Existen dos principales maneras de crear un malware [3, 4, 22]. La primera es directamente crear un código en un lenguaje de programación adecuado para el sistema a infectar. Por ejemplo, si queremos crear un malware para Windows, podemos utilizar Batch, C, Python (en realidad casi cualquiera nos vale). Después de crear el código, se compila y se trata para dar un formato común como .pdf con herramientas llamadas wrappers. Después, para que pase desapercibido para los antivirus, se tratan los archivos con programas llamados crypters. La otra manera es utilizar herramientas especializadas en crear malware, que facilitan el proceso dejando escoger qué tipo de acciones y formatos se quieren conseguir. A continuación se detallan algunas de todas estas herramientas.

Crypters: estos programas se dedican a esconder el malware o las herramientas maliciosas de tal forma que los antivirus no los puedan detectar fácilmente. Para conseguirlo, el crypter cifra en código fuente del archivo malicioso (por ejemplo un .exe) de modo que cuando la víctima ejecuta la aplicación, ésta se descifra e infecta el sistema. Normalmente los crypters vienen en dos partes, una es el 'builder' y la otra es el 'stub'. La primera es la que se encarga de ensamblar el stub con el archivo de código malicioso, un proceso relativamente sencillo y legítimo.

La segunda es más elaborada. Es la que se encargará de descifrar el malware cifrado y cargar el contenido directamente en la memoria del sistema mediante una técnica llamada RunPE o Dynamic Forking. Esta técnica consiste principalmente en crear dos procesos en memoria legítimos, uno con el malware descifrado y otro proceso 'vacío' para dejarlo en estado de suspensión. Luego sobrescribe dicho proceso con el código malicioso y reactiva el proceso suspendido consiguiendo que el malware se ejecute. El principal problema suele ser la ocultación del stub para que no sea detectado por los antivirus, para ello se editan el código fuente y el binario (modding) buscando las firmas que utilizan los antivirus para modificarlas y que estos no los detecten [15]. Algunos ejemplos de estas herramientas son AIO FUD Crypter, Hidden Sight Crypter, Galaxy Crypter, Criogenic Crypter, Heaven Crypter.

Wrapers: Estos programas se encargan de cambiar el aspecto real de un archivo que contiene el código malicioso (por ejemplo un .exe), transformándolo en uno menos agresivo para el usuario (por ejemplo un .doc) con el fin sea ejecutado. Lo que hace realmente es envolver en un mismo archivo, el archivo malicioso y uno legítimo, con lo que al ser ejecutado, el archivo legítimo se abre normalmente y el malicioso se ejecuta en segundo plano de manera transparente al usuario. Se suele utilizar mucho para ocultar troyanos aunque se puede utilizar para cualquier malware.

Creadores de virus, troyanos y gusanos: como se ha comentado, existen programas que facilitan la creación de malware, ofreciendo su creación como 'un menú a la carta'. En el caso de creación de troyanos tenemos Dark Horse Trojan Virus Maker. Para crear virus temos software como Sam's Virus Generator, JPS Virus Maker, Andreinick05's Batch Virus Maker, DeadLine's Virus Maker, Sonic Bat Batch File Virus Creator o Poison Virus Maker. En la creación de gusanos tenemos algunos como host Eye Worm o Internet Worm MAker Thing.

R.A.T's (Remote Acess Trojans): Las RATs son troyanos que buscan el control total sobre la maquina infectada. Una vez infectada la máquina con el troyano servidor, éste se conecta a través de internet a la máquina del atacante dándole absoluto control del sistema infectado. Algunas de las RATs más populares son: Optix Pro, MoSucker, BlackHole, SSH-R.A.T., etc. (ver anexo I para lista complementaria de RATs.).

Herramientas anti-malware: para no pasarlo por alto, existen muchos y variados softwares antivirus. Algunos de estos son ESET Smart Security, Kasperky Anti-Virus, MalwareBytes Anti-Malware, Norton Anti-Virus, etc. También existen algunas muy específicas que buscan determinados tipos de malware concreto, como por ejemplo TrojanHunter, Emsisoft Anti-Malware, Immunet o www.virustotal.com. (Ver anexo I para ver más).

6.4 Contramedidas

Las contramedidas más típicas se intuyen ya, una vez se ha jerarquizado y esquematizado todo el malware. Estas son algunas de las medidas que se deberían tomar [1]:

- No abrir archivos adjuntos procedentes de emails sin su previo análisis. Además, nunca abrir un archivo con extensión doble, por ejemplo 'archivo.exe.ppt'
- Instalar antivirus y antimalware no es suficiente, se debe tener actualizado debido a que el malware no solo prolifera a sus anchas una vez subido a internet, sino que se crea nuevo constantemente. También deben activarse la protección en tiempo real y deben realizarse escaneaos periódicos.
- No insertar en los sistemas dispositivos extraíbles sin saber su procedencia. Incluso aunque sean de confianza, siempre analizar el contenido previamente.
- Si se tienen dudas sobre algún archivo infectado, no correr riesgos, y realizar los análisis desde una máquina aislada de la red.
- Bloquear los puertos que no se necesitan habitualmente, para evitar que el malware (sobre todo los troyanos y las RATs) puedan comunicarse con los atacantes.
- No instalar ni descargar programas procedentes de la red, si no se sabe al 100% que son fiables. Revisar los términos de seguridad a aceptar antes de instalar para verificar que no existen permisos adicionales (en los dispositivos móviles incluidos). Si hay que hacerlo, verificar los checksum MD5 y seguir cuidadosamente las instrucciones del responsable de seguridad.
- No acceder a links externos procedentes de emails o chats de redes sociales. Pueden dirigir a webs maliciosas.
- Monitorear el tráfico de red para analizar conexiones anómalas.
- Mantener actualizado todo el software. Es útil permitir a estos que se actualicen automáticamente para evitar despistes.
- Establecer unas políticas de uso restrictivas referentes a los usuarios; si un usuario solo debe acceder a un programa, eliminar la posibilidad de que pueda acceder a otros apartados del sistema operativo, a otros programas o a los medios extraíbles.
- Realizar copias de seguridad periódicamente para poder recuperase de un ataque (por ejemplo de ransomware).
- Tener los firewalls propios de los sistemas operativos y los específicos activos.

CAPÍTULO VII

7. Seguridad en sistemas

Una vez superadas las fases de gathering y análisis de vulnerabilidades, toca pasar a la acción. Seguro que el auditor/atacante a estas alturas tiene un buen número de datos útiles procedentes de la recogida de información, del phishing e incluso tendrá preparado algún troyano o alguna RAT para infectar alguna máquina de la red. Es por eso que cambiamos de fase y se empieza a pensar, qué vulnerabilidades se pueden aprovechar para crear exploits que puedan hacer ganar privilegios al pentester. Para ellos habrá que analizar la red y los sistemas (tanto móviles como fijos), utilizando herramientas apropiadas para la tarea. Se abre ahora un abanico de posibilidades que tendrá una gran dosis de ingenio y conocimientos técnicos para conseguir los objetivos. Este tema es increíblemente amplio, así que aquí se va a intentar dar un enfoque global del panorama que se tiene por delante [6, 7].

Primeramente a modo de introducción, hay que definir algunos conceptos muy básicos.

Exploit: un exploit es un pequeño fragmento de código que busca 'explotar una vulnerabilidad', es decir, aprovecharse del fallo de configuración, programación, ejecución, etc. conocido para llevar a cabo acciones generalmente maliciosas. Normalmente los exploits se escriben en lenguaje C aunque de nuevo podemos encontrarlos en muchos más leguajes como Python, Ruby o Java. Normalmente, para crear un exploit, lógicamente hay que encontrar una vulnerabilidad en el software en el que se basa proceso. Para ello se lleva a cabo una técnica llamada ingeniería inversa, la cual se explicará en el punto 7.4. Dado que esta tarea es bastante complicada, los investigadores, una vez encuentran un la vulnerabilidad, crean el exploit y lo hacen público (en la mayoría de casos), de tal modo que queda al abasto de cualquiera que lo sepa y quiera ejecutar. Por ejemplo, el framwork por excelencia para ejecutar exploits es Metaesploit, el cual lleva asociada una base de datos online consultable para buscar un exploit que nos convenga en cada momento, siempre y cuando sea conocido.

Payload: este nombre es muy común en el lenguaje de los exploits, dado que es el pequeño fragmento de código contenido dentro del exploit, cuyo objetivo es ejecutarse en la memoria de la máquina de la víctima. Realmente es justo lo que provoca el funcionamiento anómalo que busca el exploit y suelen ejercer efectos como ejecutar una Shell en la maquina víctima, descarga/instalación de un archivo, creación de un usuario, etc. Están escritos en lenguaje máquina (ensamblador).

En general existen tres tipos de payloads [7]:

Inline: estos payloads ejecutan una acción muy concreta al inyectarse en la máquina de la víctima.

Stagers: estos se encargan de crear un conexión de red entre el intruso y la maquina víctima generalmente con el fin de realizar una descarga más completa de payloads

Staged: éstos son los payloads que los de tipo stagers descargan para ampliar el abanico de posibilidades de explotación.

7.1 Seguridad en dispositivos fijos

Por lo general, los procedimientos y conceptos que se explicarán a continuación se pueden aplicar a todo tipo de dispositivos incluyendo los móviles. No obstante, se ha querido hacer la distinción entre fijos y móviles dada las peculiaridades especiales que tienen estos últimos.

Los procedimientos de hackeo de sistemas normalmente se dividen en tres fases [1, 3]; la primera es la de obtención de acceso a la máquina, la cual incluye el ataque a contraseñas y el escalamiento de privilegios en el sistema. La segunda fase es la de mantención del acceso, es decir, asegurarnos de que tendremos el control sobre la máquina todo el tiempo que

necesitemos. Ésta incluye la ejecución y ocultación de aplicaciones/malware en la máquina de la víctima o empresa a auditar. Por último y no menos importante, está la fase de limpieza de huellas, que constará de asegurarnos de que no queda rastro de nuestras actividades una vez conseguidos los objetivos. Si lo miramos desde el punto de vista de un auditor con permiso, probablemente no haga falta asegurarse de no dejar rastro.

7.1.1 Ataques a contraseñas

En el momento en que se busca obtener contraseñas de dispositivos, tenemos cuatro escenarios principales [7, 11]:

Ataques activos online

Se incluyen las técnicas en las que se interacciona directamente con la máquina de la víctima con intenciones de craquear las contraseñas. En este apartado existe un surtido de técnicas de craqueo, algunas son ampliamente conocidas y obvias, otras no tanto. Estas son las siguientes:

Ataques de diccionario: mediante un programa y un archivo con miles de palabras (un diccionario) se realizan pruebas de contraseñas una a una, hasta dar con la acertada. Esto como es lógico, puede ser muy tedioso y largo, además de infructuoso como cabe imaginar. Se suele combinar ataques de diccionario con ataques de fuerza bruta.

Ataques de fuerza bruta: como el nombre indica, consisten en realizar todas las combinaciones posibles de unas palabras y/o símbolos y/o letras para conseguir la contraseña. Es la típica técnica de probar todo lo posible mediante ensayo y error. Estos ataques se automatizan mediante programas específicos para ello.

Ataques basados en reglas: esto ataques se hacen cuando se tiene información concreta sobre la persona que creó el password. Dicho de otra manera, si se sabe que la fecha de nacimiento del usuario o la de algún familiar, es probable que la su contraseña sea esta. En base a estas suposiciones, se pueden crear listas de posibles passwords y ataques dirigidos los cuales suelen ser fructíferos.

Ataques a contraseñas por defecto: Es sorprendente la cantidad de usuarios que introducen contraseñas obvias, muy comunes (como '1234' por ejemplo) o simplemente la dejan por defecto (por ejemplo 'Admin'). Así que en este punto, podemos suponer también las contraseñas por defecto en los sistemas.

Ataques de malware: mediante keyloggers, troyanos, spywere se recolectan gran cantidad de datos, entre los que están las propias contraseñas. Claro está que previamente deberemos infectar la máquina de la víctima y lista para la explotación.

Inyección de hash: en ocasiones, las contraseñas se almacenan en formato hash con tal de que ni siquiera los administradores sepan las contraseñas de los usuarios. Este método consiste en capturar un hash validado de un usuario ('logeado') en el sistema objetivo. En ese momento, el atacante utiliza el hash extraído para introducirlo en una sesión y poder acceder al objetivo.

Ataques pasivos online

La mayoría de ataques pasivos a contraseñas se hacen capturando archivos mediante técnicas de man in te middle o robo de datos ampliamente descrito en todo el documento para luego estudiar a fondo el craqueo de lo obtenido. La técnica de escaneo de redes MITM se explicará en el apartado dedicado a ello, el punto 8. Básicamente consiste en interponerse entre la conexión de la víctima, analizando y capturando todo el tráfico y comunicaciones. Una vez capturado el tráfico, este puede viajar en texto plano o cifrado. Cuando encontramos tramas susceptibles de contener las contraseñas pero están cifradas podemos preparar un ataque pasivo para descifrar el mensaje. Los métodos del ataque activo en estos casos siguen siendo válidos (incluso más) dado que el atacante puede tomarse el tiempo que necesite.

Colisión hash: Una vez se está en posesión de un archivo con la contraseña cifrada, si el password está en formato hash, podemos intentar un ataque puro de colisión de cadenas hash para descifrar el contenido. Este método es unidireccional, con lo que aparentemente, sin la palabra correcta no se puede descifrar el contenido. Pero esto no es del todo cierto, se puede hacer un ataque con palabras probables o de diccionario en ese formato buscando colisiones de cadenas de hash. En el momento que tengamos una colisión (coincidencia de resultados hash), sabremos la palabra cifrada.

Ataques Rainbow table: estos ataques están basados en unas tablas que almacenan palabras relacionadas entre sí mediante una cadena hash. El proceso de creación de las tablas consiste en escoger una palabra inicial y obtener la cadena hash, a la cual se le aplica un algoritmo de reducción y se obtiene otra palabra relacionada. Este proceso se repite unas 40000 veces hasta obtener una última palabra relacionada. La primera palabra resultante se almacena junto con la última obtenida. Así se crean unas tablas que se utilizan para comparar los resultados y el/los hash/es de la contraseña capturada (la que se quiere descifrar). Realizando el proceso inverso y comparando, se pueden obtener coincidencias de hashes y por lo tanto los passwords.

Ataque de red distribuido (Ditributed Network Attack, DNA): este tipo especial de ataque consiste en recuperar pequeñas porciones de la contraseña en hash, utilizando la capacidad de procesamiento de máquinas distribuidas (se entiende que infectadas por el DNA). El DNA se aloja en el servidor comprometido y controla todo el ataque de manera pasiva. Protocolos de guardado de contraseñas como los utilizados por Security Accounts Manager (SAM) o active directory database, además de protocolos de autenticaciones distribuidas como NTLM o Kerberos son susceptibles de estos ataques.

7.1.2 Escalando privilegios

Cuando se habla de escalar privilegios, en realidad se está hablando de que el atacante, ya ha obtenido acceso al sistema pero la cuenta de usuario está limitada a las tareas del propietario de esta. Con lo cual, lo que desea es elevar esos privilegios hasta, si es posible, niveles de administrador del sistema, en Linux permiso root. Si esto se consigue, el atacante tendrá control total sobre el sistema, y por lo tanto acceso a toda la información que desee. Esta es la situación ideal a la que todo atacante malicioso querrá llegar. Hay que pensar que si esto ocurre, lo normal que se cambien algunas configuraciones para mantener el acceso total a la máquina, y puede pasar largo tiempo hasta que el black hat sea descubierto (si es que alguien lo hace en algún momento). A partir de este punto ya no habrá marcha atrás, cuando se descubra el daño estará hecho, independientemente de las consecuencias penales del atacante/s. Es un jaque mate [2,6,7,8].

Pero pasar de un usuario común a uno con permisos administrativos no será fácil (en ocasiones). El atacante deberá buscar errores de configuración, de programación o debilidades de uso que le puedan servir. Nuevamente se buscarán exploits y vulnerabilidades desde dentro del sistema para dichos fines.

Existen dos técnicas fundamentales en el escalado de privilegios:

Reseteo de contraseñas: puede sonar muy obvio pero es sorprendente la cantidad de veces que un usuario sin privilegios puede resetear las contraseñas administrativas utilizando la consola de Windows. Por ejemplo, con el comando ‘net user’ se pueden ver todos los usuarios e introduciendo ‘net user nombre*’ se puede volver a introducir la contraseña como nueva substitutiva. Nunca se pueden sobreestimar las obviedades por muy simples que parezcan y esta es una prueba. No siempre funciona este método pero en muchos casos tiene éxito.

DLL Hijacking: el secuestro de archivos DLL consiste en eliminar una librería (un archvio .dll) y reemplazarla por una modificada por el atacante. Cuando el usuario acceda a la aplicación, esta accederá a la librería externa modificada y otorgará acceso al atacante (dependiendo de los privilegios del usuario en cuestión).

7.1.3 Ejecución y ocultación de aplicaciones

Lo ideal es ejecutar troyanos, backdoors, RATs, o Keyloggers en la máquina con tal de mantener el acceso en todo momento. Cualquier malware que hayamos creado en el apartado 6 puede usarse aquí, siempre y cuando se tenga el suficiente cuidado para que no sea detectado. Por ejemplo, se podría instalar un keylogger y esperar a que diversos usuarios introdujesen las contraseñas para poder hacer una escalada de privilegios. Con un poco de suerte/mala suerte el administrador del sistema también introducirá su contraseña con lo que el atacante obtendrá lo que quiere [4, 9, 22].

Rootkits

Aún y así, hay que saber ocultar las aplicaciones maliciosas. Para ello, suelen utilizarse herramientas llamadas rootkits. Estas herramientas o conjunto de ellas, reemplazan algunas llamadas al sistema importantes o críticas por versiones que subyacen actividades maliciosas propias de un malware. De este modo, aparentemente no existe ninguna actividad susceptible de ser investigada. Estos rootkits al igual que pasa con los RATs, suelen incluir un todo en uno de herramientas, como sniffers, IRC bots, programas DDoS, y en definitiva todo lo que el diseñador haya querido introducir.

Existen varios tipos de rootkits, dependiendo de los procesos del sistema que utilizan para ocultar sus acciones. Por ejemplo, tenemos los que manipulan la secuencia de arranque del sistema operativo para arrancarlo como si fuese una VM, los que utilizan el código fuente de los firmware y hardware para alojarse o los que escogen hacerlo en los kernel y drivers básicos de los dispositivos. También existen rootkits que aprovechan vulnerabilidades en el arranque para suplantar estos métodos por otro semejante y manipulado, los que escogen binarios para instalar malware suplantándolos o los que, como se ha comentado, reemplazan llamadas al sistema (SYSCALLS) para zafarse.

Esteganografía y estegoanálisis

Una manera creativa de ocultar información en las maquinas afectadas es la estaganografía. Esta técnica consiste en ocultar información utilizando típicamente una imagen, un archivo de ofimática, de audio o de video. De este modo, el usuario que abre el archivo nunca detectará la ocultación pero realmente estará ahí. Se pueden ocultar códigos fuentes, mensajes con información sensible, planes de acción, etc.

Existen varias y diversas técnicas de ocultación de información, por ejemplo inserción de bits menos significativos en cadenas (para por ejemplo, insertar una imagen dentro de otra), aprovechar los espacios en blanco en archivos ASCII para enviar mensajes ocultos, utilizar algoritmos de transformación matemática para ocultar imágenes (por ejemplo los JPEG utilizan estos método para comprimir los datos). También se puede ocultar información dentro de archivos de video, añadiendo datos al formato de compresión o al código fuente o repartidos en cada frame. En los archivos de audio, se utilizan las frecuencias muy altas o muy bajas para el oído humano para ocultar mensajes de sonido. Normalmente, mediante herramientas concretas se pueden llegar a realizar estas manipulaciones.

Estegoanálisis consiste en el análisis de los archivos en busca de manipulaciones que puedan suscitar técnicas de esteganografía, y dada la diversidad de posibilidades suele ser una labor bastante meticulosa. Normalmente, se buscan patrones en los códigos fuentes o textos, cambios injustificados en el peso de los archivos, paleta de colores en las imágenes subidas/bajadas de tono, fecha de modificación de los timestamp[13] de los archivos y en definitiva anormalidades.

[13] Timestamp: cadena de información que determina la fecha y hora de la última edición del archivo.

Borrado de huellas

Por último, el atacante, una vez haya hecho las labores que tenía por objetivo, querrá eliminar el rastro de sus acciones para evitar represalias y salir airoso [9]. Por lo normal, suelen deshabilitar las auditorias automatizadas, así como la creación de logs. También los manipulan y editan con el fin de pasar desapercibidos. Se suelen utilizar herramientas (ver punto 7.1.4) o hacer manualmente. En sistemas operativos Windows existe una parte de configuración llamada 'Herramientas administrativas', la cual incluye un 'Visor de eventos' que incluyen todos los logs que se van creando en tiempo real. En sistemas Linux, hay que acceder al path /var/log/messages que es un archivo de texto en plano donde se almacenan los registros de las acciones, logeos, etc. Bastaría en cada caso, con eliminar los logs de las acciones pertenecientes a la fecha del ataque. También hay que tener en cuenta datos típicos relativos a los navegadores de internet como coockies, caches, historiales y las mismas aplicaciones/malware que se han ido instalando.

7.1.4 Herramientas

Algunas de las herramientas más útiles de todos los campos comentados se enumeran a continuación [1, 7].

Metaesploit: esta es la herramienta por excelencia, un todo en uno que incluye una base de datos de exploits y payloads, actualizables online, categorizadas por sistema operativo, versiones y software diverso, a disposición de los auditores de seguridad. Esta herramienta es increíblemente amplia y se podría escribir un libro aparte exclusivo explicando el funcionamiento detallado.

Ataques a contraseñas: Para craquear contraseñas tenemos Jhonn The Ripper que probablemente es la más útil a la vez que conocida. También tenemos otras conocidas como Cain & Abel, WinPassword, Windows Password Cracker entre otras. LophtCrack, rtgen, Ophcrack, RainbowCrack y Wirtgen para generar Rainbow tables. Pwdump7 y fgdump para los ataques de redes distribuidas. (Ver anexo I).

Escalado de privilegios: Una muy útil es Active@ Password Changer la cual nos deja recuperar contraseñas de particiones borradas, rastros de SAM e información general de todos los usuarios locales. Otras pueden ser: PasswordLastic, Trinity Rescue Kit, ElcomSoft System Recovery,

Ejecución y ocultación de aplicaciones: algunas de las más útiles son Avatar, Necurs, Azazel o ZeroAccess.

Anti-rootkits: algunas de las principales herramientas para defenderse de los rootkits son los antivirus como Virus Removal Tool, Avira Free Antivirus, Rootkit Buster o Prevx.

Esteganografía: para estas tareas tenemos SNOW (ocultación ASCII), QuickStego, Hide In Picture, OpenStego ImageHide entre otros para ocultación de imágenes. wvStego, DataStash, Office XML o Camouflage para ocultación de documentos, OmniHide, Masker, DeepSound, MAXA Security Tools o SilentEye para edición de video y audio.

Estegoanálisis: podremos apoyarnos en herramientas como Gargoyle Investigator Forensic Pro, Stego Suite, Stegdetect, StegAnyzerAS, ImgStegano, StegSpy o Steganography Studio.

Eliminación de logs: Auditpol, clearlogs.exe (incluida en Windows por defecto), comando clearev en metaesploit, CCleaner, MRU-Blaster, Wipe, ClearProg, BleachBit, Privacy Eraser o AbsoluteShield Internet Eraser Pro.

7.1.5 Contramedidas de seguridad fija

Medidas de seguridad para evitar el craqueo y proteger las contraseñas [1, 7, 11, 22]:

- No utilizar los mismos passwords para todo e cambiarlos periódicamente. Tener especial cuidado con los métodos para recordar; escribir como contraseña M@r1a aunque parezca una contraseña fuerte, no lo es dado que se pueden deducir.
- Utilizar contraseñas fuertes, con más de ocho dígitos, que combinen números, caracteres especiales, empleando mayúsculas y minúsculas.
- Nunca compartir las contraseñas y asegurarse de que se guardan encriptadas de algún modo. También inhabilitar el recordatorio automático de contraseñas en las webs y aplicaciones.
- Lo ideal es recordar las contraseñas en la mente, es decir, nunca guardarlas en texto plano, pero si no es posible, almacenarlas debidamente encriptadas.
- Nunca poner palabras directas del diccionario ni dejar las contraseñas por defecto en los sistemas. Tampoco poner obviedades como la fecha de nacimiento o el nombre de la mascota.
- Utilizar la técnica de salt (cadena de caracteres aleatorios que se adjunta a la cadena hash que almacena a una contraseña) siempre que sea posible.
- Protegerse de los ataques de fuerza bruta habilitando los logs para que guarden el monitoreo de estos ataques. Además, limitar los intentos válidos de entrada de contraseña a no más de 3 intentos. Bloquear la cuenta en caso de más intentos fallidos.

Recomendaciones para evitar el escalado de privilegios [1, 8]:

- Encriptar siempre que se pueda los datos sensibles o críticos del sistema.
- Por defecto, crear perfiles de usuarios con los mínimos privilegios respecto al resto del sistema.
- Siempre es mejor prevenir que curar dado que una vez se ha robado la información, no ha segundas oportunidades.
- Restringir y acotar los privilegios de los códigos que se ejecutan en la máquina al mínimo.
- Existen herramientas de debug (testeo de errores) que se deben aplicar con el fin de realizar pruebas de estrés en los sistemas.
- Como siempre, mantener el equipo actualizado es la mejor manera de prevenir posibles explotaciones de vulnerabilidades (se deduce que las empresas de software actualizan y corrigen rápidamente los errores que salen a la luz con parches, aunque no siempre es así.).
- Si es posible, habilitar la doble autenticación mediante imágenes o pequeños juegos que sirvan como protección de tareas automatizadas.
- Ejecutar servicios con cuentas sin ningún permiso aparente, es decir, con la máxima restricción.

Recomendaciones para defenderse de los Rootkits [1, 22]:

- Tener siempre instalados los firewalls, tanto por software como de red
- Verificar que se entienden los documentos de instalación y que las fuentes son fiables. No instalar software innecesario o que no está verificado por el administrador.
- Realizar escaneaos del kernel periódicamente así como cerciorarse de que las restauraciones del sistema están activas
- Nunca logearse con privilegios administrativos en caso de que se pueda evitar.
- Mantener el software y los antivirus actualizados y asegurarse de que contienen protección activa y contra rootkits.

7.2 Seguridad en dispositivos móviles

Después de todo lo comentado, cabe hacer comentarios especiales para los dispositivos móviles [1, 11]. La mayoría de métodos y explicaciones dadas sieguen siendo válidas pero con la llegada de los smartphones y tabletas el abanico de posibilidades se amplía considerablemente. Sobre todo, aumentan los frentes abiertos, focos de infecciones de malware.

7.2.1 Vectores de ataque

Los principales vectores de ataque que nos podemos encontrar (algunos ya mencionados y comunes) son los siguientes (figura 7.1)

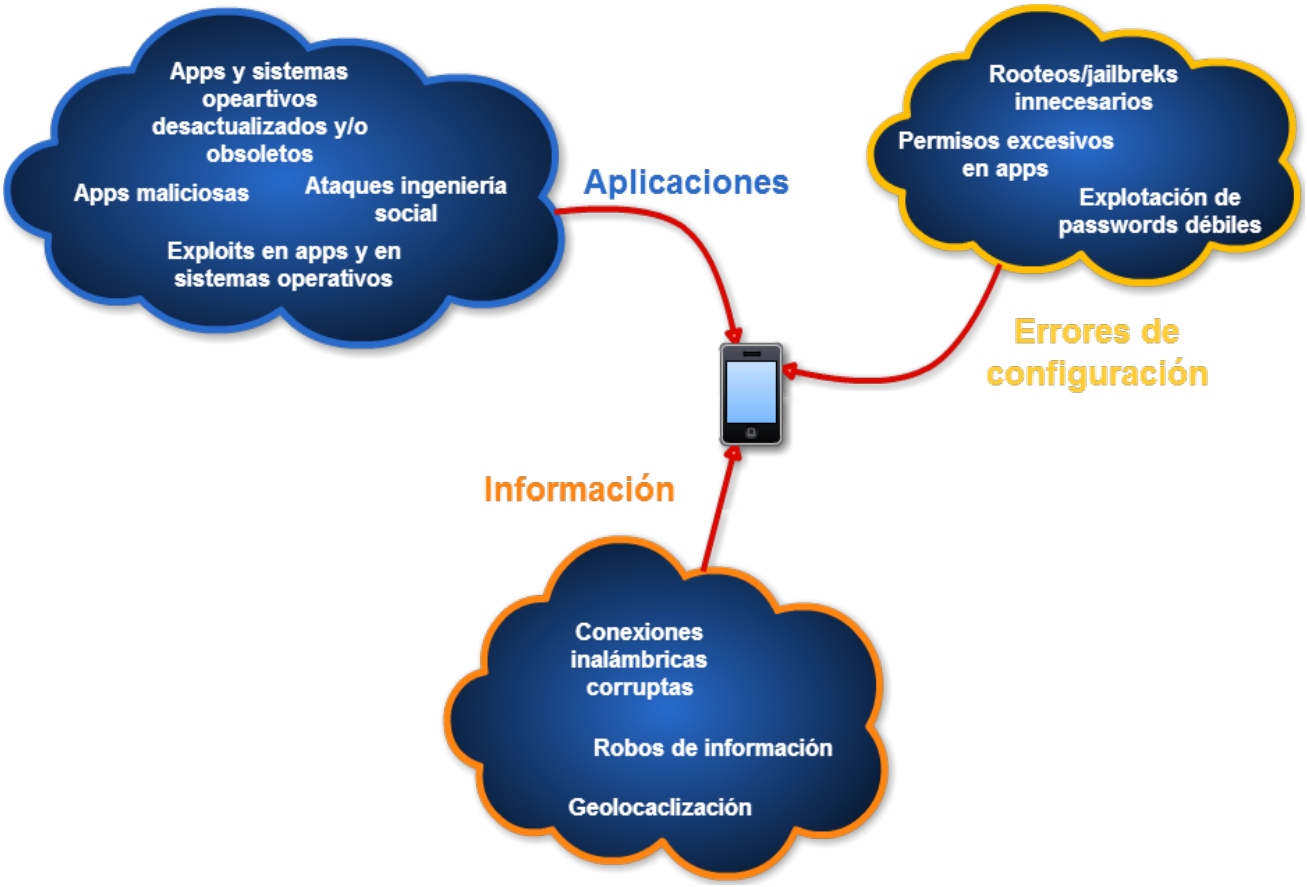


Figura 7.1. Vectores de ataque en plataformas móviles

7.2.2 Plataformas móviles

Existen numerosos sistemas operativos que son utilizados por los dispositivos móviles. Android e iOS son los más usados, pero existen otros como Symbian, BlackBerry iOS, ChromeOS, Windows Phone o Firefox OS [1] . Aquí, nos centraremos en los dos primeros dado que son los más utilizados.

Android

Sistema operativo más usado tanto en tabletas como en smartphones, propiedad de Google y basado en Linux, por lo tanto, heredero de muchos exploits descubiertos de dicho S.O.

Arquitectónicamente consta de consta de una base de datos estructurada SQLite, navegador de internet integrado basado en el software libre WebKit engine, soporta multiples formatos de audio, video, imagen, etc. Si le sumamos a todo esto un rico entorno de desarrollo (debug, emulador de dispositivos, plugin para Eclipe, etc.) y una plataforma de aplicaciones online

masiva, nos encontramos con un ordenador personal en el bolsillo, susceptible de ser vulnerado por la gran información sensible que poseen.

Su arquitectura a nivel interno se divide en 4 capas:

1. Aplicación: incluye las aplicaciones típicamente utilizadas por los usuarios comunes, contactos, navegador, teléfono, etc.

2. Framework de aplicación: este nivel incluye aspectos secundarios y de administración respecto del usuario, como la gestión de ventanas, manejo de notificaciones, instalación de paquetes, manejo de localizaciones, etc.

3. Librerías: El tercer nivel es transparente al usuario e incluye todas las librerías necesarias para el funcionamiento del sistema. Esto incluye por ejemplo gestión de SSL, libc, SQLite, WebKit, OpenGL, y las librerías de ejecución en tiempo real nombradas como Android RUNTIME etc. Por defecto, un usuario no tendrá nunca acceso a esta capa a no ser que sea desarrollador o haga rooteo al teléfono, que no es más que aplicar técnicas de elevación de privilegios en el propio dispositivo.

4. Linux Kernel: Aquí es la capa más profunda del sistema operativo y la más sensible. Es la base de éste ya que incluye todos los drivers de cámara, wifi, memoria flash, keypad, pantalla, batería, audio, etc.

Un atacante siempre intentará efectuar un ataque empezando por lo más simple posible (primeras capas) hasta llegar a la manipulación del kernel. Por defecto, los usuarios de estos sistemas operativos no tienen todos los permisos posibles para evitar daños y por motivos de seguridad, en caso de infección de malware. Es por eso, si un atacante logra acceder al dispositivo, muy probablemente quiera tener permisos administrativos (también llamados de super usuario, o root) y a partir de ese punto, podrá ejercer acciones a su antojo. Una acción común es modificar un sistema operativo (llamados ROMS) que incluyen funciones especiales para luego instalarlas en un dispositivo con permisos root. El proceso de root está expresamente prohibido por los fabricantes y anula las garantías en el mismo instante en que se hace, por lo que se puede considerar un hackeo del dispositivo. Existen herramientas de rooteo que permiten realizar estas acciones (ver punto 7.2.4).

Es posible también ejercer técnicas de spoofig y de hijacking con un terminal con Android. Estas técnicas se explicarán en el siguiente apartado de este documento, el punto 8.

iOS

El caso de iOS es diferente al de Andoird. Mientras el primero se basa en una filosofía de código parcialmente abierto, iOS es más cerrado en ese aspecto y por defecto tiene su kernel cifrado para que no sea posible manipularlo ni editarlo. Es por ello que los investigadores y crackers han de buscar vulnerabilidades en el sistema con tal de poder instalar un kernel modificado y obtener privilegios root. Por lo tanto, romper la seguridad base del sistema operativo es una tarea compleja llamada jailbraking. Una vez descubierta la vulnerabilidad, se procede como ya se ha escrito, creando un exploit y difundiéndolo para que los programadores han sus propias versiones de iOS añadiendo funciones que la compañía no permite o que cobran por ello (liberación del móvil de una compañía, instalaciones de apps de terceros, etc.). Algunas herramientas descritas para dicho fin se numeran en el punto 7.2.4.

7.2.3 Herramientas

Android

Algunas herramientas útiles para Android son [1]:
Rooting: tenemos SuperOneClick, SuperBoot, One click root, o Kingo Android ROOT.
Troyanos y RAT's: ZitMo, FakeToken, TRAMP.A, Obad, o Fakedefender, AndroRAT, Dendroid.
Sniffers y hijackers y spoofers:Packet Sniffer, Fing, NetX, tPacketCapture, Android PCAP, FaceNiff, DroidSHeep o Network Spoofe.

Herramientas de seguridad [3]: TrustGo Mobile Security, Sophos Mobile Security, DroidSheep Guard, Remote wipe y en general, la mayoría de antivirus más utilizados para los sitemas fijos, tienen su símil en plataformas móviles.
Rastreo de dispositivo: son las herramientas que se dedican a tener localizado nuestro dispositivo. Algunas son: FinfMyPhone, Prey Anti-thef, Where is my Droid, iHound, AndroidLost.com.

iOS
Jailbreaking [22]: Cydia, Pangu, Redsn0w, Absinthe, evasi0n7, GeekSn0w, Sn0wbreeze o PwnageTool.
Ratreo: iHound, FindMyPhone, iLocalis o GadgetTrak iOS Security. En las últimas versiones del S.O. ya se incluyen herramientas de rastreo por defecto.

Herramientas de pentesting
Existen algunas herramientas especiales de pentesting de ámbito general (toolkits) como son zAnti, dSploit, X-Scan o Hacode [1].

7.2.4 Contramedidas de seguridad móvil

Dicho todo esto, existen numerosas medidas de seguridad que debemos tener en cuenta [1, 20].

- Evitar instalar software de fuentes desconocidas, habilitar la subida de archivos automática a la red, incluidas las redes sociales y ejecutar demasiadas a la vez
- Asegurarse de que el bluetooth está desactivado y minimizar estas conexiones. Si no se pueden, nunca realizarse en espacios públicos, sino en entornos controlados. De igual modo pasa con las redes wifi. Jamás conectarse a una red wifi abierta o de origen desconocido. También es recomendable no conectarse a redes wifi y bluetooth simultáneamente.
- Nunca acceder a links procedentes de SMS o de clientes de mensajería instantánea. También deshabilitar la autodescarga de imágenes/videos y la previsualización. Tampoco contestar a números desconocidos y menos si piden algo con urgencia.
- Deshabilitar la geolocalización de móvil y limitar esta disposición de información a las aplicaciones.
- Deshabilitar la red wifi en caso de no utilizarse (por ejemplo cuando utilizamos datos móviles) para evitar rastreos.
- Mantener actualizados los sistemas operativos. Si el fabricante decide no mantener el soporte por más tiempo, considerar cambiar el dispositivo (aparecerán vulnerabilidades que nunca se corregirán).
- Bloquear los intentos reiterados de inicio de sesión fallidos, así como utilizar contraseñas/patrones de seguridad fuertes.
- Utilizar MDM para monitorear el tráfico de datos que existe en el dispositivo.
- Encriptar los datos del dispositivo, siempre que se pueda, así como utilizar aplicaciones (sobre todo de comunicación) que habiliten la encriptación end-to-end.
- Instalar aplicaciones como Remote Wipe o FindMyPhone para eliminar los datos o bloquear el dispositivo en caso de robo. También dar de baja temporal el número de teléfono en la compañía en caso de robo/perdida.
- Nunca prestar el dispositivo a ningún desconocido para evita instalaciones de malware encubiertas o robo de información.
- Mantener al mínimo los datos almacenados, realizando copias de seguridad periódicas y encriptadas, eliminando los datos ya guardados. Los backups es mejor tenerlos localizados en dispositivos físicos y no en la nube.
- Instalar herramientas antivirus y antimalware para prevenir infecciones y rooteos.
- No realizar jaibreaking/rooteo de los dispositivos.
- Deshabilitar herramientas de diagnóstico automáticas.

- Deshabilitar las notificaciones mientras el dispositivo está bloqueado.
- Utilizar una VPN siempre que sea posible para protegerse de posibles intrusos en la red.
- Los administradores deberán hacer públicas unas políticas de seguridad básicas a los usuarios. También deberán limitar el tiempo de sesión de los usuarios una vez iniciadas las sesiones.

7.3 Ingeniería inversa

Tiene un peso realmente importante en la creación de exploits, dado que consiste básicamente en realizar el proceso inverso de creación de un software o hardware hasta desmenuzarlo por completo o en parte. Hecho esto, empezará la tarea de comprensión total y absoluta de la información obtenida, librerías y procesos hasta conseguir algún proceso susceptible de mejora o debilidad que se pueda explotar. Otras motivaciones ilícitas pueden ser la apropiación indebida de la tecnología (para su venta por ejemplo) o legales como la mejora de estos (en caso de una auditoria, si se tienen permisos, sería completamente legal). El primer problema surge al querer descomponer los archivos binarios de un programa. Estos comúnmente se nos mostrarán como una serie de datos en lenguaje de bajo nivel, desestructurado e ininteligible (al principio). Eso si se consigue hacer. También se puede hacer en hardware, aplicando lo acabo de comentar a circuitería, chips y memorias [7, 22].

Es por todo esto que el trabajo de ingeniería inversa requiere de un especial y profundo conocimientos de los procesos del sistema así como del lenguaje de programación y compilación utilizadas. Los tres tipos principales de análisis de I.E son: I.E de datos, donde se busca analizar las bases de datos y las estructuras internas; la I.E. de lógicas de procesos, dónde se busca comprender los procesos internos subyacentes; y la I.E de interfaces de usuario, dónde se intenta desmenuzar estas interfaces para comprender los métodos de logeo, cotejo de credenciales, etc.

Algunas herramientas muy utilizadas en estos procesos son los llamados depuradores (debuggers) como OllyDBG o WinDBG, que s eutilizan paraejecutar controladamente un programa, observando la salida de éste en cada momento y así poder deducir los procesos subyacentes. También se utilizan inyectores de fallos, las cuales introducen controladamente entradas complejas y/o no validas al programa buscando el momento en que se produce el fallo, para luego estudiarlo con detenimiento. También por último hay que mencionar los desensambladores y descompliladores, que son programas que realizan el proceso de conversión de código máquina a lenguaje ensamblador, y de lenguaje de alto nivel a lenguaje de bajo nivel. Algunos ejemplos son IDA Pro, Ciadis, Bastard Disassembler, DCC Decompiler o Reverse Engineering Compler (REC) [5, 22].

De nuevo, nos encontramos ante un tema excesivamente amplio y siempre complicado, del cual se podría escribir un libro aparte. Aquí nos centramos en definir y hacer constarlo constar.

CAPÍTULO VIII

8. Seguridad en redes

En este apartado se va a proceder a analizar las principales maneras que un atacante puede interferir en las redes de sistemas. No es el objetivo describir las tipologías de red ni diversas configuraciones existentes, dado que esos conocimientos se dan por sabidos.

Principalmente tenemos varios métodos generales; Spoofing, Hijacking, Sniffing, cuyas técnicas se complementan unas entre otras, y otros tipos de ataques como ataques DoS y cracking de conexiones Wireless, las cuales se van a desglosar a continuación. Al final se hará una especial mención a las redes IPv6 dado que, aunque serán comunes en el futuro, aún siguen siendo desconocidas y semi-implementadas en muchos sistemas [1, 3, 5, 11].

8.1 Spoofing

Esta técnica consiste en establecer una comunicación con un sistema suplantando la identidad de alguna de las máquinas falseando los datos necesarios. Por ejemplo se puede falsearla dirección MAC de una tarjeta de red, suplantando una dirección de confianza. Para realizar estos ataques se emplean diferentes técnicas, dependiendo de cada aspecto técnico que se desea suplantar. Comúnmente la manipulación de paquetes e información de una red se le nombra como envenenamiento de redes (poisoning networks).

8.1.1 Tipos de ataques

Los diferentes tipos de spoofing son los siguientes:

MAC Spoofing: esta técnica es una de las más utilizadas y consiste simple y llanamente en cambiar la dirección MAC de una máquina (normalmente la del atacante) por otra con el fin cambiar la identidad y evitar el descubrimiento. Las direcciones MAC vienen fijas en cada dispositivo con el fin de mantener el control y el orden legal de cada dispositivo, pero con estas técnicas se le hace creer al sistema operativo que se ha conectado una tarjeta de red con esa nueva MAC.

IP Spoofing: consiste en reemplazar la IP de origen por otra que interesa suplantar. Para ello se escogen los paquetes TCP/UDP/ICMP/etc. y se manipula la información para que los paquetes vayan dirigidos a un destino falso. Esto ha sido un problema grave en los cuales los proveedores de internet se han volcado para erradicar y actualmente es muy complicado tener éxito en estos ataques. No obstante, siempre puede existir alguna nueva vulnerabilidad que pueda provocar la explotación de esta técnica nuevamente.

ARP Spoofing: estos ataques manipulan los paquetes ARP (recordar que son los paquetes que se encargan de enlazar la información entre IP y direcciones físicas MAC a través de conexiones Ethernet en la capa de enlace) para confundir las conexiones y que estos vayan a parar a otra máquina. Los ataques se aprovechan de que las relaciones ARP se guardan en unas tablas con memoria caché dinámicas que pueden ser manipuladas. Como se deduce, estos ataques requieren conexión Ethernet, es decir, estar conectados directa y físicamente a la red a auditar/atacar. La manera de evitar estos ataques es estableciendo direcciones IP estáticas y tablas ARP fijas, o en redes grandes, configurándolas con DHCP, que mantiene un registro de direcciones MAC válidas.

DNS Spoofing: mediante una vulnerabilidad conocida como Pharming de los servidores DNS, se manipula la información de las tablas para redirigir al usuario a una dirección de internet falsa, para por ejemplo, conecte con un servidor infectado con malware. En otras palabras, se intercambia la IP asociada a un dominio para que las consultas DNS devuelvan una dirección ilegitima.

SMTP Spoofing: siguiendo la filosofía del spoofing, ahora se manipulan los paquetes de los protocolos de emails para que el emisor sea falseado. De este modo el receptor del email verá un email aparentemente legítimo pero en realidad es una suplantación de identidad. La manera de evitarlo es usando firmas que cifren la información SMTP con PGP, configurando los firewalls para que confíen sólo en un tipo de servidor SMTP o enseñado a los usuarios a desconfiar si encuentran aspectos extraños en los contenidos de los emails.

Web Spoofing: esta técnica consiste en utilizar una web falsa a modo de proxy para monitorear todo el tráfico de internet de la víctima. Mediante el falseo de dicha página, se deriva a la víctima a las webs lícitas extrayendo toda la información que el atacante necesita. Actualmente es un ataque muy efectivo que es muy difícil de detectar.

Blue MAC Spoofing: este ataque no es tan semejante a los anteriores pero es interesante dado que su objetivo es reemplazar un dispositivo bluetooth enlazado con otro. Si se tienen los datos y se consigue, se puede utilizar para enviar RATs y obtener el manejo total del terminal, ya sea fijo o móvil.

GPS Spoofing: este ataque consisten en crear una señal manipulada ligeramente más intensa que la de un satélite GPS, semejante a las originales (tiempos de respuesta, ubicaciones cercanas, etc.) para que el dispositivo receptor indique una ruta o posición diferente a la que debería. El problema es que se debe saber información relacionada con la proximidad real de la ubicación con tal de emular efectivamente los detalles de la conexión GPS, por ejemplo el tiempo de respuesta y la potencia de señal adecuada.

8.1.2 Herramientas

Ataques spoofing: Websploit, Ethercap, torsocks, DNSChef, burpsuite, Netcut o Reaver.

Detección de Spoofing: Arpwatch

8.2 Hijacking

El hijacking o secuestro de sesión consiste en aplicar un conjunto de técnicas y herramientas para manipular, capturar y alterar el tráfico de red. Se suele complementar a las técnicas de spoofing, dado que, una vez se ha redireccionado y enmascarado una conexión, se manipulan los datos que fluyen a través de la red envenenada para el provecho del atacante.

8.2.1 Tipos de taques

El objetivo primordial de un secuestro de sesión es obtener un token válido de la comunicación entre las máquinas. Para ello se utilizan las siguientes técnicas principalmente:

Predicción de Token: este ataque consiste en predecir la variable ID de la sesión a partir de los patrones que se obtienen del análisis del tráfico de red. Estos análisis se llevan a cabo con herramientas de criptoanálisis, cuyos métodos consisten en capturar gran cantidad de tokens por los cuales se pueda deducir y prever las siguientes variables ID.

Ataque MITM: man in the middle, aquí el atacante se interpone entre dos dispositivos conectados (ya sea por redes físicas o inalámbricas) en modo promiscuo sin que ninguno de los interlocutores sepa que está ahí, capturando todo el tráfico (ver apartado 8.3). El objetivo suele ser manipular los paquetes TCP de las conexiones infectar las máquina con malware o extraer información mediante el secuestro de las sesiones.

Ataque MITM en navegador web: en este caso, el atacante debe previamente infectar la máquina de la víctima con algún tipo de malware tipo troyano o RAT especial. El objetivo de este malware será realizar un MITM entre el navegador y el resto de internet directamente, almacenando todo el tráfico de red que interese para su posterior extracción de información que busca el atacante. Cuentas corrientes, visitas de webs comprometidas, emails o chats se

verán comprometidos en estos secuestros de navegador. Ataques conocidos como el SSLstrip utilizan estos principios para evadir en este caso, conexiones seguras HTTPS.

Ataques XSS (cross-site scripting): se pueden crear links con códigos maliciosos en javasscript que serán enviados a las víctimas mediante cualquier técnica ya comentada. Dada la naturalidad de la ejecución de este código, al hacer clic se ejecutará todo un proceso que dará al atacante la información necesaria para establecer un secuestro de sesión (de nuevo el token ID), además de la posibilidad de la infección de la máquina con cualquier otro malware.

Ataque de petición falsificada cross-site (XSRF): estos ataques ocurren cuando un atacante aloja un link o una imagen infectada en un servidor web que es visitado por un usuario que mantiene una conexión legítima con un segundo servidor. El atacante obtendrá el ID del usuario cuando éste interactúa de algún modo con el link/archivo infectado.

Ataque de replicación de sesión: En estos casos, el atacante puede tener la habilidad y la casualidad de capturar el token de autenticación en el momento del intercambio cliente-servidor con lo que puede reproducir el acceso con dicho token para poder tener acceso, es decir, copiar la conexión lícita para obtener un acceso no autorizado.

Todas estas técnicas acabadas de nombrar se definen en la capa de aplicación. Pero no todo los secuestros de sesión se explotan en esta capa. En la capa de red, también se presentan algunas técnicas muy utilizadas que también se describen a continuación.

Secuestro Blind: este ataque ocurre cuando el atacante efectúa un MITM entre la víctima y la otra máquina y es capaz de capturar mediante sniffing los paquetes de peticiones de autenticación ACK, SYN e ISN. EL atacante se anticipa a la respuesta de confirmación de la sesión y establece la sesión ilegítima.

Secuestro UDP/TCP – IP: en estos casos, el atacante, mediante técnicas de sniffing, envía los paquetes capturados y manipulados, suplantando la IP de la víctima, hacia el servidor con el número de secuencia de paquete predicho. Esto provoca que la víctima acabe recibiendo paquetes con los números de secuencia incorrectos. De esta manera el atacante se hace con el control de la comunicación sin que la víctima, ni en este caso el servidor, se den cuenta.

Secuestro RST: este método vuelve a utilizar una combinación de técnicas de spoofing como en el secuestro UDP/TCP pero en este caso con los paquetes RST. Mientras la víctima-servidor se comunican con normalidad, el atacante suplanta la dirección del servidor y le responde a la víctima con un RST incorrecto para que la conexión de la se reinicie.

8.2.2 Herramientas

Herramientas hijancking: Colasoft's Packet builder, tcpdump, Zaproxy, Burp Suite, JHijack, Surf Jack, Ettercap, PerJack, sslStrip, Cookie Cadger.

Para móviles tenemos DroidSheep o DroidSniff.

8.2.3 Contramedidas

- Utilizar siempre el protocolo SSH para crear comunicaciones seguras. Además deben ser las mínimas necesarias para minimizar la exposición al riesgo de hijacking.
- Existe un problema de concepto generalizado al no acostumbrarse a hacer logout de los servicios. Es indispensable para finalizar las sesiones correctamente y evitar problemas.
- Utilizar siempre conexiones a webs HTTPS y cifrar siempre que sea posible en contenido de las comunicaciones. También resulta muy útil utilizar VPNs
- Configurar las sesiones con un timeout de modo que finalicen automáticamente.
- Usar una cadena aleatoria de sesión puede ayudar a dificultar la premonición de las tokens.
- Utilizar siempre IDS y firewalls debidamente configurados que analicen los paquetes ARP. También establecer reglas para delimitar las IP entrantes y salientes.

- Las fortificaciones más comunes deberían ser: sFTP en vez de FTP, HTTPS en vez de HTTP, IPSec sobre IP, SMB signing sobre SMB o OpenSSH o SHH en vez de telnet o rlogin. Además siempre utilizar switches en lugar de hubs.

8.3 Sniffing

Para poder llevar a cabo las tareas de spoofing e Hijacking, comúnmente se suelen utilizar técnicas de sniffing. Estas técnicas consisten en capturar todo el tráfico de red que fluye entre dos puntos. Mediante técnicas de man in the middle, el atacante se interpone entre dos dispositivos conectados (ya sea por redes físicas o inalámbricas) en modo promiscuo sin que ningúno de los interlocutores sepa que está ahí. Por consiguiente, toda la información intercambiada será capturada para analizarla, manipularla, reenviarla ya manipulada extraer información sensible (contraseñas, usernames, IPs y webs visitadas, servicios usados, etc.)

8.3.1 Herramientas y contramedidas

La herramienta estrella en sniffing es Wireshark. Esta herramienta es capaz de interceptar todos los paquetes y guardarlos para luego poder realizar búsquedas de paquetes. Después permite desglosarlos para leer internamente cada contenido en detalle. Toda una herramienta increíblemente potente y open source. También tenemos IPgrab, ntopng, EtherApe, dnsiff, Big-Mother o Ace Password Sniffer.Para móviles tenemos algunas herramientas como Wi.cap.network Sniffer Pro o FaceNiff. En el ámbito de detección se puede usar Nmap o Promqry en modos promiscuos.

Para protegerse del sniffing se pueden tomar algunas medidas preventivas pero será un frente difícil de erradicar. Las medidas son las mismas que se han descrito en el apartado 8.2.3 contra el hijacking.

8.4 Denegaciones de servicio

Los ataques de denegación de servicio (DoS) son actualmente uno de los frentes más activos de internet [10, 22]. Se trata de enviar datos innecesarios y masivos a un servidor para sobrecargarlo y conseguir entorpecer alguna de las propiedades ACID. Es más, si se colapsa el servidor completamente quedando inutilizado durante el mayor tiempo posible, el ataque DoS habrá tenido el mejor final posible.

No obstante, actualmente, los servicios web más usados no están centralizados en un único servidor, sino que están distribuidos, por lo que un solo sistema no tiene la capacidad suficiente como para provocar daños verdades. Es por ello que el concepto de DoS se amplía a DDoS (Distributed Denial of Service) y se actúa conjuntamente con muchos sistemas distribuidos atacantes para provocar los daños. Para ello, se utilizan botnets, equipos infectos en modo zombie que reaccionan conjuntamente enviando datos cuando se activa el ataque (flooding es el nombre que comúnmente se utiliza para indicar en envío masivo de datos).

Actualmente los ataques DDoS son un grave problema para webs con un tráfico de datos importante, o webs gubernamentales. A menudo es el tipo de ataque preferido por hacktivistas provocando serias pérdidas en empresas multinacionales o gobiernos.

Principalmente los objetivos y técnicas de saturación de servicios son:

1. Envío de paquetes fragmentados.
2. Envío masivo de información inútil.
3. Envío de paquetes SYN con direcciones spoofeadas (SYN flood).
4. Compartición de datos Peer to peer masivos.
5. Phlashing (ataques que de algún modo causan daños irreparables en el sistema).Por ejemplo se envían actualizaciones dañinas a los dispositivos.

8.4.1 Herramientas

Algunas de estas herramientas son [1]**:**

Herramientas de ataques DoS/DDoS: existen numerosas herramientas para realizar ataques DoS por ejemplo tenemos Pandora DDoS Bot Toolkit, Dereil, HOIC, DoSHTTP, BanglaDos, Tor's Hammer, Anonymoues-DoS, HULK; DDOSIM o PyLoris. En móviles tenemos AnDOSid o LOIC.

Herramientas de protección: FortGuard Anti-DDoS Firewall, Incapsula, SDL Regex Fuzzer, FortiDDoS, NetFlow Analyser, NetScaler o Anti DDoS guardian, entre otros.

8.4.2 Contramedidas

Describiendo algunas de las buenas prácticas tenemos [10]:

- Una manera efectiva es planificar un equipo de absorción de tráfico innecesario, desviado los paquetes detectados como ataque a estos.
- Parar los servicios críticos cuando no se utilizan y tenerlos monitoreados.
- Si se detecta un ataque, desactivar el/los servicio/s afectado/s para evitar daños. Bloquear todas las IP de las cuales proceda el tráfico permanentemente.
- Utilizar IDS y firewalls debidamente configurados para la detección de ataques. Recordar que se pueden configurar normas sobre intentos de conexión, para bloquear los intentos reiterados.
- Instalar Honeypots y DZM siempre que sea posible. Cisco tiene dispositivos específicos para paliar ataques DDoS que interceptan todos los paquetes TCP.
- Utilizar como se ha comentado VPN, IPSec y protocolos de cifrado fuertes como WPA2, algoritmos RSA de 256 bits o más y utilizar switches en vez de hubs.
- Mantener actualizados los kernel de los servidores ayuda a detectar los ataques evitar infecciones.

8.5 Cracking Wireless

En las redes inalámbricas se nos abre un abanico de posibilidades dada la movilidad de estas. Por esta razón, cabe entender bien los procesos de intercomunicación de dispositivos para entender los ataques y las amenazas existentes [1, 7, 11, 12].

Definición básica sobre la cabecera de una trama Wi-Fi (24 bytes)

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes
Control de trama	**ID/duración**	**Dirección estación receptora**	**Dirección estación transmisora**	**Dirección de origen/ destinación**	**Control de secuencia**

Figura 8.1: Cabecera de una trama Wi-Fi

Métodos de cifrado de las redes inalámbricas Wi-Fi (clave compartida)
Cifrado WEP: algoritmo de cifrado simétrico de flujo que utiliza el algoritmo RC4 caracterizado por la poca complejidad de cálculos necesarios. Normalmente se utilizan claves de 64 o 128 bits pero puede llegar hasta 2048 bits. Los pasos que sigue la estación emisora para la protección de las tramas son los siguientes:

1. Genera una cadena de 24 bits para utilizar como un vector que se utilizará para hacer cálculos
2. Concatena esa cadena con la clave original WEP para formar la clave que utilizará RC4
3. Calcula el código de control de errores (CRC) y obtiene el ICV (integrity check value)
4. Concatena la trama inicial con la ICV y la procesa con el algoritmo RC4
5. Monta los datos (ver figura 8.2) y los envía juntamente con la cabecera general.

6. El receptor descifra la trama invirtiendo los pasos.

3 bytes	1 byte	1-2304 bytes	4 bytes
Vector (IV)	**Id clave**	**Datos para proteger**	**ICV**
Datos cifrados			

Figura 8.2: Trama enviada.

Cifrado WPA: ideado para corregir las deficiencias del cifrado WEP. La idea consiste en utilizar un servidor de autenticación (RADIUS por ejemplo) que identifica a los usuarios de la red y le otorga los privilegios. En redes pequeñas (caseras por ejemplo) se usa una clave pre-compartida (PSK) uniforme para todos que se negocia una vez se establece la conexión con el dispositivo que quiere añadirse a la red. De este modo se elimina la necesidad de un servidor de autenticación. El tipo de cifrado que utiliza es TKIP **T**emporal **K**ey **I**ntegrity **P**rotocol que genera las claves algo diferentes a lo que hace WEP, generando la clave temporal que irá cambiando periódicamente.

Cifrado WPA2: nace como evolución del WPA para tapar las deficiencias encontradas. Ahora se utilizaría el algoritmo AES para la generación de las claves comparativas y el modo WPA-PSK pasa a llamarse WPA2-Personal o WPA2-Enterprise. En la figura 8.3 nos podemos hacer una idea de las diferencias principales entre los cifrados de las conexiones.

	Algoritmo de encriptación	Tamaño del vector IV	Peso de la clave de encriptación	Mecanismo de control de integridad
WEP	**RC4**	**3 bytes**	**5/13 bytes**	**CRC-32**
WPA	**RC4 o TKIP**	**6 bytes**	**16 bytes**	**Algoritmo de Michael y CRC-32**
WPA2	**AES-CCMP**	**6 bytes**	**16 bytes**	**CBC-MAC**

Figura 8.3: Comparación entre algoritmos de encriptación inalámbricos.

8.5.1 Amenazas y técnicas

A continuación se describen las principales técnicas que emplean los black hats para romper la seguridad de las redes inalámbricas.

WEP: son numerosos los ataques que ha sufrido este sistema y hoy en día es la protección más insegura en cuanto a conexiones inalámbricas se refiere:
Inyección de tramas: consiste en capturar una trama WEP e inyectarla repetidas veces hasta que el receptor da por buena la conexión, o en cuyo caso, substituir las direcciones de estación emisora y receptora de forma que siga el procedimiento hasta dar la conexión por buena.

Falsificación de la autentificación: simplemente capturando las tramas transmitidas durante el handshake de la conexión, el atacante podrá establecerse como un huésped más en la red.

Ataque chop-chop: consiste en capturar una trama WEP y descifrarle la última n bytes cifrados, enviando a la estación receptora una media de 128xn tramas. A medida de que el ICV de una trama vaya siendo dado por bueno, se irá descifrando el keystream.

Ataque de fragmentación: una vez que se sepan los n bytes de un keystream se podrán obtener 60 bytes de otro enviando 16 tramas fragmentadas al receptor.

Ataques contra vulnerabilidades del algoritmo RC4: existen tres vulnerabilidades conocidas referente exclusivamente al modo de operar del algoritmo; como el FMS, donde sabiendo el primer byte del keystream puede acceder a la clave madre, KoreK que aprovecha el conocimiento de los dos primeros bytes del keystream para averiguar el valor de la clave raiz o PTW donde se hace uso del conocimiento de los bytes 3 al 15 para hacer lo mismo que los acabados de mencionar.

WPA/WPA2: la vulnerabilidad conocida más utilizada para crackear redes WPA/WPA2 es mediante la captura del handshake inicial. Para entenderlo hace falta explicar el contexto. Al

principio de la conexión, durante el intercambio de claves, se negocia la política de seguridad a seguir, el cliente y el AP generan una clave llamada pairwise master key (PMK) y también una clave única para cada proceso de autenticación de cada cliente llamada PTK, que básicamente a su vez, se genera por medio de 4 números aleatorios, 2 por el cliente y dos por el AP. Conjuntamente se conoce el proceso como 4-way-handshake. Si se es capaz de capturar el tráfico de ese handshake (concretamente los números aleatorios), junto con la dirección MAC y el ESSID, será capaz de obtener el PSK y conectarse a la red. En caso sólo de que el cliente ya esté conectado a la red, simplemente se le hará un ataque DoS para forzar la desconexión de la red y así tener la oportunidad de capturar el handshake.

8.5.2 Herramientas

Cracking WEP/WPA/WPA2: la suite Aircrack (que incluye airmon-ng para monitorear y airodimp-ng para recolectar datos) y Reaver son las dos herramientas por excelencia en estas labores. También tenemos otras ya mencionadas como Cain & Abel o KisMAC, Elcomsoft Wireless Security Auditor, WebDecrypt, Wesside-ng, WebAttack y Wifite. En móviles tenemos algunas herramientas interesantes como Penetrate en su versión Pro, WiHack y Blacktrack Simulator.

Monitoreo de red: para examinar redes tenemos NetworkMAnager, WaveNode, SigMOn, RF Monitor o NetworkControl

8.5.3 Contramedidas

- Algunas de las recomendaciones más importantes son:
- Configurar adecuadamente las WLAN. Es to implica cambiar el nombre por defecto de las redes (para evitar dar información) y el password y usuario por defecto de configuración del router. También es importante desactivar el login inalámbrico de éste y activar la encriptación WPA2.
- Instalar firewalls siempre que sea posible
- Ocultar los SSID de los puntos de acceso si es posible, o desactivarlos si es necesario. Si no se puede evitar, no poner nunca el nombre de la empresa, sino algún nombre conocido sólo por los trabajadores.
- Mantener actualizados los firmwares de los routers, switches, etc.
- Nunca conectarse a una red wifi la cual no conocemos el origen. Que esté abierta puede indicar que sea un punto de acceso trampa (sobre todo si estamos en lugares muy concurridos).
- Limitar el alcance innecesario de las redes wifi, es decir, acotar la red inalámbrica exclusivamente al área que se necesita, por ejemplo recolocando el punto de acceso o poniendo antenas menos potentes.
- Desconectar la red cuando no se necesite (por ejemplo en vacaciones de empresa)
- Si es posible, establecer una capa adicional de protección como IPSEC.
- Un servidor centralizado que controle las autenticaciones a la red también es muy efectivo contra ataques.

8.6 IPv6

Se entiende que los conceptos explicados en la gran mayoría de ocasiones serán bajo IPv4 pero se pueden aplicar perfectamente a la configuración IPv6 por lo que no se debe olvidar. Dado que no es tan utilizada, a menudo es una configuración que se pasa por alto pese estar activa y configurada. Por ello debe ser comprendida para saber que nuevos frentes se abren y que bondades ofrece [16].

En la actualidad la gran mayoría de tráfico que se genera es en IPv4 donde las direcciones son, como es bien sabido, tienen un formato estilo 220.168.124.125, es decir, se dispone de 32 bits para asignar a las direcciones IP públicas, unas 4300 millones de combinaciones diferentes

aproximadamente. Estas direcciones públicas, mediante configuraciones NAT se transforman en direcciones de red internas dado que no es posible asignar una IP pública a cada aparato conectado a internet (simplemente no estaba previsto la expansión de dispositivos conectados a internet actual, y no son suficientes). Es por ello que nace el protocolo IPv6, donde ahora se utilizan 128 bits para las direcciones (en usuarios domésticos 64 bits para direcciones internas y 64 para información externa) y además se indican en formato hexadecimal permitiéndose la omisión de 0, por ejemplo 2DB8::AC10:H2345: … con una capacidad aproximada de $3'38 \cdot 10^{38}$ combinaciones posibles.

Dicho esto, en este protocolo es capaz de conectar directamente mediante una dirección única, dos sistemas sin pasar por la NAT interna de la red, cosa que auqne trae bondades, abre nuevos focos de ataques. Actualmente el protocolo está por explorar, quedan muchas herramientas que crear y su implantación está en camino, pero conviene tener en cuenta, en el apartado de hacking de redes, que tipos de ataques están ya descubiertos.

Para conseguir compaginar abos protocolos, lo que se hace es apilar la información IPv6 sobre IPv4 en la capa de red (ver figura 8.4), y se envía conjuntamente de manera que los enrutadores detectan los paquetes que contienen dicho protocolo y lo envía a su correspondiente destinatario.

5. Aplicación	NFS, NIS, DNS, LDAP, telnet, ftp, rlogin, rsh, rcp, RIP, RDISC, SNMP, etc.
4. Transporte	TCP, UDP,
3. Red	IPv4, **IPv6**, ARP, ICMP
2. Enlace	PPP, IEEE 802.2
1. Física	Ethernet (IEEE 802.3), Token Ring, RS-232, FDDI, etc

Figura 8.4: Capa utilizada por IPv6 en modelo TCP-IP

8.6.1 Tipos de ataques

Dada la peculiaridad de que todos los dispositivos pueden conectarse entre ellos directamente, los ISP normalmente detectan intentos de spoofing y los bloquean [11, 16].

SYN-Flooding: dada la conectividad tan extensa, los ISP no pueden controlar todas las combinaciones posibles de direcciones y por lo tanto, es posible hacer ataques DoS de tipo SYN flood a maquinas con direccionamiento IPV6, con paquetes con direcciones aparentemente spoofeadas (realmente no es un spoofing dado que los 64 bits que indican la red del propietario no se cambian ni se falsean, sino que se utilizan el resto de bits que te da el ISP para crear los paquetes). Con herramientas como Scapy, implementada en Python, es posible crear paquetes a medida con las IPv6 que deseemos y realizar dichos ataques. Con la ayuda además de Nmap podemos realizar escáneres de puertos abiertos para dirigir los ataques.

Escaneo TCP: si hacemos un escaneo de red en busca de puertos abiertos con una ipv6 diferente cada vez, obtendremos los resultados deseados sin que el firewall o IDS detecte que se está haciendo un escaneo.

Ataque NDP exhaust: esta técnica consiste en saturar, por ejemplo un router, enviando paquetes aleatorios dentro del rango de 64 bits de direcciones asignadas a dicho router. Este envío de datos provocará efectos similares a una ataque DoS en la red interna, dado que el router esperará un pequeño tiempo a que algún dispositivo con esa dirección le conteste.

Neigbor Spoofing MITM: el protocolo IPv6 no utiliza el protocolo ARP o RARP para encotrar vecinos, sino que se basa en otro llamado NDP (Neighbor Discovery Protocol) basado en paqeutes ICMPv6. Es posible enviar mensajes a los equipos de las víctimas, intercambiando las ipv6 de cada uno respectivamente, e introduciendo la dirección MAC del atacante. De este modo se realiza un MITM dentro de la red entre los equipos.

Vulnerabilidades en las adaptaciones: hay que comentar que todos los firewalls, IDS, sistemas operativos que dependan de alguna manera del protocolo IPv4, deberán adaptarse al nuevo protocolo. Decir que es muy posible que muchos protocolos de seguridad y métodos actuales ni siquiera planten la posibilidad de IPv6 por lo que están en riesgo.

8.7 Evasión de IDS, firewalls y detección de honeypots

A la hora de llevar a cabo los ataques, es obvio que no será un camino de rosas. Nos encontraremos en la mayoría de casos que las redes están protegidas por Firewalls, IDS y Honeypots a modo de trampa [1, 6].
Es por ello que los atacantes han tenido que desarrollar técnicas de detección de estas protecciones para poder evadirlas o engañarlas. Un auditor también deberá probar diferentes técnicas para encontrar debilidades de configuración en estos sistemas para extraer las mejoras necesarias.

8.7.1 Conceptos y soluciones disponibles

Conviene dejar claros algunos conceptos básicos con tal de entender las técnicas de evasión.

Intrusion Detection/Protection Systems o comúnmente llamados IDS/IPS, son algoritmos o programas capaces de analizar todo lo que pasa por su filtro de red, para buscar patrones o acciones susceptibles de ser originadas por un atacante (incluyendo no solo la explotación de éstos, sino los intentos de escanear los sistemas y redes en los procesos de gathering y análisis). Estos dispositivos suelen interponerse entre el nodo más externo de la red a auditar, y la red interna que alberga los dispositivos a proteger. Es posible también que existan varios, con una DMZ[14] en el medio de los dos, para más seguridad. Las configuraciones pueden variar y dependerán del diseñador de red/ administrador de sistemas escoger un método u otro.

Los IDS contienen unas bases de datos con las firmas de los diversos ataques conocidos, las cuales, mediante un sniffer, coteja con lo analizado en la red. Se buscan tramas incompletas, corruptas, intentos de conexión fallidos o reiterados, etc. Existen dos tipos fundamentales, los HIDS (Hosts IDS) y los NIDS (NewtorkIDS). Los primeros se dedican a analizar el rastro que ha quedado en los dispositivos una vez se han hecho los intentos de ataque, los segundos, analizan todo el tráfico entrante/saliente indiscriminadamente (las tarjetas de red que están en ese modo se les dice que están en 'modo promiscuo') en busca de rastros en el propio tráfico. Normalmente, se encuentran en modo pasivo o reactivo, los primeros están analizando lo que pasa y extraen informes para sus posteriores acciones, los segundos reaccionan en tiempo real a los diferentes posibles ataques.

Algunos factores explícitos que indicarán un intento de ataque serán la presencia de archivos/programas nuevos, cambios en los permisos y en los pesos de los archivos, archivos corruptos o perdidos. También serán indicios, repetidos intentos de conexión a diversos puertos/servicios, repetidos logins, información inusual en los logs o directorios nuevos y extraños.

Normalmente los IDS/IPS están integrados con un Firewall

Estos Firewalls compuestos por hardware y/o software, normalmente estarán interpuestos entre la red interna e la externa/pública como Internet. Su función es detectar y analizar los mensajes entrantes y salientes para bloquearlos si es necesario. Para tenerlo claro, existen 4 tipos básicos de firewalls.

[14] DMZ: siglas de Desmilitarized Zone, es una parte especial de red perimetral que permite la conexión de esta con el resto de internet pero no con la red interna. Por el contrario, la red interna solo puede conectarse a la DMZ. De esta forma se mantiene a la red interna aislada de los peligros de Internet y a la vez se pueden dar servicios al exterior desde la DMZ.

Filtradores de paquetes: estos analizan la capa de red (modelo OSI) buscando paquetes sospechosos. Estos paquetes se analizan según unas normas preconfiguradas por el administrador (IP, puerto, protocolo, etc.) para ser bloqueados o aceptados.

CLG (Circuit Level Gateway): estos firewalls actúan en la capa de sesión (modelo OSI) inspeccionando la información que pasa por la puerta de enlace para decidir si aceptan o descartan las sesiones.

ALG (Aplication Level Gateway): ahora se actúa sobre la capa de aplicación (modelo OSI) analizando los paquetes que entran y salen, fijándose en servicios de aplicaciones como telnet, FTP, etc.

Analizadores completos multicapa: estos tipos son firewalls que actúan en las tres capas anteriores siendo unos todoterreno. Las funciones son una combinación de los filtradores de paquetes, CLGs y ALGs.

Ahora queda describir los Honeypots. Estos sistemas son trampas para atacantes, que reúnen 'casualmente' las condiciones propicias para ser hackeados. Son muy permisivos en cuanto comprobaciones y escaneo de redes y están destinados a recolectar información sobre los posibles atacantes (logs habilitados, monitoreo de sistema constante, poca seguridad en el sistema, etc.). Dependiendo de si son completamente hackeables o no, si tienen unos extensísimos servicios implementados o no y si capturan gran cantidad de datos de los ataques o no, hablaremos de honeypots de alta interacción o baja respectivamente.

8.7.2 Técnicas de evasión

Evasión de IDS

Ataque DOS: En muchas ocasiones, el IDS tiene asignada una IP centralizada, la cual gestiona todos los procesos. Si el atacante llega a conocer esta IP, puede crear un ataque dirigido de denegación de servicio. Con este método, el IDS se satura de información (por ejemplo, cuándo el búffer de entrada se sobrepasa, se le llama bufferoverflow o desbordamiento de buffer) dejando pasar el exceso de información libremente. El objetivo es colapsar totalmente el detector de intrusiones, situación que el atacante aprovecharía para realizar las actividades.

Ofuscación: Se puede ofuscar el código malicioso de envío a través de algoritmos y programas dedicados a ello. Por ejemplo, se pueden reconfigurar los paquetes en Unicode para que el IDS dé estos requests como válidos. Otro ejemplo de ofuscación es manipular las firmas de las rutas de los archivos a enviar, enviar algoritmos polimórficos o encriptar los protocolos. En fin siempre será que el IDS no detecte el código malicioso a causa del 'camuflaje' de los paquetes enviados.

Generador de falsos positivos: Si se tiene suficiente información sobre el dispositivo IDS, se pueden llegar a manipular paquetes para que generen falsos positivos en sus detecciones. Al generar una gran cantidad de ellos, el IDS tendrá problemas en diferenciar los falsos positivos de los ataques reales, permitiendo dichos ataques maliciosos en un pequeño porcentaje.

Ataque de fragmentación: Si el IDS no reconstruye la información que le llega (es decir, no normaliza el tráfico) se puede trocear la información enviada para confundir al IDS. En caso de que si lo normalice, se puede crear un retardo exagerado de los paquetes troceados. Los IDS no retiene los paquetes indefinidamente, por lo que tiene estipulado un periodo de tiempo para desecharlos (por ejemplo 20 o 60 segundos).Dado que algunos IDS paran de reensamblar los paquetes en estos intervalos de tiempo y paran de trabajar dada la sesión excesivamente larga, el tráfico malicioso puede infiltrarse de todas maneras pasando desapercibido si se manipulan estos tiempo.

Solapamientos de fragmentos: Esto consiste en manipular el número del orden de los fragmentos enviados, enviando en primer lugar unos legítimos que sean aceptados y luego otros que contengan los mismos números de secuencia que los aceptados pero conteniendo la

información maliciosa. Cabe esperar en algunos casos, que los paquetes viejos queden solapados por los nuevos.

Ataques Time-to-Live: Si el atacante conoce bien la tipología de la red víctima por ejemplo utilizando traceroute, puede por ejemplo realizar ataques que consisten en enviar 3 fragmentos TLL; el primero con un TTL alto y un segundo con un TTL bajo, de modo que el IDS desechará el segundo y aceptará el primero. Luego si el atacante vuelve a enviar un TTL alto (tercer paquete) el IDS reconstruirá los paquetes 1,2 y 3 y los desechará pero enviará el 3. Si luego se envía de nuevo el paquete dos (el que falta para completar) éste se enviará a la víctima. De este modo se le filtran paquetes al IDS y se infecta el sistema.

Paquetes RST inválidos: Seste ataque consiste en enviar paquetes RST con checksums inválidos al IDS para que este piense que la comunicación ha terminado pero el paquete sí que llegará a la víctima. El IDS como es normal, acabará desechando el paquete con el RST erróneo pensando que ya la conexión ha acabado pero no será así, dado que el atacante habrá conseguido enviar paquetes TCP a la víctima.

Flags urgentes: en los protocolos TCP se pueden utilizar flags para marcar paquetes urgentes que pueden ser utilizados para manipular los últimos bytes que estos contienen.

Shells polimórficos: Si utilizamos stubs para enviar shellcodes cifrados, estos darán el efecto de ser polimórficos y serán muy difíciles de detectar por los IDS, dado que cada vez que se envíen, serán diferentes.

Pre/post- conexiones SYN: se pueden enviar paquetes SYN manipulados para despistar a los IDS de modo que las comunicaciones entre el atacante y el detector de intrusiones queden bloqueadas. Esto provocará que el IDS pare de monitorear el tráfico y deje pasar las conexiones maliciosas.

Flooding: Este método consiste en enviar cantidades masivas de tráfico innecesario para provocar lentitud en los procesos de análisis de los IDS. Si éste no analiza el tráfico adecuadamente puede llegar a dejar pasar conexiones maliciosas.

Encriptación: Si el atacante consigue encriptar la conexión entre la máquina víctima y la suya, el IDS nunca detectará nada sospechoso. Es una técnica muy efectiva pero requiere el paso previo.

Evasión de Firewalls

Escaneo de puertos: Al escanear los puertos del firewall abiertos, se pueden llegar a saber que versión de los servicios se tienen. Esto puede dar lugar a el encuentro de nuevas vulnerabilidades de dicho firewall susceptibles de ser explotadas (versiones antiguas, errores de configuración, 0days, etc.)

Firewalking: Esta técnica consiste en enviar al firewall paquetes UDP/ TCP con modificaciones en el TTL, aumentando las posibilidades de que los saltos que haga el paquete provoquen errores de exceso de tránsito. Esto más que evadir, puede ayudar a localizar el firewall y localizar nuevas vulnerabilidades.

Banner grabbing: En ocasiones el análisis de los banners puede dar lugar a un pequeño finguеprinting de firmwares y firewalls, que se conectan con este mediante servidores web, FTP, telnet, etc.

Spoofing: Este ataque se explicará en el siguiente apartado. Consiste principalmente en enmascarar la verdadera identidad del atacante o auditor para realizar tareas maliciosas. Hay varios tipos que se explicarán con detalle en el apartado 8.

Diminutas fragmentaciones: Si el firewall sólo examina las cabeceras de los protocolos en cuestión (por ejemplo UDP), se pueden fragmentar los paquetes hasta tal punto que la

información de los headers queda repartida. En estos casos el firewall solo examinará el primer fragmento dejando pasar al resto.

IPs por URLs: en ocasiones, las reglas y filtros de los firewalls controlan determinadas URL pero el administrador, por despiste, no tiene en cuenta dirección IP asociada a dicha url. Esto se puede aprovechar para realizar un ataque utilizando la IP real en vez de la url. Parece una técnica simple pero puede resultar muy efectiva como en otras ocasiones.

Proxys: Utilizar un proxy puede ayudar a evadir determinados filtros del firewall, por ejemplo discriminando regiones o determinados protocolos. Si realizamos una conexión a través de un proxy adecuado (generalmente son servidores disponibles en Internet) que haga de intermediario, no sólo los atacantes ofuscarán su rastreo sino que podrán acceder al sistema de igual modo.

Tunneling ICMP/ACK/HTTP/SSH: Consiste en camuflar backdoors Shell en forma de paquetes ICMP, ACK o HTTPS para engañar al firewall. Es uno de los métodos más eficaces y utilizados dado que aprovechan una debilidad de los firewall difícil solventar.

MITM: esta técnica que explicará detalladamente en el punto siguiente de este documento (el apartado 8) pero básicamente consiste en interponerse entre la conexión de dos dispositivos sin que estos detecten la presencia intrusa. Si realizamos este ataque entre un usuario legítimo del sistema a auditar y el mismo sistema, podremos observar todo el tráfico que pasa a través de esta conexión.

Detección de Honeypots

La detección de los honeypots es algo más ambigua que la de lso IDS o Firewalls y requiere de algo más de olfato. Por sentido común, un atacante debería sospechar si entra en un sistema o red demasiado fácilmente y obtiene lo que quiere con facilidad. Por ello, los administradores deberán configurarlos de manera que no sean demasiado evidentes. Aún y así, los atacantes pueden realizar escaneos de red buscando si están activos determinados servicios en el sistema que indiquen la posibilidad de un honeypot. Por ejemplo, pueden enviar paquetes legítimos como HTTP, SMTP o IMAP todos sobre SSL.

Algunos puertos también pueden denegar sin razón aparente el proceso de conexión three-way-handshake aunque están activos, cosa que indica la presencia de un Honeypot en un alto porcentaje.

8.7.3 Herramientas

IDS: el más conocido probablemente es Snort que utiliza una serie de reglas para analizar el tráfico. También tenemos otros como TippingPoint, SilverSky, Peek&Spy, OSSEC, SNARE, AIDE, Fortigate, Enterasys, IBM Security Network Intrusion Prevention System o Cisco Intrusion Prevention Systems. En dispositivos móviles también es interesante dada la movilidad es estos. Aquí tenemos algunas herramientas como WIFI Intrusion Detection, Wifi Intruder Detector o Wifi Inspector.

Firewalls: muchos de los routers llevan integrado en su firmware de serie un firewall configurable, al igual que pasa con algunos sistemas operativos y antivirus. Además, existen algunos conocidos como ZoneAlarm, FireWall 2016, Comodo, Ashampoo Firewall o Squared Online Armor o Sonicwall. Para móviles tenemos algunos como Android Firewall o Firewall iP, aFirewall o Mobiwol:NoRoot firewall. Firewalls físicos (hablando de hardware) son fabricados por marcas como Cisco, Alpha shield, TuxGate, Asus o D-link.

Honeypots: aquí tenemos algunos como KFSensor, SPECTER, Argos, LaBrea Tarpit, HoneyBot, WinHoneyd o Kojoney. Para su detección se pueden utulizar herramientas ya usadas en este documento anteriormente como Hping, Nessus o Send-safe Honeypot Hunter.

Evasión IDS/Firewalls: Tenemos Traffic IQ Pro, tcp-over-dns, Freenet, GTunnel, Tomahawk, Proxifier, VPN One click, AckCmd o Snare Agent for Windows. Para la fragmentación de paquetes y ataques DoS se puede utilizar CommView, fping 3, Packet generator, Ostinato o

hping 3. Para creación de conexiones tuneling, tenemos HTTPort, HTThost, Super Network Tunnel, SSH tunneling o Bitvise.

8.7.4 Medidas de seguridad y prevención

Algunas de las medidas de seguridad que se pueden tomar para protegerse de los ataques por las técnicas nombradas son:

- Deshabilitar los puertos de los switchs que seas susceptibles de ser atacados o que no se utilicen.
- Resetear las sesiones TCP (RST) maliciosas o sospechosas.
- Analizar y monitorear periódicamente e insitu el tráfico de red.
- Como siempre tener debidamente actualizado el software.
- Tener instaladas y bien configuradas en la red, IDS y Firewalls. También asegurarse de que los paquetes se están normalizando y reensamblando en el orden correcto.
- Proteger bien y actualizar, no solo los sistemas, sino los dispositivos aparentemente secundarios como routers, switches y módems.

CAPÍTULO IX

9. Seguridad en aplicaciones web y webservers

En este apartado se describirá globalmente todo el sistema de seguridad y ataques sobre servidores y aplicaciones web [1, 8, 10, 12]. Es un tema muy candente dado la exposición de estos a Internet y realmente daría para escribir interminables líneas. Hay que tener en cuenta que los conceptos se irán entrelazando entre ellos y con los apartados anteriores, dada la conglomeración de ideas que intervienen en el intento de ataque o audición de un servidor web; redes, malware, ingeniería social, explotación de sistemas, gathering, y más, convergen de un modo u otro aquí [1, 8, 10, 12].

9.1 Aplicaciones web

En este apartado se va a proceder a explicar los diferentes aspectos de seguridad referentes a las aplicaciones web. Estas aplicaciones son interfaces que se encuentran entre los usuarios y los webservers que suelen moverse entre los dos mundos, ejecutando códigos javascript en la máquina del usuario y gestionando consultas cliente-servidor, etc. Por lo tanto son susceptibles de ataques de manipulación de código y acciones, como inyecciones sql (ver apartado 9.3), explotación de vulnerabilidades XSS, hijacking, spoofing, MITM, etc. Además a partir de la llegada de la nombrada como web 2.0, estos riesgos se han ampliado considerablemente, dado que el objetivo es crear más interactividad y movimiento a las webs tradicionales estáticas (web 1.0).

Para comprender un poco mejor lo que son y representan las aplicaciones web, es interesante describir la arquitectura interna que suelen tener.

En el principio de la cadena tenemos una capa de aplicación para clientes, la cual contiene todo el conglomerado de servicios accesibles desde el navegador de Internet. Aquí se ejecuta código JavaScript, Silverlight, HTML5 o flash (venido a menos/desuso). Estos navegadores alojados en las máquinas o sistemas móviles se comunican con los webservers (se hablará de ellos en detalle en el apartado 9.2), que son los encargados de procesar las consultas, protocolos y comunicaciones. Es decir, esta es la siguiente capa de aplicación, la cual incluye procesos como logins, proceso HTTP/S, filtros de Firewall, caché de los proxys, entre otros aspectos. Es la encargada de filtrar hacia la información sensible, que representa que se aloja en las bases de datos. Pero antes de llegar a la información concreta, normalmente en empresas existe otra capa, llamada capa de negocio. Hay que recordar en estos casos aplicaciones propias empresariales ERPs (CRM, SCR, etc.) que son las que controlan en última instancia el acceso a la base de datos web. Esta capa incluye código de proceso de información (C++, .NET, J2EE, etc.) y procesos de acceso a la base de datos. Después, por último aparece la capa base de datos, esta contiene los servicios y procesos relacionados con la obtención de información, servicios en la nube, etc.

La información y comunicación entre estas capas fluye bidireccionalmente y en todo su conjunto se define como una aplicación web.

9.1.1 Frentes abiertos

Descritas la arquitectura general de una aplicación web, no hace falta mucho para darse cuenta de que los frentes abiertos para encontrar vulnerabilidades que explotar son numerosos. Por cada capa existirán algunos concretos y otros de ámbitos más globales, pero todos importantes aspectos a tener en cuenta.

Podemos esquematizar las amenazas de la siguiente manera:

Entradas inválidas: cuando las entradas no se validan antes de ser procesadas, aparece un flanco de ataque el cual se puede aprovechar para explotar/inyectar código y obtener

información interesante para el atacante. Por ejemplo inyecciones SQL o algunos de los ataques que se describen seguidamente.

Manipulación de parámetros: en ocasiones es posible manipular los parámetros de entrada (por ejemplo al conectarse con un webserver) con el fin de modificar permisos de sesión, editar variables, etc. Por ejemplo, en una url puede aparecer algún parámetro que sea susceptible de modificar manualmente su valor para acceder a un sito o modificar algún campo.

Directorios transversales: podemos encontrar vulnerabilidades asociadas a la mala configuración de los accesos a los directorios. Por ejemplo, en una url vulnerable, un atacante puede introducir manualmente código como ../ y acceder a algunos directorios root o a algunos que no debería tener acceso. También explorando los paquetes de intercomunicación, se puede introducir código malicioso explotando esta vulnerabilidad.

Malas configuraciones o carencias: aquí se han juntado algunas vulnerabilidades menores pero no menos importantes, sobre errores de configuración. Si el atacante descubre errores de configuración, podría adquirir accesos no autorizados, leer directorios/archivos restringidos, etc. Por ejemplo, no cambiar las cuentas por defecto, incluidas las contraseñas y nombres de usuario, es un claro defecto de configuración. Si existe código inseguro a causa de no estar correctamente encriptado, también supone una brecha de seguridad; la información no debe viajar en claro, siempre que sea posible.

Inyecciones de código: este apartado consiste en inyectar códigos, comandos o archivos con un determinado formato en las consultas de una web (consultas a su base de datos) o en las comunicaciones de las API's para aprovecharse de una vulnerabilidad concreta conocida. Existen tres tipos de inyecciones; SQL el cual se hablará con detalle en punto 9.3, Inyección de comandos, los cuales manipulan los códigos de HTML, PHP o Shell para ganar acceso al servidor o infectarlo con malware subiendo archivos, etc.

Uso de campos ocultados: existe la posibilidad de manipular las consultas con campos que a priori están restringidos, ocultando u ofuscando valores de variables, semejante a los ataques mencionados antes sobre manipulación de parámetros.

XSS (Cross-Site Scripting): este es probablemente el ataque más conocido a la vez que más extendido. Consiste explotar vulnerabilidades de webs consideradas 2.0 con el objetivo de inyectar un script malicioso para que los demás usuarios lo visualicen. Esto ocurre porque muchas de estas webs son generadas dinámicamente. Por ejemplo, inyectando un script en PHP dentro del marco HTML, se genera una respuesta errónea al resto de usuarios de esa misma aplicación web, por ejemplo, redirigiéndolos a un servidor infectado.

Los objetivos y medios de estos ataques pueden ser variados, pueden venir de un email malintencionado, buscar el robo de cookies o inyectar un script en un comentario de un blog.

CSRF (Cross-Site Request Forgery): esta derivación del ataque XSS se produce cuando un usuario con una sesión valida en un servicio web legítimo, accede a un servidor infectado con código que es redirigido hacia el servidor legítimo, ofreciendo el acceso no autorizado del servidor al atacante.

Ataques DoS: estos ataques ya mencionados en el apartado 8, pueden tener como objetivo también a las aplicaciones web, dado que ofrecen servicios online. Las vías por las cuales suelen venir en las aplicaciones web suelen ser intentos reiterados de registros o de logins.

Buffers overflows: al igual que se describió en el apartado 7, los buffer overflows parecen un mal interminable y las aplicaciones web no son una excepción. Si el código fuente de la aplicación webs es vulnerable a estos ataques, el black hat podrá ejecutar código, inicializar procesos o escalar privilegios a su antojo.

Envenenamiento de sesiones con cookies: similar al envenenamiento de redes, este ataque consistirá en mantener la sesión habilitada todo el tiempo posible. Modificar los contenidos de las cookies será el principal medio por el que un atacante podrá explotar esta vulnerabilidad.

Se busca mantener un acceso a la aplicación web, apoyándose en una cookie con una sesión válida, modificada maliciosamente. Estas cookies pueden ser robadas mediante técnicas de sniffing, y utilizadas para comunicarse con el webserver.

Ataques captcha: los captcha son acciones demandadas por la aplicación web que requieren la interacción directa del usuario para verificar que no es un proceso automático el que realiza la consulta. Estas validaciones se pueden aprovechar para inyectar código o escalar privilegios.

Redirecciones inválidas: en ataques de phishing por ejemplo, es posible encapsular dentro de la url del webserver, una dirección de un servidor infectado de manera que el usuario acaba accediendo a un lugar que no es el correcto.

Ataques a servicios web: estos servicios suelen basarse en sus propios protocolos XML, por ejemplo WSDL[15], UDDI[16], SOAP[17], etc. A través de las consultas y respuestas XML se pueden obtener datos sensibles de las bases de datos. También se puede envenenar las consultas para generar errores y ejecutar exploits o denegaciones de servicio.

9.1.2 Metodologías

Sabida la estructura y los vectores de ataque que sufren las aplicaciones web, al igual que ocurrirá con las metodologías de los webservers (punto 9.2), se puede confeccionar un sub esquema de auditoría/hacking de estas.

Como es de esperar, la primera tarea es recopilar toda la información específica de la aplicación web a auditar/atacar. Descubrimientos DNS, servicios activos, puertos e información oculta en los códigos y procesos serán los principales objetivos de gathering. Por supuesto todos los métodos explicados en apartados anteriores serán válidos (ingeniería social, explotación de redes, etc.).

Después se suele intentar atacar directamente al servidor web, dado que este es el que aloja a la aplicación web y si tuviese éxito, la aplicación se comprometería más fácilmente. Estos procesos específicos se explican en el apartado siguiente 9.2.

Si el ataque no resultara fructífero, se procedería a analizar los métodos de autenticación, sus esquemas y en general los protocolos que se utilizan para acceder a la aplicación. Los métodos descritos anteriormente en este apartado se pueden aplicar ahora; mensajes de error, usuarios predecibles, explotación de cookies (envenenamiento, robo o replicación de éstas), ataques a passwords, hijacking de sesiones, etc. Cambiar el password suele ser uno de los métodos más probados por los atacantes, en busca de maneras de recuperar una contraseña por la vía en que lo haría un usuario normal, pero aprovechando debilidades específicas muy diversas.

Se puede aquí empezar a predecir qué tipo de nomenclatura se está utilizando para hacer consultas, además de averiguar que motor de base de datos se está utilizando (Mysql, Postegre, etc.). El objetivo es confeccionar ataques específicos de inyección de sql. Esto se explica con detalle en el apartado 9.3

Después de toda la preparación, se procede a realizar los ataques a la aplicación web mediante todos los métodos ya conocidos. No importa si no está descrito en los vectores de ataque, una combinación de conceptos referentes a otras áreas de explotación puede dar como resultado una nueva vulnerabilidad. Por ejemplo un ataque de ingeniería inversa de un cliente de una aplicación web, puede provocar el descubrimiento de un 0day. Como se puede

[15] WSDL: Web Services Definition Language, es un protocolo utilizado para describir los puntos de conexión.

[16] UDDI: Universal Description Discovery Integration, es un protocolo usado para describir a los propios servicios web.

[17] SOAP: Simple Object Access Protocol, es un protocolo destinado a la comunicación entre servicios web.

deducir, todo vale, inyección de malware, phishing a los usuarios, explotaciones de buffer overflow, ataques XSS, inyecciones SQL, y un largo etc.

9.1.3 Herramientas

Algunas de las herramientas más utilizadas para explotar vulnerabilidades de aplicaciones web son:

Explotación de aplicaciones web: Burp Suite Professional, CookieDigger, WebScarab, Instant SOurce, WebCopier, HHTtrack, HttpBee, W3af o BlackWidow.

Seguridad en aplicaciones web: tenemos algunas como Acunetix Web Vulnerability Scanner, Watcher Web Security Tool, Netparker, N-Stalker, Web Application Security Scanner, VampireScan, OWASP ZAP, NetBrute, Syhunt Mini, SPIKE proxy o WSSA Web site Security Audit, entre otras.

Firewalls a nivel de aplicación web: dotDefender, ServerDefender VP, Radware's AppWall, Barracuda Web Application Firewall, ThreatSentry, IBM Security AppScan o ModSecurity, etc.

9.1.4 Contramedidas

Algunas de las medidas preventivas que podemos seguir son las siguientes:

- Normalizar las url y códigos html. Esto quiere decir transformarlos en formatos ASCII válidos, cambiando los caracteres especiales como espacios en blanco o '=' por '%' o en html usar &gt en vez de > (son sólo algunos ejemplos).
- Contramedidas propias de la inyección SQL (ver apartado 9.3.4).
- Utilizar las herramientas adecuadas para controlar los intentos de explotación
- Asegurarse de que las APIs son seguras.
- Controlar las cabeceras, cookies, consultas y parámetros adecuadamente validándolos antes de ser procesados
- Cifrar toda la información que se pueda que vaya a viajar por internet. Algoritmos de clave privada y pública como RSA fortalecen las sesiones.
- Usar firewalls para bloquear scripts.
- Tener en cuenta las contramedidas propias de los ataques D/DoS (ver apartado 8.4.2).
- Configurar adecuadamente los protocolos WDSL para que denieguen cualquier intento de escalamiento de privilegios.
- Desactivar los servicios que no se vayan a utilizar, así como eliminar cuentas y servicios creados por defecto.
- Se pueden evitar algunas acciones/funciones/variables de php para evadir ataques de inyección de ficheros, por ejemplo 'alllow_url_include' o 'register_globals', son solo algunos ejemplos.
- Para evitar exploits de captcha, es mejor que el usuario no tenga acceso directo a la solución. Además estos deben ser suficientemente complejos y variados, bien estructurados y seguros de encriptar los envíos.
- Utilizar siempre autenticaciones SSL.
- Almacenar siempre que se pueda, los datos cifrados y vaciar los historiales, los detalles de los *logeos* y la información derivada que pueda obtener de las consultas o de lso códigos fuentes.
- Las cookies deben caducar, estar asociadas a una ip legítima y además guardarse cifradas.
- Como siempre, mantener actualizados los programas.

9.2 Webservers

En el ámbito de los servidores, hay que aclarar varios conceptos de seguridad importantes dado que son susceptibles de muchos ataques. Son equipos conectados a Internet, que soportan muchas aplicaciones web y software, además de alojar bases de datos, webs, servicios, etc. Así que son objetivos expuestos que hay que proteger. Procesos como el hardening se ocupan de fortalecer estos webservers ante todo tipo de ataques. De nuevo, este apartado daría para un libro entero así que nos centraremos es los aspectos a considerar más importantes con el fin de tener una idea global focalizada [1, 8, 10, 12].

Los impactos de los ataques que comprometen a los servidores suelen ser muy graves dado que se extrapolan a muchos otros campos; redes, servicios, maquinas conectadas, bases de datos sensibles, etc. Los objetivos más comunes suelen ser comprometer la información de los usuarios pivotar para acceder a los servicios web o simplemente provocar un defacement[18].

Normalmente la arquitectura de un webserver suele ser la de un sistema junto a un sistema operativo especialmente diseñado para las tareas de servir/recibir (por ejemplo Windows server o Apache). Estos sistemas suelen ser escalables y ejecutan tareas web basadas en lenguajes de programación destinados a tal fin como PHP, bases de datos y por supuesto alojan los archivos físicamente. A su vez necesitan un cliente, un programa que sea capaz de comunicarse con él, pude ser un navegador web, o un programa especial. A través de esto, los usuarios comunes realizan las tareas. Además, el administrador suele tener permisos especiales para poder gestionar y analizar el contenido del servidor, ya puede ser remota o localmente. Dado que estas conexiones se realizan a través de Internet normalmente, los atacantes aprovechan este flanco para encontrar vulnerabilidades y explotarlas.

Como se puede prever, se abre un abanico de posibles intrusiones y técnicas de explotación que se explicarán a continuación.

9.2.1 Vectores de ataque

Muchos de los conceptos de ataque ya se han ido viendo a lo largo de todo este documento pero hay que tenerlos en cuenta.

Ataques D/DoS: mediante consultas incorrectas y/o falsas, procedente de una botnet por ejemplo, se puede provocar la caída de los servidores, como se ha descrito en el apartado 8.

DNS hijacking: si se puede infectar un servidor DNS, es posible realizad un secuestro de sesión modificando las tablas de mapeo de las direcciones de dominio. El usuario al buscar el dominio web, realizará una consulta al servidor DNS que le derivará a otro lugar, probablemente a un servidor web infectado.

Recursive DNS: este ataque es derivado del anterior, y consiste en comprometer los servidores DNS para que haga más consultas de las habituales a sus servidores DNS vecinos en busca de un dominio concreto. Esto puede usarse para amplificar los ataques a diversos servidores DNS.

Directorios transversales: al igual que pasaba con las aplicaciones web, es posible que una atacante pueda acceder a un directorio que no debería introduciendo rutas ambiguas como ../. En realidad es un error de configuración pero bastante habitual.

Ataques MITM: como también se explicó en el apartado 7 dedicado a redes, un ataque MITM es perfectamente viable en estos casos. Por supuesto, ataques de sniffing se combinan para interceptar y manipular el tráfico interponiéndose entre los interlocutores, en estos casos, cliente-servidor.

[18] Defacement: es un término comúnmente utilizado cuando lo que se busca es una modificación de algún contenido o funcionamiento de una web, una aplicación web, etc. Los autores se les llama defacers.

Phishing: Como se describió en el apartado 4 de ingeniería social, los ataques de phishing son un gran frente de ataques constantes hacia los usuarios incluso administradores de los servidores. Los objetivos siempre serán robar las credenciales derivando a los usuarios legítimos a servidores comprometidos, o infectar sus máquinas.

Defacement web: como se ha comentado en el punto anterior, estos ataques buscan cambiar la configuración de alguno de los servicios del servidor, su funcionamiento algún proceso específico. Suelen ocurrir casos en que las webs alojadas en servidores comprometidos devuelven la típica imagen de 'you have been hacked' o alguna foto burlesca. Muchas veces no llegan a ocasionar daños, simplemente grupos de atacantes lo hacen para dejar patente la debilidad del sistema públicamente. Para conseguir estos objetivos se suelen utilizar métodos como el SQL injection (ver apartado 9.3) o la explotación de cualquier vulnerabilidad encontrada.

Defectos de configuración: como siempre ocurre, existen defectos de configuraciones susceptibles de ser debilidades. Passwords y users por defecto, mensajes de error que dan demasiada información, certificados SSL configurados pro defecto, servicios y puertos activos innecesarios, posibilidades de introducir scripts en las consultas, descontrol de las consultas, url no normalizadas, y un largo etc.

Ataque splitting: este ataque basado en las respuestas HTTP consiste en añadir una cabecera modificada a los paquetes de respuestas de la víctima (un usuario). Es decir, cuando un usuario realiza una consulta legítima HTTP a un servidor, el atacante le responde con un paquete que simula una respuesta real del servidor (una segunda respuesta, derivada de la primera que recibió el atacante). Esto provoca que el atacante puede utilizar la respuesta legítima de la víctima y el servidor crea que habla con el usuario legítimo.

Ataque SSH por fuerza bruta: como su nombre indica, consiste en aplicar a taques de fuerza bruta a los logins del protocolo SSH utilizado por los administradores para obtener acceso no autorizado con privilegios.

Ataques a contraseñas: los ataques a contraseñas, como se comentaron en el apartado 8, son objeto de black hats. Normalmente los objetivos suelen ser protocolos SMTP, SSH, FTP, etc. mediante uso de ingeniería social, phishing, spoofing, etc.

Ataques a las aplicaciones web: como se ha comentado en el apartado 9.1, comprometer la aplicación web puede suponer vulnerar la seguridad del servidor.

9.2.2 Metodologías

Sabiendo los frentes abiertos y teniendo un concepto general de lo que son los webservers y sus amenazas, se puede realizar un esquema que confeccione las pautas de un ataque.

Primeramente, si no se ha hecho antes, se realiza una tarea de gahtering que recopile la máxima información posible sobre el servidor web a auditar/atacar. Todos los métodos mecionados en el apartado 2 son válidos (whois, traceroute, noticias, emails, etc.). Aquí aparece un fichero interesante en los buscadores, el robots.txt. Estos ficheros pueden aportar información importante y sensible sobre los directorios y archivos de los servidores. Estos archivos muchas veces aparecen adjuntos en las búsquedas por Internet como google.

Después, hay que realizar un escaneo de puertos, servicios, volcado de la web para analizar el código, etc. Es un proceso de enumeración en toda regla, el objetivo previo de la búsqueda de vulnerabilidades, será crucial dado que va a identificar claramente los focos de ataque.

Terminado toda esa ardua tarea de recopilación de información y enumeración, hay uqe buscar vulnerabilidades, a partir de lo encontrado. Utilizando las técnicas específicas descritas anteriormente podremos observar que vectores son más susceptibles de explotación. Sesiones de hijacking,a taques de diccionario o explotación de versiones antiguas de protocolos, etc. Como siempre depende de la información obtenida y de la creatividad de atacante. Si n auditor

encuentra muchas brechas de seguridad, será aquí cuando deba demostrar los daños y confeccionar un documento que sea capaz de reunir toda esta información

Después del ataque, si ha tenido éxito el supuesto atacante, deberá realizar tareas de borrado de huellas. Estas técnicas se explican en el apartado 7 y 10.

9.2.3 Herramientas

La herramienta por excelencia para explotar servidores es la misma que la utilizada para vulnerar sistemas, y es metasploit, junto con sus plugins y módulos. Este framework tiene una interfaz especializada en web (msfweb). No obstante, existen otras como UrlScan, Nikto, Nessus, Wfetch, THC-Hydra o Brutus.

Algunas herramientas para controlar los parches de seguridad interesantes pueden ser Altiris Client manager, Prism Suite, Secunia CSI, Security Manager Plus o Kaseya Security PAch Management.

En cuanto a protección y hardening tenemos: Syhunt Dynamic, n-Stalker Web Application, Wikto, Acunetix, Hackalert retina CS, Arirang, Nscan o Infiltrator entre otras.

9.2.4 Contramedidas

Algunas de las recomendaciones y prevenciones que se pueden tomar son las siguientes. En algunos casos son semejantes a muchas de las ya comentadas a lo largo de todo el documento:

- Separar las redes que incluyen a los servidores de las redes locales. Las redes deben estar protegidas por firewalls e IPS/IDS, incluyendo DMZ y monitorización constante.
- Siempre realizar el mantenimiento de los parches y actualizaciones de seguridad que aparecen, tanto del sistema operativo del servidor, como de los protocolos de seguridad implementados.
- Cada día aparecen nuevas vulnerabilidades. Por ello, el administrado debe estar al día de todos para mejorar todos los procesos que vayan apareciendo en la menor brevedad posible.
- Realizar backups periódicos y cifrar la información más sensible que se aloje en las páginas. Si la información almacenada no es usada habitualmente, lo más sensato es eliminarla y alojarla en una copia de seguridad. Las contraseñas deben guardarse cifradas.
- Eliminar servicios innecesarios y bloquear puertos que no se utilicen.
- Si se utilizan protocolos inseguros de por si como FTP, SMTP o Telnet, existen maneras de fortificarlos (comentados en el apartado 8).
- Las comunicaciones deben ser encriptadas y tuneadas.
- Eliminar módulos web y software que no sea necesario, pueden contener vulnerabilidades.
- Controlar las cuentas de usuario, manteniéndolas con los mínimos privilegios posibles y nunca dejarlas por defecto (incluyendo los routers). Lo mismo pasa con los procesos de sistema. Estos deben ejecutarse en modos restringidos.
- Monitorear los servidores DNS y protegerlos igual que los servidores que soportan aplicaciones web.
- Crear listas negras para las IP que intenten realizar ataques de fuerza bruta, además de logins reiterados. Además es necesario normalizar los directorios y los códigos fuente.
- El administrador debe mantener los certificados válidos y accesibles. Jamás puede ser revocados, dado que provocaría una grave deficiencia de seguridad.
- Tener en cuenta las contramedidas de los puntos 9.1.4 y 9.3.4 referentes a las aplicaciones web y a las inyecciones SQL. Estos dos apartados funcionan conjuntamente con los webservers y la vulneración de uno de estos flancos puede suponer la explotación de todo el sistema y redes.

9.3 SQL injection

La técnica de sql-injection es un proceso malicioso por la cual, alguien desde el exterior de la red, intenta manipular la base de datos de un servidor o una aplicación mediante comandos sql que aprovechan vulnerabilidades conocidas o deficiencias de seguridad. La intención es alterar el procesamiento normal de las consultas sql o accesos no autorizado, añadiendo y ejecutando código malicioso [11, 15].

Las respuestas que aceptan los servidores web y aplicaciones deberán tener medidas de seguridad para poder mitigar con esta problemática que busca la extracción de información sensible o la malversación de ésta. Estas vulnerabilidades de seguridad están en la capa de aplicación, es decir, son las aplicaciones web las que deben protegerse de estos tipos de ataques y no es un defecto en sí del lenguaje SQL.

Dicho esto, para entender exactamente en qué consisten en estos ataques, se van explicar los principales tipos que nos podemos encontrar.

9.3.1 Tipos y técnicas asociadas

Principalmente tenemos dos tipos de inyección SQL; errores basados en SQL e inyecciones SQL Blind

Errores Basados en SQL

Estas inyecciones de comandos buscan provocar un error o perturbación en las bases de datos afectadas. Este tipo a su vez se divide en 5 subtipos dependiendo de la naturaleza del error:

Procesos guardados en sistemas: como bien indica el nombre, se basa en provocar errores en procesos guardados de las bases de datos.

Comentarios de final de línea: cuando se inyecta código legítimo en una entrada, al final se puede añadir un apartado de comentarios que puede usarse para inyectar código. Por ejemplo:

- SELECT * FROM user WHERE name = ‘David’ AND userid IS NULL; --‘;

Consultas ilógicas: se pueden introducir parámetros, tipos de datos o nombres de tablas para obtener resultados de interés de la base de datos:

Tautología: son las inyecciones que siempre retornarán resultados verdaderos y provoca que las bases de datos pueden devolver información que no debería, por ejemplo:

- SELECT * FROM users WHERE login= ’user’ or 1=1 --’;

Union SQL: el comando UNION en SQL se puede utilizar para enlazar varias consultas que por separado no dejaría el sistema, por ejemplo:

- SELECT name, phone FROM users WHERE id=$id
- $id=1 UNION ALL SELECT creditCardNumber FROM cctable

SQL Blind

También llamado ataque a ciegas, se aprovecha de una vulnerabilidad web. Concretamente se observa cuando al realizar consultas erróneas, esta no contesta adecuadamente. Para entenderlo mejor se defienden los tres casos posibles:

Página general: procede cuando ante la inyección SQL, aparece una ventana o página web generada automáticamente.

No error: esto ocurre cuando el atacante no recibe ningún tipo de información al inyectar un código SQL malicioso excepto cuando la consulta es legítima. Por ejemplo podemos deducir si un número de tarjeta de crédito, un usuario, un dato concreto existe en la base de datos o no.

Tiempos excesivos: se pueden introducir comandos que jueguen con los lapsos de tiempo entre servidores y consultas para deducir cierta información, por ejemplo que versions de MySQL tiene, por ejemplo:

- SELECT * FROM users WHERE id=1-IF(MID(VERSION(),1,1) = '5', SLEEP(15),0);

9.3.2 Metodología de un ataque

Las metodologías de SQL injection son todo un arte y requiere de un proceso meticulosos para llegar a explotar una base de datos. Una manera de esquematizar claramente las metodologías es dividirlas en dos sub conjuntos [1, 7, 11]:

Gathering: volvemos a la recogida de información pero esta vez única y exclusivamente dedicados a obtener toda la información que se pueda de la base de datos y todo lo que le rodea. Con herramientas como wireshark podemos desmenuzar los paquetes GET, POST, etc. para descubrir métodos de entrada, procesos ocultos, cookies etc.

De igual modo, provocar errores, es decir, realizar pruebas con entradas inválidas, incoherentes, etc. nos informará de datos como sistemas operativos y tipos de bases de datos, privilegios, etc. También se pueden deducir las estructuras de las consultas y el motor de ejecución de las búsquedas en la base de datos.

Otro punto interesante es el de analizar el código fuente en busca de vulnerabilidades susceptibles de ser explotadas por un ataque de SQL injection. Existen herramientas concretas que realizan análisis exhaustivos en busca de estas vulnerabilidades (ver apartado 9.3.3).

Creación y explotación de los códigos: una vez reunida toda la información toca organizarla y construir posibles códigos que puedan tener éxito en los ataques de inyección. Utilizando las técnicas del apartado anterior 9.3.1 se irán confeccionando ataques hasta obtener algún tipo de éxito, obtención de alguna información, escalado de privilegios, etc. Hay que tener un conocimiento solido sobre todas las bases de datos utilizadas en el mercado, por ejemplo tenemos MySQL, Oracle DB, PostgreSQL, Microsoft Access, etc. y cada una con sus vulnerabilidades y sus nomenclaturas particulares.

Aquí además se pueden combinar técnicas y herramientas de explotación nombradas en el apartado 7 (por ejemplo metaesploit). También se pueden hacer scripts que vallan probando comandos con pequeñas diferencias e ir almacenando los resultados para posteriormente extraer la información. Como siempre depende de la inspiración y creatividad del atacante.

Además hay que tener en cuenta que siguen habiendo barreras como los firewalls o IDS. Normalmente para evadir estos casos se utilizan técnicas como la ofuscación de codigo, el cifrado de los comandos en hexadecimal, manipular los espacios vacíos, entre otras.

9.3.3 Herramientas

Herramientas SQL injection: por ejemplo tenemos el framwork BSQLHacker, o programas como SQLInjector, SQLbftools, bfsl, Marathon Tool, SQL Power Injector, Havij, SQL Brute, sqlmap, Pangolin, sqlninja, sqlget o Absinthe. Para móviles también tenemos algunas interesantes como sqlmapchik o DroidSQLi.

Análisis de codigo: Microsoft Source Code Analyzer, CodeSecure o HP QAinspect

Protección ante ataques: dotDefender, IBM Security Appscan, Webcruiser, SQL Block Monitor, SQLDict, o HP WebInspect.

9.3.4 Contramedidas

- Algunas de las medidas preventivas que podemos tomar son:
- Configurar correctamente las reglas de los IDS y Firewalls para detectar los ataques (por ejemplo, a SNORT se le puede decir que bloquee ciertos caracteres como /((\%27)|(\ '))union/ix).
- Utilizar herramientas para detectar los ataques.
- Configurar adecuadamente el retorno de errores para que retorne siempre resultados aleatorios (se entiende que dentro de los permisos que pueda tener un usuario normal), en vez de mensajes de error que puedan dar información derivada.
- Si es posible, reprogramar el código fuente de algunas aplicaciones para que no trate consultas como las de tautología.
- Por defecto, los usuarios con acceso a las bases de datos deben tener los mínimos permisos necesarios.
- Repudiar consultas que contengan comentarios, datos binarios o secuencias de escape.
- Nunca permitir la concatenación de consultas.
- Utilizar un hash fuerte (por ejemplo el SHA256) para almacenar las contraseñas en las bases de datos como medida preventiva.
- No dejar información (comentarios en código, explicaciones de funcionamiento, etc.) en los códigos fuente.
- Controlar los tiempos de aceptación de consultas.
- No utilizar sentencias sql construidas dinámicamente por motores

CAPÍTULO X

10. Análisis forense

Este último apartado se dedicará al concepto del análisis forense. Estos análisis no están dentro de las fases descritas anteriormente de hacking o de auditoría de seguridad profesional. Una vez el atacante haya alcanzado su objetivo el administrador del sistema deberá recoger evidencias y analizar los sistemas en profundidad para averiguar qué pasó y sobretodo cómo pasó y que alcances y repercusiones tiene el ataque. Se podría decir que es la fase de post-hackeo de un sistema, donde el black hat no tiene nada que ver, a parte de las pruebas que haya dejado a su paso.

Para ello, existen un conjunto de técnicas y herramientas específicas cuyos objetivos son desmenuzar minuciosamente la información que el atacante haya podido dejar. Estas huellas ya están descritas en los apartados anteriores, pero por lo general serán aplicaciones instaladas (malware en general, backdoors, troyanos, etc.), actividades realizadas y métodos emprados.

Este apartado (como todos los anteriores) es muy extenso y específico para cada sistema operativo, y de nuevo se podrían escribir extensas páginas sobre él. Dado que los más utilizados son entornos Windows, este apartado se centrará particularmente en ellos, pero la mayoría de conceptos generales se pueden extrapolar a cualquier otro [9].

10.1 Recopilando evidencias

La recopilación de evidencias es la tarea de recogida de hechos que puedan ser contrastados, tales como fechas de registros, cookies almacenadas en los navegadores, archivos temporales, archivos de instalaciones de software que haya podido quedar en el sistema después de su desinstalación, ediciones de ficheros, medios físicos, y un largo etc. Para que la recogida de evidencias sea productiva hay que seguir unas ciertas pautas que se incluyen en el documento RFC 3227 (Request for Comments) público aceptado por la comunidad forense. Este documento se centra en aspectos como acciones que se deben evitar, consideraciones sobre la privacidad y uso de datos privados, además de aspectos legales y relacionados con la volatilidad, transparencia y almacenamiento de los datos incluyendo la cadena de custodia.

Este documento también reúne una serie de recomendaciones para llevar a cabo. Estas recomendaciones incluyen:

1. Capturar la imagen exacta del sistema a inspeccionar
2. Almacenar toda la información posible para su análisis
3. Recolectar las evidencias teniendo en cuenta la volatilidad de la información. Esto incluye registros de CPU, información de caché, tablas de enrutamiento, tablas de procesos y estadísticas, memoria almacenada en el sistema mientras está en ejecución (por ejemplo volcado de RAM), logs, configuraciones de entorno y de red, unidades de copias de seguridad y datos de monitorización.

Teniendo en cuenta estos factores, se debe proceder a realizar los análisis de los datos Principalmente tenemos tres tipos principales de aspectos a analizar; discos, ficheros temporales y tramas de red, además de otros complementarios.

10.2 Análisis de discos

Para realizar la analítica de discos y medios digitales no existe ningún método infalible ni una línea fija que seguir, sino que cada analizador utilizará las técnicas y métodos más adecuados

para obtener sus resultados. Por ejemplo, uno de los métodos más usados son los que dicta la Digital Forensics Research Workshop (DRFW).

En general, se suelen distinguir una serie de pautas generales en los métodos:

1. Definir la línea temporal y la indexación de los datos.
2. Diferenciación de la información basada en firmas y formatos.
3. Búsqueda y recuperación de datos eliminados.

Definir la línea temporal y la indexación de los datos

Definir la línea temporal crea un marco sobre el que el resto del análisis forense se va a basar. Es por ello que se creará un filtro de información sobre el antes y después de los sucesos ocurridos. Sin este punto de referencia base, difícilmente se seguirá un hilo de investigación productivo con resultados coherentes. Aquí seleccionaremos los ficheros modificados y/o eliminados, ficheros que no han sido modificados o incoherentes con el resto del sistema, con fechas anteriores al punto temporal al análisis o logs modificados entre las horas del ataque. En definitiva son las evidencias palpables de que algo ha pasado y en las cuales nos basaremos.

A menudo esta información será cuantiosa y no toda será útil, por lo que se deberá tratar de algún modo para no perder ningún detalle de las acciones. Es por ello que se debe ordenar e indexar adecuadamente para poder realizar los análisis coherentemente.

Diferenciación de la información basada en firmas y formatos

Normalmente una vez tengamos una cuantiosa cantidad de información nada despreciable a analizar, se deberá catalogar en fichero o datos concretos que son especialmente relevantes para el caso. Por ejemplo, si la investigación está causada por el robo de información confidencial, será lógico analizar datos que contengan dicha información (.pdf, correos electrónicos, etc.). Para ello existen herramientas que realizan estos procesos de refinado de la información indexada. También hay que tener en cuenta que el atacante puede haber escondido premeditadamente información en ficheros, con lo que no se puede dar por supuesto que dichos datos estarán en una categoría concreta indiscutiblemente.

Búsqueda y recuperación de datos eliminados

Y llegamos al núcleo y grueso del análisis. Se tienen todos los datos útiles destacados y se debe proceder a buscar pruebas empíricas de que ha habido un ataque. Por ejemplo, un log con una conexión a un IP desconocida, un residuo de un backdoor alojado en la línea temporal en la cual se está trabajando, etc. Aquí dependerá en gran medida del olfato y la experiencia del analista para encontrar dichas pruebas, dado que no siempre son tan evidentes.

También se deberá tener muy en cuenta el hecho de que el atacante haya borrado sus huellas antes de terminar. Es por ello que se deberá realizar una recuperación y análisis de los datos eliminados. Es bien sabido que el borrado de un fichero no elimina por completo dicho archivo del medio en el que está guardado, sino que le da permiso a la dirección de memoria para que lo reescriba cuando sea necesario. Así que el archivo sigue ahí, hasta que se reescriba con otra información, es decir, o bien pase suficientemente tiempo como para que se reutilice el espacio de memoria, o bien se reescriba a propósito, por ejemplo formateando repetidamente la memoria o reescribiéndola completa e intencionadamente. Mediante herramientas de recuperación, se puede analizar el espacio libre en busca de estos archivos residuales pero no existe ninguna garantía de éxito. Además de todo suelen ser procesos que tardan en exceso y puede devolver ficheros incompletos (por ejemplo, un archivo que ocupa más de un clúster y parte de él ya se ha sobrescrito).

10.3 Análisis de ficheros temporales

Hay que tener un especial cuidado con los archivos temporales que se alojan en los sistemas cuando se ha realizado un ataque, por su volatilidad. Es primordial recuperar esos datos lo más rápidamente posible después de la intrusión maliciosa puesto que con muchas probabilidades contendrán indicios y pruebas relevantes. En general existen los siguientes tipos de archivos temporales

Navegadores de internet

Las cookies, historiales, caché y las pequeñas descargas que los navegadores realizan son las principales bazas para inspeccionar estos datos. Cada vez que se carga un web, parte de sus datos acaban en la caché del navegador que son eliminados cuando se llega a un límite impuesto. Pasa igual con el resto de tipos de archivos, por lo cual, pueden contener información importante sobre los movimientos realizados (por ejemplo, un servidor web que se ha utilizado para descargar un malware en el equipo de la víctima).

Desinstalaciones

A menudo en sistemas Windows, al realizar instalaciones quedan restos. No siempre serán temporales, por ejemplo la ruta de una carpeta, pero otras veces quedarán rastros en los registros de Windows que puede indicar que tipo de programa se ha utilizado en el ataque. Además hay que tener en cuenta que si las funciones de restaurar sistema están habilitadas, la información borrada seguirá en el sistema operativo por si ha de recuperarse.

También hay que mencionar la papelera de reciclaje, que puede estudiarse para encontrar indicios. Las papeleras realmente no eliminan el archivo y se deshacen de él, sino que renombran el archivo para que cuando el usuario dé la confirmación de vaciar la papelera, sepa dónde está alojado y lo elimine de la manera descrita en el apartado 10.2. Cada vez que es renombrado, lógicamente el nombre cambia. Todo esto se almacena en un fichero llamado INFO2 que se limpia cuando se vacía la papelera. Dicho esto, existen herramientas que recomponen el archivo INFO2 para su análisis.

Colas de impresión

Las colas de impresión son otro punto que puede ser importante. Es muy probable que no se haya utilizado claro está, pero se deben comprobar las colas de impresión temporales. En Windows se suelen alojar en System32\SPOOL\PRINTERS y podremos encontrar archivos con extensiones SPL (Spool File Format), SHD (Shadow File Format), RAW o EMF (Enhanced Metafile). Existen herramientas específicas para el análisis de estos archivos respectivamente que permitirá extraer información.

Memoria RAM

Sabemos que la memoria RAM es la utilizada por el sistema operativo para volcar gran parte de esos datos que se van a utilizar. Al ser una memoria que se vacía cada vez que el sistema es reiniciado, habrá que mantener el sistema encendido. Además es posible que los procesos de ejecución de hilos, notificaciones al subsistema, carga de librerías DLL, etc. que pertenezcan al proceso malicioso no se estén ejecutando en el momento o ya hayan sido reescritas, por lo que es probable que no surja efecto. Existen herramientas que realizan volcados íntegros del contenido de las memorias RAM para su posterior análisis en detalle, en busca de procesos, accesos a disco, información relacionada con la CPU, alojamientos de variables, claves de registro, etc.

10.4 Análisis de red

Mediante el sniffing se podrá analizar capturar todo el tráfico de la red para su análisis. Es posible que en el momento posterior al ataque la información recogida no sea útil, pero podemos pensar en que se podría estar monitoreando todo el tráfico siempre (cómo si de una cámara de vigilancia se tratara). Si esta medida de prevención sucede, se podrá analizar las tramas de red que han circulado con una alta probabilidad, saber dónde, cómo y cuándo se han realizado las conexiones malintencionadas.

Si analizamos estos paquetes con detenimiento se podrá deducir los protocolos utilizados para las conexiones o datos muy llamativos como identificación de remitentes/destinatarios de emails (SMTP), intentos de escaneos de puertos (mediante el three way hadshake), si se ha utilizado alguna herramienta concreta que se conozca su modus operandi (por ejemplo Nmap suele utilizar puertos superiores a 20000) y en general las acciones descritas en los apartados 8 y 9 de este documento.

10.5 Otros análisis

Algunos otros aspectos relevantes se deben tener en cuenta y merece la pena mencionar para acabar de tener una visión completa y global de los análisis forenses. Son los siguientes.

Registros de windows

Existen los famosos registros de Windows que suelen contener pruebas de toda actividad en dicho sistema. Por ejemplo, suelen contener información sobre aplicaciones instaladas, aplicaciones nativas, configuraciones de la máquina, manipulaciones de usuarios, etc. Suelen alojarse de nuevo en la carpeta \System32\Config. Por lo tanto resulta importante realizar un exhaustivo análisis de estos archivos en busca de evidencias.

Accesos de usuario

Es posible rastrear los accesos al sistema de un usuario concreto, buscando datos como horas y fechas del login, directorios de trabajo, nombres y contraseñas, que puedan dar luz al modo en que se ha realizado la intrusión. Esto queda controlado por la API de Windows Local Security Authority API.

Metainformación

Este apartado realmente es transversal a todos los procesos de análisis de datos, dado que incluye toda esa información adicional oculta al usuario que se utiliza para interactuar con el sistema. Podemos realizar análisis de archivos con extensiones .pdf o .doc y encontrar versiones de programas, nombres de usuarios y redes asociados, fecha de creación del documento original, etc. Existen herramientas son especialmente útiles para extraer metadatos.

10.6 Herramientas

Algunas de las herramientas útiles para el análisis forense son las siguientes, además de las ya incluidas en Kali Linux 2.0:

General: la suite Forensic Adquisition Utilities.
Recomponer INFO2: Rifiuti.
Volcado de RAM: WinDbg, Helix o SystemDump.
Análisis de red: Whiresharko o logs de Snort.
Extracción de metadatos: FOCA.
Análisis de accesos de usuario: Netusers (incluida en Windows) o PsLoggedOn.

CAPÍTULO XI

11. Conclusiones finales

Llegados al final del documento, lo primero que vedará a la cabeza del lector es que el mundo de la seguridad informática es completamente enorme, inabarcable en un solo documento y que está claramente influenciado por las intenciones del que está detrás de la máquina. Además de todo ello, se intuye una mezcla de curiosidad, picardía y conocimientos informáticos avanzados y matemáticos, además de comprender y tener presente el comportamiento humano. Todo esto confluye de una manera especial dado lugar a la temática del documento, el apasionante mundo de la seguridad de la información.

Hemos visto que la figura del hacker está llena de luces y sombras, dependiendo de los objetivos de white o black hat. A partir de esa base contextual previa a todos los conceptos que se iban a explicar, se ha intentando extraer una visión periférica de las fases de un ataque o una auditoria y así, a medida que se profundiza en cada tema, se van abriendo vías de investigación y aprendizaje en la cuales el lector se puede embarcar.

En la primera fase sobre recolección de información, se ha visto como la mezcla de creatividad y conocimientos se fusionan con el fin de extraer toda la información posible de las máquinas objetivo, incluyendo todos los posibles vectores de ataque susceptibles de ser explotados. Innumerables herramientas y técnicas están disponibles para facilitar la fuga de información, tanto personal como empresarial y la mejor manera de poderse proteger es conociéndolas a fondo. Hemos visto como existen muchas contramedidas adecuadas para evitar esta recolección de información pero en la era de la información, donde todo tiende a estar conectado, cada vez será más difícil la labor de contención, por no decir imposible. Muchas de las veces se ha visto también que el eslabón más débil de la cadena es la propia persona, independientemente de si el sistema está mejor o peor protegido. Por ello, las labores transversales de ingeniería social se han tenido presente en todo momento, evidenciando las flaquezas.

Después del gathering, se ha visto todo el tipo de malware que circula por la red. Resulta curioso ver el amplio abanico de código malicioso que los black hat han ido creando a lo largo de los últimos años para poder explotar los sistemas.

La labor de explotación de los sistemas y redes es la más concentrada de todo el documento a la vez que la más extensa. Las vulnerabilidades de software parece ser que siempre van a estar, puesto que los programas están creados por personas, y las personas cometemos errores. La inyección de un payload en un brecha de seguridad para escalar privilegios y obtener todo el control de la máquina, o el secuestro de una sesión para controlar el tráfico que circula por la red son sólo algunos ejemplos que seguro llaman la atención y ponen al descubierto aspectos no muy simpáticos de la tecnología.

Una vez el atacante a tenido éxito, lo que es seguro es que no existirá una segunda oportunidad. Aunque todo ocurra de manera virtual, la información es real e influye a personas reales, con lo que la educación adecuada sobre el uso de la tecnología es y será, vital para garantizar la privacidad de sus usuarios. El sentido común y la concienciación de la información que se difunde por la red son asignaturas pendientes que en un futuro muy cercano se deberán abordar.

Indice de figuras

Glosario básico

0day: vulnerabilidad poco o nada conocida y por la que no se conoce remedio aparente.

ACK, SYN, ICMP, etc: son tipos de paquetes enviados entre máquinas para poder comunicarse entre ellas.

Backdoor: malware que se oculta en el sistema para que el atacante pueda acceder a él siempre que quiera.

Dirección MAC: código que identifica a una tarjeta de red.

DMZ: Zona desmilitarizada. Es un segmento de una red que está aislada del resto para proteger el sistema principal de ataques.

DNS: servidor de internet encargado de traducir las ips de las web a su nombre común www.nombredeweb.com por ejemplo.

D/DoS: ataque de red que busca atorar algún sistema de tal forma que no se pueda acceder al servicio por un tiempo o indefinidamente.

El juego de la vida: Juego matemático basado en reglas simples que va evolucionando evolucionando por si solo provocando un efecto de movimiento de objetos en pantalla.

Explotación: aprovechamiento de una vulnerabilidad de un sistema o red.

Firewall: programa o hardware informático configurable para bloquear y filtrar el tráfico de red.

Flags: son atributos dentro de los paquetes que detallan aspectos técnicos de éstos.

Footprinting y fingerprinting: huellas que deja un programa o un usuario respectivamente al realizar acciones en un sistema operativo.

Framework: entorno de trabajo desarrollado especialmente para la herramienta, con el fín de facilitar el trabajo.

FTP: File Transfer Protocol. Es un protocolo utilzado para transferir archivos entre servidores y maquinas de cualquier tipo.

Gathering: proceso de recolección de información sobre un sistema.

Glider: una de las formas participantes en 'el juego de la vida'.

Hacker: experto en seguridad informática que da usos no convencionales a las herramientas, sistemas y redes, independientemente de sus intenciones.

Hacking: acciones llevadas a cabo por hackers

Hardening: proceso de fortificación de un servidor para protegerlo de ataques.
CIFS/SMB: Common Internet File System y Server Message Blocks respectivamente

Hash: código resultante de la aplicación de un algoritmo de cifrado a un dato/programa que sirve para identificar inequívocamente ese programa. Así se puede ocultar la información (contraseñas) o verificar la integridad de un programa

Honeypot: Sistema trampa para atraer la atención de un atacante y tenerlo localizado y analizar sus intenciones y acciones.

HTML: Hyper Text Markup Language. Pseudo lenguaje de programación utilizado para la creación de páginas web.

HTTP: Hyper Text Transfer Portocol. Es un protocolo utilizado para el intercambio de información de páginas web.

IP pública/privada: código de números i/o letras que identifica a un punto de conexión a internet, o un sistema. Unos son conocidos por internet y los otros son privados a nivel interno de empresa/vivienda.

IDS: programa o hardware programable para detectar ataques y reaccionar a éstas.

Kernel: Núcleo de un sistema operativo Linux. Se encarga de ejecutar las tareas y servicios más básicos y esenciales de éste.

LDAP: Leghtweight Directory Access Protocol, base de código abierto en la que se fundó active directory.

Log: Archivo que almacena algún suceso en un sistema operativo.

Malware: programa con intenciones maliciosas que se instala en un sistema operativo.

Metadatos: datos ocultos a los usuarios que utiliza el sistema operativo para manipular el archivo.

Metaesploit: programa muy versátil que engloba muchas disciplinas de la seguridad informática. Muy utilizado por los auditores y atacantes de la seguridad informática hoy en dia.

NIS: Network Information Service. Protocolo usado por servicios en internet.

Paquetes: las comunicaciones en internet se llevan a cabo mediante la fragmentación y empaquetado de la información. Al llegar a su destino, se ensamblan para formar la información.

Payload: pequeño codigo que utiliza un exploit para aprovechar alguna vulnerabilidad y realizar así acciones maliciosas en el sistema.

Pentesting: acción de auditar un sistema referente a la seguridad informática

Ping: Paquete simple que se utiliza para comprobar si existe comunicación entre dos máquinas.

Plugin: Pequeño programa que se acopla a otro más completo para mejorar y ampliar sus funciones

Protocolo SMTP: convenio de comunicaciones de internet para intercambiar emails.

Protocolo VoIP: protocolo que utiliza las IP comunes para realizar llamadas a través de internet.

Protocolos TCP/UDP: convenio de comunicaciones en internet para el intercambio de datos.

Proxy: Un servidor que ejerce de intermediario entre un sistema e internet con el fin de proteger la dirección IP origen.

PTES: siglas de Penetration Testing Execution Estándar que reúnen un conjunto de estándares para realizar auditorías de seguridad.

Root: Usuario de un sistema operativo con acceso total al sistema.

Shellcode: conjunto de ordenes traducidas a un lenguaje de de bajo nivel (muy básico para el sistema) con el objetivo de ser ejecutado en una pequeña porción de memoria

Sniffer: Programa que se encarga de capturar todo el tráfico de paquetes de una red para su posterior análisis.

SSH: Secure Shell. Protocolo utilizado para comunicarse con una máquina remotamente y poder ejercer labores administrativas en la máquina.

SSL: Secure Socket Layer. Protocolo para cifrar los paquetes de datos que se envían y reciben a través de Internet. Se usa para proteger las comunicaciones.

uPnP: Unerversal Plug and Play. Protocolo utilizado en los sistemas operativos para conectar dispositivos sin instalar ningún programa adición por el usuario.

URL: dirección de una página web en formato legible comúnmente.

VM: siglas de Virtual Machine. Es un programa que tiene la capacidad de instalar un sistema operativo aislado dentro de otro.

VPN: red privada virtual por la cual el trafico de red está controlado.

.

.

.

Referencias

[1] EC-Council (2016). *Certified ethical hacker ((CEH) 9ª ed.)*. USA: Autor: EC-Council.

[2] Jara, H.; Pacheco, F.G. (2012). *Ethical hacking 2.0.* Buenos Aires: Fox Andina.

[3] Wrightson, T. (2015). *Advanced persistent threat hacking: the art and science of hacking any organization*. New York: McGraw-Hill Education.

[4] Baloch, R. (2015). *Ethical hacking and penetration testing guide*. Boca Raton: CRC Press.

[5] Regalado, D.; Harris, D.; Harper, A.; Eagle, C.; Ness, J.; Spasojevic, B.; Linn, R.; Sims, S.; (2015). *Gray hat hacking: the ethical hacker's handbook (4ª ed.)*. New York: McGraw-Hill Education.

[6] Astudillo, K.; (2013). *Hacking ético 101*. Argenitina: Autor.

[7] González, P.; Sánchez, G.; Soriano, J.M.; (2013).*Pentesting con Kali*. 0xWord: Madrid.

[8] Puente, D. ;(2013). *Linux exploiting*. 0xWord: Madrid.

[9] Garrido, J.; (2012). *Análisis forense digital en entornos Windows*.Informática64: Madrid.

[10] Álvarez, C.; (2013). *Hardening de servidores GNU/Linux*. 0xWord: Madrid.

[11] García, J.L.; (2014). *Ataques a redes de datos*. 0xWord: Madrid.

[12] Kurose, J.F.; Ross, K.W.; (2013). *Computer networking (6ª ed.)*. Pearson: Harlow.

[13] PTES Technical guideline (2014) [En línea]. Disponible: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines [Último acceso: 14/04/2016].

[14] Ministerio de educación cultura y deporte (2012) [En línia]. Disponible: http://recursostic.educacion.es/observatorio/web/gl/software/software-general/1040-introduccion-a-la-seguridad-informatica?start=3 [Último acceso: 13/4/2016]

[15] SqlInjection.net (2016) [En línea]. Disponible: www.sqlinjection.net [Último acceso 01/05/2016]

[16] XII Ciclo UPM TASSI (2016). *Conferencia 2: Montando un escenario de hacking IPv6 por 2 €.* [En línea]. Disponible: www.criptored.upm.es/multimedia/multimedia [Último acceso: 05/05/2016]

[17] Potrias, L.; Bonnefoy, M.; Wilutzky, D. (productores) y Potrias, L. (directora). (2014). *Citizenfour* [película documental].Estados Unidos.

[18] WebWiz (2016). *DNS, Network & IP Tools.* [En línea]. Disponible:https://network-tools.webwiz.co.uk [Último acceso: 15/05/2016].

[19] Ip Adress Location [2016]. *IPLoockup*.Disponible:www.ipadresslocation.org [Último acesso: 15/05/2016]

[20] Alonso, C. (2015) [En línea]. Disponible: www.elladodelmal.com [Último acceso: 13/05/2016]

[21] International Organization for Standardization y International Electrotechnical Commission. (2005-2016). [En línea]. Disponible: http://www.iso27000.es [Último acceso: 13/05/2016]

[22] SecuritybyDefault. (2016). [En línea]. Disponible: www.securitybydefault.com [Último acceso 28/04/2016]

Anexo - Software complementario

Web donde descargar software: http://sectools.org

Apartado 3 - Gathering

- Propósito general sobre footprinting web: Burp suite, Zaproxy, Paros Proxy, Website Informer, Firebug.
- Análisis de CMS: Nikto, Plecost, Wpscan, Joomscan.
- Mirroring web: BlackWidow, NCollector Studio, PageNest, Backstreet Browser, Website Ripper Copier, Offline Explorer Enterprise, GNU Wget, Teleport Pro, Hooeey Webprint, Portable Offline Browser.
- Monitorización de actualizaciones web: Change detection, onWebChange, Follow That Page, Infominder, Page2RSS, TrackedContent, Watch That Page, Websnitcher, Check4Change, Update scanner.
- Preguntas DNS: DIG, DNSWatch, myDNSTools, DomainTools, Professional Toolset, DNS Query Utility, DNS Records, DNS Lookup, DNSData View, DNS Query Utility, dnsdict6, dnsrecon, dnsrevenum6, dnstracker, dnswalk, fierce, urlcrazy.
- Rastreadores de emails: Wesware, Zendio, Pointofmail, ContactMonkey, Read Notify, WhoReadMe, DidTheyReadIt, GetNotify, Tracce Email, G-Lock Analytics.
- Herramientas traceroute: Network Pinger, Magic NetTrace, GEOSpider, 3D Traceroute, vTrace, AnalogX HyperTrace, Trout, Network Systems Traceroute, Ping Plotter, Roadkil's Trace Route.
- Servicios online útiles para la labor de hacking con motores de búsqueda: AnyWho, PeopleSmart, US Search, Veromi, Intelius, PrivateEye, People Search Now, PeopleFinders, Public Background Checks.
- Algunos softwares que ayudan a explotar el protocolo Whois: LanWhois, HotWhois, Batch IP Converter, ActiveWhois, CallerIP, WhoisThisDomain, SoftFuse Whois, Whois Lookup Multiple Adresses, Whois Analyzer Pro.
- Otros softwares de propósito general que abarcan varias de las áreas mencionadas en el apartado 3: Prefix Whois, NetMask, NetScanTools Pro, Binging, Tctrace, SearchNug, Autonomous System Scanner (ASS), TinEye, DNS-DIgger, Robtex, Dig Web Interface, SpiderFoot, White Pages, NSlookup, Email Tracking Tool, Zeba Search, yoName, GeoTrace, Ping-Probe, DomainHostingView, MetaGoofil, GMapCatcher, Wikto, Search Diggity, SiteDigger, GoogleHACK DB, Google Hacks, Gooscan, BiLE Suite, Trellian.

Apartado 5 – Análisis de vulnerabilidades

- Softwares interesantes sobre análisis devulnerabilidades: Retina, Nexpose, Spike, Yersinia.
- Enumeración de usuarios: Suite Pstechnet de microsoft (PsList, PsLoggedOn, PsLogList, PsPasswd, PsShutdown, etc)
- Enumeración SNMP: SNMP Scanner, Getif, Net-NSMP, SNMP Informant, SoftPerfect Network Scanner, Nsauditor Network Security Auditor, Spice, iReasoning MIB Browser.
- Enumeración LDAP: Active Directory Explorer, Jxplorer, LDAP Admin tool, LDAP Account Manager, LEX- The LDAP Explorer, LDAP Browser Editor.
- Enumeración NTP: NTP Server Scanner, Wireshark, AtomSync, NTPQuery, PresenTense NTP Auditor, PresentTense Timer Server, LAN Time analyser.
- Enumeración SMTP: NetScanTools, Telnet, Nslookup,

Apartado 6 – Malware

Link hacia un google doc que adjunta todos los ransomware conocidos hasta la fecha, sus extensiones, sus decrypts, screenshoots y más. :

https://docs.google.com/spreadsheets/d/1TWS238xacAto-fLKh1n5uTsdijWdCEsGIM0Y0Hvmc5g/htmlview?sle=true