



Universidad
Francisco de Vitoria
UFV Madrid



Hacking Ético

Metasploit 1

Laboratorio Básico



Gonzalo Pascual Romero

Fecha: 23/11/2023

Índice

1. Alcance.	3
2. Desarrollo del estudio.....	4
3. Conclusiones.....	14



Alcance

1. Se usará motor PostgreSQL, aunque si el alumno quiere puede usar MySQL o el que tenga instalado en Metasploit
2. Arrancar la base de datos PostgreSQL con el comando start, desde la ubicación del proceso
3. Desde línea de comando Linux:
 - a. "sudo su postgres -c psql"
 - b. "Alter user postgres with password 'laquequerais';"
 - c. "ALTER ROLE"
4. Desde Metasploit
 - a. "db connect postgres:lapassword@127.0.0.1/test_ufv"
 - b. Comprobar el estado del fwk respecto a la bbdd
 - c. Ejecutar nmap y que vuelque el resultado en un fichero XML (-oX)
 - d. Importar el resultado desde metasploit con el comando:"db import resultadoNMAP.xml"
5. Una vez importado, interactuar con los comandos
 - a. "db_host"
 - b. "db_services"
 - c. "db_notes"
6. Ejecutar nmap y que los datos se guarden en la base de datos
 - a. Asegurarse que la bbdd está conectada
 - b. Ejecutar db_nmap contra una IP
 - c. Revisar lo que hay en la bbdd
7. Escanear un FTP desde Metasploit
 - a. Cargar el módulo auxiliar de FTP (use +. Módulo de FTP)
 - b. Configurar variable
 - c. Ejecutar
8. Escanear un SSH desde metasploit
 - a. Cargar el módulo auxiliar de FTP (use +. Módulo de FTP)
 - b. Configurar variable
 - c. Ejecutar



Desarrollo del estudio

Metasploit: Metasploit es un marco de desarrollo de código abierto que proporciona herramientas para desarrollar, probar y ejecutar exploits contra sistemas informáticos. Facilita a los profesionales de la seguridad y a los hackers la automatización de tareas comunes relacionadas con la penetración y prueba de seguridad. El marco Metasploit incluye módulos para realizar diversas tareas, como la explotación de vulnerabilidades, el análisis de contraseñas, la recopilación de información y la creación de payloads personalizados.

PostgreSQL: PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto y orientado a objetos. Es altamente configurable y ofrece características avanzadas como soporte para procedimientos almacenados, disparadores, índices y replicación. Metasploit utiliza PostgreSQL como base de datos para almacenar información sobre exploits, payloads, hosts comprometidos y otra información relevante.

Nmap: Nmap, que significa "Network Mapper", es una herramienta de código abierto utilizada para explorar redes y realizar escaneos de seguridad. Nmap utiliza paquetes de red para determinar qué dispositivos están activos en una red, qué servicios (puertos) están abiertos en esos dispositivos, qué sistemas operativos están en ejecución y otra información detallada sobre la red.

1. Se usará motor PostgreSQL, aunque si el alumno quiere puede usar MySQL o el que tenga instalado en Metasploit

Usé PostgreSQL porque ya está instalado en Kali y así evitar complicaciones que podría tener al usar MySQL o cualquier otro

2. Arrancar la base de datos PostgreSQL con el comando start, desde la ubicación del proceso

Primero arranqué metasploit con el comando "msfconsole"

Después dentro de Metasploit ejecute el comando db_status para comprobar el estado de la base de datos y me decía que postgresql estaba seleccionado, pero no conectado, por lo que habría que conectarlo.



```
msf6 > db_status  
[*] postgresql selected, no connection
```

Para conectarlo hice un “msfdb init” y un “service postgresql start” para crear las bases de datos e iniciarla

```
(root@kali)-[/home/kali]  
# msfdb init  
[+] Starting database  
[+] Creating database user 'msf'  
[+] Creating databases 'msf'  
[+] Creating databases 'msf_test'  
[+] Creating configuration file '/usr/share/metasploit-framework/config/database.yml'  
[+] Creating initial database schema
```

```
(root@kali)-[/home/kali]  
# service postgresql start
```

Una vez hecho, repetí el comando “db_status” para comprobar que ya se había conectado la base de datos y efectivamente ponía que se había conectado a msf que es la base de datos por defecto que se crea.

```
msf6 > db_status  
[*] Connected to msf. Connection type: postgresql.
```

3. Desde línea de comando Linux:

Ahora desde otra terminal de la misma máquina o desde la misma terminal (de la cual habría que salirse de Metasploit) ejecuté el comando “sudo su postgres -c psql” y abrirá en forma root (super administrador) el programa postgres.

Una vez dentro de postgres alteré la contraseña por la que quise con el comando “alter user postgres with password 'kali';” y dirá que ya hemos alterado el role

Por último, salí de postgres con \q y enter



```
(kali㉿kali)-[~]  
$ sudo su postgres -c psql  
[sudo] password for kali:  
could not change directory to "/home/kali": Permission denied  
psql (15.3 (Debian 15.3-0+deb12u1))  
Type "help" for help.  
  
postgres=# alter user postgres with password 'kali';  
ALTER ROLE  
postgres=# ALTER ROLE
```

Ahora antes de salir de postgres voy a crear la base de datos a la que en el punto 4 me voy a conectar. Para ello escribo CREATE DATABASE gon_ufv; para crear mi base de datos y para comprobar que se ha creado \list o \l para que nos aparezca una lista completa de todas las bases de datos del sistema en la que estará la nueva que hemos creado.

```
postgres=# CREATE DATABASE gon_ufv;  
CREATE DATABASE  
postgres=# \list
```

List of databases						
Name	Owner	Encoding	Collate	Ctype	ICU Locales	
Access privileges						
gon_ufv	postgres	UTF8	en_US.UTF-8	en_US.UTF-8		

Para salir de vuelta a la terminal se haría con el comando \q

```
postgres-# \q
```



4. Desde Metasploit

Metasploit:

De vuelta en Metasploit ejecuto el comando “db_connect postgres:password@127.0.0.1/test_ufv”

Hago un db_status para saber si estoy conectado, al estar conectado a otra base de datos lo desconecto con db_disconnect porque sino no me dejara crear otra conexión. Y ahora ya conecto la base de datos que he creado con el comando “db_connect postgres:kali@127.0.0.1/gon_ufv”

```
msf6 > db_status
[*] Connected to msf. Connection type: postgresql.
msf6 > db_disconnect
Successfully disconnected from the data service: local_db_service.
msf6 > db_connect postgres:kali@127.0.0.1/gon_ufv
[*] Connected to Postgres data service: 127.0.0.1/gon_ufv
```

Terminal:

Por otra parte fuera de metasploit, voy a ejecutar el nmap para que haga el resultado en un fichero xml. Para ello primero me moví con “cd” a la carpeta donde quiero que se guarde mi archivo xml y una vez dentro ejecuté el comando “nmap -oX output.xml as.com” en el que con nmap abrimos la herramienta, con -oX establecemos que vamos a pasarle el fichero xml que queremos que se cree y después pasamos la página web

```
(kali@kali)-[~]
$ cd Downloads

(kali@kali)-[~/Downloads]
$ nmap -oX output.xml as.com
Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-15 17:25 EST
Nmap scan report for as.com (5.255.145.192)
Host is up (0.027s latency).
Other addresses for as.com (not scanned): 5.255.145.161 5.255.145.160 5.255.145.136 185.43.18
1.34 5.255.145.202 5.255.145.203 5.255.145.147 2a02:26f0:980:f::b81f:804 2a02:26f0:980:f::b81
f:809 2a02:26f0:980:f::b81f:828
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 10.68 seconds

(kali@kali)-[~/Downloads]
$ ls
output.xml
```



Metasploit:

Ahora vuelta en metasploit, importo el resultado desde Metasploit con el comando “db_import (enlace a archivo) output.xml”

```
msf6 > db_import /home/kali/Downloads/output.xml
[*] Importing 'Nmap XML' data
[*] Import: Parsing with 'Nokogiri v1.13.10'
[*] Importing host 5.255.145.192
[*] Successfully imported /home/kali/Downloads/output.xml
```

Y ya se habría importado en la base de datos. Para comprobarlo examinaré los siguientes comandos

5. Una vez importado, interactuar con los comandos

db_hosts o **hosts** hace una lista de las máquinas que se encuentran en la base de datos. Al ejecutarla se puede ver que el host que hay coincide con el que acabo de importar en el paso anterior.

```
msf6 > hosts

Hosts
=====

address      mac      name      os_name      os_flavor      os_sp      purpose      info      comments
-----
5.255.145.192      Unknown      device
```

Para editar información se pueden ejecutar comandos como por ejemplo “hosts -n nombre address” para cambiar el nombre, en este caso voy a ponerle el nombre de la página web que utilice:

```
msf6 > hosts -n As.com 5.255.145.185
msf6 > hosts

Hosts
=====

address      mac      name      os_name      os_flavor      os_sp      purpose      info      comments
-----
5.255.145.185      As.com      Unknown      device
```


db_services o **services** muestra información de los servicios de las máquinas analizadas, puertos abiertos y protocolos. Al ejecutarlo también podemos ver el host que hemos importado

```
msf6 > services
Services
=====
```

host	port	proto	name	state	info
5.255.145.192	80	tcp	http	open	
5.255.145.192	443	tcp	https	open	

Como en el db_hosts también se pueden ejecutar comandos. Por ejemplo se pueden mostrar solo las columnas que se deseen con el comando "services -c columna1,columna2,columna3".

```
msf6 > services -c port,name,state
Services
=====
```

host	port	name	state
5.255.145.185	80	http	open
5.255.145.185	443	https	open

db_notes o **notes** hace un comando que permite ver las notas que se han generado en los comandos de escaneo entre otros, retorna información muy útil en especial sobre los comandos nmap ejecutados

```
msf6 > notes
Notes
=====
```

Time	Host	Service	Port	Protocol	Type	Data
2023-11-15 22:27:27 UTC	5.255.145.192		92		host.imported	{:filename=>"/home/kali/Downloads/output.xml", :type=>"Nmap XML", :time=>2023-11-15 22:27:27 .556834404 UTC}



6. Ejecutar nmap y que los datos se guarden en la base de datos

Para recopilar información es posible utilizar Nmap desde la propia consola de Metasploit. Para ello, basta con solo invocar el comando “db_nmap”. Los parámetros que pueden utilizarse son los mismos que acepta Nmap. De esta manera, los resultados serán almacenados en la base de datos de Metasploit.

Ejecutar db_nmap contra una IP en la que pondré la IP de la máquina virtual de Windows XP

Para saber la IP en Windows ejecutamos un “ipconfig”

```
Adaptador Ethernet Conexión de área local :  
Sufijo de conexión específica DNS : Home  
Dirección IP. . . . . : 192.168.1.74  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada : 192.168.1.1
```

```
msf6 > db_nmap 192.168.1.74  
[*] Nmap: Starting Nmap 7.94 ( https://nmap.org ) at 2023-11-22 07:49 EST  
[*] Nmap: Nmap scan report for 192.168.1.74  
[*] Nmap: Host is up (0.0075s latency).  
[*] Nmap: Not shown: 997 closed tcp ports (reset)  
[*] Nmap: PORT      STATE SERVICE  
[*] Nmap: 135/tcp open  msrpc  
[*] Nmap: 139/tcp open  netbios-ssn  
[*] Nmap: 445/tcp open  microsoft-ds  
[*] Nmap: MAC Address: 34:60:F9:19:B1:44 (TP-Link Limited)  
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 0.79 seconds
```

Si ejecutamos hosts nos saldrá toda la información de todos los sistemas que fueron analizados.

```
msf6 > hosts  
  
Hosts  
=====
```

address	mac	name	os_name	os_flavor	os_sp	purpose	info	comments
5.255.145.18			Unknown			device		
192.168.1.73	08:00:27:CB:7E:F5							
192.168.1.74	34:60:f9:19:b1:44		Unknown			device		

Y pasará de la misma manera con el comando services



```
msf6 > services
Services
=====
```

host	port	proto	name	state	info
5.255.145.185	80	tcp	http	open	
5.255.145.185	443	tcp	https	open	
192.168.1.74	135	tcp	msrpc	open	
192.168.1.74	139	tcp	netbios-ssn	open	
192.168.1.74	445	tcp	microsoft-ds	open	

7. Escanear un FTP desde Metasploit

Elegí para escanear el `auxiliary/scanner/ftp/Anonymous` el cual escaneará un rango de direcciones IP en busca de servidores FTP que permitan el acceso anónimo y determinará dónde se permiten permisos de lectura o escritura. Para buscarlo lo hago con el comando `search`.

```
msf6 > search scanner/ftp
Matching Modules
=====
```

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/ftp/anonymous		normal	No	Anonymous FTP Access Detection
1	auxiliary/scanner/ftp/bison_ftp_traversal	2015-09-28	normal	Yes	BisonWare BisonFTP Server 3.5 D
2	auxiliary/scanner/ftp/colorado_ftp_traversal	2016-08-11	normal	Yes	ColoradoFTP Server 1.3 Build 8
3	auxiliary/scanner/ftp/easy_file_sharing_ftp	2017-03-07	normal	Yes	Easy File Sharing FTP Server 3.
4	auxiliary/scanner/ftp/ftp_login		normal	No	FTP Authentication Scanner
5	auxiliary/scanner/ftp/ftp_version		normal	No	FTP Version Scanner
6	auxiliary/scanner/ftp/konica_ftp_traversal	2015-09-22	normal	Yes	Konica Minolta FTP Utility 1.00
7	auxiliary/scanner/ftp/pcman_ftp_traversal	2015-09-28	normal	Yes	PCMan FTP Server 2.0.7 Directory
8	auxiliary/scanner/ftp/titanftp_xcrc_traversal	2010-06-15	normal	No	Titan FTP XCRC Directory Travers

Para usarlo escribo el comando `use` seguido del número que tenga en la búsqueda anterior y una vez dentro escribo `show options` para mostrar las opciones que tiene el módulo y posteriormente configurarlo.

```
msf6 > use 0
msf6 auxiliary(scanner/ftp/anonymous) > show options
Module options (auxiliary/scanner/ftp/anonymous):
```

Name	Current Setting	Required	Description
FTPPASS	mozilla@example.com	no	The password for the specified username
FTPUSER	anonymous	no	The username to authenticate as
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/
RPORT	21	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads (max one per host)

Para cambiar la opción del RHOST (máquina objetivo) escribo el comando “set RHOST IP”

Y para ejecutar el módulo con la configuración escribo el comando “run” y nos dirá que la ejecución se ha completado.

```
msf6 auxiliary(scanner/ftp/anonymous) > set RHOST 192.168.1.74
RHOST => 192.168.1.74

msf6 auxiliary(scanner/ftp/anonymous) > run

[*] 192.168.1.74:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

8. Escanear un SSH desde Metasploit

Se hace de la misma manera que el FTP, pero con otro módulo.

Para buscarlo lo hago con search modulo

Voy a escanear el módulo el auxiliary/scanner/ssh/ssh_login el cual sirve para probar un conjunto de credenciales en un rango de direcciones IP y también puede realizar intentos de inicio de sesión por fuerza bruta.

```
msf6 > search scanner/ssh

Matching Modules
-----
#  Name
-  -
0  auxiliary/scanner/ssh/apache_karaf_command_execution 2016-02-09 normal No Apache Karaf Default Cre
ls Command Execution
1  auxiliary/scanner/ssh/karaf_login                    2016-02-09 normal No Apache Karaf Login Utili
2  auxiliary/scanner/ssh/cerberus_sftp_enumusers        2014-05-27 normal No Cerberus FTP Server SFTP
ame Enumeration
3  auxiliary/scanner/ssh/eaton_xpert_backdoor           2018-07-18 normal No Eaton Xpert Meter SSH Pr
Key Exposure Scanner
4  auxiliary/scanner/ssh/fortinet_backdoor              2016-01-09 normal No Fortinet SSH Backdoor Sc
5  auxiliary/scanner/ssh/juniper_backdoor               2015-12-20 normal No Juniper SSH Backdoor Sca
6  auxiliary/scanner/ssh/detect_kippo                   normal No Kippo SSH Honeypot Detec
7  auxiliary/scanner/ssh/ssh_login                       normal No SSH Login Check Scanner
8  auxiliary/scanner/ssh/ssh_identity_pubkeys           normal No SSH Public Key Acceptanc
ner
```

Una vez buscado el módulo voy a escribir “use” más el número de la búsqueda del paso anterior y después una vez dentro escribo “show options” para mostrar las opciones



```
msf6 > use 7
msf6 auxiliary(scanner/ssh/ssh_login) > show info

Name: SSH Login Check Scanner
Module: auxiliary/scanner/ssh/ssh_login
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Check supported:
  No

Basic options:
```

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 t
DB_ALL_PASS	false	no	Add all passwords in the current database
DB_ALL_USERS	false	no	Add all users in the current database to
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the c
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/basics/using-metasploit.html
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for
THREADS	1	yes	The number of concurrent threads (max one
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separ
USER_AS_PASS	false	no	Try the username as the password for all
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

Para cambiar la opción del RHOST (máquina objetivo) escribo el comando “set RHOST IP”

Y para ejecutar el módulo con la configuración escribo el comando “run” y nos dirá que la ejecución se ha completado.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.1.74
RHOSTS => 192.168.1.74

msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 192.168.1.74:22 - Starting bruteforce
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



Conclusiones

En esta práctica básica de Metasploit se han aprendido las herramientas básicas de este framework como el uso de la base de datos PostgreSQL y la conexión desde Metasploit junto a la verificación del estado del framework en relación con la base de datos. También se ha estudiado la herramienta Nmap para poder ver resultados importados y gestionados en Metasploit. Y se ha interactuado con la base de datos utilizando comandos específicos, y la ejecución de escaneos de servicios FTP y SSH, cargando módulos auxiliares, configurando variables y realizando los escaneos correspondientes.



Final del documento



Metasploit 1

Laboratorio Básico