C|EH Certified Ethical Hacker

Ethical Hacking and Countermeasures
Metasploit Cheat Sheet

EC-Council
Building A Culture Of Security

Metasploit
Source:
https://www.metasploit.com

Metasploit is an open-source project that provides the infrastructure, content, and tools to perform penetration tests and extensive security auditing. It is a tool that provides information about security vulnerabilities and aids in penetration testing. Metasploit framework is also used for developing and executing exploits which promotes in gaining remote access to a system by exploiting any vulnerability present in that server. Meterpreter is a payload inside the framework. The following table lists the various Metasploit commands and their respective scanning methods.

1.Metasploit General Information

Metasploit Command	Description
`msfconsole`	Launch program
`version`	Display current version
`msfupdate`	Pull weekly update
`makerc <FILE.rc>`	Saves recent commands to file
`msfconsole -r <FILE.rc>`	Loads resource file

2. Executing an Exploit / Scanner / Module

Metasploit Command	Description
`use <MODULE>`	Set the exploit to use
`set payload <PAYLOAD>`	Set the payload
`show options`	Show all options
`set <OPTION> <SETTING>`	Set setting
`exploit or run`	Execute exploit

3. Session Handling

Metasploit Command	Description
`sessions -l`	List all sessions
`sessions -i <ID>`	Interact to session
`background or ^Z`	Detach from session

4. Using Database

i.First Time Setup (Linux command line)

Metasploit Command	Description
`service postgresql Start`	List all sessions
`msfdb Init`	Init database

ii. Inside msfconsole

Metasploit Command	Description
`db_status`	Should display connected
`hosts`	Display hosts in database
`services`	Show ports in database
`vulns`	Exhibit all vulnerabilities

5. Meterpreter Session Commands

i.Base commands

Metasploit Command	Description
`sysinfo`	Display system name and OS type
`shutdown / reboot`	Shutdown system
`exit / quit`	Exit Meterpreter session

ii. Process Commands

Metasploit Command	Description
`ps`	Show running processes list
`kill <PID>`	Terminate process
`getuid`	Show user ID
`getpid`	Show process ID that Meterpreter is running inside
`migrate <PID>`	Start another process
`execute`	Execute given program with the privileges of the process

iii. File System Commands

Metasploit Command	Description
`pwd / lpwd/getwd`	Display current working directory (local / remote)
`cd`	Change directory
`lcd`	Change directory (local)
`mkdir`	Make directory
`rmdir`	Remove directory
`cat`	Show contents of a file
`edit <FILE>`	Edit a file in default editor (vi)
`upload / download`	Upload / download a file from target machine

iv. Escalate Privileges

Metasploit Command	Description
`use priv`	Load script
`getsystem`	Gain administrative-level privileges
`getprivs`	Elevate privileges

v. Networking Commands

Metasploit Command	Description
`ipconfig`	Show network interface information
`route`	Manage/view the system's routing table
`c`	Forward packets through TCP session
`route add <Target IP/ Subnet>`	Pivot through session by adding route in MSF
`route add <Target IP/ Subnet> -d`	Delete route inside MSF
`sniffer`	Allow network sniffing interaction commands
`portfwd`	Port forwarding connections
`portfwd -L`	Local host to listen
`portfwd -l`	Local port to listen
`portfwd -p`	Remote port to connect
`portfwd -r`	Remote host to connect

vi. Additional Commands

Metasploit Command	Description
`shell`	Drop into a shell on the target machine
`hashdump`	Show all password hashes in Windows
`idletime`	Display idle time of the machine
`screenshot`	Save the screenshot
`clearev`	Clear the logs
`uictl [enable/disable] [keyboard/mouse]`	Enable or disable the mouse or keyboard of the machine
`use`	Extension load
`channel`	Display active channel
`reg`	Access machine registry
`steal_token`	Attempts to steal impersonation token from target
`espia`	Desktop spying by screenshots
`incognito`	Impersonation commands
`msf> search`	Search for any module
`msf > use exploit`	Specify and exploit to use

6. Session Management

Metasploit Command	Description
`msf > exploit -z`	Run exploit in background expecting one session
`msf > session -i [SessionID]`	Interact with backgrounded session
`msf > exploit -j`	Run exploit in background expecting one or more sessions
`msf > sessions -l`	List all backgrounded sessions
`msf > jobs -l`	List all current jobs
`msf > jobs -k [JobID]`	Kills job

`meterpreter > <Ctrl+Z> /`
`meterpreter > background`
	Background current interactive session

7. Interface / Output Commands

Metasploit Command	Description
`enumdesktops`	Display all existing desktops
`getdesktop`	Display current desktop
`keyscan_start`	Start keylogger in target machine
`keyscan_stop`	Stop keylogger in target machine
`set_desktop`	Configure desktop
`keyscan_dump`	Dump keylogger content

8. Msfvenom Command Options

Metasploit Command	Description
`-p (Payload option)`	Show payload standard options
`-l (list type)`	List module type
`-f (format)`	Output format
`-e(encoder)`	Define which encoder to use
`-a (Architecture or platform`	Define which platform to use
`-s (Space)`	Define maximum payload capacity
`-b (characters)`	Define set of characters not to use
`-i (Number of times)`	Define number of times to use encoder
`-x (File name)`	Define a custom file to use as template
`-o (output)`	Save payload
`-h`	Help

9. Important Auxiliary Modules

Metasploit Command	Description

`msf > use auxiliary/scanner/portscan/tcp`
`msf > set RHOSTS <Target IP/Subnet>`
`msf > set PORTS 1-1000`
`msf > run`
	Port scanning module

`msf > use auxiliary/gather/dns_enum`
`msf > set DOMAIN target.tgt`
`msf > run`
	DNS Enumeration module

`msf > use auxiliary/server/ftp`
`msf > set FTPROOT /tmp/ftproot`
`msf > run`
	FTP Server module

`msf > use auxiliary/server/socks4`
`msf > run`
	Proxy Server module

`msf > use auxiliary/scanner/snmp/snmp_enum`
`msf > set RHOSTS <Target IP>`
`msf > exploit`
	SNMP Enumeration module

`msf > use auxiliary/scanner/sip/enumerator`
`msf > set RHOSTS <Target IP/Subnet>`
`msf > run`
	SIP Enumeration module

`msf > use auxiliary/scanner/ftp/ftp_version`
`msf > set RHOSTS <Target IP>`
`msf > exploit`
	FTP Enumeration module

`msf > use auxiliary/scanner/discovery/arp_sweep`
`msf > set RHOSTS <Target IP-Range>`
`msf > set SHOSTS <Target IP>`
`msf > set SMAC <MAC Address>`
`msf > set THREADS < Number of concurrent threads>`
`msf > run`
	ARP Sweep module

`msf > use auxiliary/scanner/discovery/ipv6_neighbor`
`msf > set RHOSTS <Target IP-Range>`
`msf > set SHOSTS <Target IP>`
`msf > set SMAC <MAC Address>`
`msf > set THREADS < Number of concurrent threads>`
`msf > run`
	IPV6 Neighbor module

`msf > use auxiliary/scanner/discovery/udp_probe`
`msf > set RHOSTS <Target IP-Range>`
`msf > set THREADS < Number of concurrent threads>`
`msf > run`
	UDP Probe module

`msf > use auxiliary/scanner/discovery/udp_sweep`
`msf > set RHOSTS <Target IP-Range>`
`msf > set THREADS < Number of concurrent threads>`
`msf > run`
	UDP Sweep module

`msf > use auxiliary/scanner/scada/modbus_findunitid`
`msf > set RHOSTS <Target IP>`
`msf > run`
	Scan and detect Modbus Slaves

`msf > use x86/opty2`
`msf nop(opty2) > generate -h`
`Usage: generate [options] length`
	Generates a NOP sled of a given length