Module 14: Hacking Web Applications

Scenario

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

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

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

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

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

Objective

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

- Footprinting a web application using various information-gathering tools
- Performing web spidering, detect load balancers, and identify web server directories
- Performing web application vulnerability scanning
- Performing brute-force and cross-site request forgery (CSRF) attack
- Exploiting remote command execution vulnerability

- Gaining backdoor access via a web shell
- Detecting web application vulnerabilities using various web application security tools

Overview of Web Applications

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

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

Lab Tasks

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

1. Footprint the web infrastructure
    - Perform web application reconnaissance using Nmap and Telnet
    - Perform web spidering using OWASP ZAP
    - Perform web application vulnerability scanning using SmartScanner
2. Perform web application attacks
    - Perform a brute-force attack using Burp Suite
    - Perform Remote Code Execution (RCE) attack
3. Detect web application vulnerabilities using various web application security tools
    - Detect web application vulnerabilities using wapiti web application security scanner
4. Perform Web Application Hacking using AI.
    - Perform web application hacking using ShellGPT.

Lab 1: Footprint the Web Infrastructure

Lab Scenario

The first step in web application hacking for an ethical hacker or pen tester is to gather the maximum available information about the target organization website by performing web

application footprinting using various techniques and tools. In this step, you will use techniques such as web spidering and vulnerability scanning to gather complete information about the target web application.

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

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

Lab Objectives

- Perform web application reconnaissance using Nmap and Telnet
- Perform web spidering using OWASP ZAP
- Perform web application vulnerability scanning using SmartScanner

Overview of Footprinting the Web Infrastructure

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

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

Task 1: Perform Web Application Reconnaissance using Nmap and Telnet

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

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

1. Perform a Whois lookup to gather information about the IP address of the web server and the complete information about the domain such as its registration details, name servers, IP address, and location.

2. Use tools such as **Netcraft** (https://www.netcraft.com), **SmartWhois** (https://www.tamos.com), **WHOIS Lookup** (https://whois.domaintools.com), and **Batch IP Converter** (http://www.sabsoft.com) to perform the Whois lookup.

3. Perform DNS Interrogation to gather information about the DNS servers, DNS records, and types of servers used by the target organization. DNS zone data include DNS domain names, computer names, IP addresses, domain mail servers, service records, etc.

4. Use tools such as, **DNSRecon** (https://github.com), and **Domain Dossier** (https://centralops.net) to perform DNS interrogation.

5. Now, we will perform port scanning to gather information about the open ports and services running on the machine hosting the target website.

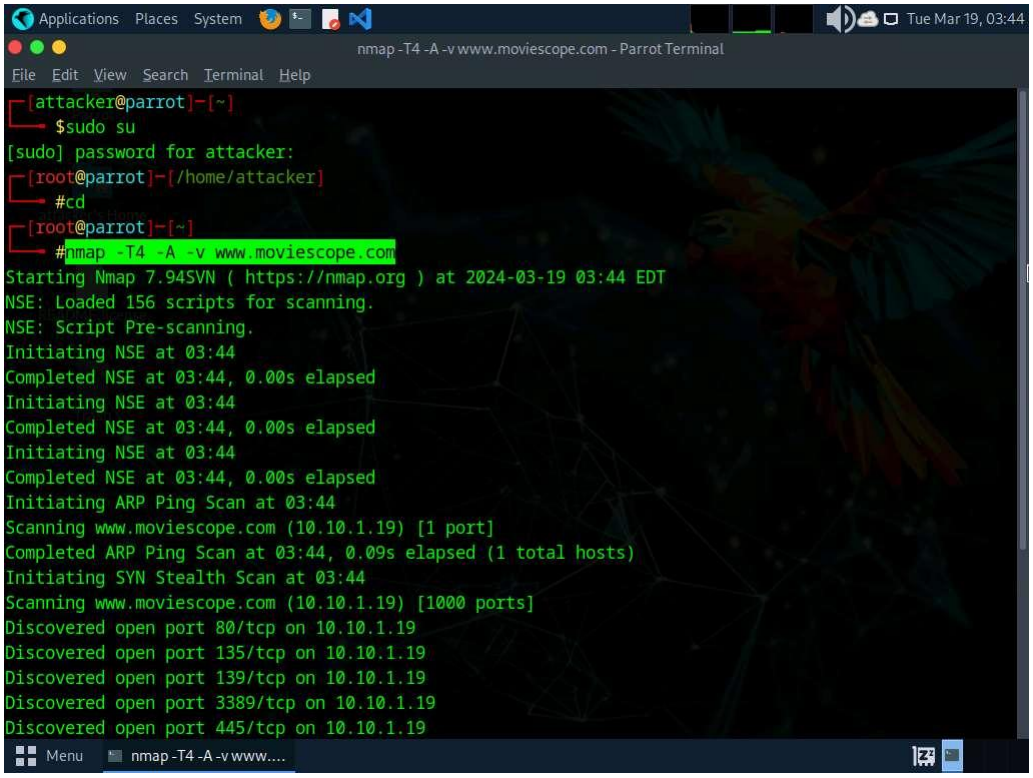
6. Click Parrot Security to switch to the **Parrot Security** machine. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

   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.

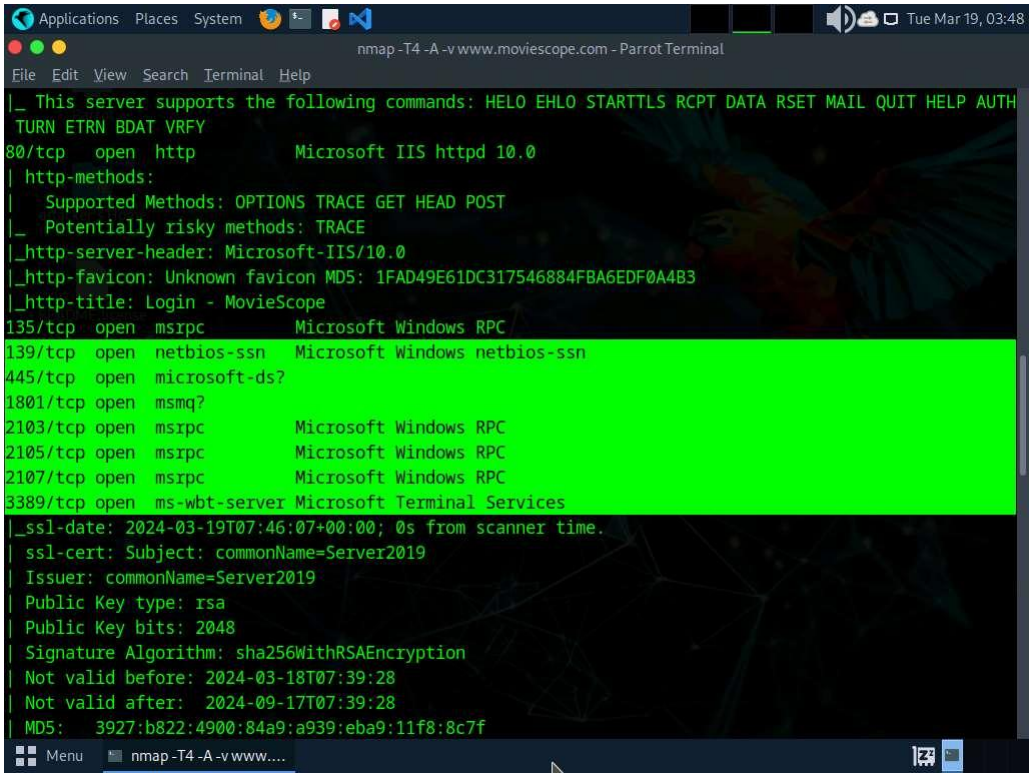
7. Now, type **cd** and press **Enter** to jump to the root directory.

8. In the **Parrot Terminal** window, run **nmap -T4 -A -v [Target Web Application]** command (here, the target web application is **www.moviescope.com**) to perform a port and service discovery scan.

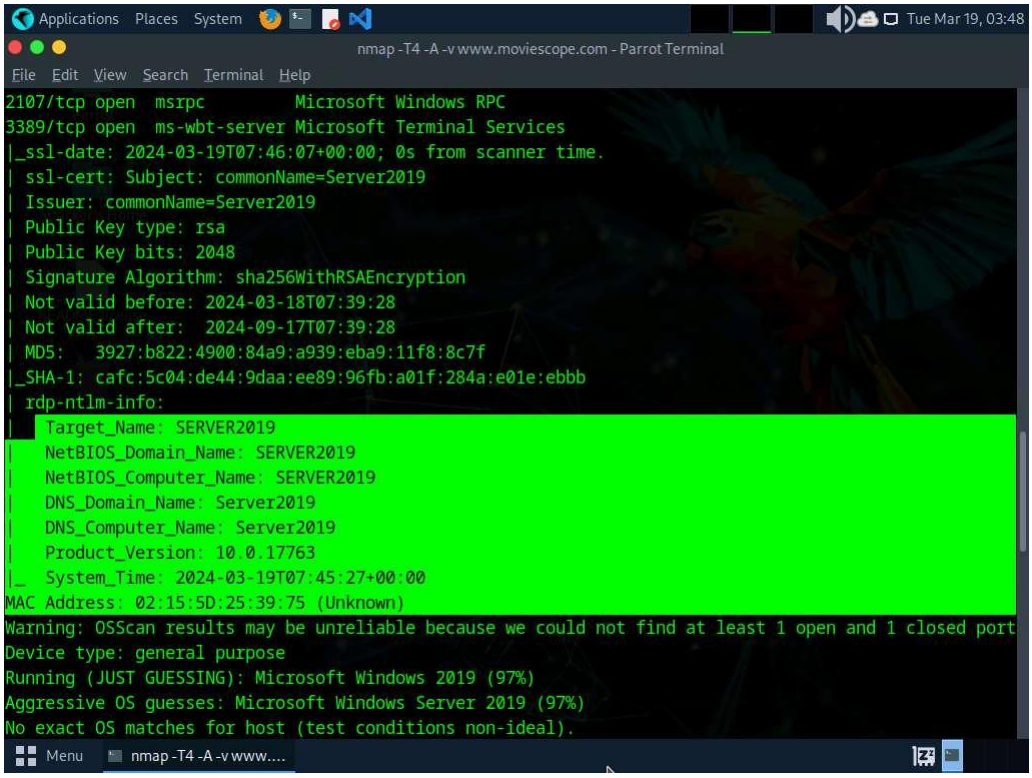
   In this command, **-T4**: specifies setting time template (0-5), **-A**: specifies aggressive scan, and **-v**: enables the verbose output (include all hosts and ports in the output).



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



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

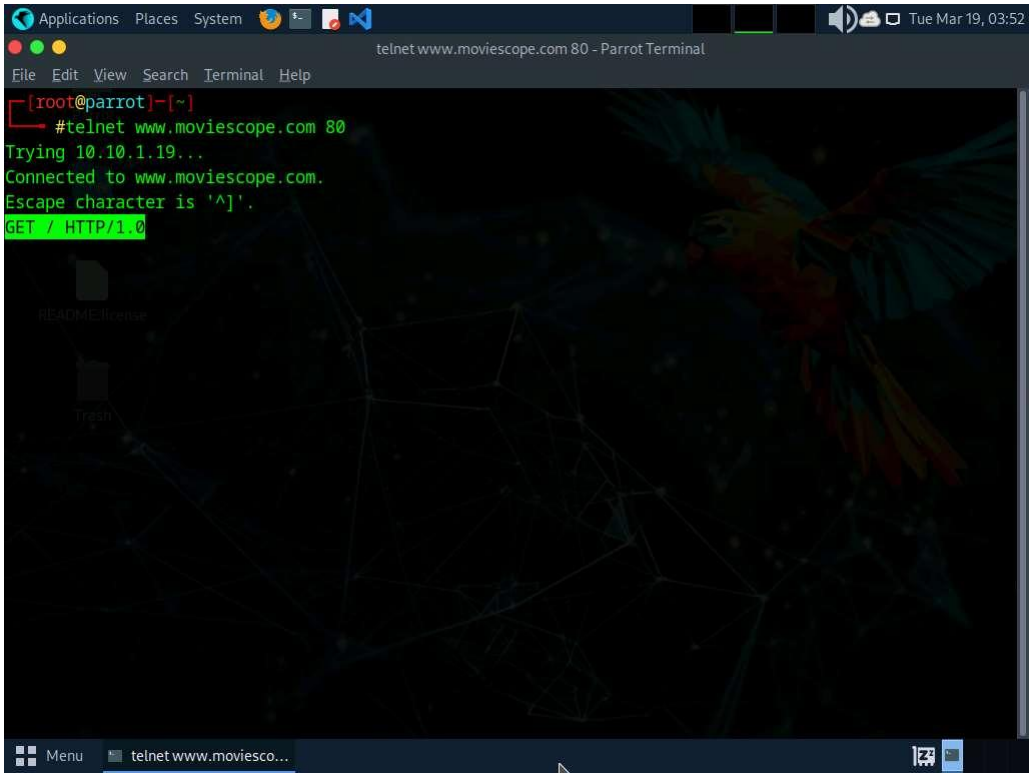


11. Now, perform banner grabbing to identify the make, model, and version of the target web server software.

12. In the terminal window, run command **telnet www.moviescope.com 80** to establish a telnet connection with the target machine.

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

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

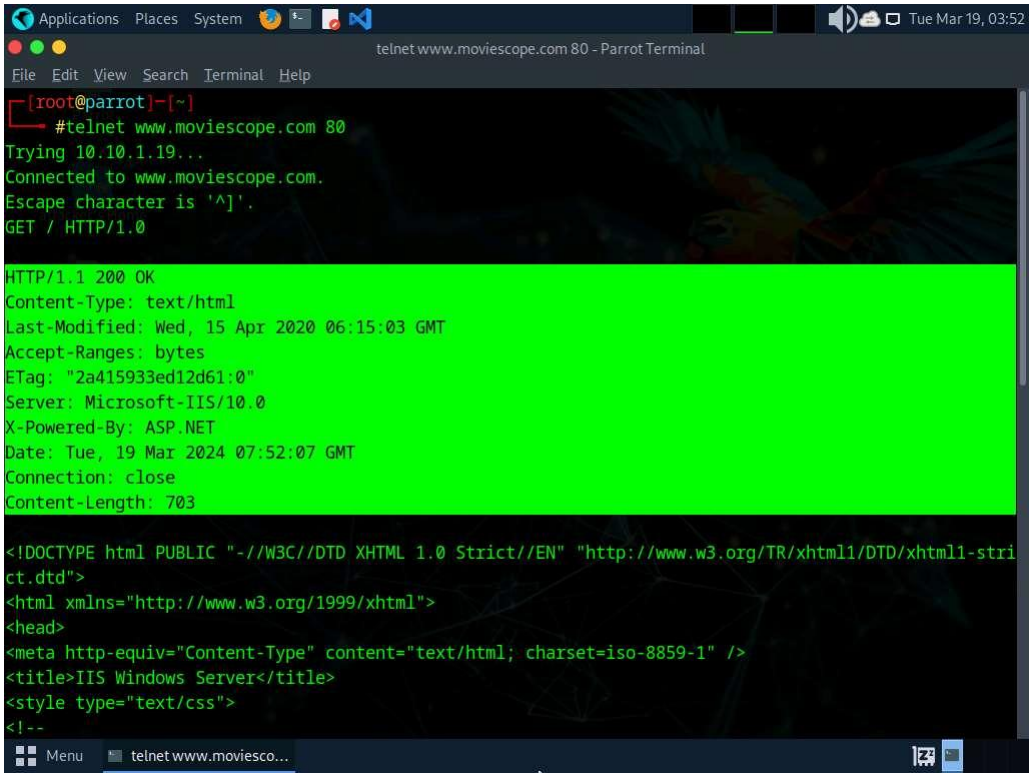


14. The result appears, displaying information related to the server name and its version, technology used.

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

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

    more...



16. This concludes the demonstration of how to perform web application reconnaissance (Whois lookup, DNS interrogation, port and services discovery, banner grabbing, and firewall detection).

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

Question 14.1.1.1

Perform a port and service discovery scan using Nmap on the website www.moviescope.com. Enter the IP address of the machine hosting www.moviescope.com.

Question 14.1.1.2

Perform a scan using Nmap on the website www.moviescope.com. Enter the name of the DNS server hosting the domain name for www.moviescope.com.

Question 14.1.1.3

Perform banner grabbing using Telnet on the website www.moviescope.com to identify the make, model, and version of the target web-server software. Identify the server-side application used to develop the web pages.

Task 2: Perform Web Spidering using OWASP ZAP

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

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

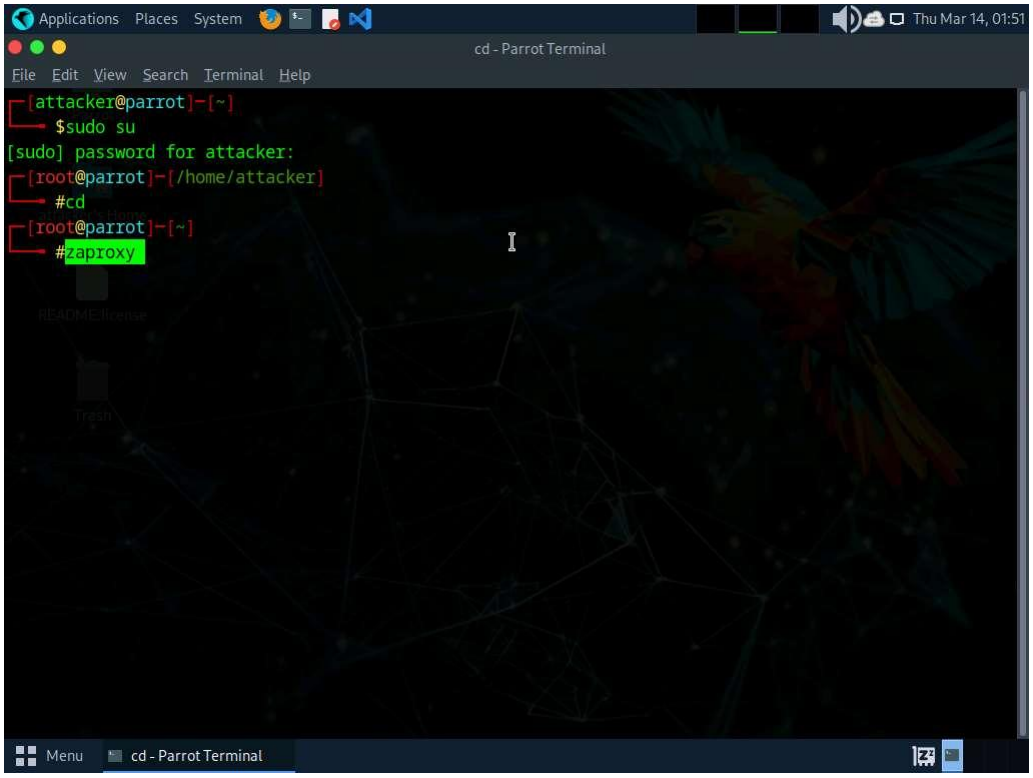
In this task, the target website (**www.moviescope.com**) is hosted by the victim machine (**Windows Server 2019**). Here, the host machine is the **Parrot Security** machine.

1. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

   The password that you type will not be visible.

2. Now, run **cd** command to jump to the root directory.

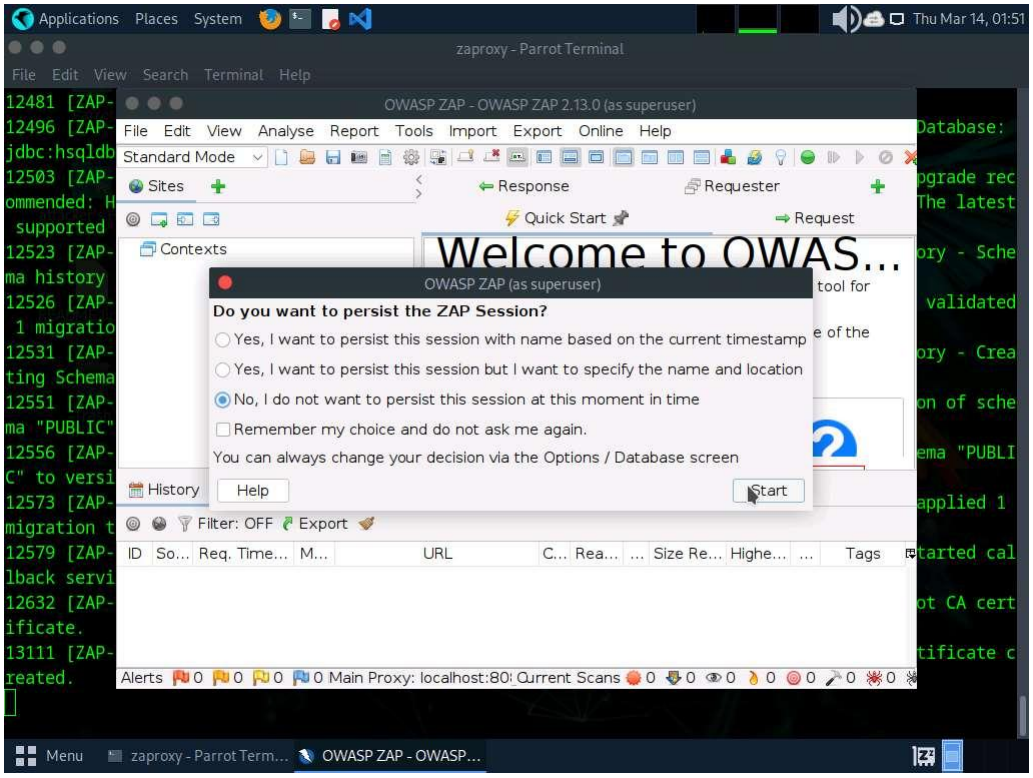
3. In the **Terminal** window, type **zaproxy** and press **Enter** to launch OWASP ZAP.



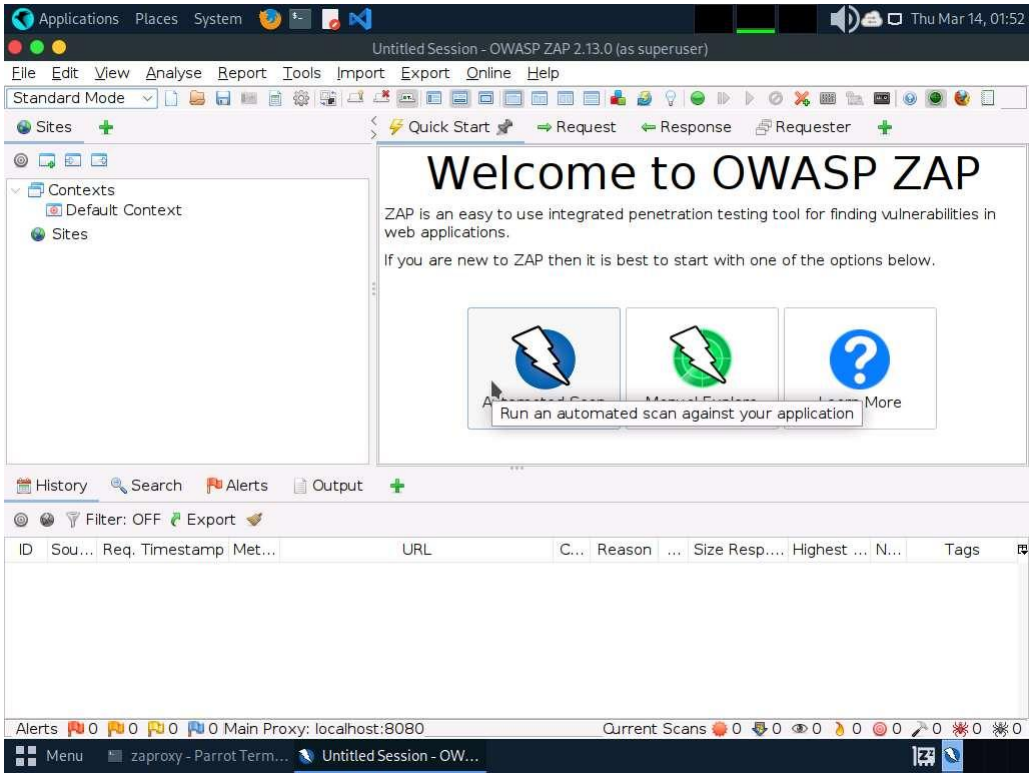
4. The **OWASP ZAP** initializing window appears; wait for it to complete.

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

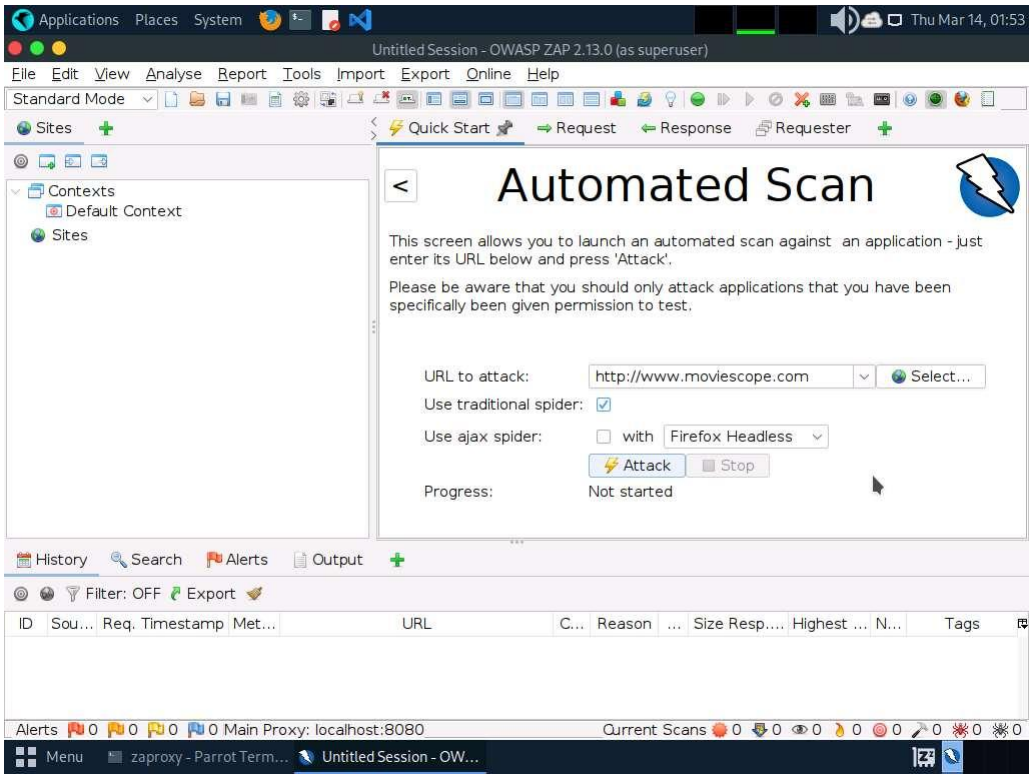
If a **Manage Add-ons** window appears, click the **Close** button.



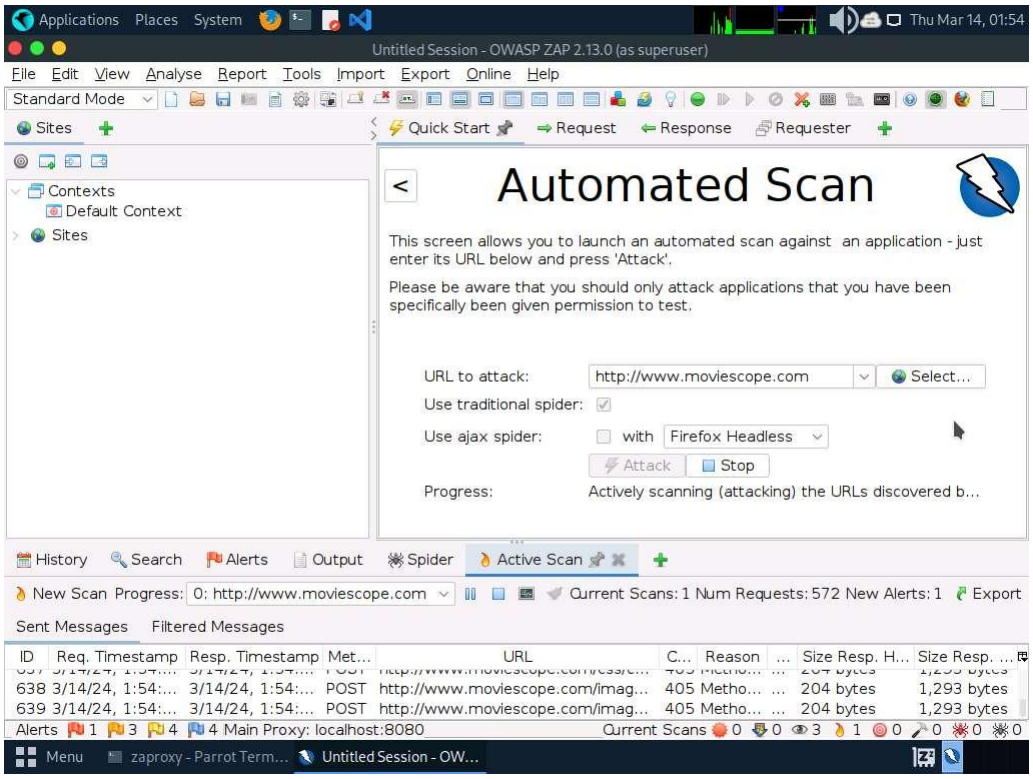
6. The **OWASP ZAP** main window appears. Under the **Quick Start** tab, click the **Automated Scan** option under **Welcome to OWASP ZAP**.



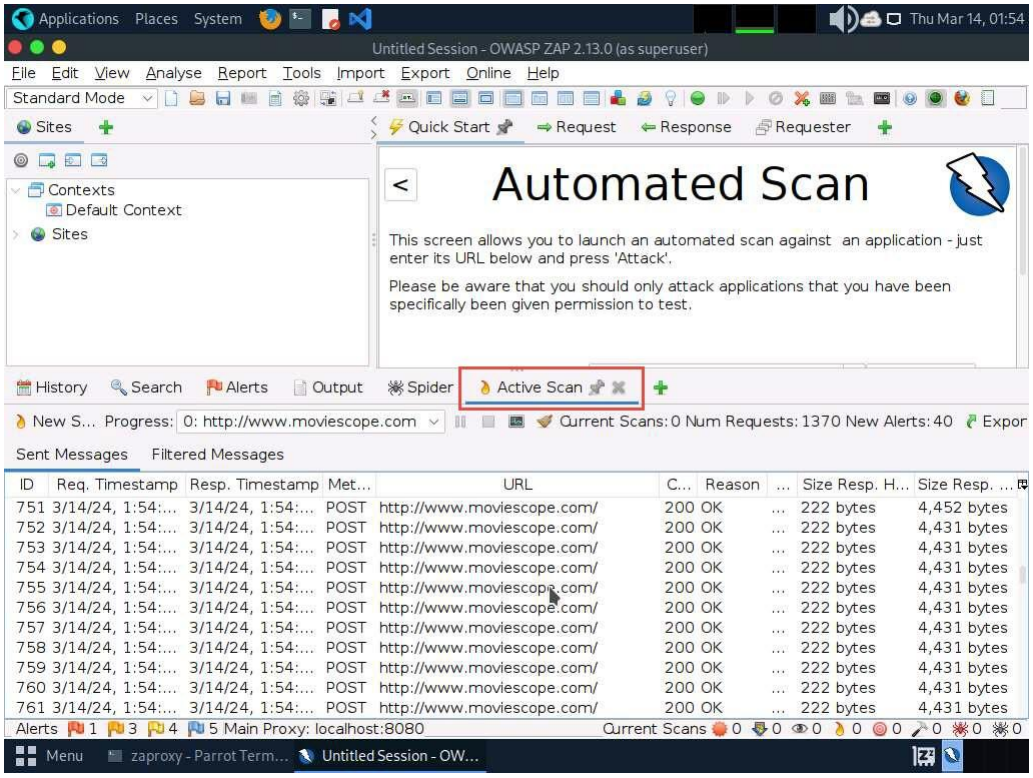
7. The **Automated Scan** wizard appears; enter the target website under the **URL to attack** field (here, **www.moviescope.com**). Leave the other settings to default and click the **Attack** button.



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

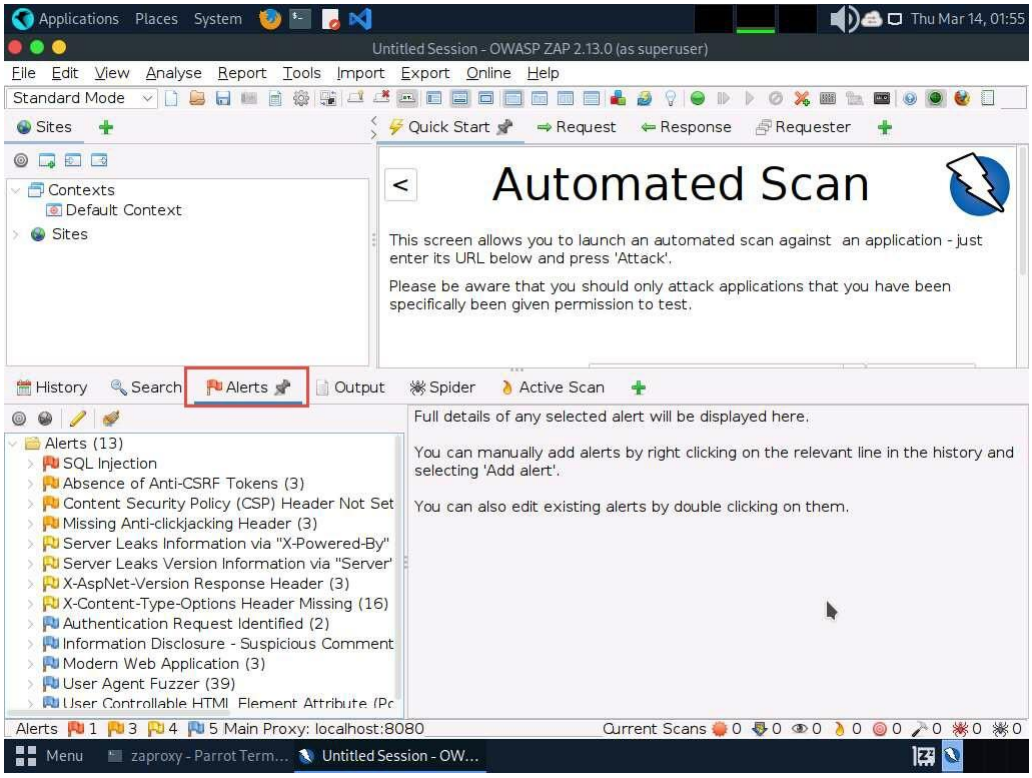


9. After performing web spidering, **OWASP ZAP** performs active scanning. Navigate to the **Active Scan** tab to observe the various scanned links.



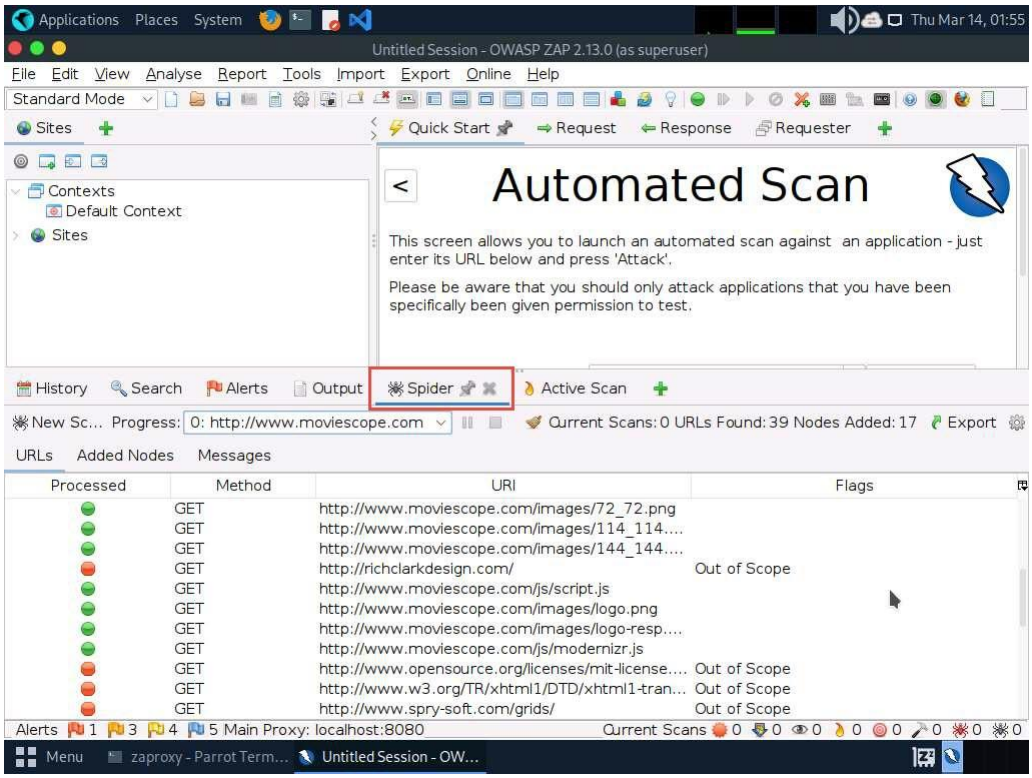
10. After completing the active scan, the results appear under the **Alerts** tab, displaying the various vulnerabilities and issues associated with the target website, as shown in the screenshot.

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



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

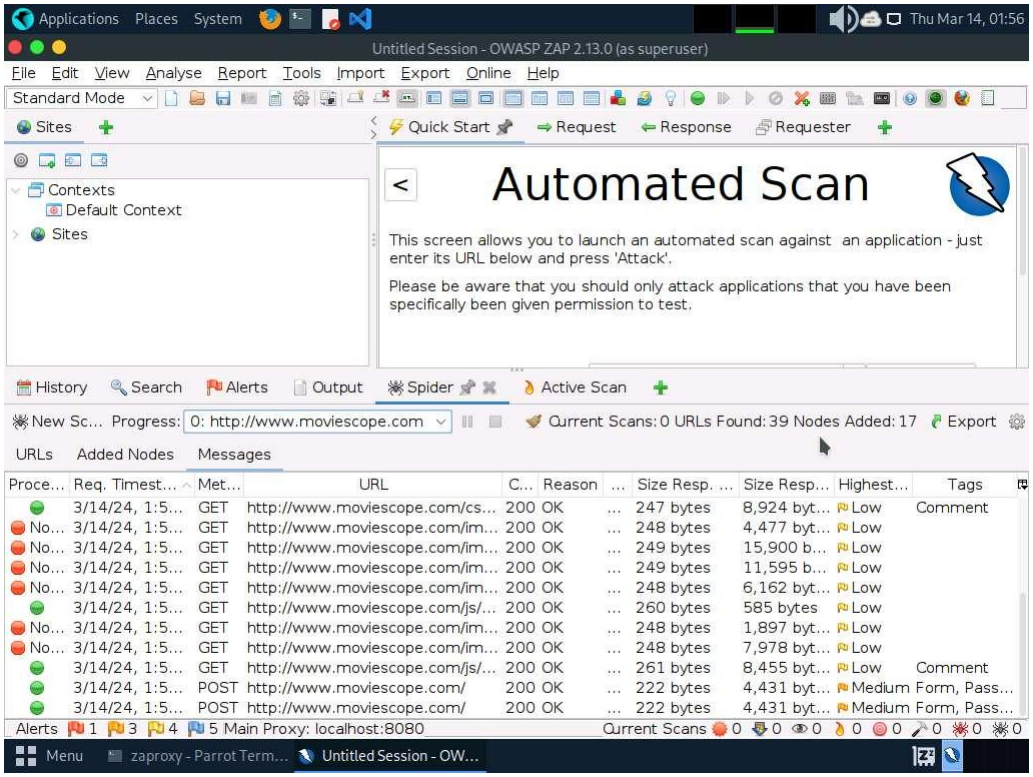
12. The **URLs** tab contains various links for hidden content and functionality associated with the target website (**www.moviescope.com**).



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

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

    more...



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

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

Question 14.1.2.1

Perform web spidering on the www.moviescope.com website using OWASP ZAP. Enter the name of the tab on the OWASP ZAP application that allows you to view detailed information regarding the URLs obtained while performing web spidering.

Task 3: Perform Web Application Vulnerability Scanning using SmartScanner

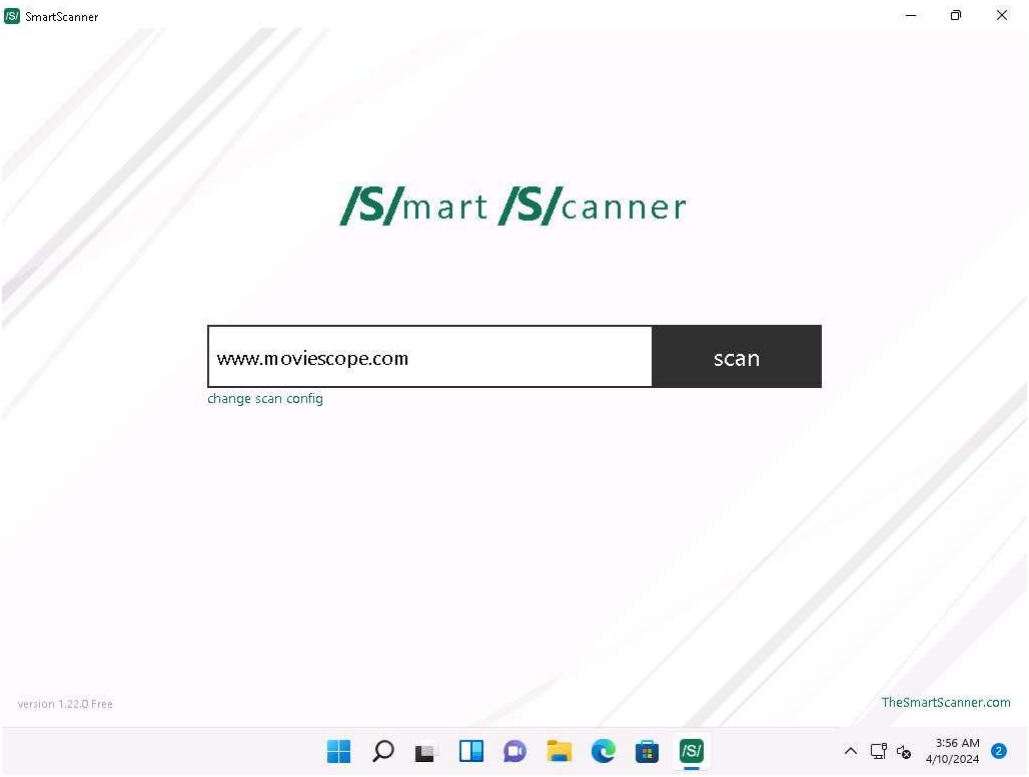
SmartScanner leverages machine learning (ML) and artificial intelligence (AI) techniques to adapt its methodologies to the behavior of the target. This integration allows SmartScanner to minimize false positives. It uses AI for identifying vulnerable pages, detecting 404 custom pages, identifying input vectors, fingerprinting the target and calculating the security risk.

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

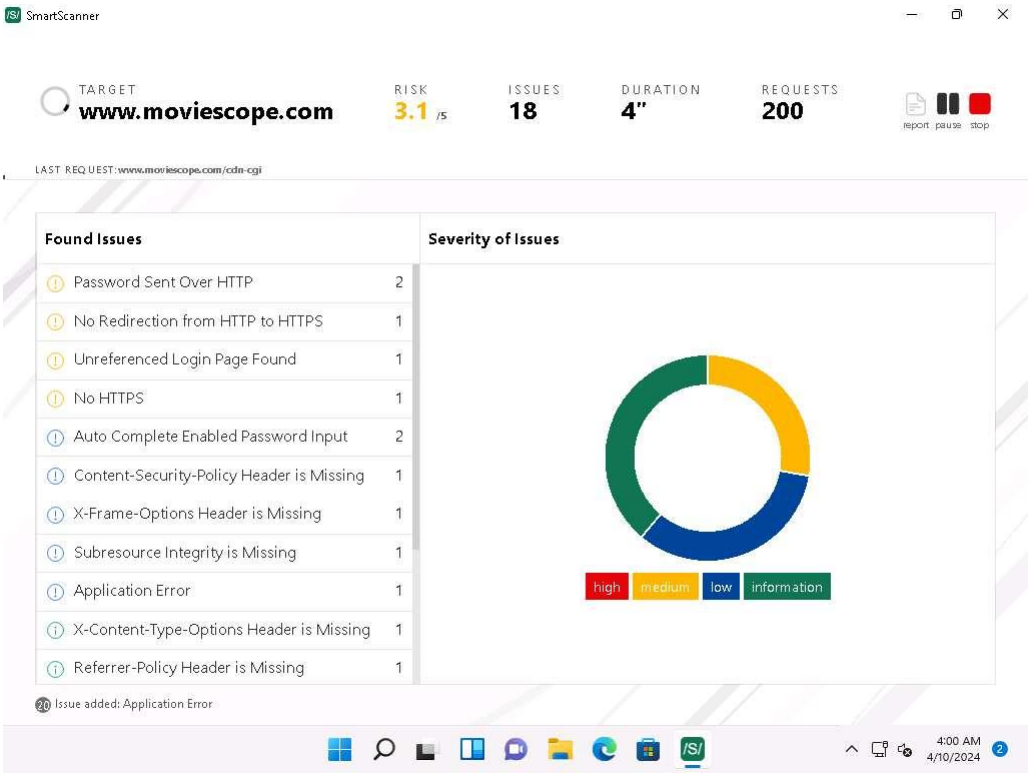
1. Click Windows 11 to switch to the **Windows 11** machine, click Ctrl+Alt+Delete to activate the machine and login using **Admin/Pa$$w0rd**.

2. Click **Search** icon ( ) on the **Desktop**. Search **smartscanner** in the search field, the **SmartScanner** appears in the results, click **Open** to launch it.

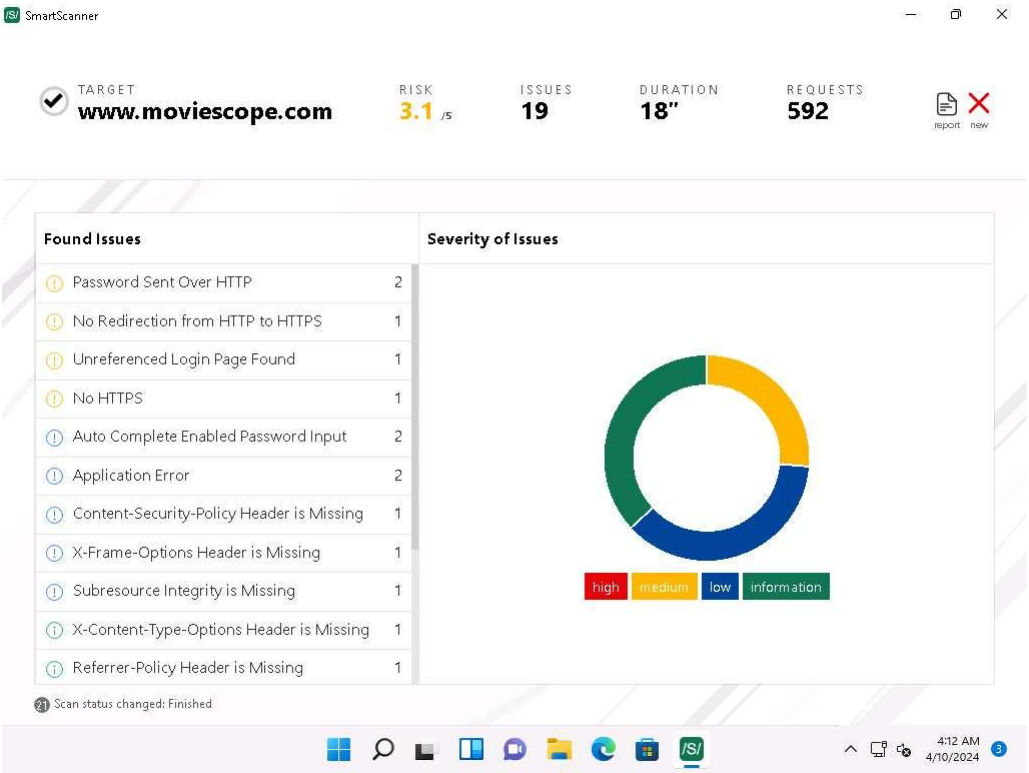
3. **SmartScanner** window appears. In the **enter site address to scan** field, enter **www.moviescope.com** and click **scan** button.



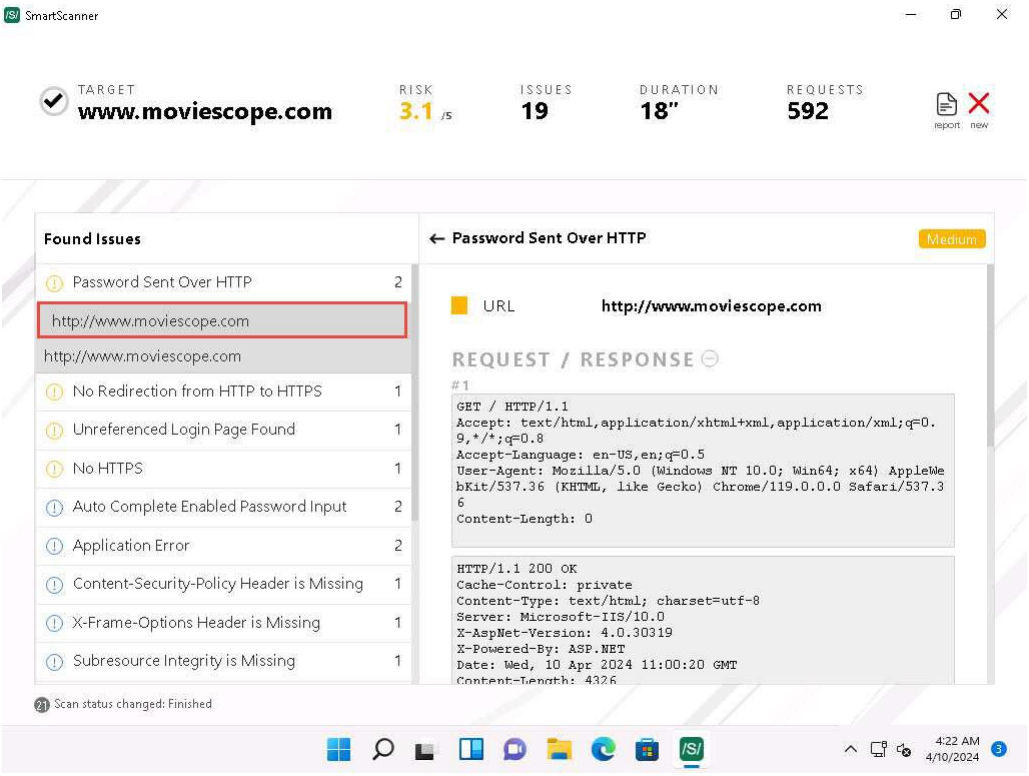
4. The tool starts scanning the target website for vulnerabilities.



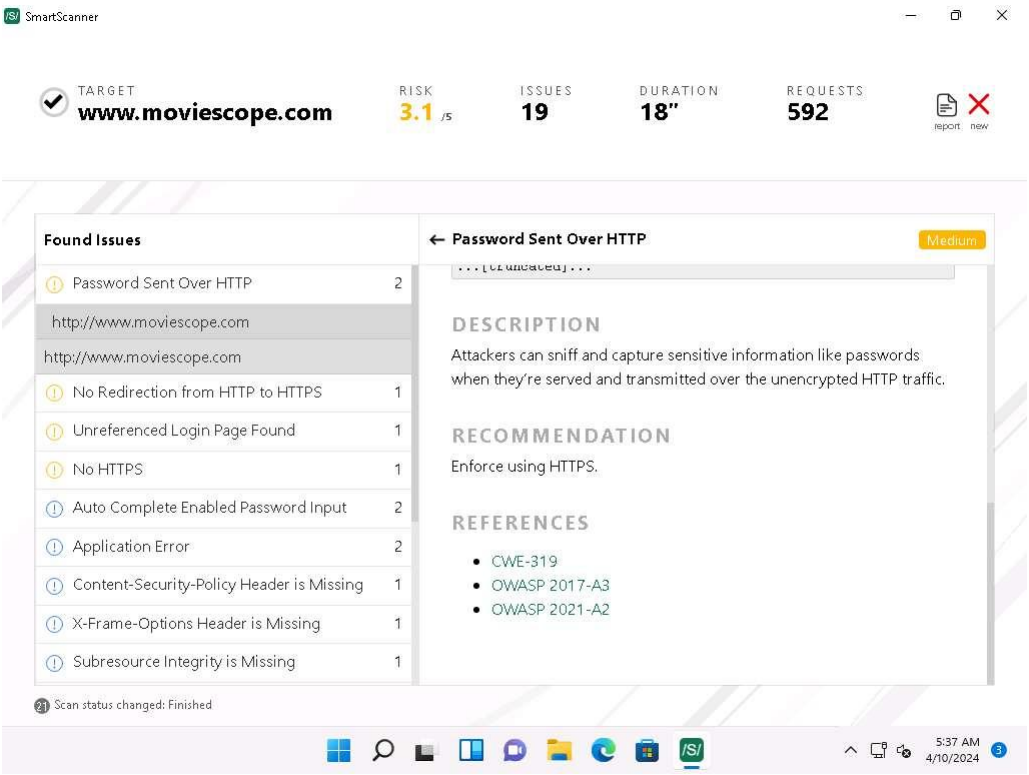
5. Once the tool completes scanning, it will display the issues that are found under **Found Issues** section and **Severity of Issues**.



6. Now, expand **Password Sent Over HTTP** and click on first **http://www.moviescope.com** link from the left pane to view the details of the vulnerability.



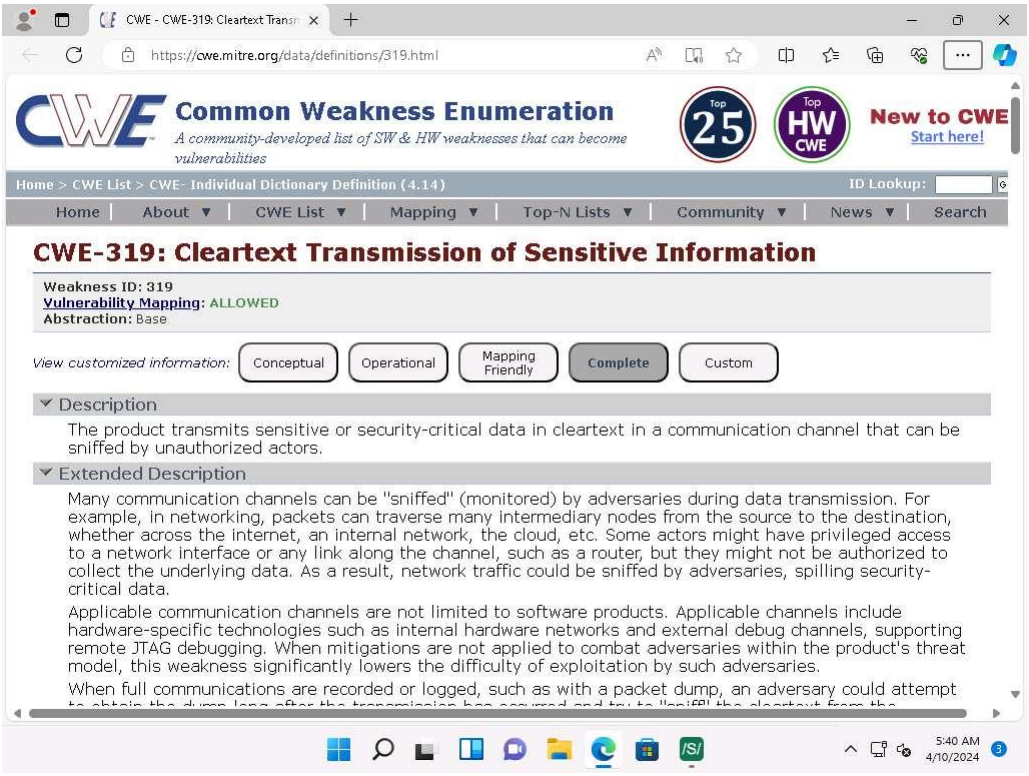
7. In the right pane, scroll down to the **DESCRIPTION** part. We can observe that this website contains a vulnerability, which could be exploited by attackers to intercept sensitive information like passwords during transmission over unencrypted HTTP traffic.



8. You can also go through the **RECOMMENDATION** section to check for the recommended actions to patch the vulnerability.

9. Now, under **REFERENCES** section, press **Ctrl** and click on **CWE-319** hyperlink .

10. A CWE website appears in **Microsoft Edge** web browser, displaying the details of **CWE-319 ClearText Transmission of Sensitive Information**.

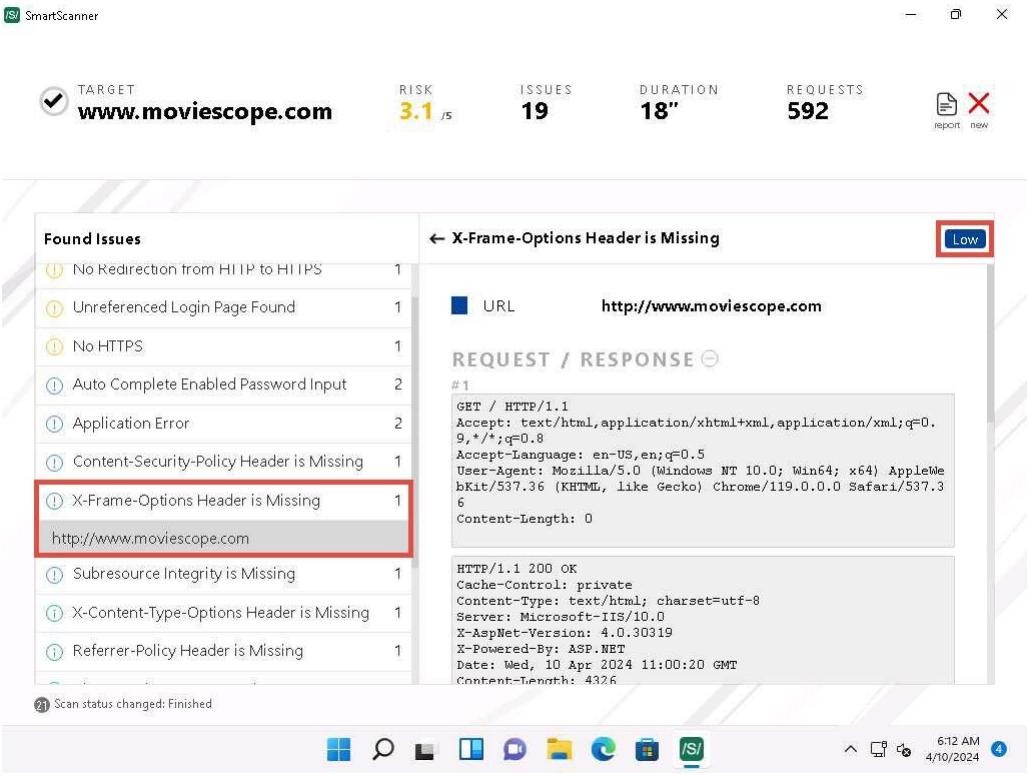


11. In the CWE page, we can see that the attackers can gather sensitive information such as passwords etc. by sniffing the network, if the information is transmitted in cleartext format.

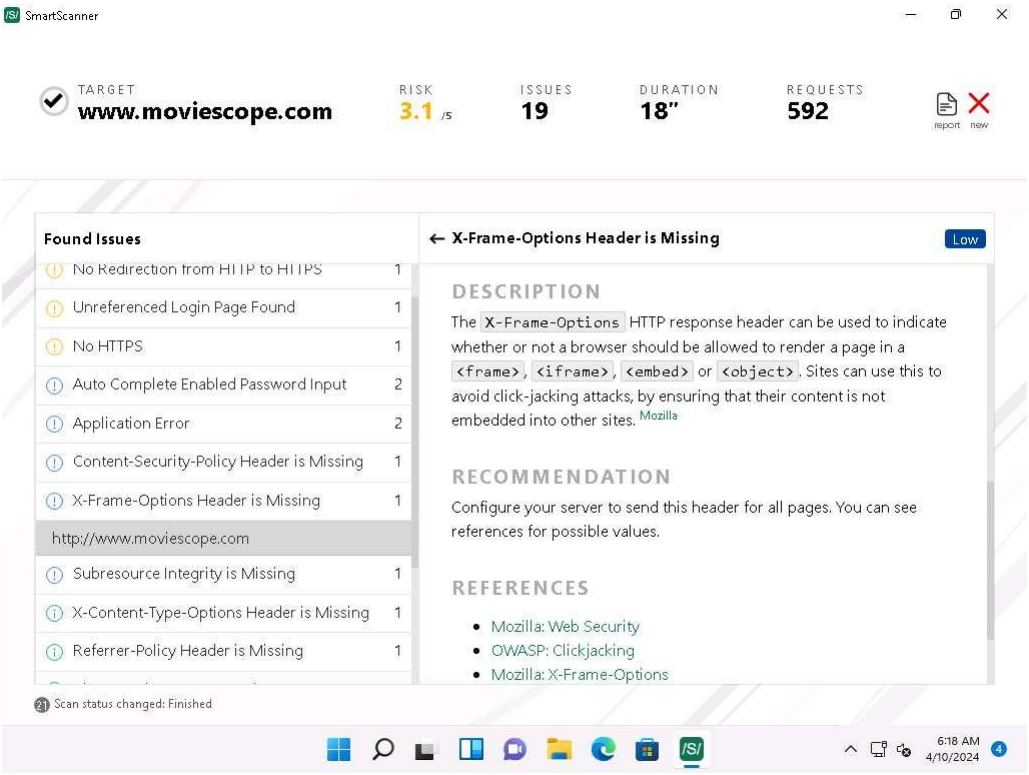
    We have already performed a lab about **Password Sniffing using Wireshark** in **Module 08: Sniffing**.

12. Close the browser window and switch to the SmartScanner window.

13. Similarly, click the **http://www.moviescope.com** link available under **X-Frame-Options Header is Missing** node which is termed as **Low** severity.



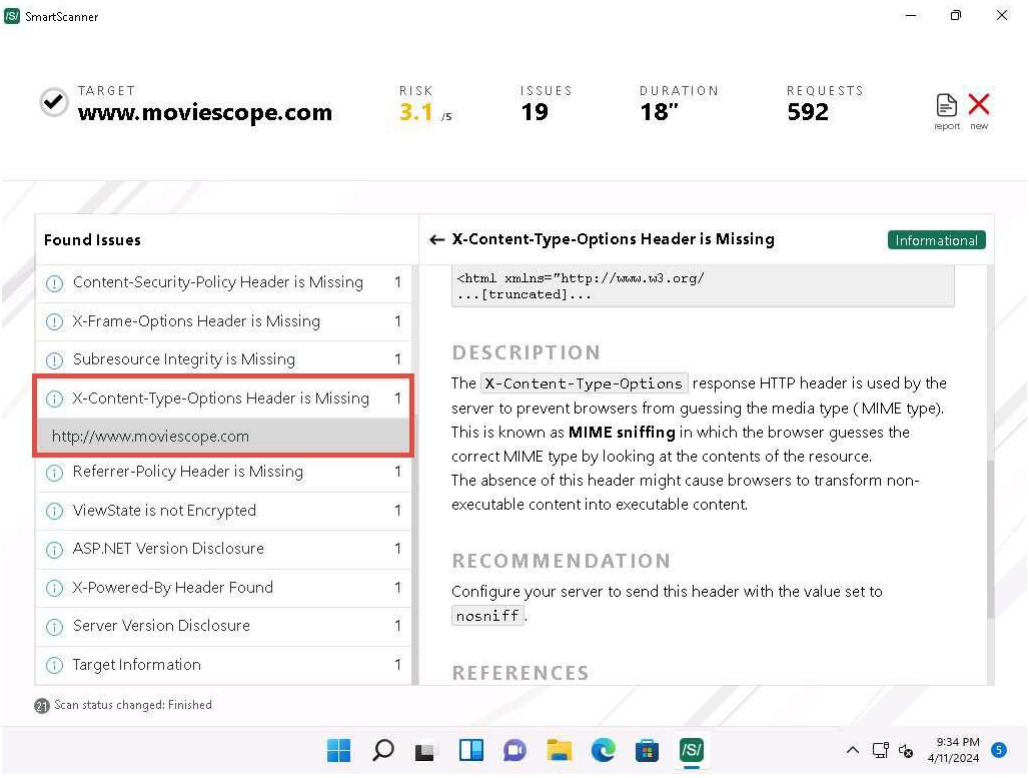
14. Scroll down to the **DESCRIPTION** here, we can observe that the **X-Frame-Options Header is Missing** which will make this site vulnerable to click-jacking.



15. Similarly, you can view the **RECOMMENDATION** section and click on the reference link under **REFERENCES** section.

16. Now, expand **X-Content-Type-Options Header is Missing** node and click on **http://www.moviescope.com** link to view its contents.

17. Under **DESCRIPTION** section we can observe that the browsers can perform **MIME sniffing** which can cause the browsers to transform non-executable content into executable content.



18. Similarly, you can view the the **RECOMMENDATION** section and click on the reference link under **REFERENCES** section.

19. You can also click on any other vulnerability to view its detailed information.

20. This concludes the demonstration of discovering vulnerabilities in a target website scanning using SmartScanner.

21. You can also use other web application vulnerability scanning tools such as **WPScan Vulnerability Database** (https://wpscan.com), **Codename SCNR** (https://ecsypno.com), **AppSpider** (https://www.rapid7.com), **Uniscan** (https://github.com) and **N-Stalker** (https://www.nstalker.com).

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

Question 14.1.3.1

On the windows 11 machine use SmartScanner tool to perform vulnerability scan on www.moviescope.com and analyse the report. Enter the CWE ID that is connected to No redirects from HTTP to HTTPS vulnerability that is found on the target website while scanning.

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 Remote Code Execution (RCE) attack

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.

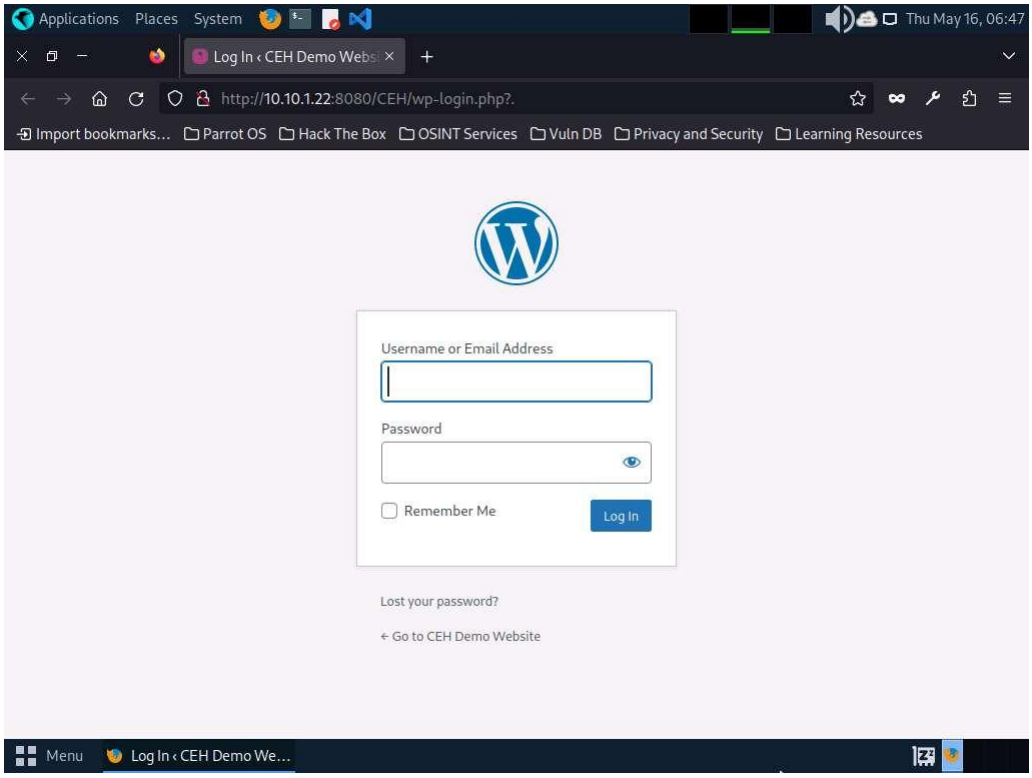
Ensure that the **Wampserver** is running in **Windows Server 2022** machine. To run the **WampServer**, execute the following steps:

- Click Windows Server 2022 to switch to the **Windows Server 2022** machine Click Ctrl+Alt+Delete to activate the machine and login with **CEH\Administrator** / **Pa$$w0rd**.
- Now, click **Type here to search** field on the **Desktop**, search for **wampserver64** in the search bar and 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. Launch the **Mozilla Firefox** web browser and go to **http://10.10.1.22:8080/CEH/wp-login.php?**.

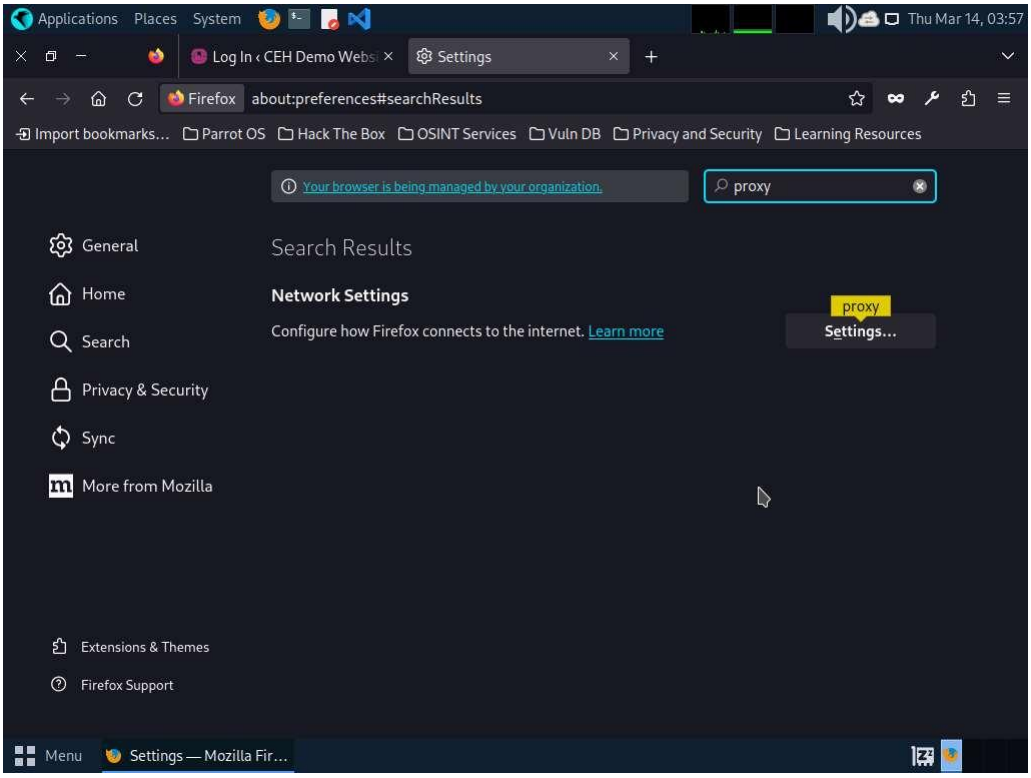
   Here, we will perform a brute-force attack on the designated WordPress website hosted by the **Windows Server 2022** machine.



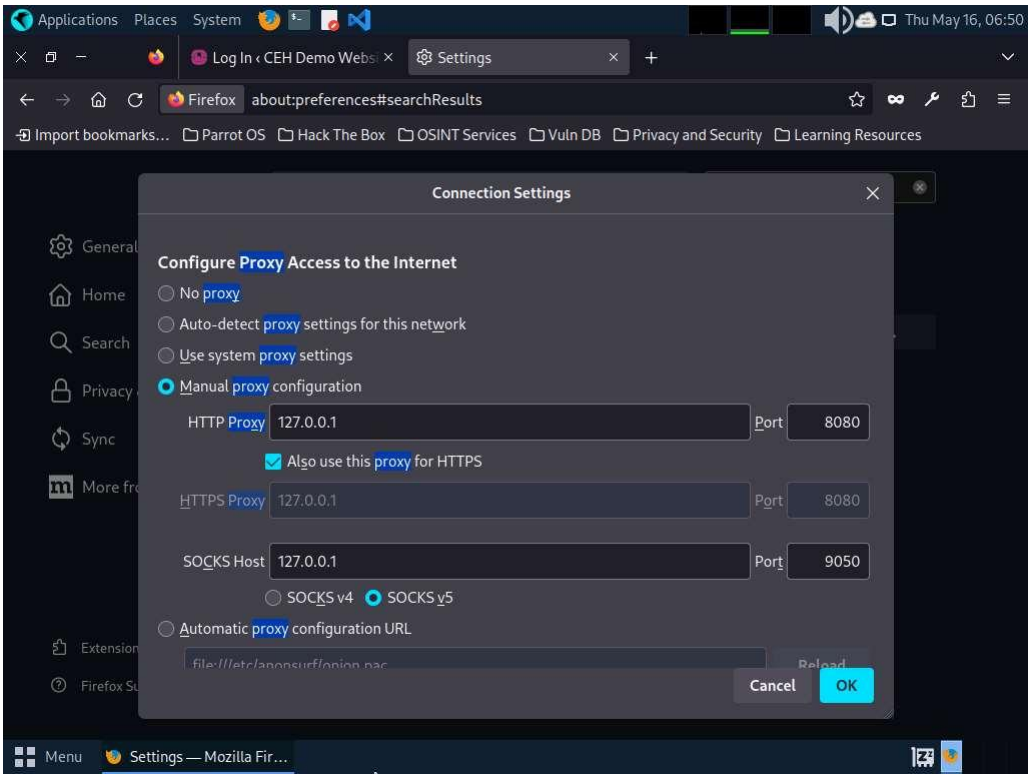
3. Now, we shall set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.

4. In the **Mozilla Firefox** browser, click the **Open application menu** icon ( ) in the right corner of the menu bar and select **Settings** from the drop-down list.

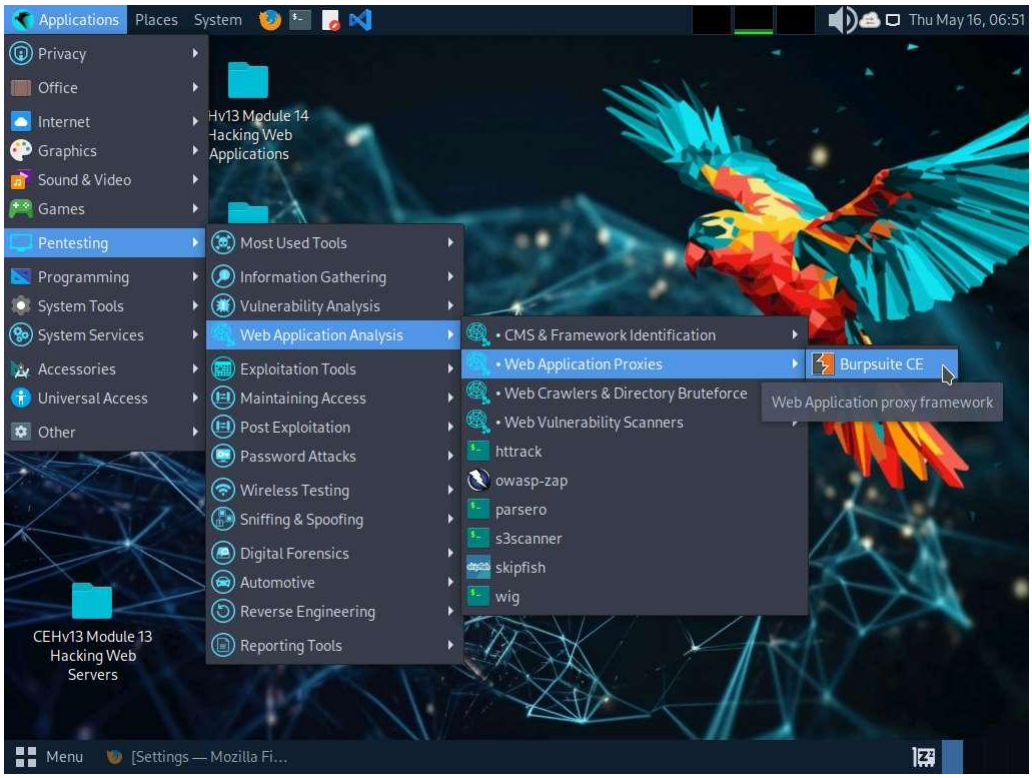
5. The **General** settings tab appears. In the **Find in Settings** search bar, search for **proxy** and in the **Search Results**, click the **Settings** button under the **Network Settings** option.



6. 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 HTTPS** checkbox and click **OK**. Close the **Settings** tab and minimize the browser window.



7. Now, minimize the browser window, click the **Applications** menu form the top left corner of **Desktop**, and navigate to **Pentesting** --> **Web Application Analysis** --> **Web Application Proxies** --> **Burpsuite CE** to launch the **Burpsuite CE** application.



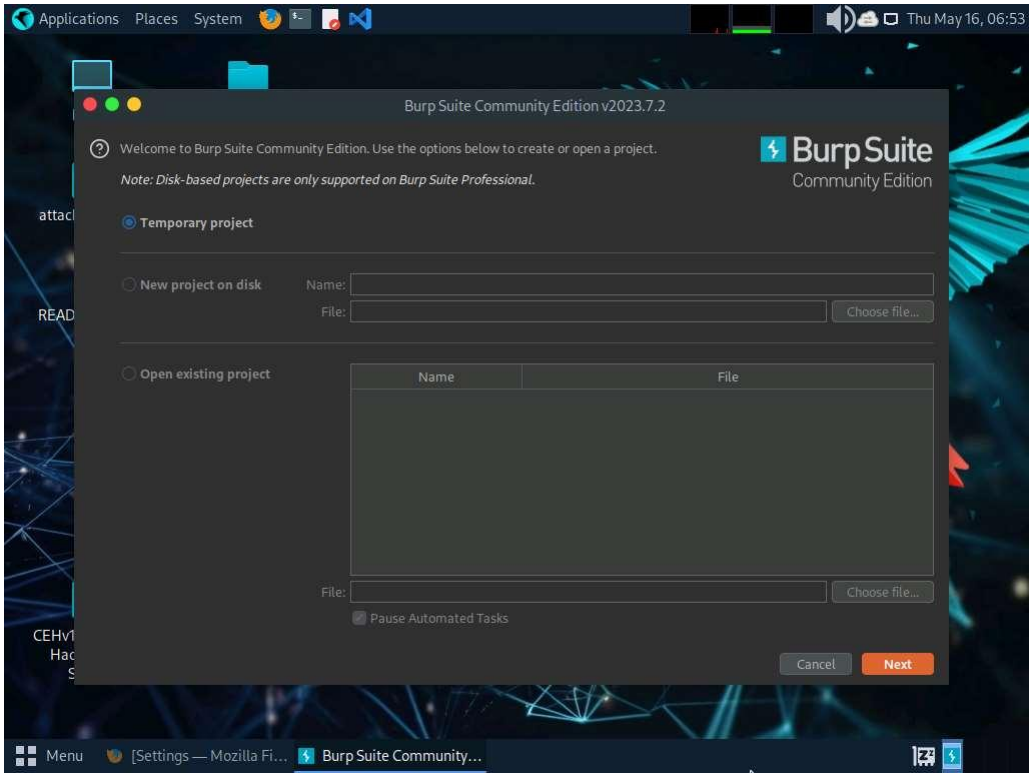
8. The **Burp Suite Community Edition** pop-up appears, click **OK**.

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

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

10. 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.

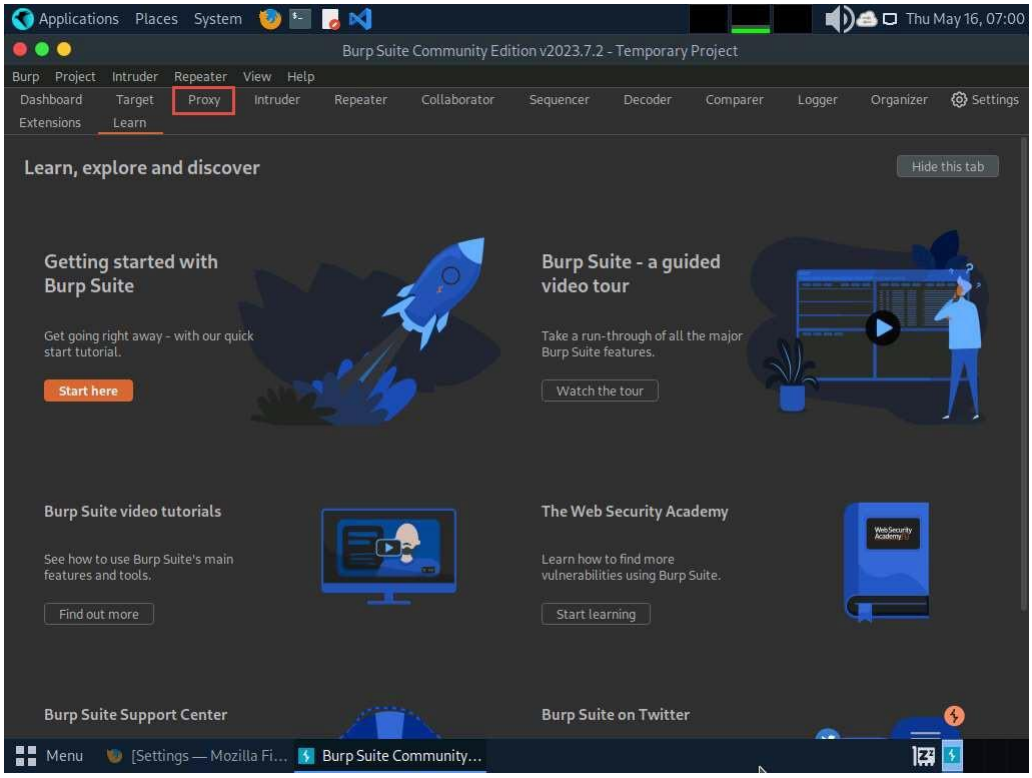
    If an update window appears, click **Close**.



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

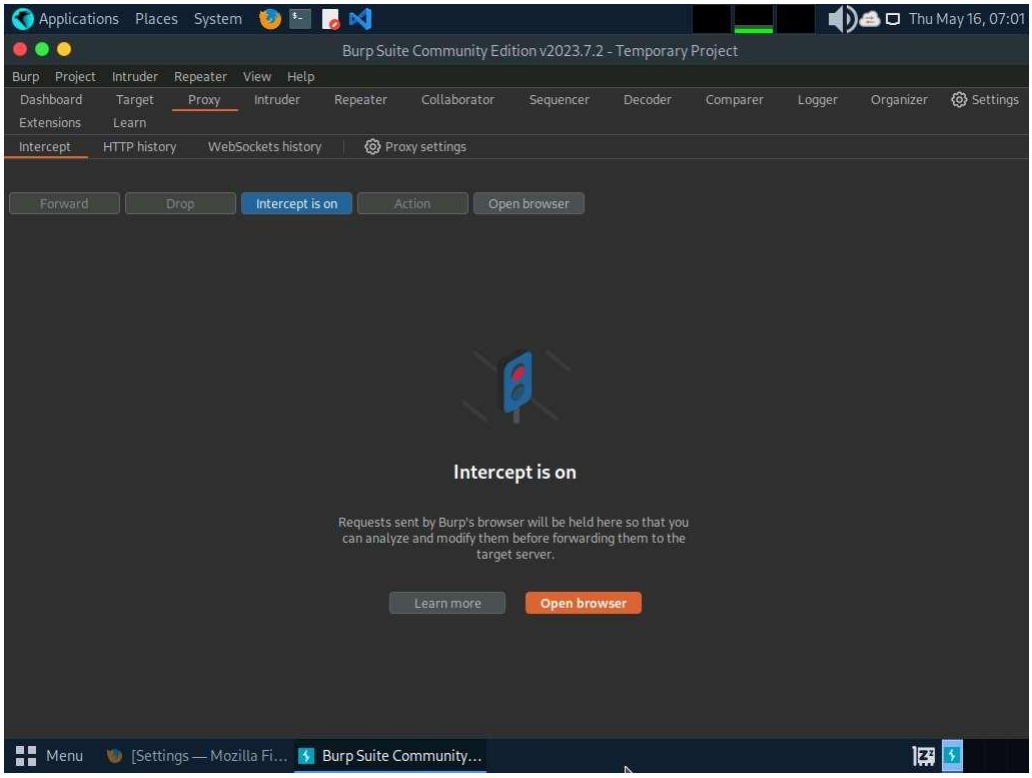
    If **Burp Suite is out of date** pop-up appears check **Don't show again for this version** checkbox and click **OK**.

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



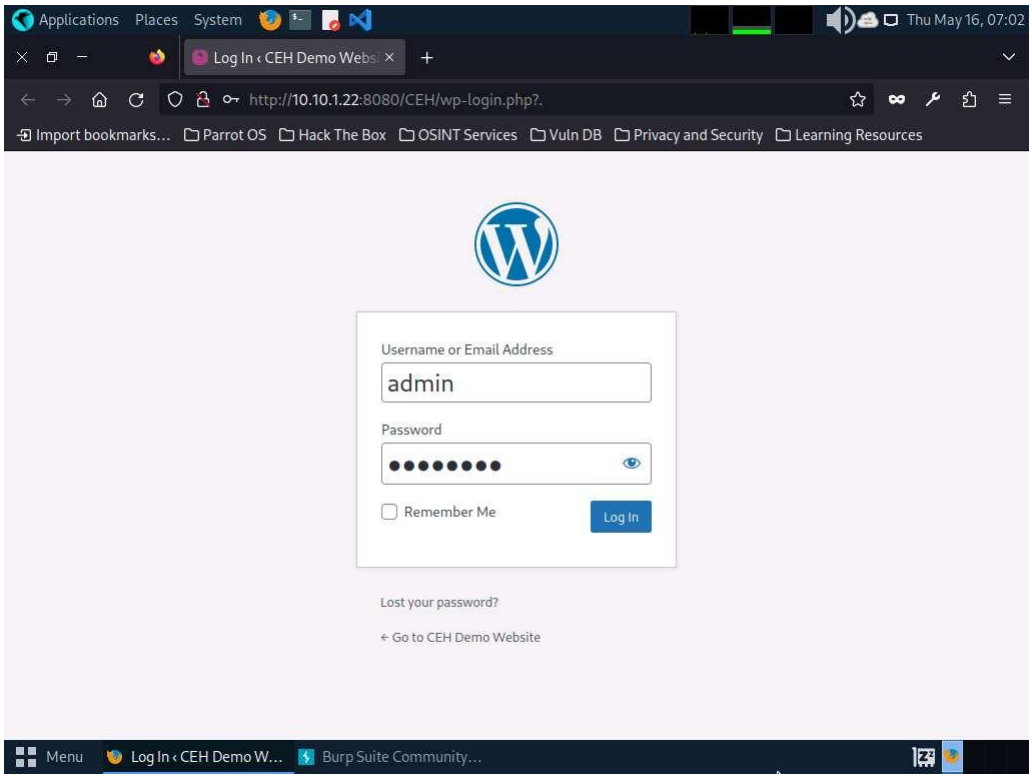
13. In the **Proxy** settings, by default, the **Intercept** tab opens-up. Observe that by default, the interception is active as the button says **Intercept is on**. Leave it running.

    Turn the interception on if it is off.



14. 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.

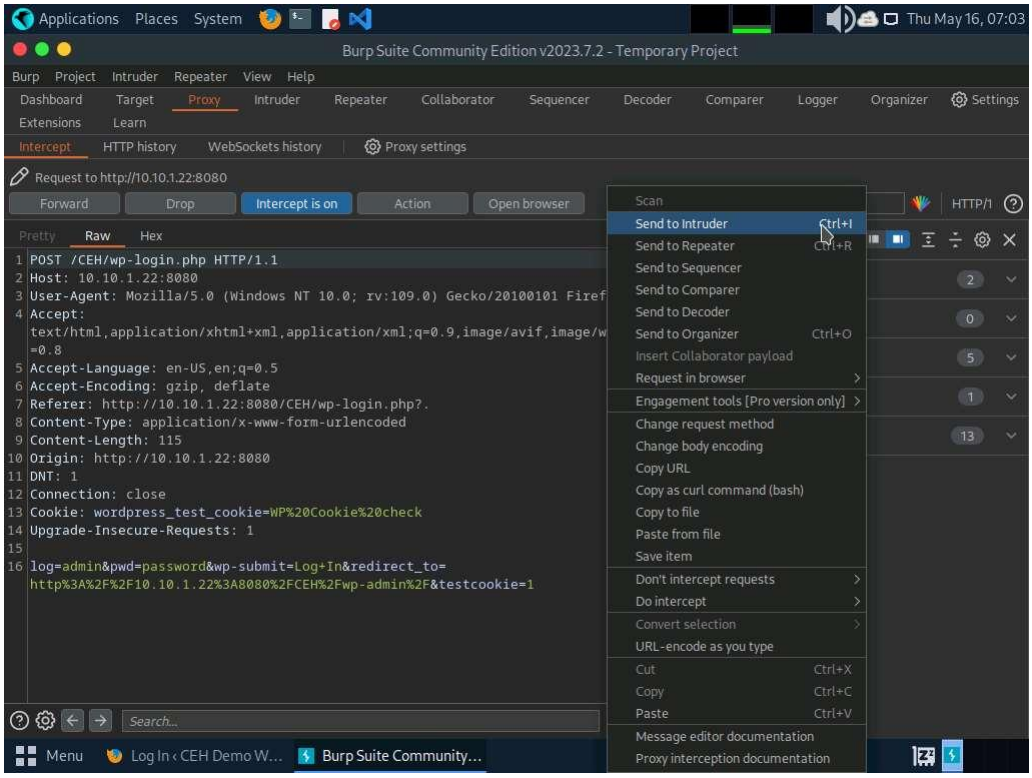


15. Switch back to the **Burp Suite** window; observe that the HTTP request was intercepted by the application.

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

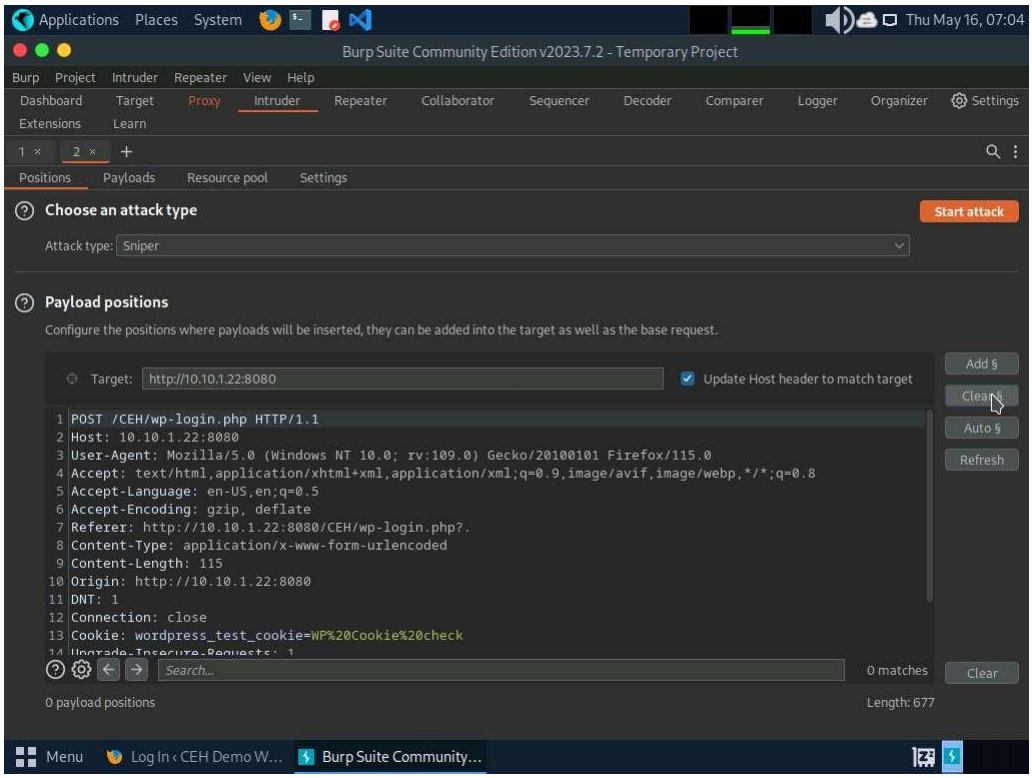
    Observe that Burp Suite intercepted the entered login credentials.

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



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

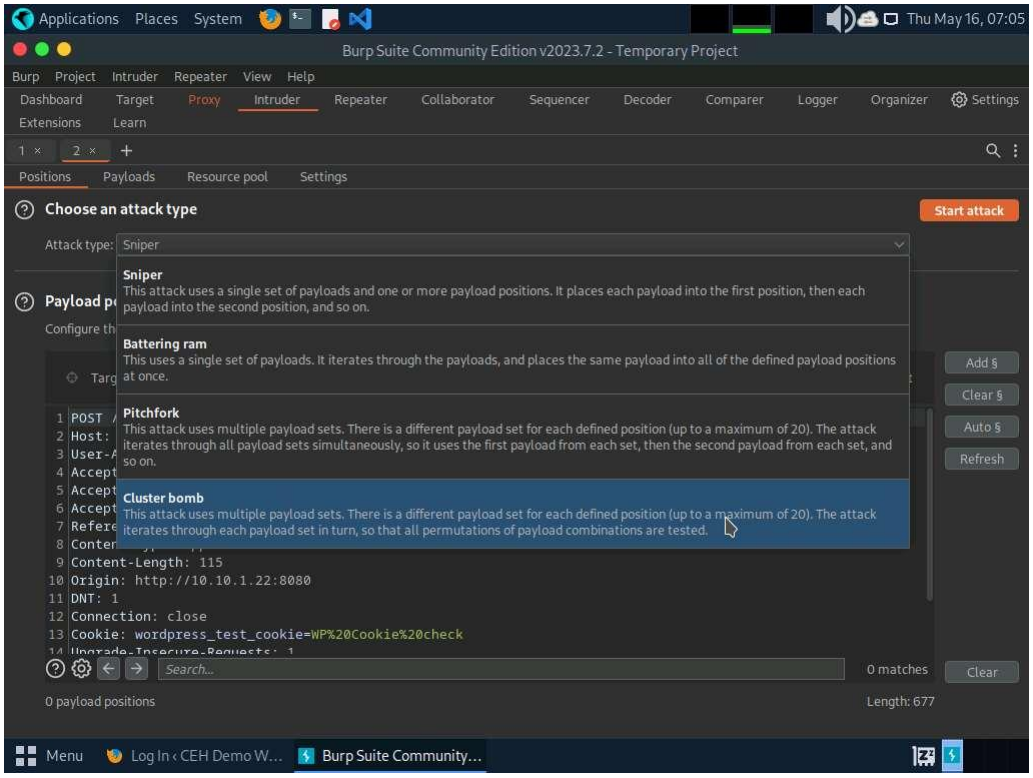
18. In the **Positions** tab under the **Intruder** tab observe that Burp Suite sets the target positions by default, as shown in the HTTP request. Click the **Clear §** button from the right-pane to clear the default payload values.



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

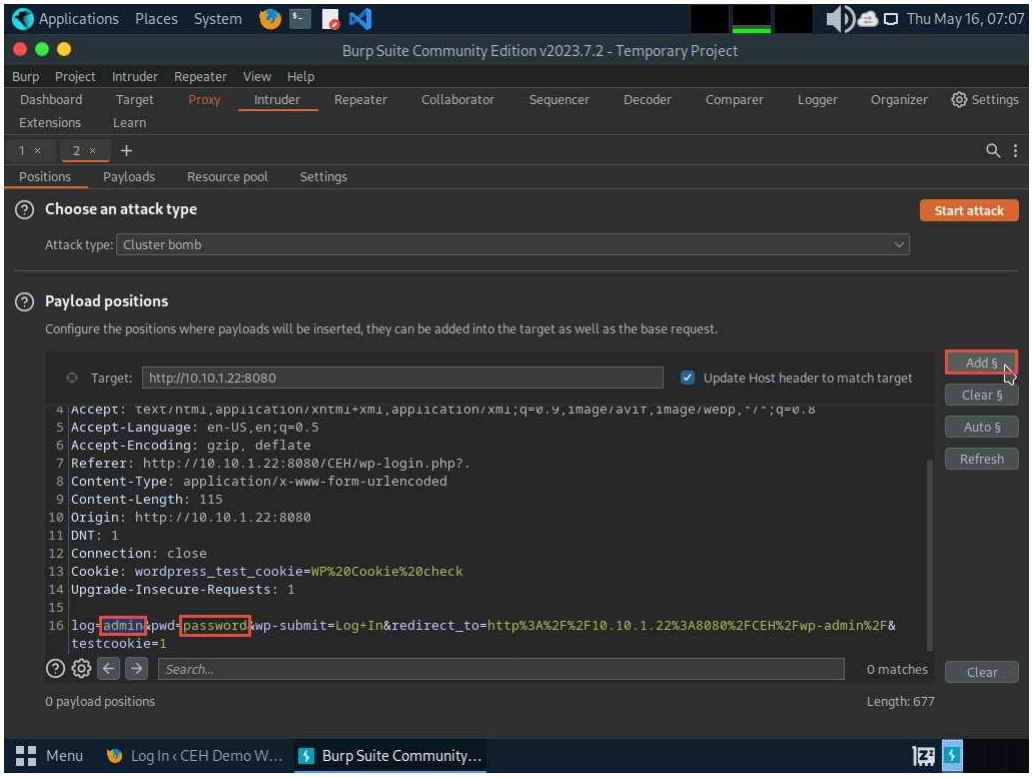
    Cluster bomb uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn so that all permutations of payload combinations are tested. For example, if there are two payload positions, the attack will 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...

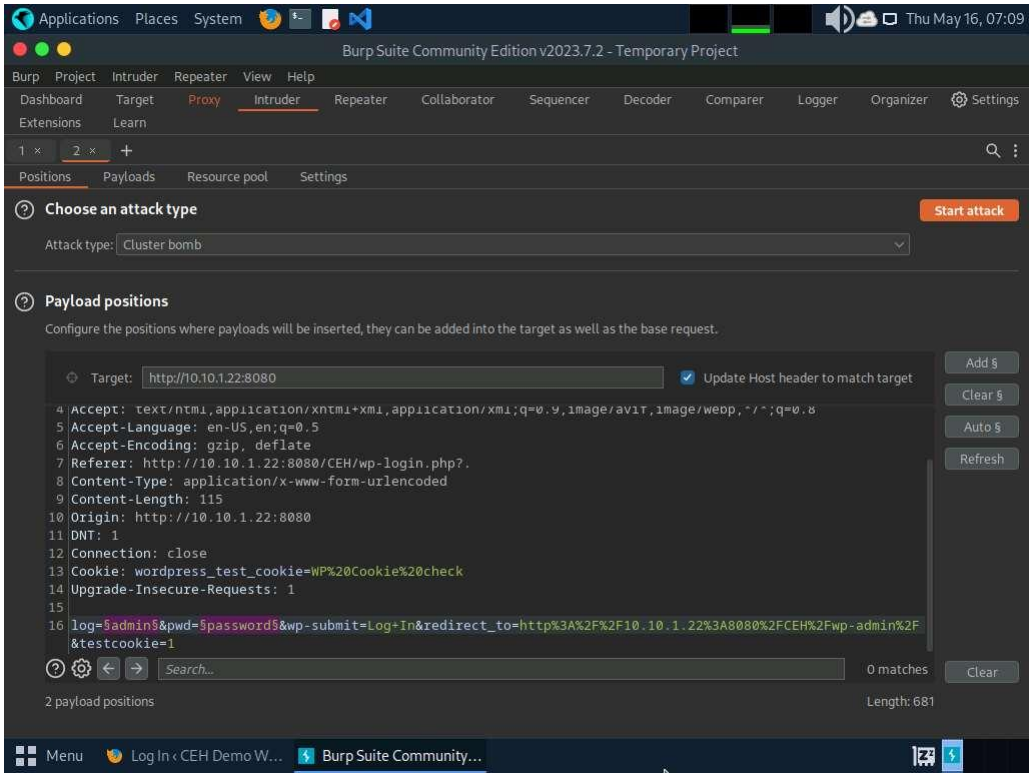


20. Now, we will set the username and password as the payload values. To do so, select the username value entered in **Step#14** and click **Add §** from the right-pane. Similarly, select the password value entered in **Step#14** and click **Add §** from the right-pane.

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



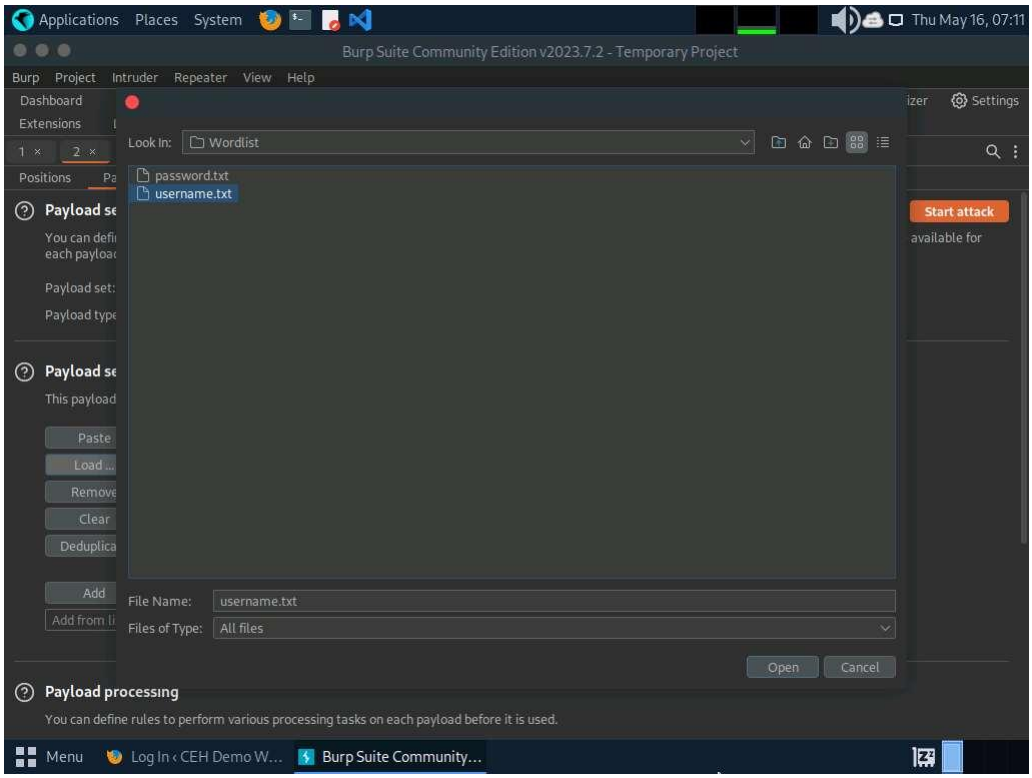
21. Once the username and password payloads are added. The symbol **'§'** will be added at the start and end of the selected payload values. Here, as the screenshot shows, the values are **admin** and **password**.



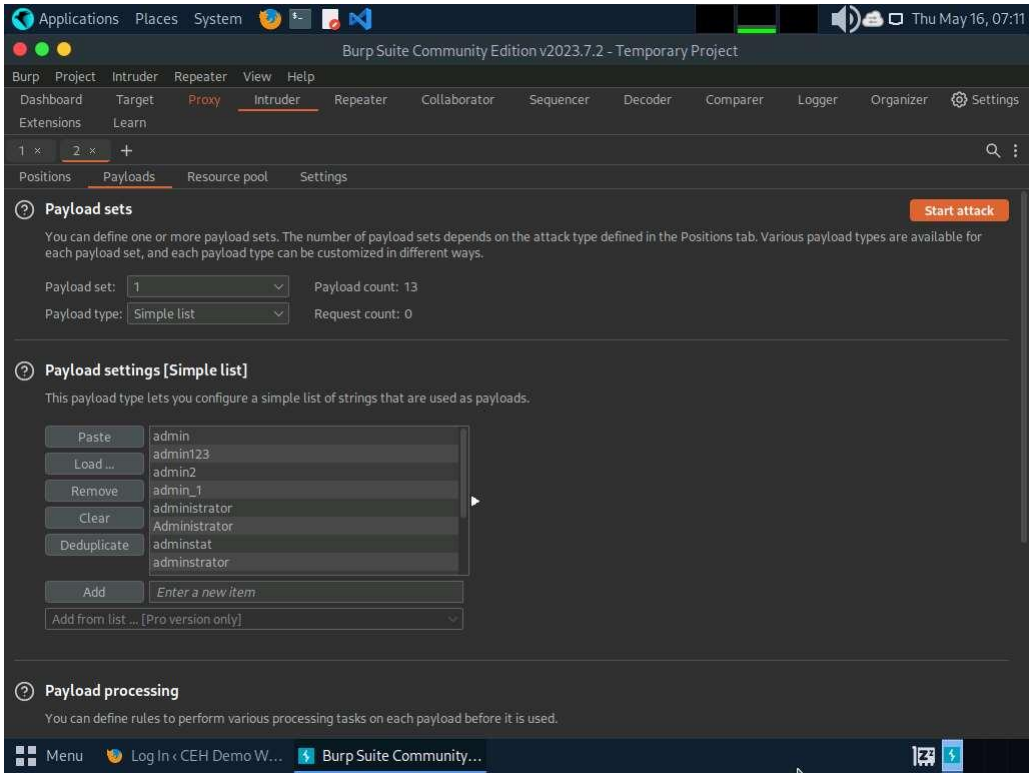
22. Navigate to the **Payloads** tab under the **Intruder** tab and ensure that under the **Payload Sets** section, the **Payload set** is selected as **1**, and the **Payload type** is selected as **Simple list**.

23. Under the **Payload settings [Simple list]** section, click the **Load…** button.

24. A file selection window appears; navigate to the location **/home/attacker/Desktop/CEHv13 Module 14 Hacking Web Applications/Wordlist**, select the **username.txt** file, and click the **Open** button.



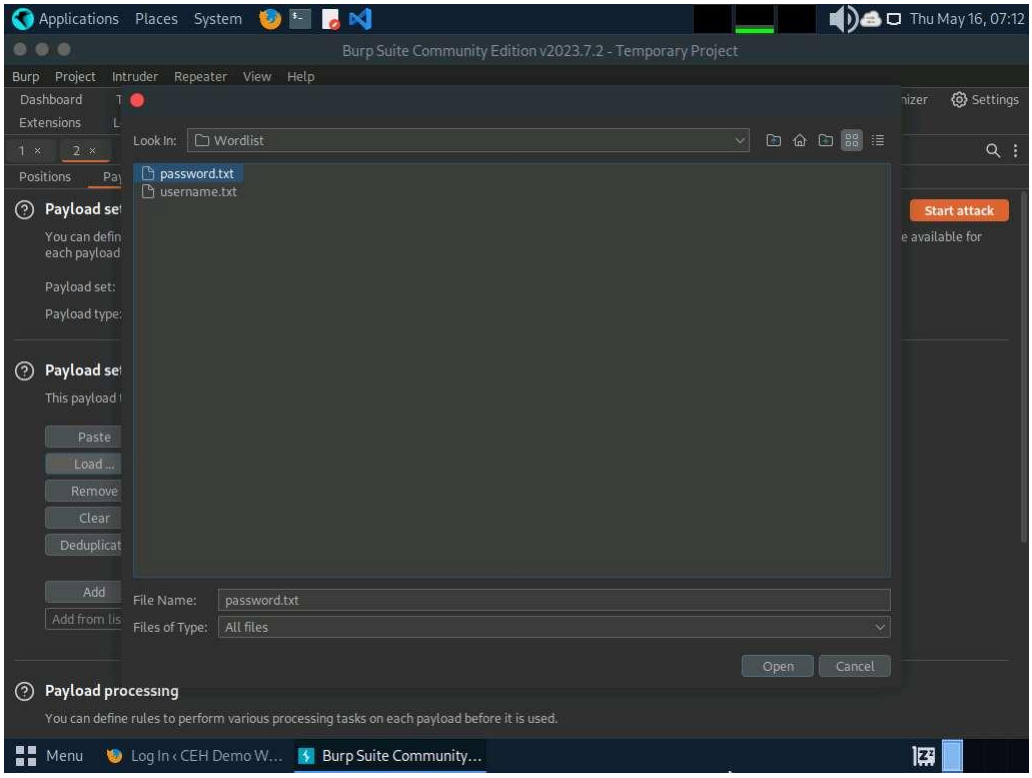
25. Observe that the selected **username.txt** file content appears under the **Payload settings [Simple list]** section, as shown in the screenshot.



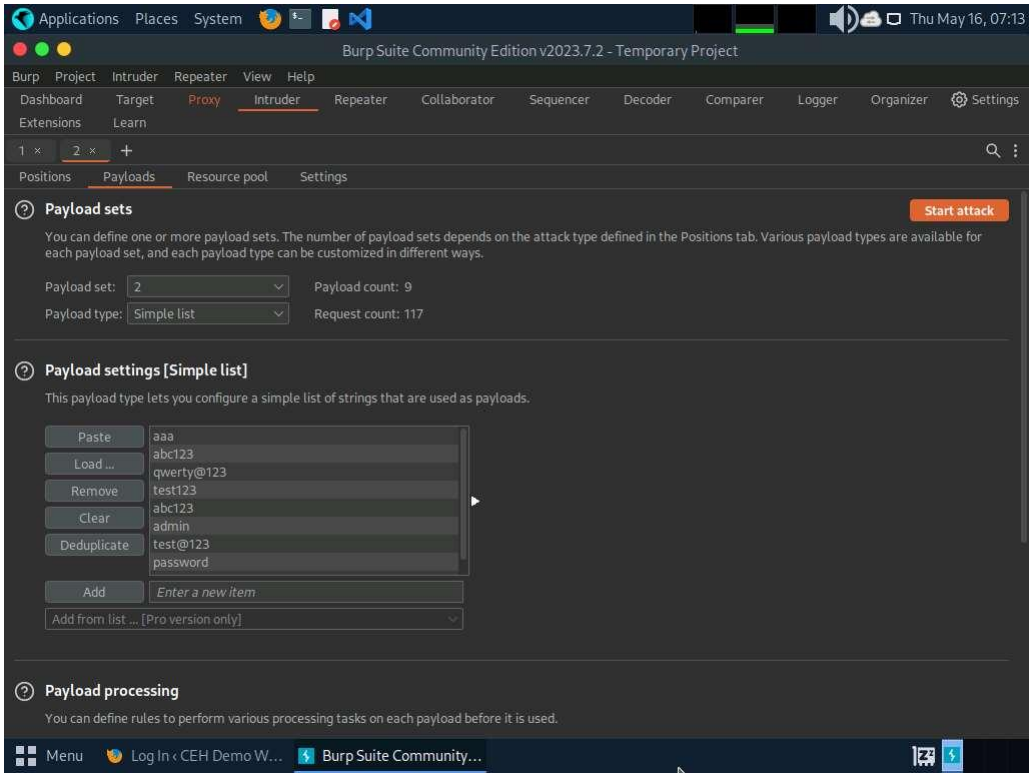
26. Similarly, load a password file for the payload set 2. To do so, under the Payload Sets section, select the **Payload set** as **2** from the drop-down options and ensure that the **Payload type** is selected as **Simple list**.

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

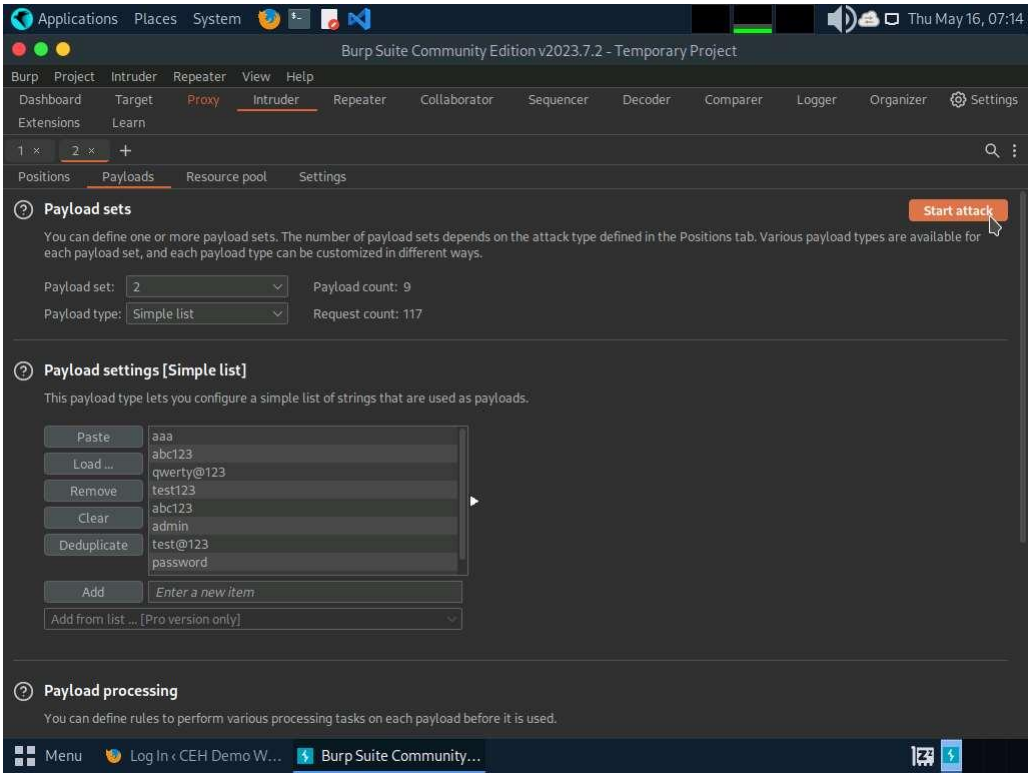
28. A file selection window appears; navigate to the location **/home/attacker/Desktop/CEHv13 Module 14 Hacking Web Applications/Wordlist**, select the **password.txt** file, and click the **Open** button.



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



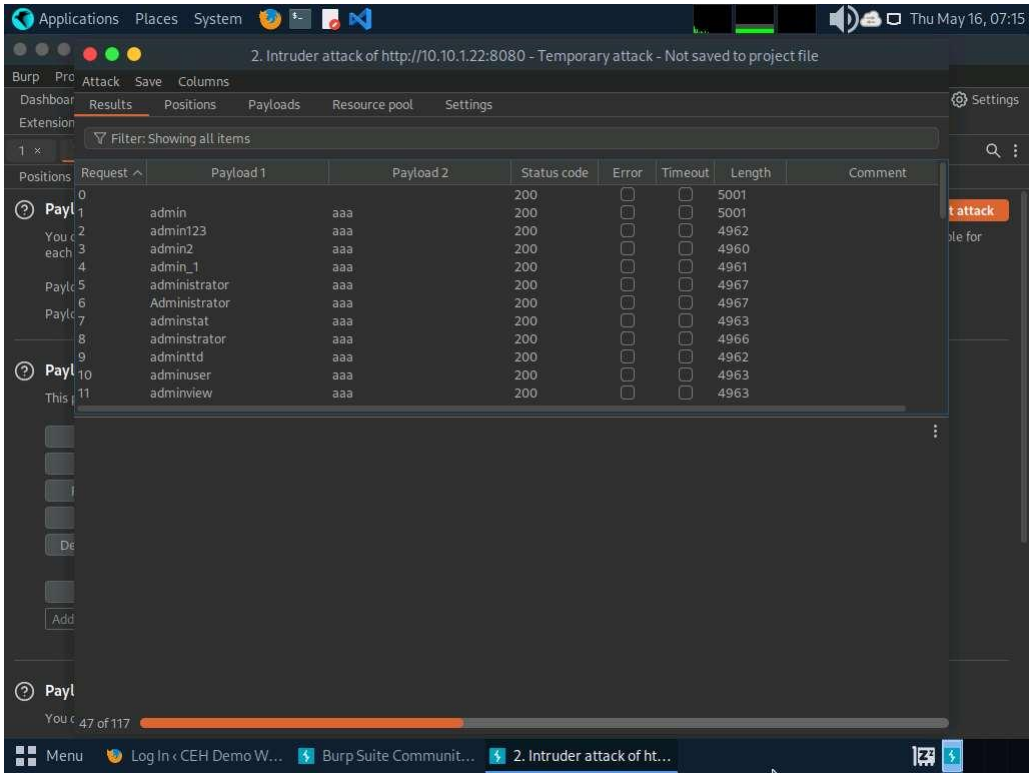
30. Once the wordlist files are selected as payload values, click the **Start attack** button to launch the attack.



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

32. 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**.

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

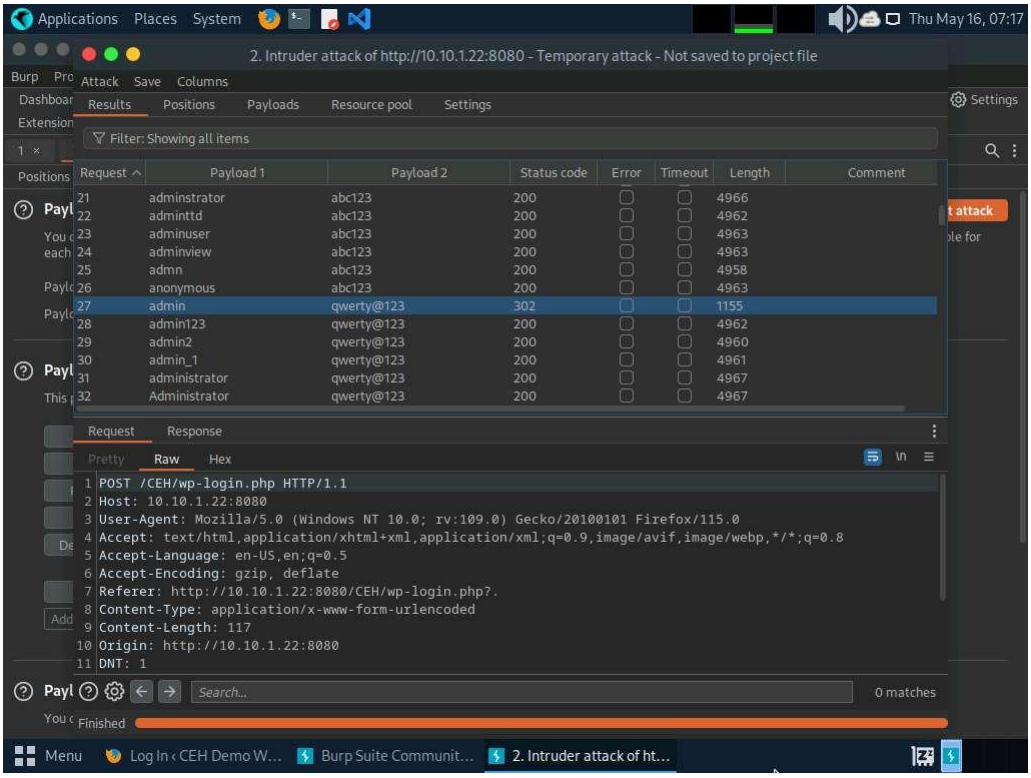


34. After the progress bar completes, scroll down and observe the different values of **Status** and **Length**. Here, Status=**302** and Length= **1155**.

    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.

35. 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.

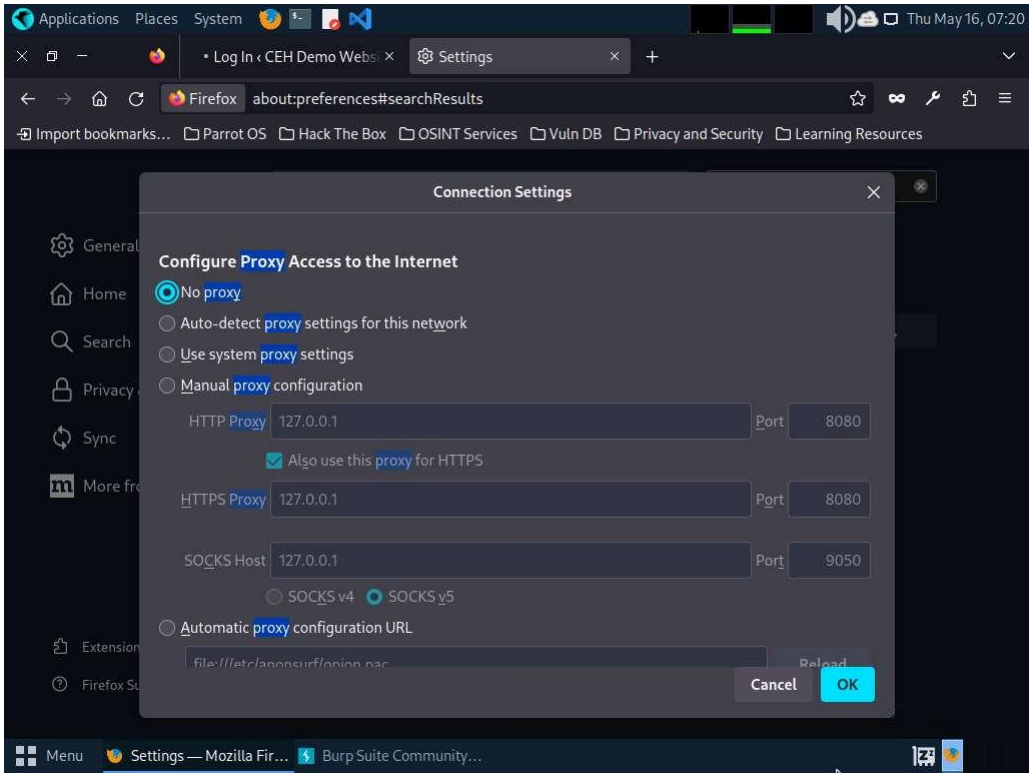


36. Now, that you have obtained the correct user credentials, close the **Intruder attack of 10.10.1.22** window.

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

37. 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.

38. Switch to the browser window and perform **Step#4-5**. Remove the browser proxy set up in **Step#6**, by selecting the **No proxy** radio-button in the **Connection Settings** window and click **OK**. Close the tab.

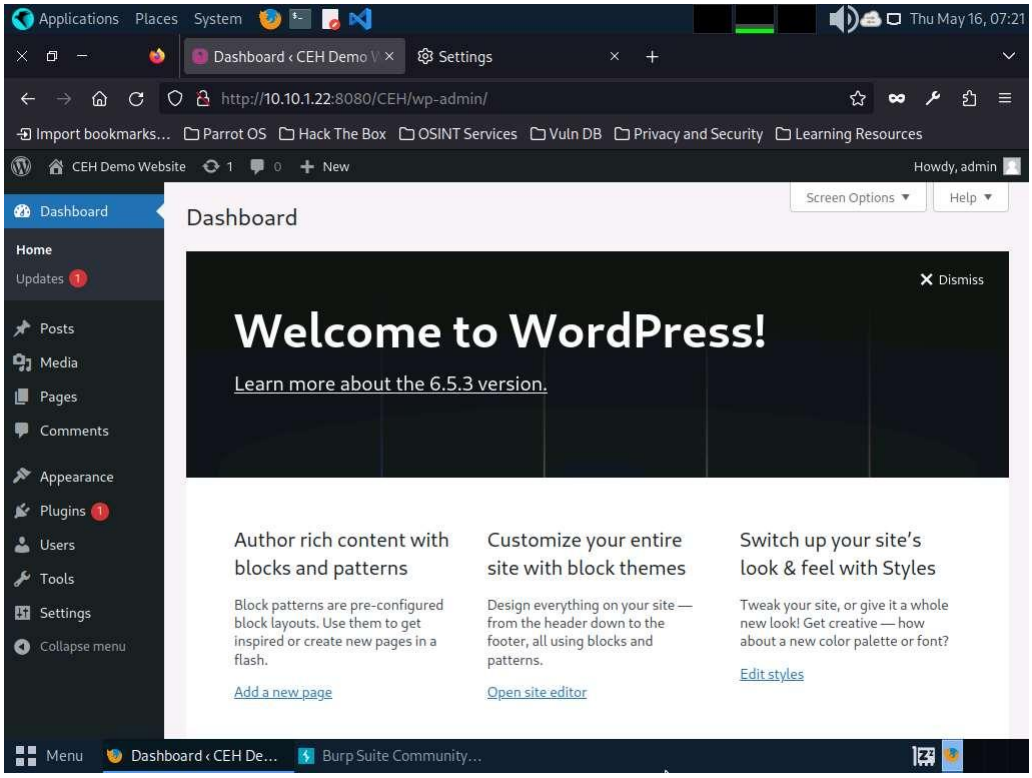


39. Reload the target website **http://10.10.1.22:8080/CEH/wp-login.php?**, enter the **Username** and **Password** obtained in **Step#35** and click **Log In**.

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

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

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



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

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

Question 14.2.1.1

Perform a brute-force attack on the WordPress website (http://10.10.1.22:8080/CEH) using Burp Suite. Enter the username/password obtained. Note: username and password files are available at /home/attacker/Desktop/CEHv13 Module 14 Hacking Web Applications/Wordlist.

Task 2: Perform Remote Code Execution (RCE) Attack

Remote Code Execution (RCE) Attack vulnerability is a critical security flaw that allows an attacker to execute arbitrary code on a target system remotely, without needing physical access to the system. This type of vulnerability is particularly dangerous because it enables attackers to take control of the target system, potentially gaining unauthorized access, stealing data, or causing damage to the system or network.

Attackers exploit these vulnerabilities by injecting malicious code into the target system through various means such as input fields, file uploads, or network protocols. Once the malicious code is executed, the attacker can gain control over the system and perform actions as if they were an authenticated user or system administrator.

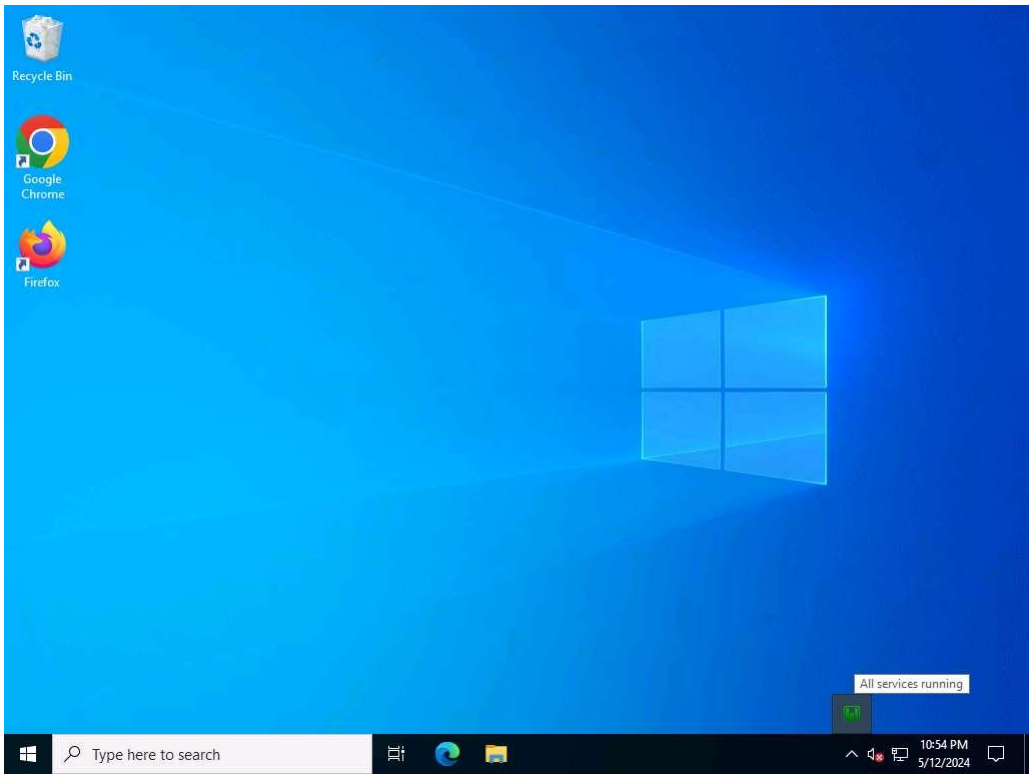
Here, we will perform a CSRF attack using vulnerability present in the wp-upg plugin.

1. Click Windows Server 2022 to switch to the **Windows Server 2022** machine and login with **CEH\Administrator / Pa$$w0rd**.

2. Click **Type here to search** field on the **Desktop**, search for **wampserver64** in the search bar and select **Wampserver64** from the results.

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

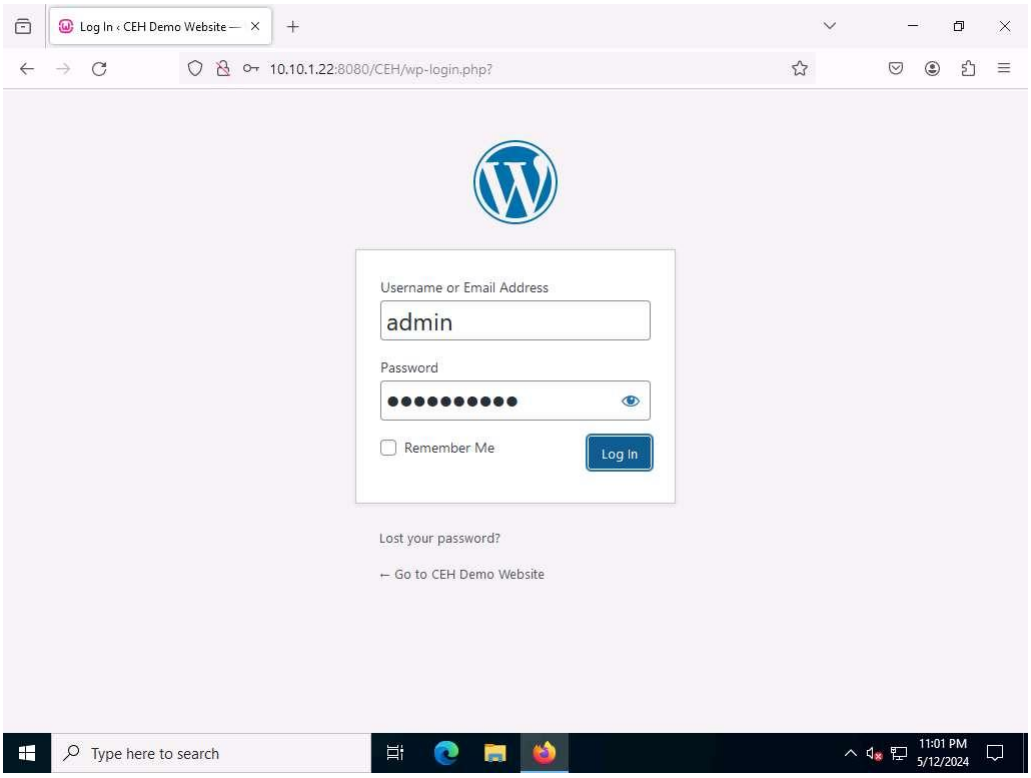
4. Wait for this icon to turn green, which indicates that the **WampServer** is successfully running.



5. Now, open any web browser, and go to **http://10.10.1.22:8080/CEH/wp-login.php?** (here, we are using **Mozilla Firefox**).

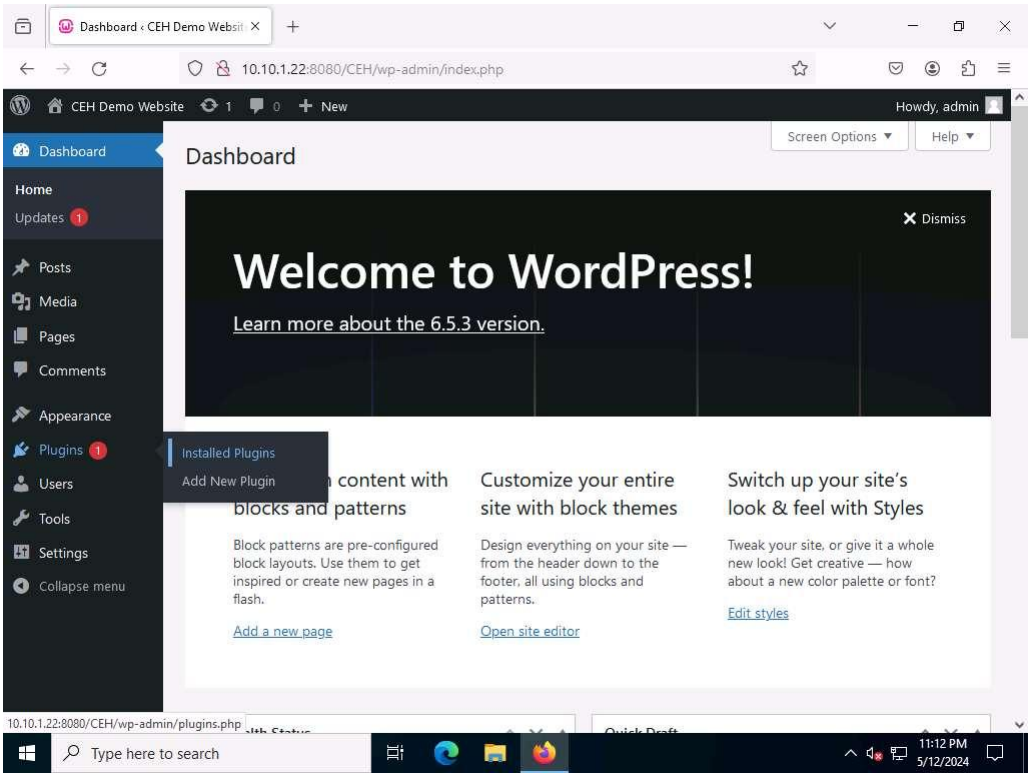
   Here, we are opening the above-mentioned website as the victim.

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

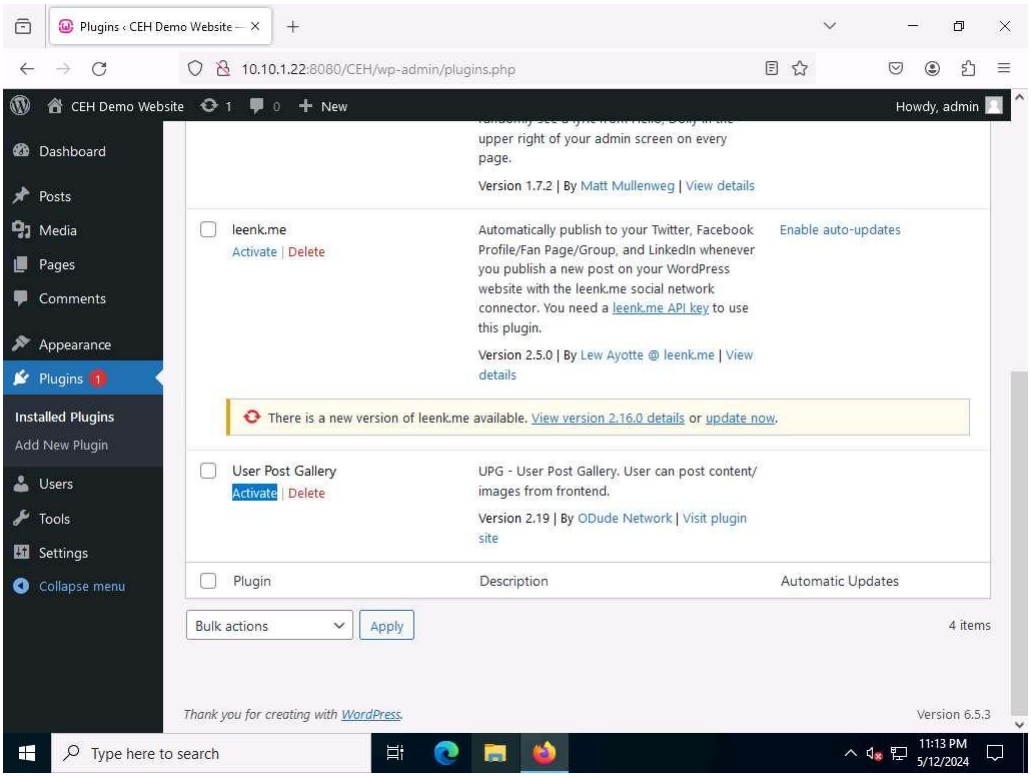


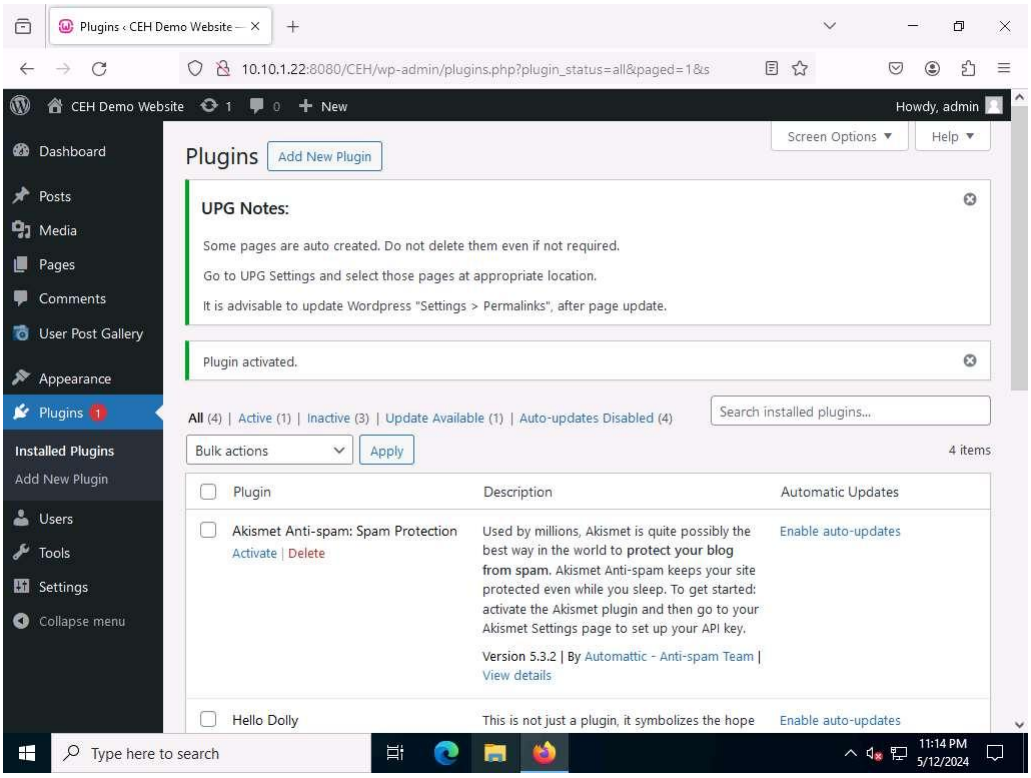
7. Assume that you have installed and configured User Post Gallery plugin

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



9. In the **Plugins** page, observe that **User Post Gallery** is installed.
   Click **Activate** under the **User Post Gallery** plugin to activate the plugin.



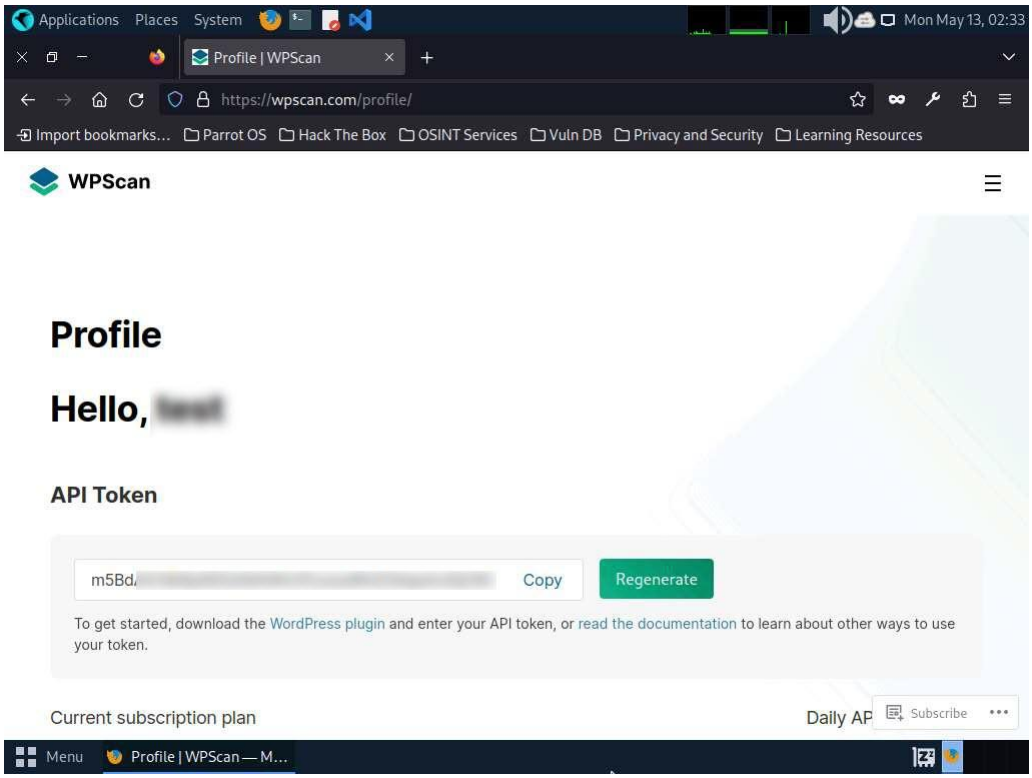


10. Click Parrot Security to switch to the **Parrot Security** machine.

11. Open Mozilla Firefox web browser and go to **https://wpscan.com/** and login to the wpscan account that you have created in previous task.

12. You get signed in successfully in the website. Now, click the **Get Started** button and click **Start for free** button under **Researcher** section.

13. 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.

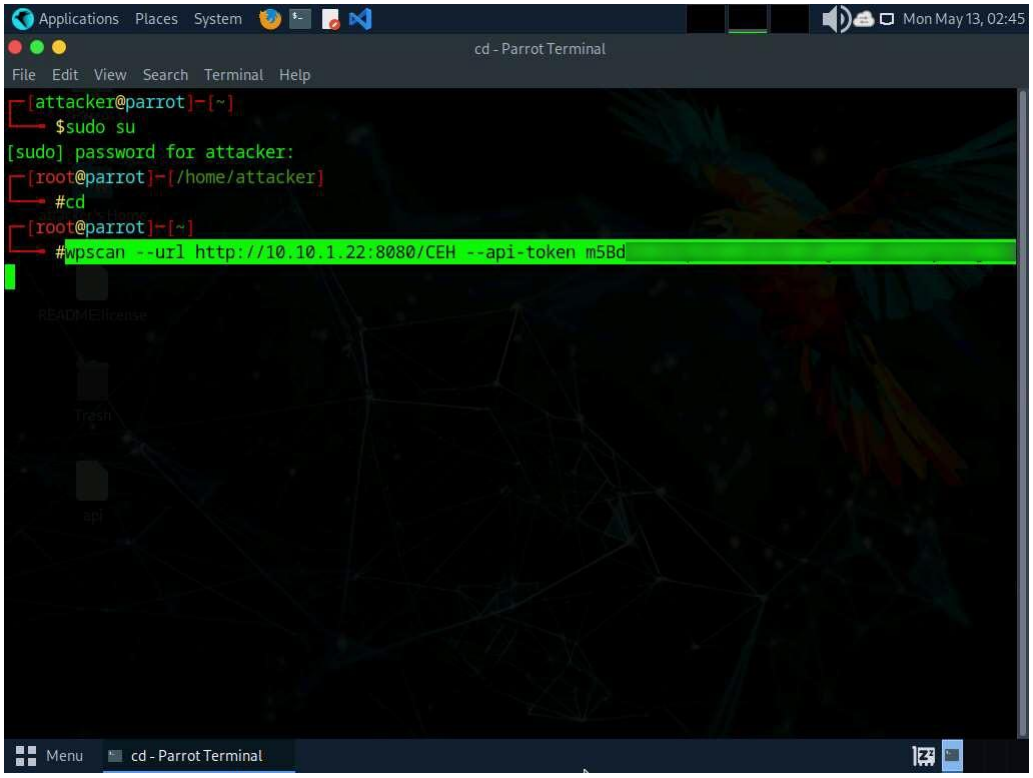


14. Close the **Firefox** browser window.

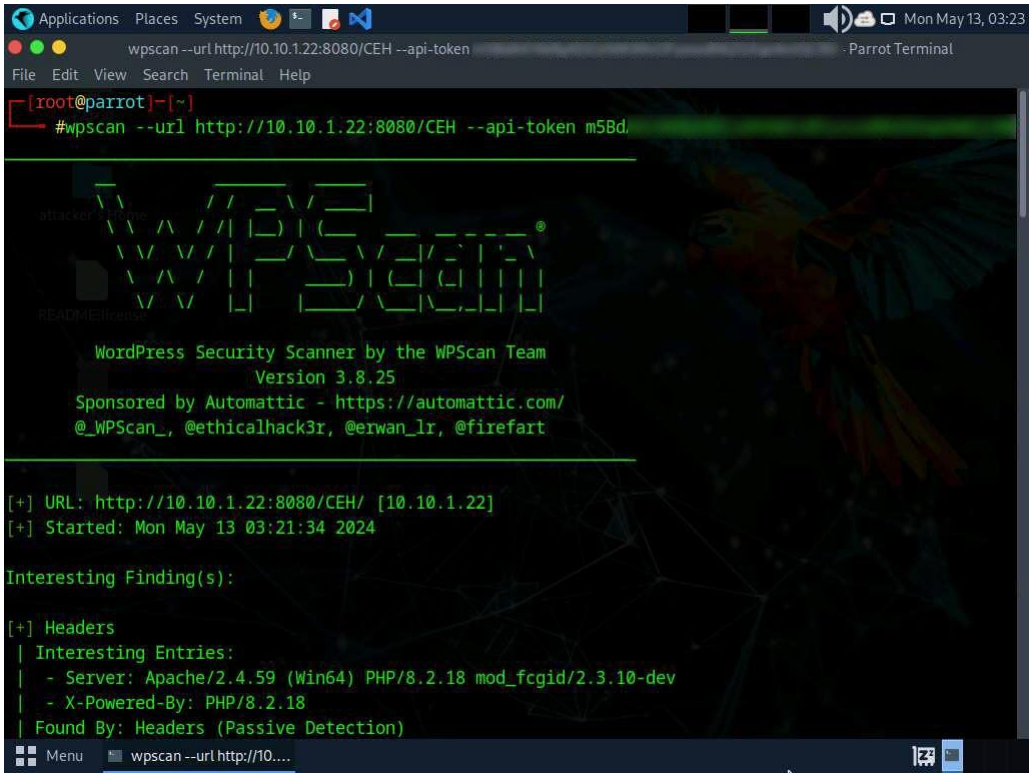
15. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

16. Now, run **cd** command to jump to the root directory.

17. In the Terminal window, run **wpscan --url http://10.10.1.22:8080/CEH --api-token [API Token from Step#13]** command.



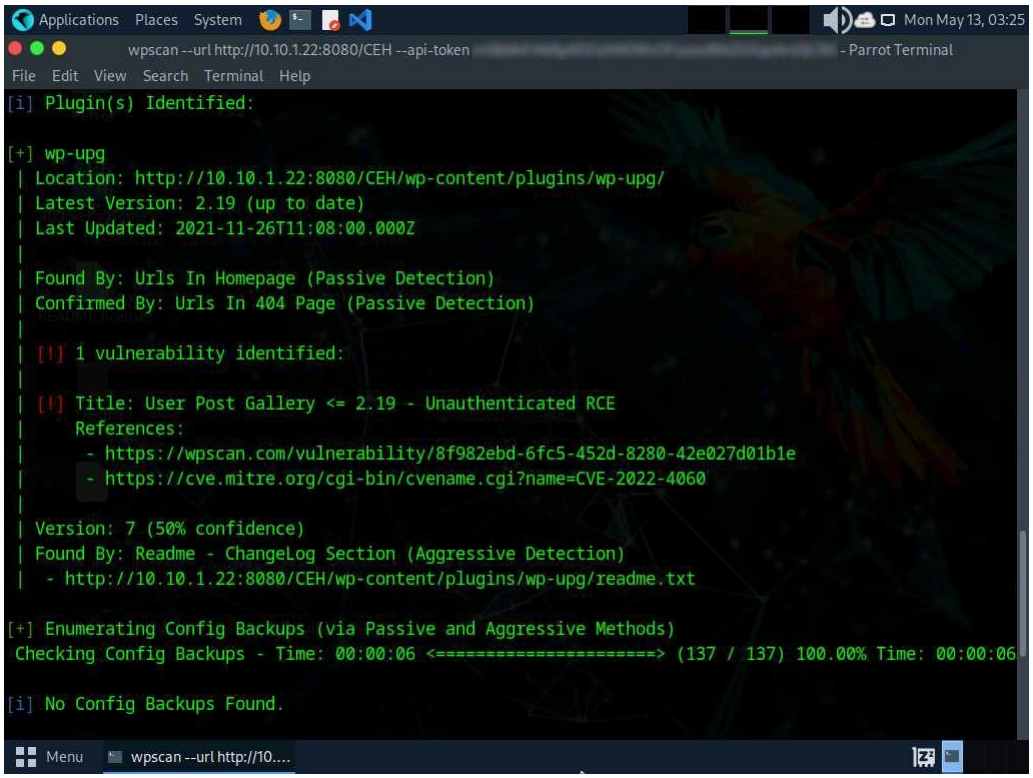
18. The result appears, displaying detailed information regarding the target website.



19. Scroll down to the **Plugin(s) Identified** section, and observe the installed vulnerable plugins (**wp-upg**) on the target website.

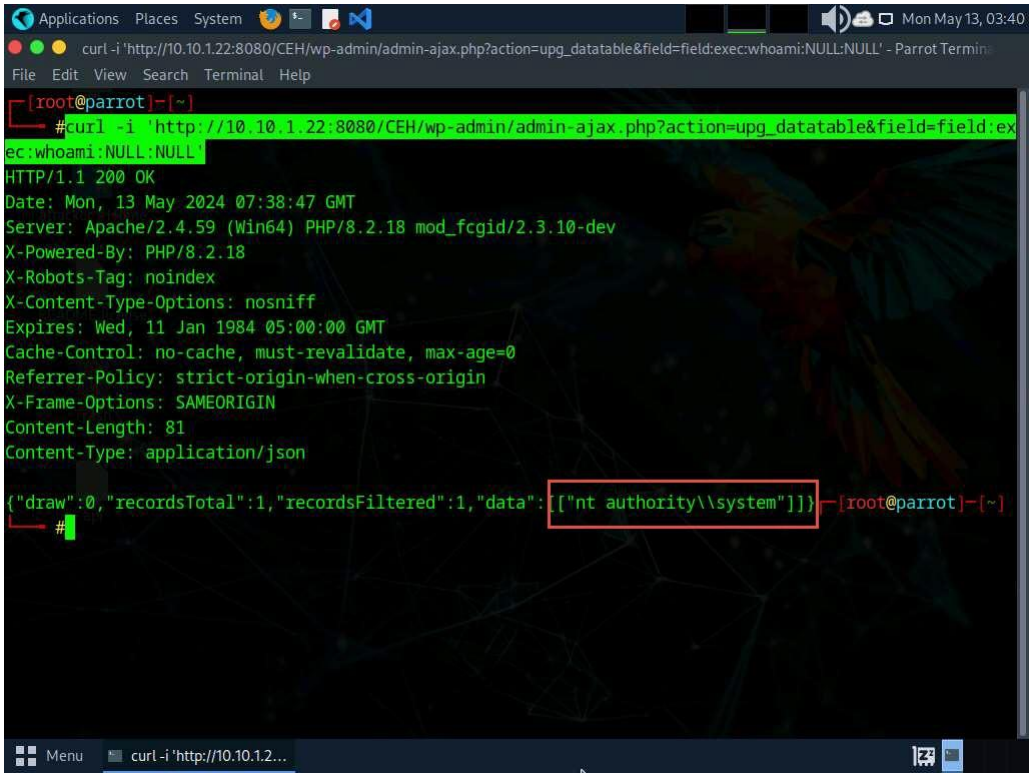
20. In the **Plugin(s) Identified** section, within the context of the **wp-upg** plugin, an **Unauthenticated Remote Code Execution (RCE)** vulnerability has been detected as shown in the screenshot.

    The number of vulnerable plugins might differ when you perform this lab.



21. In this task, we will exploit the **RCE** vulnerability present in the **wp-upg** plugin.

22. To perform RCE attack, run **curl -i 'http://10.10.1.22:8080/CEH/wp-admin/admin-ajax.php?action=upg_datatable&field=field:exec:whoami:NULL:NULL'** com mand.



23. This curl command exploits a WordPress plugin vulnerability by sending a malicious request to the **admin-ajax.php** file, allowing an attacker to execute arbitrary system commands via the **exec** function, potentially leading to **remote code execution**.

24. In the last step, **whoami** command was executed, yielding the outcome **nt authority\ \system**

25. This concludes the demonstration of performing RCE attack.

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

Question 14.2.2.1

In Windows Server 2022 machine activate User Post Gallery plugin which is installed in http://10.10.1.22:8080/CEH web application. From Parrot Security machine, scan for vulnerable plugins on the http://10.10.1.22:8080/CEH web application hosted in Windows Server 2022 machine using WPScan and perform Remote code execution attack on the http://10.10.1.22:8080/CEH website. Enter the plugin name that was identified exploited in the target web application to perform RCE attack.

Lab 3: Detect Web Application Vulnerabilities using Various Web Application Security Tools

Lab Scenario

When talking about web applications, organizations consider security to be a critical component, because web applications are a major source of attacks. Attackers try various application-level attacks to compromise the security of web applications to commit fraud or steal sensitive information.

Web application attacks, launched on port 80/443, go straight through the firewall, past the OS and network-level security, and into the heart of the application, where corporate data resides. Tailor-made web applications are often insufficiently tested, have undiscovered vulnerabilities, and are, therefore, easy prey for hackers.

A professional ethical hacker or pen tester needs to determine whether their organization's website is secure, before hackers download sensitive data, commit crimes using the website as a launchpad, or otherwise endanger the business. There are various web application security assessment tools available to scan, detect, and assess the security and vulnerabilities of web applications. These tools reveal the web application's security posture and are used to find ways to harden security and create robust web applications. These tools automate the process of accurate web-app security assessment, thus enabling cybersecurity staff to protect their business from impending hacker attacks!

The tasks in this lab will assist in discovering the underlying vulnerabilities and flaws in the target web application.

Lab Objectives

- Detect web application vulnerabilities using wapiti web application security scanner

Overview of Web Application Security

Web application security deals with securing websites, web applications, and web services. Web application security includes secure application development, input validation, creating and following security best practices, using WAF Firewall/IDS, and performing regular auditing of a network using web application security tools.

Web Application security tools are automated tools that scan web applications, normally from the outside, to look for security vulnerabilities such as XSS, SQL injection, command injection, path traversal, and insecure server configuration. This category of tools is frequently referred to as Dynamic Application Security Testing (DAST) Tools.

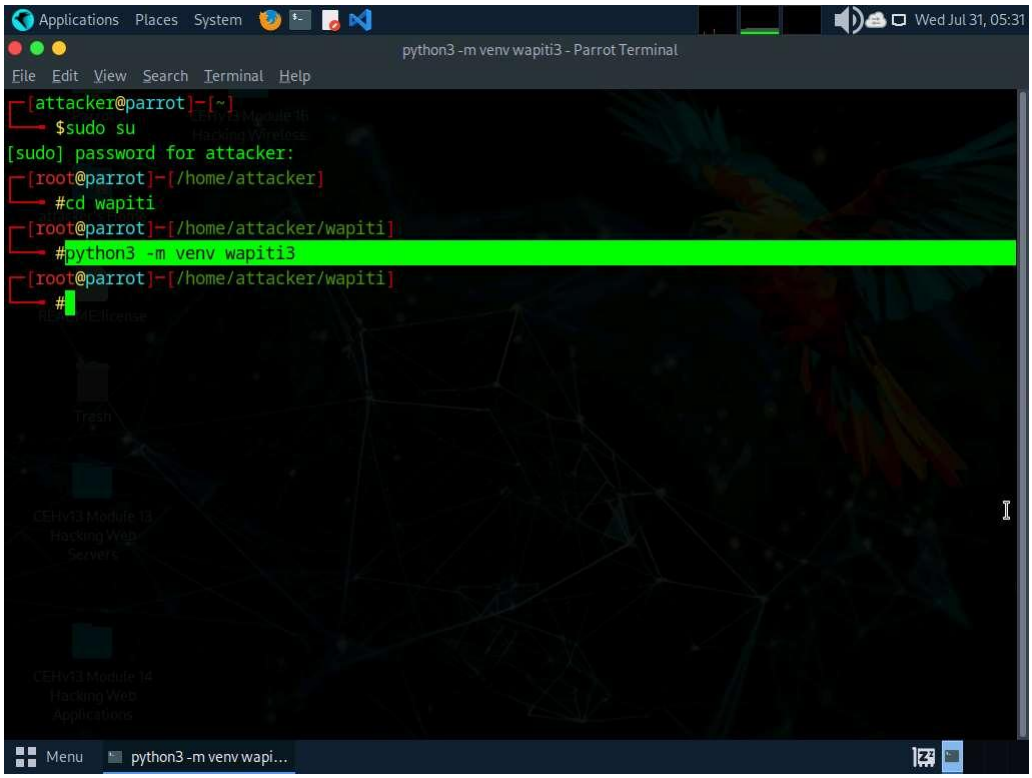
Task 1: Detect Web Application Vulnerabilities using Wapiti Web Application Security Scanner

The Wapiti web-application vulnerability scanner identifies security weaknesses in web applications by crawling websites and performing black-box testing. It detects issues like SQL injections, XSS, and other vulnerabilities.

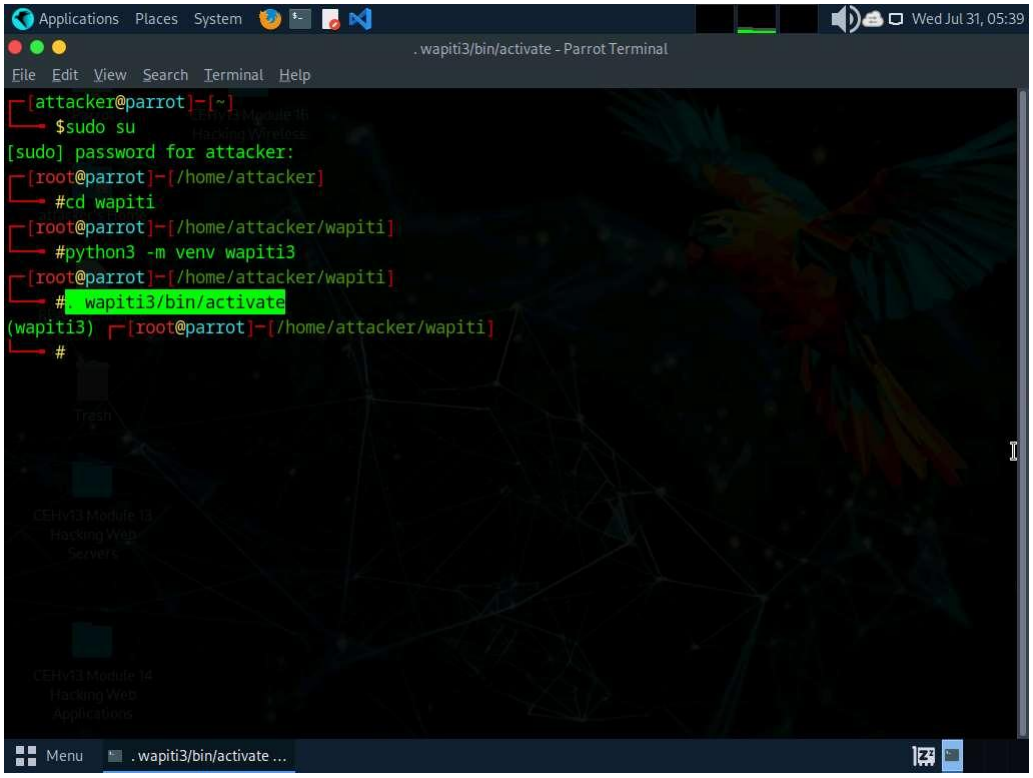
1. Click Parrot Security to switch to the **Parrot Security** machine. Open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).

   The password that you type will not be visible.

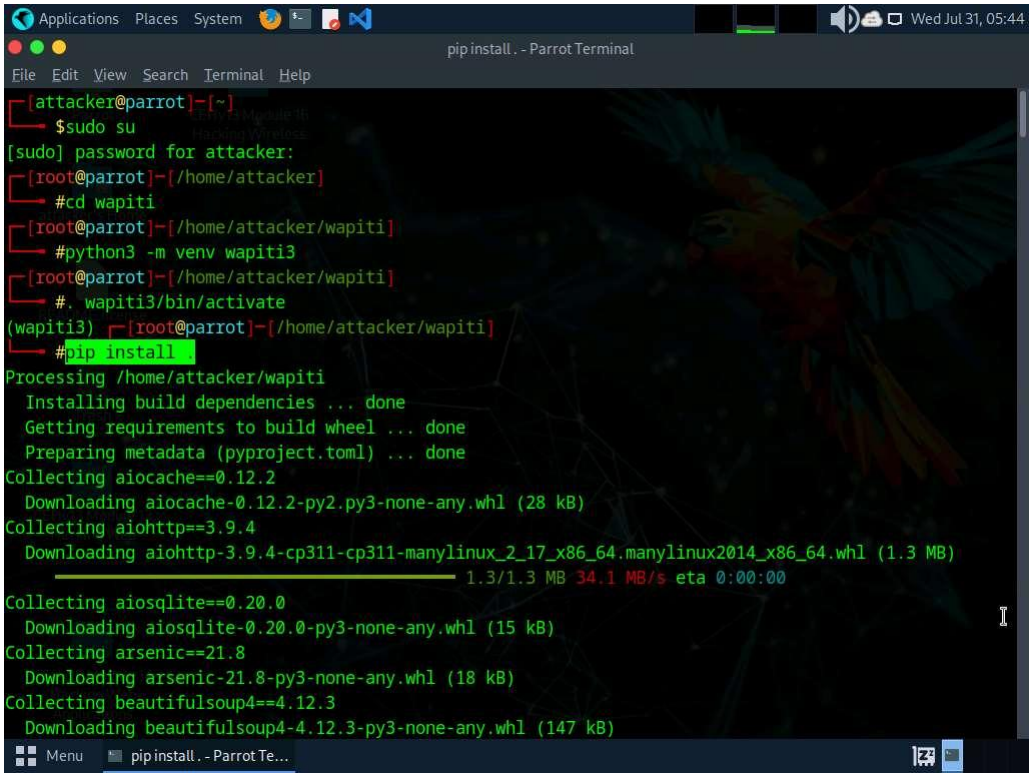
2. In the terminal window run **cd wapiti** command to navigate into wapiti directory and run **python3 -m venv wapiti3** command to create virtual environment in python.



3. Now, run **. wapiti3/bin/activate** command to activate virtual environment.

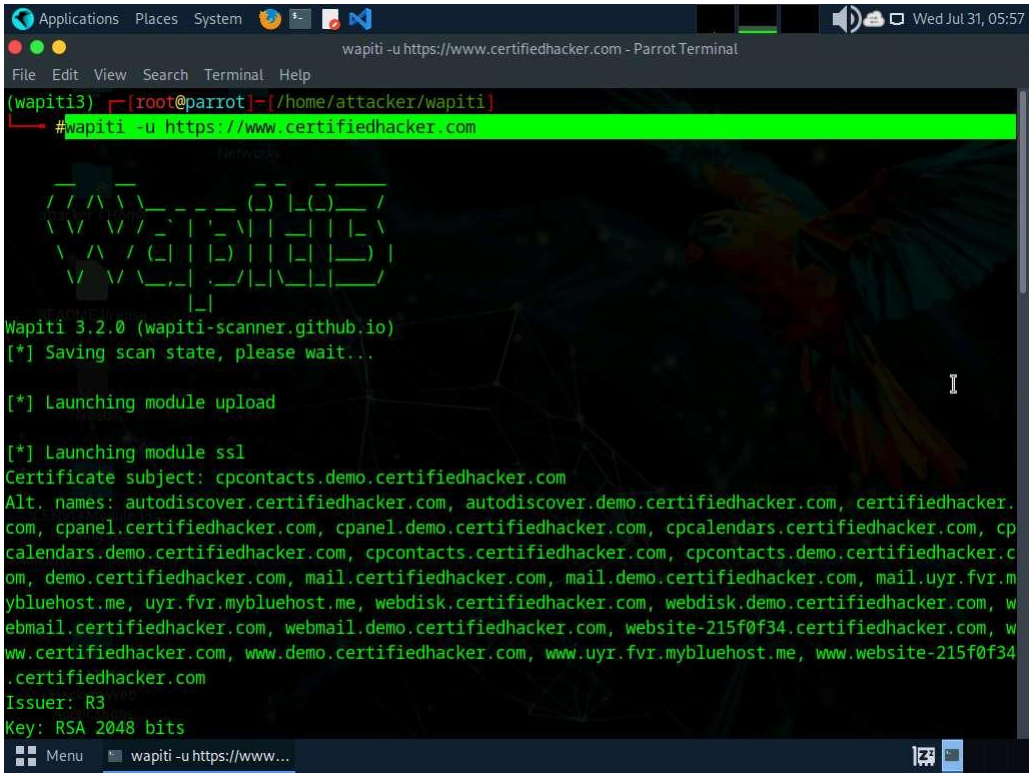


4. Run **pip install .** command to install wapiti web application security scanner.

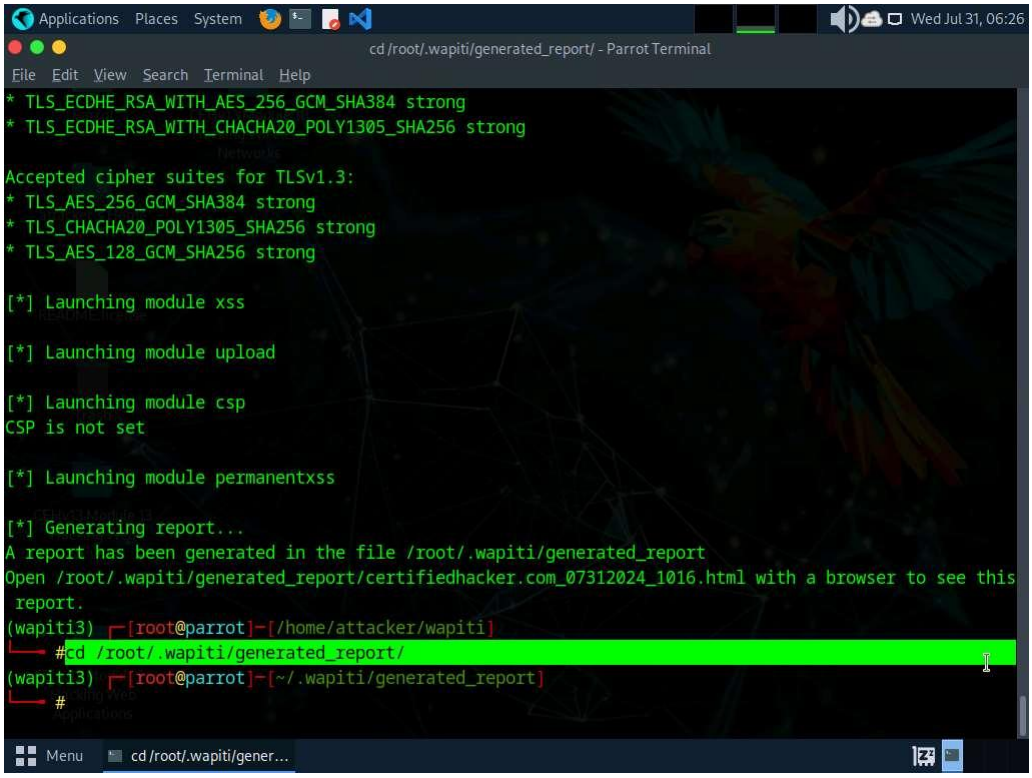


5. After installing the tool run **wapiti -u https://www.certifiedhacker.com** command to perform web application security scanning on certifiedhacker.com website.

It takes approximately 10 minutes for the scan to complete.

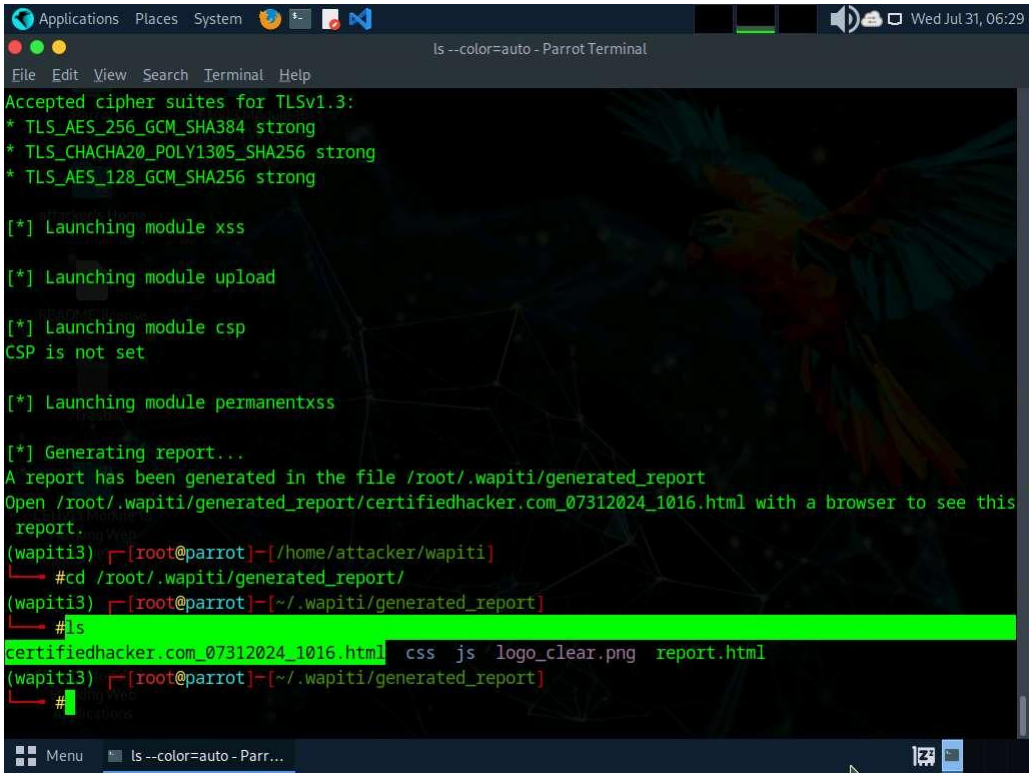


6. Now, in the terminal run **cd /root/.wapiti/generated_report/** to navigate to generated_report directory.

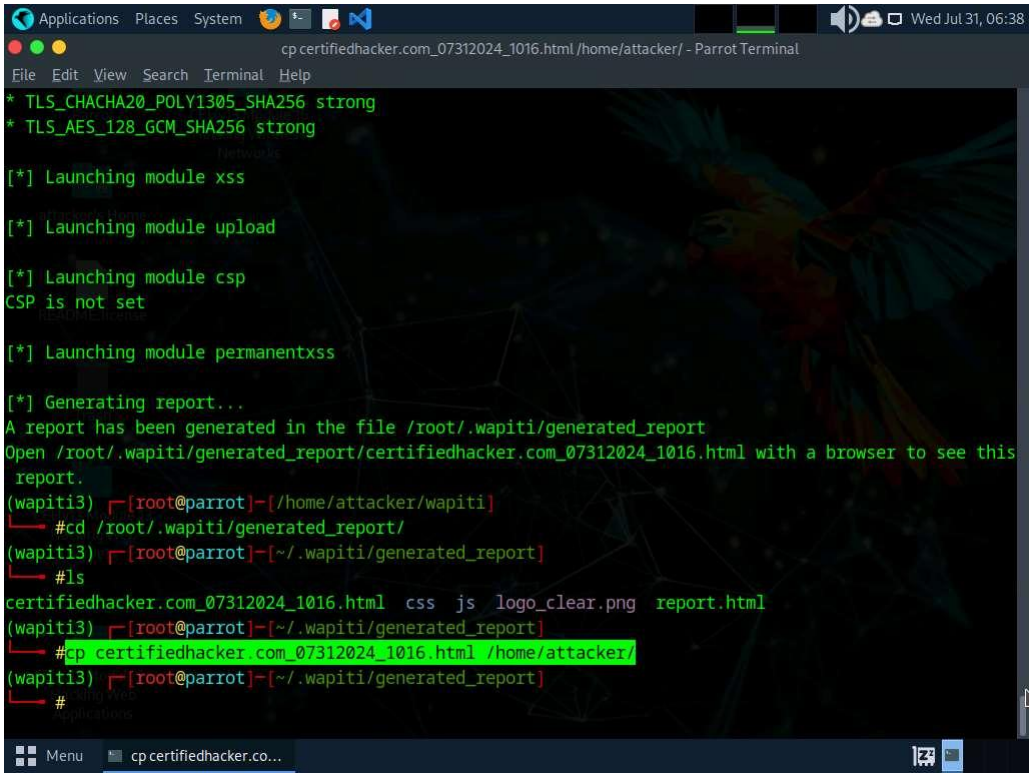


7. Run **ls** command to view the contents of the directory. we can see that the **certifiedhacker.com_xxxxxxxx_xxxx.html** file is created.

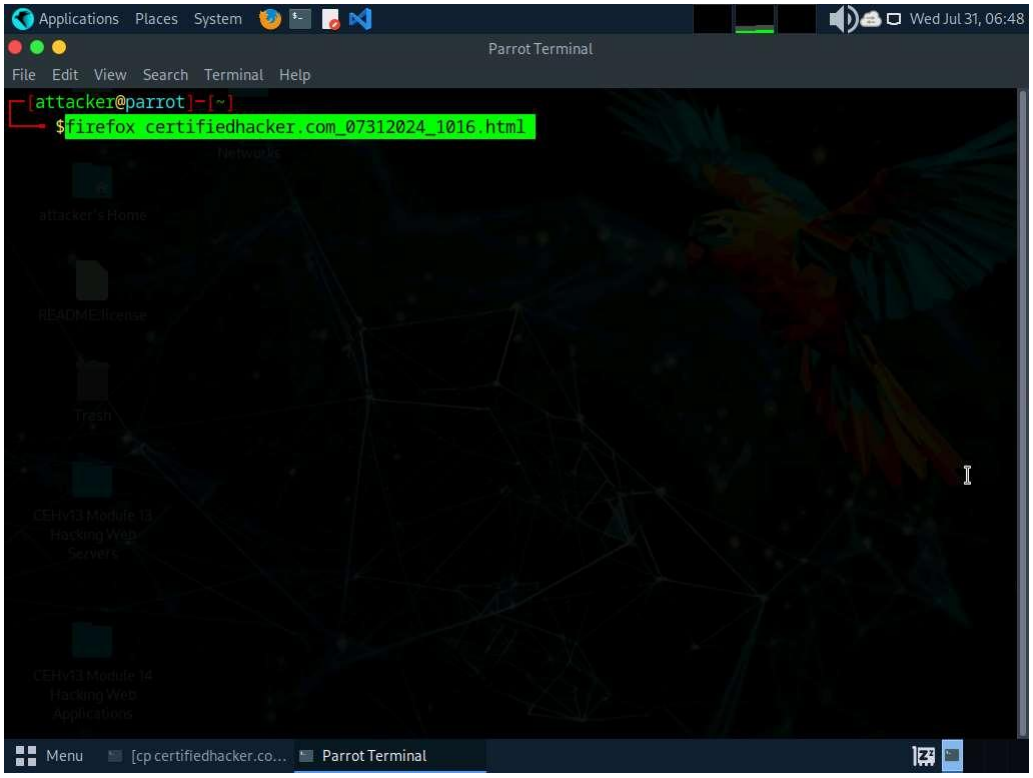
   The name of the .html file varies when you perform this lab.



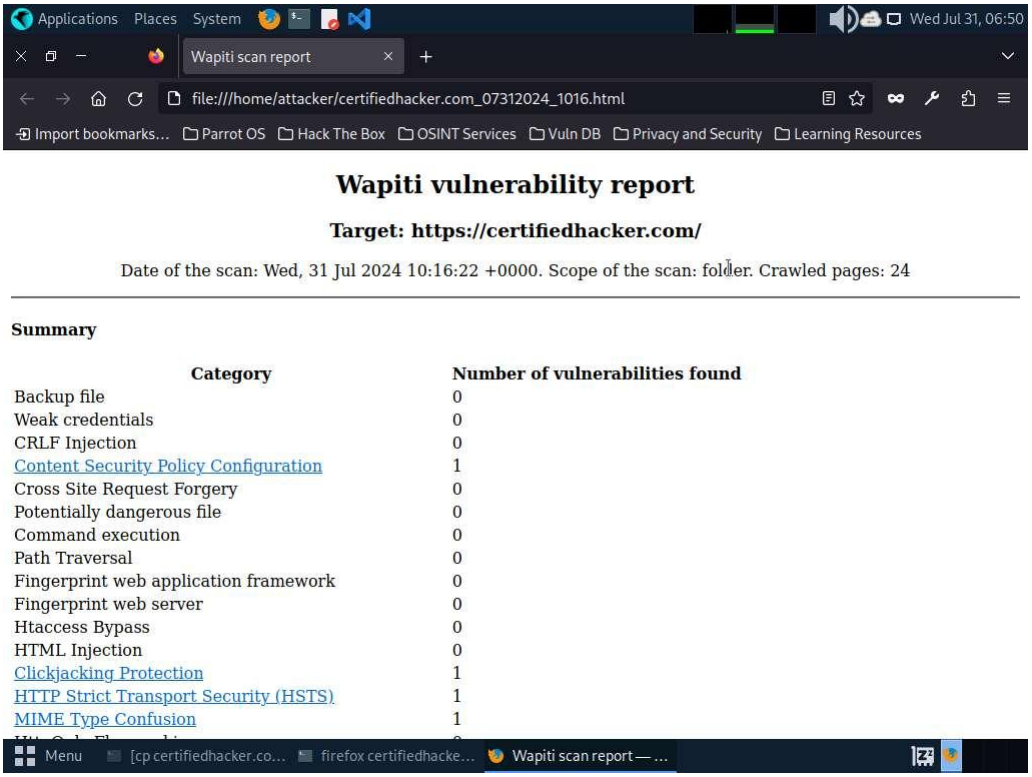
8. Run **cp certifiedhacker.com_xxxxxxxx_xxxx.html /home/attacker/** command to copy the .html file to **/home/attacker** location.



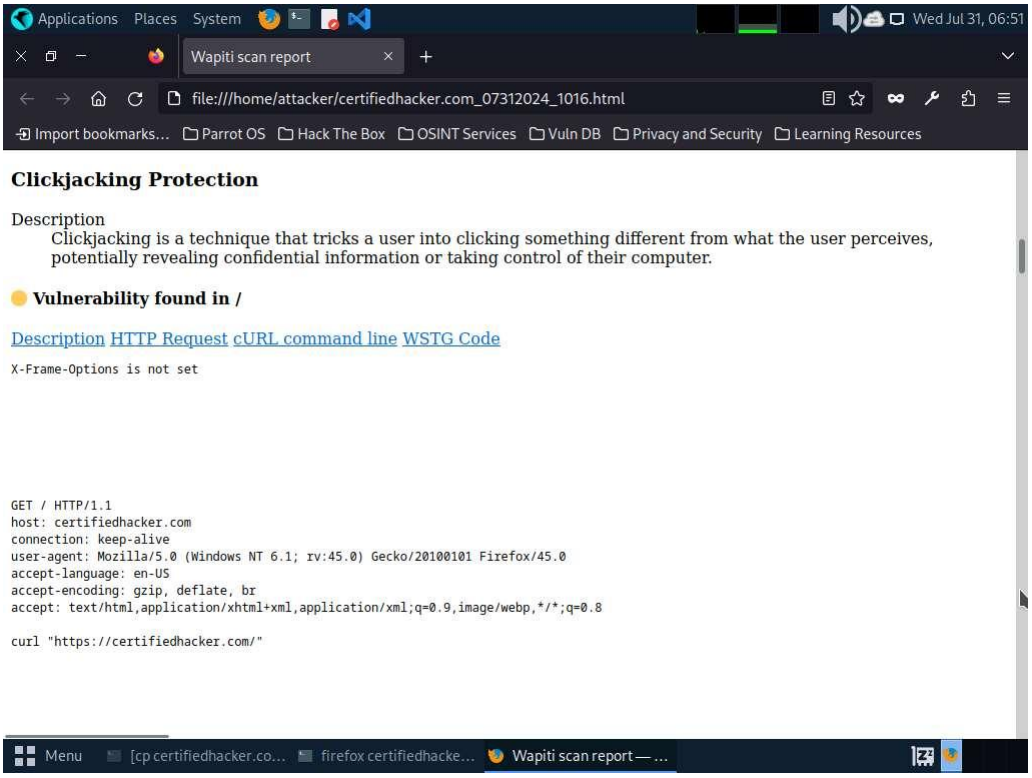
9. Open a new terminal and run **firefox certifiedhacker.com_xxxxxxxx_xxxx.html** command to open the .html file in Firefox browser.



10. Wapiti scan report opens upp in Firefox browser, you can analyze the scan result with the discovered vulnerabilities.



11. Scroll down to view the detailed information regarding each discovered vulnerability.



12. This concludes the demonstration of discovering vulnerabilities in a target website scanning using wapiti.

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

Question 14.3.1.1

In Parrot Security machine use wapiti web application security scanner to detect web application vulnerabilities of https://www.certifiedhacker.com web application and generate a .html report. Enter the WSTG code of the Clickjacking Protection vulnerability.

Lab 4: Perform Web Application Hacking using AI

Lab Scenario

Hacking web applications using AI involves leveraging advanced machine learning techniques to exploit vulnerabilities in web applications. This approach can automate and enhance the traditional methods of penetration testing and vulnerability assessment.

The labs in this exercise demonstrate how to perform web application hacking using AI.

Lab Objectives

- Perform web application hacking using ShellGPT

Overview of Web Application Hacking using AI

Web application hacking using AI represents a sophisticated evolution in cyber threats, leveraging advanced machine learning algorithms and techniques to identify vulnerabilities, create exploits, bypass defenses, and extract sensitive information from web applications.

Task 1: Perform Web Application Hacking using ShellGPT

Web application hacking with ShellGPT involves leveraging AI-generated commands to exploit vulnerabilities, execute code injections, bypass security measures like WAFs, and extract sensitive data. It automates attack vectors, adapts to defenses, and poses a sophisticated threat requiring advanced defensive strategies.

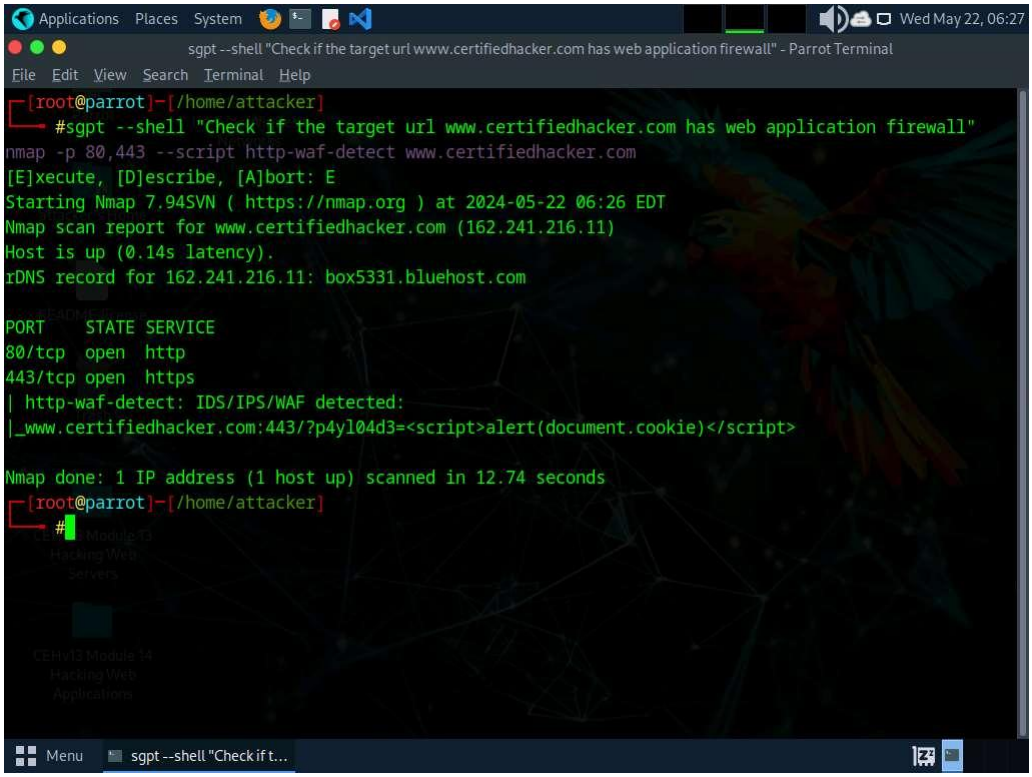
The commands generated by ShellGPT may vary depending on the prompt used and the tools available on the machine. Due to these variables, the output generated by ShellGPT might differ from what is shown in the screenshots. These differences arise from the dynamic nature of the AI's processing and the diverse environments in which it operates. As a result, you may observe differences in command syntax, execution, and results while performing this lab task.

1. Before starting this lab, click Parrot Security to switch to the **Parrot Security** machine and incorporate ShellGPT by following steps provided in Integrate ShellGPT in Parrot Security Machine.pdf.

   Alternatively, you can follow the steps to integrate **ShellGPT** provided in **Module 00: Integrate ShellGPT in Parrot Security Machine**.

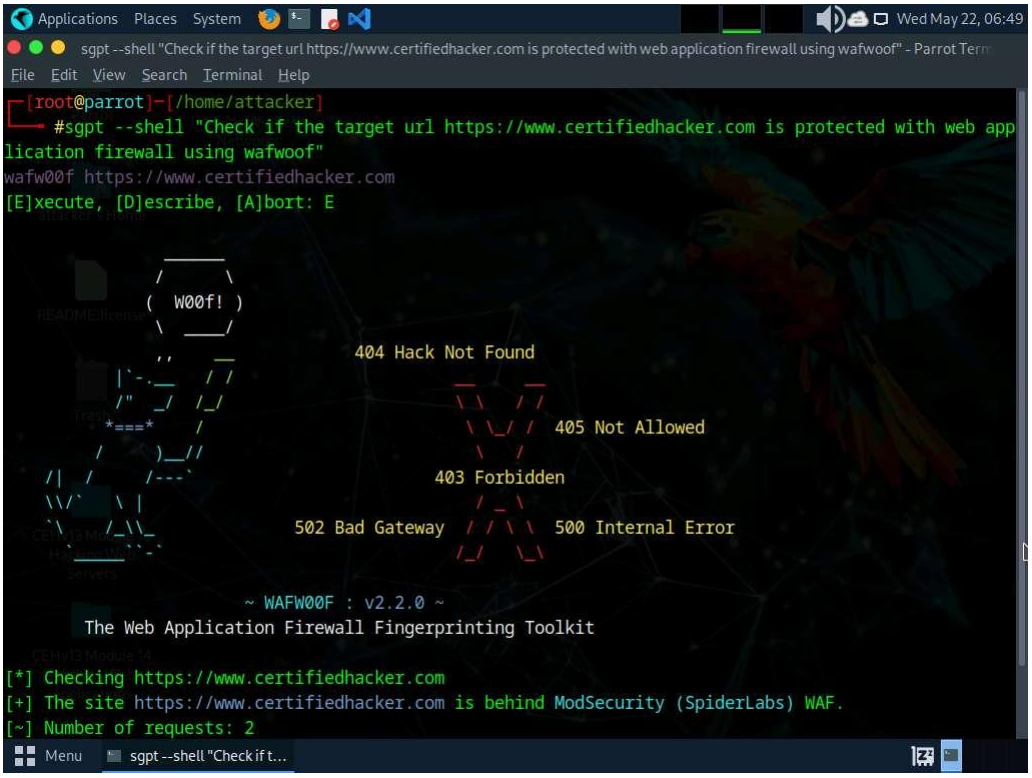
2. After incorporating the ShellGPT API in Parrot Security Machine, in the terminal window run **sgpt --shell "Check if the target url www.certifiedhacker.com has web application firewall"** command to detect WAF using ShellGPT.

   In the prompt type **E** and press **Enter** to execute the command.



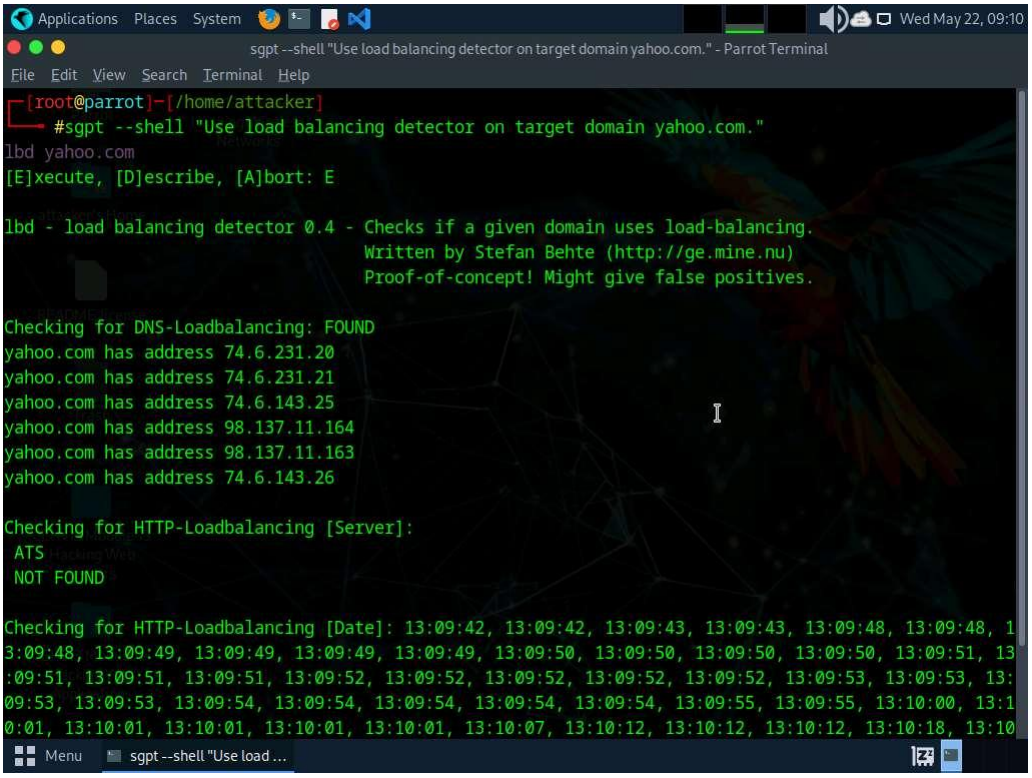
3. Now, run **sgpt --shell "Check if the target url https://www.certifiedhacker.com is protected with web application firewall using wafwoof"** command to check for WAF using wafwoof.

   In the prompt type **E** and press **Enter** to execute the command.



4. To detect load balancers using ShellGPT run **sgpt --shell "Use load balancing detector on target domain yahoo.com."** command.

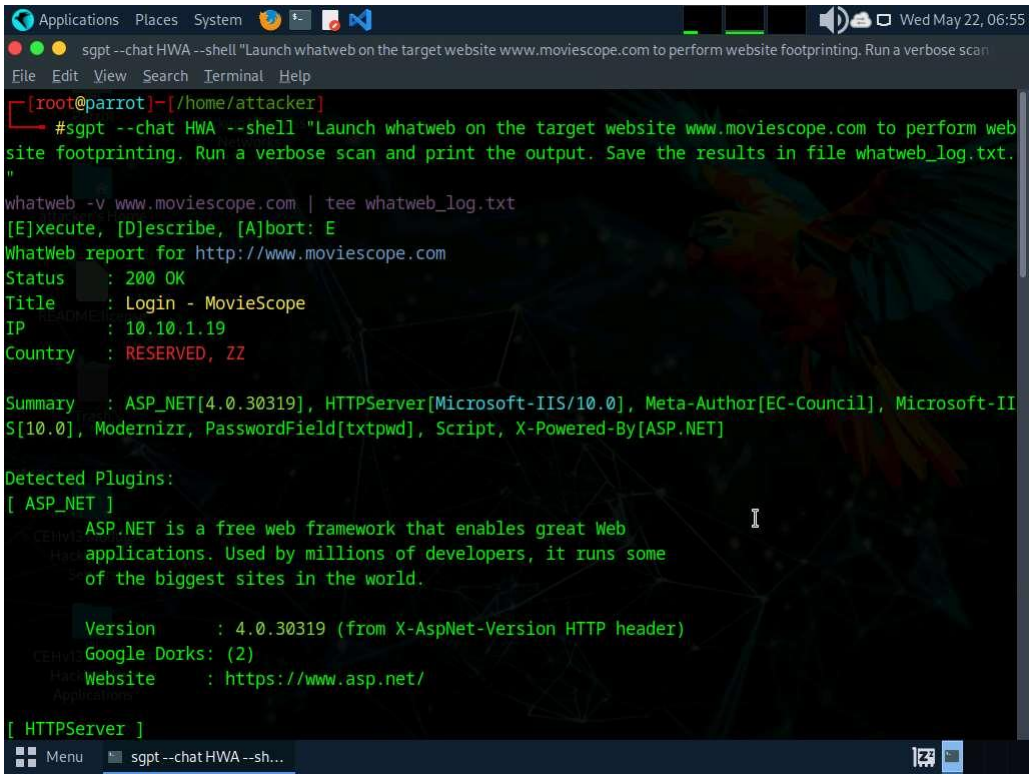
   In the prompt type **E** and press **Enter** to execute the command.



5. To identify server side technologies using ShellGPT run **sgpt --chat HWA --shell "Launch whatweb on the target website www.moviescope.com to perform website footprinting. Run a verbose scan and print the output. Save the results in file whatweb_log.txt."** command.

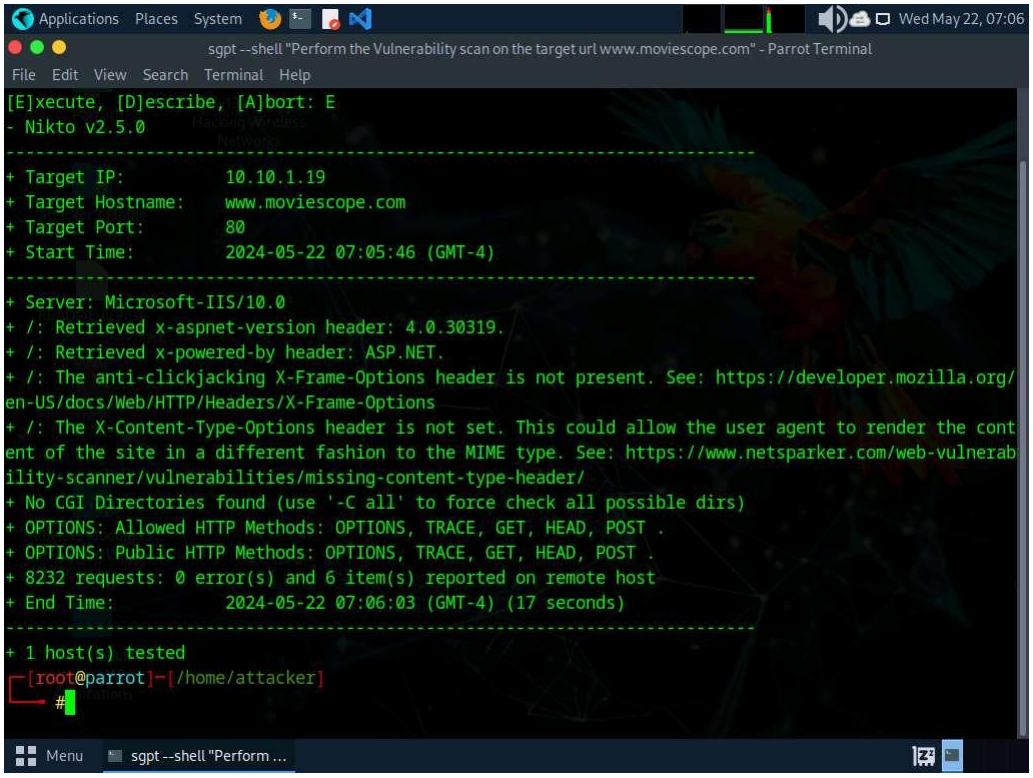
   In the prompt type **E** and press **Enter** to execute the command.

   To view the generated **whatweb_log.txt** file contents, navigate to **/home/attacker** and double-click on whatweb_log.txt file.



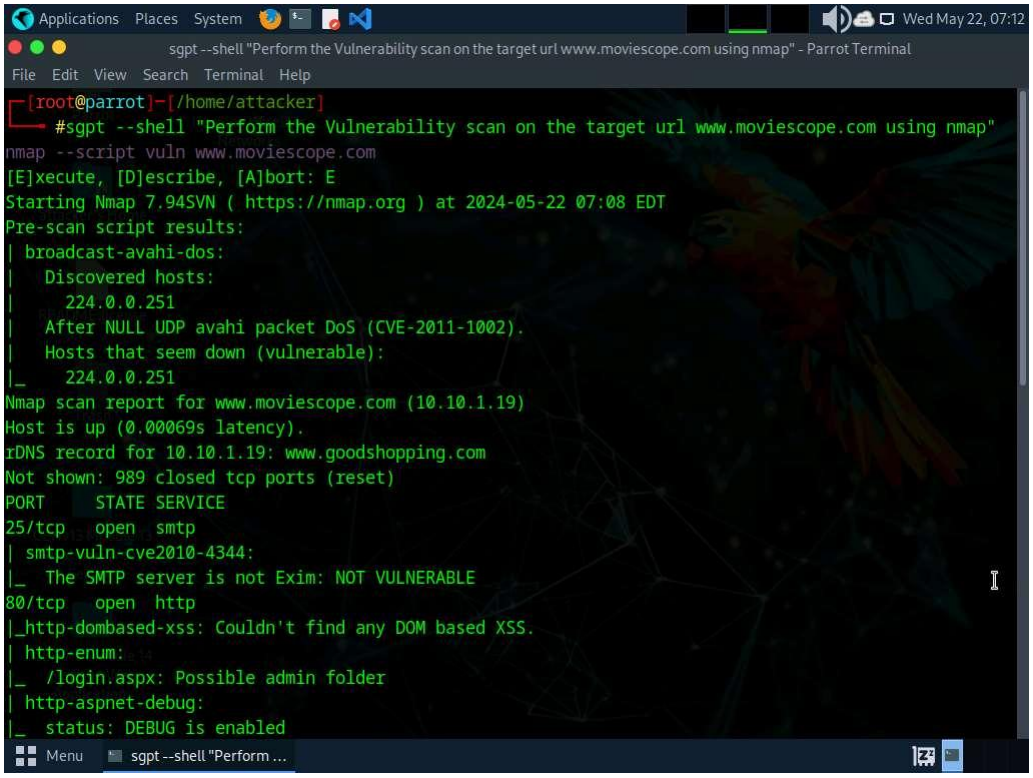
6. Now run **sgpt --shell "Perform the Vulnerability scan on the target url www.moviescope.com"** command to identify web application vulnerabilities on a target website.

   In the prompt type **E** and press **Enter** to execute the command.



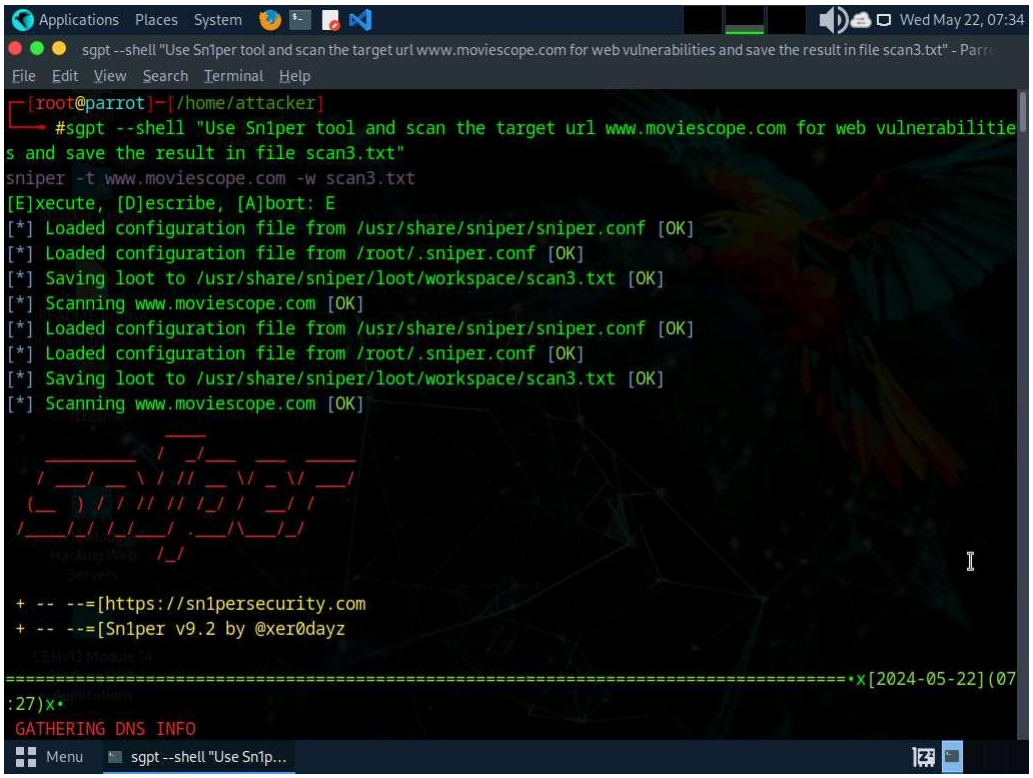
7. Run **sgpt --shell "Perform the Vulnerability scan on the target url www.moviescope.com using nmap"** command to perform web application scanning using Nmap.

   In the prompt type **E** and press **Enter** to execute the command.



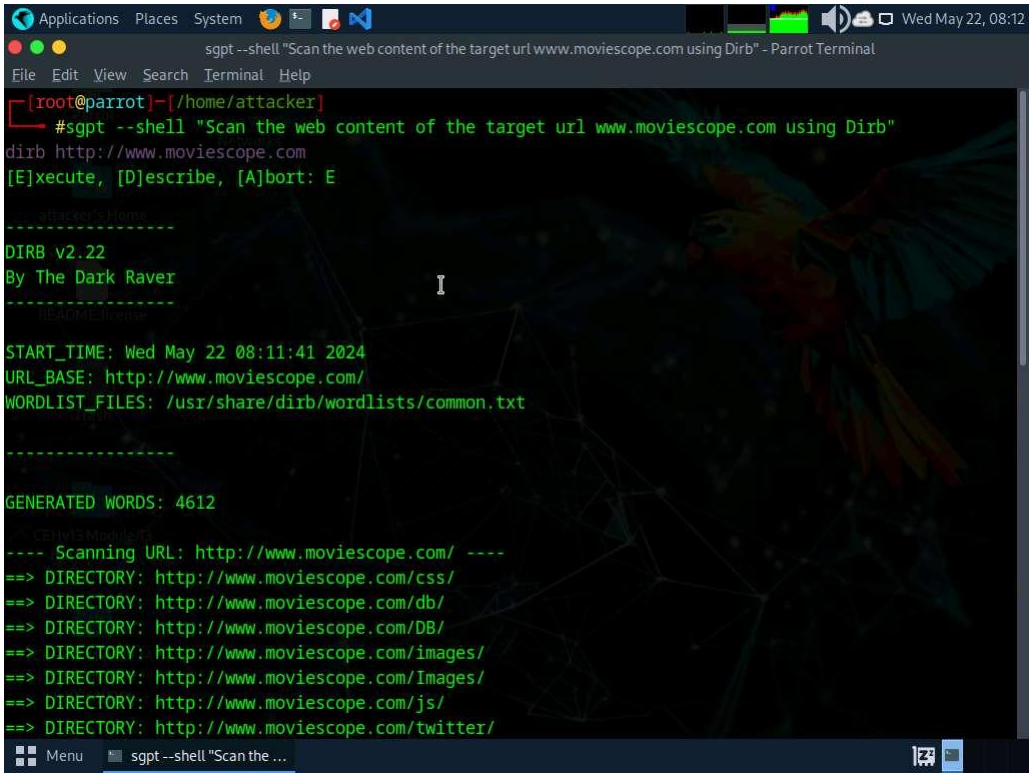
8. To perform a vulnerability scan on web application using Sniper tool run **sgpt --shell "Use Sn1per tool and scan the target url www.moviescope.com for web vulnerabilities and save result in file scan3.txt"** command.

   In the prompt type **E** and press **Enter** to execute the command.



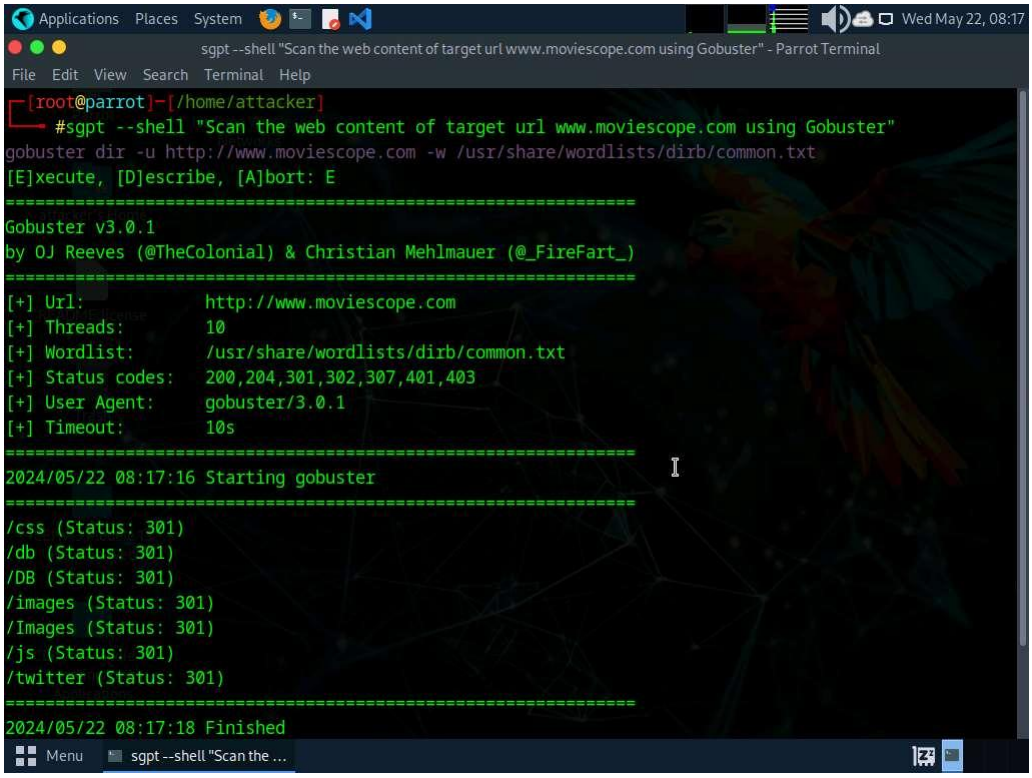
9. To identify files of a web application run **sgpt --shell "Scan the web content of target url www.moviescope.com using Dirb"** command.

   In the prompt type **E** and press **Enter** to execute the command.



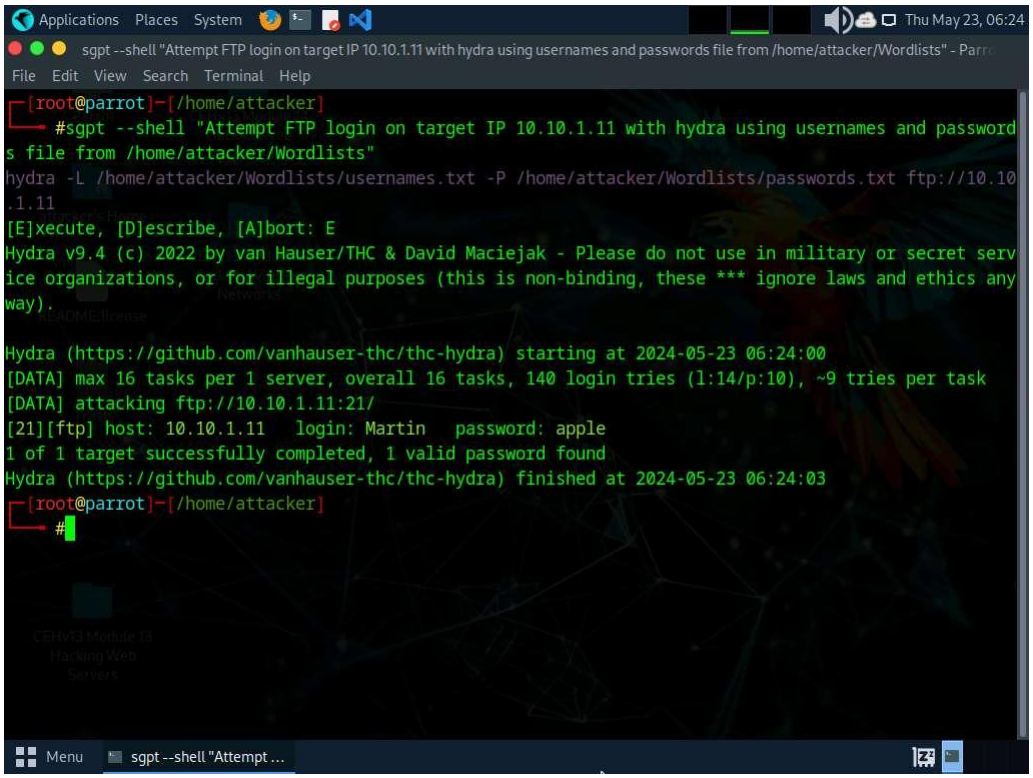
10. Run **sgpt --shell "Scan the web content of target url www.moviescope.com using Gobuster"** command to identify directories using Gobuster.

    In the prompt type **E** and press **Enter** to execute the command.



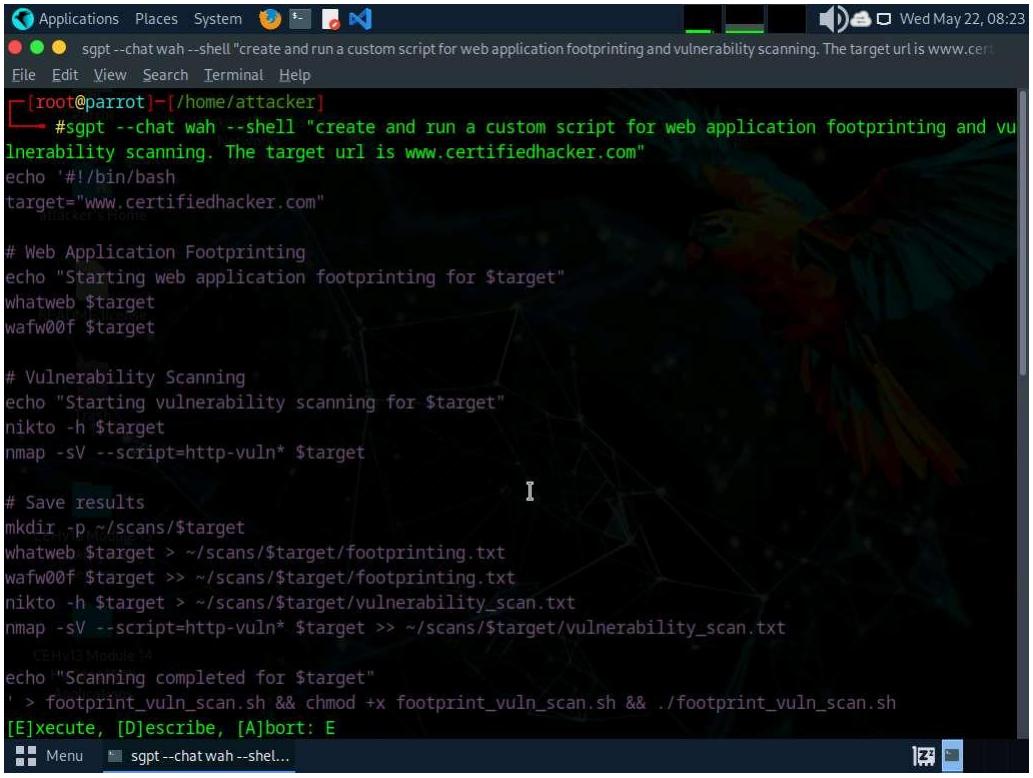
11. To perform FTP bruteforce attack run **sgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists"** command.

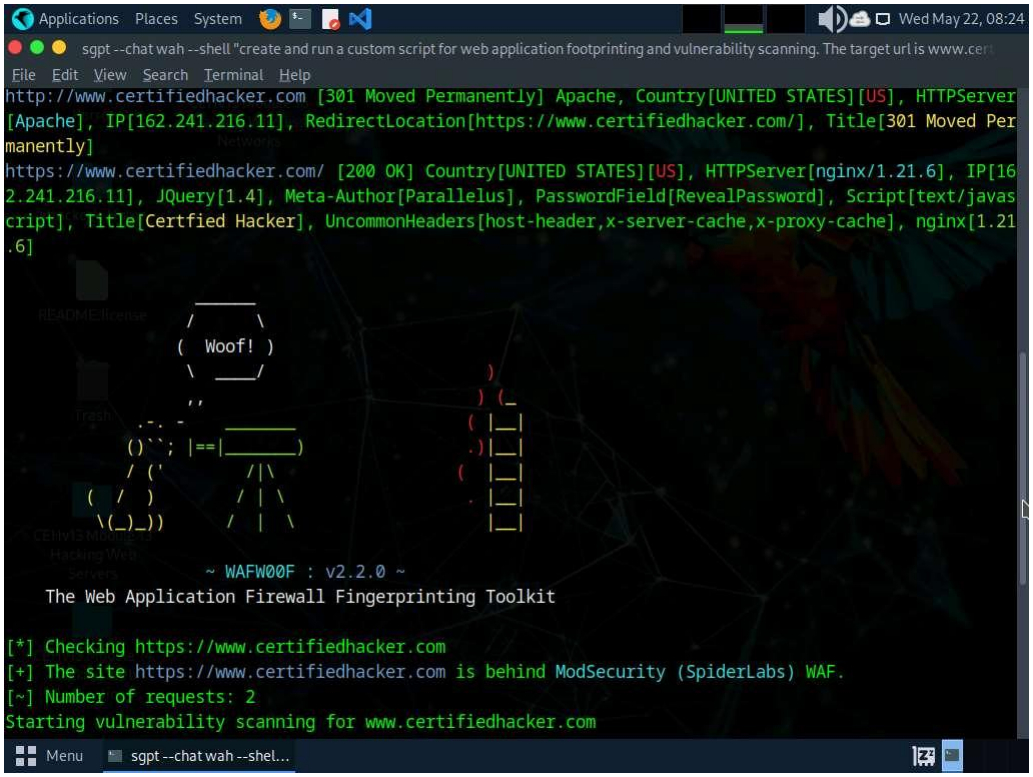
    In the prompt type **E** and press **Enter** to execute the command.

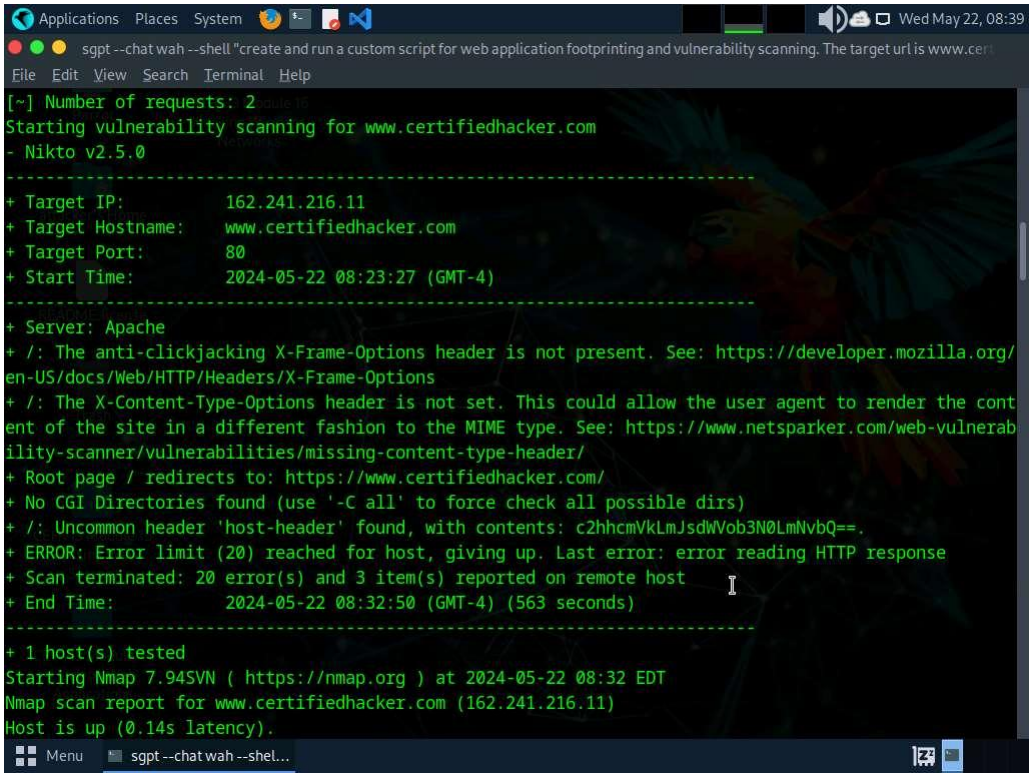


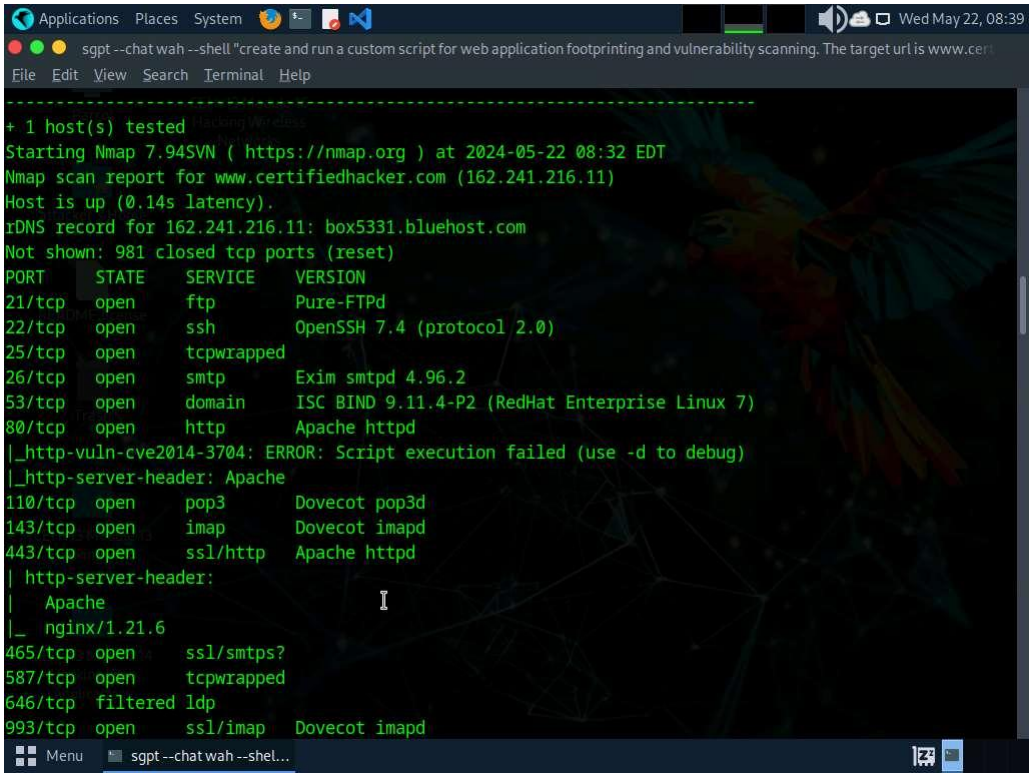
12. Run **sgpt --chat wah --shell “create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com”** to automate web application hacking tasks with custom scripts.

    In the prompt type **E** and press **Enter** to execute the command.


Applications Places System
Wed May 22, 08:23
sgpt --chat wah --shell "create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.cert
File Edit View Search Terminal Help
[root@parrot]-[/home/attacker]
#sgpt --chat wah --shell "create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com"
echo '#!/bin/bash
target="www.certifiedhacker.com"
# Web Application Footprinting
echo "Starting web application footprinting for $target"
whatweb $target
wafw00f $target
Vulnerability Scanning
echo "Starting vulnerability scanning for $target"
nikto -h $target
nmap -sV --script=http-vuln* $target
Save results
mkdir -p ~/scans/$target
whatweb $target > ~/scans/$target/footprinting.txt
wafw00f $target >> ~/scans/$target/footprinting.txt
nikto -h $target > ~/scans/$target/vulnerability_scan.txt
nmap -sV --script=http-vuln* $target >> ~/scans/$target/vulnerability_scan.txt
echo "Scanning completed for $target"
' > footprint_vuln_scan.sh && chmod +x footprint_vuln_scan.sh && ./footprint_vuln_scan.sh
[E]xecute, [D]escribe, [A]bort: E
Menu
sgpt --chat wah --shel...

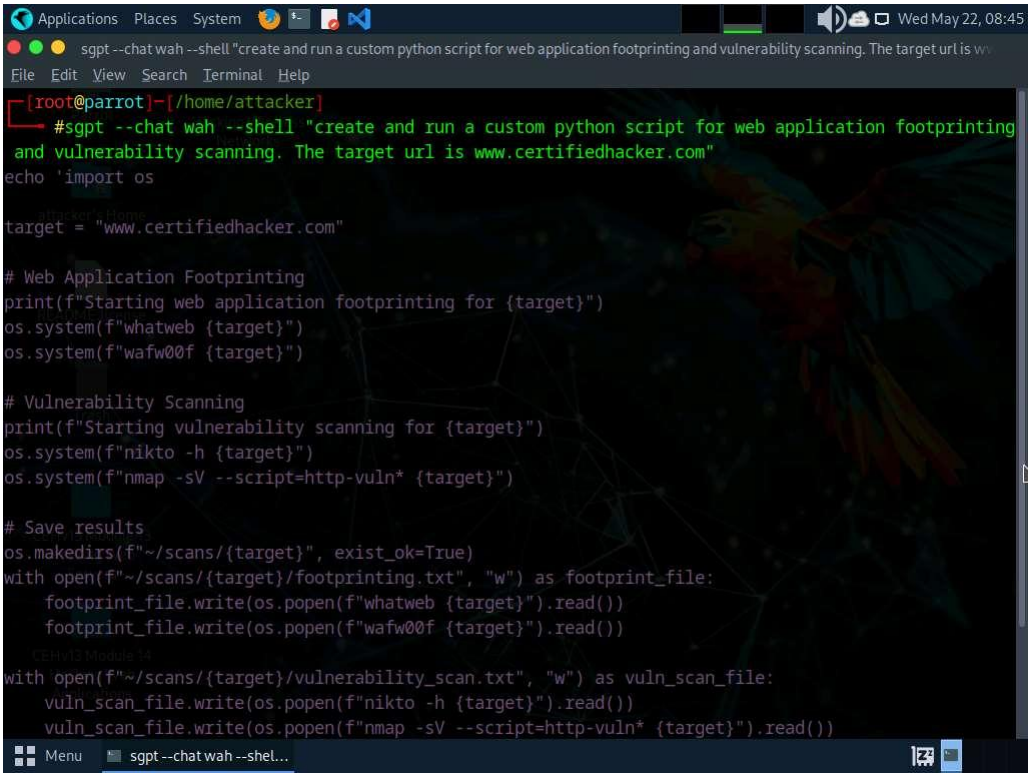

Applications Places System
Wed May 22, 08:24
sgpt --chat wah --shell "create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.cert
File Edit View Search Terminal Help
http://www.certifiedhacker.com [301 Moved Permanently] Apache, Country[UNITED STATES][US], HTTPServer[Apache], IP[162.241.216.11], RedirectLocation[https://www.certifiedhacker.com/], Title[301 Moved Permanently]
https://www.certifiedhacker.com/ [200 OK] Country[UNITED STATES][US], HTTPServer[nginx/1.21.6], IP[162.241.216.11], JQuery[1.4], Meta-Author[Parallelus], PasswordField[RevealPassword], Script[text/javascript], Title[Certfied Hacker], UncommonHeaders[host-header,x-server-cache,x-proxy-cache], nginx[1.21.6]
Woof!
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit
[*] Checking https://www.certifiedhacker.com
[+] The site https://www.certifiedhacker.com is behind ModSecurity (SpiderLabs) WAF.
[~] Number of requests: 2
Starting vulnerability scanning for www.certifiedhacker.com
Menu
sgpt --chat wah --shel...

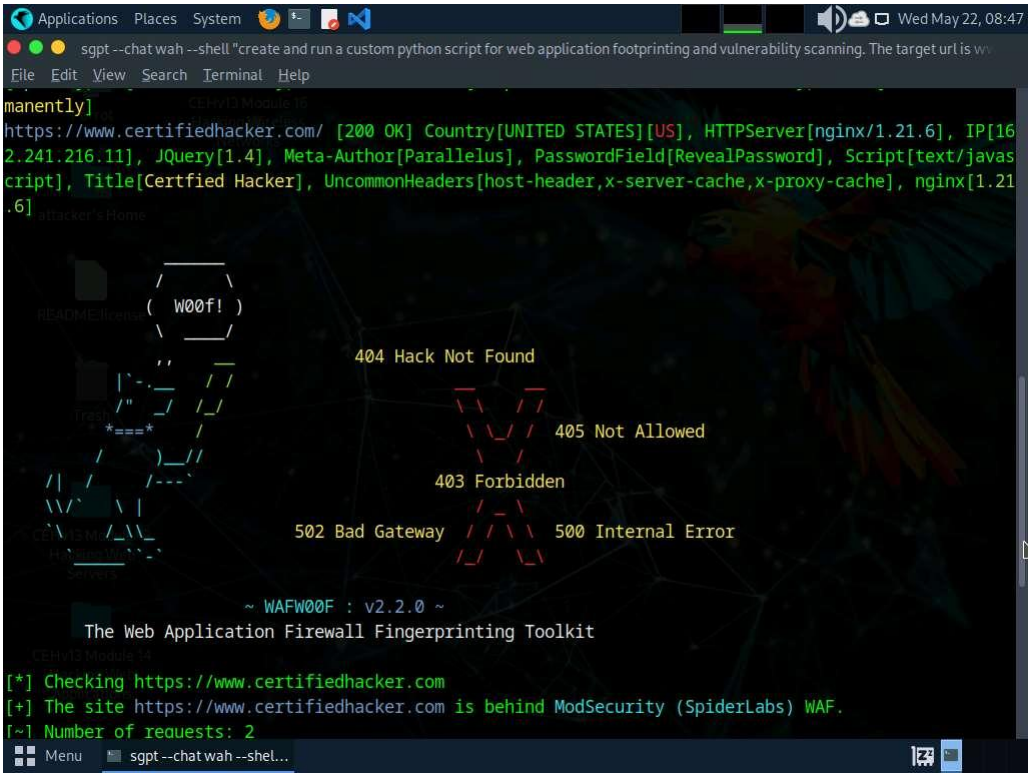


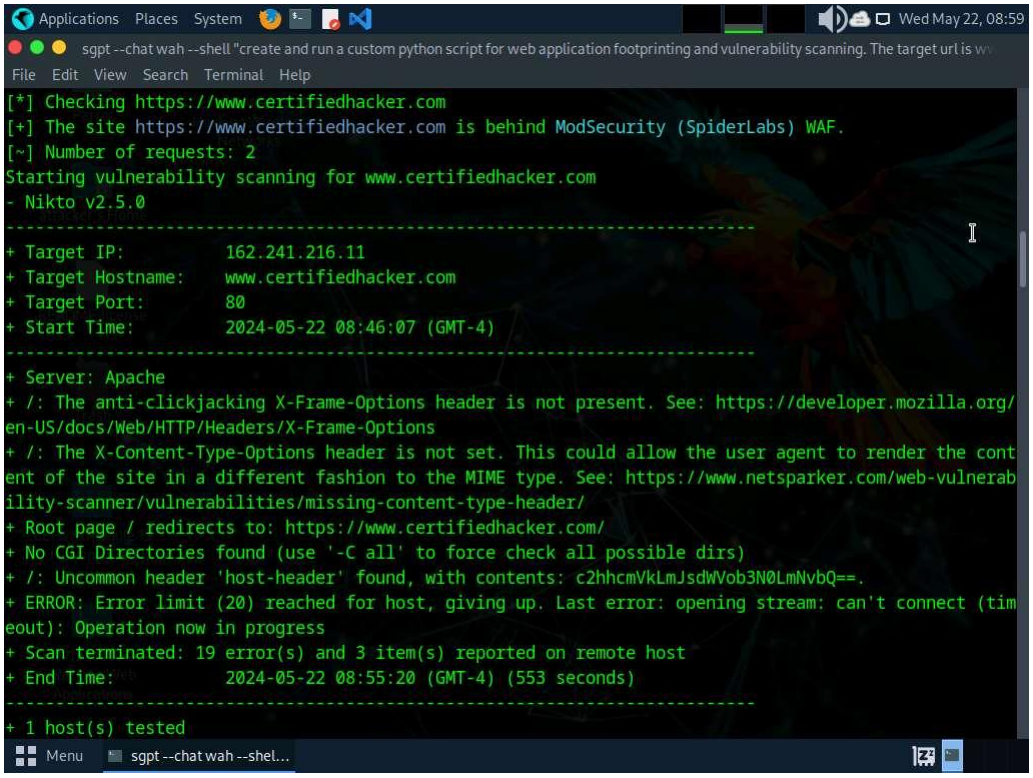


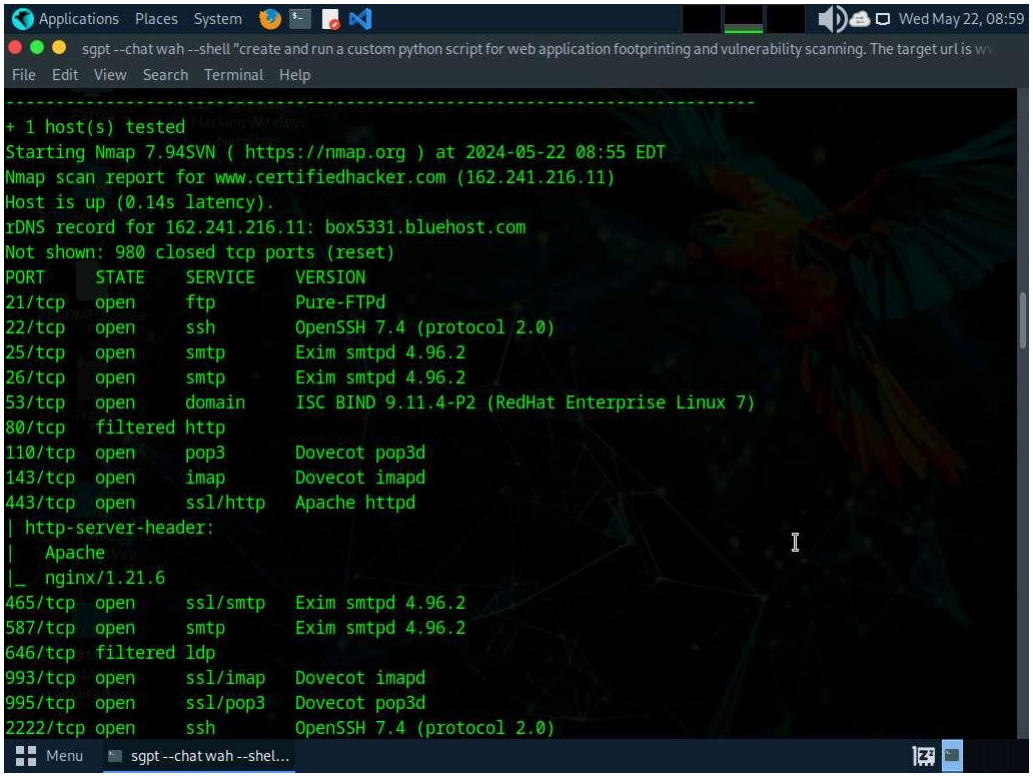
13. To create a custom python script for web application scanning run **sgpt --chat wah --shell "create and run a custom python script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com"** command.

    In the prompt type **E** and press **Enter** to execute the command.


Applications Places System
Wed May 22, 08:45
sgpt --chat wah --shell "create and run a custom python script for web application footprinting and vulnerability scanning. The target url is ww
File Edit View Search Terminal Help
[root@parrot]-[/home/attacker]
#sgpt --chat wah --shell "create and run a custom python script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com"
echo 'import os
target = "www.certifiedhacker.com"
# Web Application Footprinting
print(f"Starting web application footprinting for {target}")
os.system(f"whatweb {target}")
os.system(f"wafw00f {target}")
# Vulnerability Scanning
print(f"Starting vulnerability scanning for {target}")
os.system(f"nikto -h {target}")
os.system(f"nmap -sV --script=http-vuln* {target}")
# Save results
os.makedirs(f"~/scans/{target}", exist_ok=True)
with open(f"~/scans/{target}/footprinting.txt", "w") as footprint_file:
footprint_file.write(os.popen(f"whatweb {target}").read())
footprint_file.write(os.popen(f"wafw00f {target}").read())
with open(f"~/scans/{target}/vulnerability_scan.txt", "w") as vuln_scan_file:
vuln_scan_file.write(os.popen(f"nikto -h {target}").read())
vuln_scan_file.write(os.popen(f"nmap -sV --script=http-vuln* {target}").read())
Menu
sgpt --chat wah --shel...

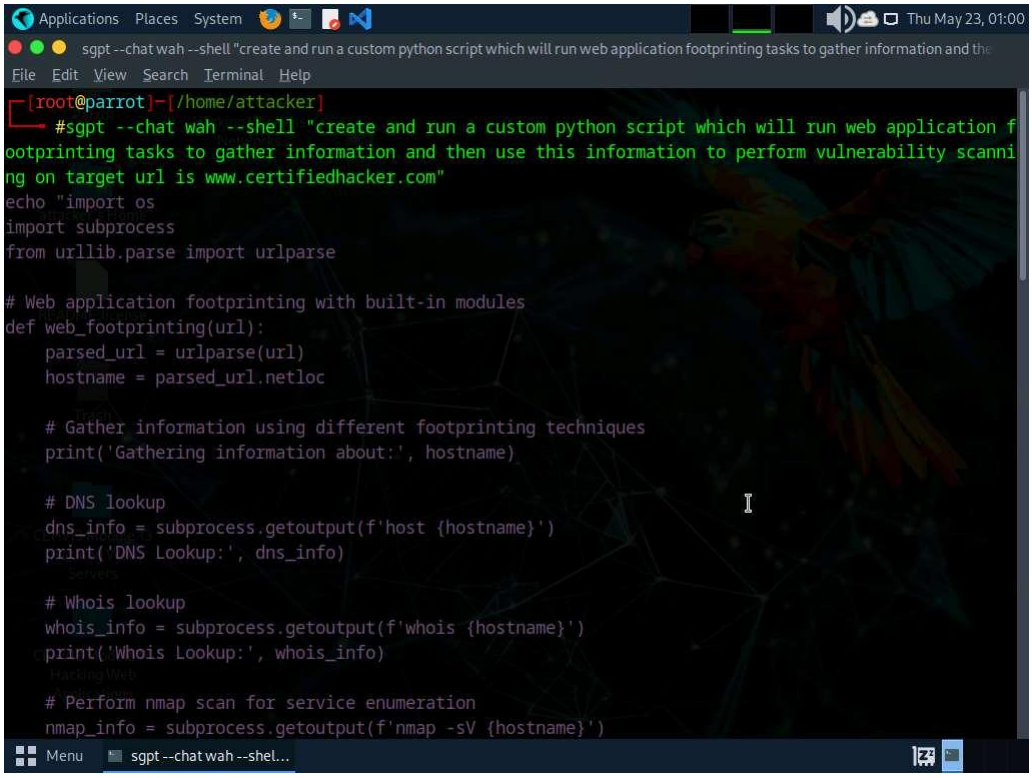

Applications Places System
Wed May 22, 08:47
sgpt --chat wah --shell "create and run a custom python script for web application footprinting and vulnerability scanning. The target url is ww
File Edit View Search Terminal Help
manently]
https://www.certifiedhacker.com/ [200 OK] Country[UNITED STATES][US], HTTPServer[nginx/1.21.6], IP[162.241.216.11], JQuery[1.4], Meta-Author[Parallelus], PasswordField[RevealPassword], Script[text/javascript], Title[Certfied Hacker], UncommonHeaders[host-header,x-server-cache,x-proxy-cache], nginx[1.21.6]
W00f!
404 Hack Not Found
405 Not Allowed
403 Forbidden
502 Bad Gateway
500 Internal Error
~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit
[*] Checking https://www.certifiedhacker.com
[+] The site https://www.certifiedhacker.com is behind ModSecurity (SpiderLabs) WAF.
[~] Number of requests: 2
Menu
sgpt --chat wah --shel...

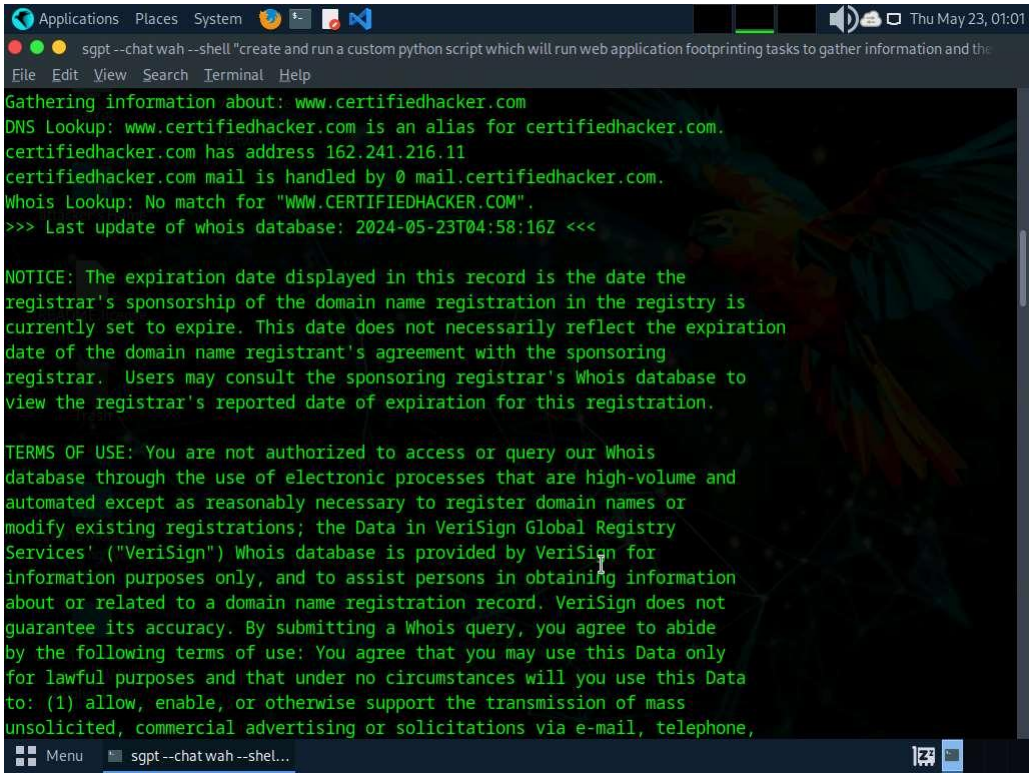


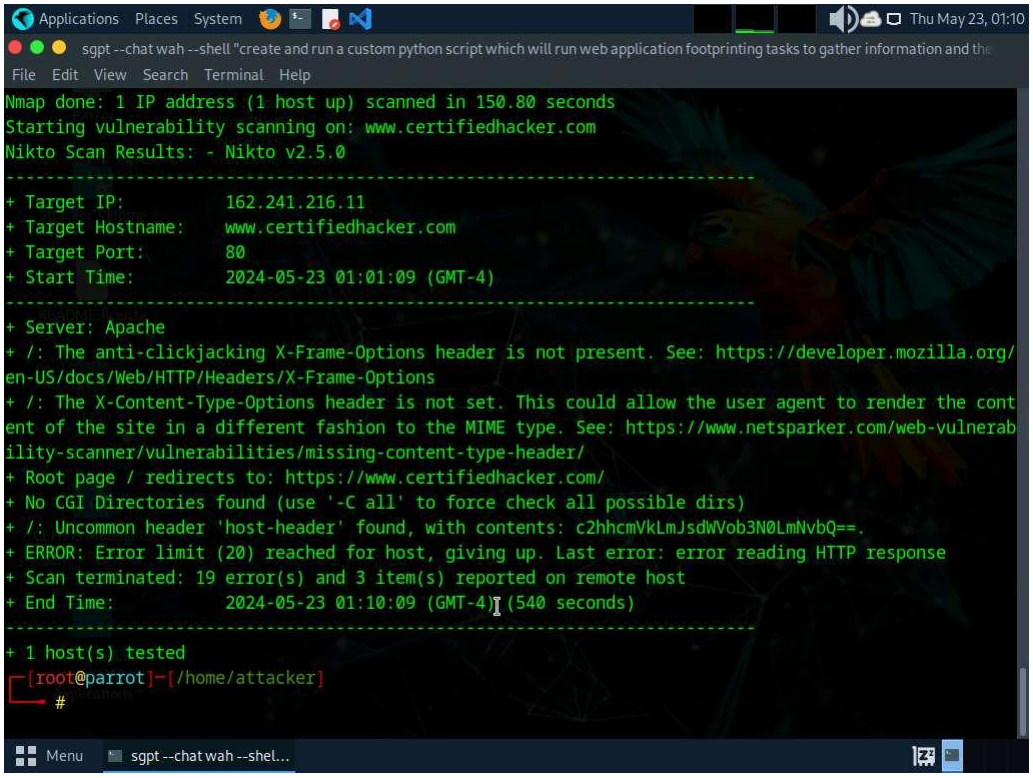


14. To create a custom python script for web application scanning run **sgpt --chat wah --shell "create and run a custom python script which will run web application footprinting tasks to gather information and then use this information to perform vulnerability scanning on target url is www.certifiedhacker.com"** command.

In the prompt type **E** and press **Enter** to execute the command.

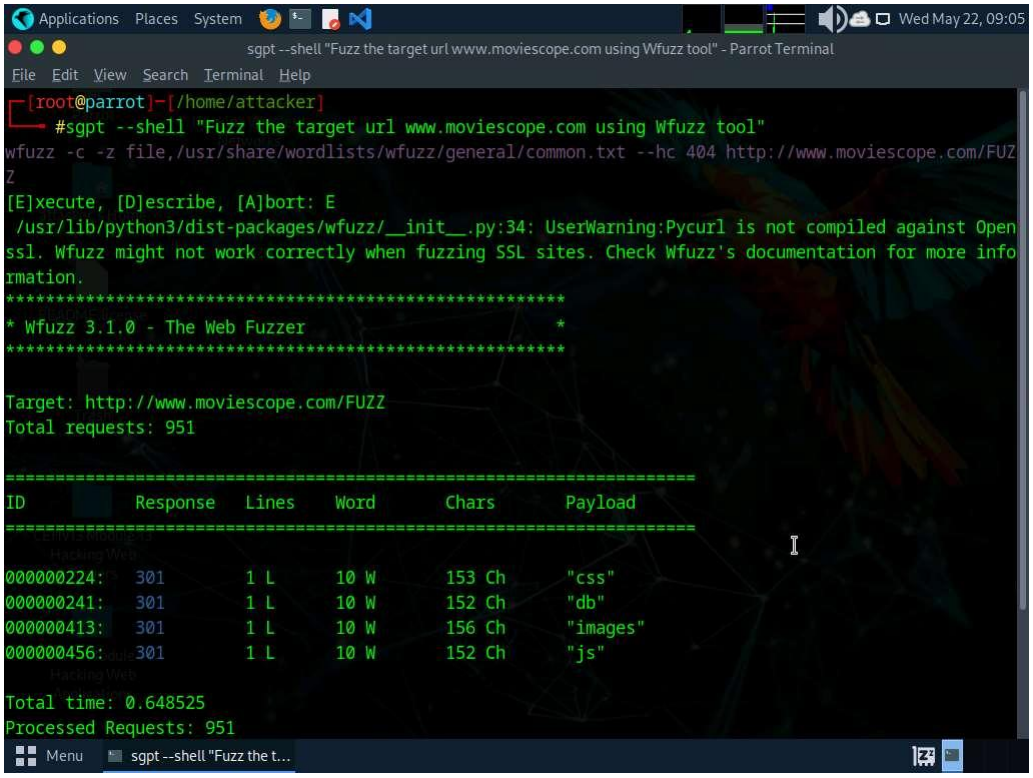






15. To perform Web application fuzz testing using ShellGPT run **sgpt --shell "Fuzz the target url www.moviescope.com using Wfuzz tool"** command.

    In the prompt type **E** and press **Enter** to execute the command.



16. Apart from the aforementioned commands, you can further use ShellGPT prompts to perform Web Application Hacking.

17. This concludes the demonstration of Webserver footprinting and attacks using ShellGPT.

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

Question 14.4.1.1

Write a prompt using ShellGPT and execute it on the Parrot Security machine to check if website https://www.certifiedhacker.com is protected with a web application firewall using wafwoof. Enter the name of the web application firewall found during the scan.