The Binding of Cyber

CTF Challenge Write-up

Penetration Testing Report

ENUMERATION

EXPLOITATION

PRIVILEGE ESC

Executive Summary

This write-up documents the methodology and techniques used to successfully compromise a deliberately vulnerable CTF machine. The challenge required systematic enumeration of over 900 open ports, NFS share exploitation, password cracking, and Linux privilege escalation through capability abuse.

Attack Path Summary: Port Scanning ⟶ NFS Discovery ⟶ Password Cracking ⟶ SSH Access ⟶ Capability Exploitation ⟶ Root Flag

Reconnaissance Phase

Initial Network Enumeration

The reconnaissance phase began with standard port scanning, which immediately revealed an unusual characteristic: the target system had an exceptionally large number of open ports, making conventional scanning approaches inefficient.

Port 109 Investigation

Initial scanning of port 109 revealed an interesting message suggesting the service was "lost" and typically operates on the top 100 ports.

```
nmap -p 109 -sC -sV <TARGET_IP>
```

Tool: Nmap Flags Explained

- `-p 109`: Scans only port 109
- `-sC`: Executes default NSE scripts for service detection
- `-sV`: Performs version detection on discovered services

Scanning Top 100 Ports

Following the hint from port 109, a comprehensive scan of the top 100 commonly used ports was conducted.

```
nmap --top-ports 100 -sC -sV <TARGET_IP>
```

Tool: Additional Nmap Options

- `--top-ports 100`: Focuses the scan on the 100 most commonly used ports
- This approach significantly reduces scan time while maintaining good coverage

The results revealed multiple services with unusual configurations across various ports, though most appeared to be decoys or red herrings.

Sequential Port Probing

To systematically explore the first 100 ports, a bash script was developed to connect to each port using netcat and capture any responses.

```
for i in {1..100}; do
  echo "Attempting connection to port $i"
  nc <TARGET_IP> $i
done
```

Tool: Netcat (nc)

Netcat is a versatile networking utility that can read and write data across network connections using TCP or UDP protocols. It's often called the "Swiss Army knife" of networking tools.

✓ Hidden message discovered across ports 51-64, directing to port 23456

Port 23456 Discovery

Connecting to port 23456 yielded critical intelligence about the location of flag files.

```
nc <TARGET_IP> 23456
```

The response indicated that flags were stored in a chest accessible via an NFS (Network File System) share.

NFS Share Enumeration

Discovering Exported Shares

Using the showmount utility, available NFS exports were enumerated.

```
showmount -e <TARGET_IP>
```

Tool: Showmount

Showmount queries the NFS server to display information about exported directories. The `-e` flag specifically lists all exported filesystems.

✓ **Exported directory discovered:** `/home/nfs`

▸ Mounting the NFS Share

The discovered NFS share was mounted to the local filesystem for inspection.

```
mkdir /mnt/nfs
mount <TARGET_IP>:/home/nfs /mnt/nfs
ls -la /mnt/nfs
```

Within the mounted share, a password-protected archive named `chest.zip` was discovered.

Credential Attacks

Password Cracking the Archive

The protected zip file required brute-force password cracking to access its contents.

```
zip2john chest.zip > zip.hash
john --wordlist=/usr/share/wordlists/rockyou.txt zip.hash
john --show zip.hash
```

Tool: Zip2john

- zip2john : Extracts password hash from a ZIP archive
- zip2john has no cracking flags, it only converts the ZIP into a crackable hash for John

✓ **Password recovered:** `isaaciscrazy`

Archive Contents Analysis

The extracted archive revealed an SSH configuration directory containing:

- Private SSH key (`id_rsa`)
- Public SSH key (`id_rsa.pub`)
- Authorized keys file
- A hint file containing the port range `5000-6500`
- The first flag EPI{4ch13V3M3N7_7R0Phy_90lD3n_90d}

Initial Access

SSH Port Discovery

Based on the hint file, the port range 5000-6500 was scanned to identify SSH services.

```
nmap -p 5000-6500 -sV --open <TARGET_IP> | grep ssh
```

This scan identified multiple SSH services running on non-standard ports, requiring systematic testing with the recovered private key.

SSH Authentication

Through methodical testing of discovered SSH ports, successful authentication was achieved on port 5555.

```
ssh isaac@<TARGET_IP> -i id_rsa -p 5555
```

Tool: SSH Flags

- `-i id_rsa`: Specifies the private key file for authentication
- `-p 5555`: Connects to SSH on the non-standard port 5555

The initial shell presented as an Interactive Ruby (IRB) prompt, requiring escape to obtain a standard bash shell.

```
exec '/bin/bash'
```

✓ **Shell access obtained as user `isaac`**

Privilege Escalation

User Flag Recovery

Upon gaining standard shell access as the `isaac` user, the user flag was immediately accessible in the home directory.

```
cat ~/user.txt
```

USER FLAG: EPI{4Ch13V3m3N7_7r0pHy_Pl471NUm_90D}

Capability Enumeration

Linux capabilities provide a fine-grained permission system that can sometimes be exploited for privilege escalation. A recursive capability scan was performed across the filesystem.

```
getcap -r / 2>/dev/null
```

Tool: Getcap

The `getcap` utility displays file capabilities. The `-r` flag performs recursive searching, while `2>/dev/null` suppresses error messages from inaccessible directories.

▸ Critical Finding:

The "tar" binary had the “cap_dac_read_search” capability, which allows bypassing file read permission checks.

Exploiting TAR Capabilities

Leveraging the GTFOBins database for capability exploitation techniques, a method to read arbitrary files was identified.

```
tar xf /root/root.txt -I '/bin/sh -c "cat 1>&2"'
```

Tool: TAR Capability Exploit

This technique abuses tar's ability to execute arbitrary commands through the `-I` flag. The capability `cap_dac_read_search` allows tar to read files regardless of standard Unix permissions, effectively granting read access to root-owned files.

✓ **Root flag: EPI{4Ch13v3m3n7_7R0pHy_7H3_r44l_pl471Num_90d}**

Conclusion

This CTF challenge demonstrated several important penetration testing concepts:

- **Systematic Enumeration:** The importance of methodical port scanning and service discovery, even when faced with an unusually large attack surface
- **NFS Security:** The risks associated with improperly configured network file shares
- **Weak Authentication:** The vulnerability of password-protected archives to dictionary attacks
- **SSH Hardening:** The security implications of running SSH on non-standard ports
- **Linux Capabilities:** How misconfigured file capabilities can lead to privilege escalation without traditional SUID binaries

The successful compromise highlighted the critical need for defense in depth, proper capability management, and secure configuration of network services.