Penetration Testing and Ethical Hacking

RELQ technology school

- Date: May 2025
- Author: Babakhanyan Tamara

View the GitHub Repository

Starting out in Cybersecurity, HackTheBox (HTB) has been the go-to resource provided to me or anyone interested in Penetration Testing and Ethical Hacking for that matter.

👩‍🎓 Introduction

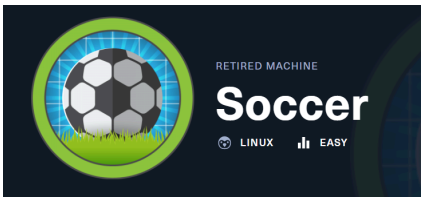
As a student exploring the world of cybersecurity, I quickly discovered that learning about penetration testing and ethical hacking isn't just about reading theory—it's about **thinking like an attacker** and **solving problems hands-on**. These fields focus on testing the security of systems by simulating real cyberattacks, but with permission and the goal of helping organizations fix their weaknesses before real threats exploit them.

This documentation is a reflection of my learning journey into **ethical hacking**, where I've practiced through solving retired machines on **Hack The Box (HTB)**—a popular online platform for cybersecurity training. Each machine presents a unique scenario that requires scanning, exploiting vulnerabilities, escalating privileges, and thinking critically to reach the goal.

By solving and documenting at least 10 HTB retired boxes, I aim to strengthen my skills in recon, exploitation, and post-exploitation techniques. This project not only demonstrates my technical progress but also shows my commitment to responsible hacking practices and continuous learning in the cybersecurity field.

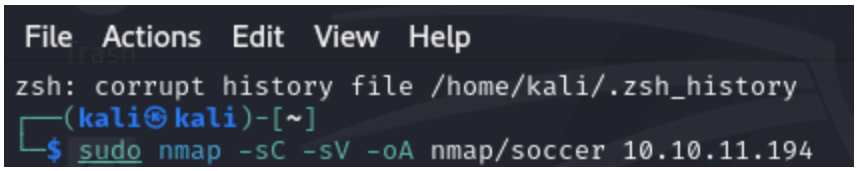
Linux Boxes	Difficulty	Tags
Soccer	Easy	Nmap, Gobuster, BurpSuite, SQLMap
Precious	Easy	Nmap,, BurpSuite, sudo
MetaTwo	Easy	Nmap, FTP, WPScan, SQLMap,, PassPie
Paper	Easy	Nmap, Feroxbuster, Gobuster, WPScan
Shoppy	Easy	Nmap, BurpSuite,ffuf, CrackStation
UnderPass	Easy	Nmap, Gobuster, BurpSuite, ffuf
Knife	Easy	Nmap,BurpSuite,Cookie

This section outlines the systematic approach I employed to tackle each Hack The Box (HTB) retired machine.

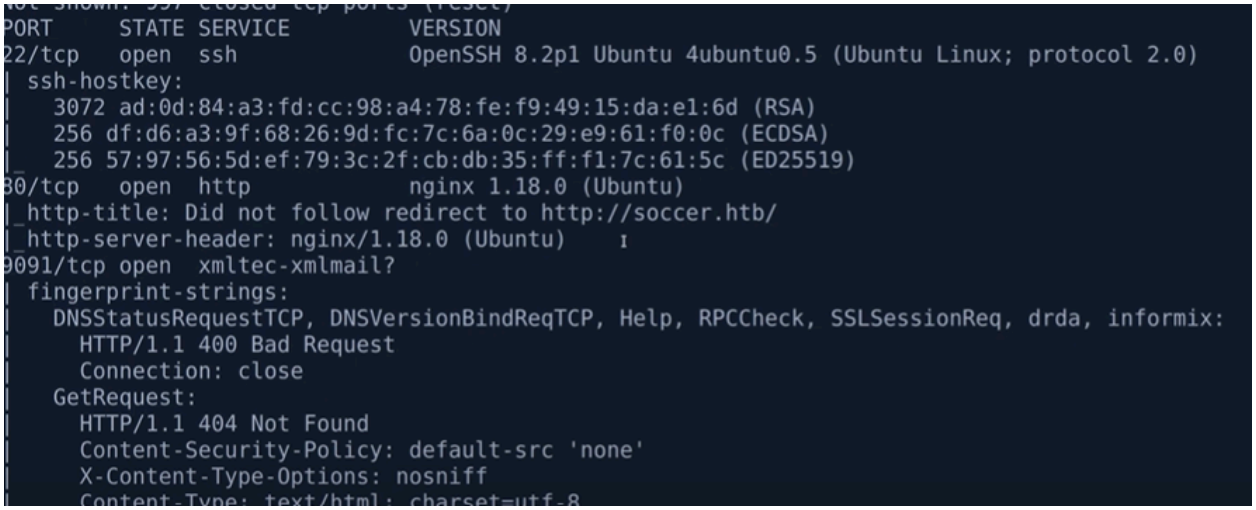


#Soccer/easy

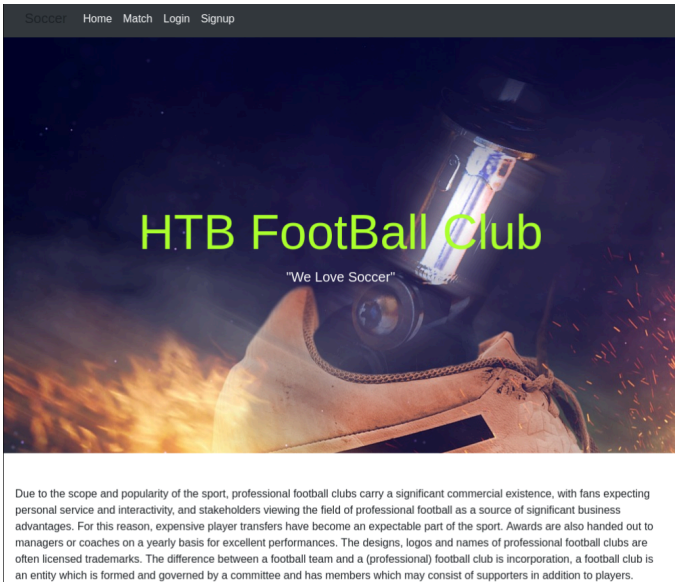
1.To begin the enumeration phase, I used **Nmap** to perform an initial scan of the target machine.

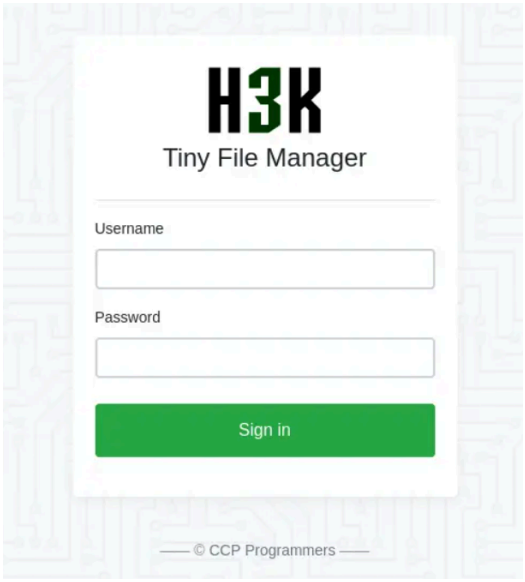


I've already ran it looking at the results we have just three ports open the first one being SSH on Port 22 from the banner we can see it's a new Ubuntu Server we also have HTTP open on Port 80. it is running engine X also oUbuntu and it is telling us it is forwarding all requests to soccer.hdb son

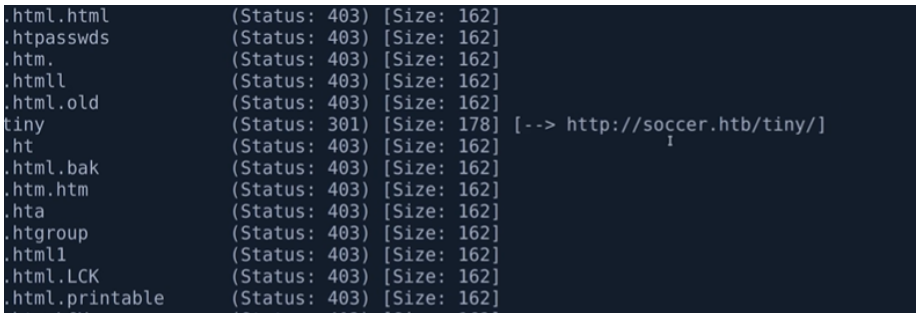


2.we could try hitting it so if we went to 10.10.11.194 was it we just get that cannot get page we see an error page from which going back takes us to the website.I used Burp suit but it didn't show anything.

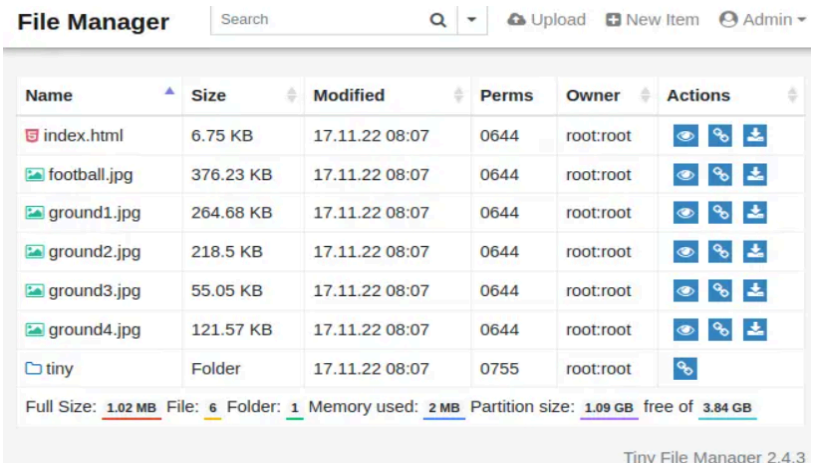




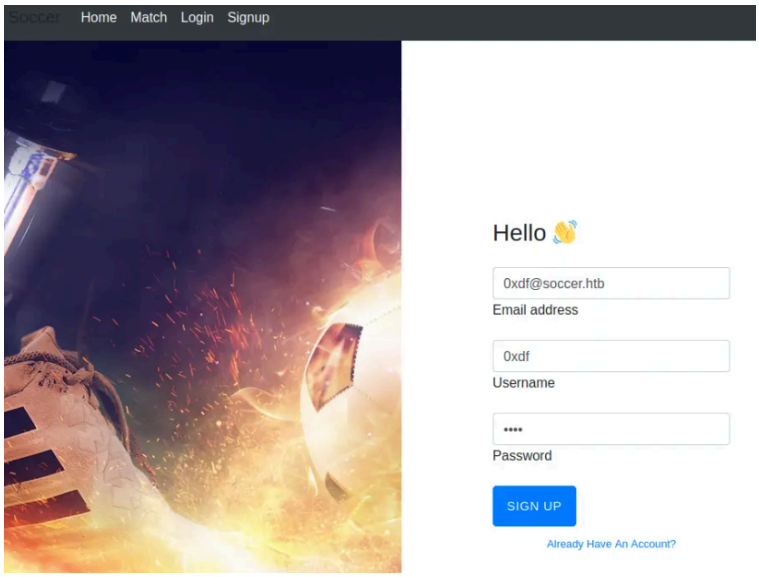
3.We are Running Gobuster and discovering Tiny File Manager and using the link, we find the login form

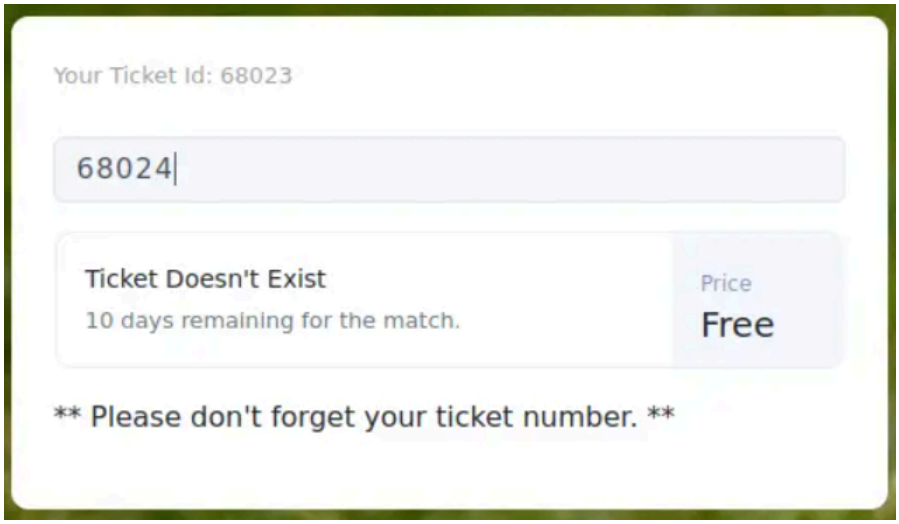


4.Looking for the source code and finding a default password of admin we can try login and then this password and we get logged in so since we are at a file manager

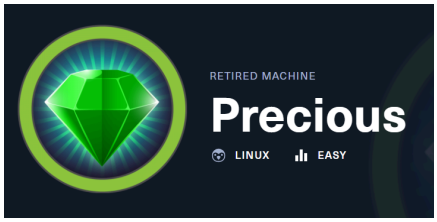


5.Navigating to soc-player.soccer.htb and discovering a few more pages and we will trying to sign up





6.SQL injection is a technique that allows an attacker to manipulate a web application's database by injecting malicious SQL queries into input fields. In this case, I exploited a Boolean-based blind SQL injection via a WebSocket endpoint by inserting crafted payloads into the `username` parameter. Using SQLMap with custom JSON data.



#Precious/easy

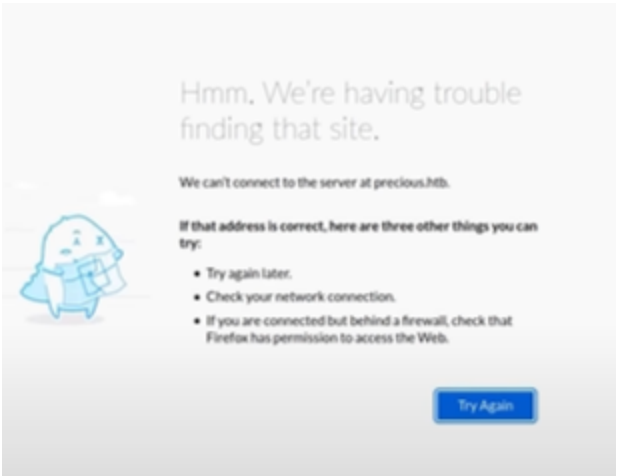
1.To begin the enumeration phase, I used **Nmap** to perform an initial scan of the target machine.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV -oA nmap/precious 10.10.11.189
[sudo] password for kali:
```

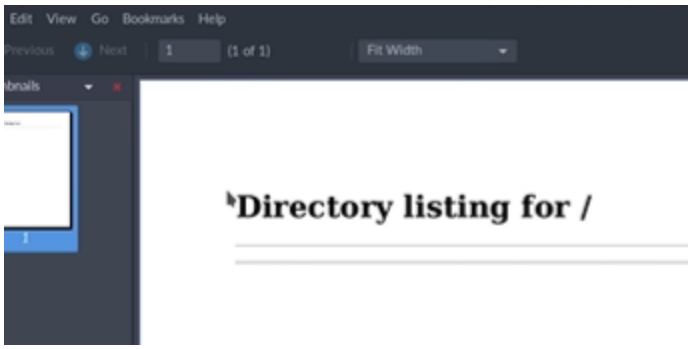
2.looking at the results we have just two ports open the first one being SSH on Port 22 and its Banner tells us it's a Debian server we also have nginx on Port

```
Nmap scan report for 10.10.11.189
Host is up (0.095s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 84:5e:13:a8:e3:1e:20:66:1d:23:55:50:f6:30:47:d2 (RSA)
|   256 a2:ef:7b:96:65:ce:41:61:c4:67:ee:4e:96:c7:c8:92 (ECDSA)
|_  256 33:05:3d:cd:7a:b7:98:45:82:39:e7:ae:3c:91:a6:58 (ED25519)
80/tcp open  http    nginx 1.18.0
|_http-title: Did not follow redirect to http://precious.htb/
|_http-server-header: nginx/1.18.0
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

3.we could go to 10.10.11.189 address we see it changes to precious.htb so then we can do Sudo Vi /etc at the host and then 10.10.11.189 put in the hostname refresh the page and we can see convert web page.

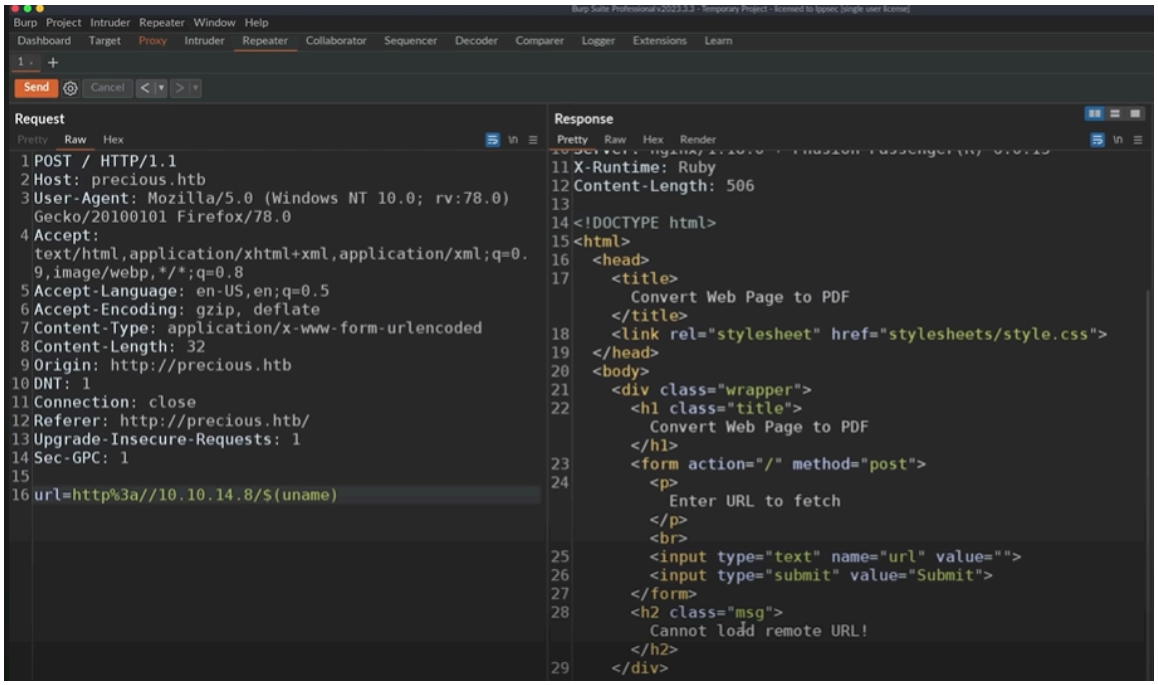


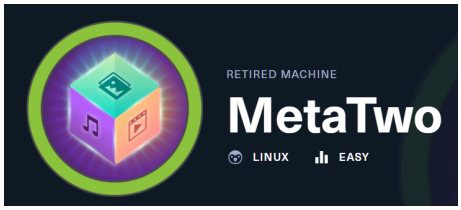
4.Checking out the web page and finding command injection in the URL so we could provide our own server so 10.10.14.8 is my IP address ,we going to use BurpSuite and ReverseShell tools in this section In the end of using we find PDF file



#Directory listing for/PDF file

5.We are going to use BurpSuite for finding something interestiing we can see PDF file generated by pdfkit, so now try reverse shell so, Finding a different payload

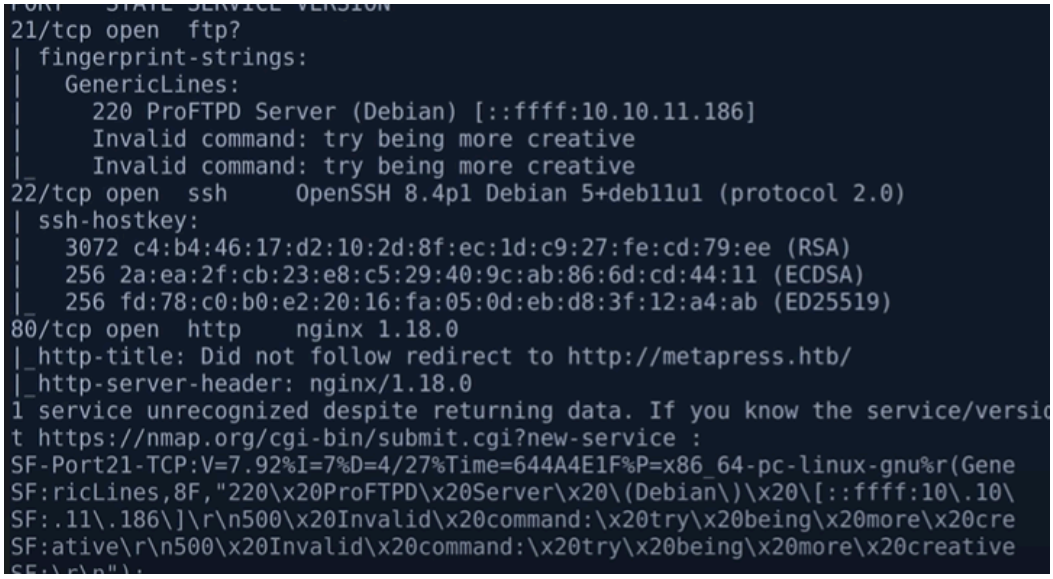




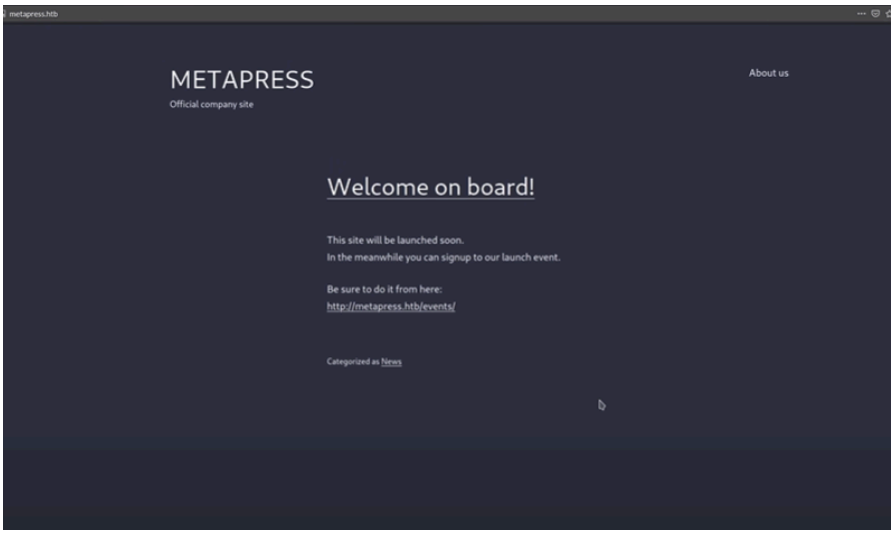
#MetaTwo/easy

1.At the beginning of the task I used **Nmap** to perform an initial scan of the target machine

2.At the results we have three ports open the first one being FTP on Port 21 and it's running a pro ftp server um this one is relatively unique we haven't seen FTP .And we can find metapress.htb and let's open up

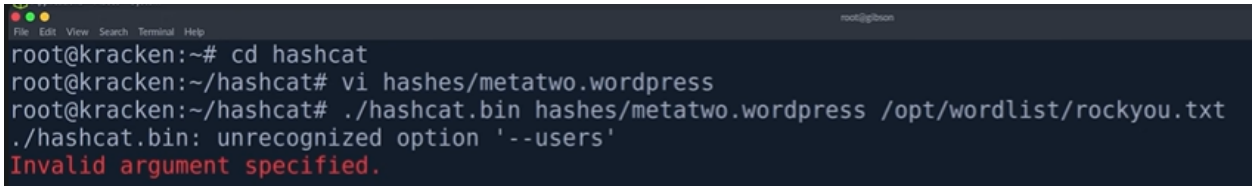


3.We are open up and see the website and i looking for their web code this WordPress so if I go back to the page this is just the HTML source of the events plugin



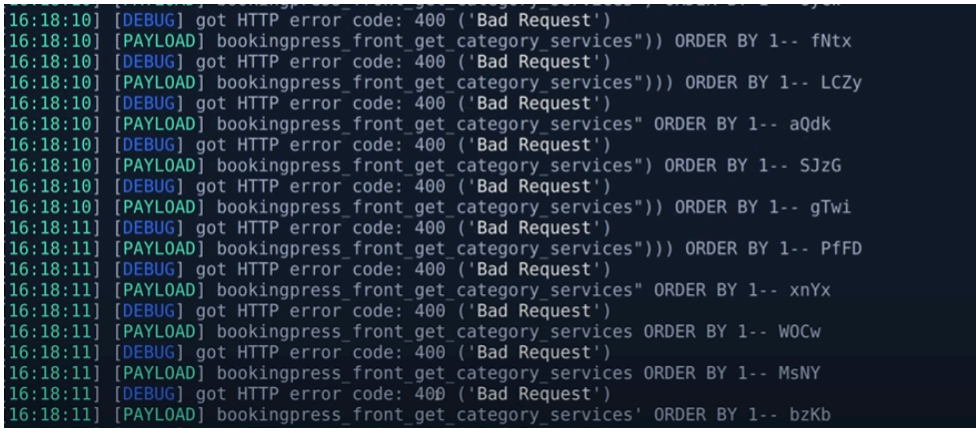
#source code

4.Using SQLMap to dump everything, while we attempt.Manually dumping the WP_USERS table with the SQL Injection



try to krack them, kracking the wordpress hashes to get a user credential

5.Playing with SQLMap to get it to dump this database



6.Searching for Wordpress 5.6.2 exploits, discovering an XXE in WAV Files

7.Using the XXE to exfil files off the webserver

8.Logging in as JNelson and seeing PassPie, which is a CLI Password Manager that uses PGP/GPG Keys

#Paper/Easy

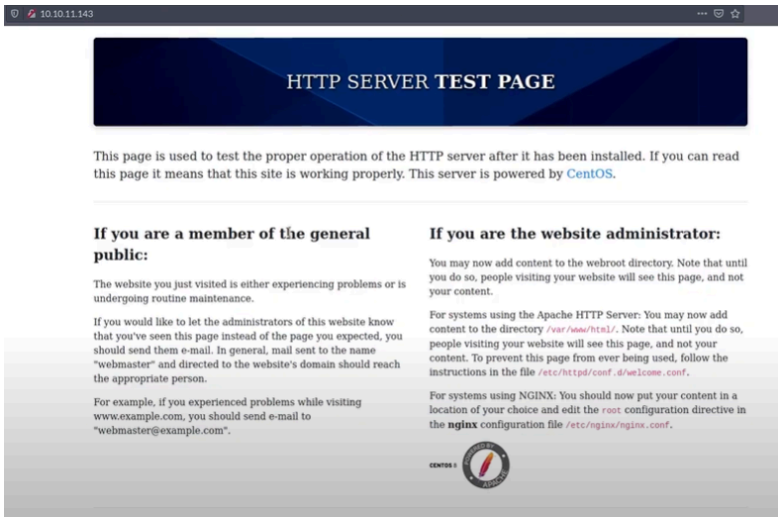
1.At the beginning of the task I used **Nmap** to perform an initial scan of the target machine

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sC -sV -oA nmap/paper 10.10.11.143
```

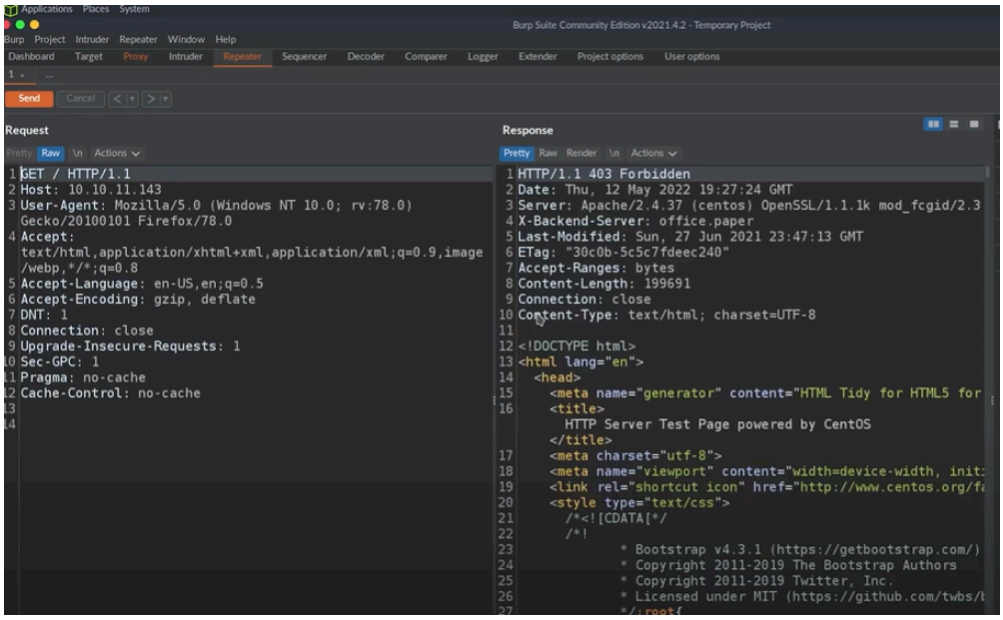
2.Looking at the results we have just three ports open the first being ssh on port 22 and http on port 80. we have https on port 443

```
PORT    STATE SERVICE  VERSION
22/tcp  open  ssh      OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   2048 10:05:ea:50:56:a6:00:cb:1c:9c:93:df:5f:83:e0:64 (RSA)
|   256 58:8c:82:1c:c6:63:2a:83:87:5c:2f:2b:4f:4d:c3:79 (ECDSA)
|_  256 31:78:af:d1:3b:c4:2e:9d:60:4e:eb:5d:03:ec:a0:22 (ED25519)
80/tcp  open  http     Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_http-title: HTTP Server Test Page powered by CentOS
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
|_http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
| http-methods:
|_  Potentially risky methods: TRACE
443/tcp open  ssl/http Apache httpd 2.4.37 ((centos) OpenSSL/1.1.1k mod_fcgid/2.3.9)
|_http-generator: HTML Tidy for HTML5 for Linux version 5.7.28
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.37 (centos) OpenSSL/1.1.1k mod_fcgid/2.3.9
```

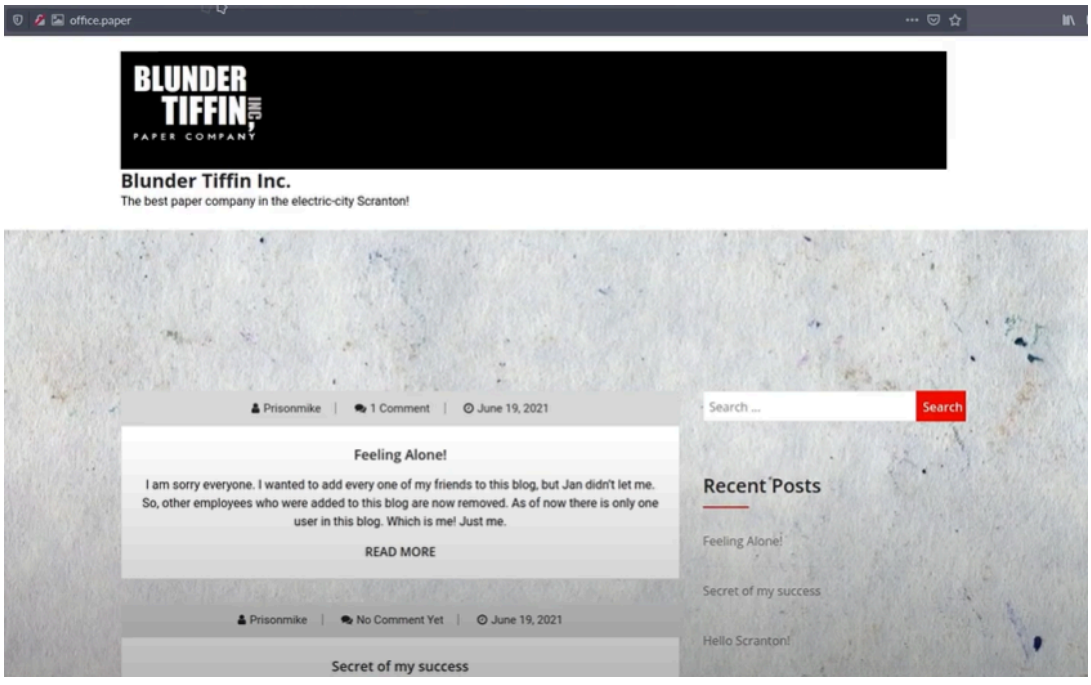
3.And we are going to looking 10.10.11.143 address and we get an http server test page



4.Running Feroxbuster and GoBuster and we're getting a lot of errors and i decided use BurpSuite to find something interesting there i noticing a X-Backend-SErver header that leaks the virtual host Office.Paper We decided nmap 10.10.11.143 80 port and there are some web page scripts

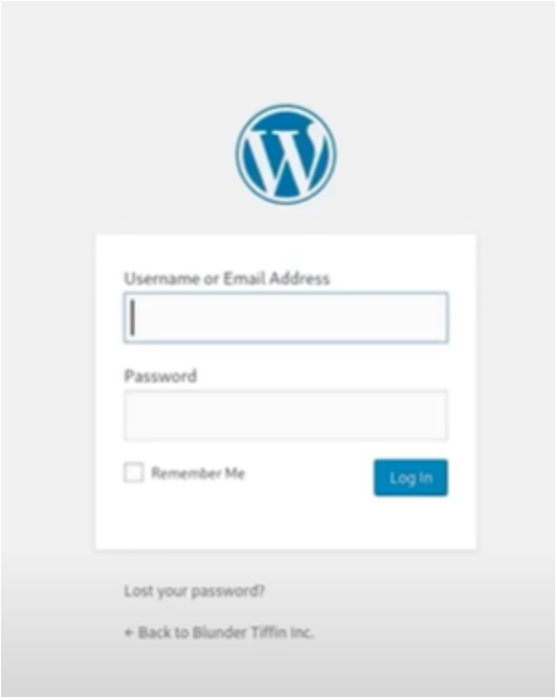


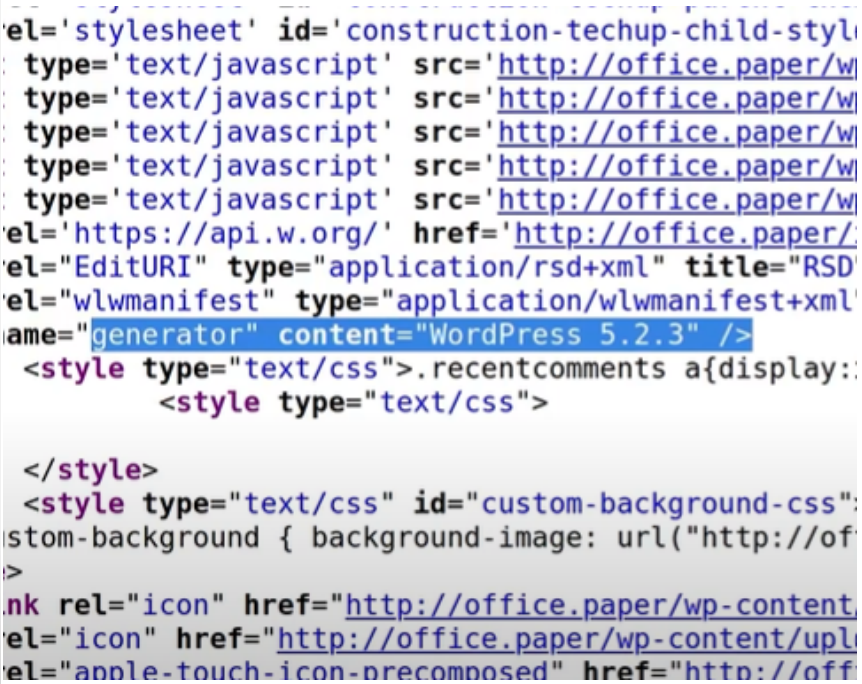
5. i find source of Office.paper and there is wordpress webpage



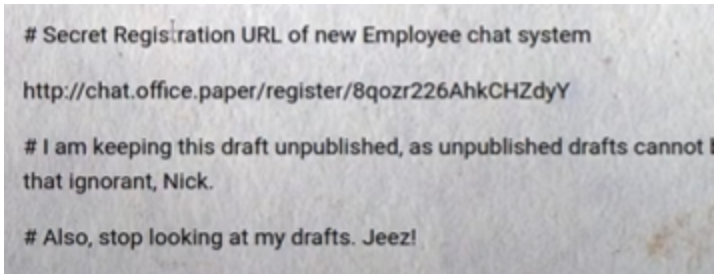
And if it wordpress web page we can looking for admin page office.paper/wp-admin

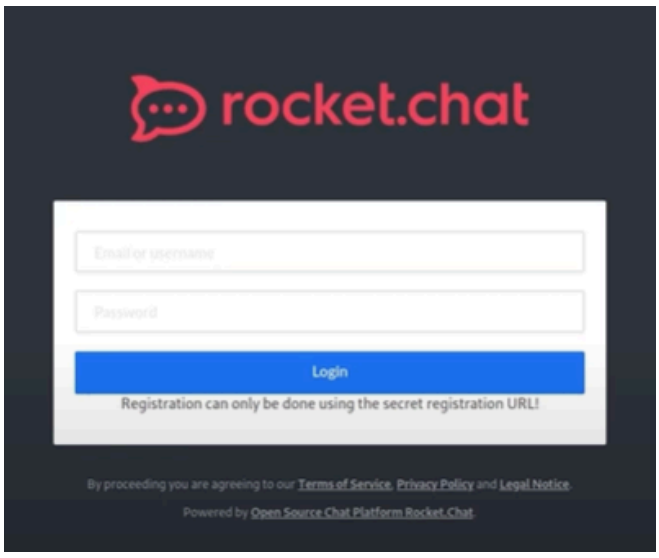
6.There are i find admin login page i decided looking for office.paper web page source code to find login or password and there are i find wordpress version is (5.2.3) i googling about it and find that it insecure version and i decided to scanning this login page





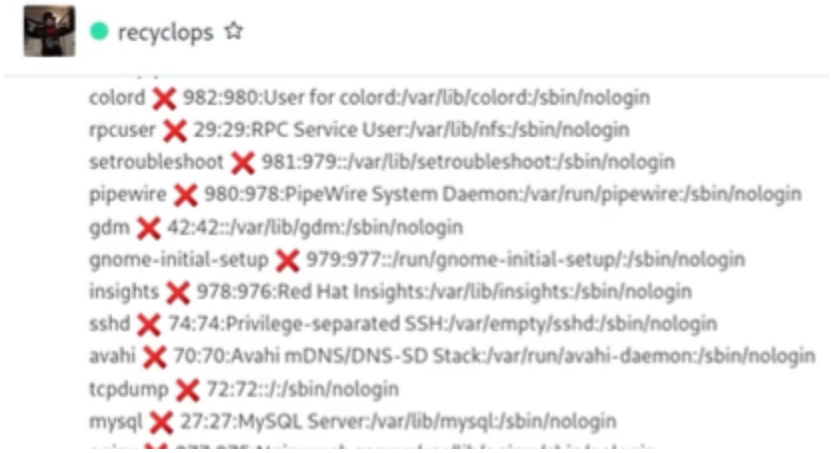
7.In Footer of web page i find some link chat.office.paper/register and going to link i find some login page to rocket.chat and i register fake user with this secret registration page and login some rocket chat server

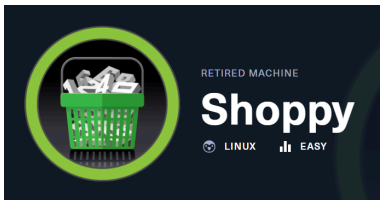




There are i find some general chat and there are some scripts and filename information and i send file name to user Direct and i have automatic response hint and i try this Command injection

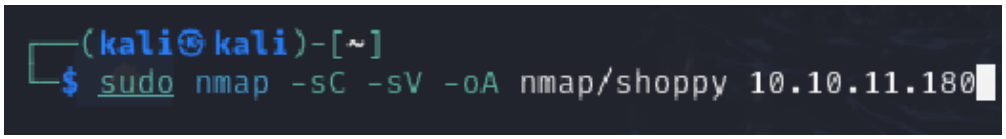
8.I use this command injection on this chat and receive some passwords content





#Shoppy/easy

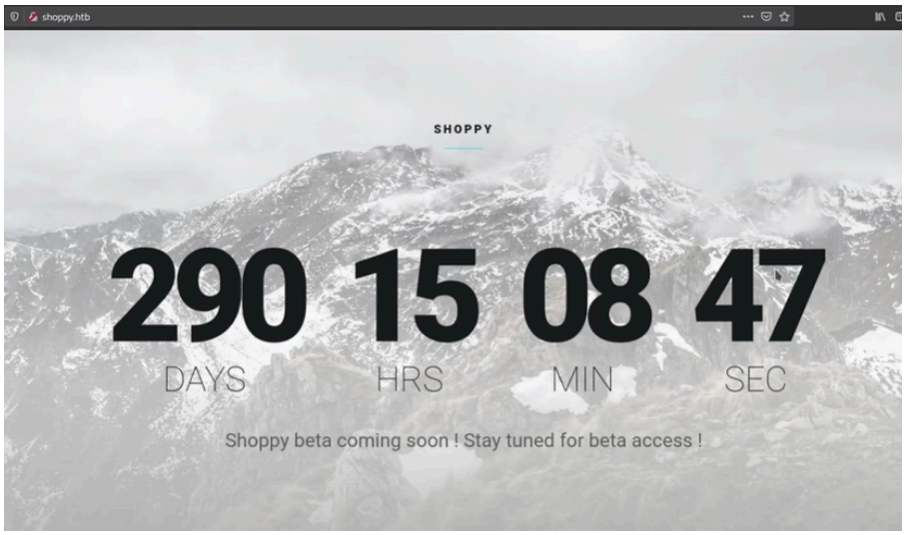
1.At the beginning of the task I used **Nmap** to perform an initial scan of the target machine



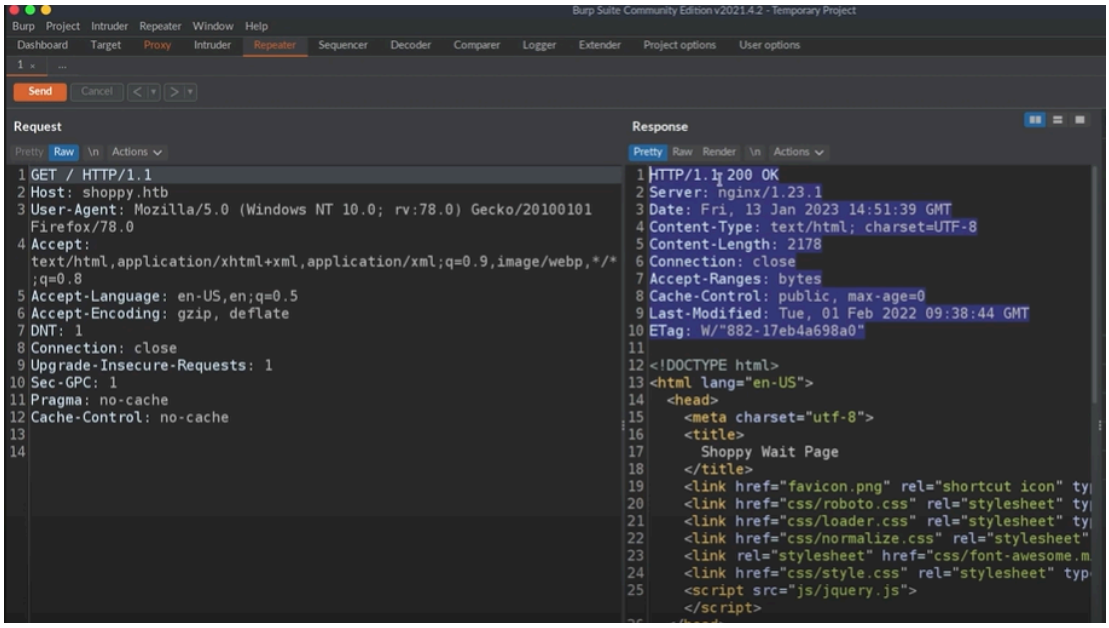
2.Looking at the results we can see two ports open we can see SSH on 22/tcp and HTTP on 80/tcp

3.Switching tactics, I tried brute-forcing directories with **ffuf**, followed by an attempt to brute-force subdomains. Unfortunately, both approaches led to dead ends.With no significant progress, I turned to UDP scanning:

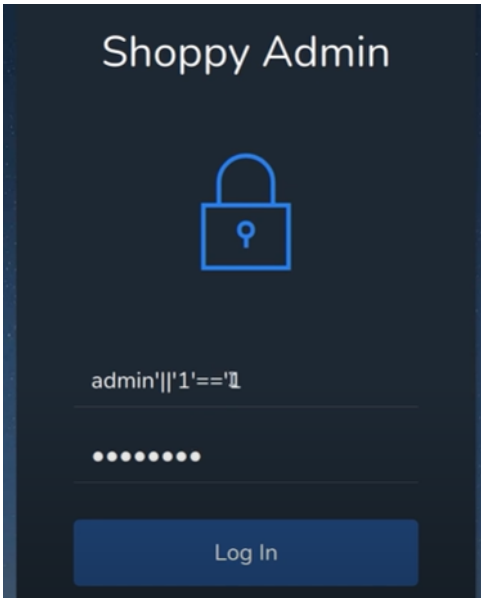
3 And i looking for open 10.10.11.180 to http://shoppy .htb and there are some web page and first thing i did i am checking html source of this web page and just finding that page written by javascript

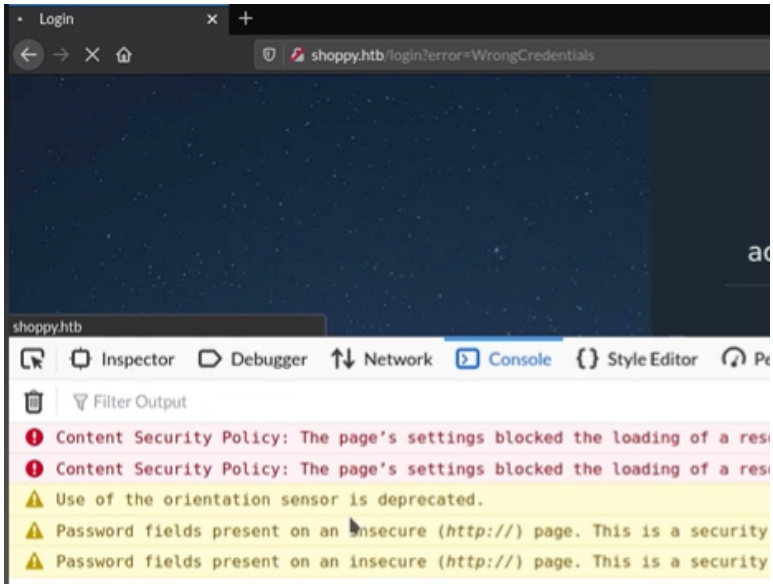


4.I decided use BurpSuite for finding something interesting and we are finding that page written by framework nodejs



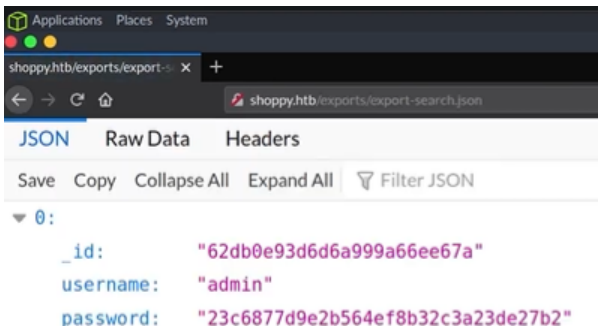
5.And i decided to check some basic things shoppy.htb/admin page there are some admin login page i trying some basic pass login but i see just error and decided to BurpSuite this side login error and i find it's just PHP script And i trying to find sql injections

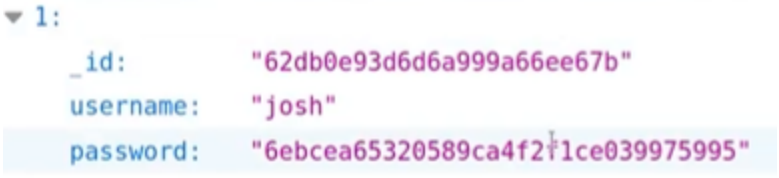




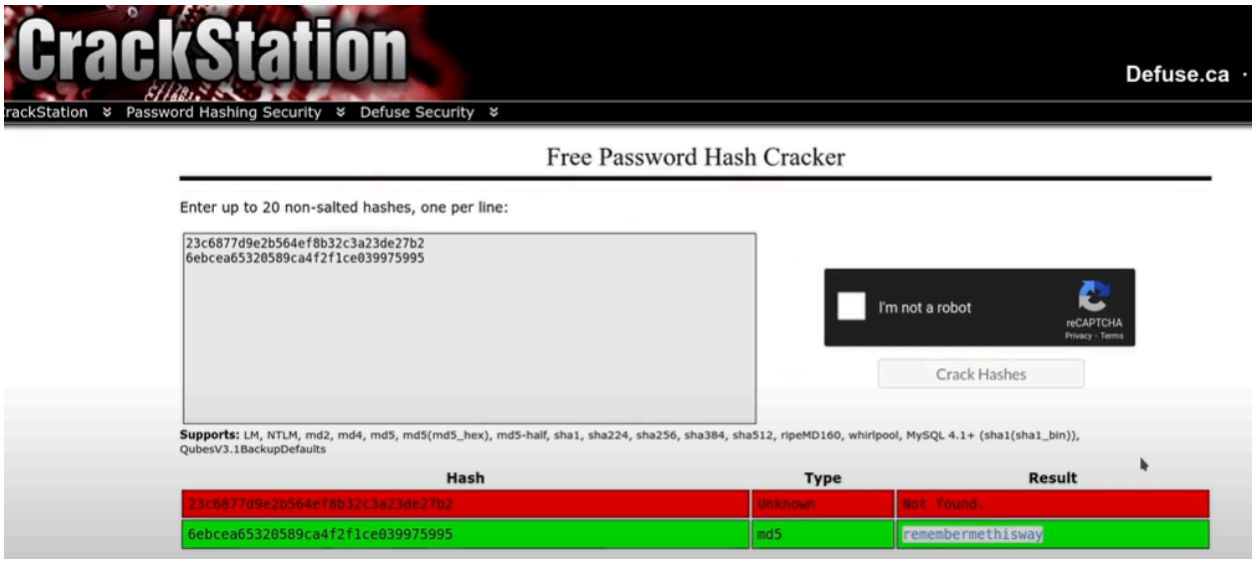
And we are finding admin page and there I trying to find something interesting in “Search For Users” and i jus searching user “Admin” and find some file where

Where we can find json file where some information about user Id ,password,login and trying to find other users information in this side

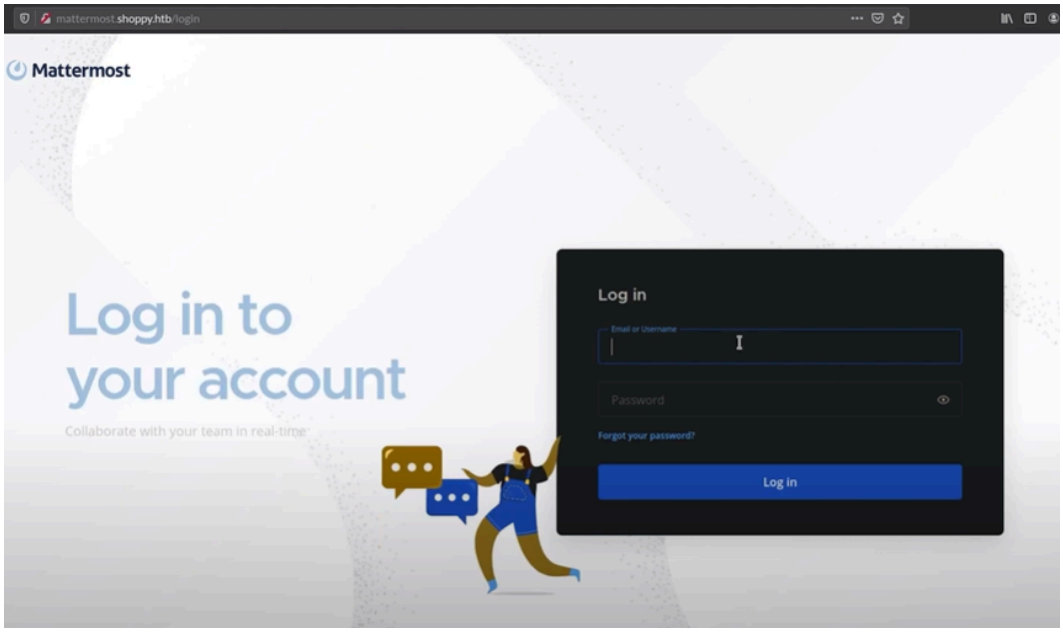




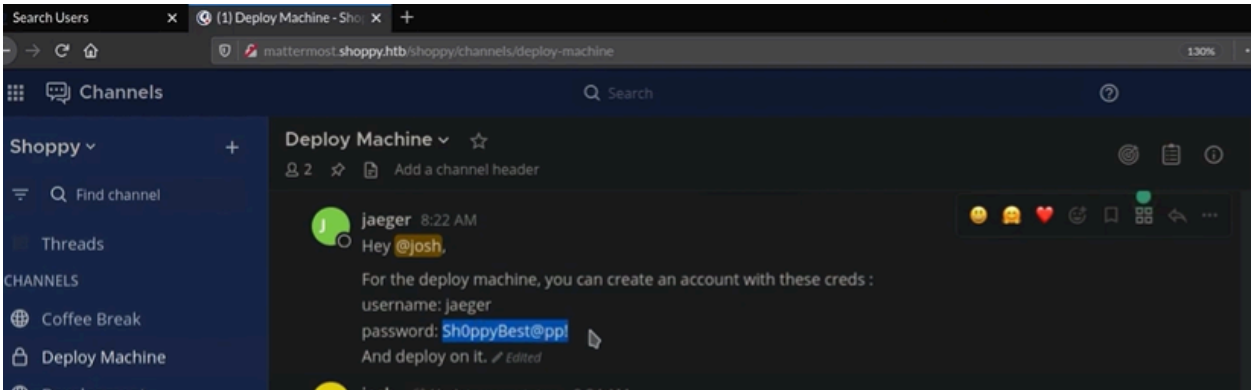
6.And i trying to Crack the passwords and use password hash cracker

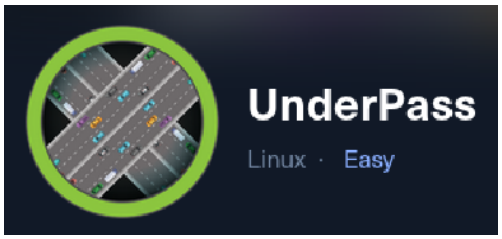


7.I am use ffuf and find new subdomain matthermost.shoppy.htb and i trying to login with user which i find, i am login and i find there a lot of information



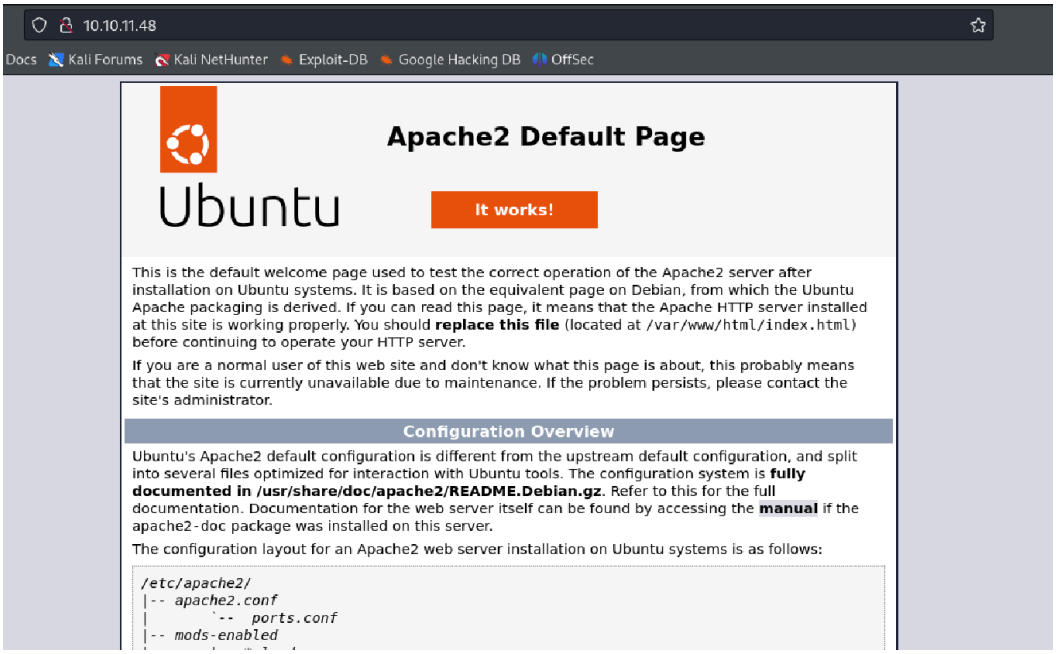
There are some secret private channels,chats of this user





#UnderPass/easy

1.First i looking for webpage and find Apache Default Page then i decided nmap ip address and find something interesting



I find Open Ports 22/tcp SSH and 80/tcp HTTP and we can find underpass.htb domain Apache web page we can see some hints for nmap-ing

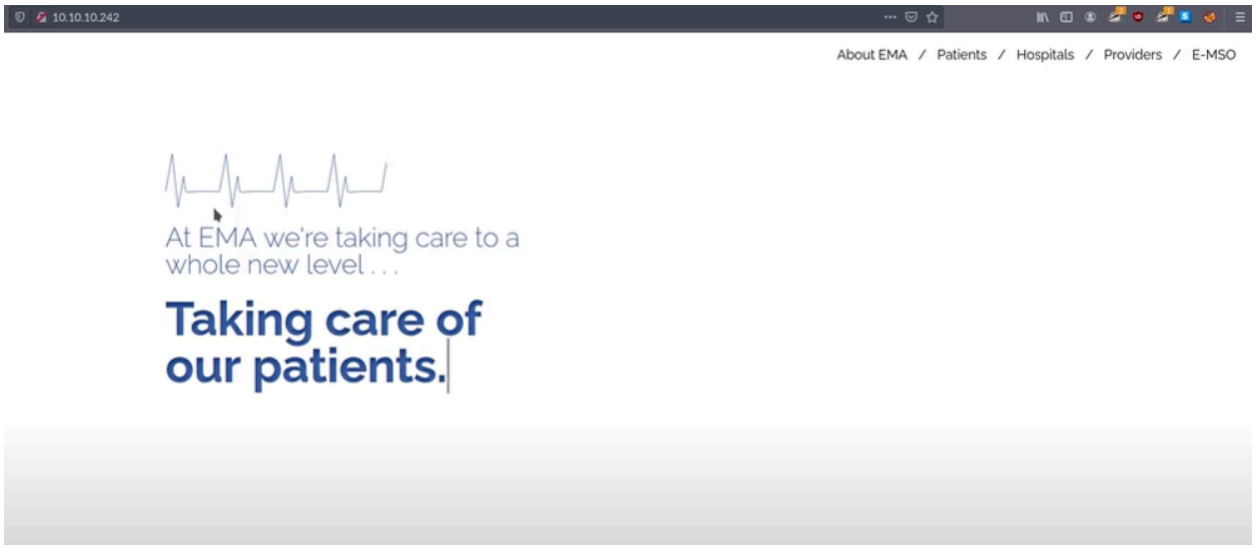
Given the use of domain based routing (or virtual hosts), I'll use `ffuf` to scan for any subdomains of that respond differently from the default case and find

#knife/easy

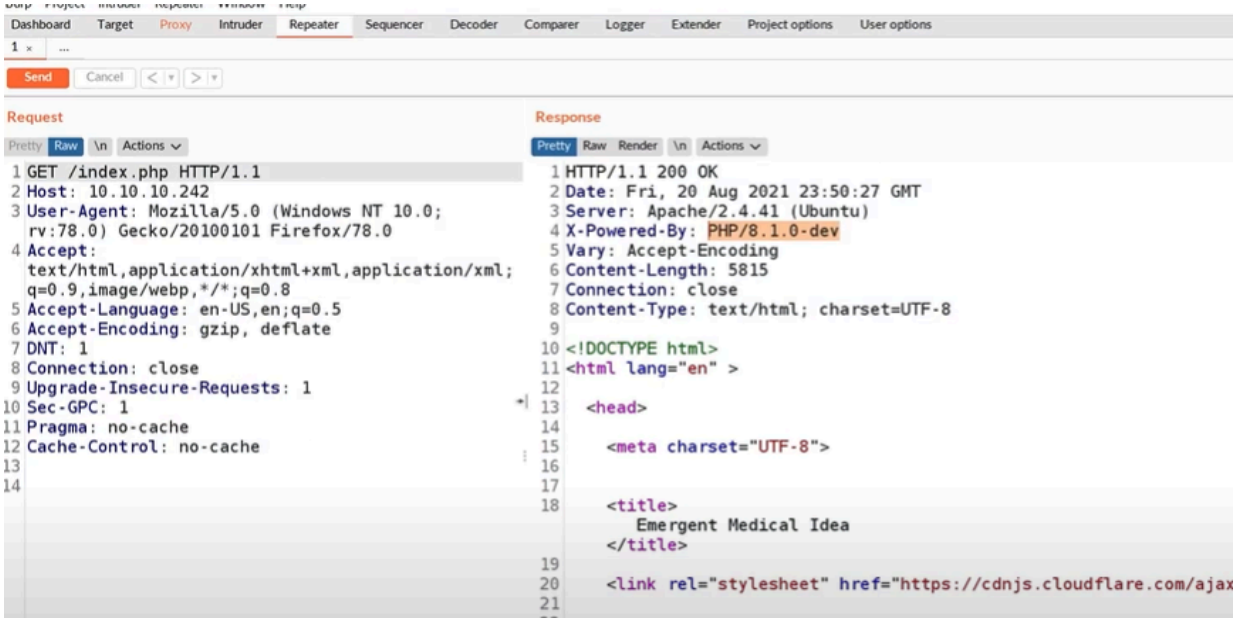
1.We are just *nmap* ip address 10.10.10.242 and find 2ports open 22/tcp SSH And 80/tcp HTTP

```
Scanning 10.10.10.242 [65535 ports]
Discovered open port 80/tcp on 10.10.10.242
Discovered open port 22/tcp on 10.10.10.242
Completed SYN Stealth Scan at 00:42, 6.96s elapsed (65535 total ports)
Nmap scan report for 10.10.10.242
Host is up (0.10s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http
```

2.Looking for web page and find and its look like some hospital website i see the source code of website



And i decided to use *Gobuster dir* to http://10.10.10.242 and look up to nmap scan and i finding 22/tcp SSH open port on Ubuntu server and i also decide look to cookies and use BurpSuite to find something interesting



And discovering we can run knife .

This tool has an entry in GTFOBins to elevate privileges to root when run with sudo

By checking the permissions with sudo -1, it shows that the user can run */usr/bin/knife* with root privileges

.. / knife

☆ Star 11,559

Shell Sudo

This is capable of running `ruby` code.

Shell

It can be used to break out from restricted environments by spawning an interactive system shell.

```
knife exec -E 'exec "/bin/sh"'
```

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo knife exec -E 'exec "/bin/sh"'
```

🔚 Conclusion

Throughout this penetration testing project, I applied a variety of essential tools and techniques to simulate real-world attacks and identify potential vulnerabilities in the target machines. The tools I utilized were selected to cover multiple phases of the pentesting process, including reconnaissance, enumeration, exploitation, and post-exploitation analysis.

I began with **Nmap**, a powerful network scanner, to identify live hosts, open ports, and the services running on them. This allowed me to build a clear map of the target environment. I then used **Gobuster** and **FFUF** to perform content and directory enumeration, revealing hidden endpoints and sensitive resources that are not accessible through normal browsing.

For web vulnerabilities, I used **Burp Suite**, which provided a proxy-based environment to intercept, modify, and replay HTTP requests. With this tool, I was able to test for common web application flaws such as **XSS (Cross-Site Scripting)** and **SQL Injection**, successfully demonstrating how an attacker could exploit user input and gain unauthorized access or extract sensitive information.

To analyze and crack exposed credentials, I performed **password hash cracking** using dictionary-based attacks, and examined **JavaScript files** for hidden endpoints and logic flaws that could aid in further exploitation.

Each tool used during this assessment served a specific purpose:

- **Nmap**: Network reconnaissance and service detection
- **FFUF & Gobuster**: Fuzzing for directories, subdomains, and web content
- **Burp Suite**: Manual and semi-automated web vulnerability testing

- **Hash cracking tools**: Extracting plaintext from leaked hashes
- **JavaScript analysis**: Reverse engineering frontend logic
- **Manual testing for XSS & SQLi**: Demonstrating web injection attacks

This project significantly enhanced my understanding of the penetration testing process. I learned how to think like an attacker, chain multiple vulnerabilities, and approach systems from both a technical and logical perspective. Through hands-on use of these tools in real and simulated environments, I've deepened my practical skills in cybersecurity and am more confident in conducting structured, ethical penetration tests