**Security Audit (PenTesting)**

# Introduction

This report will consist of 3 different machines and how I managed to hack into them to get the flag. This will outline the vulnerabilities in these machines.

Contents

[Introduction 1](#_Toc89899833)

[Machine 1 (VM3) 1](#_Toc89899834)

[User Flag 1](#_Toc89899835)

[Recon 1](#_Toc89899836)

[Nmap scripts for recon 2](#_Toc89899837)

[Getting the flag 4](#_Toc89899838)

[Root flag 8](#_Toc89899839)

[Machine 2 (VM2) 9](#_Toc89899840)

[User flag 9](#_Toc89899841)

[Root Flag 16](#_Toc89899842)

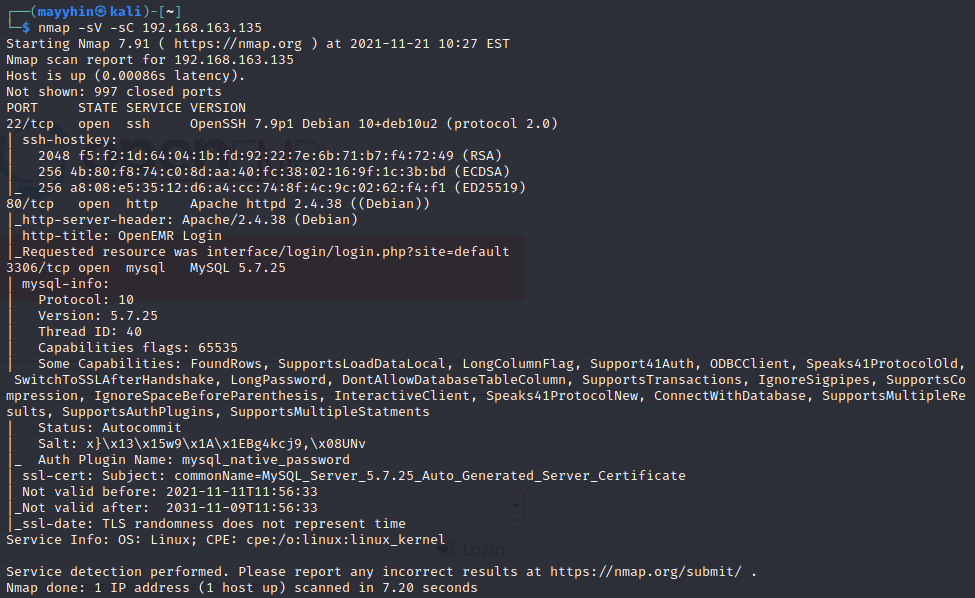
[Overflow Machine 18](#_Toc89899843)

# Machine 1 (VM3)

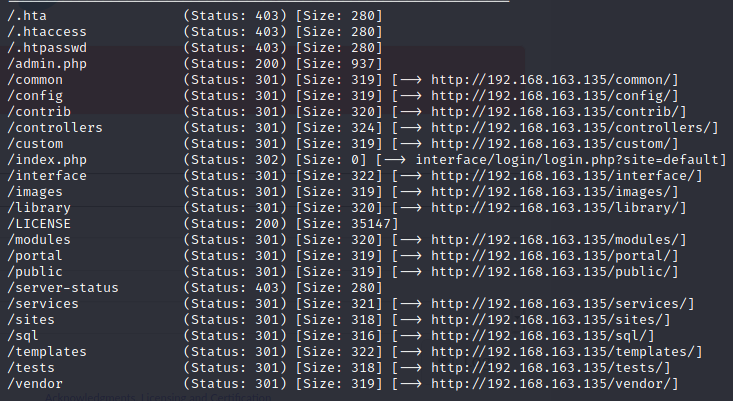
## User Flag

### Recon

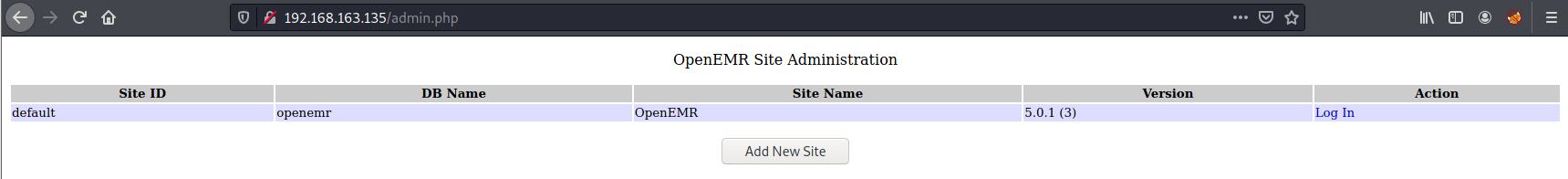
First, I used Nmap to see services and ports running on the system. This showed me that ssh, http and MySQL services were running.



Then I used Gobuster to see the hidden webpages.

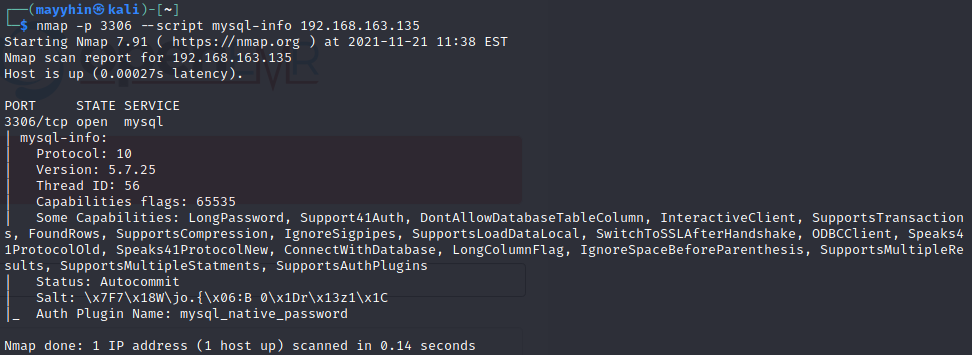


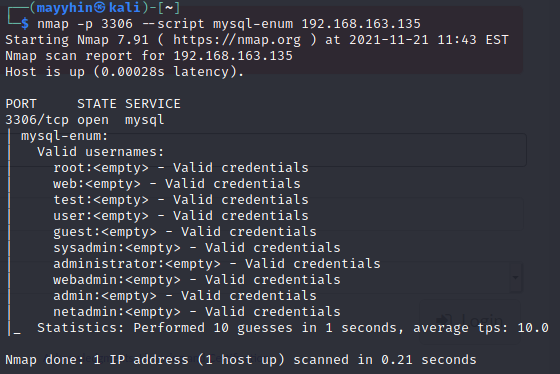
Just going to do some recon on what I can find and screenshot anything that could be useful.



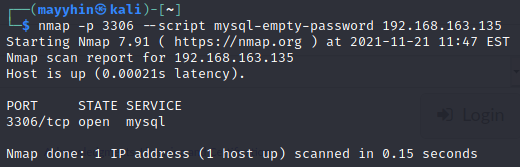
### Nmap scripts for recon

Using some nmap scripts I can find useful information about the mysql service running on the machine. Information and scrips used will be shown in the screenshots below.

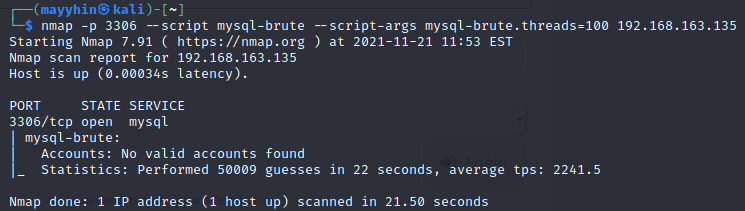




All users I can try to log into.

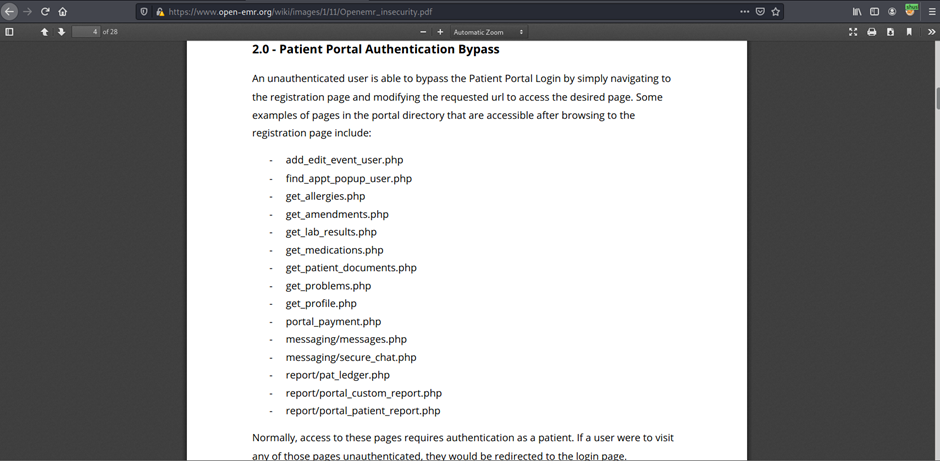


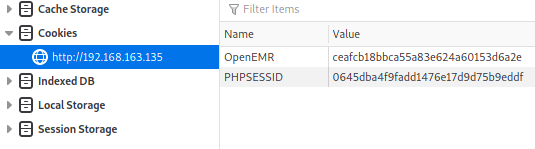
No empty passwords.



Not able to brute force my way into the database.

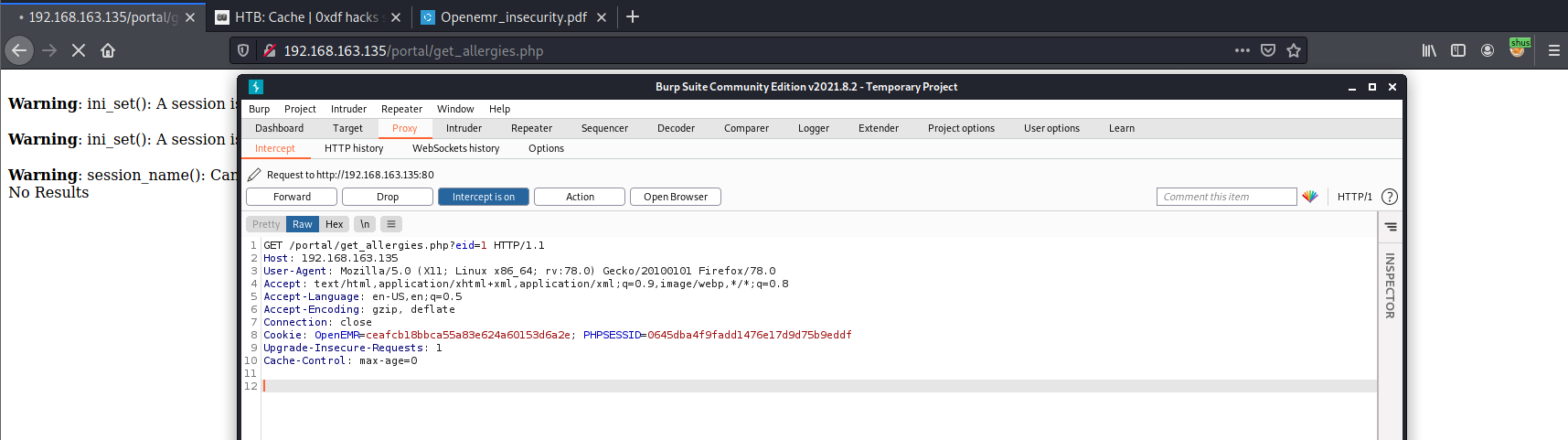
### Getting the flag



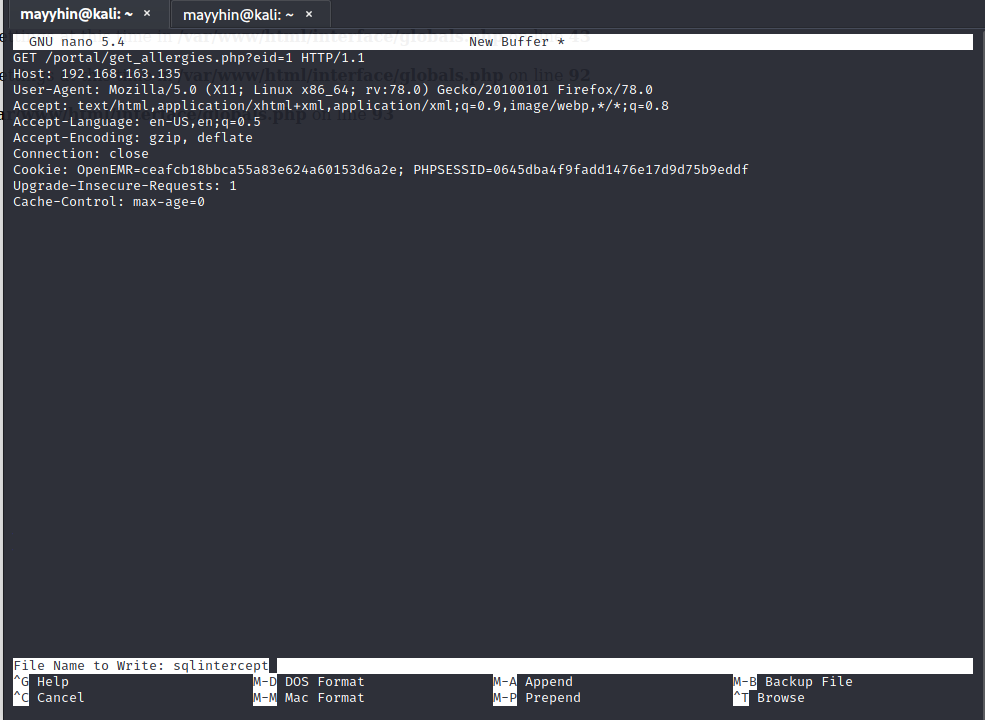


Found some authentication bypass from the pdf above that I screenshotted. Changed url to “portal/account/register.php” to give me a new cookie. Following a similar HTB I found sqlmap commands to use to also bypass authentication.

<https://0xdf.gitlab.io/2020/10/10/htb-cache.html>

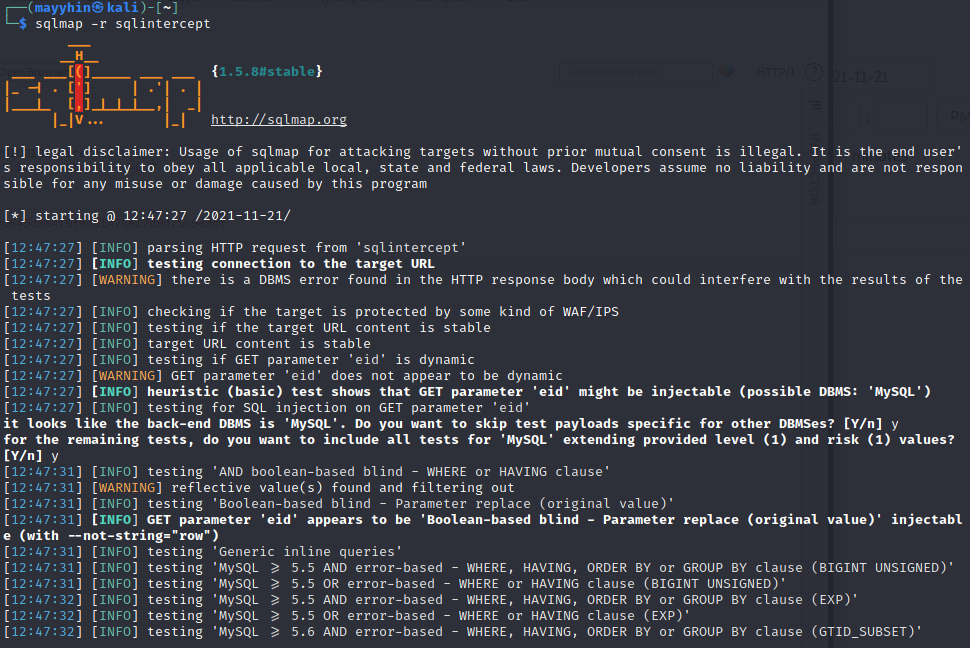


I used burpsuite to intercept the webpage so I could copy and paste everything shown on proxy into a file, named sqlintercept, to use sqlmap on.



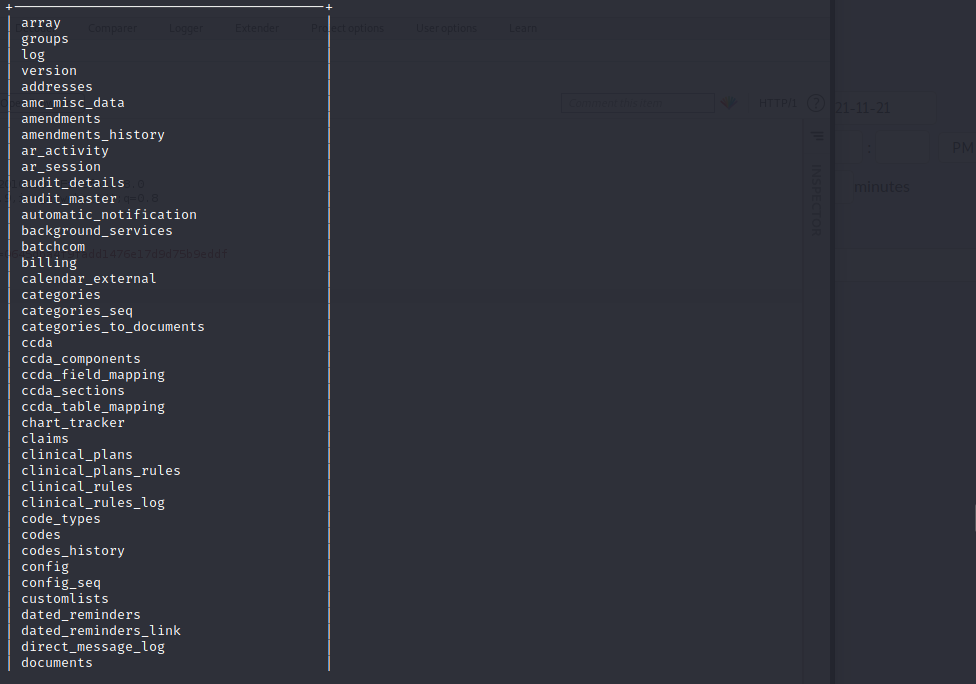
Intercept did not work with this directory, so I’ll try another directory until it works. With the same way I showed above using the command “sqlmap -r MYFILENAME”

It worked with the register.php directory.

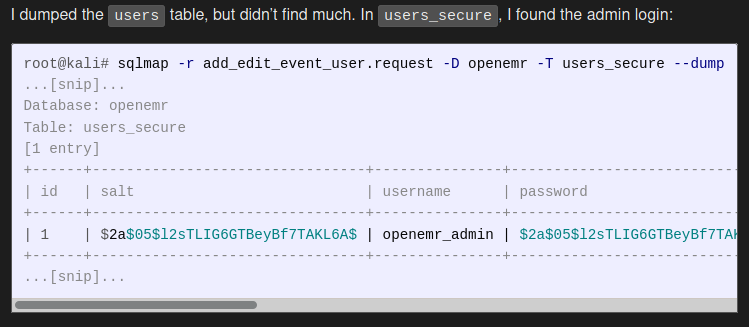




This command was used to show me the tables running on the MySQL server on the target machine.

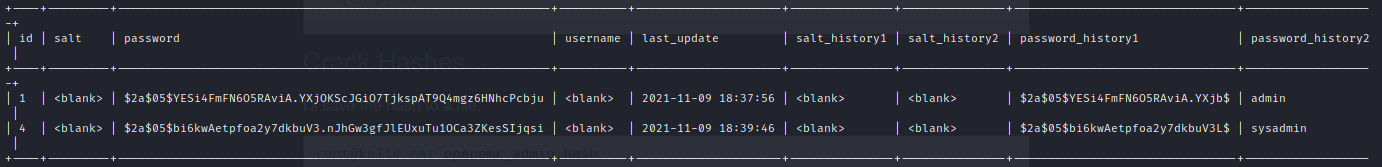


Got a list of tables.

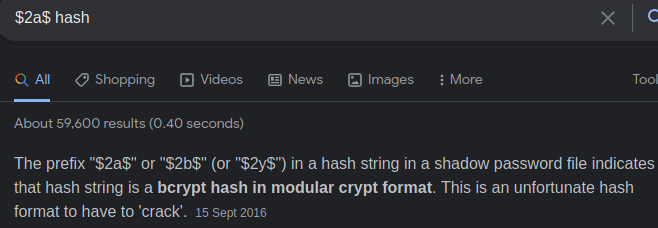


Following this command from the link provided earlier from a HTB in order to output the table with users and hashed passwords.



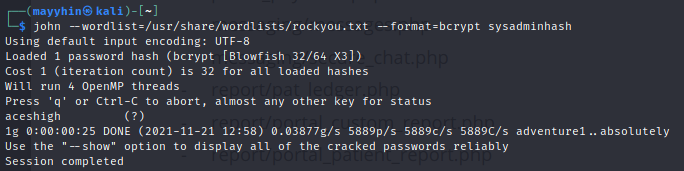


Time to use hashcat, john the ripper, etc to find the password from the hashes.

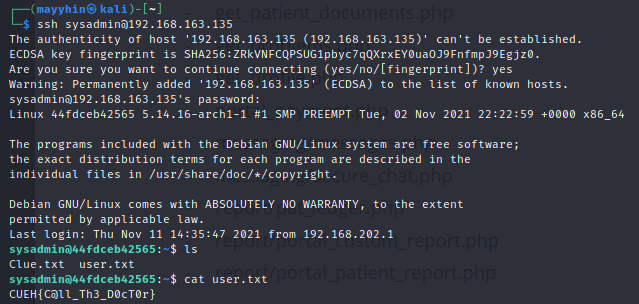


It’s a bcrypt hash.

I used hashcat and it wasn’t working or finding anything, but john the ripper found some passwords.



Password aces high was decrypted.



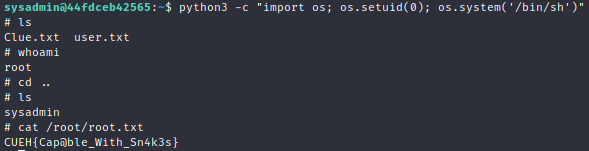
**Logged in with user sysadmin and password aceshigh using ssh and found the flag.**

## Root flag

I used the command “getcap -r / 2>/dev/null” to see all files with capabilities on the system.



From <https://github.coventry.ac.uk/pages/aa9863/5063CEM/5_Privesc/Capability_Walkthrough/> found a python command I can use to get root access since python was the only capability shown.

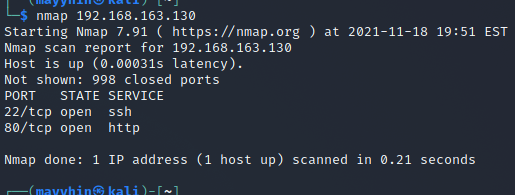


**Got the Flag.**

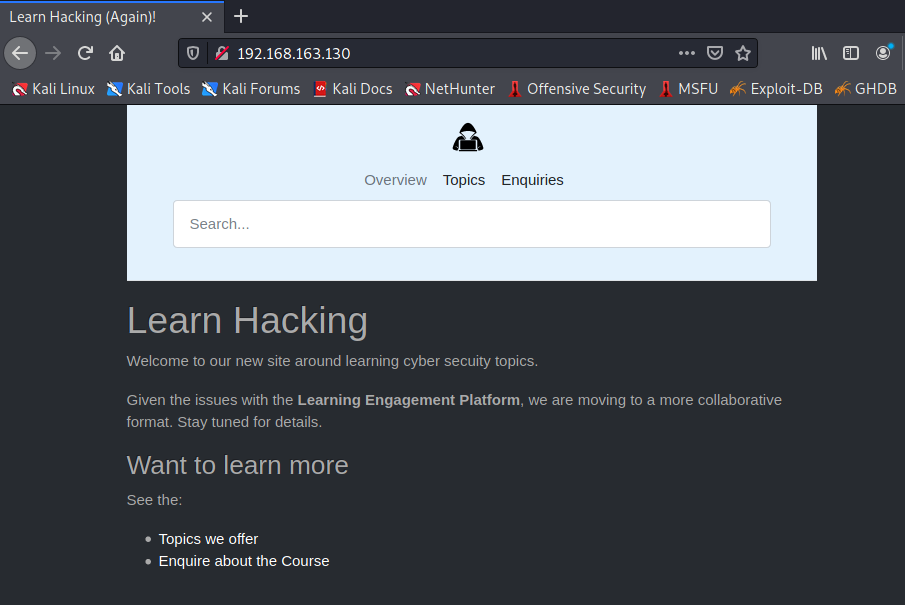
# Machine 2 (VM2)

## User flag

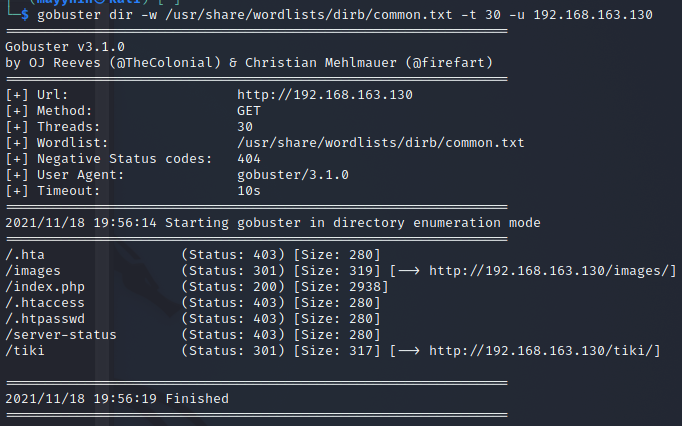
First I used Nmap to scan the ports running on the ip.

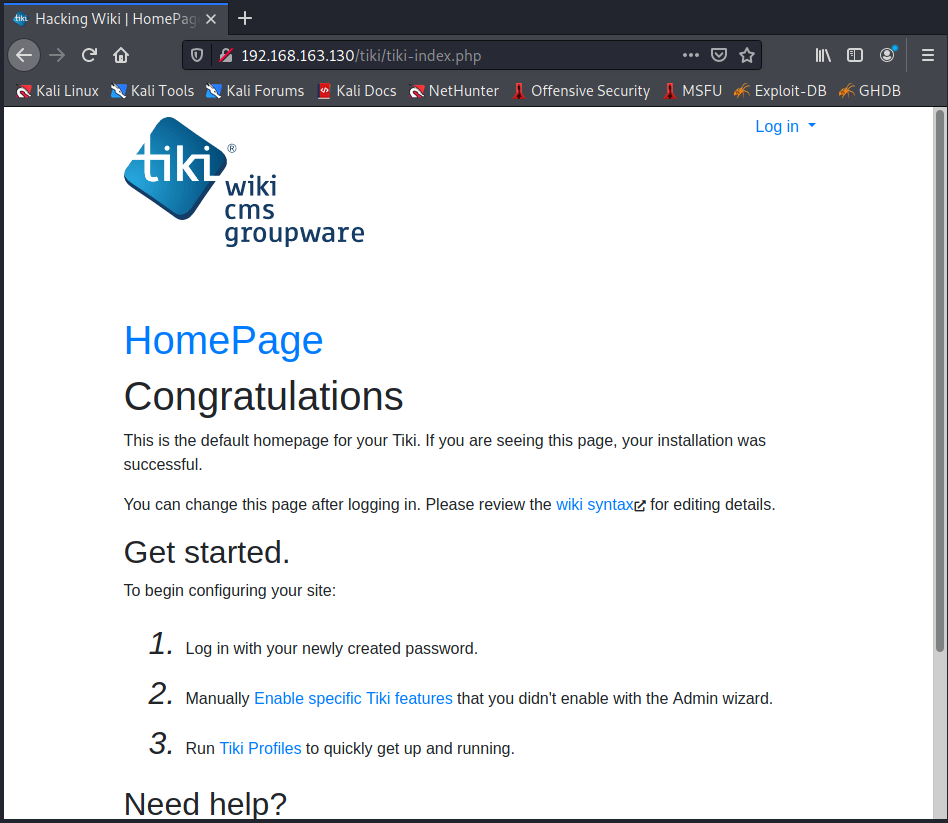


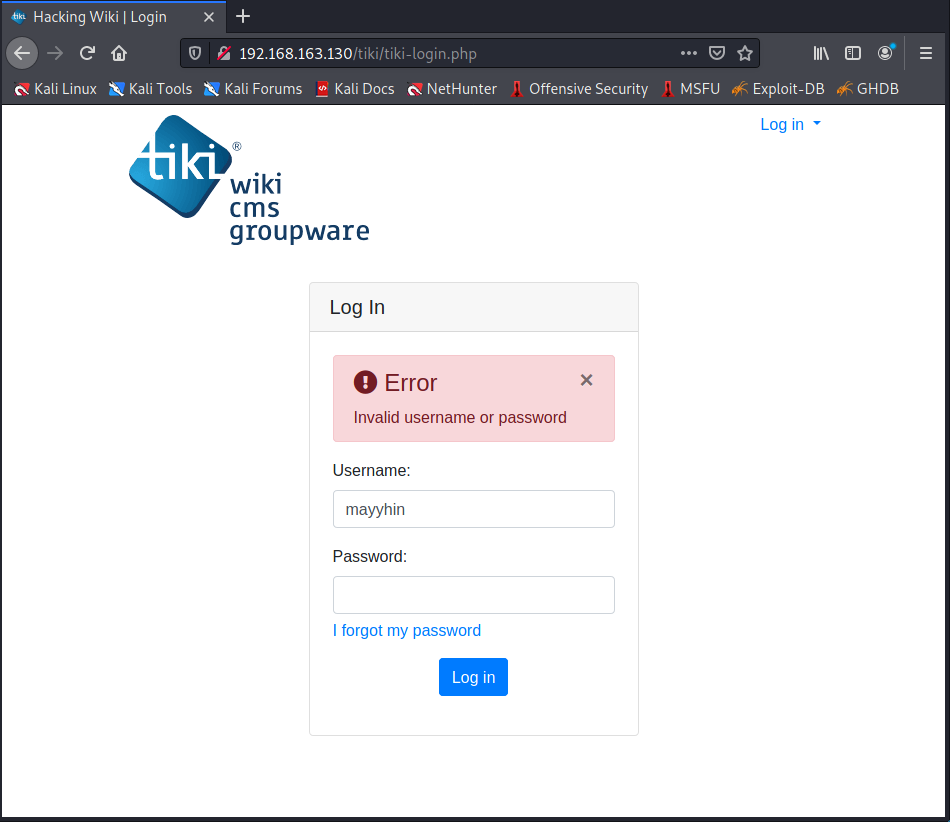
http and ssh service running meaning there should be a web page.



Then I used Gobuster to find any hidden webpages. Found some that may be useful.



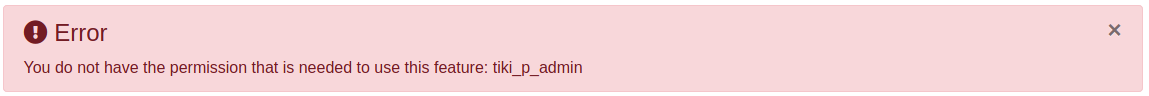




Tried putting in some credentials. Didn’t work.

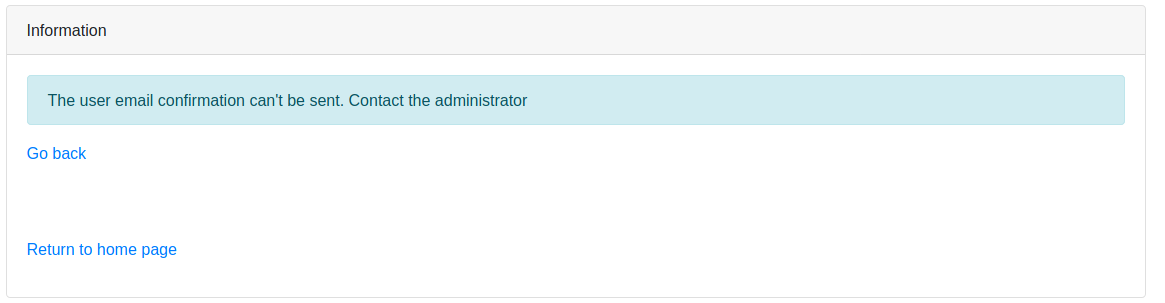
After some googling I found that this website should have a default admin user with admin as the password, but this also shows up with the same error but at least I know an admin user should be on the system.

On another directory of the tiki website it does not give me access to two of the pages and gives me an error.

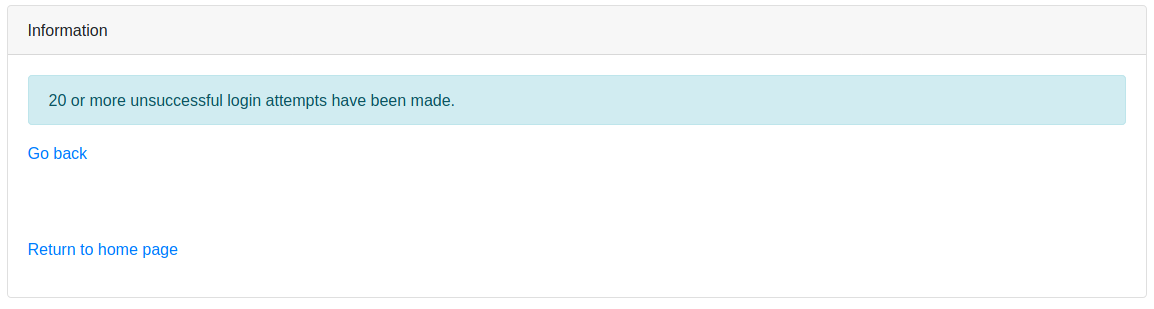


Found an exploit for tiki 21.1 so I’m going to try that.

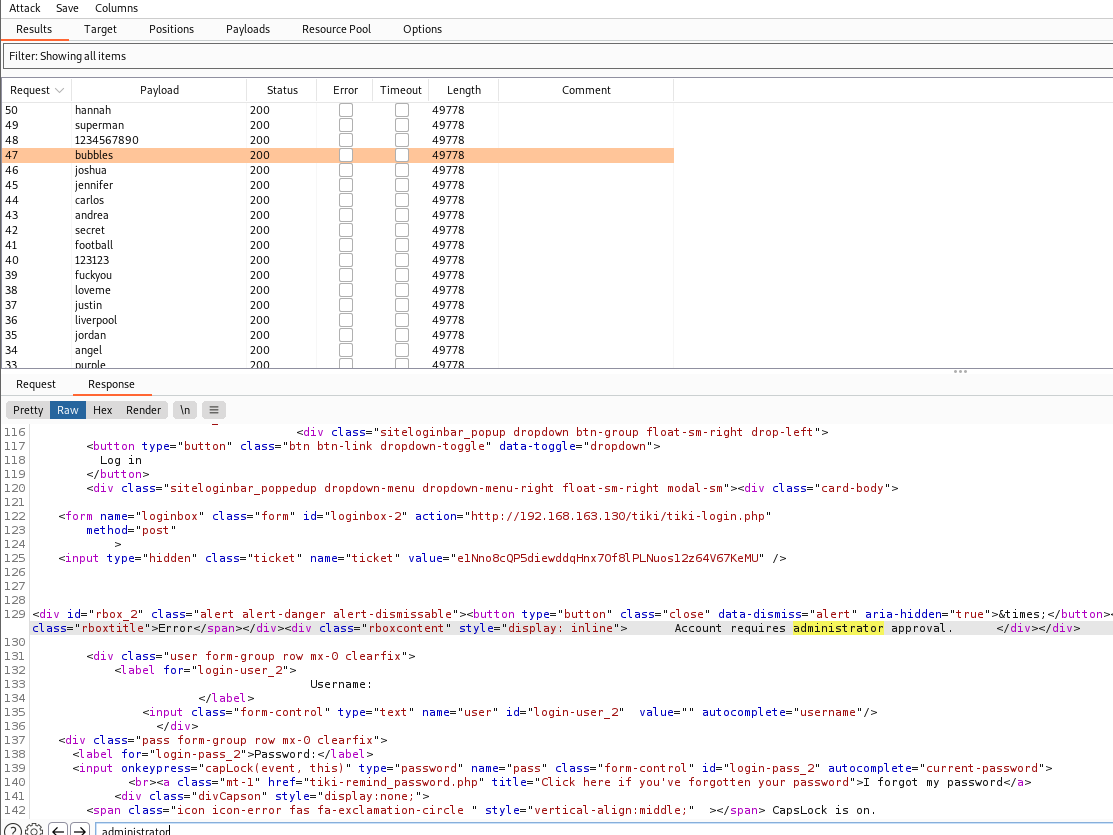
First, it says to keep brute forcing the site. After doing it about 15 times I got this.



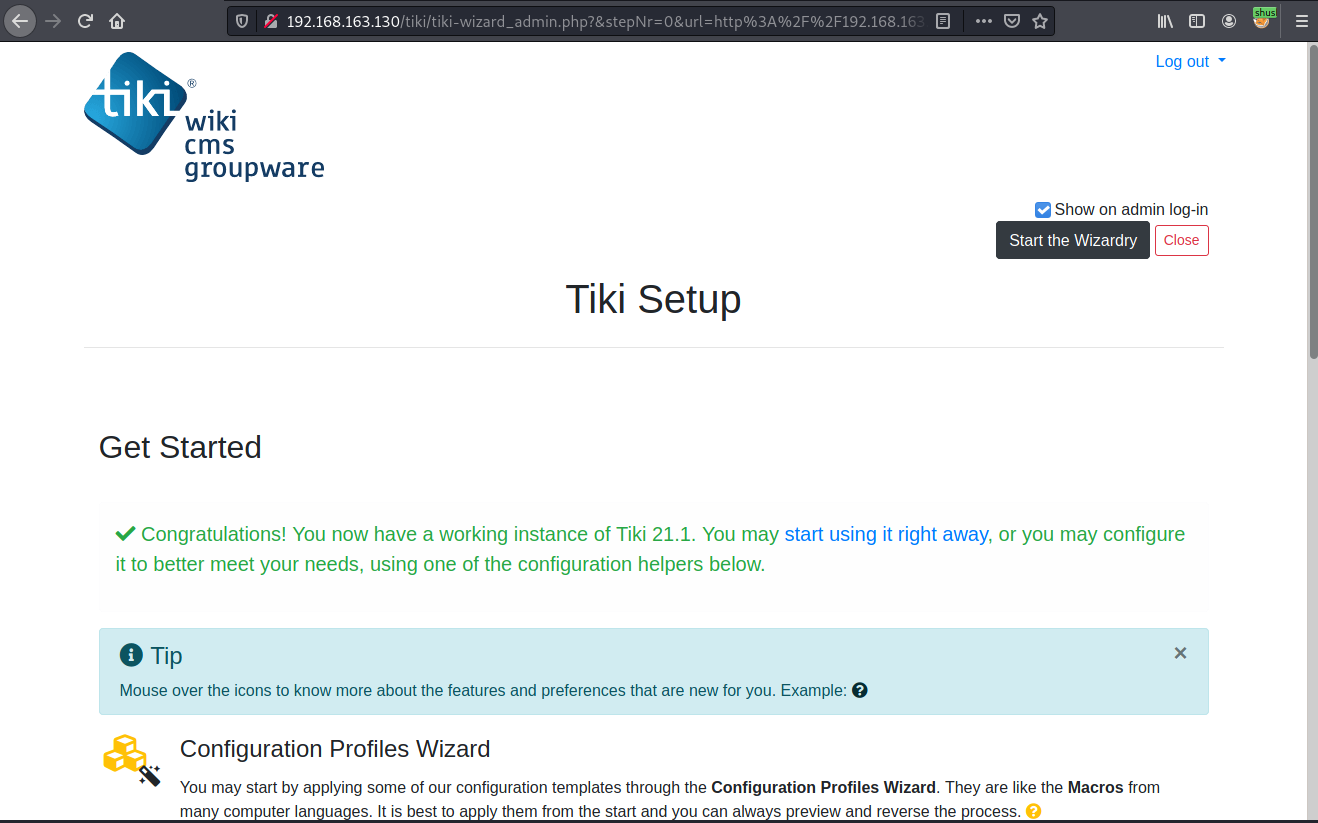
I Keep doing this.



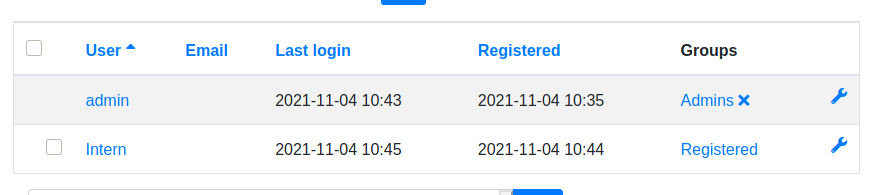
Used burpsuite to automatically keep trying passwords with its built in function to send payloads. Until I get the message ‘’Account requires administrator approval’’.

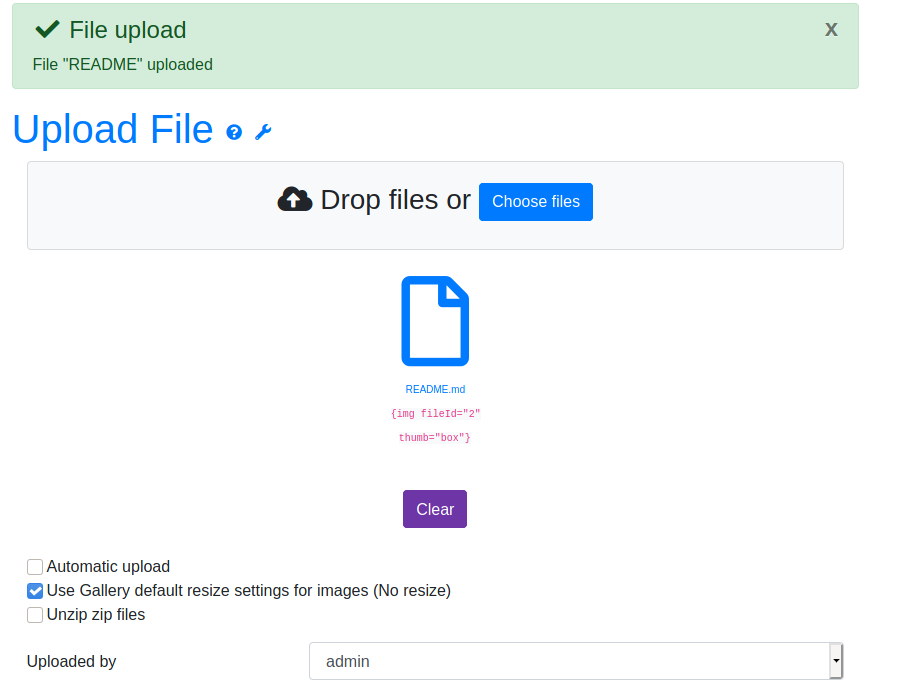


After turning intercept off I was given access to Tiki Setup.

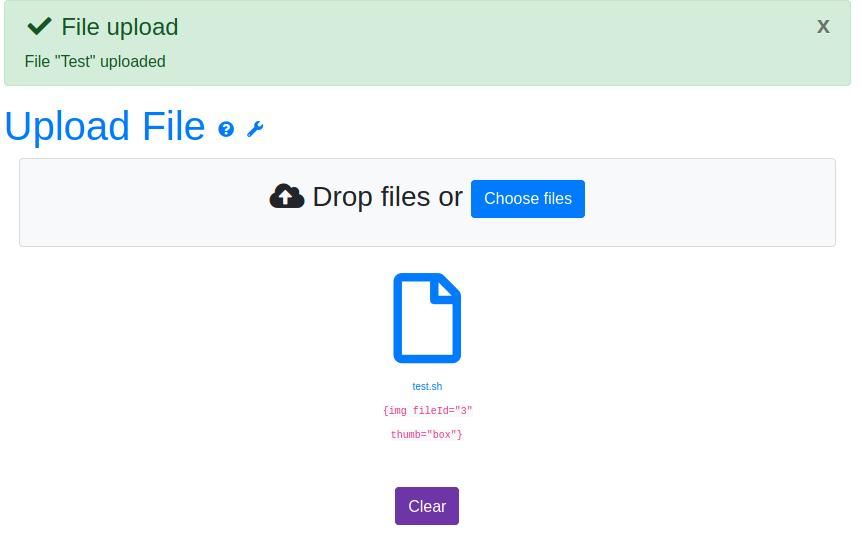


Now I can look around and see what I can find.



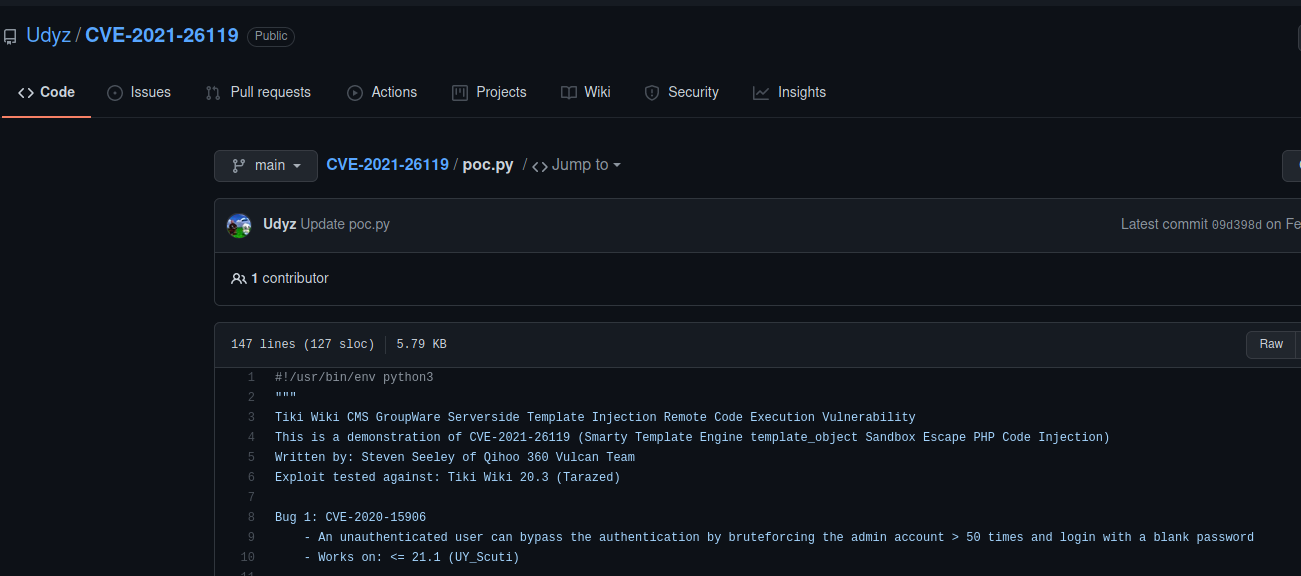


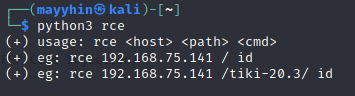
Allows me to upload images.



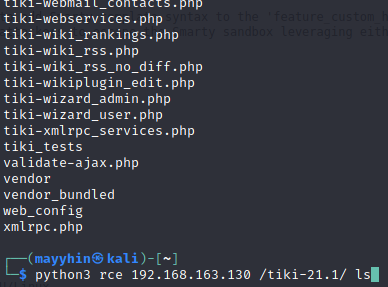
Made a file with a script in to try and get me a reverse shell, but did not work so I am going to try to find different RCE’s to use.

Found RCE on exploit db for tiki wiki.

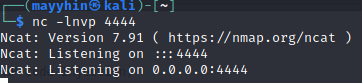




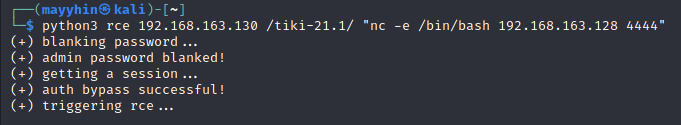
Telling me how to execute it.



just did ls to see if the RCE worked which it did as it listed the directories.

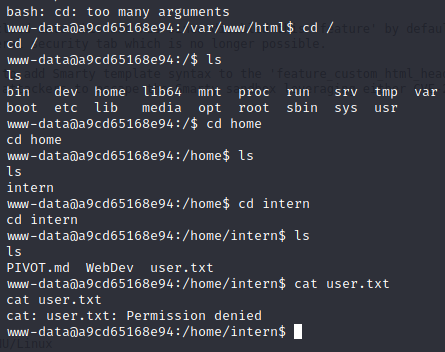


I set up a netcat listener on a different terminal.

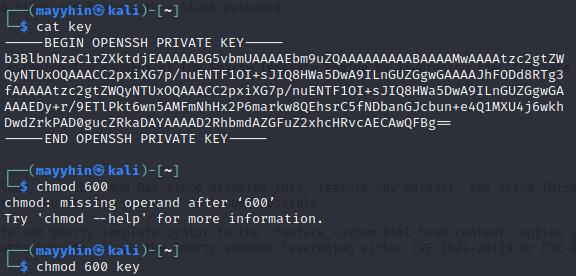


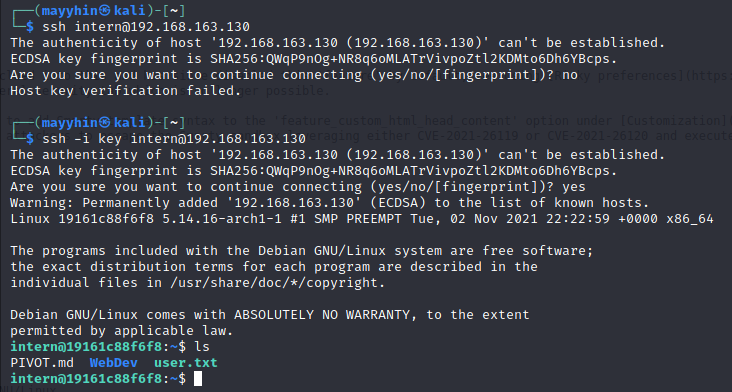
Connected to my netcat listener with this script. The script runs through the target machine so it gives me a connection between the target machine and my own.

Got a reverse shell just by running the script in the ip. This script basically did all the authentication part for me as well.

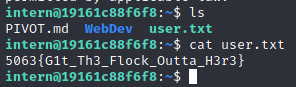


Looked through directories until I found the user flag. Permission is denied so I need to ssh into it. Found the keys in the system so I copied it onto my own system, using nano and changed permissions using chmod 600.





‘ssh -I key’ allows me to login using the key I found.

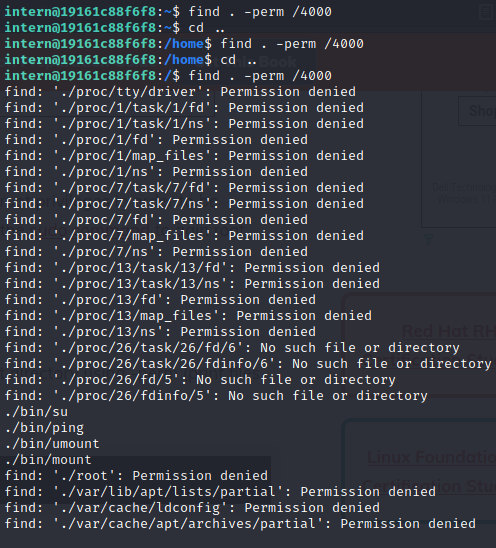


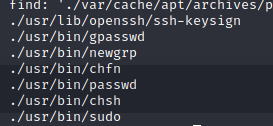
First flag!

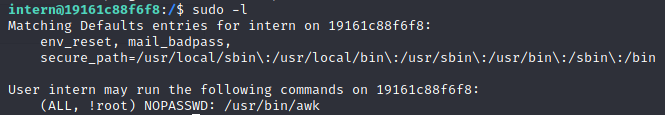
## Root Flag

Now I have access to the system I need to find a way to escalate privileges to gain access to root user and flag.

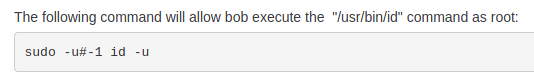
Searching for files with SUID set.

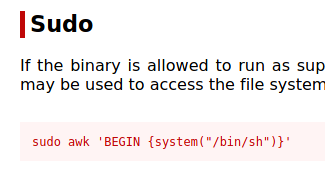




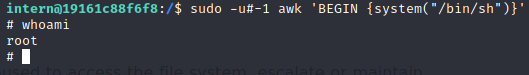


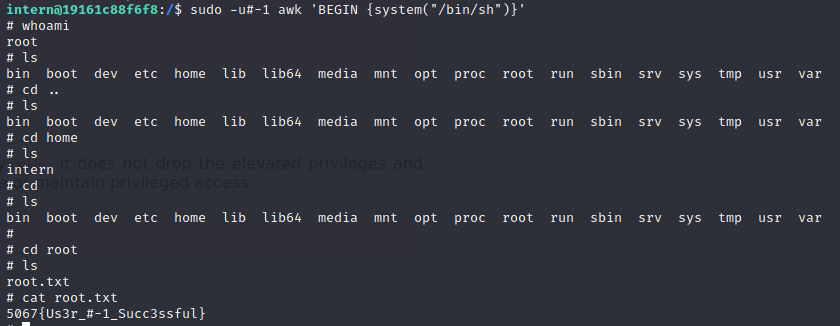
Usr/bin/sudo is available. Sudo –version lets me know it is version number 1.8.19p1 so I can google some exploits. Awk is also available to the system.





So the full code I used is ‘’ ’sudo -u#-1 awk 'BEGIN {system("/bin/sh")}' ‘’





Got the Flag!