

---

Designing and implementation of CTF VM which  
teaches aspects of cybersecurity.

AIM: To design and implement a CTF VM for beginners and amateurs who are trying to get into the cybersecurity world and provide them with actual real word scenarios which gives them a leap for their future journey

---

PURPOSE: A “Capture The Flag” (CTF) is a gamified set of technical exercises designed to teach and test practical cybersecurity skills in a safe and secure environment. CTFs are flexible, multi-faceted training tools which allow for practical problem solving and the development of technical and analytical skills. Such “game based environments” are well suited to group-work and foster team skills as the participants partake in various simulated scenarios.

---

## TABLE OF CONTENTS

---

1. Introduction
  2. Requirements for this project
  3. Implementation
  4. Screenshots and supportive files.
  5. Conclusion
  6. References
- 

### Introduction

Capture The Flag (CTF) is a cyber exercise where participants look for a hidden clue or file, a.k.a. the flag, by using cybersecurity tools. They are very common and no experience is necessary to play. The game gives you a taste of real world cybersecurity with activities often designed by cyber pros.

Participating in hacking challenges like Capture-the-Flag (CTF) offers numerous benefits to individuals interested in pursuing a career in cyber security. These challenges provide a platform for individuals to showcase their skills, gain prestige, learn from

experts, earn monetary rewards, and network with professionals in the industry.

CTF challenges require participants to think critically and solve complex problems. This helps to develop problem-solving skills, which are essential in the field of cybersecurity.

In this project we have designed a machine with the theme of an educational cybersecurity playground for beginners. The main goal of this project is to give a little bit of taste of the cyber world at a initial level for beginners or students.

### Requirements for this project

---

- Ubuntu 18.04
- ftp
- ssh
- apache
- http
- four users
- HTML page

## Implementation

---

The machine was implemented and designed in a 18.04 version of ubuntu server. Four flags were totally implemented in this machine. Various services like ssh, ftp, http, https, etc were installed and various ports such as 21,22,443,8080 were configured for the machine, one of the requirement for this project was to have at least 4 open ports, this requirement was satisfied here. the next requirement was to have at least 3 users in the server. We have created four users named user, albert, cryptoacid and finalflag to meet this requirement. an HTML page was designed as well for this.

Designing and setting up the flag was the next step. The requirements for the flag was to setup at least five flags. Ive tried to implement five flags. But due to the challenges we faced during the setup for the 4<sup>th</sup> flag I were forced to quit and make it only 4 flags. The first flag was placed in a directory called user, in this directory we created different files and directories to make it a little hard for the participants. The participant must provide a password and username inorder to access the user directory. The username and password was provided in the HTML page, but we setup it to only show for 3 seconds. Just to make it more interesting.

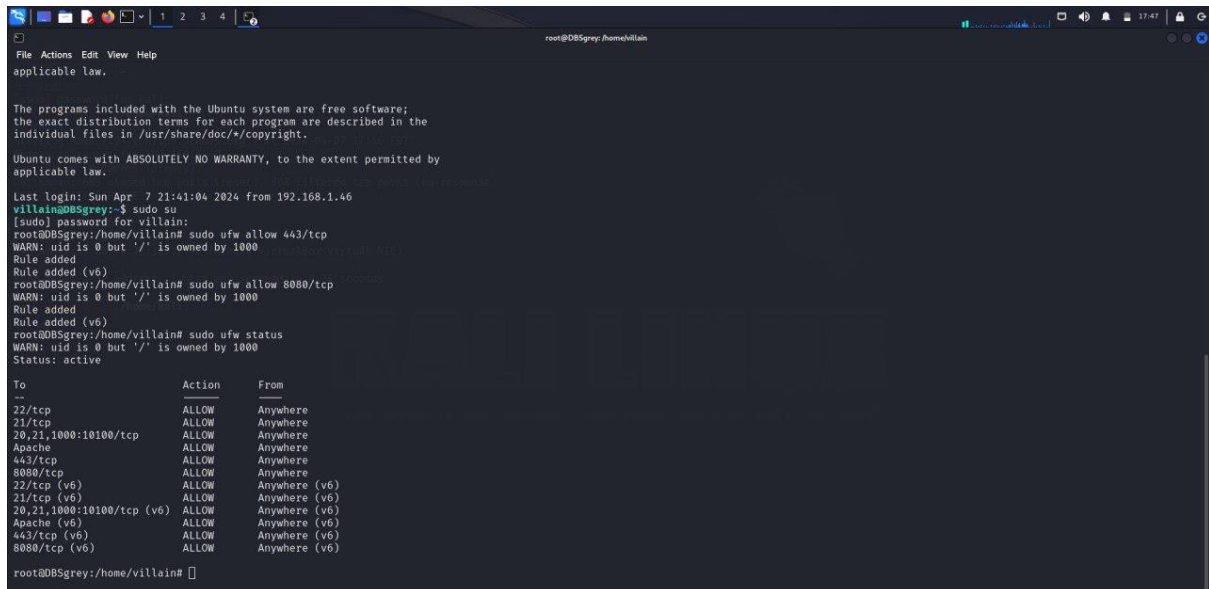
For the second flag we decided to embed the flag inside the HTML page as command. We placed a very long string inside the HTML page, inside the string we carefully placed our second flag. The second flag contains username and password to access the third directory which contains the third flag. The participants who doesn't have very knowledge on HTML gets to know about the page source section, that was our intention behind the placing of second flag. Placing a flag inside a HTML page gives the idea about users the importance of page source viewing and gives them a little real taste of cyberworld

For the third flag we give it a little twist. The third flag was placed inside the directory called albert. The directory albert is only accessible to user called albert, we make sure other users doesn't have access to the third flag directory using chown and other commands. We placed our third flag using steganographic method. We make sure to embed the flag inside a file in this directory with a passphrase. The passphrase to extract the embedded file was also hidden in another file using steganography, but for this it doesn't require any passphrase to extract. So to retrieve the the third flag, the participant should find the file that contains the passphrase, after extracting the passphrase, the participants should use this passphrase to extract the third flag. The thirdflag contains username and password for the next flag.

For the fourth flag we decide to create a puzzle, which is solved gives the fourth flag. For this we decided to create a script file for this, unfortunately to execute a script flag, the user must have r+x permissions, so this will make participant to read the script in a editor, which would make the puzzle useless, we tried various ways such as creating files with passwords using gpg, etc but it didn't make it, so we were forced to make the final flag the fourth flag.

The final/fourth flag was placed in a directory called crypto which is only accessible to user crypto acid. In this directory we created a text file which contained a hash value. The participant must decode the hashed value to plain text in order to have the clue for the fourth flag. The hashing technique used was whirlpool. This hashing algorithm is a rare and unpopular one which makes it somewhat difficult. The decoded hashed value will provide the username for the directory containing the final flag. For the password for this directory, the participants must brute force their way into it. We provide them with multiple wordfiles which contain 1000s of passwords, so it forces them to try brute forcing the password with each wordlist till they get the password using tools like hydra, john, hashcat, etc...

## Screenshots and supportive files



The screenshot shows a terminal window with the following content:

```
root@DB5grey:/home/villain#
File Actions Edit View Help
applicable law.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

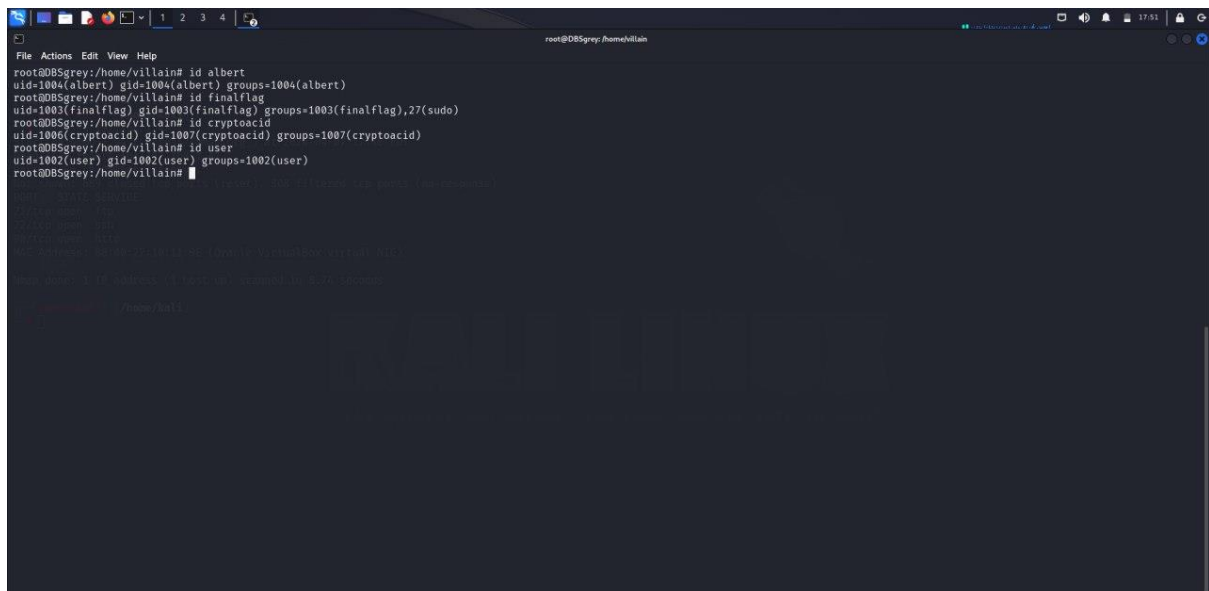
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sun Apr  7 21:41:04 2024 from 192.168.1.46
villain@DB5grey:~$ sudo su
[sudo] password for villain:
root@DB5grey:/home/villain# sudo ufw allow 443/tcp
WARN: uid is 0 but '/' is owned by 1000
Rule added
Rule added (v6)
root@DB5grey:/home/villain# sudo ufw allow 8080/tcp
WARN: uid is 0 but '/' is owned by 1000
Rule added
Rule added (v6)
root@DB5grey:/home/villain# sudo ufw status
WARN: uid is 0 but '/' is owned by 1000
Status: active

To Action From
--
22/tcp ALLOW Anywhere
21/tcp ALLOW Anywhere
20,21,1000:10100/tcp ALLOW Anywhere
Apache ALLOW Anywhere
443/tcp ALLOW Anywhere
8080/tcp ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
21/tcp (v6) ALLOW Anywhere (v6)
20,21,1000:10100/tcp (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
8080/tcp (v6) ALLOW Anywhere (v6)

root@DB5grey:/home/villain#
```

There are more than 4 ports opened here.

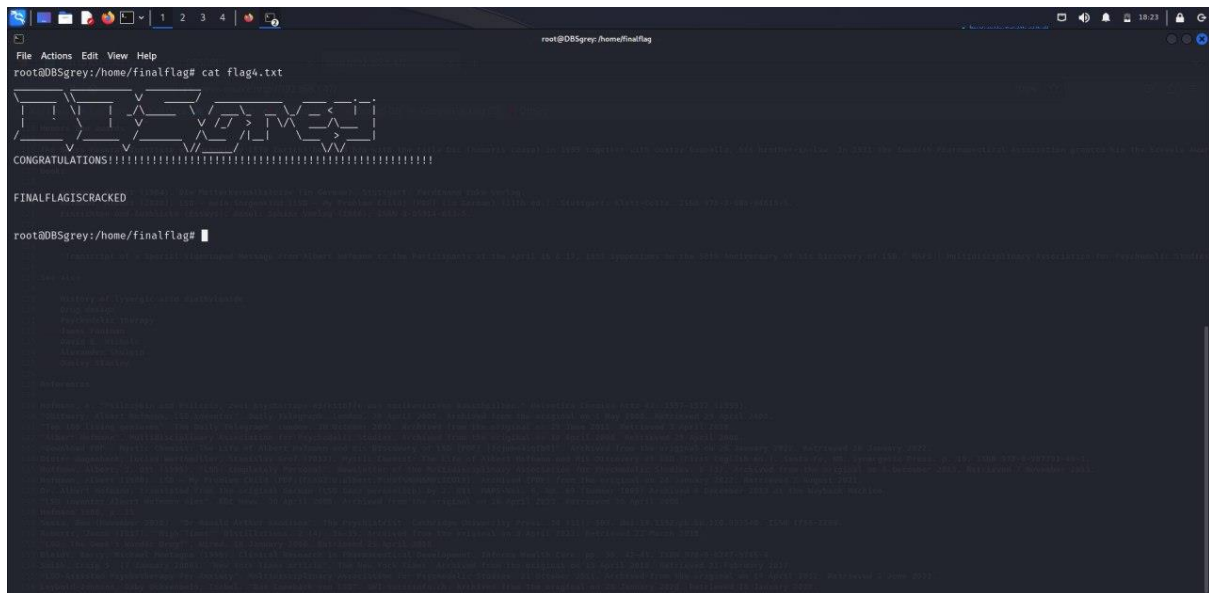


The screenshot shows a terminal window with the following content:

```
root@DB5grey:/home/villain# id albert
uid=1004(albert) gid=1004(albert) groups=1004(albert)
root@DB5grey:/home/villain# id finalflag
uid=1003(finalflag) gid=1003(finalflag) groups=1003(finalflag),27(sudo)
root@DB5grey:/home/villain# id cryptoacid
uid=1006(cryptoacid) gid=1007(cryptoacid) groups=1007(cryptoacid)
root@DB5grey:/home/villain# id user
uid=1002(user) gid=1002(user) groups=1002(user)
root@DB5grey:/home/villain#
```

More than 3 users are here as well, the last user which is finalflag is also an sudo user which meet two requirements.





We have implemented and designed atleast four flags for this machine.

## CONCLUSION

---

So in conclusion i have created a CTF machine on a ubuntu server. The theme of this project is education cybersecurity playground for beginners. We have placed one main clue for the flag and a flag in the html page. This would give the beginners who is trying to get to know cybersecurity field and who are experimenting CTF challenges for the first time, to check if theres any leading clues or anything helpful for them in the localhost page in the future when they are trying more CTF challenges. Hiding a flag in HTML source code and finding them will give any new schoolers an idea, there might be something in a webpages source code and to check them out.

For the next flag i used steganography to hide the clue for the flag in a mp3 file and the main flag in a image file. So this type of flag embedding gives the knowledge about steganography in cyber security. This would make them more vigilant the next time when they pass by any image or audio files in future. This also would give them knowledge about using steganography we can pass any important, personal or secret messages through images, audio files, etc. then i made a hashed file as a clue for the next flag. This gives a initial idea on hashing,decrypting how we can hash any strings using CLI, how we can utilize hashing.,etc. for the final flag we created a bruteforcing challenge. Ive provided the username, and multiple password lists, with one containing the actual password. So this would give them idea about bruteforcing, tools like hydra, john, etc. this also gives them the idea that only with the password list containing the original password gives them the correct password. And also still nowadays there are people who still make common passwords, such passwords can be bruteforced very easily using any common password word list. To conclude this CTF machine was implemented and designed for beginners and amateurs who are trying to get into the cybersecurity world

and provide them with actual real word scenarios which gives them a leap for their future journey, and how a CTF machine works, what are the common objectives in a CTF challenges, etc.

## References:-

- [TryHackMe: Simple CTF Walkthrough | by Skylar | Medium](#)
- [BEST FORENSIC TOOLS TO HIDE SECRETS PASSWORDS AND RECOVER FILES - Hackonology](#)
- [Online Hash Generator | Password Hash Generator \(onlinewebtoolkit.com\)](#)
- [How to host a CTF | Self-hosting a CTF Platform \(CTFd\) | csictf | csictf \(medium.com\)](#)
- <https://www.youtube.com/watch?v=a-7zTK9j1Ss>
- <https://releases.ubuntu.com/18.04/>
- [https://www.youtube.com/watch?v=HwztkS7oF5s&list=PLZzYJlCioXjZuz4pcsyWj\\_KSLwZUVO-Sx](https://www.youtube.com/watch?v=HwztkS7oF5s&list=PLZzYJlCioXjZuz4pcsyWj_KSLwZUVO-Sx)
- <https://www.youtube.com/@Hackergrity>
-