

The logo is centered on a solid black rectangular background. It features the word "escape" in a lowercase, sans-serif font, colored in a vibrant magenta. Below it, the word "ROOM" is written in a large, bold, uppercase, sans-serif font, colored white. The letter "O" in "ROOM" is replaced by a white keyhole icon. The entire text is framed by a white, stylized square border that is open on the left and right sides, resembling a door frame.

escape

ROOM

Introduction

Capture the flag is a challenge series which are designed to test the challenge solving skills under the various difficulty levels. Some CTF are design to simulate real life hacking scenarios. Participants have to find specific piece of text call flag which are hidden in the server.

We have planned to make this web based CTF call “Escape Room”. Escape Room is a locked room, which has many clues and hints to unlock the room. A room is called as a level. Normally an escape room has many levels, also our digital escape room has many levels. Players, who want to play our escape room must find a flag to unlock next level\room.

We used below techniques in our CTF,

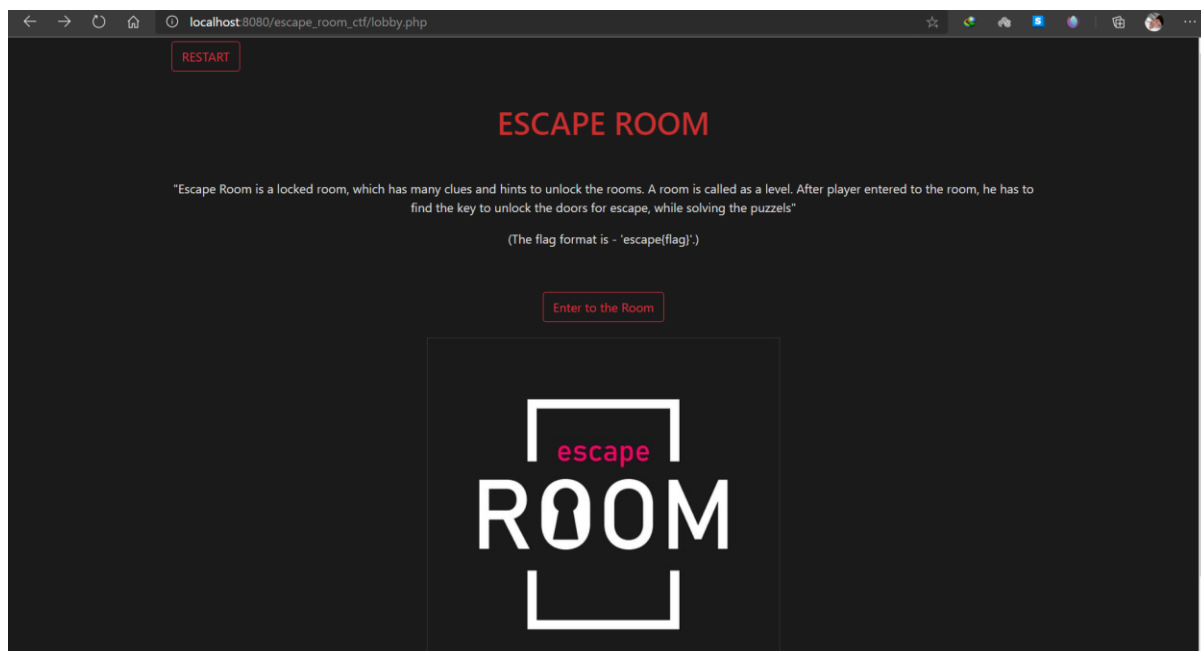
- Cryptography - Typically involves decrypting or encrypting a piece of data
- Steganography - Tasked with finding information hidden in files or images
- Binary - Reverse engineering or exploiting a binary file
- Web - Exploiting web pages to find the flag
- Pwn - Exploiting a server to find the flag

Audience

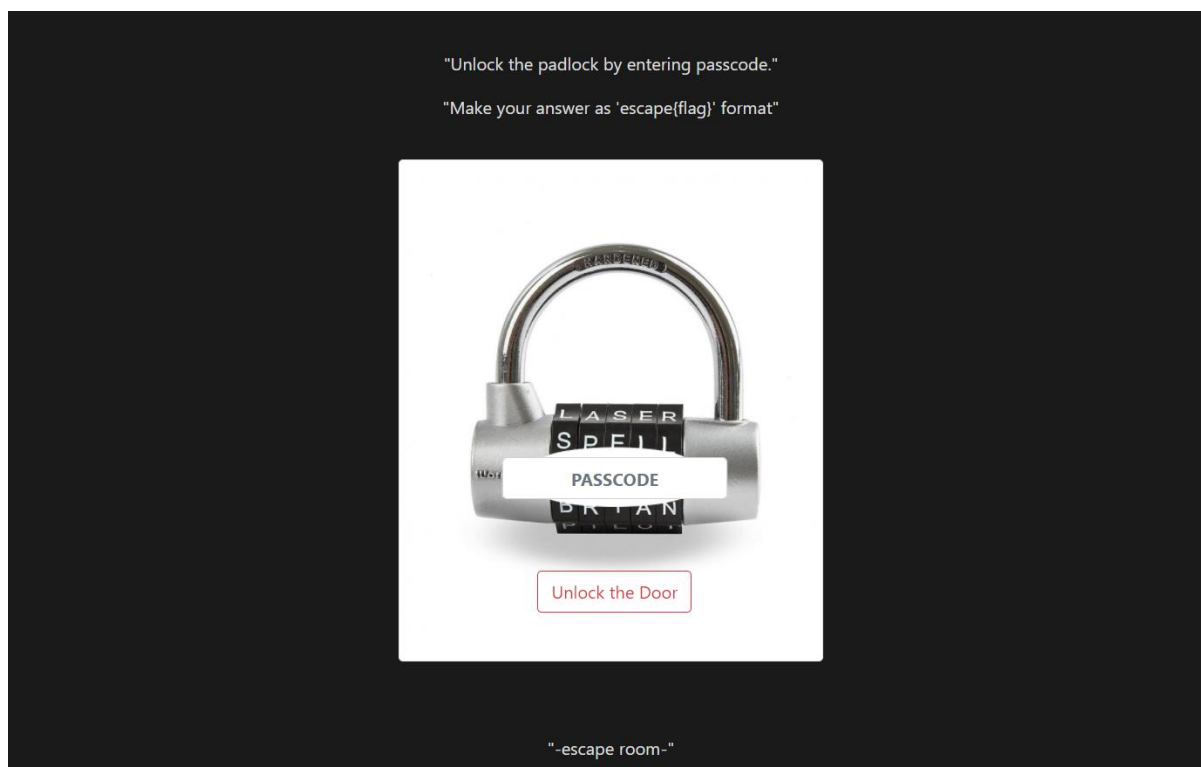
We are targeting on cyber security related organizations. When the organization is recruiting trainees or intern students, they can test and get idea about new recruits’ skill level and knowledge about the cyber security and analytical skills as in our CTF covered all basic technologies. It is better than asking theory questions.

Implementation

The landing page of the CTF showed like below. It is the lobby of the escape room and, in there gives brief intro about the escape room CTF to players.

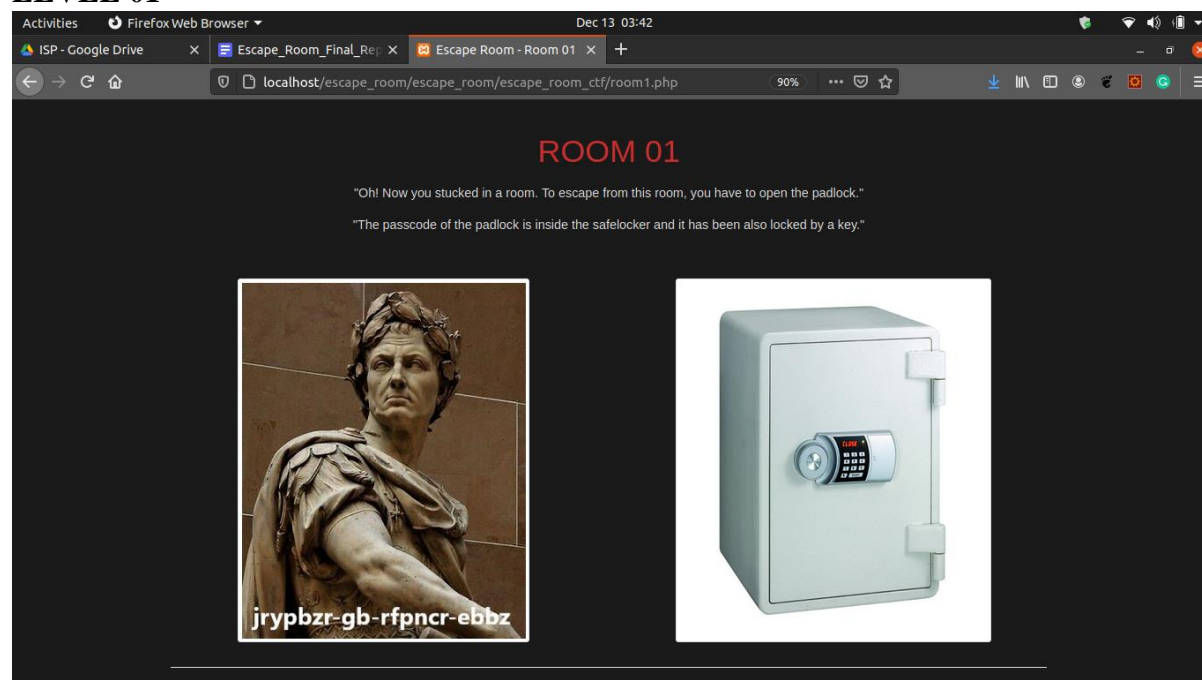


Every room locked and to escape those rooms, player has to find passcode/flag of the padlock to unlock the door. And submit passcode in the format of 'escape{flag}'.



When unlocking new room, it creates sessions for each room. Therefore players cannot go to other levels without unlocking them. And, as the flag is checked in server side, it cannot retrieve from client side.

LEVEL 01



This is the first level in our CTF. We call simply rooms. when the user enters room 1, they can see 2 images. one is an image of a safe and the other one is an image of the caesar.

We have hidden the flag inside the safe image using the steghide tool. It is a popular tool to do steganography. but to do that, the user needs a paraphrase, which is shown in the caesar image. encrypted using popular rot 13v caesar cipher.

rot13.com

[About ROT13](#)

jrypbzr-gb-rfpncr-ebbz



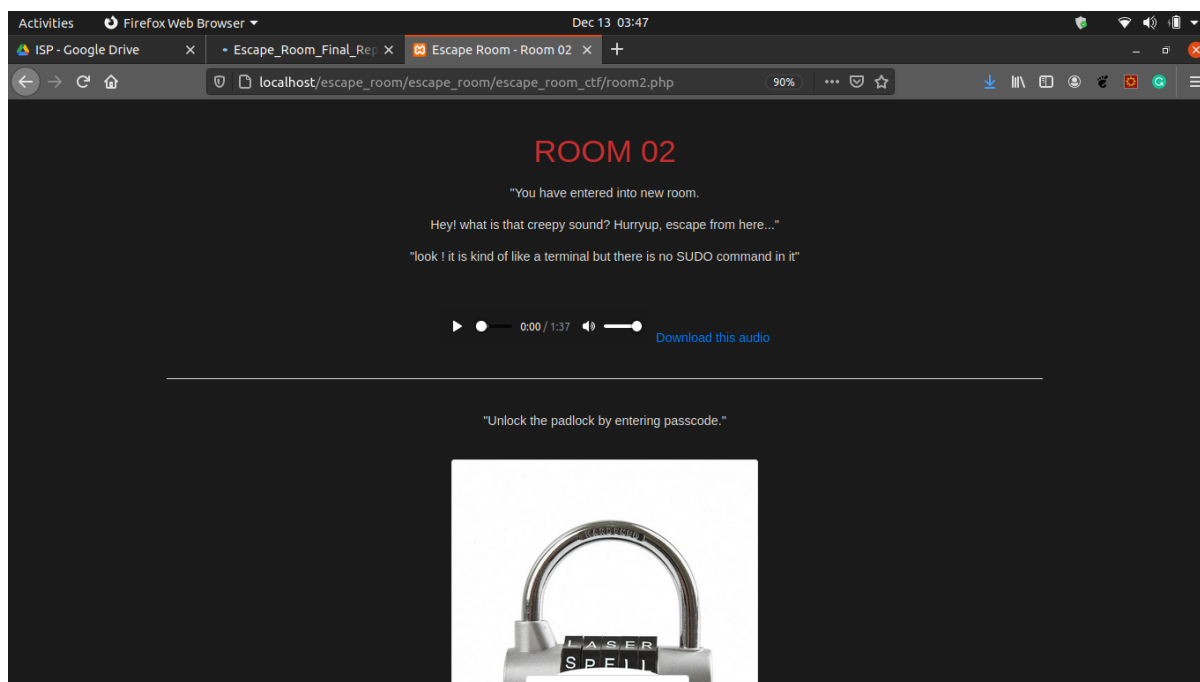
ROT13 ▾



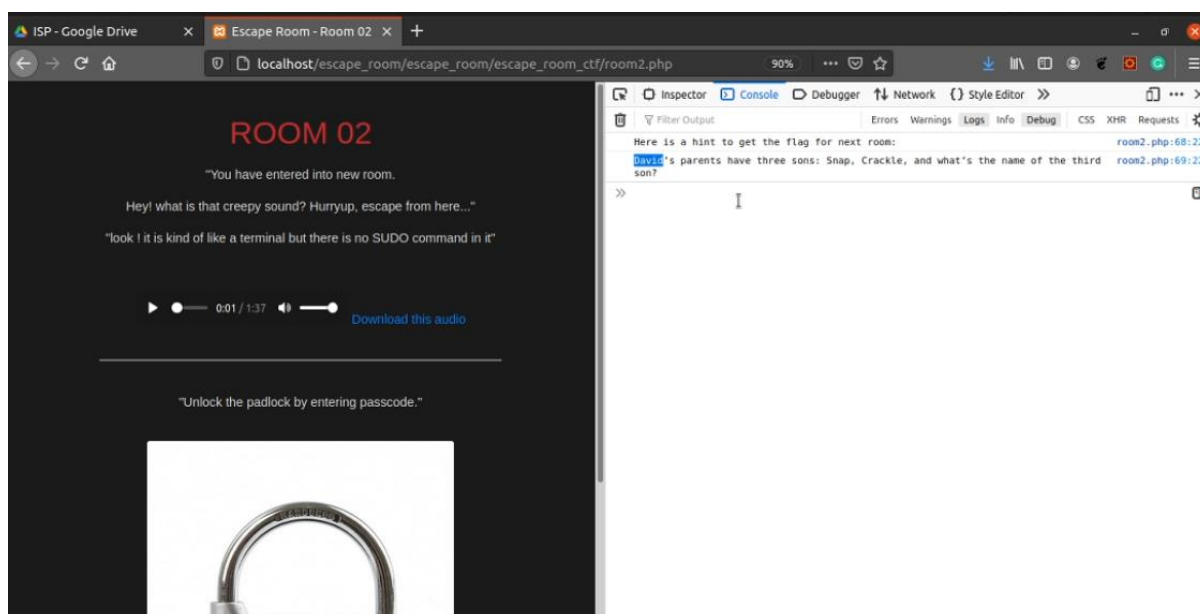
welcome-to-escape-room

```
To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ steghide extract -sf safe.jpg
Enter passphrase:
wrote extracted data to "emb.txt".
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ ls
emb.txt  escape_room  safe.jpg  vid
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ cat emb.txt
escape(welcome to escape room)
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$
```

LEVEL 02



We used the same steganography technology in this room too. but this time we used an audio file to hide the flag. Because our theme is “escape room”, we have provided the paraphrase to the audio file as a riddle in the browser’s console.

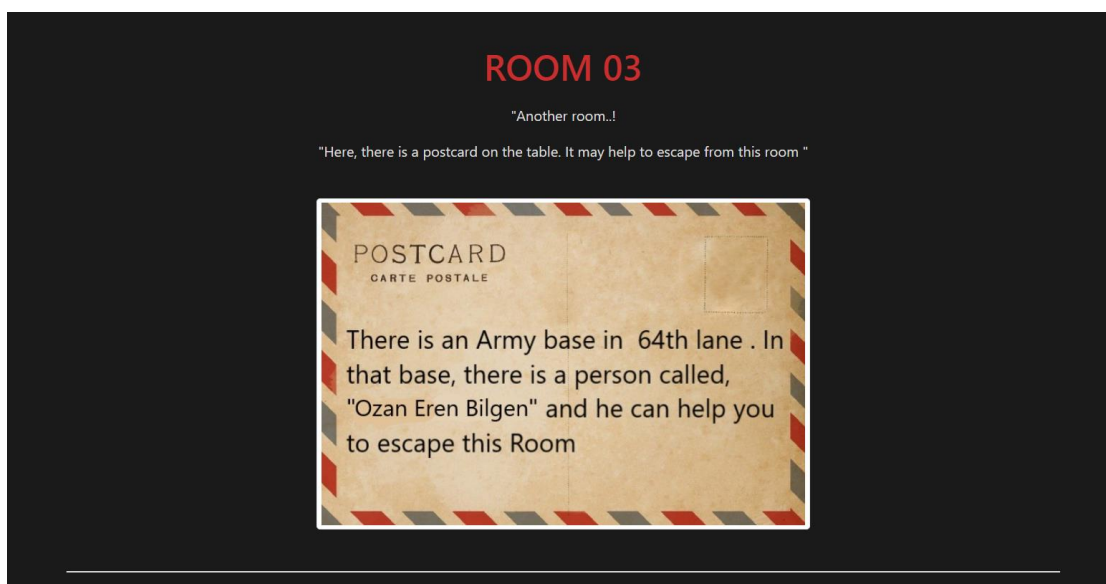


```

To embed emb.txt in cvr.jpg: steghide embed -cf cvr.jpg -ef emb.txt
To extract embedded data from stg.jpg: steghide extract -sf stg.jpg
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ steghide extract -sf lock.wav
Enter passphrase:
wrote extracted data to "emb.txt".
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ cat ebt.txt
cat: ebt.txt: No such file or directory
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ cat emb.txt
escape{12h3jgviyyhdgfu1234jhb1u2}
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$

```

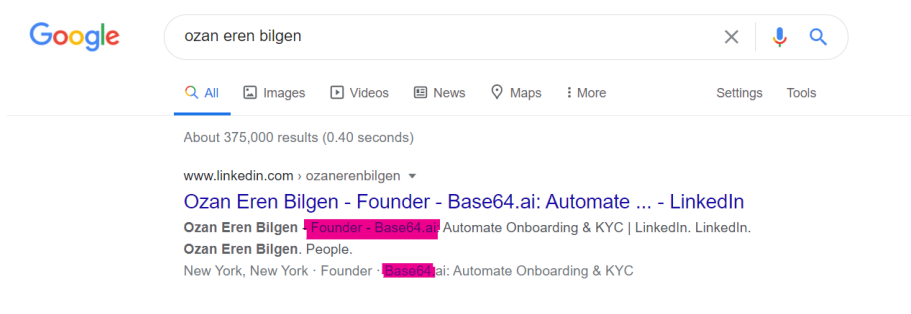
LEVEL 03



When players enter into room 3, they can see a postcard with message.

In this level we try to test players' thinking level and information gathering information. If player is more talented, they can understand the hints given in the postcard. As there mentioned, 'base' '64' words in the message, player could be getting idea, this is about 'base64' encoding.

Or players can search the name which mentioned in message and realized about the base64 according to search results.

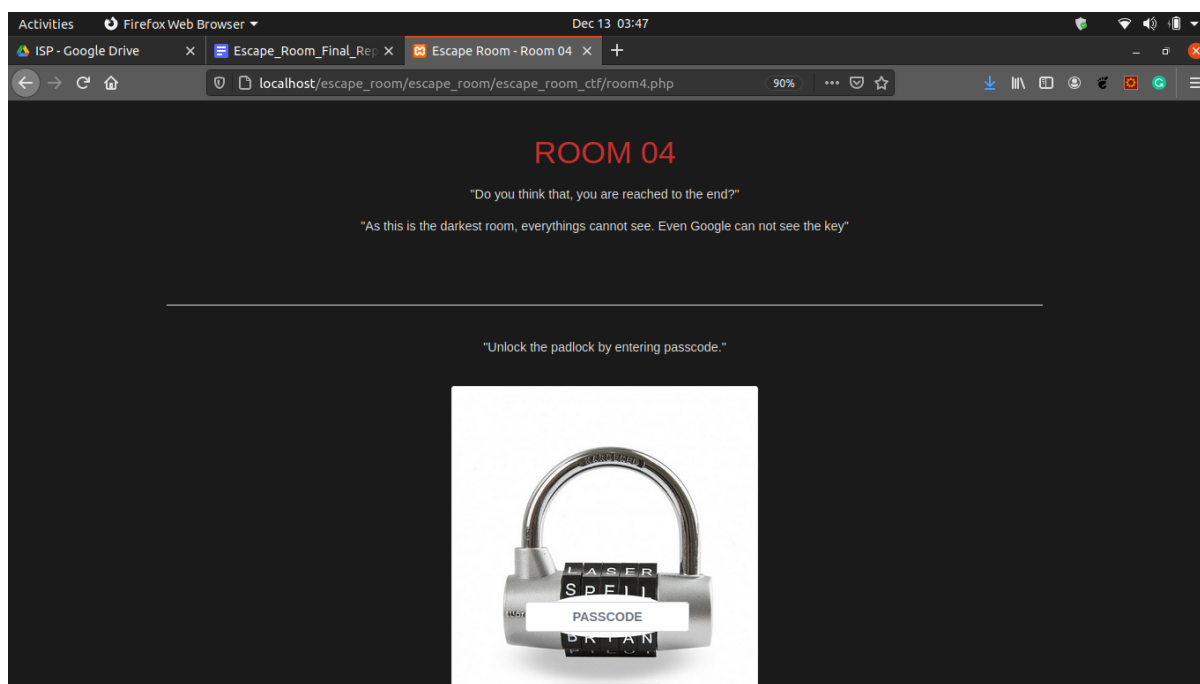


Then players have to encode the name using base64. They can use linux terminal or online tools to encoding the name.

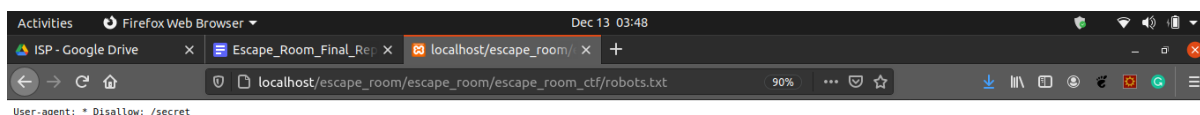
Finally submit the passcode in valid flag format.

```
escape{T3phbiBFcmVuIEJpbGdlbg==}
```

LEVEL 04

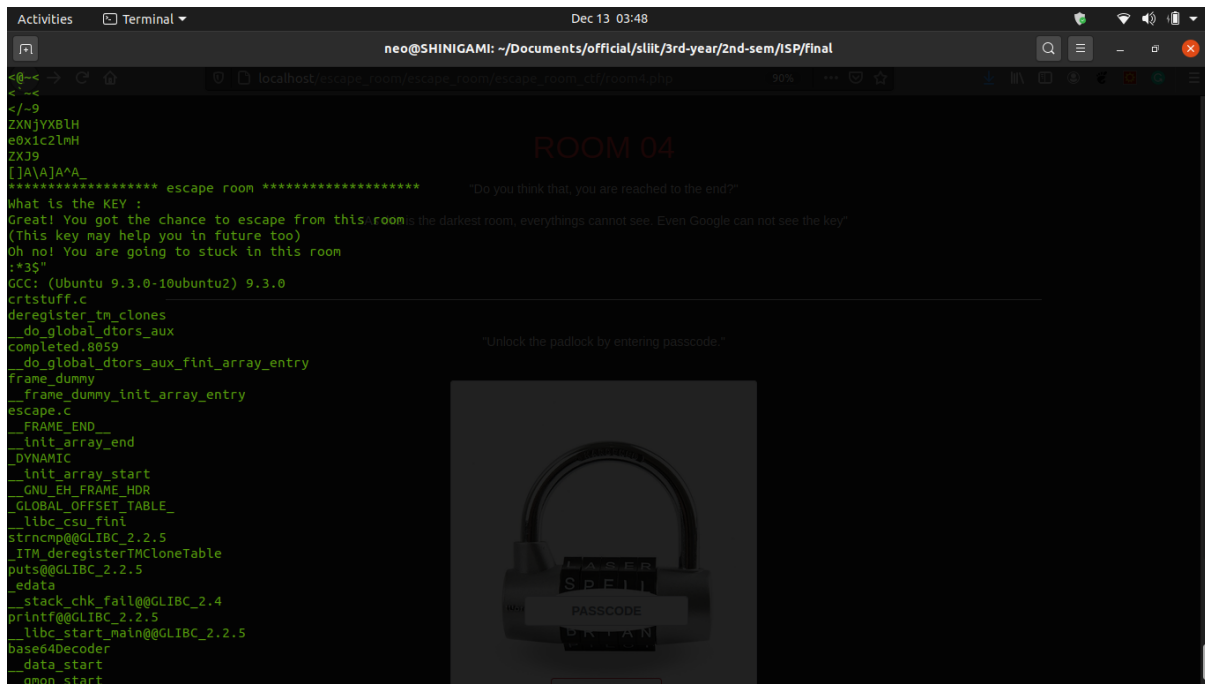


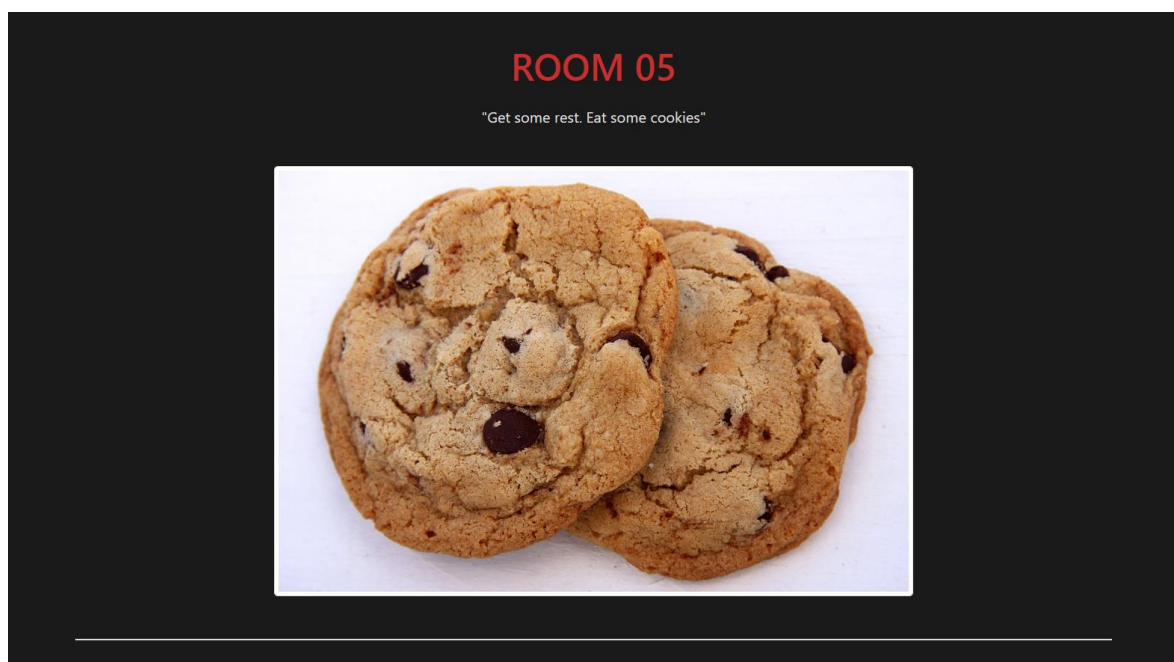
In this room, there is a hint which says “even google can not see...”. yes , there is something google can not see on our website. we indicate the files and folders, which should be hidden from the google’s crawler in robot.txt file.



According to the above screenshot, a folder called 'secret' is hidden from the crawler. and there is a c program. the user needs to download the c programme and the user has to reverse engineer it and find the flag.



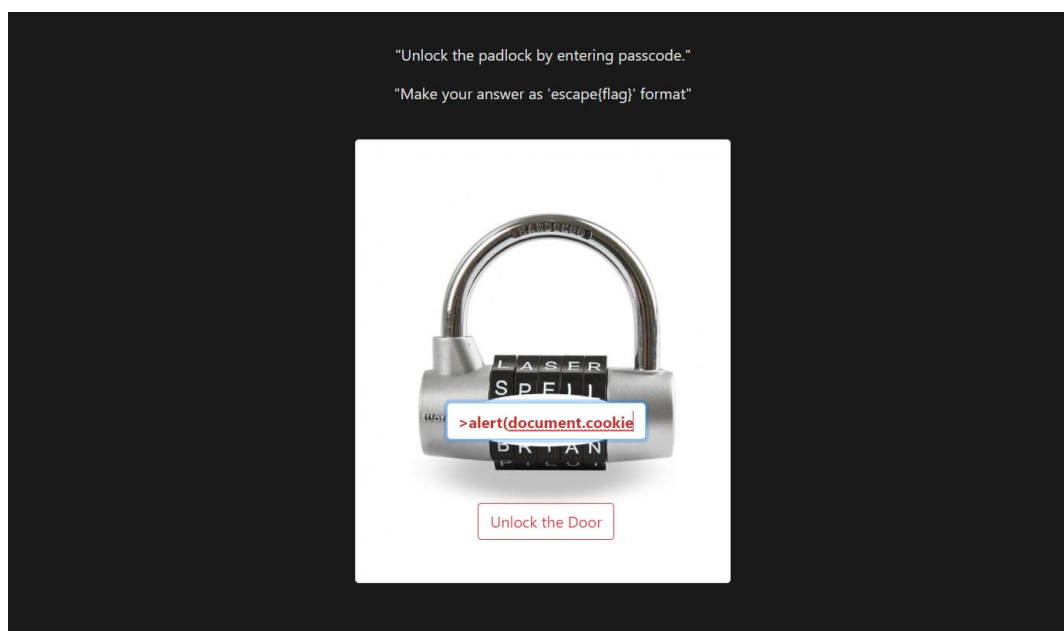


LEVEL 05

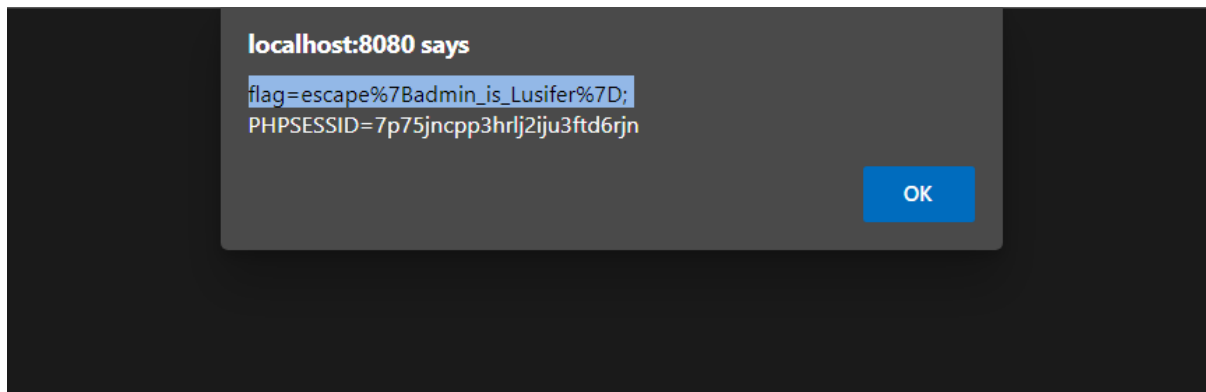
In this level we can test players XSS knowledge. Here we give hints about the cookies. Players have to realize that here we are not talking about food.

As there is a textbox in the padlock, players can retrieve web cookies using JS alert.

```
<script>alert(document.cookie)</script>
```



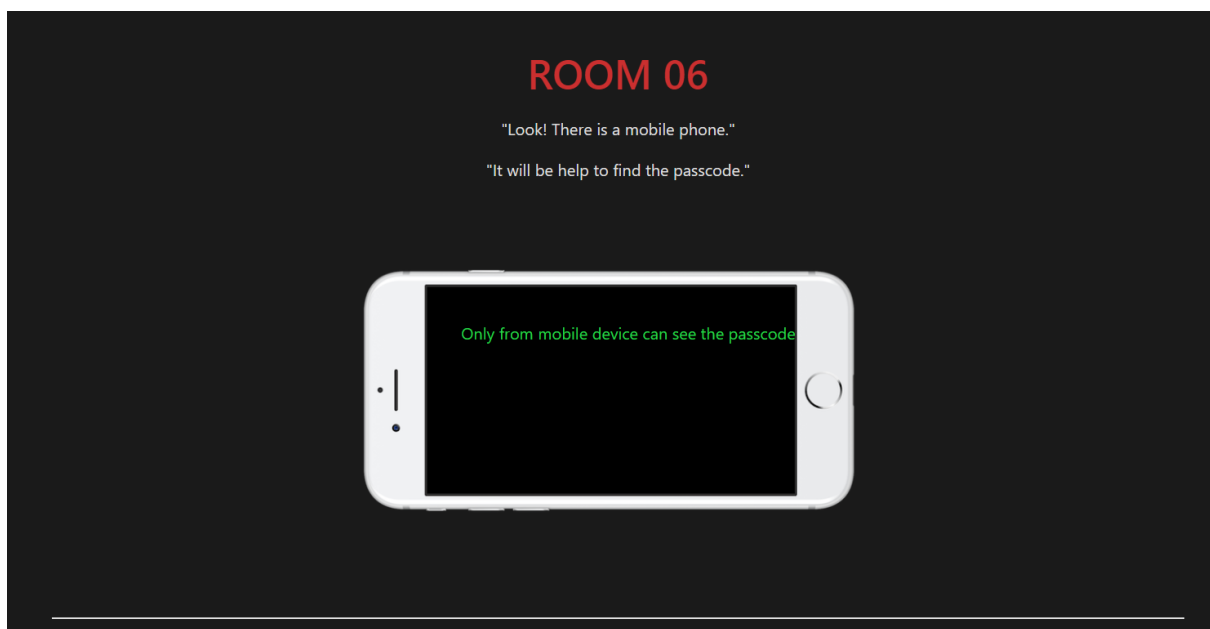
Then player can retrieve the flag from cookies.



Also, they have to arrange the flag by replacing html characters to ascii characters.

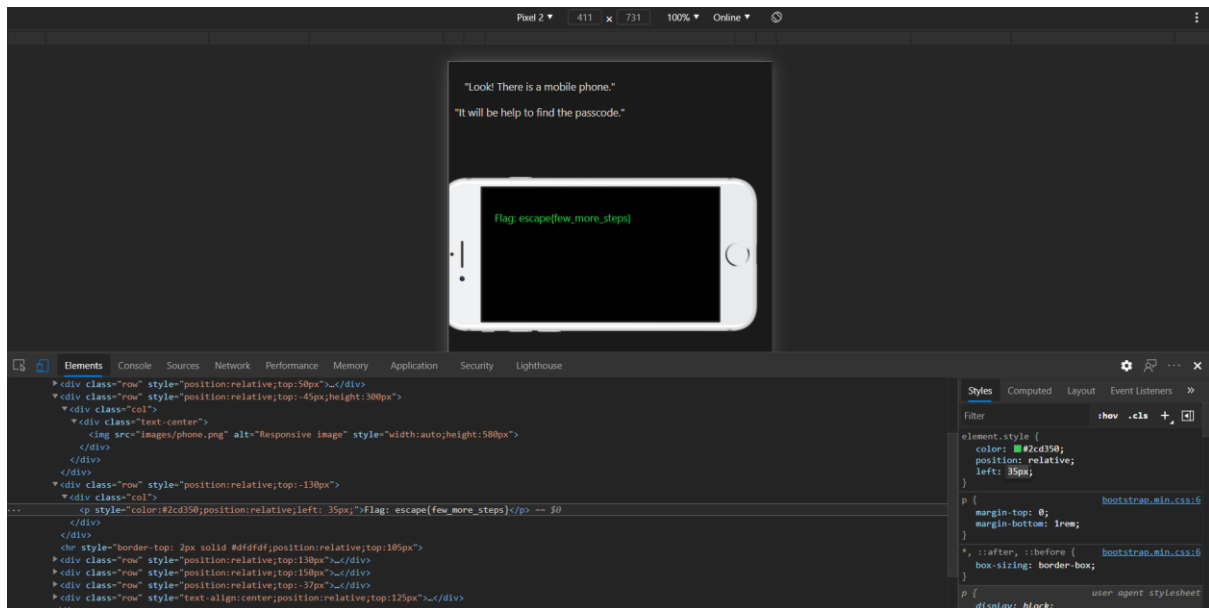
```
escape{admin_is_Lusifer}
```

LEVEL 06



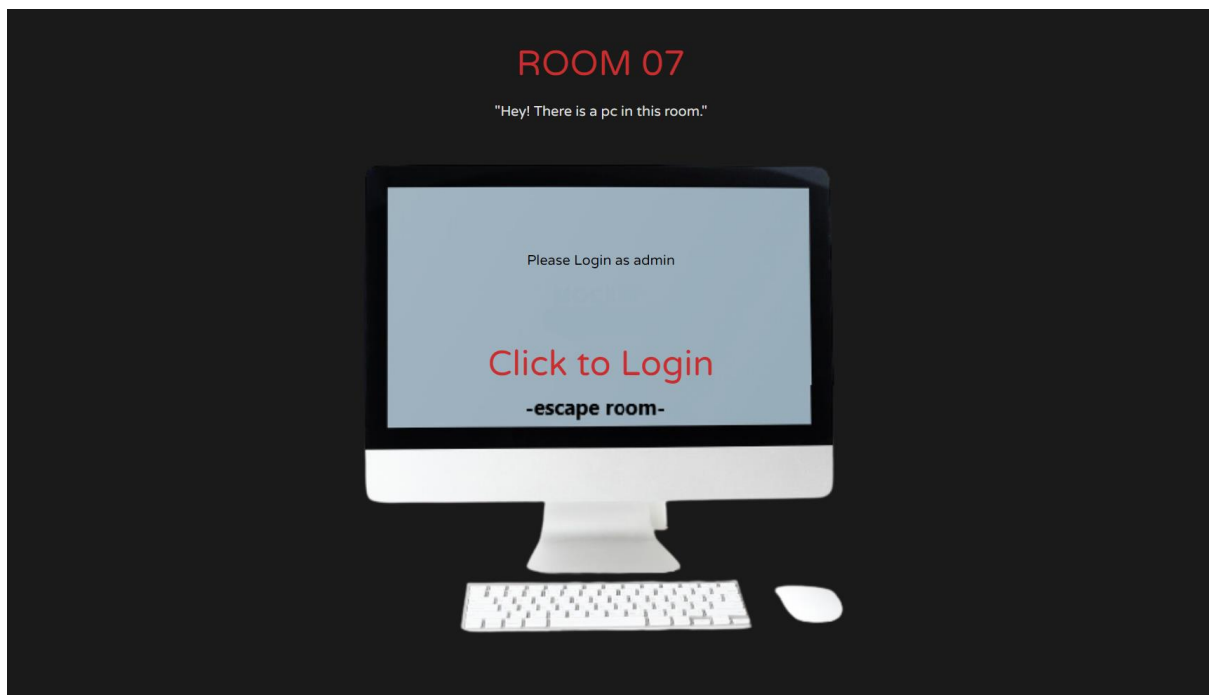
Room 06, it says that only through mobile device can retrieve the passcode.

In this level we checked the device which the website is on and only mobile view show the flag. Therefore, players use inspect element to toggle view into mobile device. Then they will be able get the flag.



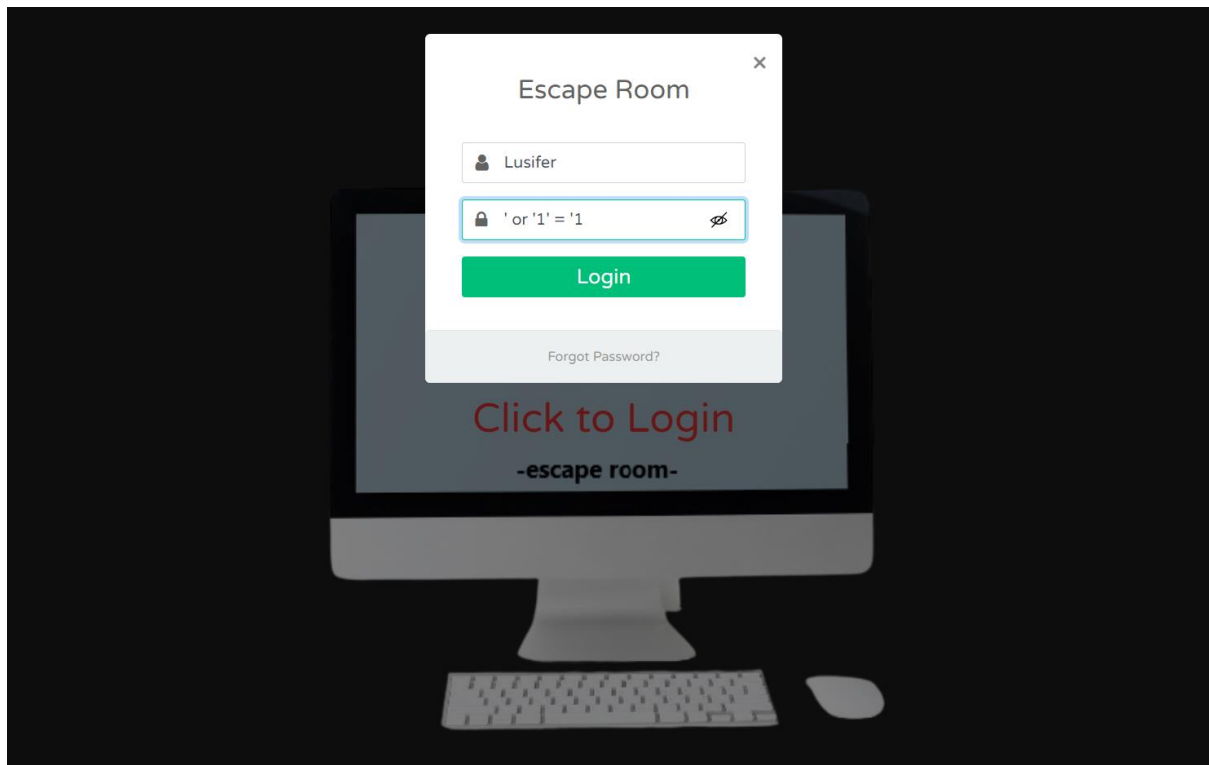
escape{few_more_steps

LEVEL 07

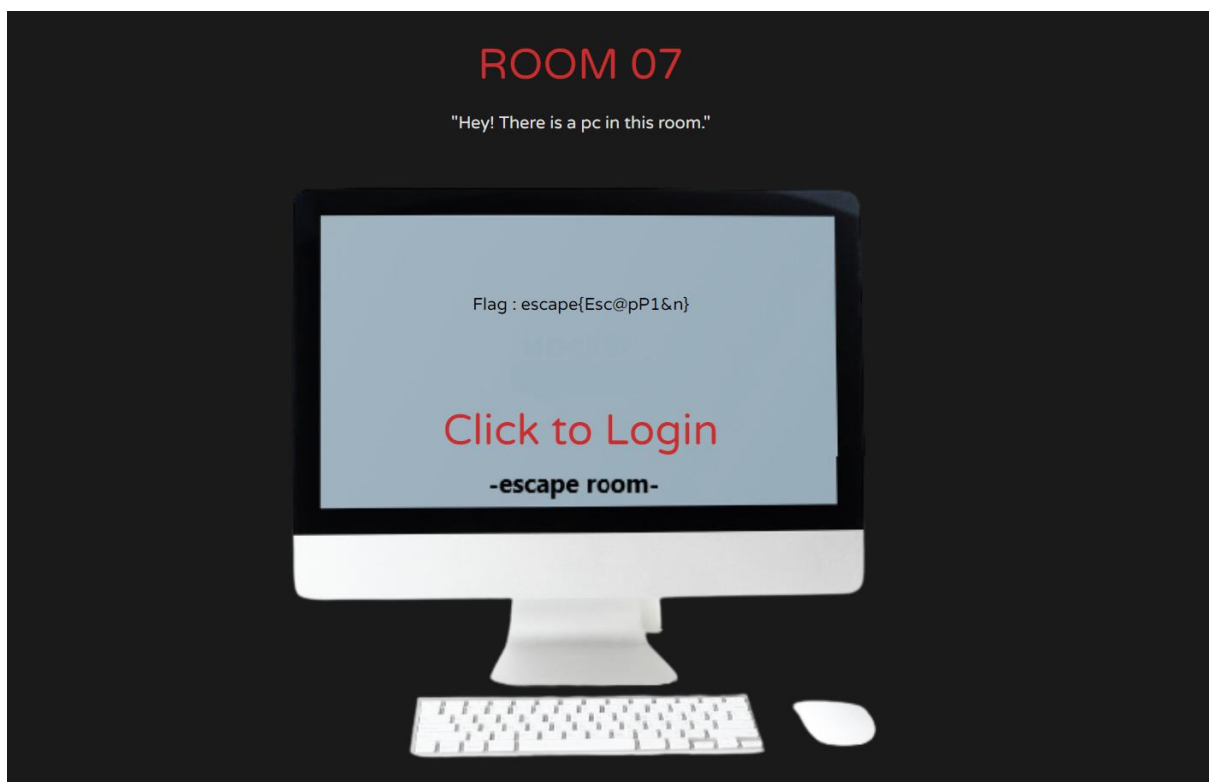


In this room, players can find a computer and it displays 'login as admin'.

When completing previous levels, players could get clues the admin is 'Lucifer'. But they do not have any idea about the password. Therefore they have to use 'bruteforce' attack or 'sql injection' attack to log into this computer. But easiest way is using 'sql injection'.

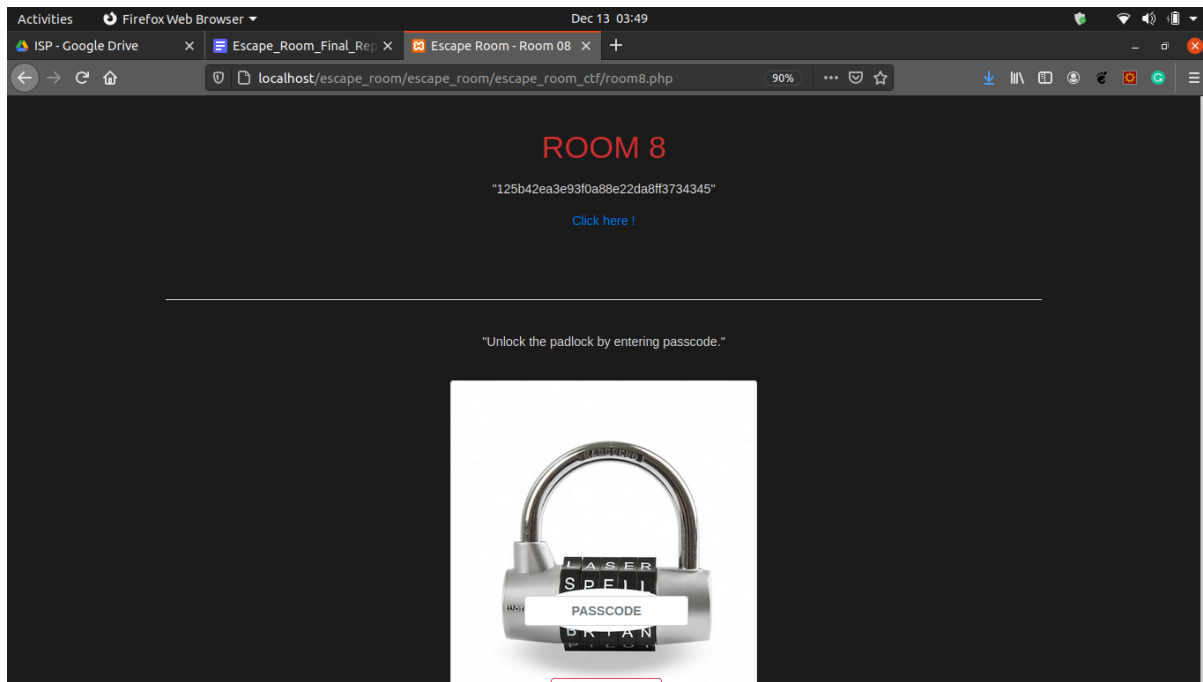


After players could login to computer successfully, it will show the flag.



Also, it is the correct login password. And players can unlock padlock using this flag.

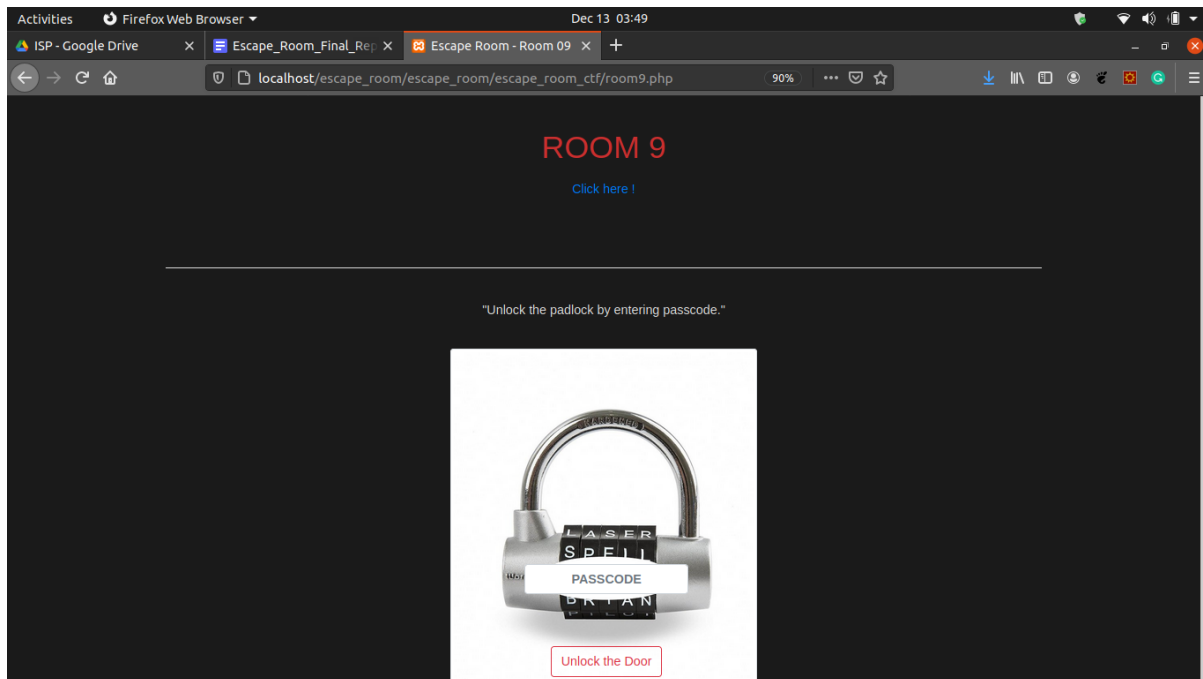
LEVEL 08



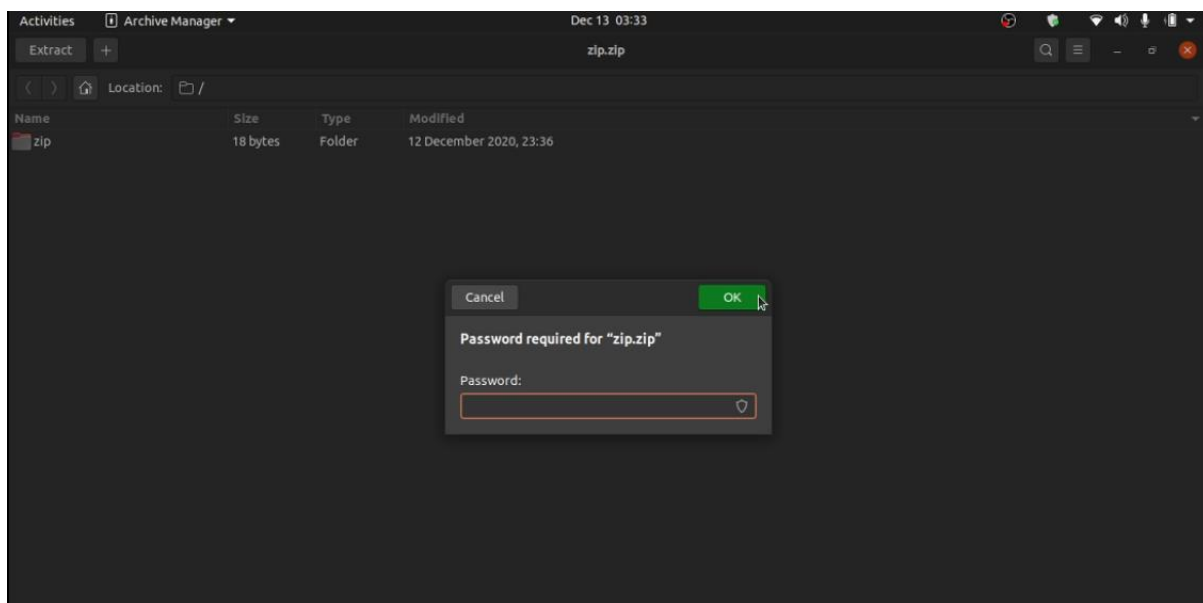
in this level users have to find the plain text value of a hashed value. the only way of finding the plain text of a hash is brute forcing. We have provided a wordlist which contains the plaintext of the hash, but users have to find the way to bruteforce it.

but there are some online popular rainbow pages, which may help the user to do this task for themselves

LEVEL 09




In this room, we provide the user with a zipped folder using a password. users have to find the password to unzip this folder and get the flag. users can use the same wordlist which we provided in the previous room. users have to find a way to brute force the unzip command.in unzipped folder there is a file called “flag.txt”




```

neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ fcrackzip -d rockyou.txt zip.zip
fcrackzip: invalid option -- 'd'
unknown option
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ fcrackzip -D rockyou.txt zip.zip
found id 34333231, 'rockyou.txt' is not a zipfile ver 2.xx, skipping
aaaaaa: No such file or directory
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ fcrackzip -D "rockyou.txt" zip.zip
found id 34333231, 'rockyou.txt' is not a zipfile ver 2.xx, skipping
aaaaaa: No such file or directory
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$ fcrackzip -Dup "rockyou.txt" zip.zip
PASSWORD FOUND!!!!: pw == bankrobber
neo@SHINIGAMI:~/Documents/official/sliit/3rd-year/2nd-sem/ISP/final$

```

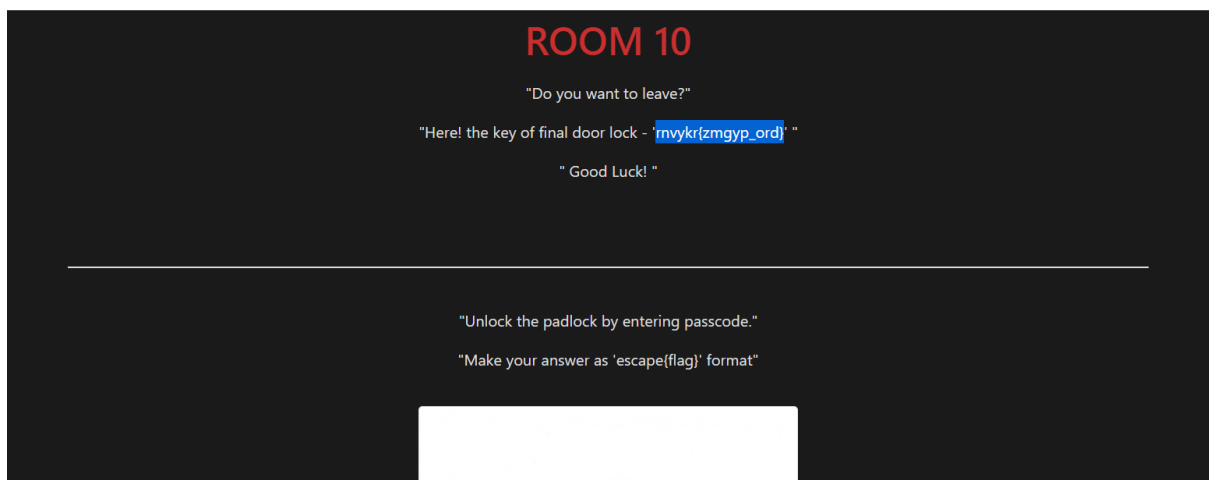


```

Open  flag.txt  Save  ~/Documents/official/sliit/3rd-year/2nd-sem/l...
1 escape{gotcha}
2

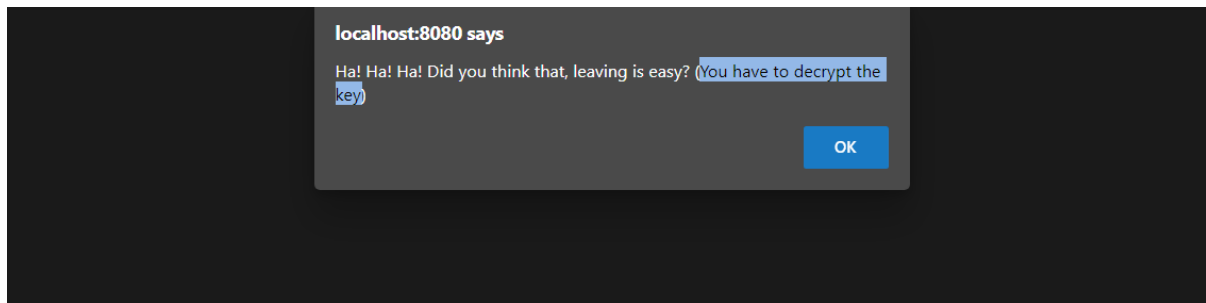
```

LEVEL 10

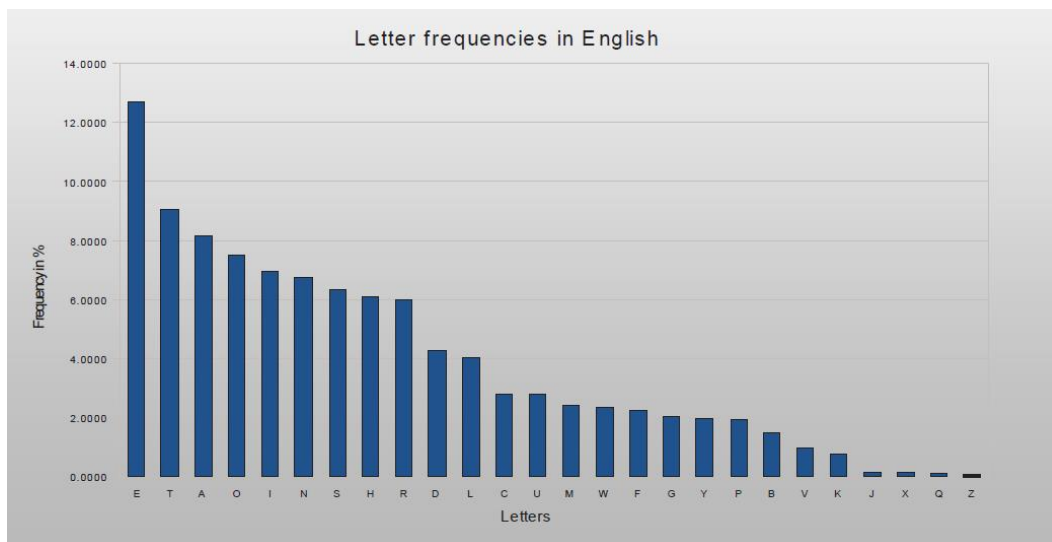


Room 10 is the final level of the escape room. When players escape from this room, the CTF will be completed.

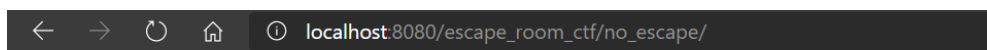
Here we have given the key for the padlock. Players can submit it. But it is an encrypted key. Therefore, players have to decrypt this key.



This key was encrypted by using ‘Substitution Cipher’. Because of that it hard to decrypt using bruteforce. Therefore they have to do ‘Letter Frequency Analysis’ manually to decrypting.



But if the player was able to find our ‘substitution alphabet’ which we used to encrypt the key while doing previous levels or in this level also, player will easily decrypt the key. If players did ‘dir search’ or manually tried to move into other directories they could find this file.



Index of /escape_room_ctf/no_escape

Name	Last modified	Size	Description
Parent Directory	-		
substitutue_alphabet	2020-12-11 13:46	180	

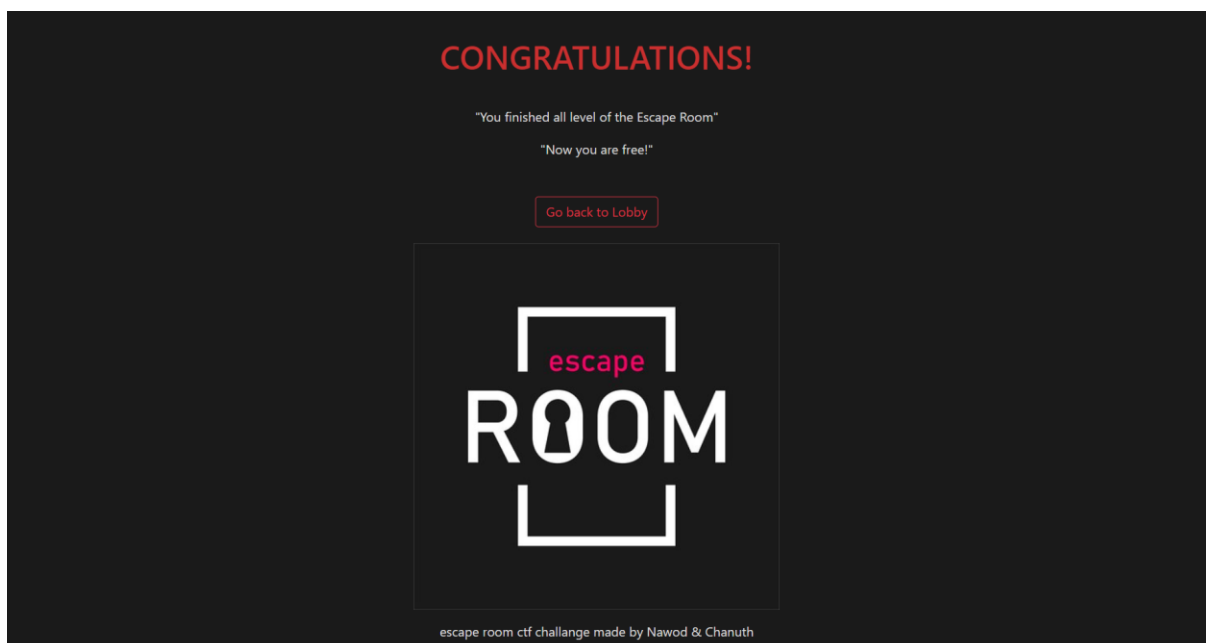
Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.4.4 Server at localhost Port 8080

```
no_escape > ≡ substitue_alphabet
1  A = Y
2  B = X
3  C = V
4  D = T
5  E = R
6  F = Z
7  G = W
8  H = U
9  I = M
10 J = Q
11 K = O
12 L = P
13 M = I
14 N = G
15 O = E
16 P = K
17 Q = L
18 R = S
19 S = N
20 T = C
21 U = A
22 V = J
23 W = H
24 X = F
25 Y = D
26 Z = B
```

The decrypted key is

```
escape{final_key}
```

And it is the final flag of Escape Room CTF. Using it players can escape from the challenge.



THANK YOU!