TryHackMe

Biohazard

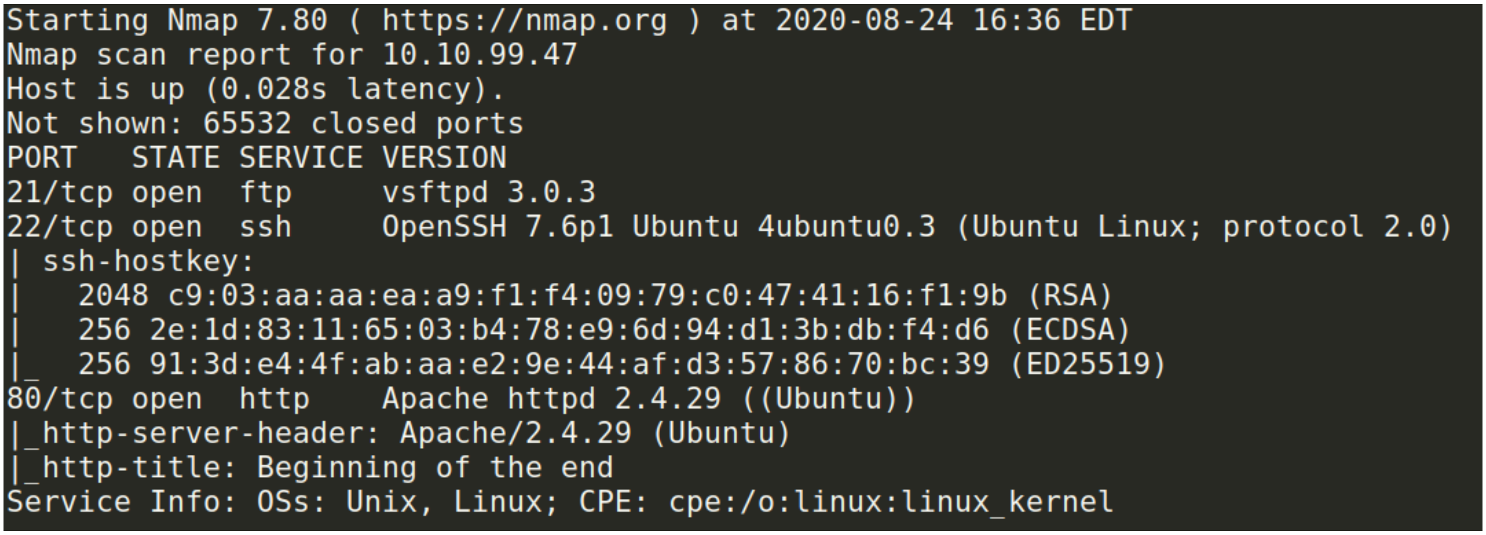
https://tryhackme.com/room/biohazard

Walkthrough

By

https://tryhackme.com/p/iLinxz

NMAP Scan



```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-24 16:36 EDT
Nmap scan report for 10.10.99.47
Host is up (0.028s latency).
Not shown: 65532 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c9:03:aa:aa:ea:a9:f1:f4:09:79:c0:47:41:16:f1:9b (RSA)
|   256 2e:1d:83:11:65:03:b4:78:e9:6d:94:d1:3b:db:f4:d6 (ECDSA)
|_  256 91:3d:e4:4f:ab:aa:e2:9e:44:af:d3:57:86:70:bc:39 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Beginning of the end
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

As we can see, there are a few ports open:

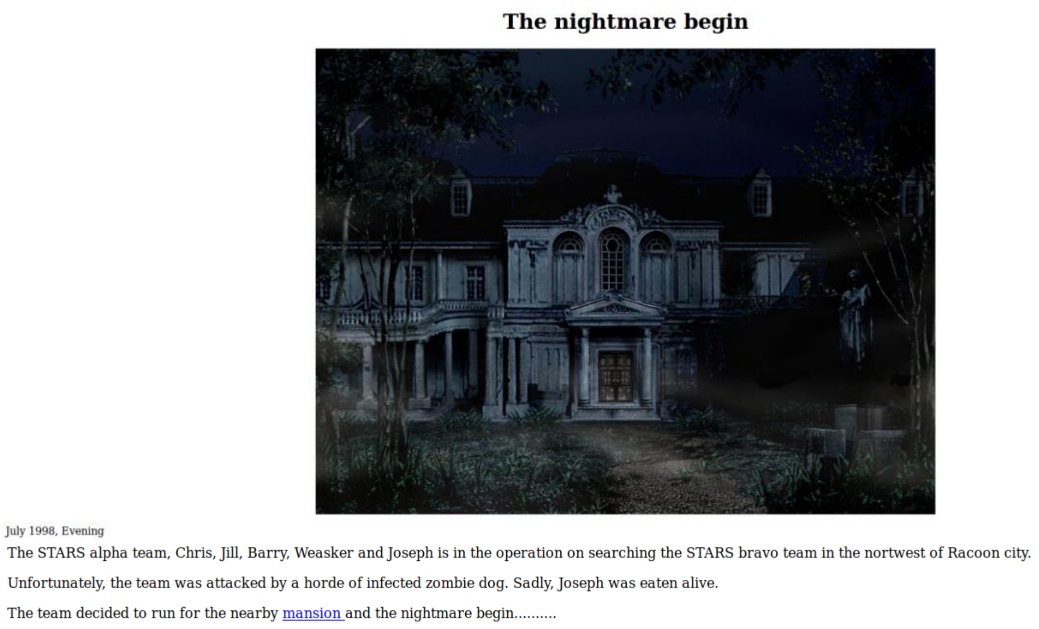
1. Port 21 – running FTP
2. Port 22 – running SSH
3. Port 80 – running HTTP

Great, what can we do?

Checking the searchsploit service to find any vulnerabilities on the services I see, the FTP and SSH, I found no vulnerabilities tied to them.

I then decided to open the victim’s website in my browser.

When entering the website, we are greeted by this page:



The nightmare begin

July 1998, Evening

The STARS alpha team, Chris, Jill, Barry, Weasker and Joseph is in the operation on searching the STARS bravo team in the nortwest of Racoon city.

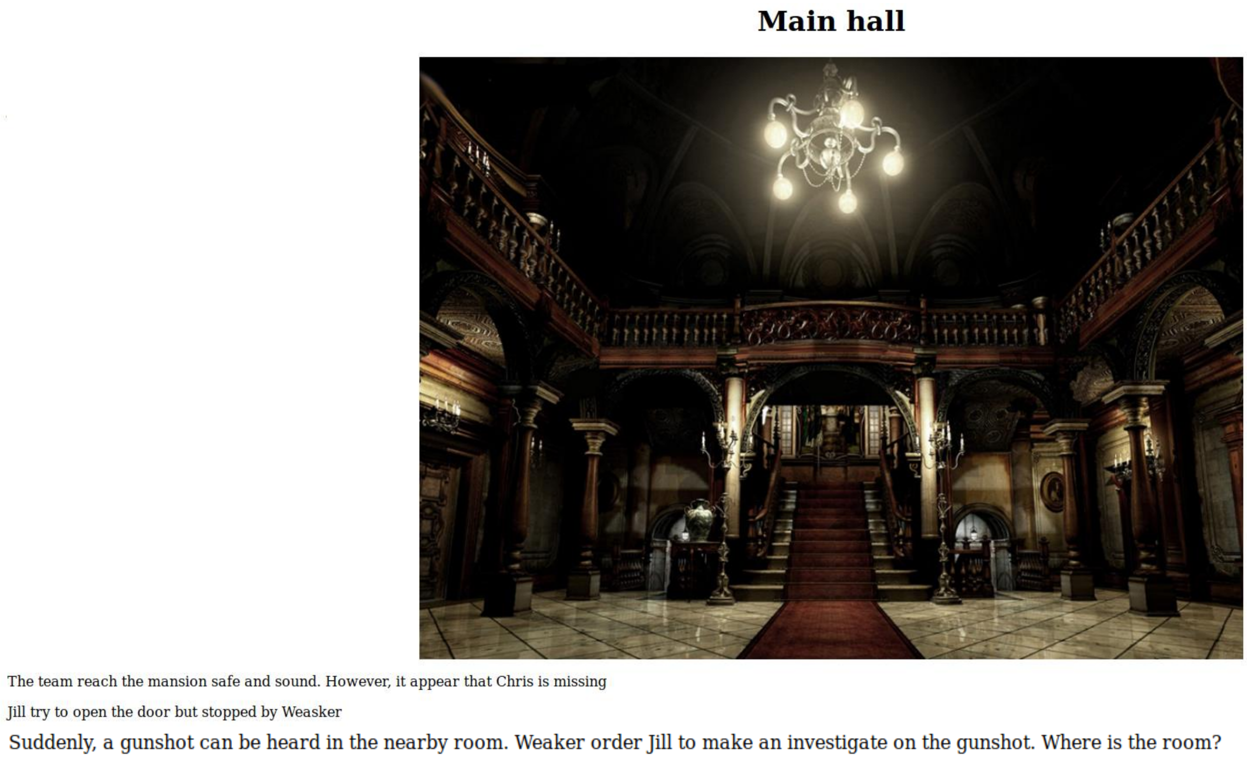
Unfortunately, the team was attacked by a horde of infected zombie dog. Sadly, Joseph was eaten alive.

The team decided to run for the nearby mansion and the nightmare begin..........

I see, I suppose this really is going to be a story-driven challenge. Nice, props to the maker: https://tryhackme.com/p/DesKel

Coming back to the challenge, checking the source code of this page will prove useless as there’s no useful information there.

Following the ‘mansion’ link gets us here:



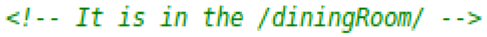
Main hall

The team reach the mansion safe and sound. However, it appear that Chris is missing

Jill try to open the door but stopped by Weasker

Suddenly, a gunshot can be heard in the nearby room. Weaker order Jill to make an investigate on the gunshot. Where is the room?

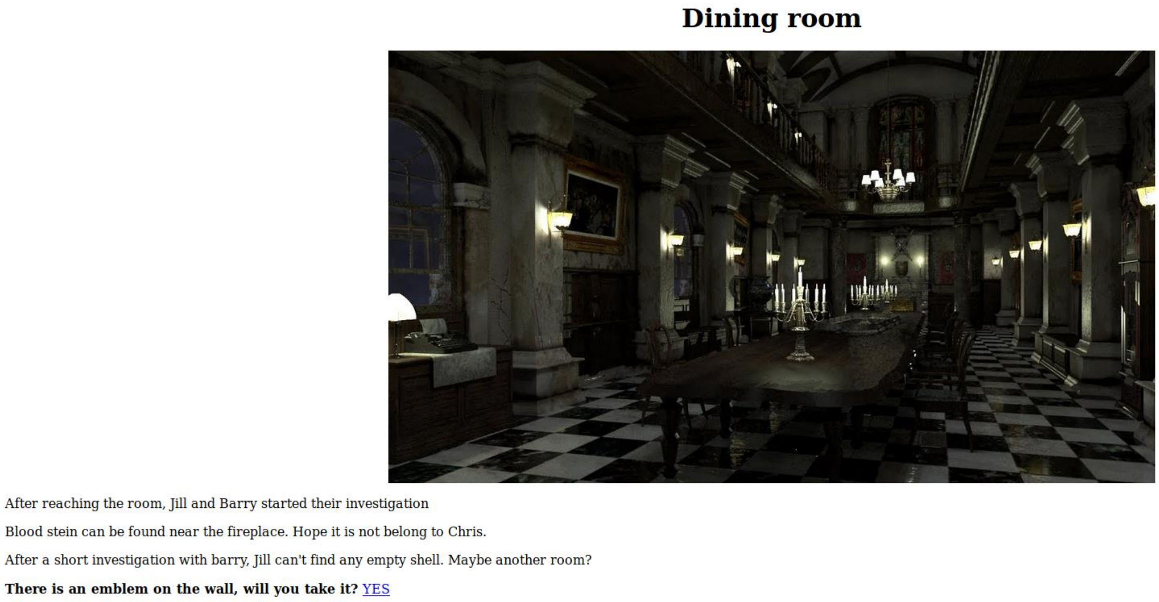
So we must find where the gunshot sound came from… Source code?



```
<!-- It is in the /diningRoom/ -->
```

%Gotcha%

Following this comment, we get to this page:



Dining room

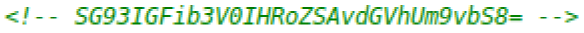
After reaching the room, Jill and Barry started their investigation

Blood stein can be found near the fireplace. Hope it is not belong to Chris.

After a short investigation with barry, Jill can't find any empty shell. Maybe another room?

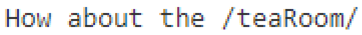
There is an emblem on the wall, will you take it? YES

Before pressing ‘YES’, I checked the source code and spotted this encrypted string.



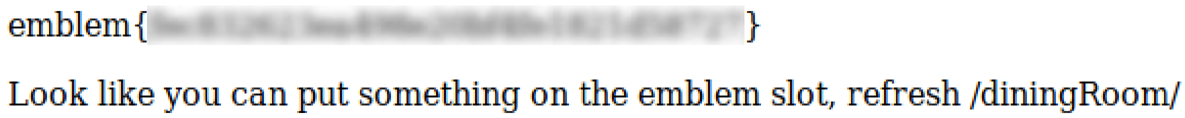
```
<!-- SG93IGFib3V0IHRoZSAvdGVhUm9vbS8= -->
```

It looks like base 64. Let’s decode it! GCHQ’s https://gchq.github.io/CyberChef/ repository is a great decoder and cryptanalysis tool.

Output: 

```
How about the /teaRoom/
```

So I suppose that’s where the gunshot sound came from? Let’s grab the emblem and be on our way.

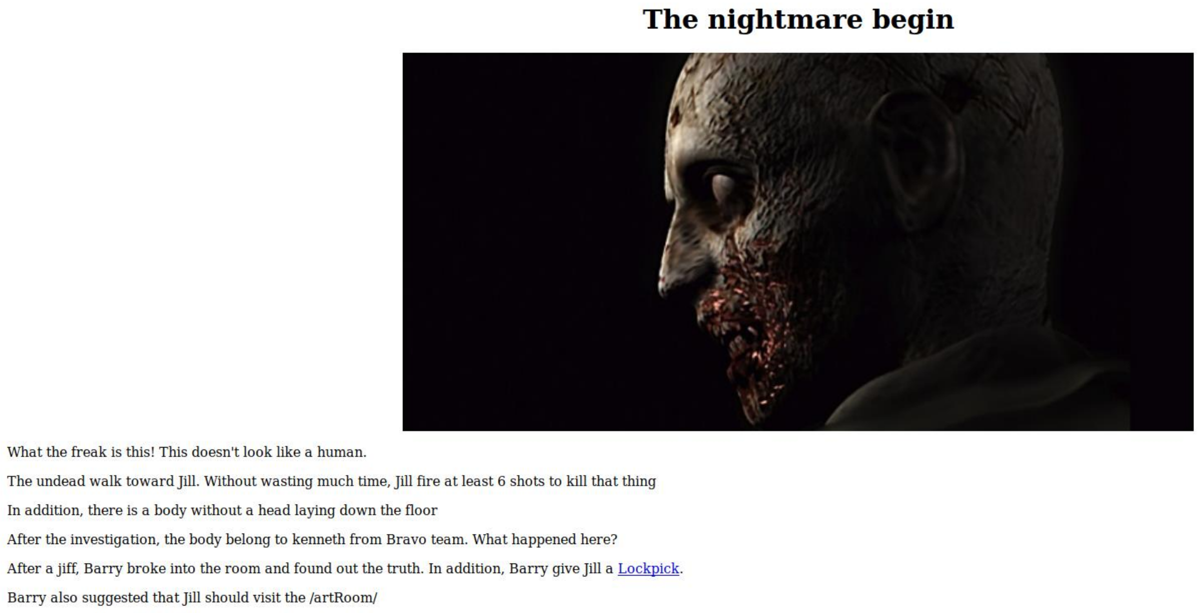


emblem{[redacted]}

Look like you can put something on the emblem slot, refresh /diningRoom/

We received the emblem flag. Let’s move on.

Visiting the /teaRoom, we get to this page:



The nightmare begin

What the freak is this! This doesn't look like a human.

The undead walk toward Jill. Without wasting much time, Jill fire at least 6 shots to kill that thing

In addition, there is a body without a head laying down the floor

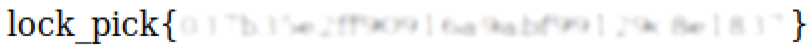
After the investigation, the body belong to kenneth from Bravo team. What happened here?

After a jiff, Barry broke into the room and found out the truth. In addition, Barry give Jill a Lockpick.

Barry also suggested that Jill should visit the /artRoom/

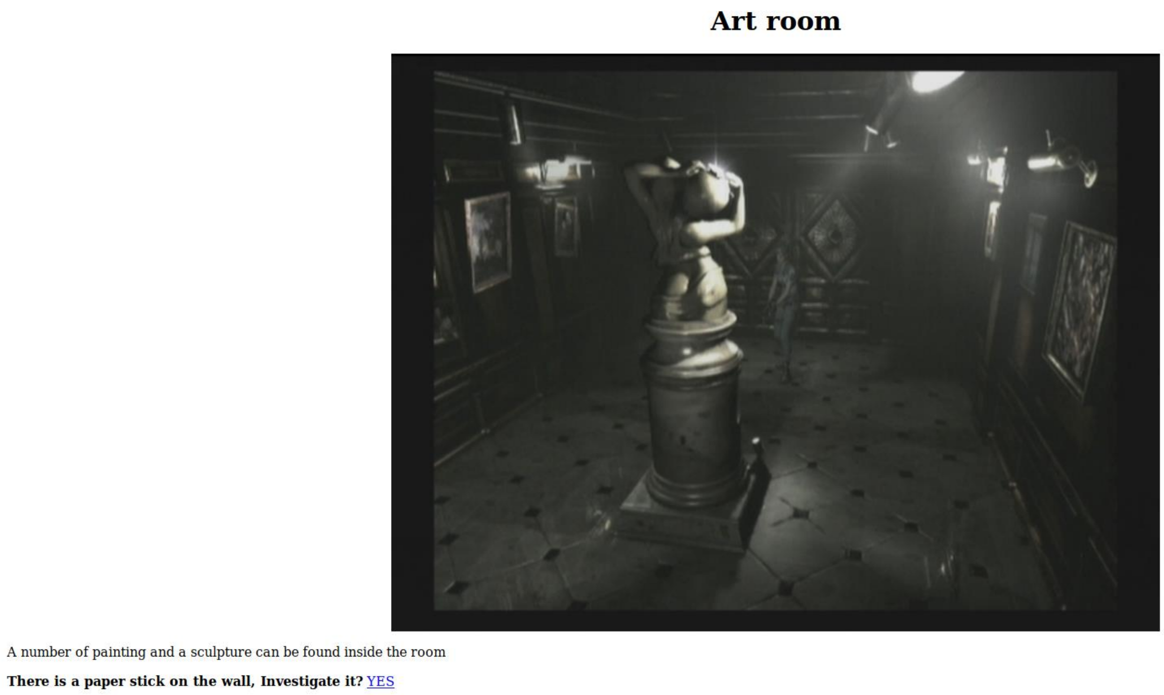
This section suggests we should visit /artRoom/ and we can also pick up a Lockpick. The source code does not provide any more info here.

Let’s pick up the Lockpick and be on our way to /artRoom.



lock_pick{[redacted]}

Visiting /artRoom directs us to this page:



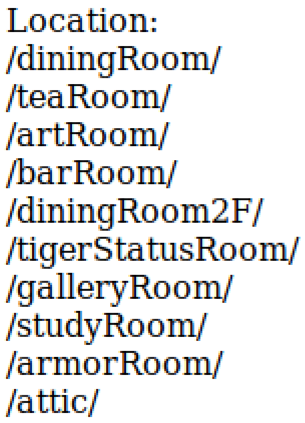
Art room

A number of painting and a sculpture can be found inside the room

There is a paper stick on the wall, Investigate it? YES

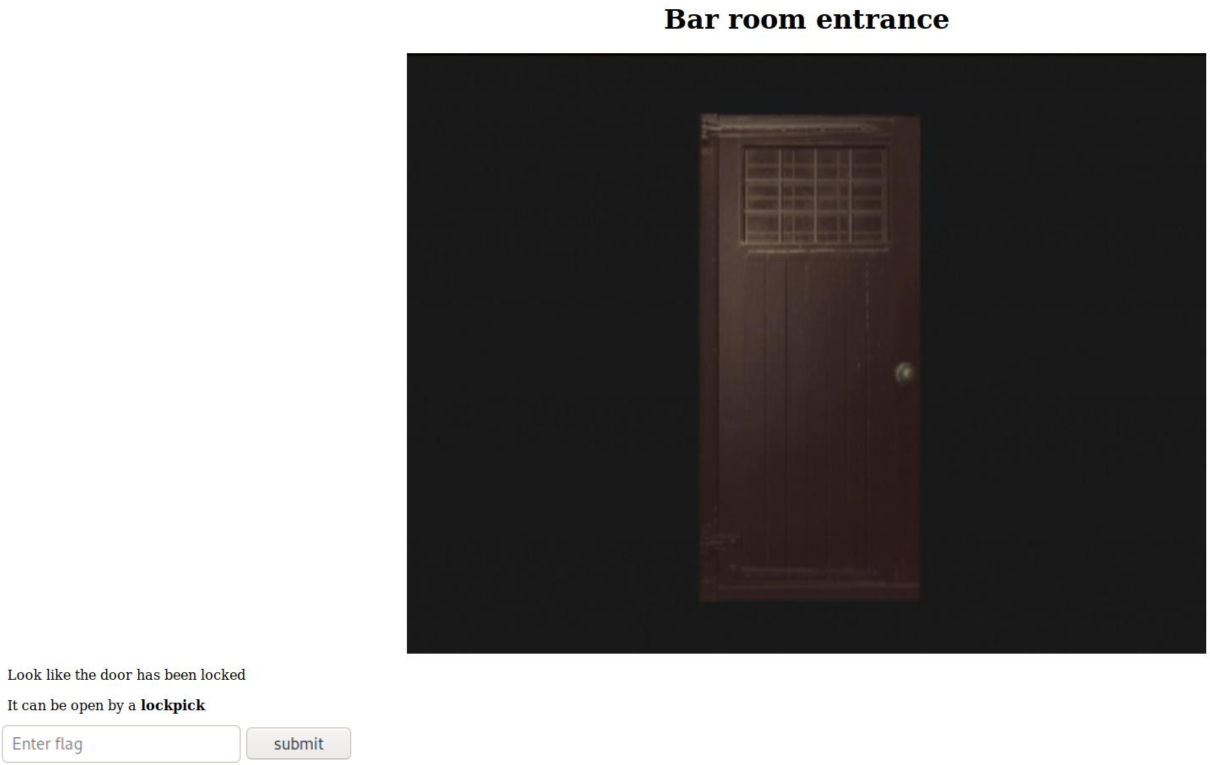
Before checking the paper stick on the wall, I had to check the source code for any hidden messages. But no hidden messages present here either.

Pressing on the ‘YES’ hyperlink shows us a map of the mansion.



Location:
/diningRoom/
/teaRoom/
/artRoom/
/barRoom/
/diningRoom2F/
/tigerStatusRoom/
/galleryRoom/
/studyRoom/
/armorRoom/
/attic/

Okay, so, we’ve already visited /diningRoom/, /teaRoom/ and the /artRoom/. Let’s see how the /barRoom/ is like.



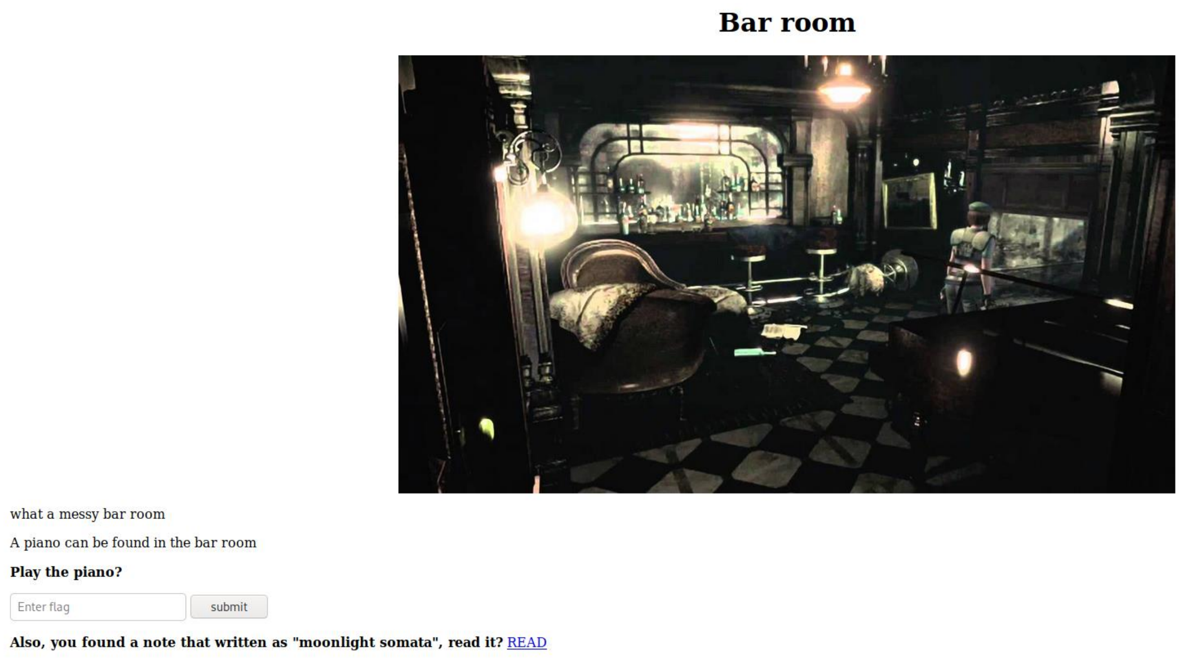
Bar room entrance

Look like the door has been locked

It can be open by a **lockpick**

Before submitting the lockpick flag into the form, I checked the source code but yet again, nothing hidden in there.

Input lockpick{flag}



Bar room

what a messy bar room

A piano can be found in the bar room

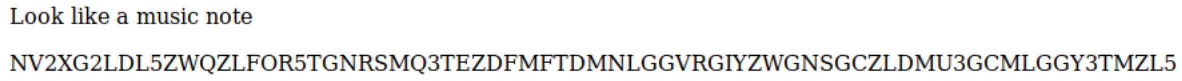
Play the piano?

Enter flag submit

Also, you found a note that written as "moonlight somata", read it? READ

We've arrived at the bar… nothing in the source code again.

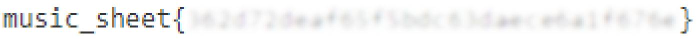
The page wants us to play the piano by using the music sheet flag. Let's read the note I suppose.



Look like a music note

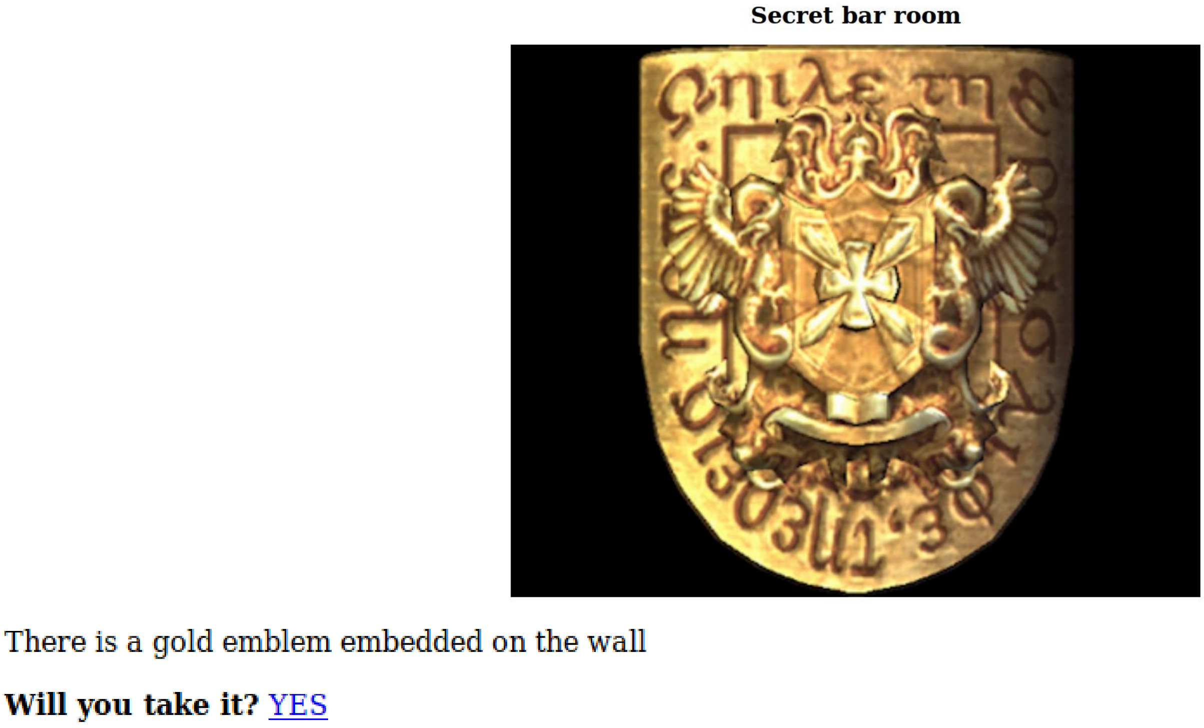
NV2XG2LDL5ZWQZLFOR5TGNRSMQ3TEZDFMFTDMNLGGVRGIYZWGNSGCZLDMU3GCMLGGY3TMZL5

It's an encrypted string, I see. I ran it through a few decoders up on the GCHQ git repository and found out it is encoded using base32. Decoding it gives us this output.



```
music_sheet{[redacted]}
```

Entering this flag into the 'Play the piano?' form, we get this:

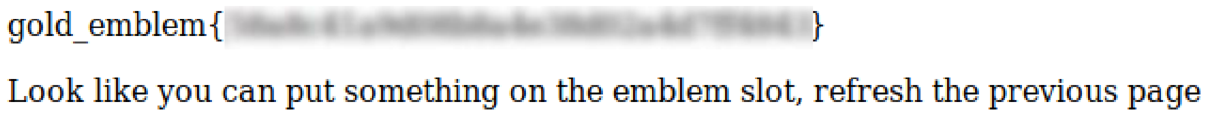


Secret bar room

There is a gold emblem embedded on the wall

Will you take it? YES

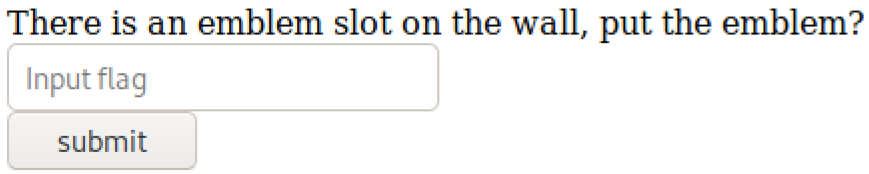
Yes, I will take the emblem, thank you very much.



gold_emblem{[redacted]}

Look like you can put something on the emblem slot, refresh the previous page

We got the gold emblem flag! Refreshing the previous page now changed the page a little bit, instead of the prompt to to take the gold emblem, we have this ‘put an emblem’ prompt.



There is an emblem slot on the wall, put the emblem?

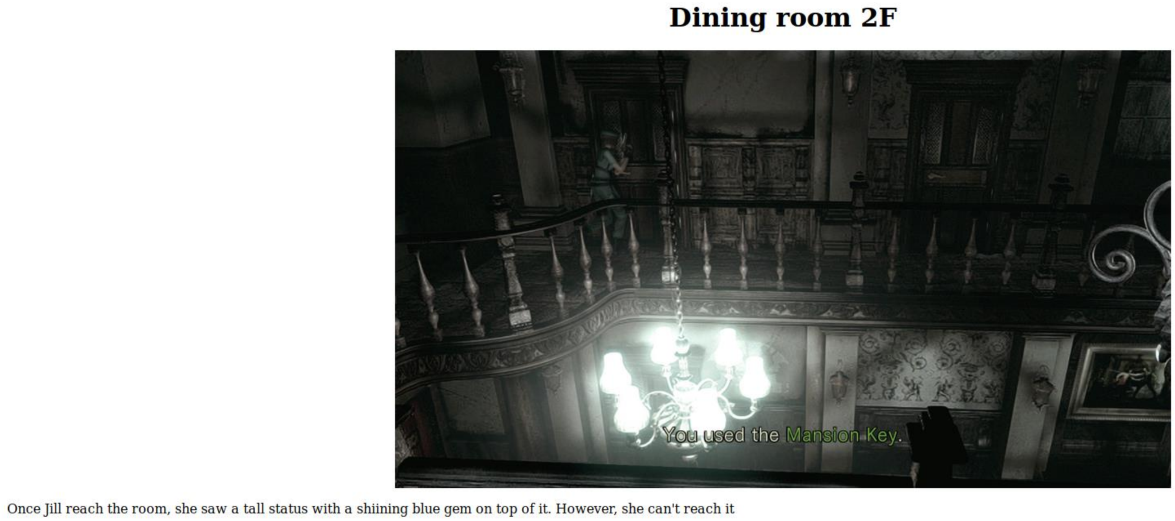
I suppose we will come back to this later as it looks like the beginning of a puzzle.

==

We are done with the /barRoom… for now…

==

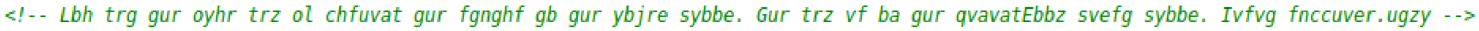
Checking with our map, we should head to the next room, /diningRoom2F/



Dining room 2F

Once Jill reach the room, she saw a tall status with a shiining blue gem on top of it. However, she can't reach it

Looks like there is nothing to do here. Checking the source code however, proved useful this time.

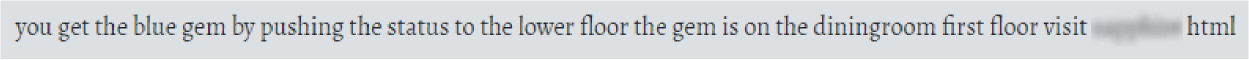


```
<!-- Lbh trg gur oyhr trz ol chfuvat gur fgnghf gb gur ybjre sybbe. Gur trz vf ba gur qvavatEbbz svefg sybbe. Ivfvg fnccuver.ugzy -->
```

Looks like another cryptography mini challenge.

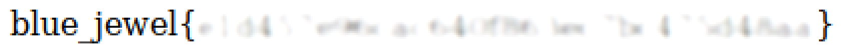
After tinkering with it for a while, I discovered that it the string is encoded using the Vigenere Cipher with the key ‘nnn’.

The string itself reads



you get the blue gem by pushing the status to the lower floor the gem is on the diningroom first floor visit [redacted] html

We visit the URL and receive the blue jewel flag!



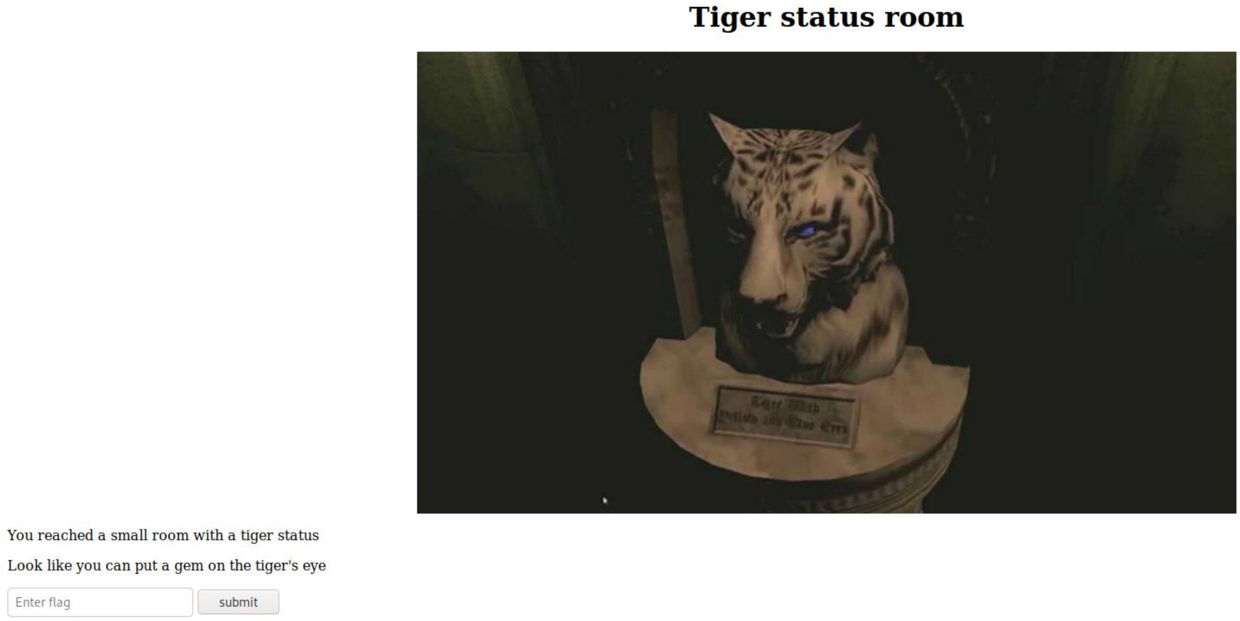
blue_jewel{[redacted]}

==

We're done with the /diningRoom2F/

==

Let's see what we can find in the /tigerStatusRoom/

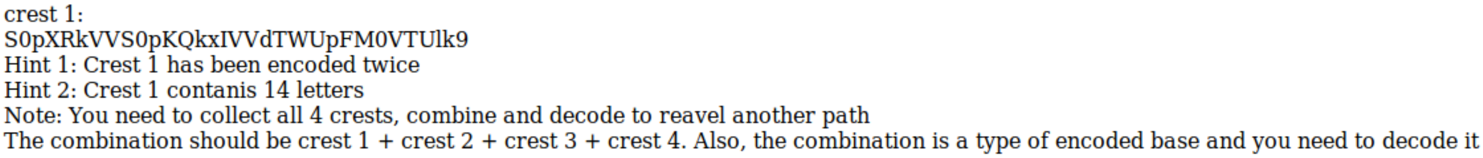


Tiger status room

You reached a small room with a tiger status

Look like you can put a gem on the tiger's eye

The room asks for the gem flag. Let' submit it and see what happens.



crest 1:
S0pXRkVVS0pKQkxIVVdTWUpFM0VTUlk9
Hint 1: Crest 1 has been encoded twice
Hint 2: Crest 1 contanis 14 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it

Hmph… So, there are 4 crests and they all make up a whole message. But they're all encoded on their own. So I have to find them, decode them, put them back to back, decode the whole message and see what the output reads.

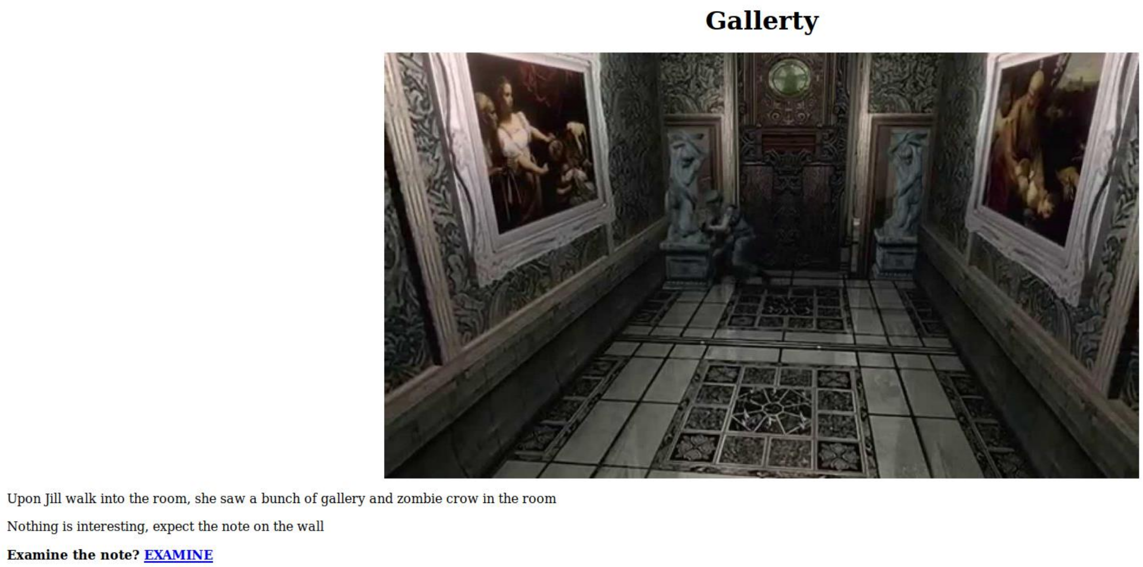
I wil acquire all the crests first and then I will decode the final message.

==

We’re done with the /tigerStatusRoom/

==

Let’s check the /galleryRoom/.



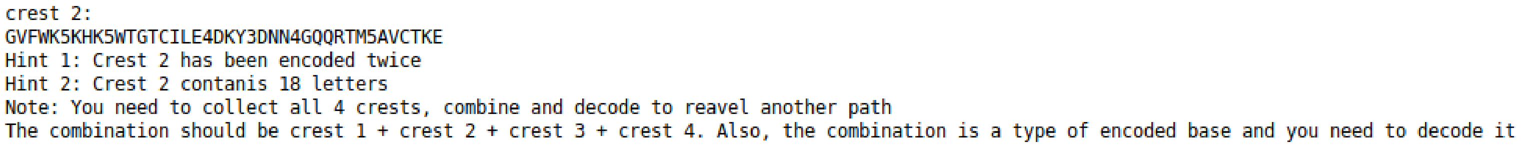
Gallerty

Upon Jill walk into the room, she saw a bunch of gallery and zombie crow in the room

Nothing is interesting, expect the note on the wall

Examine the note? EXAMINE

Examining the not prints out this:



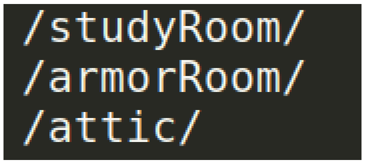
```
crest 2:
GVFWK5KHK5WTGTCILE4DKY3DNN4GQQRTM5AVCTKE
Hint 1: Crest 2 has been encoded twice
Hint 2: Crest 2 contanis 18 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

==

We’re done with the /galleryRoom/

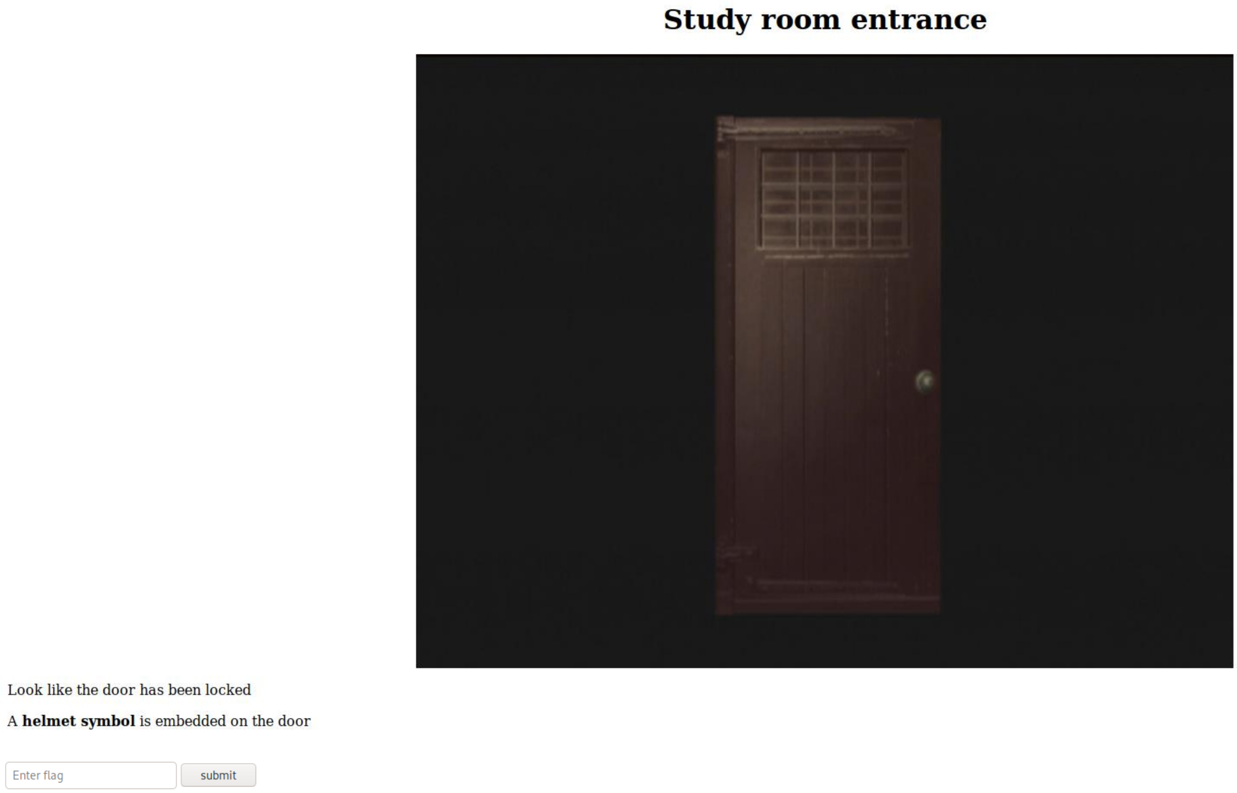
==

The only rooms left to look at are the following



```
/studyRoom/
/armorRoom/
/attic/
```

Let’s visit the /studyRoom/



Study room entrance

Look like the door has been locked

A **helmet symbol** is embedded on the door

The page is asking for a helmet symbol… we don’t have that yet.

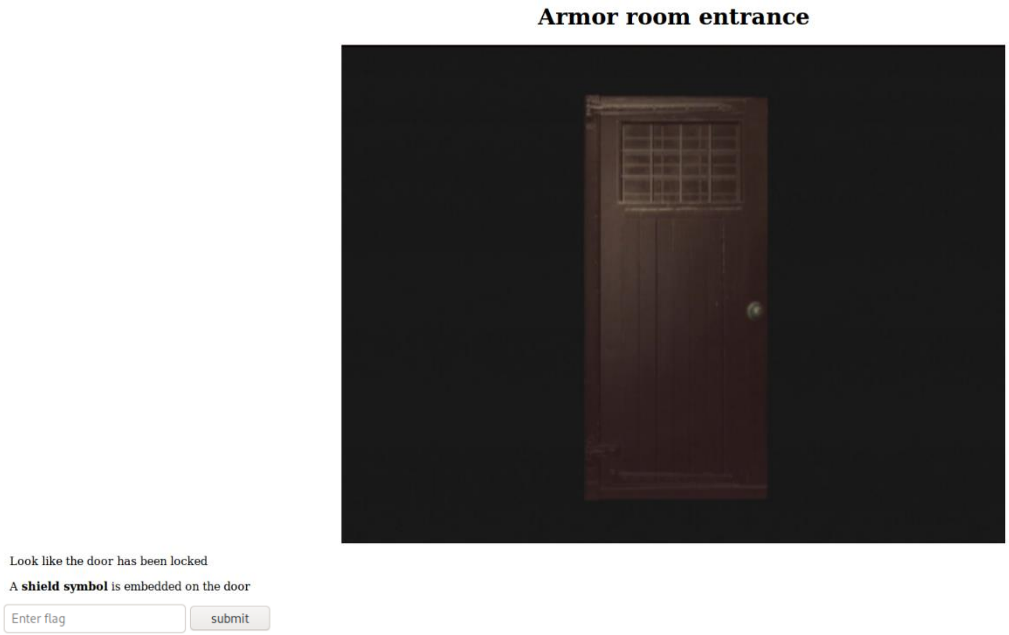
Let’s move on to the /armorRoom/

==

We’re done with the /studyRoom/… for now…

==

Visiting the /armorRoom/ gets us to this page:



Armor room entrance

Look like the door has been locked

A **shield symbol** is embedded on the door

This page asks us for the shield symbol. We don't have it, so let's move forward to the next room.

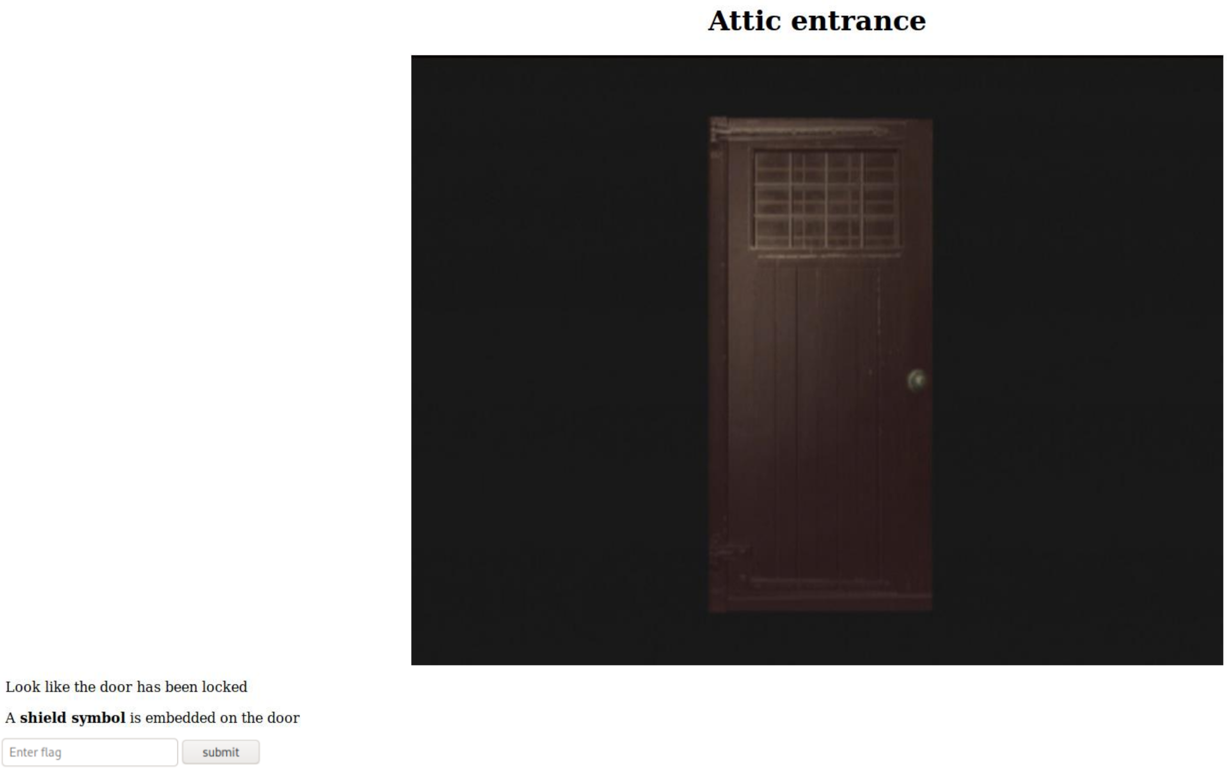
==

We're done with the /armorRoom/… for now…

==

Let's check the last room

The /attic/



Attic entrance

Look like the door has been locked

A **shield symbol** is embedded on the door

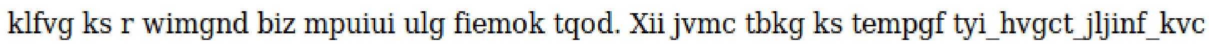
Yet again, a page asking us for the shield flag. We don't have it yet.

==

Let's backtrack a bit…

Going back to the /diningRoom/, we have an empty emblem string we can input. Let's try the gold emblem flag.

That action gives us this encrypted string

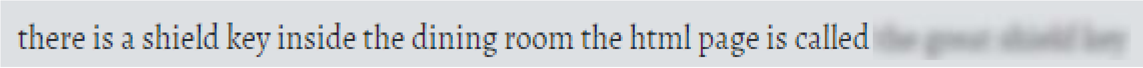


klfvg ks r wimgnd biz mpuiui ulg fiemok tqod. Xii jvmc tbkg ks tempgf tyi_hvgct_jljinf_kvc

After tinkering with it, I found out this is still a Vigenere encoded string with the key 'rebecca'.

https://www.boxentriq.com/code-breaking/vigenere-cipher

So, the note tells us we can find a flag in the dining room by accessing the supposed .html page with that name.

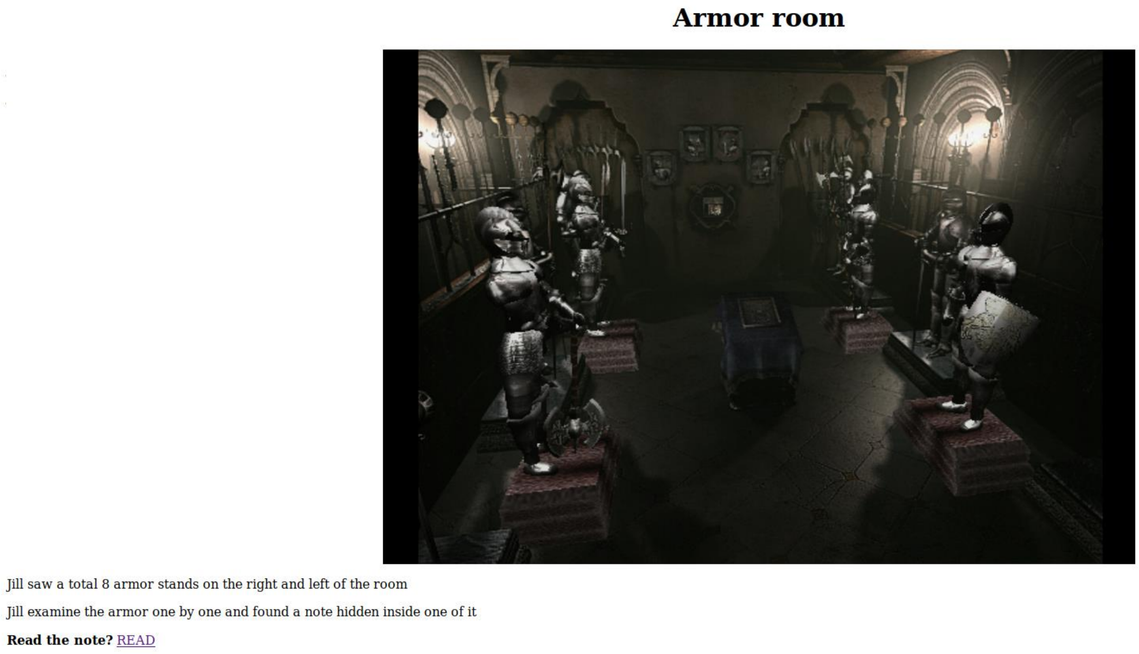


there is a shield key inside the dining room the html page is called [redacted]

We've gotten the shield key flag!

shield_key{[redacted]}

We can now access the /armorRoom/ by inputting the shield_key flag into its form.



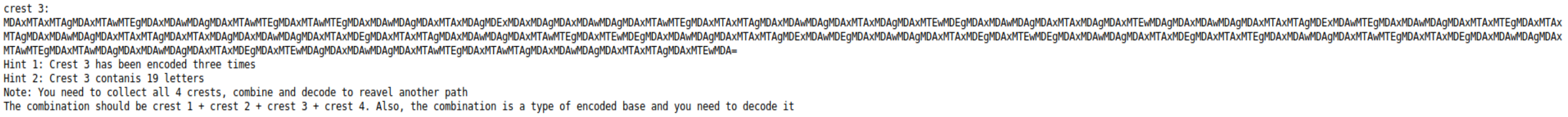
Armor room

Jill saw a total 8 armor stands on the right and left of the room

Jill examine the armor one by one and found a note hidden inside one of it

Read the note? READ

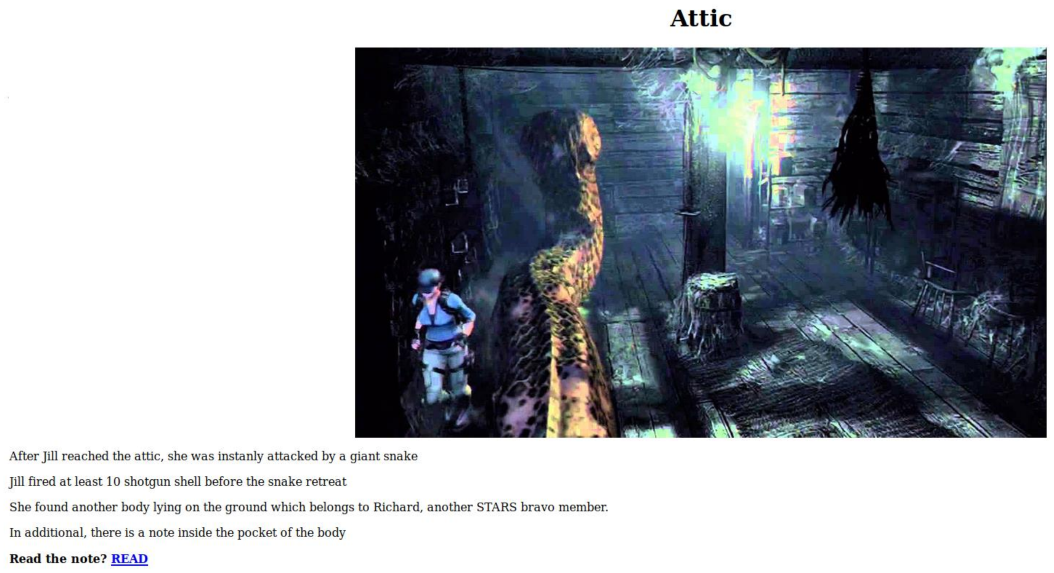
The note reads:



```
crest 3:
MDAxMTAxMTAgMDAxMTAwMTEgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAwMTEgMDAxMDAwMDAgMDAxMTAxMDAgMDExMDAxMDAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAxMTAgMDAxMDAwMDAgMDAxMTAxMDAgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMDAgMDAxMTEwMDAgMDAxMDAwMDAgMDAxMTAxMTAgMDExMDAwMTEgMDAxMDAwMDAgMDAxMTAxMTEgMDAxMTAx
MTAgMDAxMDAwMDAgMDAxMTAxMTAgMDAxMTAxMDAgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTAxMTAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMTAgMDExMDAwMDEgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTAxMTEgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAxMDEgMDAxMDAwMDAgMDAx
MTAwMTEgMDAxMTAwMDAgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTEwMDAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAwMTAgMDAxMDAwMDAgMDAxMTAxMTAgMDAxMTEwMDA=
Hint 1: Crest 3 has been encoded three times
Hint 2: Crest 3 contanis 19 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

What about the /attic/?

Inputting the shield_key into the /attic/ page redirects us to this page



Attic

After Jill reached the attic, she was instanly attacked by a giant snake

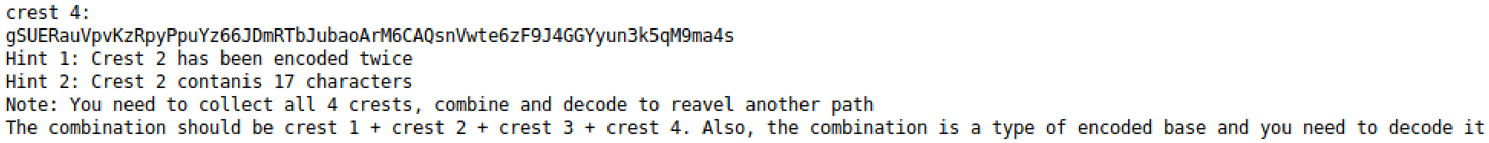
Jill fired at least 10 shotgun shell before the snake retreat

She found another body lying on the ground which belongs to Richard, another STARS bravo member.

In additional, there is a note inside the pocket of the body

Read the note? READ

The source code says nothing important. Let’s read the note.



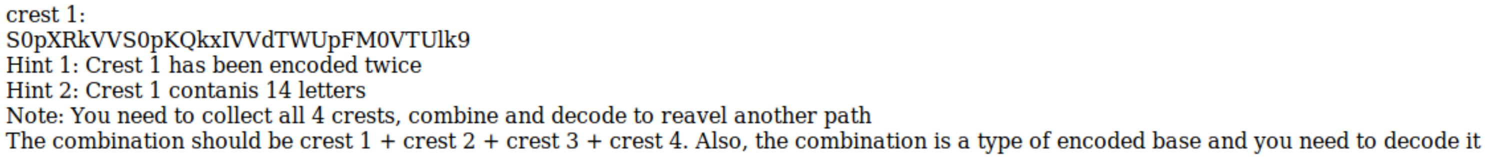
```
crest 4:
gSUERauVpvKzRpyPpuYz66JDmRTbJubaoArM6CAQsnVwte6zF9J4GGYyun3k5qM9ma4s
Hint 1: Crest 2 has been encoded twice
Hint 2: Crest 2 contanis 17 characters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

==

We’re done with /attic/

We have all four crests. Let’s decode them.

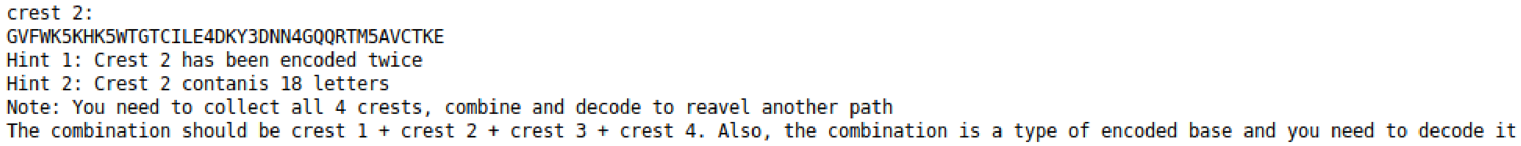
Crest 1:



```
crest 1:
S0pXRkVVS0pKQkxIVVdTWUpFM0VTUlk9
Hint 1: Crest 1 has been encoded twice
Hint 2: Crest 1 contanis 14 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

It’s been encoded twice and made of 14 letters. Tinkering with it on the GCHQ git repo, I’ve cracked it by first decoding it from base64 then base32.

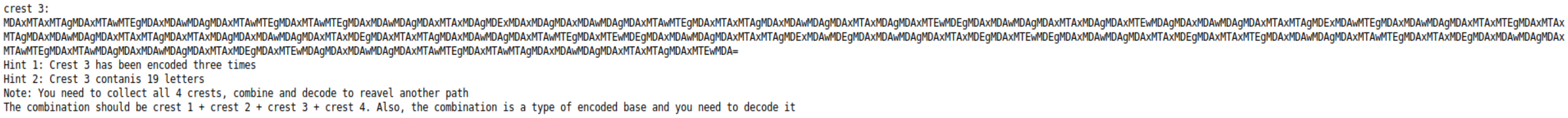
Crest 2:



```
crest 2:
GVFWK5KHK5WTGTCILE4DKY3DNN4GQQRTM5AVCTKE
Hint 1: Crest 2 has been encoded twice
Hint 2: Crest 2 contanis 18 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

This crest I decoded by first decoding it through base32 then base58.

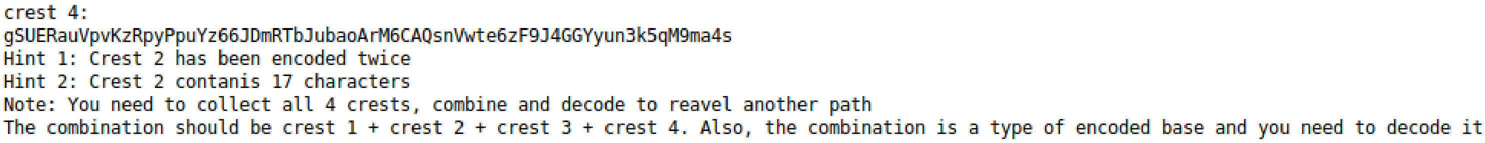
Crest 3:



```
crest 3:
MDAxMTAxMTAgMDAxMTAwMTEgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAwMTEgMDAxMDAwMDAgMDAxMTAxMDAgMDExMDAxMDAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAxMTAgMDAxMDAwMDAgMDAxMTAxMDAgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMDAgMDAxMTEwMDAgMDAxMDAwMDAgMDAxMTAxMTAgMDExMDAwMTEgMDAxMDAwMDAgMDAxMTAxMTEgMDAxMTAx
MTAgMDAxMDAwMDAgMDAxMTAxMTAgMDAxMTAxMDAgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTAxMTAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMTAgMDExMDAwMDEgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTEwMDEgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTAxMTEgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAxMDEgMDAxMDAwMDAgMDAx
MTAwMTEgMDAxMTAwMDAgMDAxMDAwMDAgMDAxMTAxMDEgMDAxMTEwMDAgMDAxMDAwMDAgMDAxMTAwMTEgMDAxMTAwMTAgMDAxMDAwMDAgMDAxMTAxMTAgMDAxMTEwMDA=
Hint 1: Crest 3 has been encoded three times
Hint 2: Crest 3 contanis 19 letters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

I first decoded with base64, then with binary, then with hex.

Crest 4:



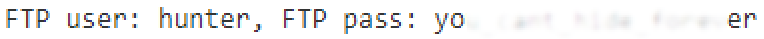
```
crest 4:
gSUERauVpvKzRpyPpuYz66JDmRTbJubaoArM6CAQsnVwte6zF9J4GGYyun3k5qM9ma4s
Hint 1: Crest 2 has been encoded twice
Hint 2: Crest 2 contanis 17 characters
Note: You need to collect all 4 crests, combine and decode to reavel another path
The combination should be crest 1 + crest 2 + crest 3 + crest 4. Also, the combination is a type of encoded base and you need to decode it
```

I first decoded this with base58 and then hex.

After decoding all the messages, I put them back to back:

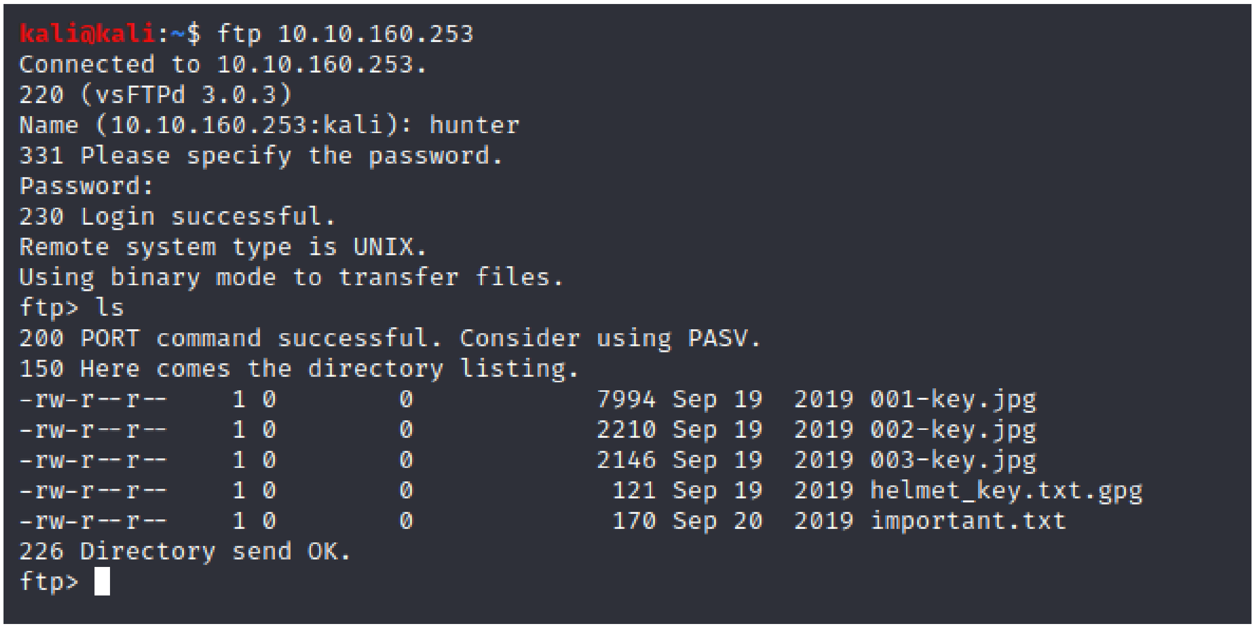
[redacted]

Decoding this message with base 64 gives us the first set of credentials:



```
FTP user: hunter, FTP pass: yo[redacted]er
```

Finally! Let's log into the FTP service.

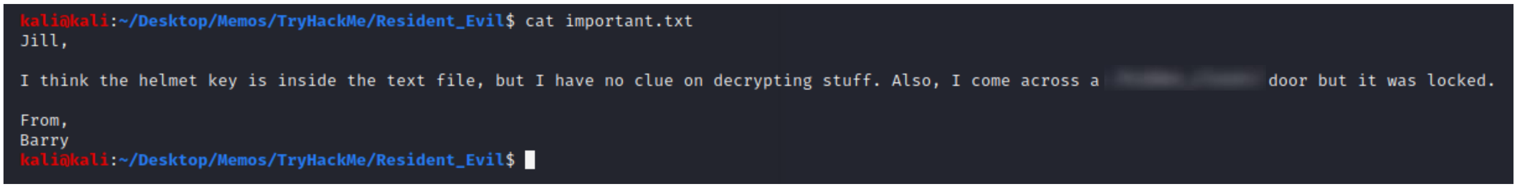


```
kali@kali:~$ ftp 10.10.160.253
Connected to 10.10.160.253.
220 (vsFTPd 3.0.3)
Name (10.10.160.253:kali): hunter
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0            7994 Sep 19  2019 001-key.jpg
-rw-r--r--    1 0        0            2210 Sep 19  2019 002-key.jpg
-rw-r--r--    1 0        0            2146 Sep 19  2019 003-key.jpg
-rw-r--r--    1 0        0             121 Sep 19  2019 helmet_key.txt.gpg
-rw-r--r--    1 0        0             170 Sep 20  2019 important.txt
226 Directory send OK.
ftp> 
```

Let’s download all the files!

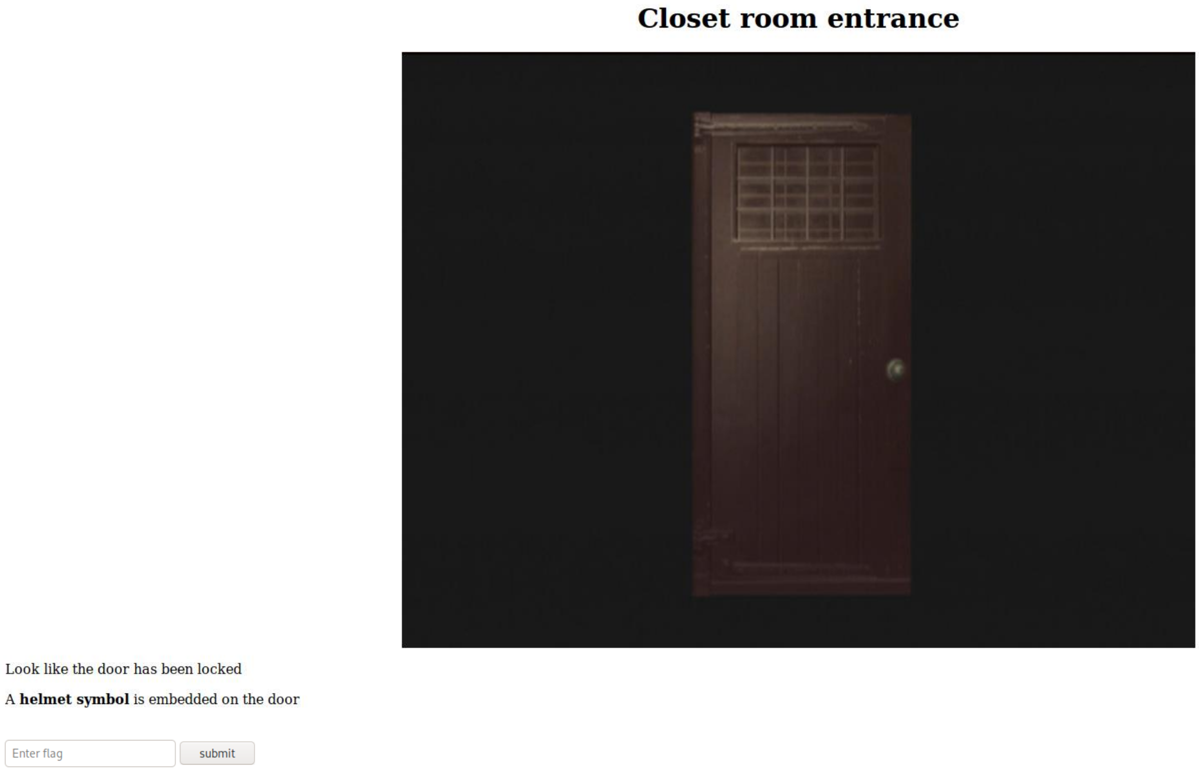
We have three .jpg files, a .gpg encrypted file and a .txt file. I smell steganography.

The .txt file reads:



```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ cat important.txt
Jill,

I think the helmet key is inside the text file, but I have no clue on decrypting stuff. Also, I come across a [redacted] door but it was locked.

From,
Barry
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ 
```

Accessing the URL hinted at in the .txt file brings us here:



Closet room entrance

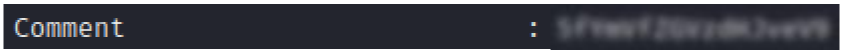
Look like the door has been locked

A **helmet symbol** is embedded on the door

This page is asking us for the helmet flag. We don’t have it yet thus let’s go back to the pictures and .gpg file.

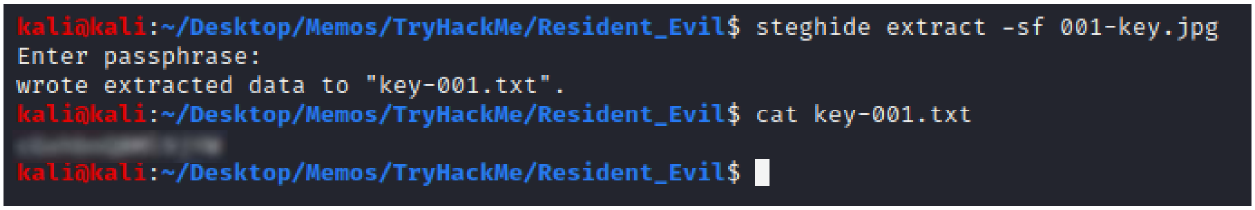
I started to use several steganography tools on the pictures.

I tried exiftool on all three keys. I found a comment on the second key.



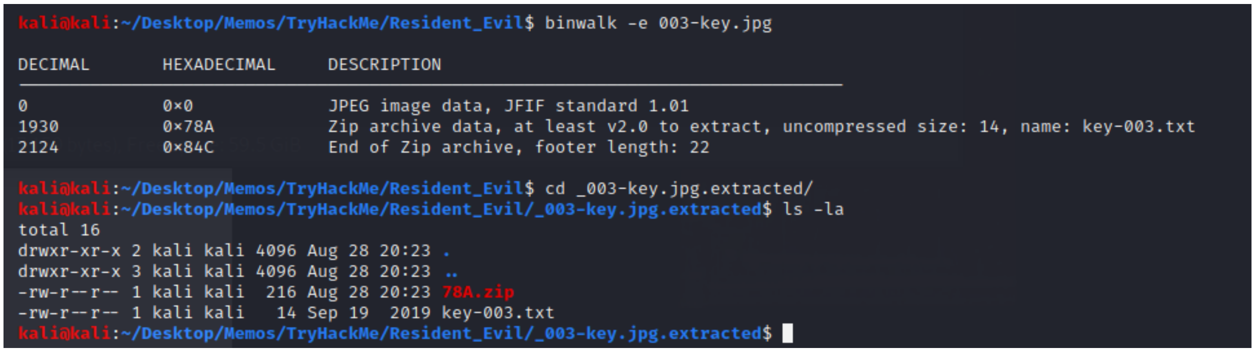
```
Comment                         : [redacted]
```

I tried to extract any hidden info with steghide (no passphrase) and found a .txt file was embedded into the first key. It reads:



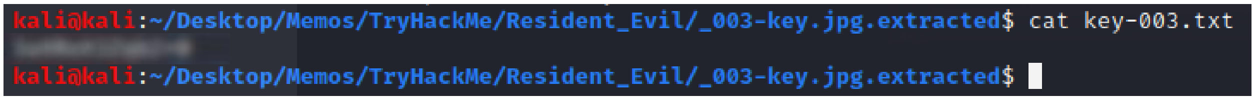
```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ steghide extract -sf 001-key.jpg
Enter passphrase:
wrote extracted data to "key-001.txt".
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ cat key-001.txt
[redacted]
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ 
```

In the end, I used binwalk to check if there were any .zip files embedded in the any of the pictures. And unsurprisingly, I found a .zip file embedded into the third key.



```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ binwalk -e 003-key.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
1930          0x78A           Zip archive data, at least v2.0 to extract, uncompressed size: 14, name: key-003.txt
2124          0x84C           End of Zip archive, footer length: 22

kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ cd _003-key.jpg.extracted/
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil/_003-key.jpg.extracted$ ls -la
total 16
drwxr-xr-x 2 kali kali 4096 Aug 28 20:23 .
drwxr-xr-x 3 kali kali 4096 Aug 28 20:23 ..
-rw-r--r-- 1 kali kali  216 Aug 28 20:23 78A.zip
-rw-r--r-- 1 kali kali   14 Sep 19  2019 key-003.txt
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil/_003-key.jpg.extracted$ 
```

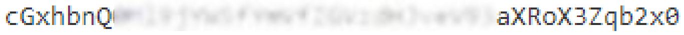
In the .zip file embedded in the image, we find another .zip file and a .txt file. The .txt file reads:



```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil/_003-key.jpg.extracted$ cat key-003.txt
[redacted]
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil/_003-key.jpg.extracted$ 
```

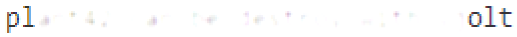
Unzipping the '78A.zip' will make me realize that within it, there is another key-003.txt, just in case I lose the one I've just been given.

Let's put the keys back to back and decode the bigger message!



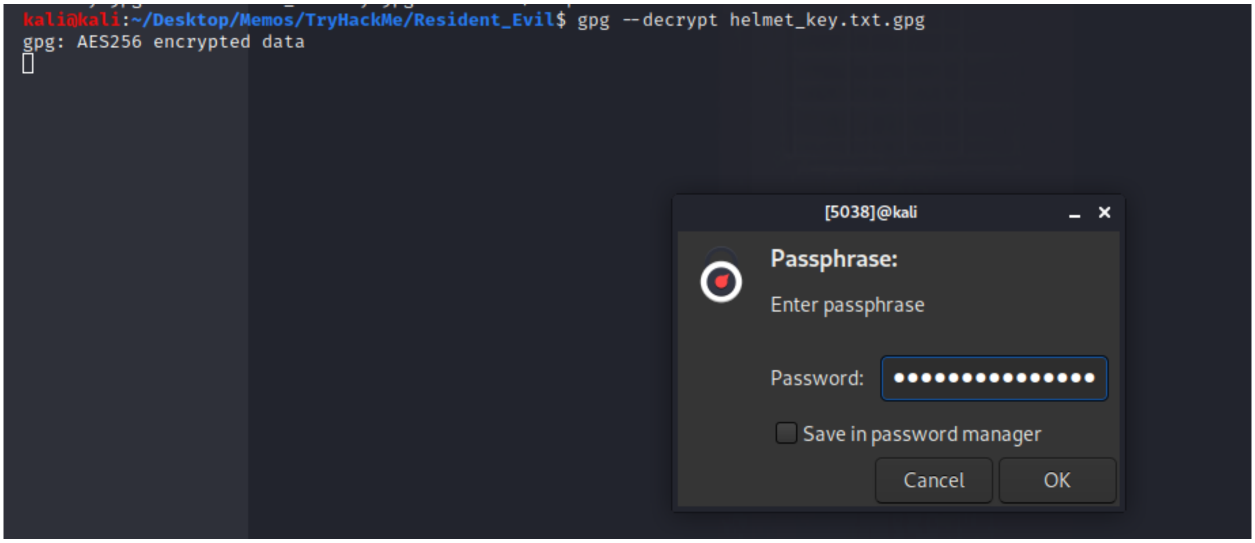
```
cGxhbnQ[redacted]aXRoX3Zqb2x0
```

Decoding the string with base64 gives us this output.

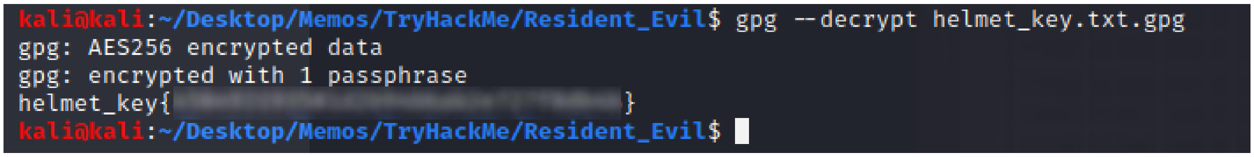


```
pl[redacted]olt
```

We're ready to get the helmet key now. We have the key that will allow us to decrypt the .gpg file now.

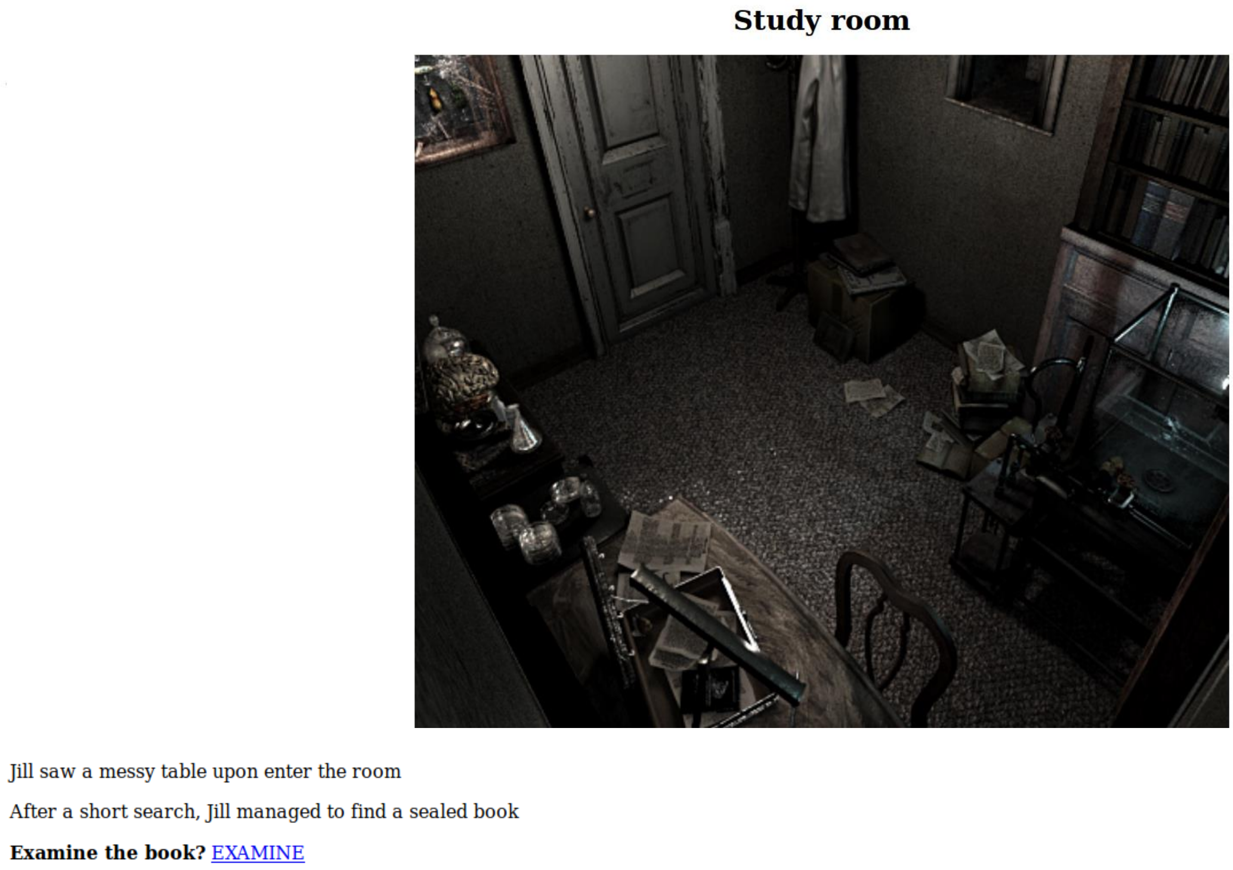


```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ gpg --decrypt helmet_key.txt.gpg
gpg: AES256 encrypted data
```



```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ gpg --decrypt helmet_key.txt.gpg
gpg: AES256 encrypted data
gpg: encrypted with 1 passphrase
helmet_key{[redacted]}
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ 
```

We found the helmet flag! Let's access the /studyRoom/ and input the helmet flag.



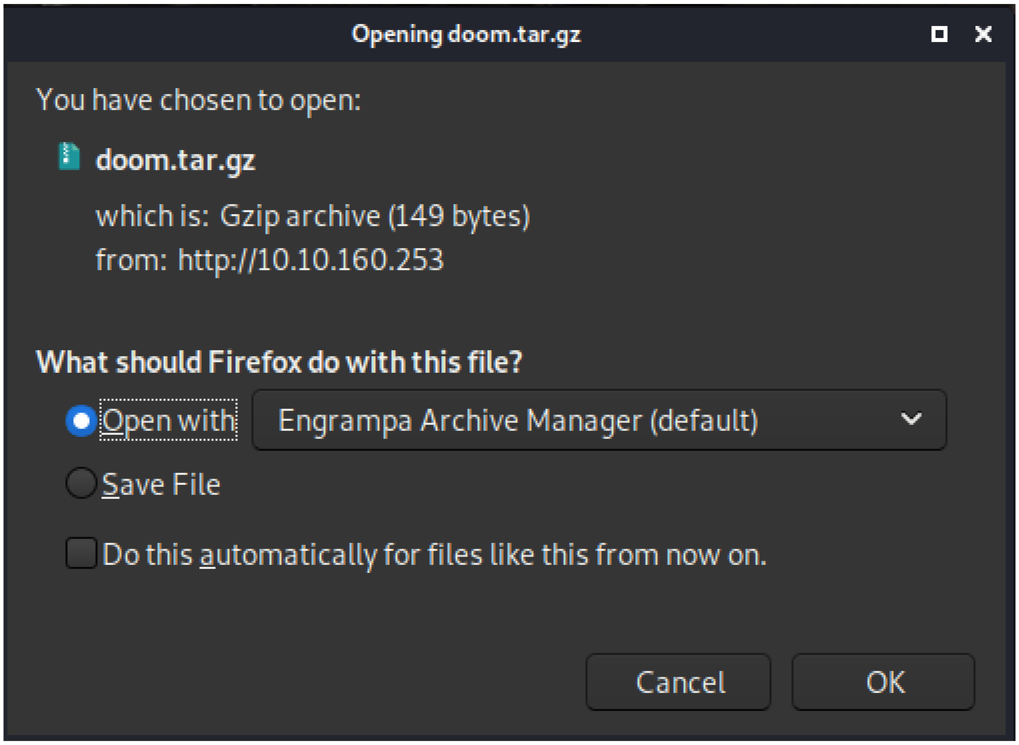
Study room

Jill saw a messy table upon enter the room

After a short search, Jill managed to find a sealed book

Examine the book? EXAMINE

Let's examine the book.



Opening doom.tar.gz

You have chosen to open:

doom.tar.gz

which is: Gzip archive (149 bytes)

from: http://10.10.160.253

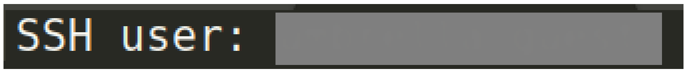
What should Firefox do with this file?

● Open with Engrampa Archive Manager (default)

○ Save File

☐ Do this automatically for files like this from now on.

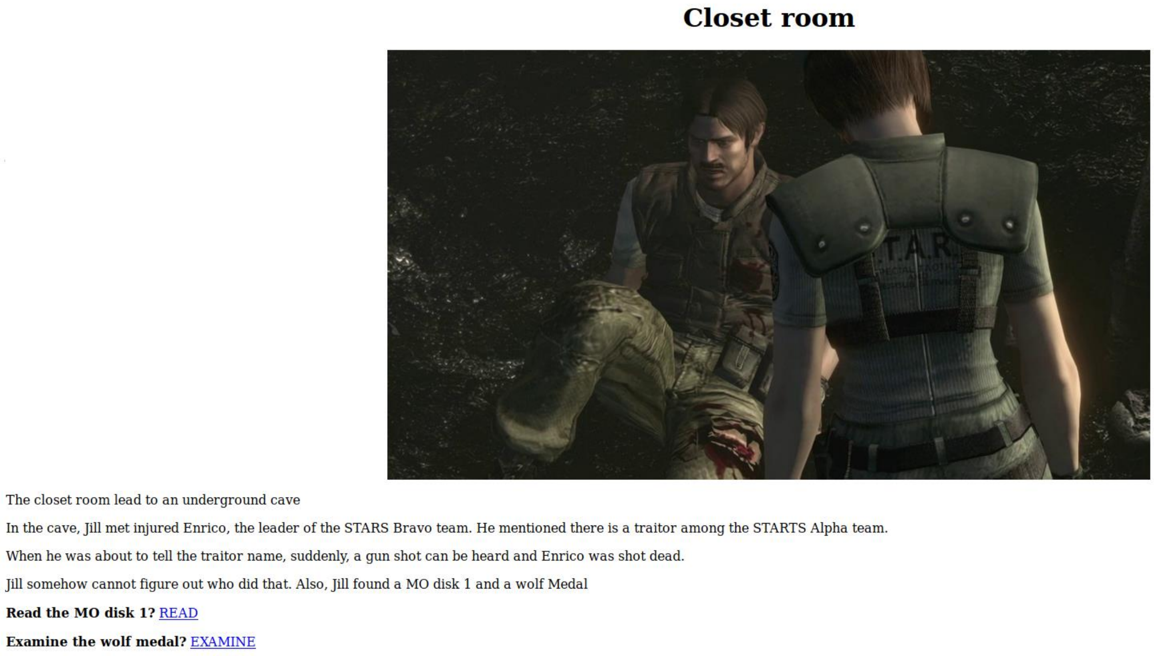
Trying to examine the book makes the browser try to download this .tar.gz file. Decrypting this file will write a .txt file called ‘eagle_medal.txt’



SSH user: [redacted]

We find an SSH user username.

Let’s check the Closed door entrance since we have the helmet flag.



Closet room

The closet room lead to an underground cave

In the cave, Jill met injured Enrico, the leader of the STARS Bravo team. He mentioned there is a traitor among the STARTS Alpha team.

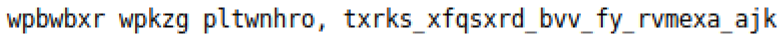
When he was about to tell the traitor name, suddenly, a gun shot can be heard and Enrico was shot dead.

Jill somehow cannot figure out who did that. Also, Jill found a MO disk 1 and a wolf Medal

Read the MO disk 1? READ

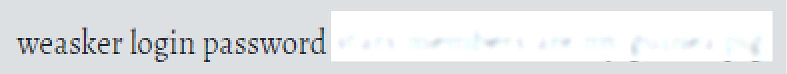
Examine the wolf medal? EXAMINE

Reading the first MO disk outputs this:



```
wpbwbxr wpkzg pltwnhro, txrks_xfqsxrd_bvv_fy_rvmexa_ajk
```

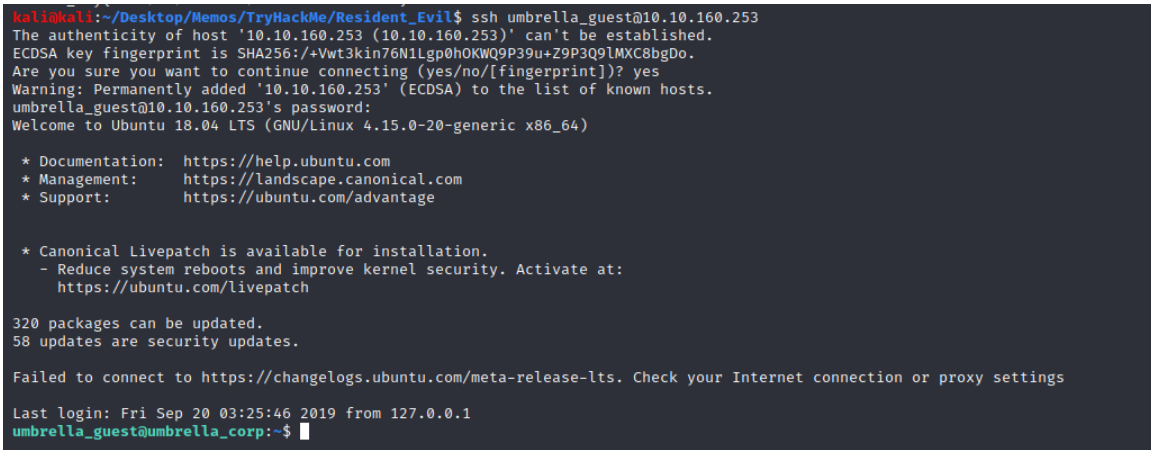
This encryption looks like Vigenere. Let’s crack it.



weasker login password [redacted]

Examining the wolf medal gives us the SSH password for the SSH user we just found.

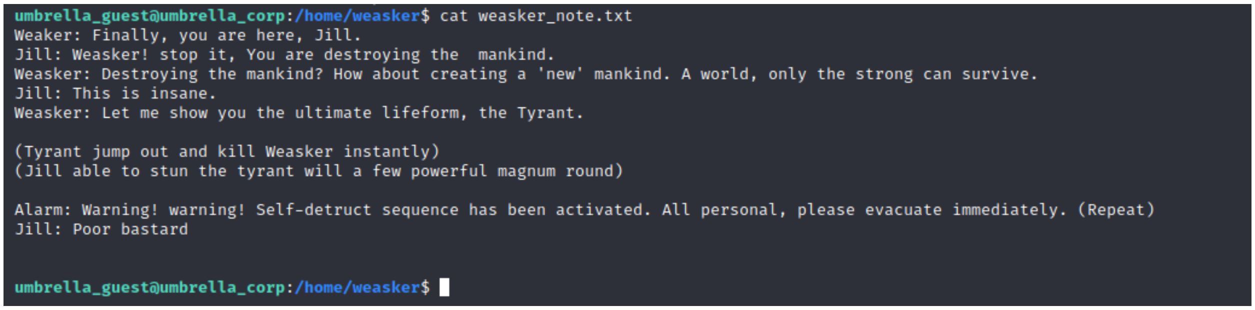
Let’s try to SSH over.



```
kali@kali:~/Desktop/Memos/TryHackMe/Resident_Evil$ ssh umbrella_guest@10.10.160.253
The authenticity of host '10.10.160.253 (10.10.160.253)' can't be established.
ECDSA key fingerprint is SHA256:/+Vwt3kin76N1Lgp0hOKWQ9P39u+Z9P3Q9lMXC8bgDo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.160.253' (ECDSA) to the list of known hosts.
umbrella_guest@10.10.160.253's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 4.15.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

320 packages can be updated.
58 updates are security updates.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Sep 20 03:25:46 2019 from 127.0.0.1
umbrella_guest@umbrella_corp:~$ 
```

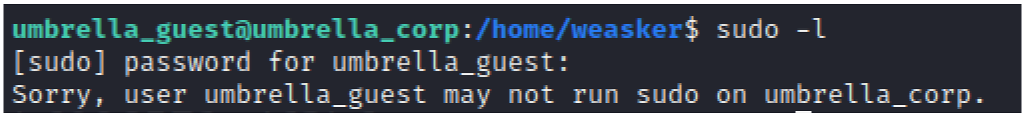
[Hacker Voice] I'm in.

Let's look for interesting files. I found this .txt file inside weasker's home directory.



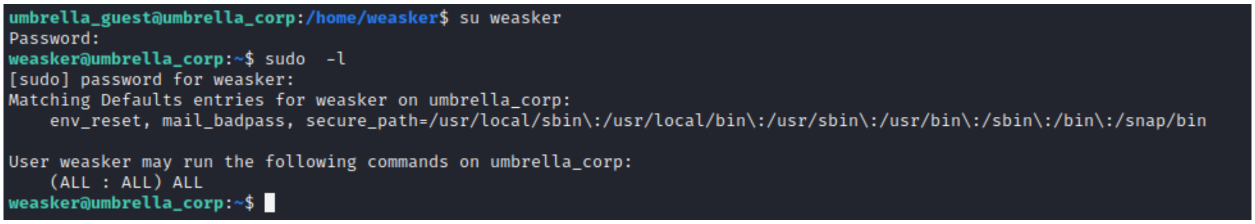
```
umbrella_guest@umbrella_corp:/home/weasker$ cat weasker_note.txt
Weaker: Finally, you are here, Jill.
Jill: Weasker! stop it, You are destroying the  mankind.
Weasker: Destroying the mankind? How about creating a 'new' mankind. A world, only the strong can survive.
Jill: This is insane.
Weasker: Let me show you the ultimate lifeform, the Tyrant.

(Tyrant jump out and kill Weasker instantly)
(Jill able to stun the tyrant will a few powerful magnum round)

Alarm: Warning! warning! Self-detruct sequence has been activated. All personal, please evacuate immediately. (Repeat)
Jill: Poor bastard


umbrella_guest@umbrella_corp:/home/weasker$
```

sudo -l?



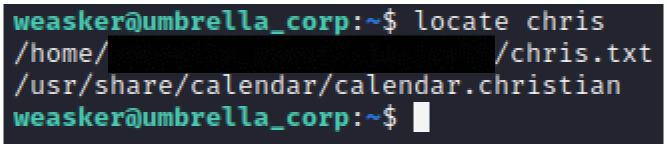
```
umbrella_guest@umbrella_corp:/home/weasker$ sudo -l
[sudo] password for umbrella_guest:
Sorry, user umbrella_guest may not run sudo on umbrella_corp.
```

Well, guess we'll have to su to weasker since we already know his password.



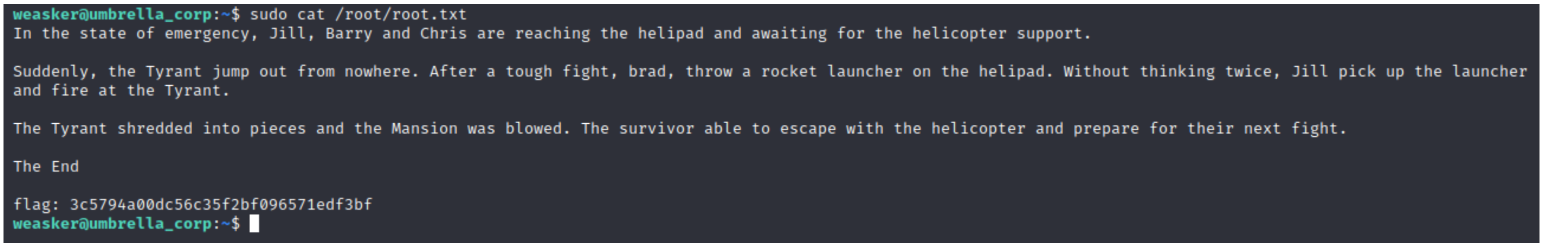
```
umbrella_guest@umbrella_corp:/home/weasker$ su weasker
Password:
weasker@umbrella_corp:~$ sudo  -l
[sudo] password for weasker:
Matching Defaults entries for weasker on umbrella_corp:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User weasker may run the following commands on umbrella_corp:
    (ALL : ALL) ALL
weasker@umbrella_corp:~$
```

Weasker can run any command as root. But before that, let's find Chris!



```
weasker@umbrella_corp:~$ locate chris
/home/[redacted]/chris.txt
/usr/share/calendar/calendar.christian
weasker@umbrella_corp:~$
```

And finally, let's get the root flag!



```
weasker@umbrella_corp:~$ sudo cat /root/root.txt
In the state of emergency, Jill, Barry and Chris are reaching the helipad and awaiting for the helicopter support.

Suddenly, the Tyrant jump out from nowhere. After a tough fight, brad, throw a rocket launcher on the helipad. Without thinking twice, Jill pick up the launcher
and fire at the Tyrant.

The Tyrant shredded into pieces and the Mansion was blowed. The survivor able to escape with the helicopter and prepare for their next fight.

The End

flag: 3c5794a00dc56c35f2bf096571edf3bf
weasker@umbrella_corp:~$
```

==

END