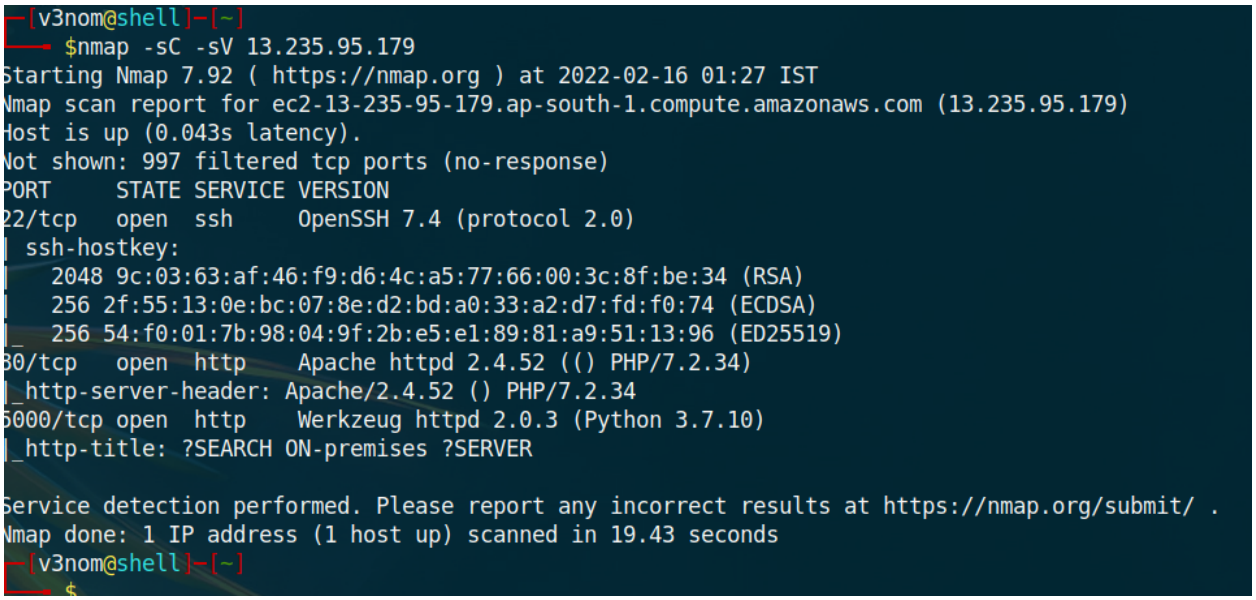
CTF Challenge-Cybersecurity-EWYL 2022

The challenge starts with a url : http://13.235.95.179/

NMAP:

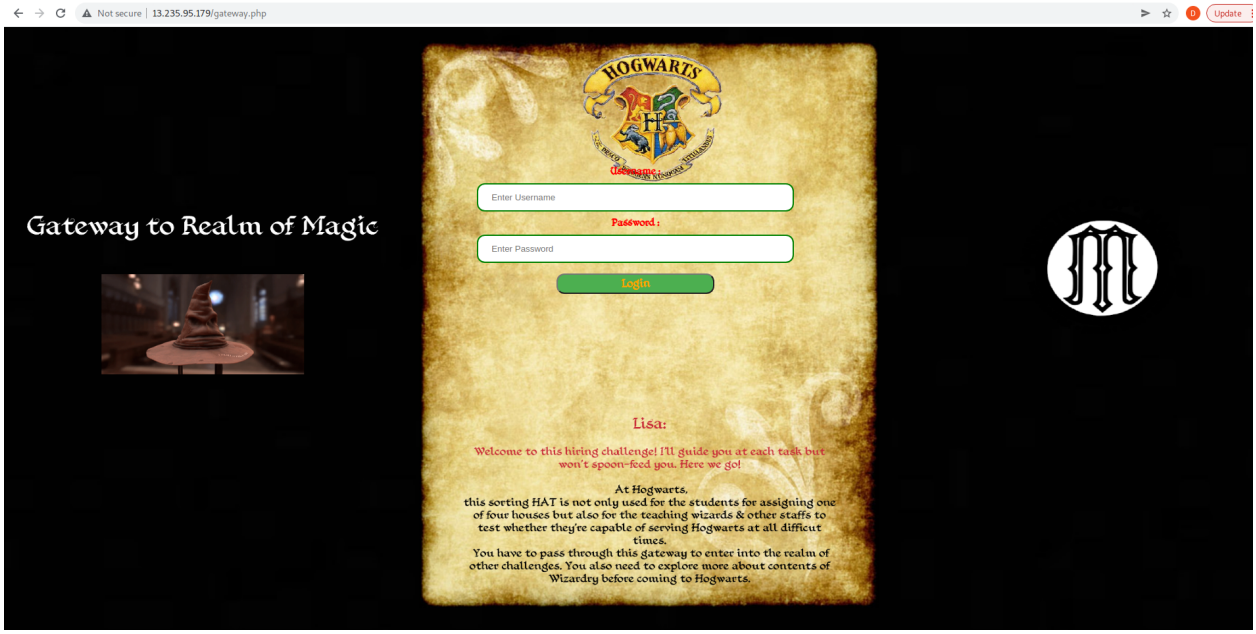
```
nmap -sC -sV 13.235.95.179
```



PORT 80:

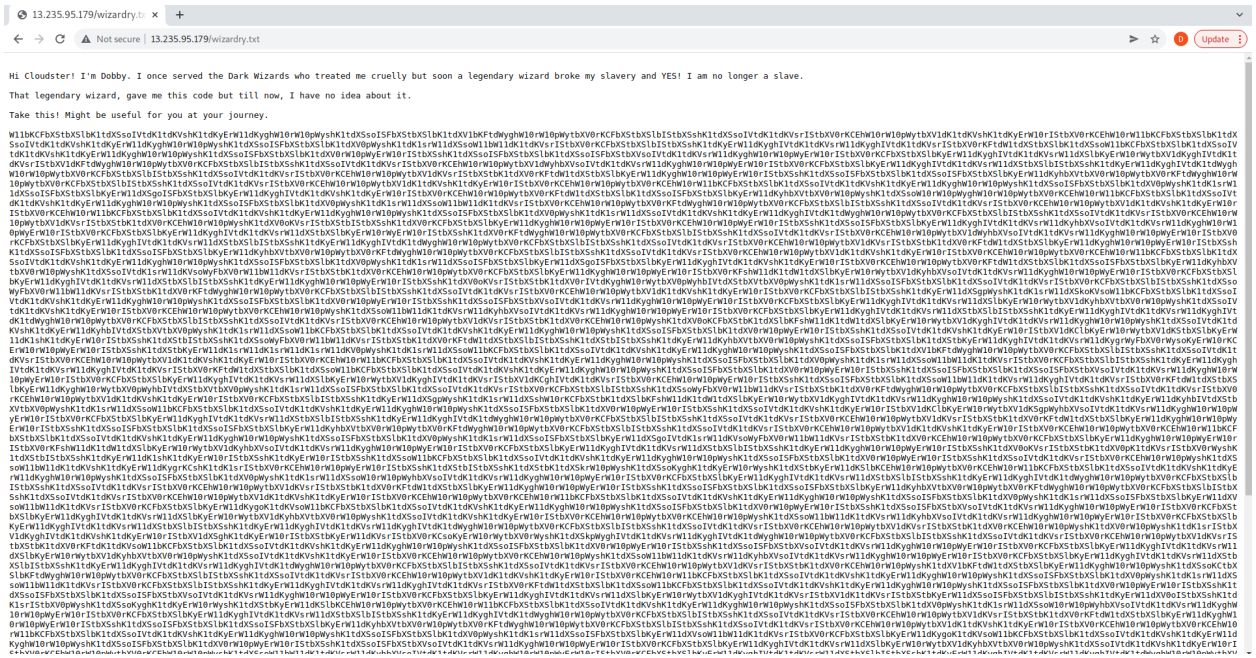
On visiting http://13.235.95.179, you are presented with a page which describing bit about the theme of ctf and a link to gateway.php, It is a static html page so we move on to gateway.php

Gateway.php

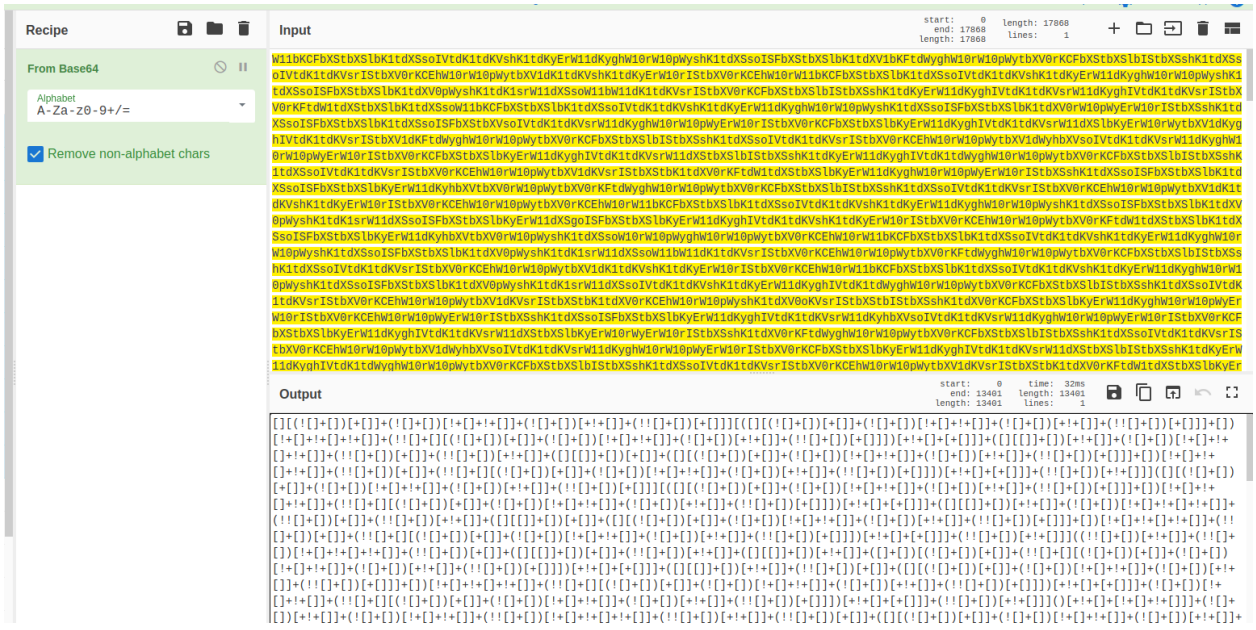


A Login portal asking for username & password.
Looking at the text at the last paragraph saying we need to read the contents of wizardry to pass the gateway.

On trying http://13.235.95.179/wizardry.txt we get some interesting page. On analyzing this long string, it comes out to be base64 encoded.



Decoding base64 string:

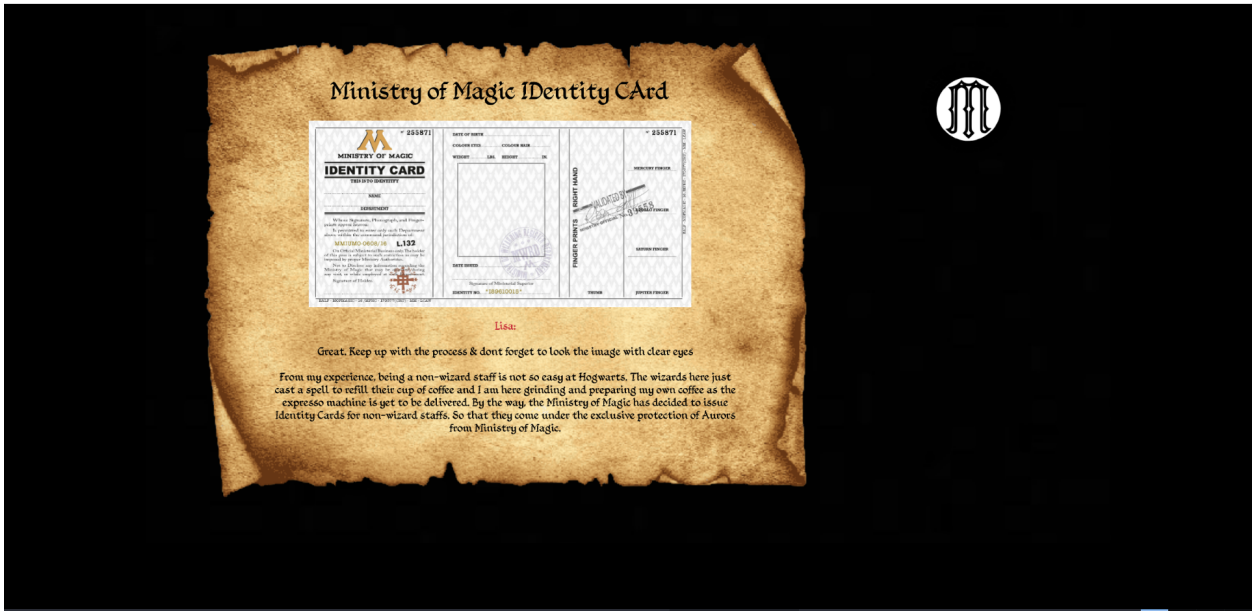


This gives another encoded string which can be recognized as encoded with jsfuck. Let's decrypt it

```
alert("id3nt1ty_card.php");
```

It gives us another directory.

Id3nt1ty_card.php

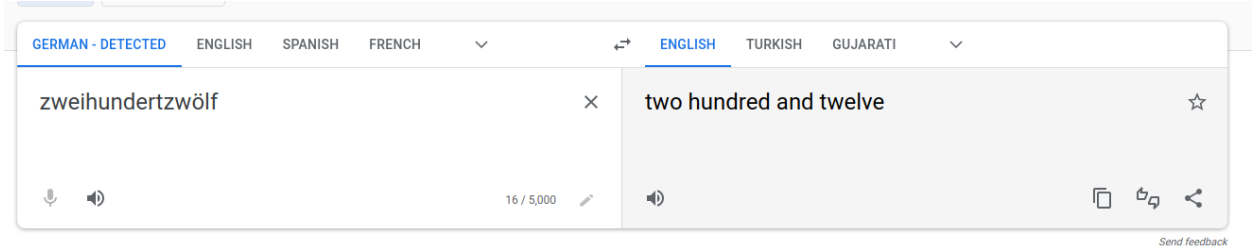


This page shows an Identity card but nothing written on it. Reading the text it says they are also issuing Identity cards to non-wizard also and there is a number on this identity card .
So I tried to see if it returns something if I add
`Id3nt1ty_card.php/?<param>=1`

I create a list of parma to try ,i will list few here: **Identity_Card, Id_card, idcard, id_card, id**

On trying these params, **Id3nt1ty_card.php?id=1** returned with different content length with text "**Seriously! How did you find this thing!?**"

On id=4 we are given a text "**zweihundertzwölf**". On looking at the way **ö** is written it is likely to be a language so on translating it to english. I got



Visiting id=212

Looking at the page we get x with some hex encoded value

The value of x on hex decoding returned: **Cl0uD_SeK_**

and a y with a link to **https://controlc.com/0e1fbc43**

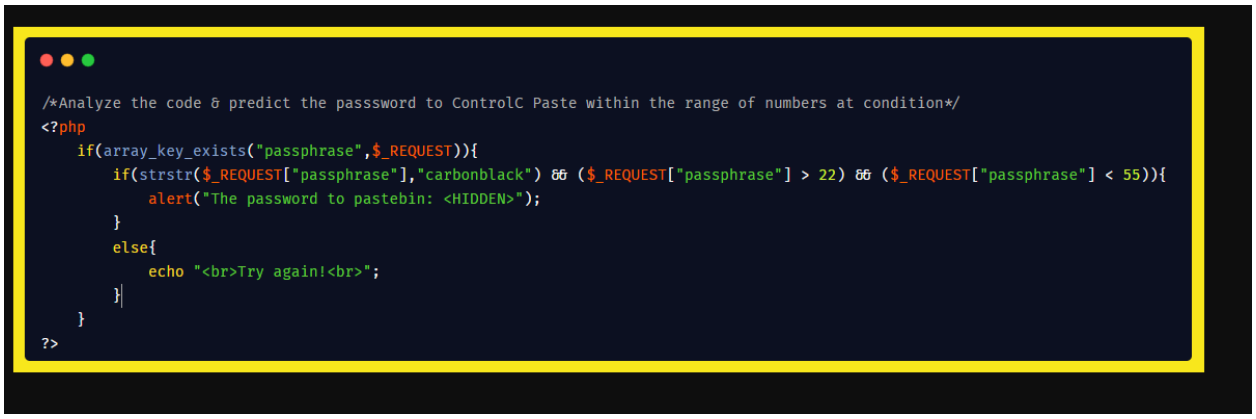
On visiting this link we need to enter the password to access the page.

Looking at Source code of the page id=212:

```
<html>
<style>
p {
  color: red;
}
div {
  border: 1px solid black;
  background-color: Coral;
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
</style>
<body style="background-color:black;">

<title>Ministry of Magic IDentity CArds</title>

<center><img src="images/hogwart0.png" alt="ID" style="width:78%"></center><br><br><center><p><b>Awesome, you found it. But still a long way to go.<b></p></center><html>
<body>

<div>
<center><h3>|_____Hey Cloudster, it's Dobby again!_____|</h3></center>
<br><center>I know that you are trying to solve the challenges made by Lisa. I found a scroll down near Lisa's room, as she was in a hurry to Headmaster's room, she didn't notice it. But I took it! :)</center>
<br>
<center><img src="images/idscroll.png" alt="Arithmancy" width="800" height="500"></center>
<br>
<br><br>
=======> x  = 436c3075445f53654b5f
<br>=======> y  = <a href="https://controlc.com/0e1fbc43">ControlC - password protected paste</a>
<br>
<a href="images/phpcard.PNG" hidden>PHP</a>

<br><br><center>####    This is just a beginning!   ####</center>
</div>
</body>
</html>
<br>
</body>
</html>
```

There is a hidden link to a phpcard.PNG on visiting the image we get

a php code saying it will return the possible password for controlc.com/0e1fbv43



```php
/*Analyze the code & predict the passsword to ControlC Paste within the range of numbers at condition*/
<?php
    if(array_key_exists("passphrase",$_REQUEST)){
        if(strstr($_REQUEST["passphrase"],"carbonblack") && ($_REQUEST["passphrase"] > 22) && ($_REQUEST["passphrase"] < 55)){
            alert("The password to pastebin: <HIDDEN>");
        }
        else{
            echo "<br>Try again!<br>";
        }
    }
?>
```

Analyzing the php code:

```
array_key_exists("passphrase",$_REQUEST)

#It will check if there is a key passphrase in $_REQUEST
array.

strstr($_REQUEST['passphrase'],'carbonblack')

#This function returns true if there is carbonblack
substring in $_REQUEST['passphrase']

$_REQUEST['passphrase']>22 && $_REQUEST['passphrase']<55

#When a string is evaluated in a numeric context, the
resulting value and type are determined as follows.
The string will be evaluated as a float if it contains
```

```
any of the characters '.', 'e', or 'E'. Otherwise, it
will be evaluated as an integer.
The value is given by the initial portion of the string.
If the string starts with valid numeric data, this will
be the value used. Otherwise, the value will be 0 (zero).
Valid numeric data is an optional sign, followed by one
or more digits (optionally containing a decimal point),
followed by an optional exponent. The exponent is an 'e'
or 'E' followed by one or more digits.
```

Refer to this link:
https://stackoverflow.com/questions/672040/comparing-string-to-integer-gives-strange-results

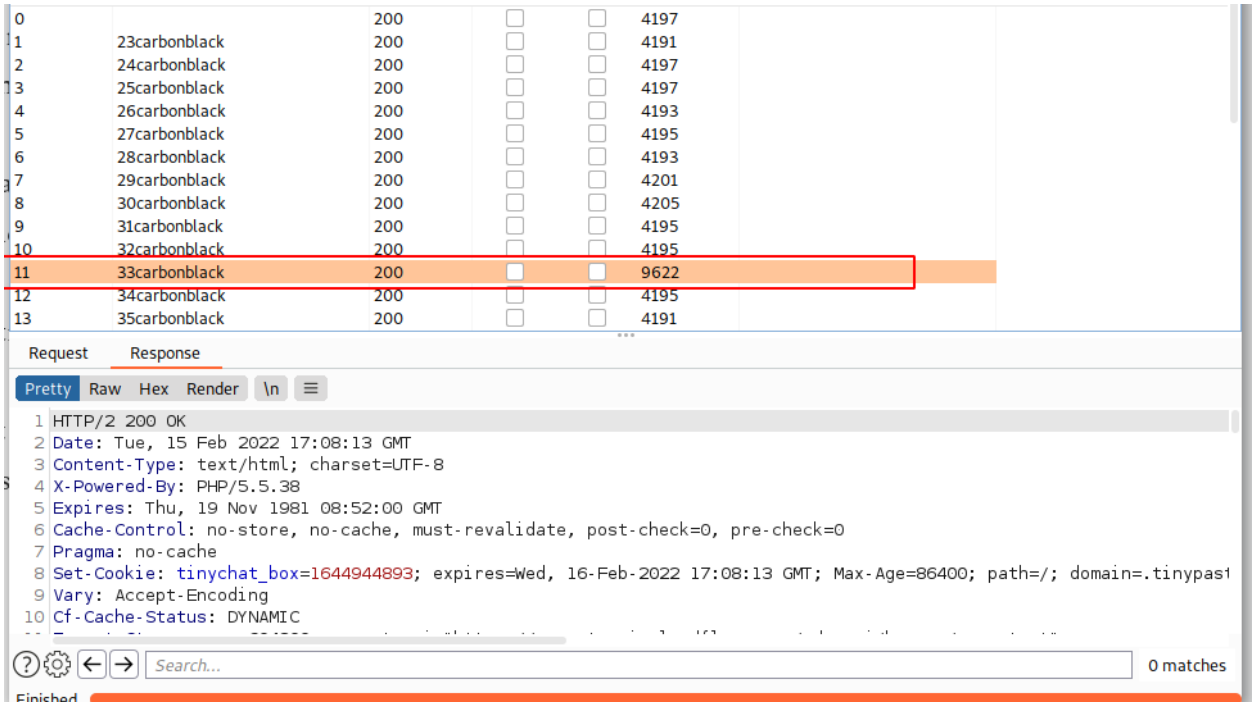
Constructing List of Valid passwords:

"Constraints: Should have carbonblack in it, greater than 22 and less than 55"

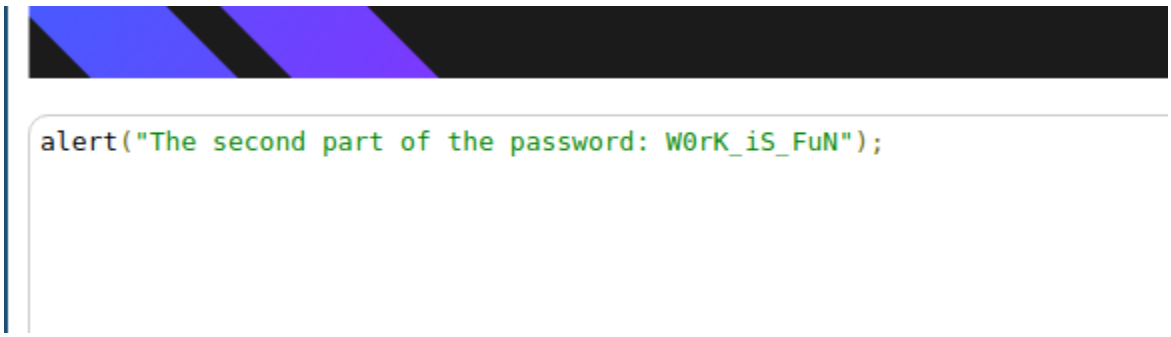
23carbonblack,24carbonblack,25carbonblack .....up to 54carbonblack
OtherPossible strings: 23[a-z]*carbonblack

Setting Burp intruder for trying list of password:

Out of all possible password "33carbonblack" comes out to valid password



On visiting the page after enter password we got the rest of the password string.



Password: `Cl0uD_SeK_W0rK_iS_FuN`

Visiting back to **gateway.php**

Now we need to identify the username, list of possible username can

be: Lisa,Dobby,Cloudster

Cloudster comes out to be the valid one for this password and we are sent to http://13.235.95.179/we1c0mE_start.php

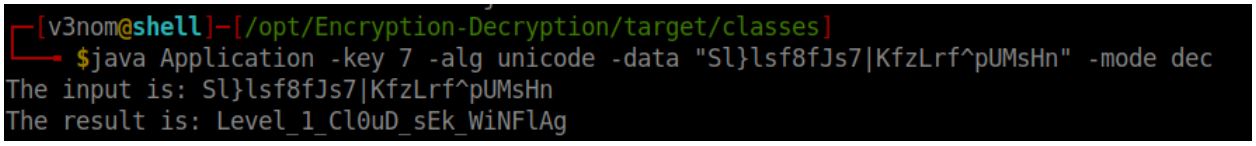
Source Code revealed nothing much, now looking at text they are emphasizing on cards/pictures. So i fetched the image **images/cfcard.png**.

Running: `strings -n 10 cfcard.png` returned some unicode.

It is pointing to unicoding with Krasy8 and to a domain on port 5000, *nmap also found this port.*

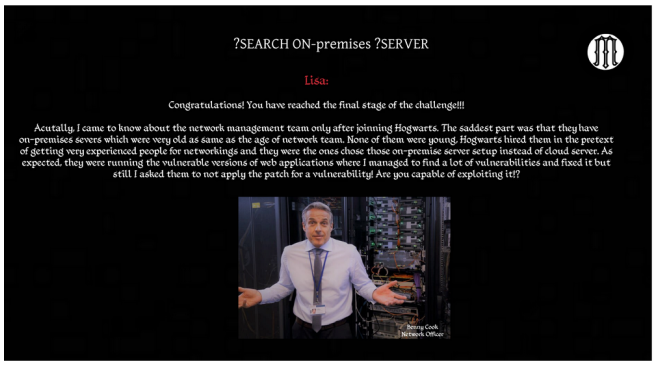
Git clone this repo to decode this unicode.
https://github.com/Krasy8/Encryption-Decryption

On key 7 it returned the flag.



Flag: Level_1_Cl0uD_sEk_WiNFlAg

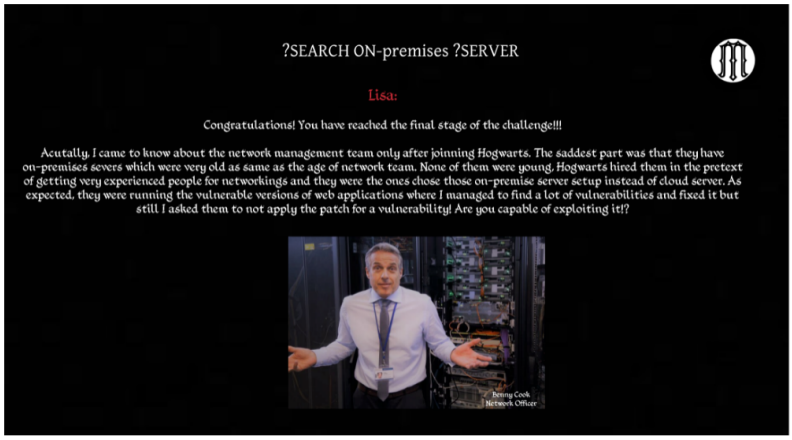
PORT 5000:



None

Straight away after page loads this None got my attention. Looking at the title of the page "**?SEARCH ON-premises ?SERVER**".

I tried to use **?search=hello**



hello

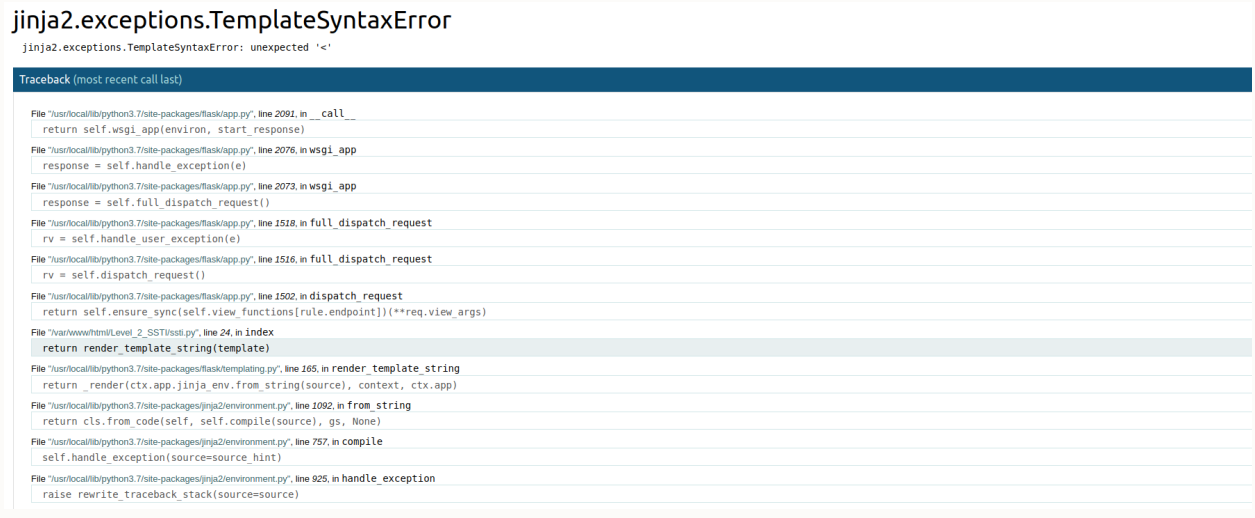
It got reflected on the page. *Then I tried ?server=hello nothing returned so i stuck with search.*

Now from nmap i know that it is a **Werkzeug/2.0.3 Python/3.7.10** server.

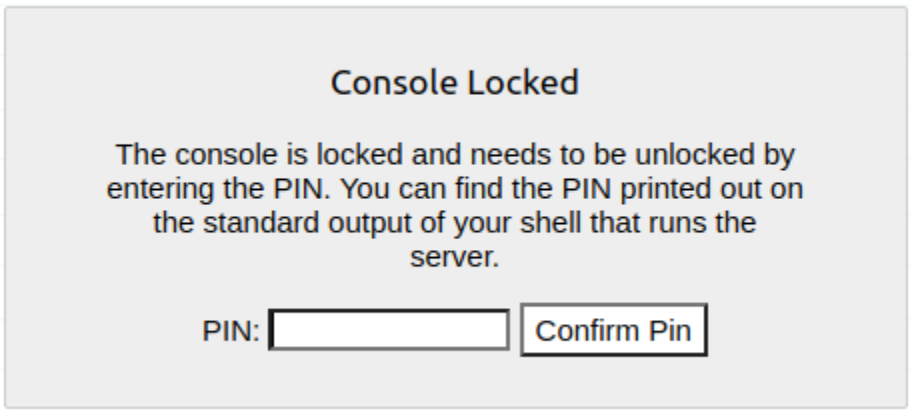
So first thing came to mind was Server Side Template Injection(SSTI)

Entering polyglot string: `${{<%[%'"}}%\`

Info it is a jinja2 template

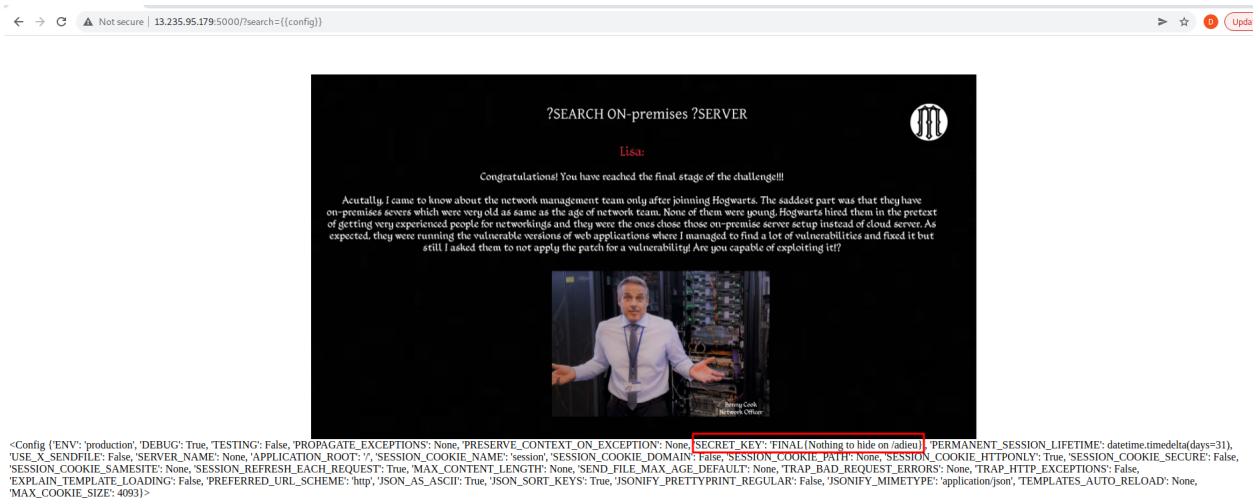


Console pops up but we need to have the DEBUG PIN to access it.



I tried some generic jinja payloads for remote code execution but it didn't work.

Moving on i tried a basic payload : {{**config**}} which returns configurations

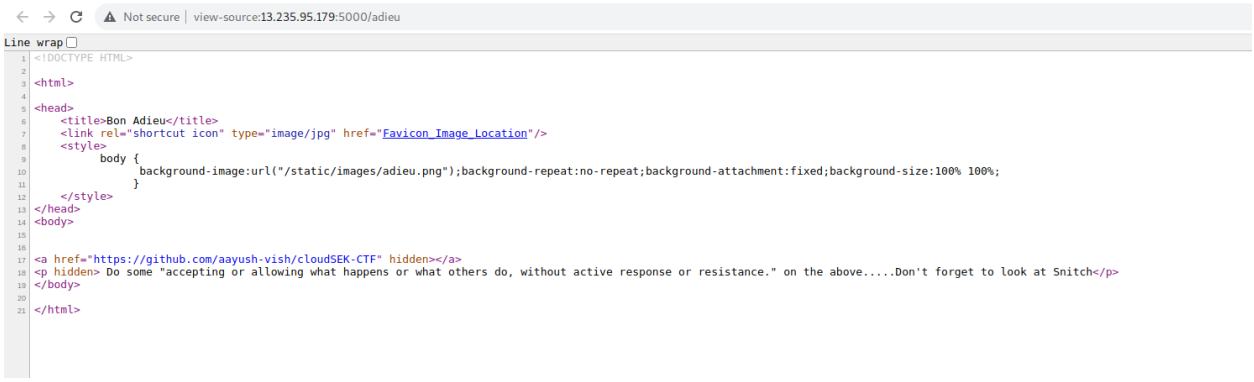


We get 'SECRET_KEY': 'FINAL{Nothing to hide on /adieu}

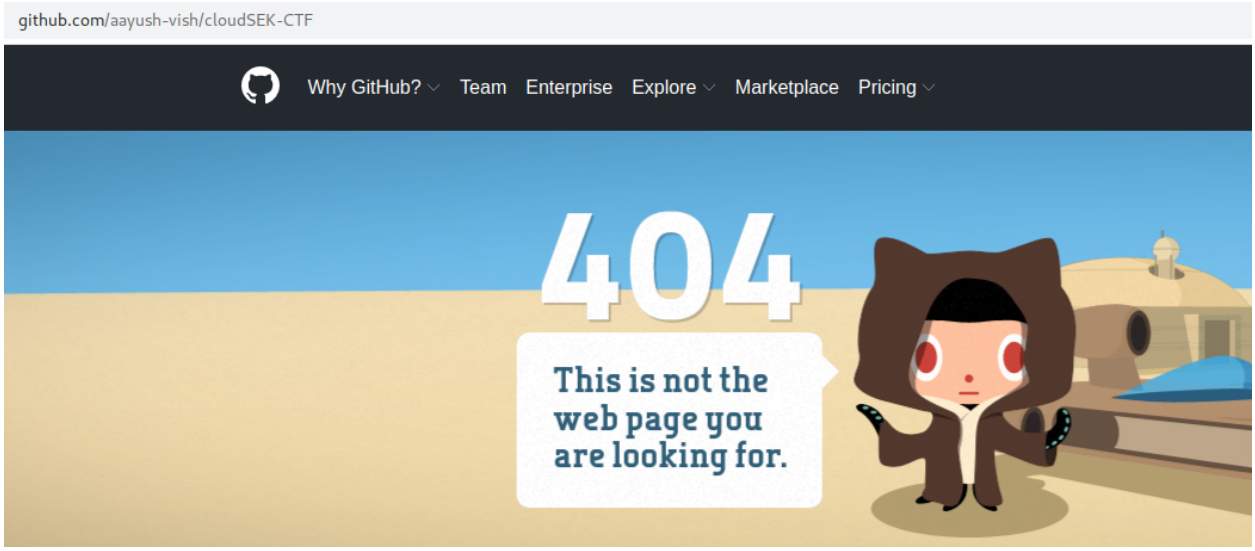
Let's Visit /adieu

So this is not the ending it is saying to complete the last challenge.

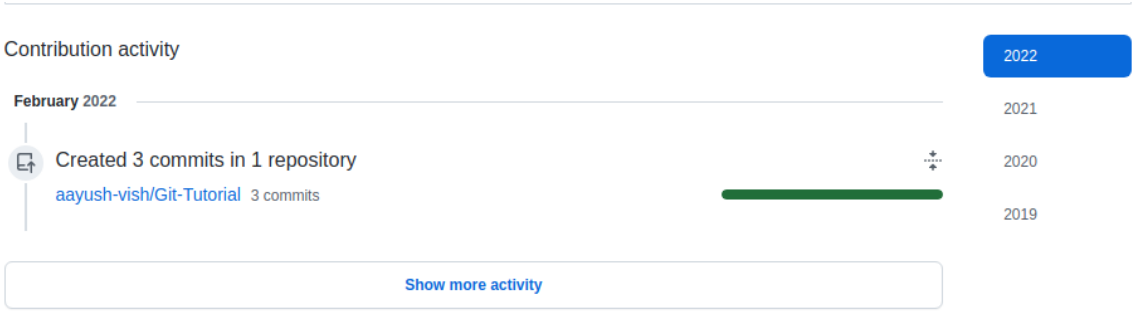
Source Code of the page adieu:



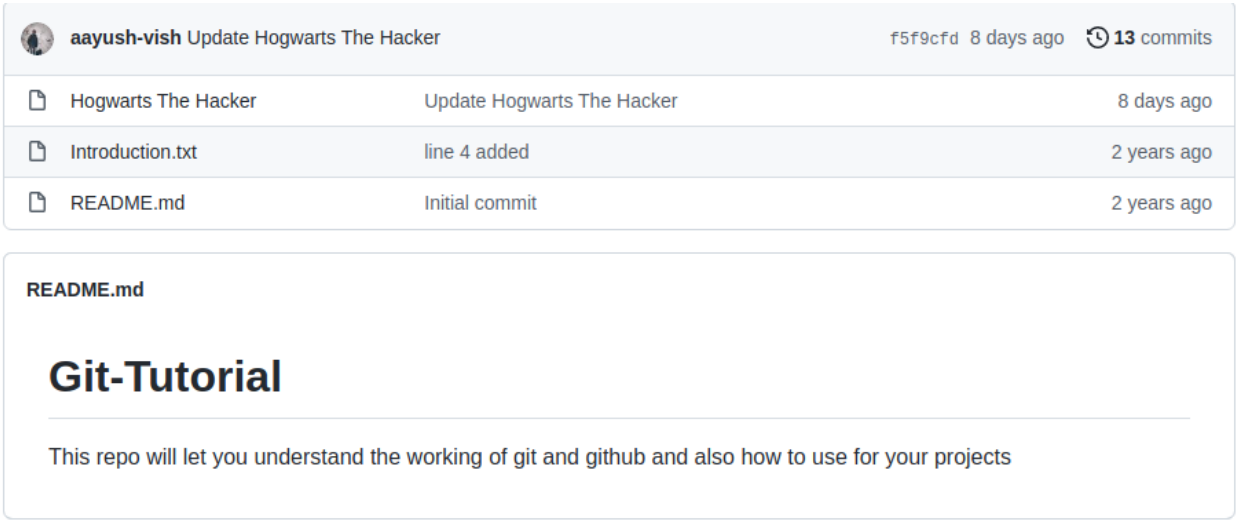
It is pointing to some git repo. Let's visit the link



It is returning 404. The thing we can do is look at aayush-vish recents commits.



On feb 1 repo is created. Let's visit.



Hogwarts The Hacker was updated just 8 days ago and it is using the ctf challenge theme, which is interesting!

4 lines (4 sloc) | 134 Bytes

```
1  {
2     "github-token-to-access-the-repo": Z2hwX2g2c1NKdXpHbUJ1SEszM0FVMnNwT3V5WHpVMDVJeDB6OXN6cA==
3  }
4  #_#Happy Hacking Hogwarts !!!!!#_#
```

I also looked into the git logs and found out that there are other tokens. In case one token doesn't work we can try these 2.

```
...  ...  @@ -1,4 +1,4 @@
 1    1   {
 2        -    "github-token-to-access-the-repo": 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
          2 +    "github-token-to-access-the-repo": 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
 3    3   }
 4    4   #_#Happy Hacking Hogwarts !!!!!#_#
```

On Decoding the above base64 token we get

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

"In git we can access private repo using access_token so let's use this token and see if we can access repo that is showing 404"

Command: `git clone https://<username>:<access-token>@github/username/repo`

```
┌─[v3nom@shell]─[/tmp]
└──╼ $git clone https://aayush-vish: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@github.com/aayush-vish/cloudSEK-CTF.git
Cloning into 'cloudSEK-CTF'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 21 (delta 4), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), 1.34 MiB | 183.00 KiB/s, done.
Resolving deltas: 100% (4/4), done.
```

Successfully cloned the repo Now let's look into the repo.

README file said we need to perform recon on this repo:

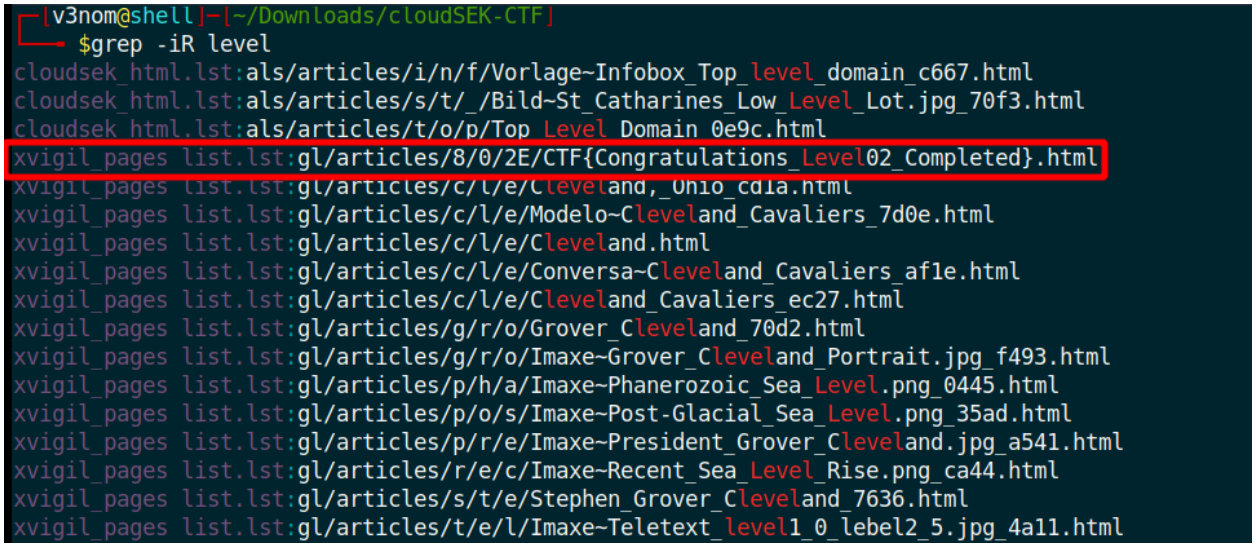
```
┌─[v3nom@shell]─[~/Downloads/cloudSEK-CTF]
└──╼ $cat README.md
# cloudSEK-CTF
### This is the repository you to need to visit to solve the CTF. Try your RECON TECHNIQUES.
```

Now i know that flag has "level" for sure in it so i performed a grep command:

```
grep -iR level
```

```
#-i flag for ignoring cases
#-R flag for recursively searching
```

Which returned the second flag.



FLAG: **CTF{Congratulations_Level02_Completed}**

.....