Abhedya Tech
Technical Round Challenges

First of all, I would like to thank Abhedya Tech for providing the set of interesting challenges for this ctf. I get to know about the ctf in the evening and it is almost a day or a day and a half long ctf. There were 8 web, 2 forensic, 2 reverse engineering and 2 cryptography questions present.

At night I started the Ctf. I first started with my favorite topic which is reverse engineering.

Reverse:

- **Challenge 11: Crackme**

  https://drive.google.com/file/d/1GA-nDGCgGeYBOth7ElLaOzLDQjPn1QJA/view?usp=sharing

  After analysing the code, I get to know the secret message was just the right rotation of actual flag similar to Caesar cipher. The only problem was to find the number of rotations. Which can be find using simple mathematical algo comparing the 1st element of secret text with first letter of flag i.e. “A”.

main.py — Run — Shell

```
alp="!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ"+ \
        "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~"
secret = "p3965J2Or%uLreo4<b504_5bN"

#finding the Rotation value
rotate_const =(alp.index("A")+len(alp)-alp.index(secret[0]))
#47 shifts

decoded = ""
for c in secret:
    index = alp.index(c)
    original_index = (index + rotate_const) % len(alp)
    decoded = decoded + alp[original_index]

print(decoded)

```

```
Abhedya_CTF{C6@ck3d_c0d3}
>
```

  And we got our first flag: Abhedya_CTF{C6@ck3d_c0d3}

- **Challenge 12: Keygenme**

  https://drive.google.com/file/d/1DGC2smkMoWuUZQrh7CBjyAbSit_XUOO5/view?usp=sharing

  This one’s code was little tricky, but you would get the idea of finding the remaining part of the flag from check key function. Inside the code. And then just write a simple python script and we got are next flag:

  AbhedyaCTF {Pwn_th3_Tr0n_fd9f6c17}

main.py — Run — Shell

```
import hashlib
a = "AbhedyaCTF{Pwn_th3_Tr0n_"
hid = "xxxxxxxx"
c = "}"
hid=hashlib.sha256(b"Abhedya").hexdigest()[4]+hashlib.sha256(b"Abhedya").hexdigest()[5]+hashlib.sha256(b"Abhedya").hexdigest()[3]+hashlib.sha256(b"Abhedya").hexdigest()[6]+hashlib.sha256(b"Abhedya").hexdigest()[2]+hashlib.sha256(b"Abhedya").hexdigest()[7]+hashlib.sha256(b"Abhedya").hexdigest()[1]+hashlib.sha256(b"Abhedya").hexdigest()[8]
key=a+hid+c
print(key)

```

```
AbhedyaCTF{Pwn_th3_Tr0n_fd9f6c17}
>
```

Now in the morning time after attending couple of my online classes. I again started the ctf now I looked into the cryptography challenges.

Cryptography:

- **Subs:**
  https://drive.google.com/file/d/17vmzZov-eNy0rHfzWe6GgvszfMSCO91o/view?usp=sharing

  After analysing the cipher, I understand that it is encoded in base64. So, after comparing result of base64 encryption of AbhedyaCTF and starting few values of given cipher. I get to know that capital letters are shifted right by 16, small letters are shifted by 13 and numbers are not rotated at all.
  So after writing a basic python script we retrieved the flag:

  AbhedyaCTF{it_was_substituted}

main.py — Run

```
import base64
check_var=base64.b64encode(b"AbhedyaCTF")

val="AGTbJGB5IEXEBagcqP93IHXsp3Fvp3BcqRFOJGB9"
print("Compare the shift\n",check_var.decode("utf-8"),"\n","AGTbJGB5IEXEBag\n\n")
#there is a shift of 13 in small alphabetes, 16 in capital and none in numeric

new=""
for i in val:
    if ord(i)<60:
        new+=i
    elif ord(i)>95:
        i=chr(97+((ord(i)-97)+13)%26)
        new+=i
    else:
        i=chr(65+((ord(i)-65)+16)%26)
        new+=i
print(base64.b64decode(new).decode("utf-8"))

```

Shell

```
Compare the shift
 QWJoZWR5YUNURg==
 AGTbJGB5IEXEBag


AbhedyaCTF{it_was_substituted}
>
```

- **Triple-H:**

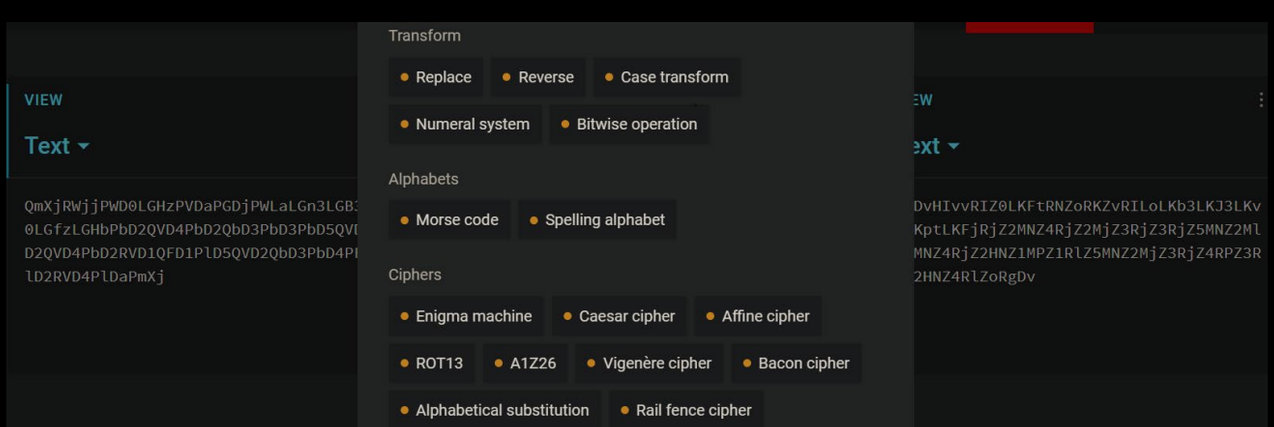
  https://drive.google.com/file/d/17J67fRKaiG_LVZnkfgIy9GHuTFMjSqMi/view?usp=sharing

```
QmXjRWjjPWD0LGHzPVDaPGDjPWLaLGn3LGB3LGj0LGfzLGHbPbD2QVD4PbD2QbD3PbD3PbD5QVD2QlD2QVD
4PbD2RVD1QFD1PlD5QVD2QbD3PbD4PFD3PlD2RVD4PlDaPmXj
```

I tried bunch of things to decipher it. Tried to find patterns in base 64 not any luck, I thought there were 2 H in cipher so the 3 part of cipher can be done but no luck with that information.

I also tried other cipher techniques.

I know a great website for that matter https://cryptii.com/ I tried bunch of different cipher but again no luck☹.



Then after completing my college lectures, I started again in the evening. Now I look into the forensics challenges.

Forensics:

Both the questions contains an image.

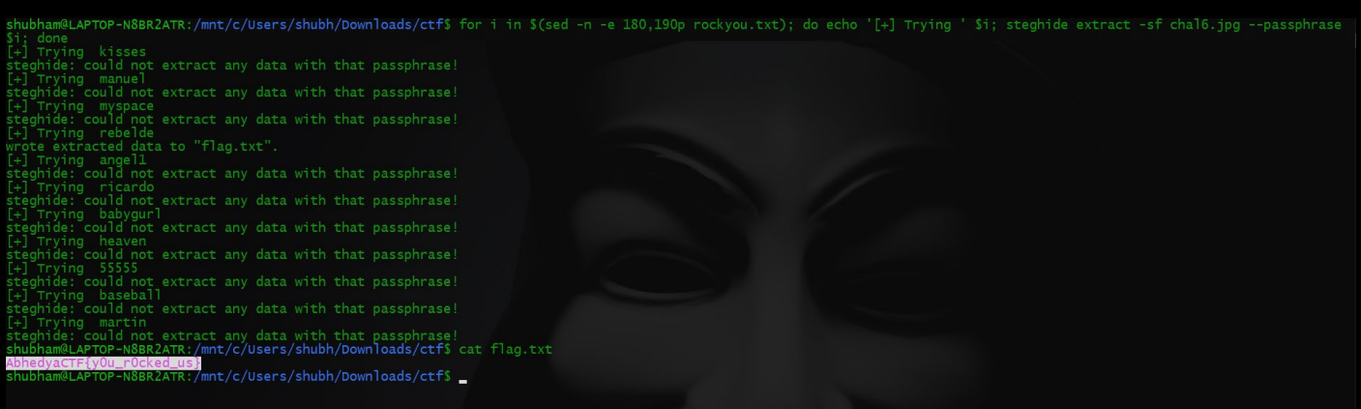
- First I tried “strings” cmd to find for any embedded text or flag inside the image.
- Than I use “Exiftool” to find whether something wasn’t hidden the metadata.
- Than used “Binwalk” to find any embedded file.

Than finally I used staghide tool for finding the hidden data.

- **Challenge 9:**

  https://drive.google.com/file/d/16VIStLDh3lRAroHTxwle0xnQRsd2TKAT/view?usp=sharing

  steghide extract -sf chal6.jpg asked for passphrase. The image highlights rockyou text file so after writing a bash cmd
  (for i in $(sed -n -e 80,150p rockyou.txt); do echo '[+] Trying ' $i; steghide extract -sf chal6.jpg --passphrase $i; done) # I used sed tool as rockyou.txt file is very large
  to iterate over rock you file a flag.txt file was obtained thus finding our next Flag:
  AbhedyaCTF{y0u_r0cked_us}



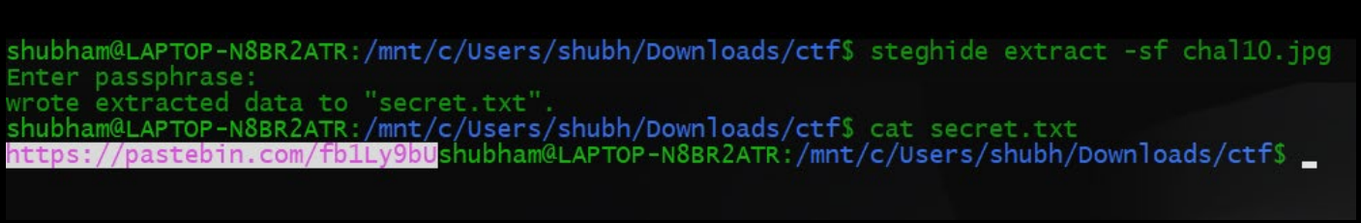
- **Challenge 10:**

  https://drive.google.com/file/d/1Ar023pe8GsSmt46crc4nFhiSXq2L3C-Q/view?usp=sharing

  I used the same tool on this file also. This file image says about password being “incorrect” so after entering the passphrase we retrieved a secret file. Containing the url of a pastebin the pastebin contains this message.

  “P s s t! Th e r e i s n o s e c r e t h ere l o ok somewhere else.”

  I sepreted out the character in different size.
  Small: st!Thisoctereok
  Large: Pserenserehlo
  I tried them as passphrase but no luck there. But the small letter combination looks like a flag content so
  Maybe the flag is AbhedyaCTF{st!Thisoctereok}
  ## I also tried deciphering the text but doesn’t find anything there##



Finally, I started web questions at 2nd night.

Web:

- **Challenge 1:** http://164.52.213.208:8002/

  In challenge 1 we have to access the website through localhost. For that matter we have to add ***"X-Forward-For:127.0.0.1"*** into the header. So for that we have to fire up burp to intercept the request and then add this header file.
  Or you can use curl cmd

  "curl -k http://164.52.213.208:8002/ -H "X-Forwarded-For: 127.0.0.1"

  And wohlaa we got the flag:
  AbhedyaCTF{N0_int3rnal_address_is_s3cure}

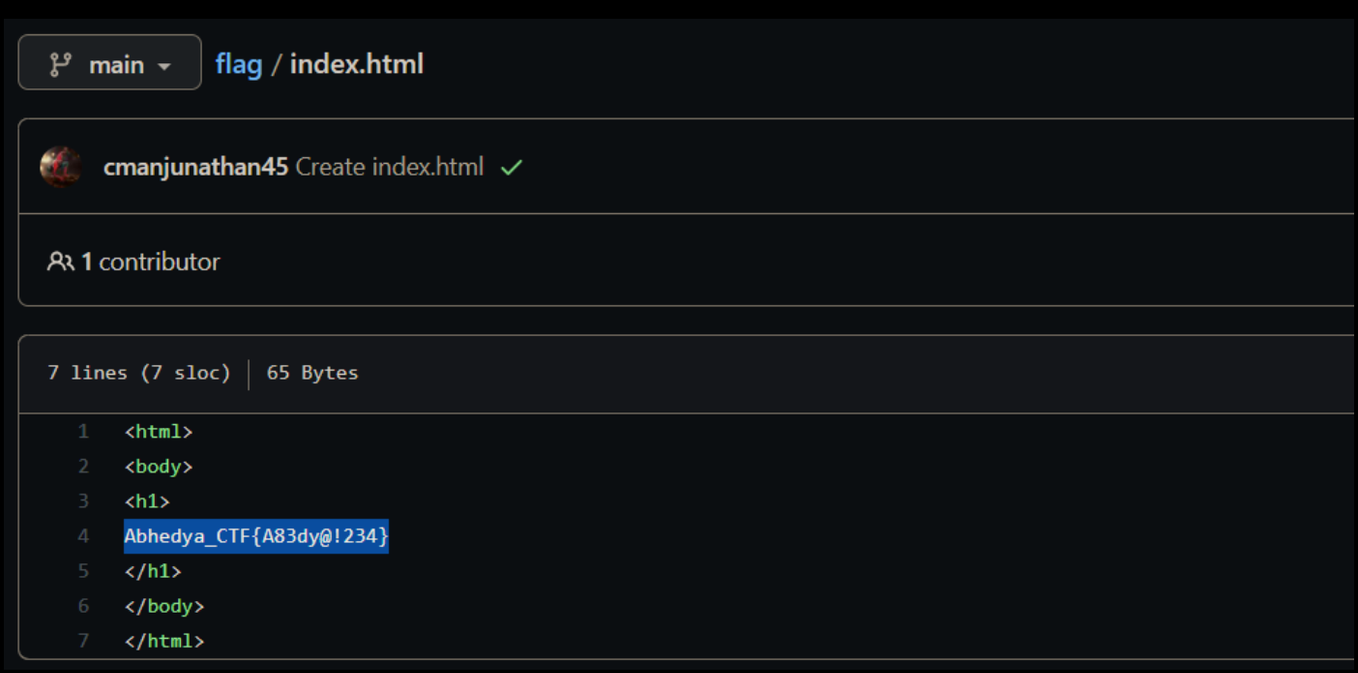
```
shubham@LAPTOP-N8BR2ATR:/mnt/c/Users/shubh/Downloads/ctf$ curl -k http://164.52.213.208:8002/ -H "X-Forwarded-For: 127.0.0.1"
Congrats on getting the flag: AbhedyaCTF{N0_int3rnal_address_is_s3cure}<html>
<title>
        Portal Login
</title>
</html>
shubham@LAPTOP-N8BR2ATR:/mnt/c/Users/shubh/Downloads/ctf$
```

##

For Challenge 2, Challenge 3, and Challenge 4. I injected various html and JavaScript but there was nothing seems to be working. And the same was with SQL injection, I inserted various queries but there were no responses something was just not adding up so I skipped these questions for now

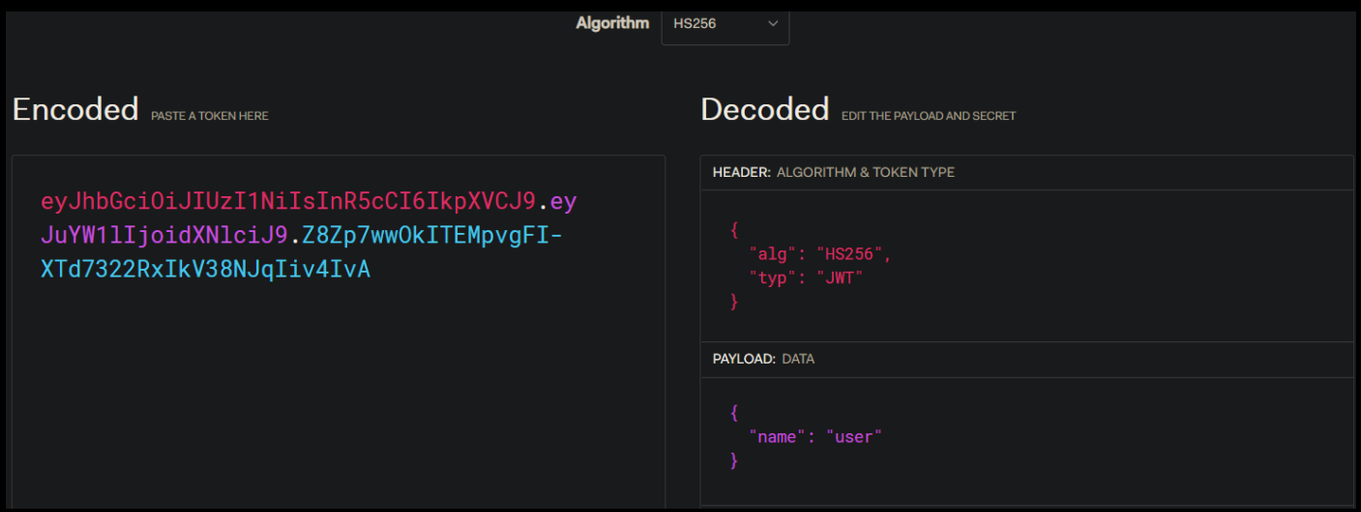
##

- **Challenge 5:** http://164.52.213.208:9007/
  The website contains a word "cmanjunathan45" so It looks like some kind of username I thought maybe we have to login via this user or something like that but first of all I googled it, I found a git repository of this user so I go through his other repos, and there was 1 repo named flag which contains the flag for this problem.
  Abhedya_CTF{A83dy@!234}



- **Challenge 6:** http://164.52.213.208:8004/

  After inspecting the website, I found a cookie named login and the value seemed to be like Jason web token so visited https://jwt.io/ to decode the jwt and a Jason file is obtained I edited the web token and entered the value in cookie value but it doesn't seem to be working I even used burp and send the request to repeater and made some changes in the header files or posting as admin. I googled various ways to login otherwise or via jwt but doesn't got anything useful ☹.



- **Challenge 7:** http://164.52.213.208:8006/

  In this problem also it also referred to as look in the headers or it is unacceptable. so I tried googling accept related headers and modifications. But does not get anything interesting I tried burb with some modifications in header file but it shows bad request most of the time. So no luck here also ☹.

As it was getting late so decided to take a nap and look for the next and remaining problems in the Morning.

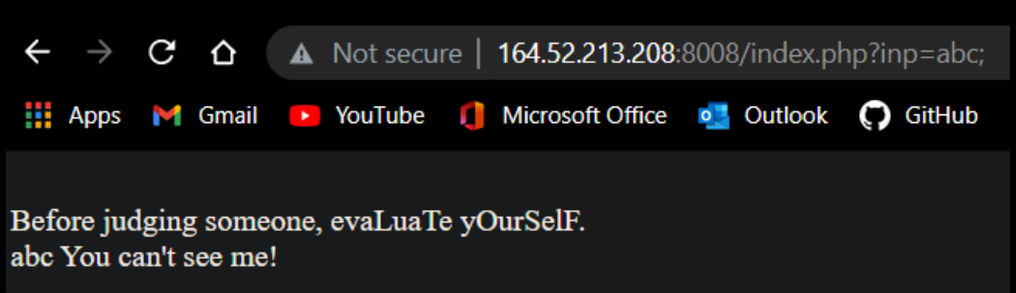
The next day after a lab of college, There is only time for the last question and writeup, So I started the last challenge.

- **Challenge 8:** http://164.52.213.208:8008/

  The website shows "*You can't see me*". After inspecting it I find something in comment "*?inp*" seems like a query for a GET request. So tried

  http://164.52.213.208:8008/index.php?inp=abc;

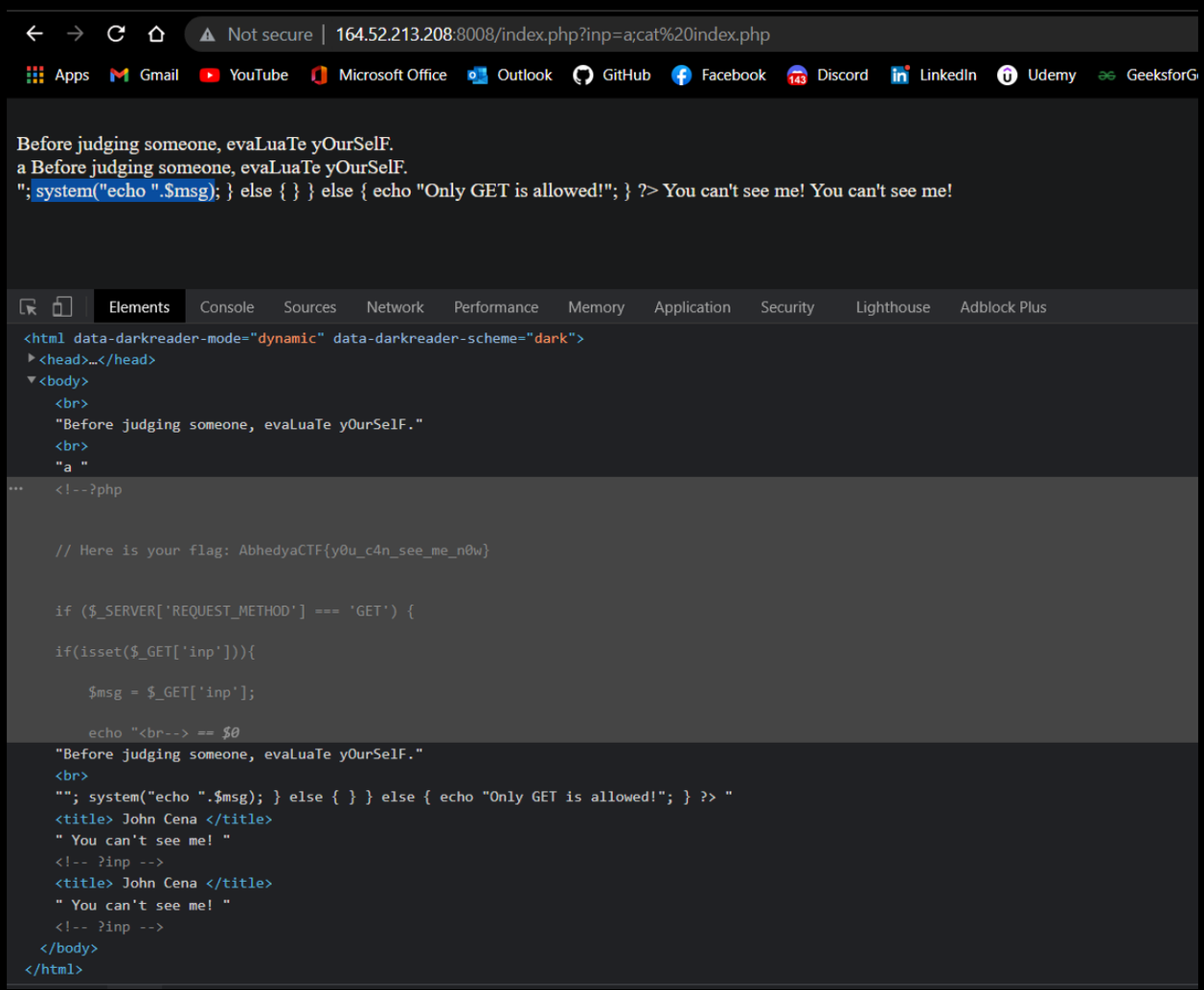
  It returns this:



As it is printing the input so maybe it is using echo cmd, I tried as a hunch to insert multiple cmds using semicolon. So I printed the index.php using this url.

http://164.52.213.208:8008/index.php?inp=a;cat%20index.php

And *hurrahhh* we got this result:



And here is our Flag:

AbhedyaCTF{y0u_c4n_see_me_n0w}

Thank you

ctf{Shubham_malik}