CITS1003 Project Report

Student ID: [24412097]

Student Name: [Kiet Le]

Part 1 – Cryptography

Advanced Emu Standard

Step 1

Following the rule that each character in a plaintext is considered as 8 bits, or 1 byte. Considering that the given command “deactivate_special_procedure_123“ is in fact 32 characters long, therefore having 32 bytes (according to ASCII). We need to make two 16 byte pieces in order to put into the website. this will look like this:

First half: deactivate_speci
Second half: al_procedure_123

Step 2

We will then need to go to the given website “http://34.87.251.234:3001”.

After that we will then need to insert both the first and second half of the 16 byte commands in the “command Encrypter” part in the website, then press the blue “get Ciphertext” button.

The website will give result of the following for the first half:

3155433d53ed30c89aef89b2e7273924

And the second half:

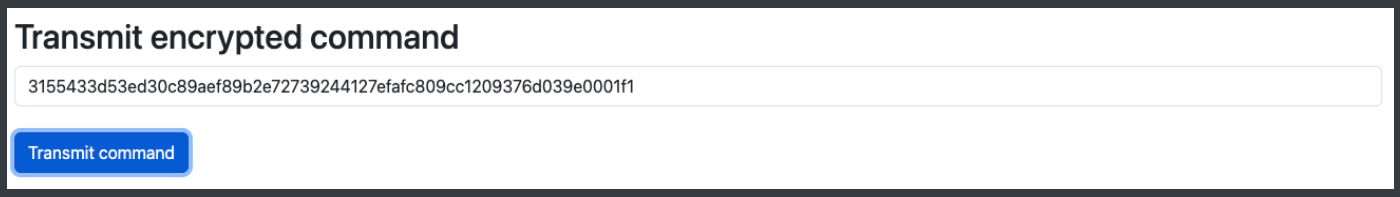
4127efafc809cc1209376d039e0001f1

Step 3

With the given ciphertext we will need to understand that to transmit and get the final result we must first need to combine the two parts of the given ciphertext together. Like this:

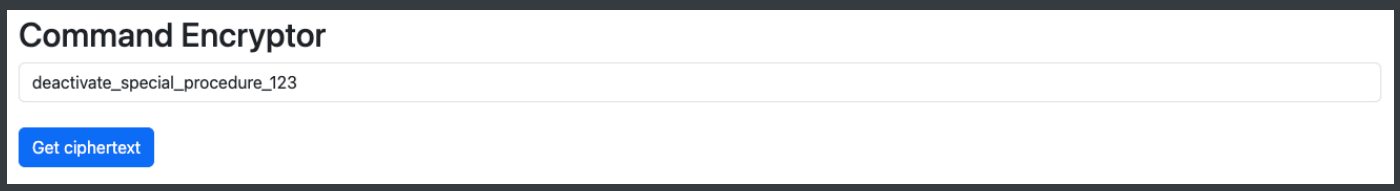
3155433d53ed30c89aef89b2e72739244127efafc809cc1209376d039e0001f1

Now just Input the full cipher text into the “transmit encrypted command” input space. (see attached image)



Step 4

Input the full plain text of “deactivate_special_procedure_123“ into the “Command Encryptor” input space. (see attachment)



Step 5

Proceed to click the “transmit command” button after inputting the content in step 4 and 5. This will provide you with the following result.

Result:**

Self-destruct protocol successfully deactivated!

UWA{3CB_i5_bL0cK_Ind3peNd3nt!}

Flag Found:

```
UWA{3CB_i5_bL0cK_Ind3peNd3nt!}
```

Emu Cook Book

Step 1

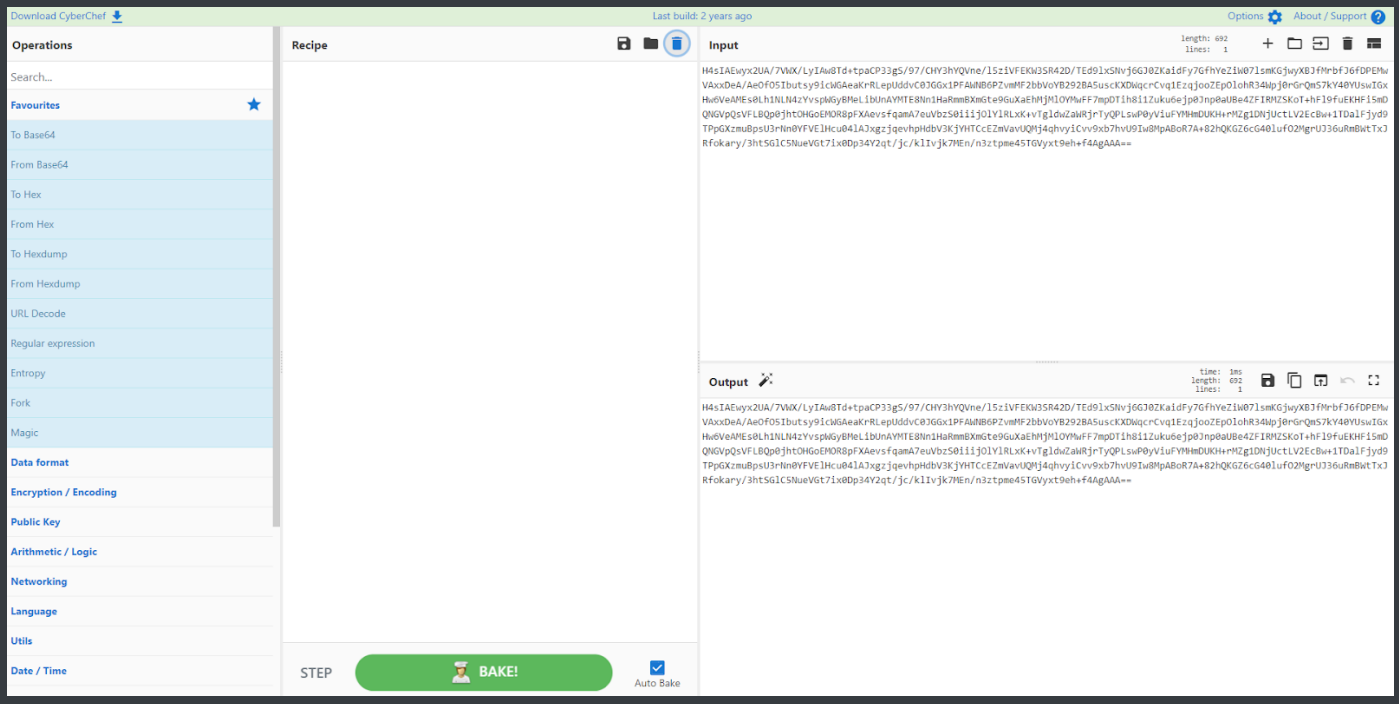
Firstly it is important that to decode something you first must identify what type of code it is. Given that the given code has an ending of “+f4AgAAA==” this would only tell that it is a “Base64” text.

Therefore the identified given code is Base64.

Step 2

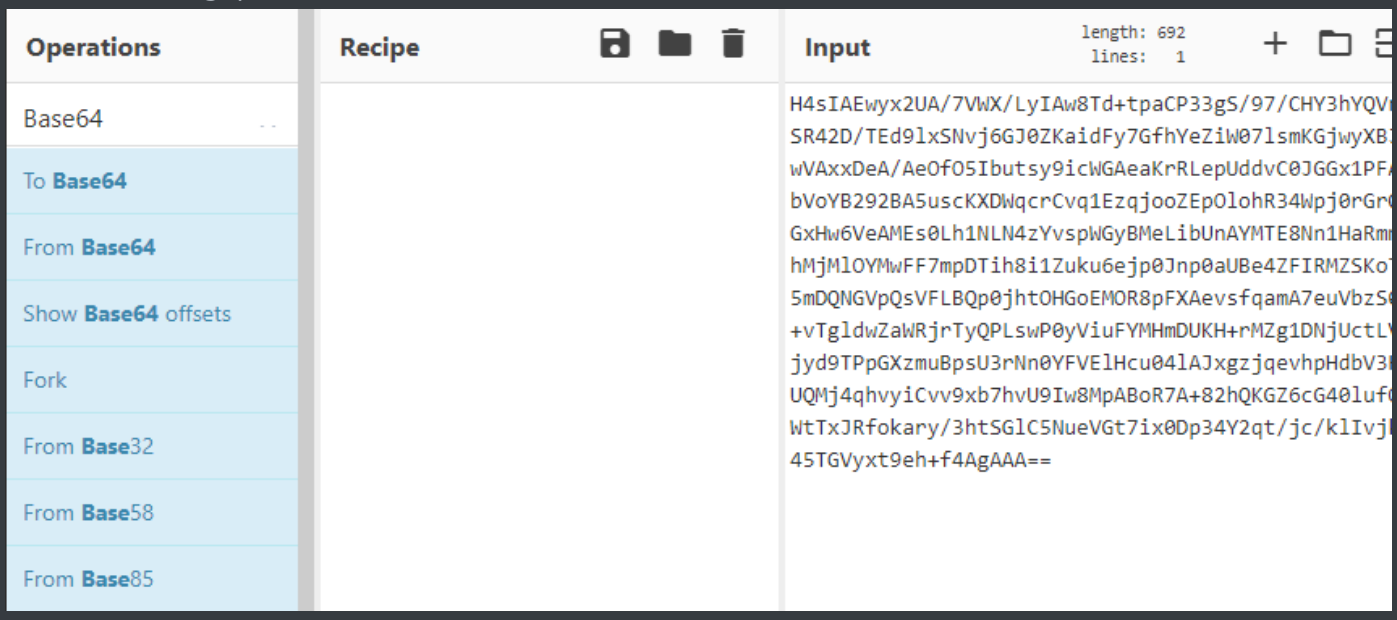
In order to decode this we will need to input the code into “CyberChef.io”.

Which will look something like this (see attached image):



Step 3

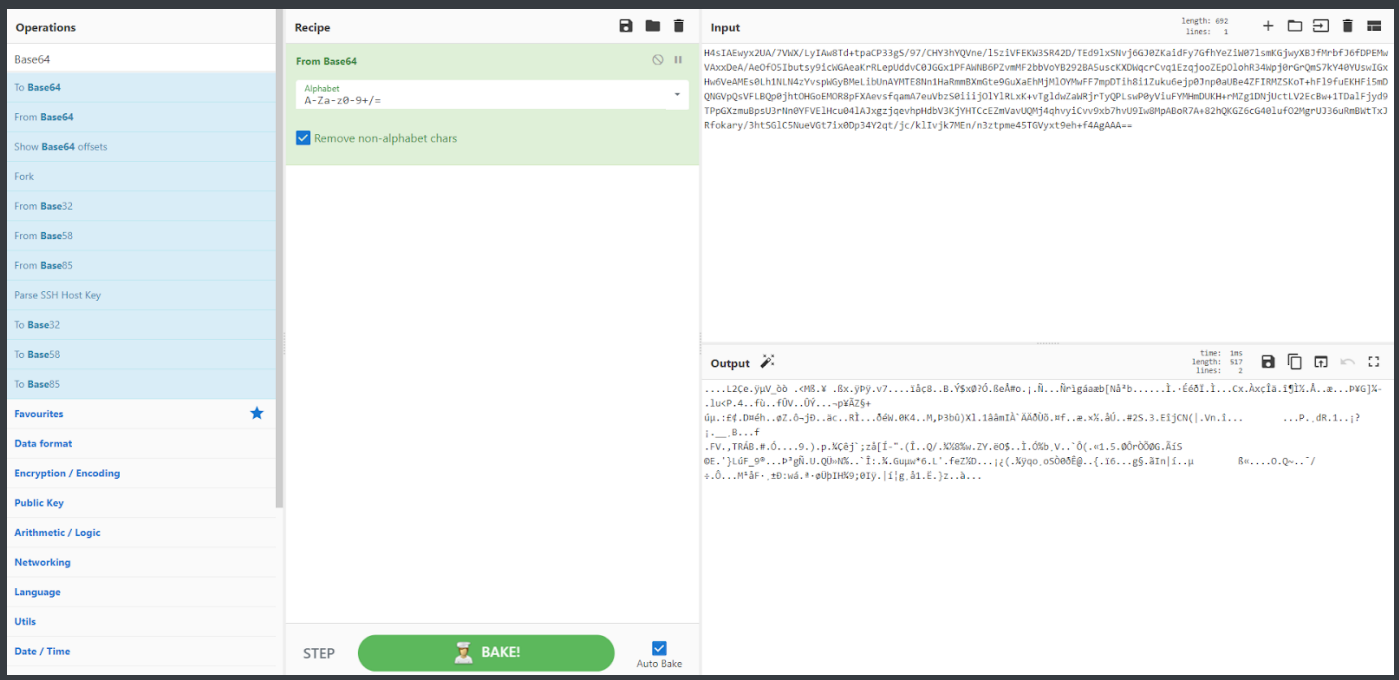
We will then need to select an operation to decode the Base64 code. Knowing that it is already in Base64 format we will need to search for “Base64” in the search bar (see attached image):



Step 4

We will then need to click and drag the “from Base64” operation into the “recipe” section. This is because the given code is already in Base64 format, so we will need to decode it from Base64 into something else.

Once selected it should look something like this (see attached image):

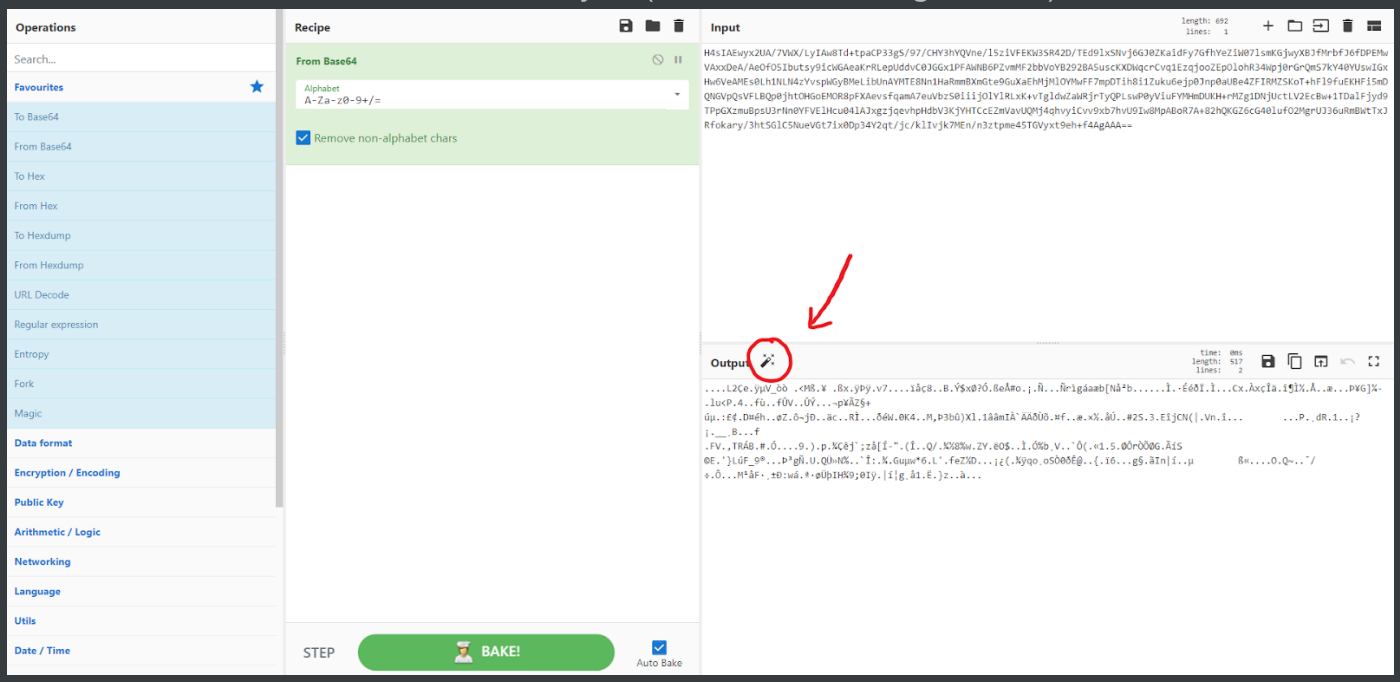


Step 5

Identifying the given output in the output box it is required that we use “Gunzip” to further decode the text.

To do this we will simply click on the “wand” button next to the output box.

Here's a red circle and arrow to show you (see attached image below):



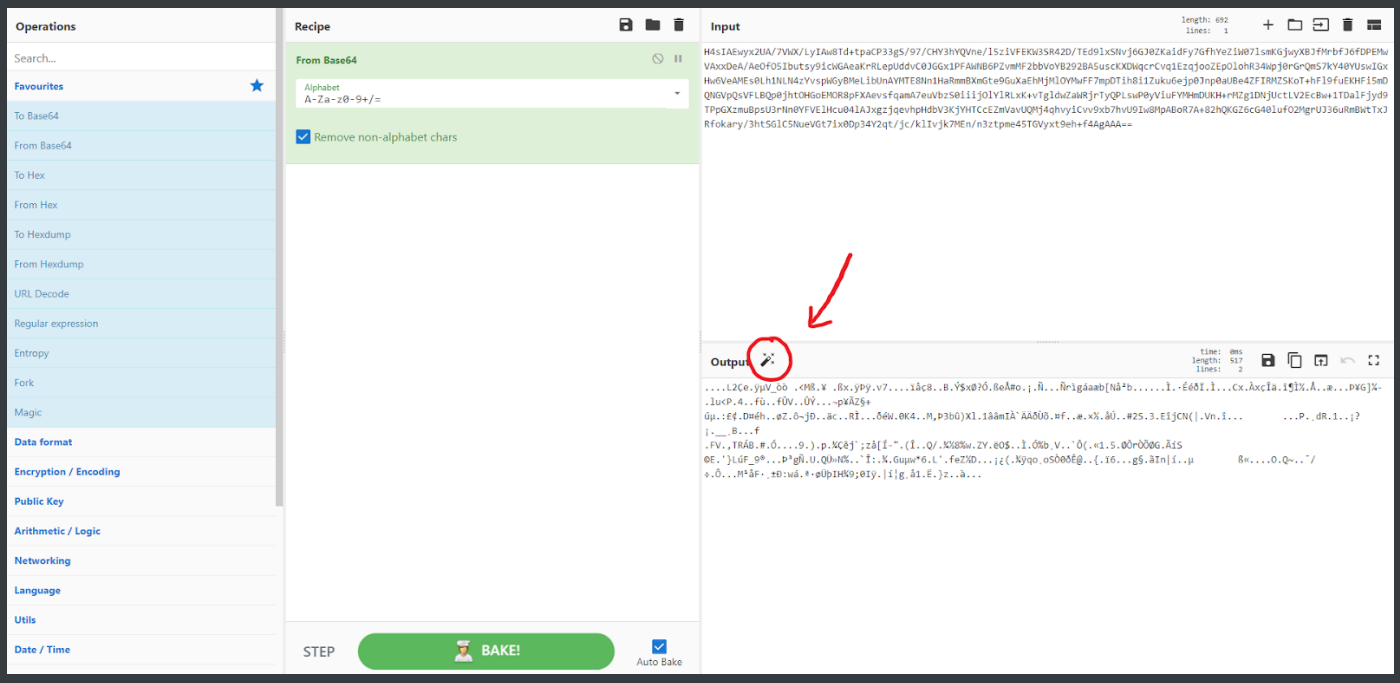
Once clicked it should look something like this:

“56 56 64 42 65 33 52 49 4d 31 39 6c 54 57 39 50 4e 56 39 33 4d 55 78 4d 58 32 34 7a 56 6a 4e 79 58 33 4e 55 62 31 42 66 5a 44 41 78 62 6b 64 66 64 45 67 7a 4e 57 56 66 5a 46 5a 74 51 6c 39 6a 4d 55 4a 79 58 32 4e 49 4d 32 56 47 58 32 4e 6f 4e 45 78 73 55 33 30 3d”

The fact that we see not only Gunzip but “URL decode and “From Hexdump” too is that it will decode everything to “Hex” is the default format to display binary data.

Step 6

We will then need to press the “wand” once more, this will be the easier alternative to search, drag and drop in the recipe section for “From Hex”. (see the wand once again below):



This will result in the following result in the “Output box”:

“VVdBe3RIM19lTW9PNV93MUxMX24zVjNyX3NUb1BfZDAxbkdfdEgzNWVfZFZtQl9jMUJyX2NIM2VGX2NoNExsU30=”

Step 7

Seeing that the final result in step 6 we can conclude that it ends with “oNExsU30=”. This will mean that it is yet another “Base64” text.

In order to decode this we will need to acquire another “From Base 64”. We do this by simply clicking the “wand” button once more.

Resulting in the following see below:

Result:

UWA{tH3_eMoO5_w1LL_n3V3r_sToP_d01nG_tH35e_dVmB_c1Br_cH3eF_ch4LlS}

Flag Found:

```
UWA{tH3_eMoO5_w1LL_n3V3r_sToP_d01nG_tH35e_dVmB_c1Br_cH3eF_ch4LlS}
```

Emu Casino

Step 1

It is important to identify what the given hits tell you, as stated by ChatGPT and Google(Palisade) “A seed in a random number generator serves as the initial input that determines the starting state of the algorithm.”

Looking at the given code “flip_coin.py” we can see that the seed is based on two given aspects, the “round” and the “session id”.

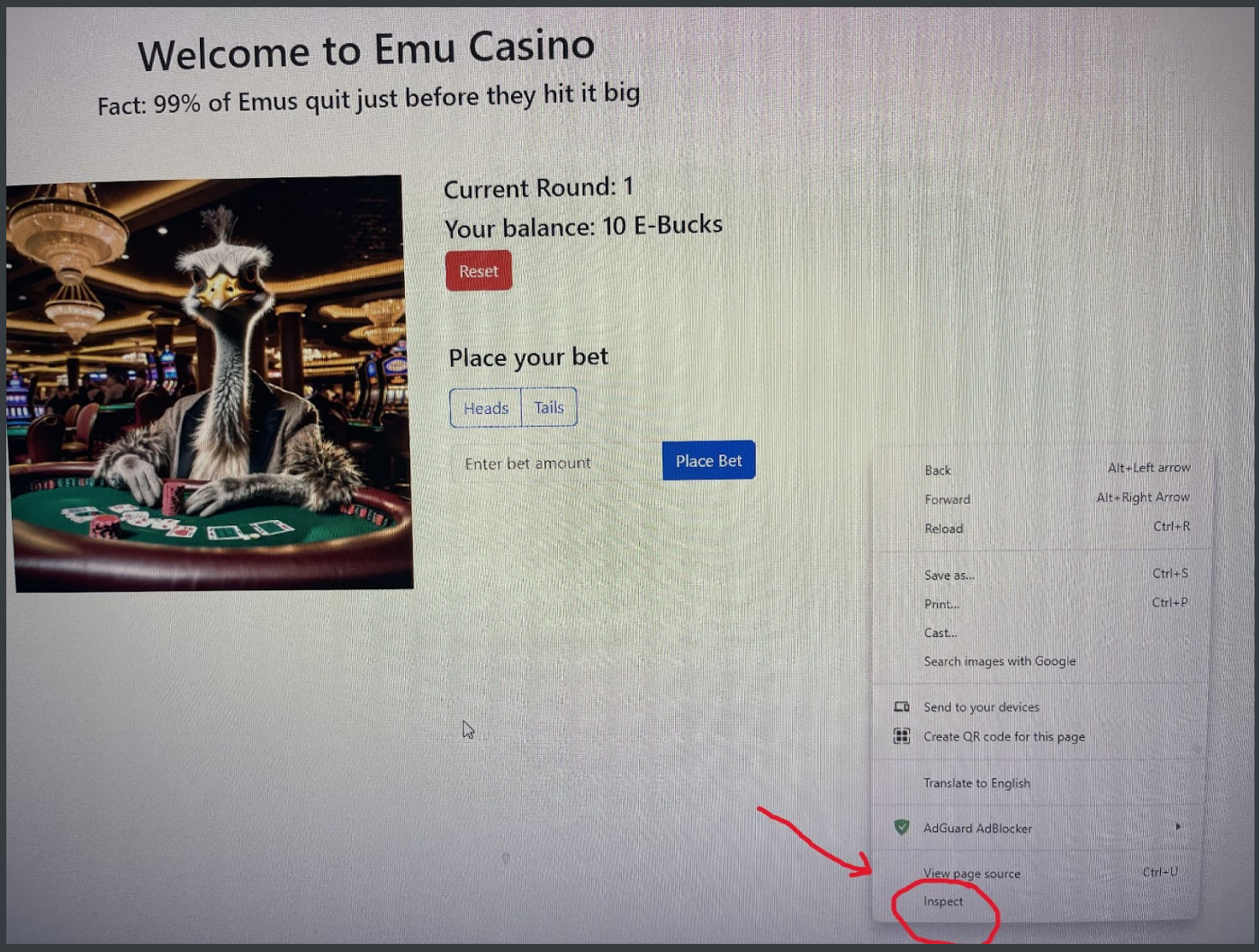
Therefore the outcome of the coin flip will be based on the “round” and “session id”.

Step 2

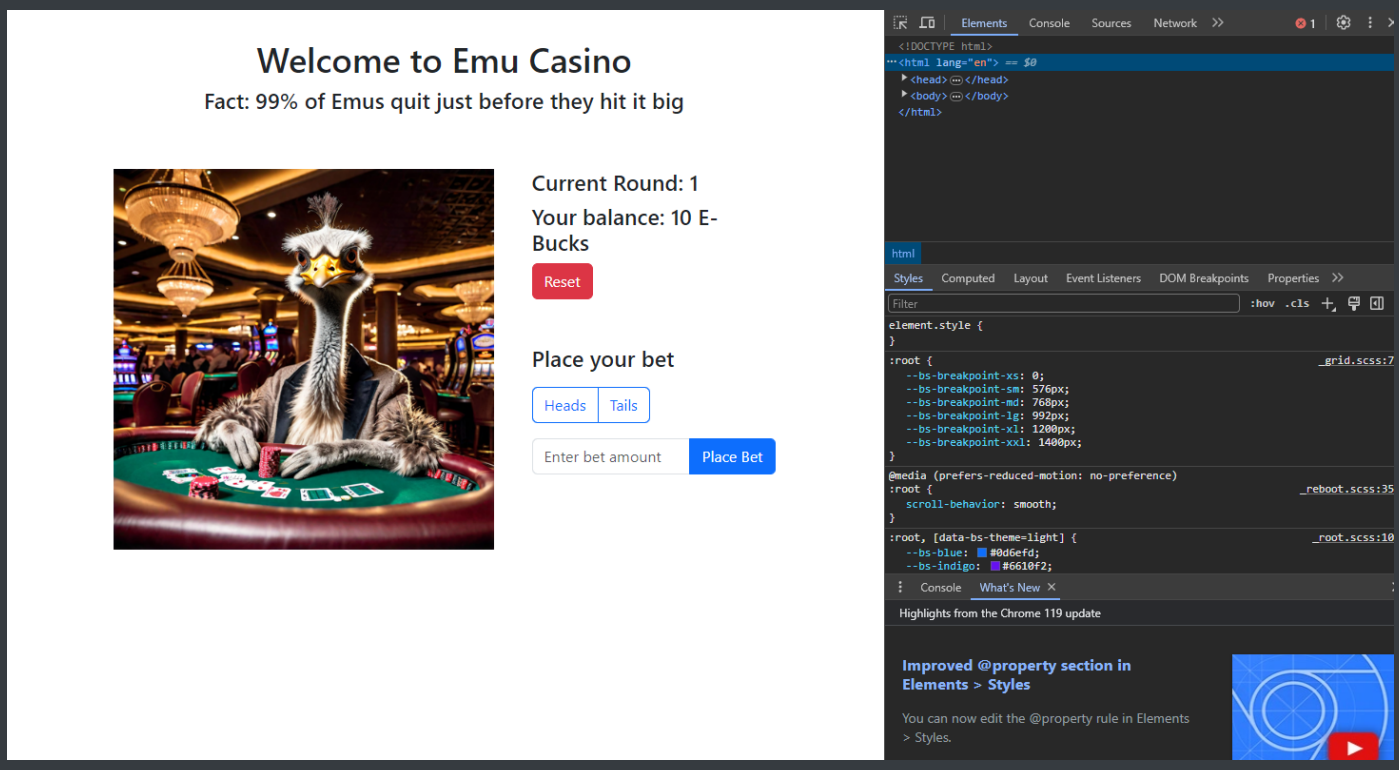
We will then need to collect this information, in order to collect the session id we will need to get the cookie of the website. We do this by going to the given website on the question “ http://34.87.251.234:3000”

Step 3

We will then need to right click on the website, Click on the “inspect” button (see attached image bellow):

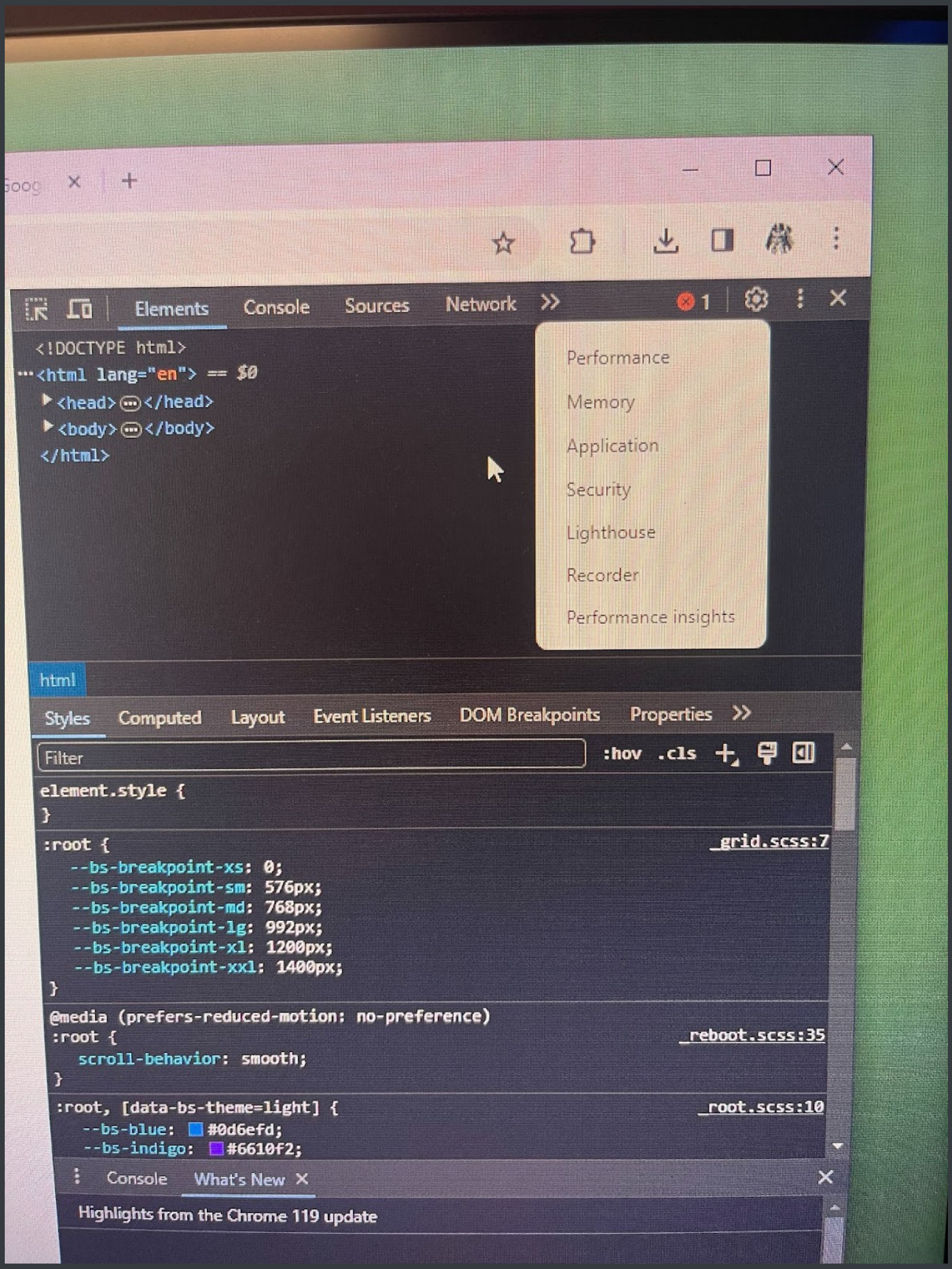


This will result in something like this afterwards.(see attachment below):



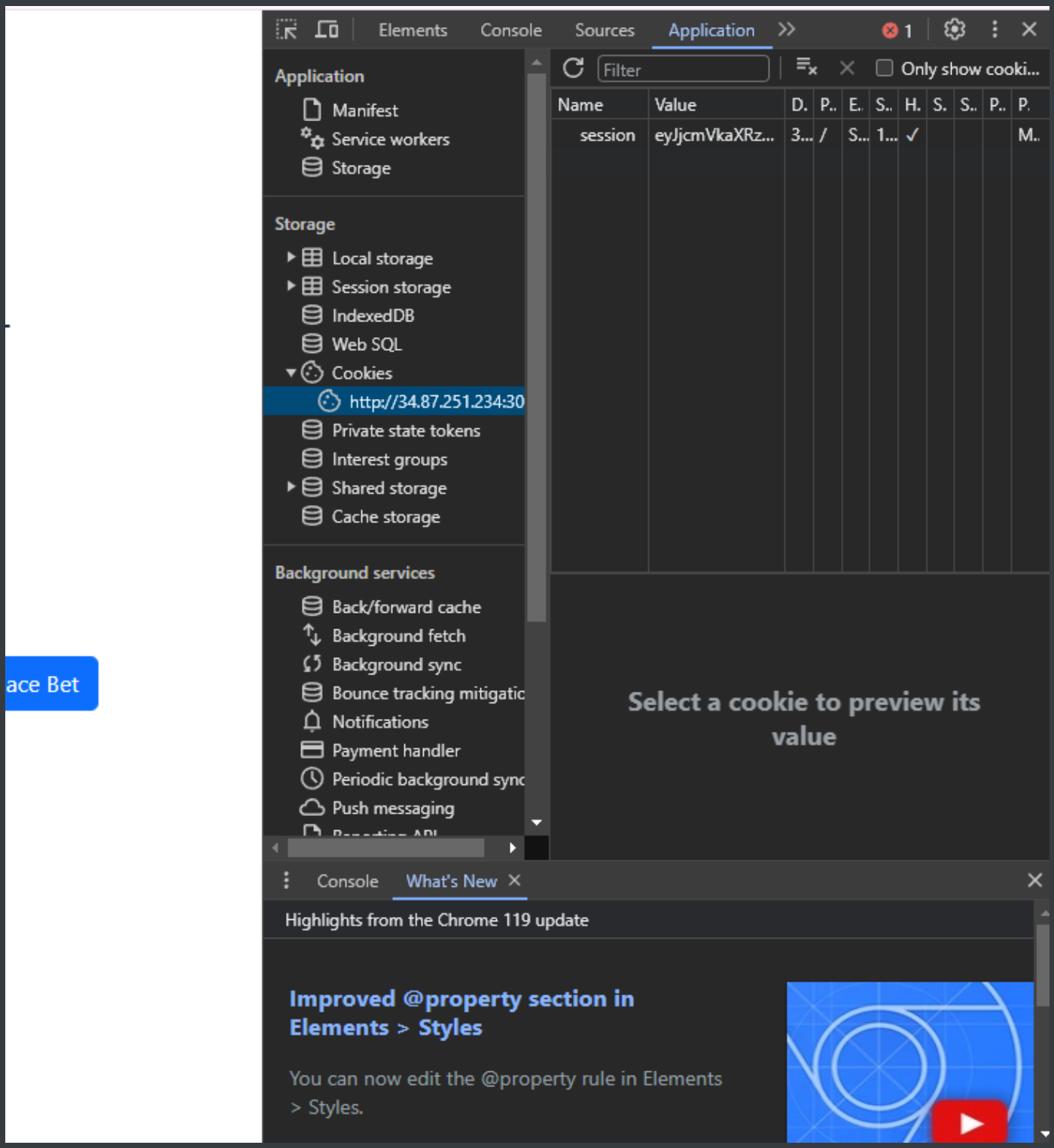
Step 4

We will then need to click on the arrow keys next to the word “network” on the top. This will result in something like this afterwards.(see attachment below):


Elements
Console
Sources
Network
1
<!DOCTYPE html>
<html lang="en"> == $0
<head></head>
<body></body>
</html>
Performance
Memory
Application
Security
Lighthouse
Recorder
Performance insights
html
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Filter
:hov .cls
element.style {
}
:root {
_grid.scss:7
--bs-breakpoint-xs: 0;
--bs-breakpoint-sm: 576px;
--bs-breakpoint-md: 768px;
--bs-breakpoint-lg: 992px;
--bs-breakpoint-xl: 1200px;
--bs-breakpoint-xxl: 1400px;
}
@media (prefers-reduced-motion: no-preference)
:root {
_reboot.scss:35
scroll-behavior: smooth;
}
:root, [data-bs-theme=light] {
_root.scss:10
--bs-blue: #0d6efd;
--bs-indigo: #6610f2;
Console
What's New
Highlights from the Chrome 119 update

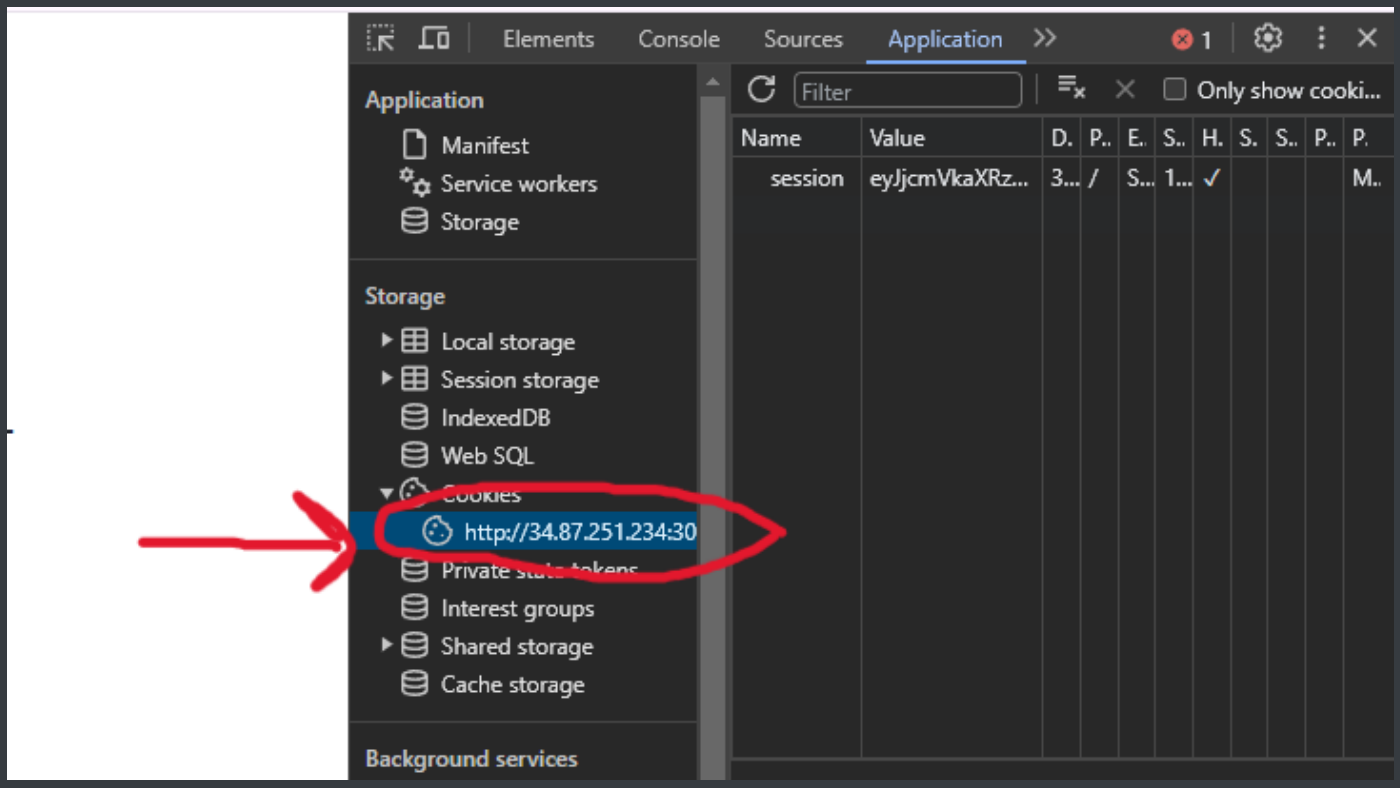
Step 5

We will proceed to click on the “Application” button. This will result in something like this afterwards.(see attachment):

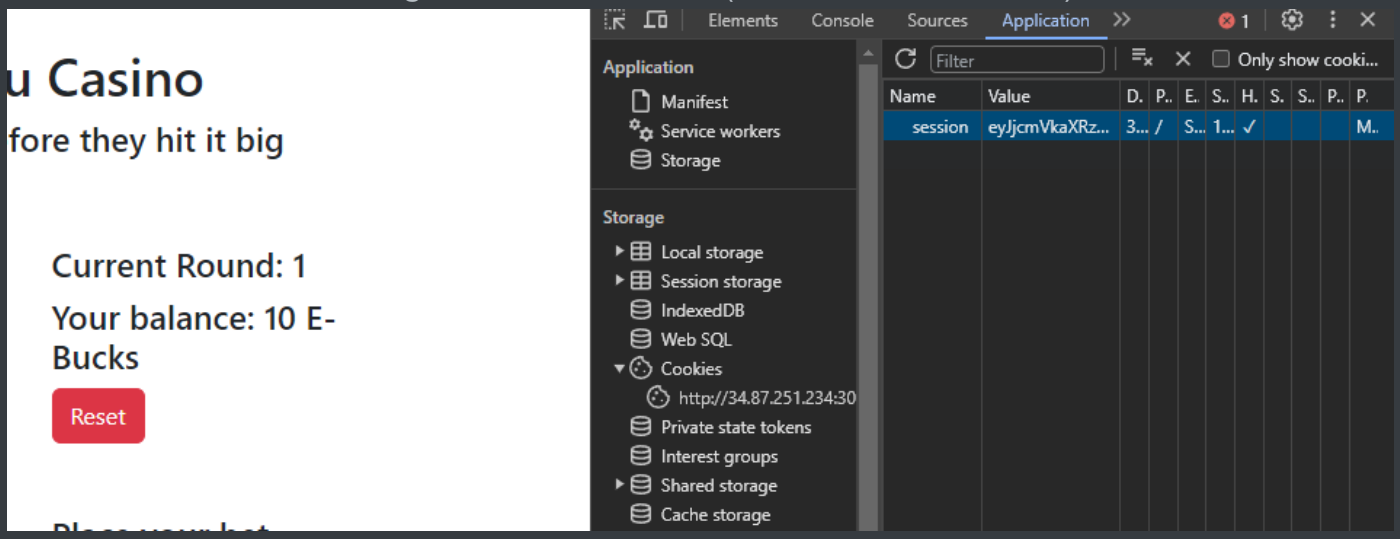


Step 6

Click on this button, this will provide you with the “value” of the website.

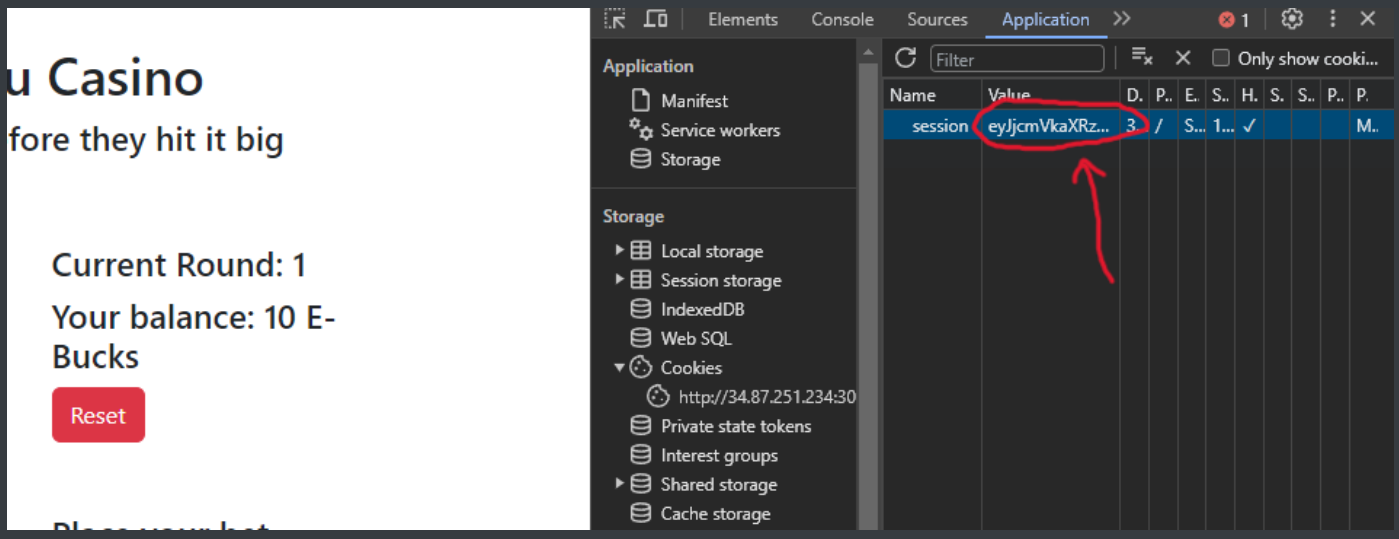


This will result in something like this afterwards.(see attachment below):



Step 7

We will then need to click on the content under the “value” button (see attached image):



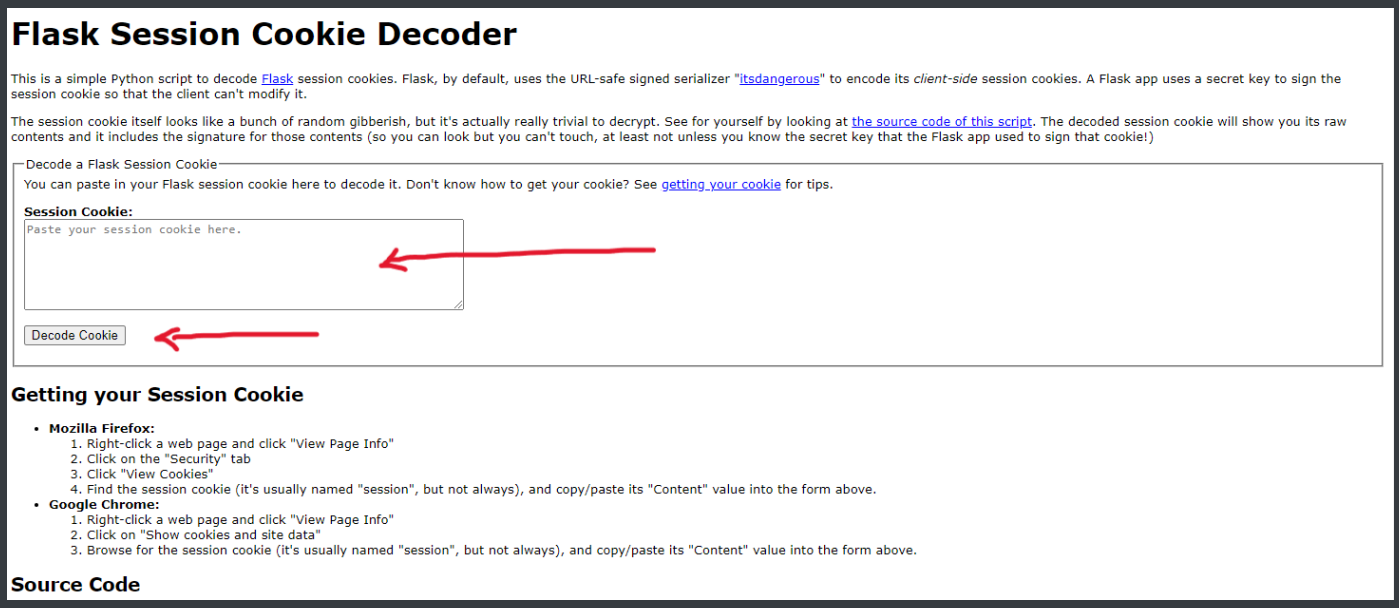
This will result in something like this afterwards. This will give the result with the cookie value of:

“eyJjcmVkaXRzIjoxMCwicm91bmQiOjEsInNlc3Npb25faWQiOiIzZTRjYWI2NDUyMDI5MGVhZWU4NDIwM2IwMjgyNTQ4ZCJ9.ZkjrIQ.VjQ5DkGJK28mF032j0laFFb4dLc”

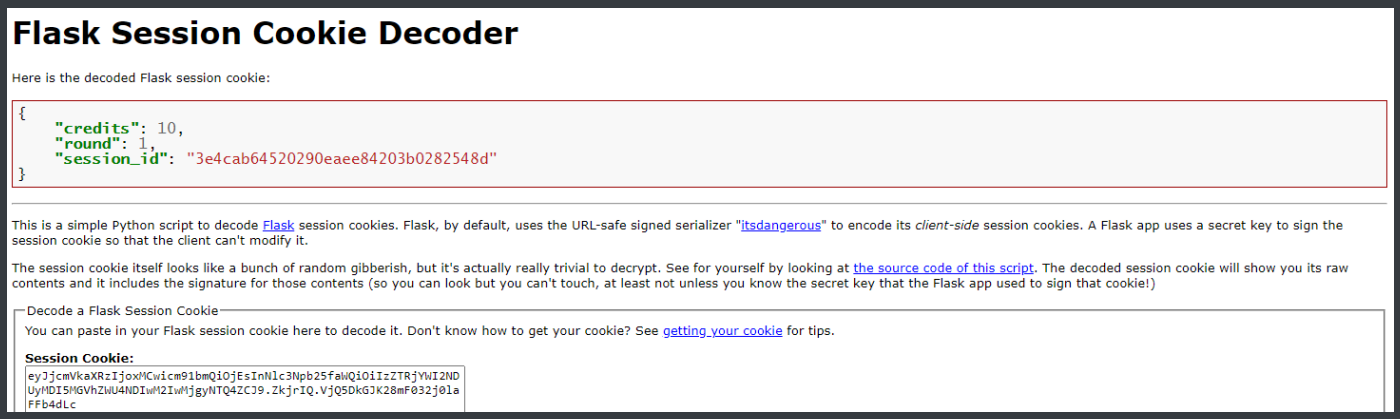
Step 8

To use this cookie value we must decode it, i find that the best tool to do this is the website “https://www.kirsle.net/wizards/flask-session.cgi”

You just simply input the cookie value in the “session cookie” input space and click the “decode cookie” button. (see attachment below):



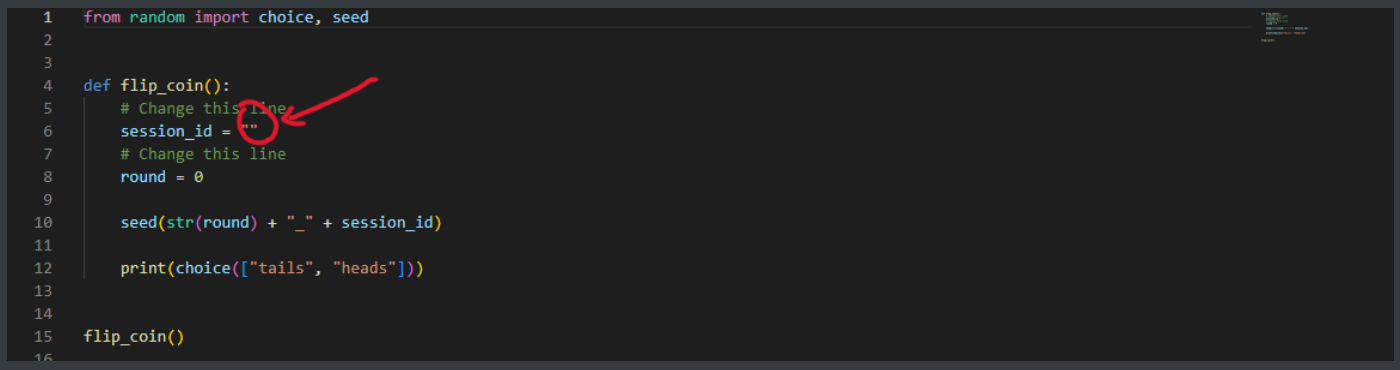
This will result in something like this afterwards.(see attachment below):



Step 9

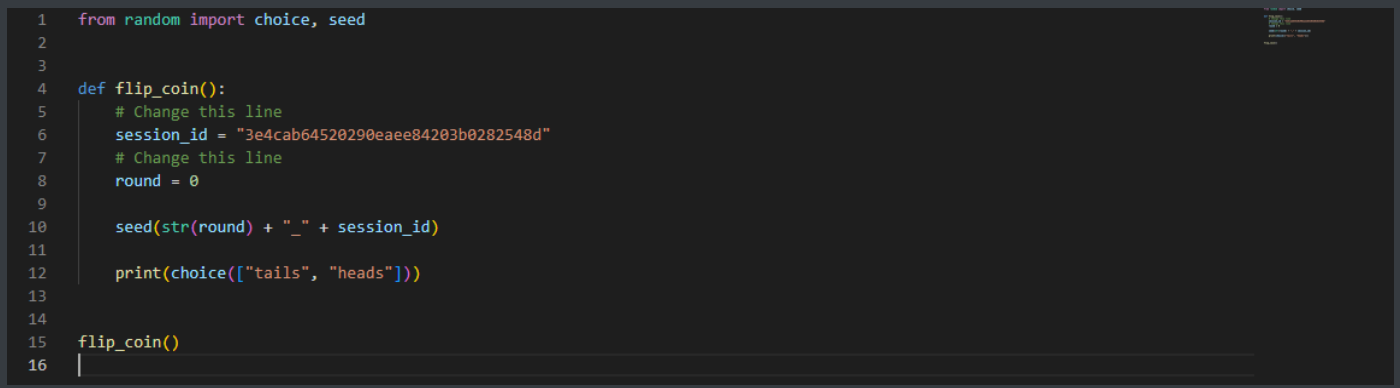
With the given session id of “3e4cab64520290eaee84203b0282548d” from step 9we can now predict the outcome of the coinflip.

We do this by adding the session id into the (“”) section within the “solution_template” That was given in the question nex to the “session_id” (see attached image):



```python
from random import choice, seed


def flip_coin():
    # Change this line
    session_id = ""
    # Change this line
    round = 0

    seed(str(round) + "_" + session_id)

    print(choice(["tails", "heads"]))


flip_coin()
```

This will result in something like this afterwards.(see attachment below):



```python
from random import choice, seed


def flip_coin():
    # Change this line
    session_id = "3e4cab64520290eaee84203b0282548d"
    # Change this line
    round = 0

    seed(str(round) + "_" + session_id)

    print(choice(["tails", "heads"]))


flip_coin()
```

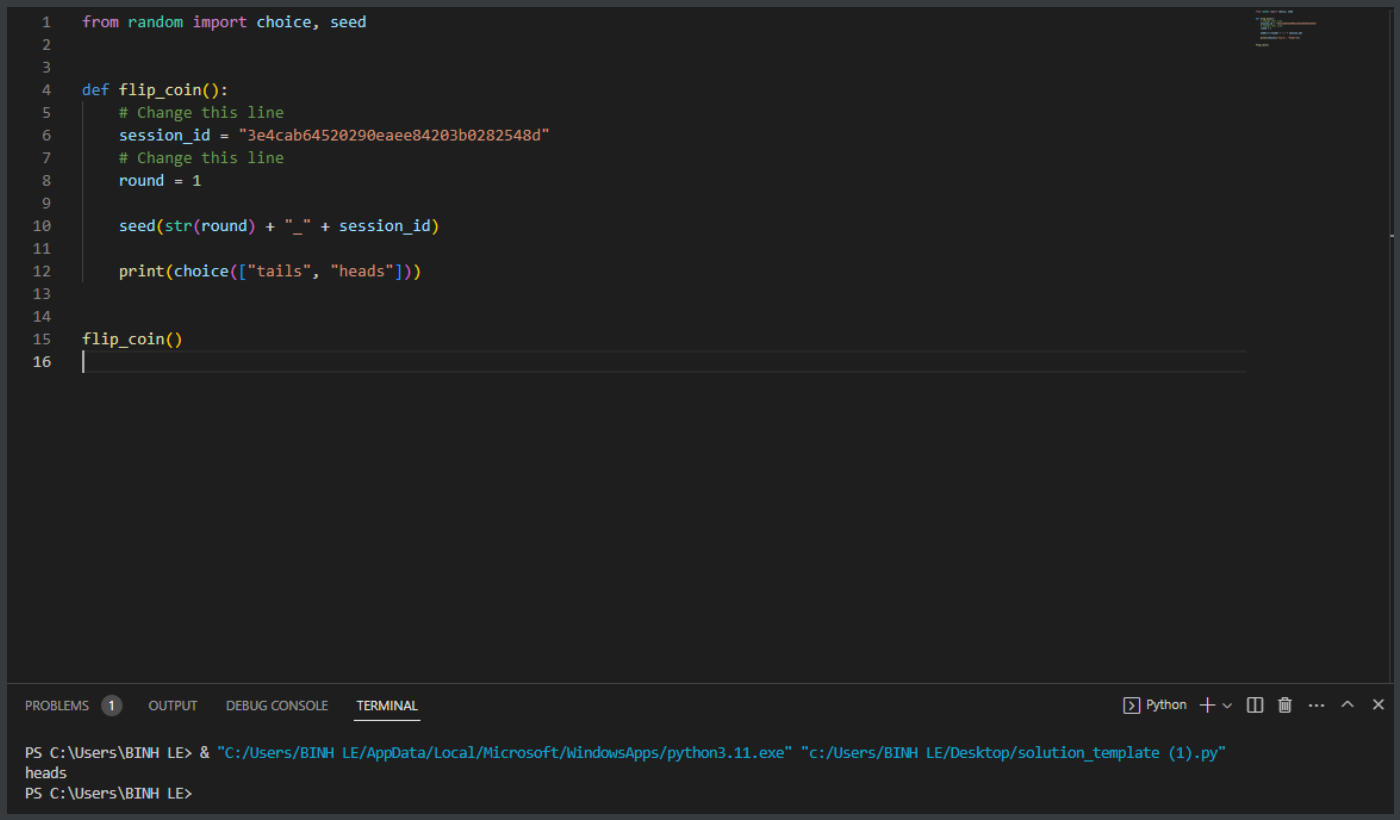
Step 10

Now that we have the session id, we need the round number, this will depend on what round you are in the coin flip website.

For example if you are in round 1 the the code will looks something like this (see attached image):

```python
from random import choice, seed


def flip_coin():
    # Change this line
    session_id = "3e4cab64520290eaee84203b0282548d"
    # Change this line
    round = 1

    seed(str(round) + "_" + session_id)

    print(choice(["tails", "heads"]))


flip_coin()
```

This will result in an answer of either head. Keep in mind It could be either head or tails, it depends on the round. (see attachment below):



Step 11

We will do this until we reach 10 thousand “E-Bucks” on the

Website. This will result in the given flag:

Result:

UWA{R0LLl111Llli1iNg_1N_C4$$$$h!11!}

Flag:

```
UWA{R0LLl111Llli1iNg_1N_C4$$$$h!11!}
```

EWT

Step 1

A clear, and detailed description.

Step 2

Step X

Flag Found

```
UWA{xxxxxxxxxx}
```

Part 2 - Forensics

Caffeinated Emus - Part 1

Step 1

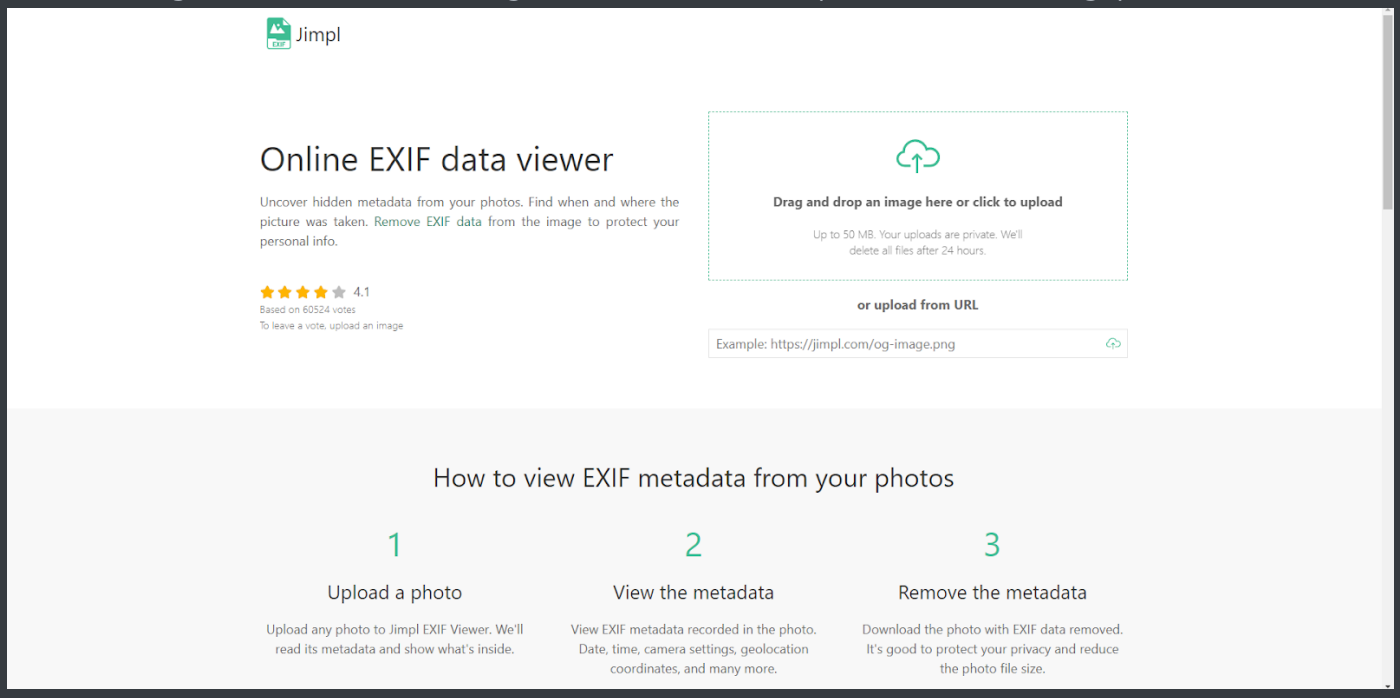
In order to acquire the flag we must need to understand the question. To find the flag we must find a way to access the metadata within the given ".png" image. We can do this through many ways, i found that the best tool would be through a website called "https://jimpl.com/results/2484d3q4wc6ankWtKnV6GmrJ?target=exif".

Step 2

Entering the given website of

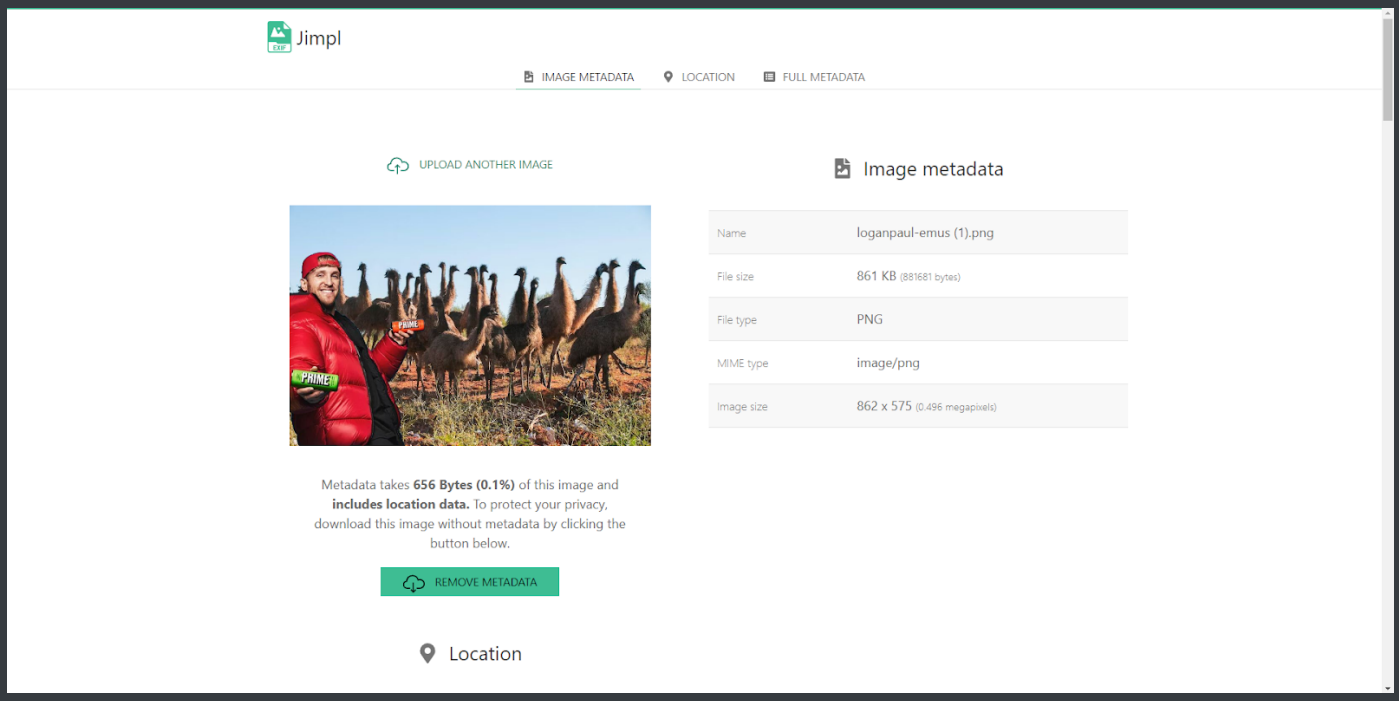
“https://jimpl.com/results/2484d3q4wc6ankWtKnV6GmrJ?target=exif”

We will be greeted with something that looks like this (see attached image):



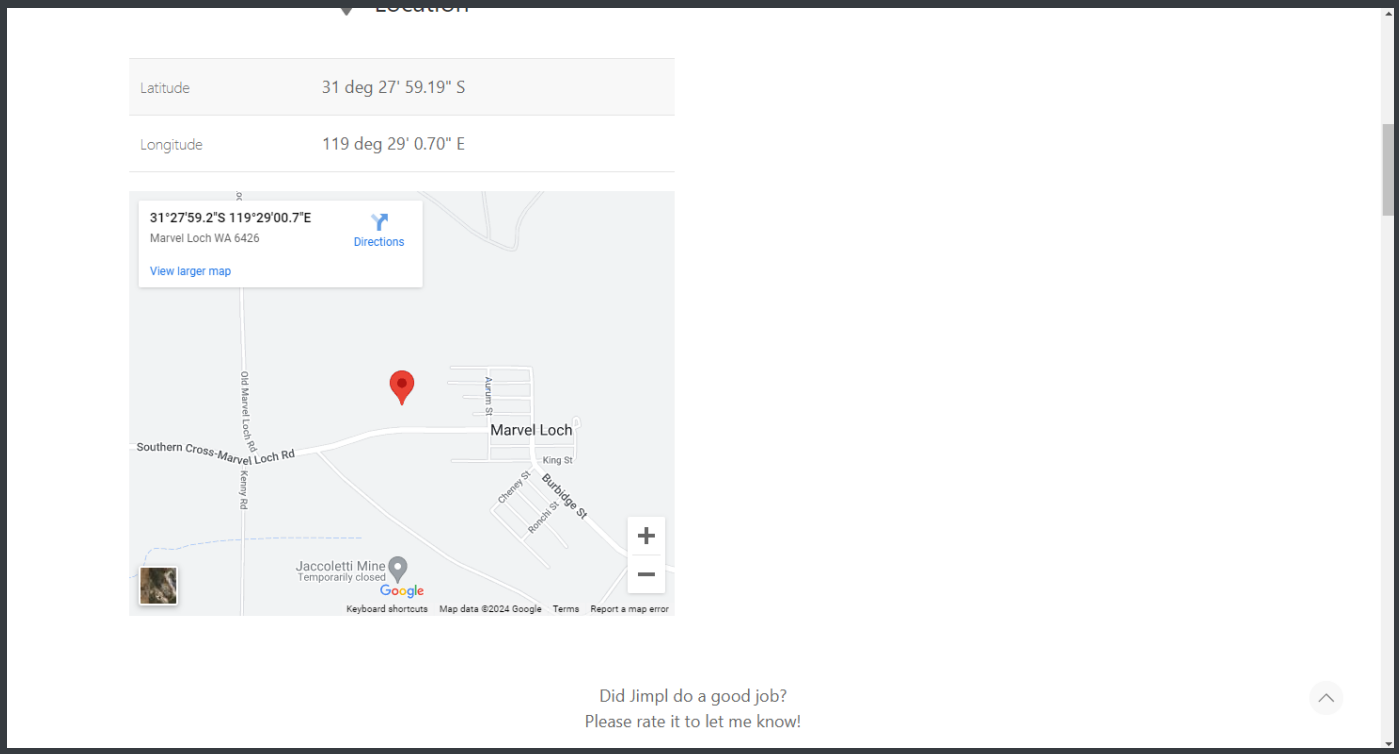
Step 3

We will then need to input the given PDF file of “loganpal_emus.png” in the “drag and drop” box. This will result in something like this afterwards.(see attachment below):



Step 4

If we just scroll down in the website from step 3, we will see something like this (see attached image):



In the given box of the map on the left we can easily identify that there is a town located near the given pinpoint, that being “Marvel Loch”.

In the original question, it was stated that the flag would be “UWA{town name}”. So gathering what was stated before the final flag for the answer would be “UWA{Marvel Loch}.

Resulting in:

Result:

UWA{Marvel Loch}

Flag Found:

```
UWA{Marvel Loch}
```

Flightless Data

Step 1

we will need to first download the html and open it on a tab, this wil allow us to download the image and save it,I will be naming mine "password.jpeg" in downloads.

we will no go into Kali Linux and put in the following which will put us in access to the dowloaded files:

```
cd Downloads
```

Step 2

we will need to then input the following, this will allow us to access the access the message. this is mentioned in the hint within the question:

```
steghide extract -sf password.jpeg
```

It will ask you for a passphrase, this is located at the bottom of the image of the secret message that being:

```
theEMusWiLLRoOld3w0oRlD
```

this will result in the following:

```
wrote extracted data to "secret.txt".
```

Step 3

next we just simply enter:

```
cat secret.txt
```

This will enter the secret message and read whats inside. this will return the following:

```
Hello my fellow Emu.
container by ru
Fortunately the hoomans arent big brain like use and would not look for
this secret message in the least significant bits of an image!

UNAIfLigHtL355_d4Ta_uNd3r_tH3_r4dAT]
```

Flag Found

```
UWA{fLigHtL355_d4Ta_uNd3r_tH3_r4dAT}
```

Ruffled Feathers

Step 1

we need fitst to download ghex within Kali Linux as said in the given hint. we do this by entering the following:

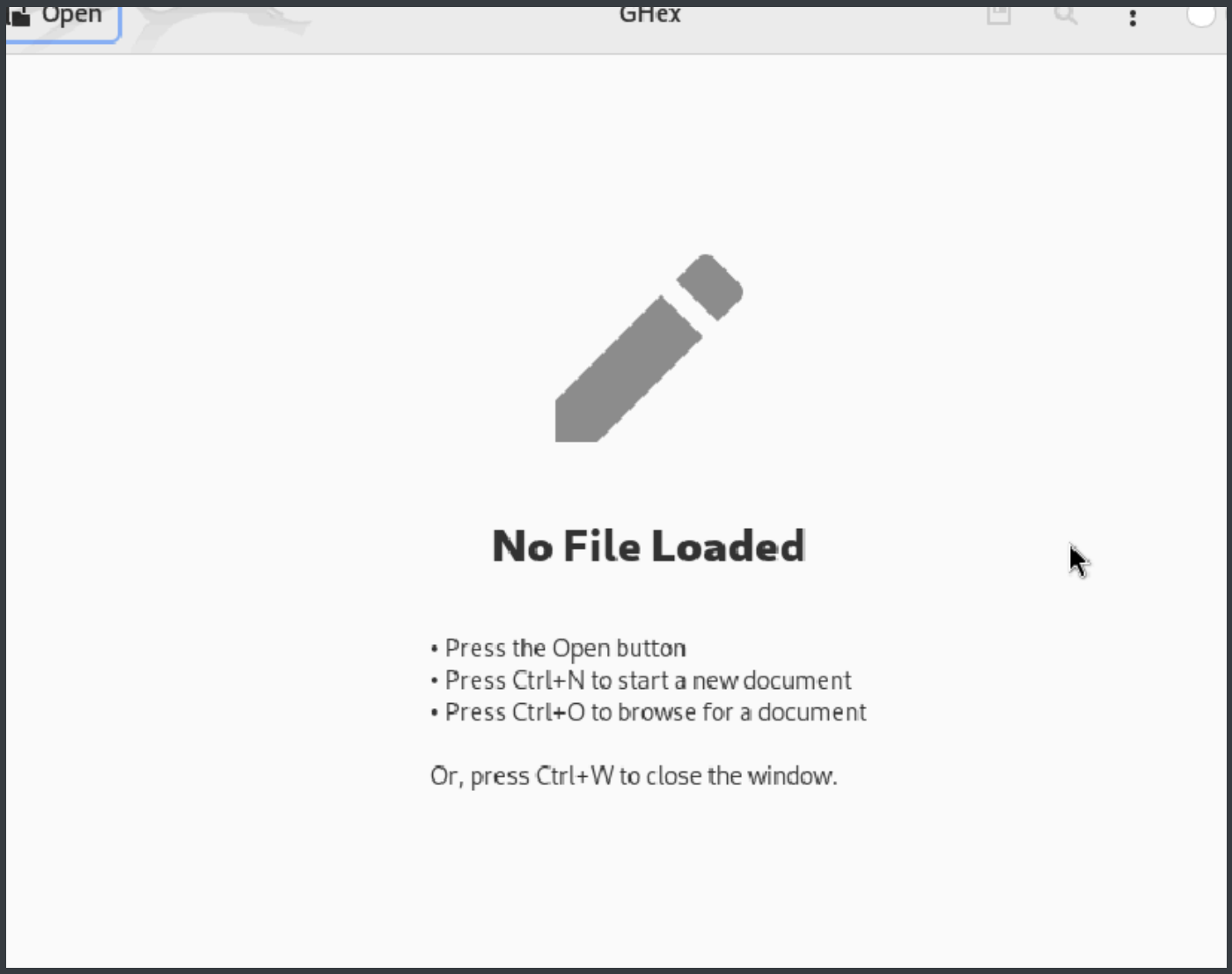
```
sudo apt install ghex
```

We will need to wait for everything to download, if an error comes up simply enter the following and re-enter the above:

```
sudo apt update
```

Step 2

once that is done simply enter "ghex" into the terminal, this will display a display like this:



Step 3

we will then need to dowload the given pdf file and click and drag it into this box. this iwll give us something like this:

```
00000000  25 50 44 46 2D 31 2E 35 0A 25 D0 D4 C5 D8 0A 36  %PDF-1.5.%.....6
00000010  20 30 20 6F 62 6A 0A 3C 3C 0A 2F 43 6F 72 72 75   0 obj.<<./Corru
00000020  70 74 32 37 32 20 20 20 20 20 20 20 0A 2F 46 69  pt272       ./Fi
00000030  6C 74 65 72 20 2F 46 6C 61 74 65 44 65 63 6F 64  lter /FlateDecod
00000040  65 0A 3E 3E 0A 73 74 72 65 61 6D 0A 78 DA 5D 51  e.>>.stream.x.]Q
00000050  B1 52 C3 30 0C DD FD 15 62 73 86 38 B6 15 3B CE  .R.0....bs.8..;.
00000060  CA 95 72 65 E3 EA 8D 63 08 89 21 5C 48 5C 92 16  ..re...c..!\H\..
00000070  7E 1F 27 6E 4A C1 1E A4 A7 27 E9 C9 32 87 37 E0  ~.'nJ....'..2.7.
00000080  70 4F F8 3F 7B 6B 49 B6 45 05 22 67 98 6B 09 F6  pO.?{kI.E."g.k..
00000090  15 04 22 2B B4 01 AD 0A 26 03 65 1B 78 A2 FB D3  .."+....&.e.x...
000000A0  21 49 51 50 37 06 53 28 BA 59 B0 BC 60 9B 94 39  !IQP7.S(.Y..`..9
000000B0  F5 87 08 F6 AE 1E DD 31 79 B6 0F D9 56 96 50 B2  .......1y...V.P.
000000C0  52 4B 3D B7 4E 39 43 65 20 95 82 19 19 1B DB D6  RK=.N9Ce .......
000000D0  85 22 44 DA FA 24 95 86 FA BE 1A A6 18 99 5A 7F  ."D..$........Z.
000000E0  FA 68 A2 3F B8 AF 24 B0 8B 1C E6 B4 1B FC 0C BF  .h.?..$.........
000000F0  23 B9 8B C1 AA 8F B0 3A 97 BF D7 73 4E 17 C9 A9  #......:...sN...
00000100  AB 8E EE 25 48 14 D4 57 63 E3 C6 9B 79 3E 72 67  ...%H..Wc...y>rg
00000110  89 08 8B E0 20 2E EF 56 5C 31 A5 0A A8 7B F2 49  .... ..V\1...{.I
00000120  04 E3 E1 60 4C B9 F2 23 29 25 33 26 72 BF 6E A0  ...`L..#)%3&r.n.
00000130  B2 5D 2F 60 E3 C9 63 B8 EB A6 57 99 74 D5 49 AF  .]/`..c...W.t.I.
00000140  84 96 6F F8 B3 2A E4 C8 42 26 98 92 A1 3E 7F 82  ..o..*..B&...>..
00000150  58 46 5E 1B 86 D1 7F 00 34 A3 6A 10 0A 65 6E 64  XF^.....4.j..end
00000160  73 74 72 65 61 6D 0A 65 6E 64 6F 62 6A 0A 34 20  stream.endobj.4
00000170  30 20 6F 62 6A 0A 3C 3C 0A 2F 54 79 70 65 20 2F  0 obj.<<./Type /
```

we will only be focusing on the right section of it, comparing the corrupted file to a normal one we can see that in the corrupted we can see the following:

```
%PDF-1.5.%.....6
 0 obj.<<./Corru
pt272       ./Fi
lter /FlateDecod
```

Step 4

caomparing it to a normal PDF we can see that instead of "Corrupt272" it has "Length 5 0 R" (see attachment):

```
%PDF-1.3.%......
......4 0 obj.<<
/Length 5 0 R /
Filter /FlateDec
```

step 5

knowing this to fix the corrupted PDF we must change the "Corrupt272" to "Length 5 0 R", this will fix the PDF completly and will give you the output:

Flag

```
UWA{uNrUffLed_p}
```

Emu in the Shell

Step 1

A clear, and detailed description.

Step 2

Step X

Flag Found

```
UWA{xxxxxxxxxx}
```

Part 3 – Linux and Networking

EMU Hack#1 - Backdoored

step 1

As said in the question we first must use the "nmap" command to scan. In order tosee the message we must create a code that can access a port to access. We will aslo need to use th -Pn option as well as the given range, which is between 61000-61500 with the server at the end. this is a lot but if we put all of it together it should look like this:

```
nmap -Pn -p 61000-61500 34.116.68.59
```

this will take a while (around 40 seconds for me). it will spit out a bunch of things but the msot mportant part would be the given port of 61337:

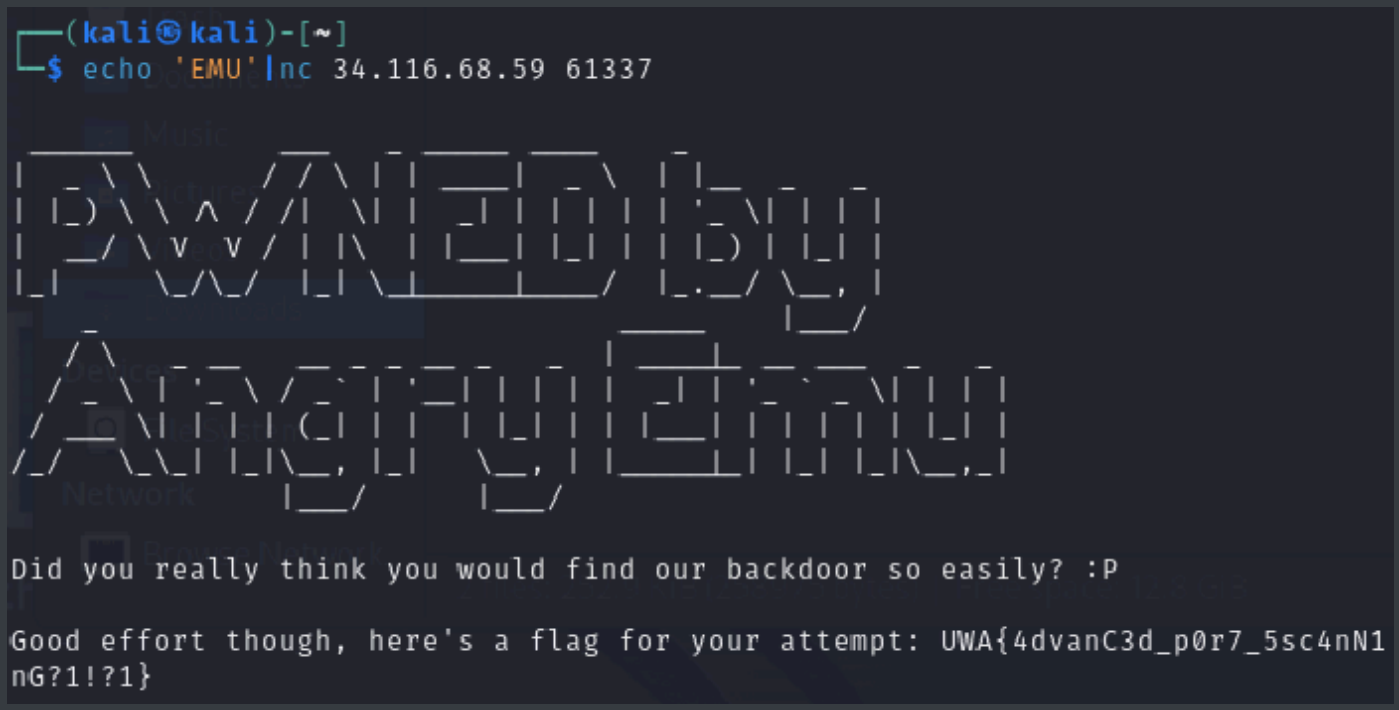
```
PORT      STATE SERVICE
61337/tcp open  unknown
```

Step 2

we will then need to send a message through the port that we found using "nc", having trouble i utilised researched on google and came to the following:

```
echo 'EMO'|nc 34.116.68.59 61337
```

this will give us the following as a response:



This looks about right as it is pwered by the "angry Emus", just like the one in the question.

Flag

```
UWA{4dvanC3d_p0r7_5sc4nN1nG?1!?1}
```

EMU Hack #2 - Git Gud

Step 1

first we must figure out how to get the access to the file using "git". this was quite challenging for me so i used ChatGPT and it came out with this:

```
git clone http://34.116.68.59:8000/emu.git
```

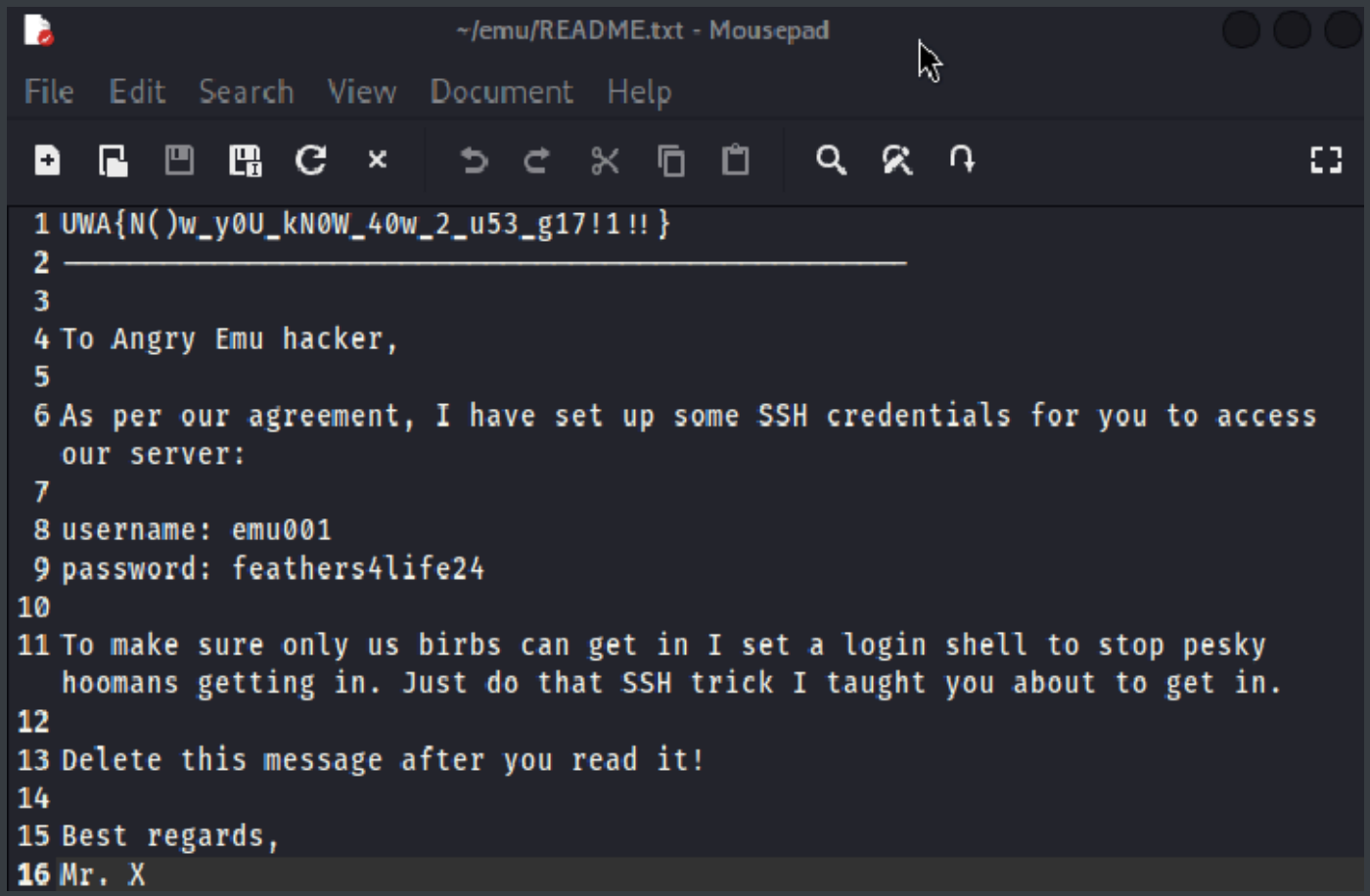
by using "clone" we are trying to clone the already existing Git repository. Once we put this in it will return this:

```
┌──(kali㉿kali)-[~]
└─$ git clone http://34.116.68.59:8000/emu.git
Cloning into 'emu' ...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

step 2

Step 1 would have downloaded a file withon your device, if you are having trouble finding the file it would be easier if you typed "ls -al", this will provide you with files that you have and when they were downloaded. the name of the dowloaded would be "emu" or "README.txt"

once located open the file and within should be:



Flag

```
UWA{N()w_y0U_kN0w_40w_2_u53_g17!1!!}
```

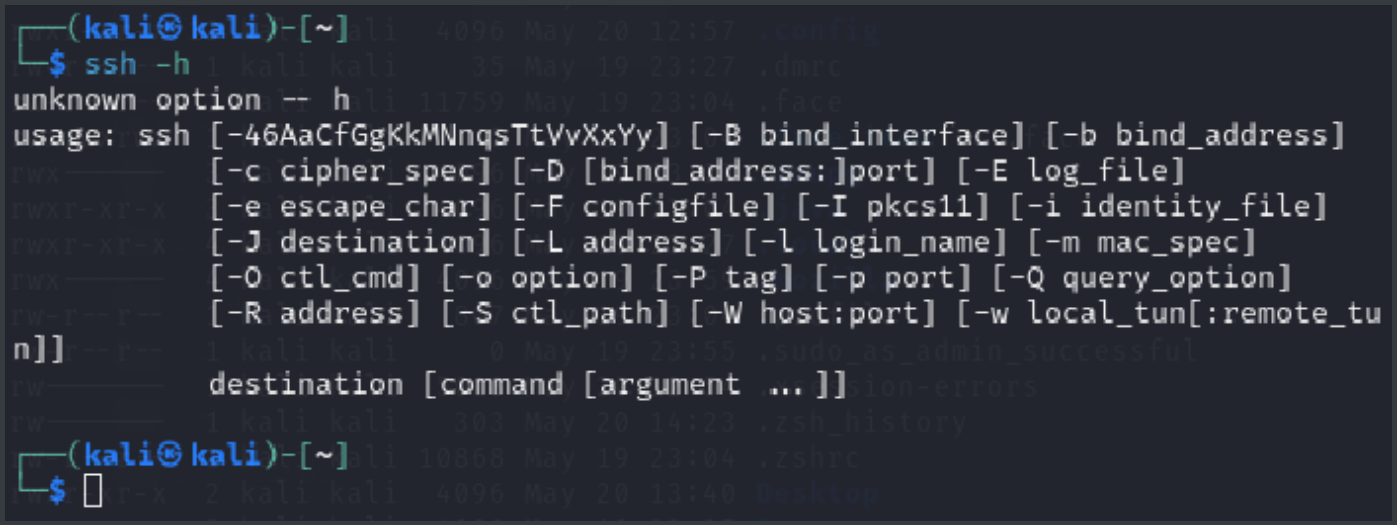
EMU hack #3 - SSH Tricks

Part 1

as mentioned we will have to do a SSH command, a bash would be thebest option as any other command like SSH will cause corruption, so we cant use it, the only solution would be:

```
ssh -h
```

this will give us the following:



part 2

no to create the bash, we will be using the port 2022 and the hosted server to do this, it will looks somthing like:

```
ssh -p 2022 -t emu001@34.116.68.59 'bash -i'
```

once we input this a passphrase is asked, this will b the one from last question, that being "feathers4life24"

step 3

now we will need to find the name of the given image, just like last question we will enter "ls -al", this will give us all the files, (see attached image):

```
drwxr-xr-x 1 root root   4096 Mar 10 12:03 .
drwxr-xr-x 1 root root   4096 Mar 10 12:02 ..
-rwxr-xr-x 1 root root   6121 Mar 10 12:02 .profile
-rw-r--r-- 1 root root    313 Mar 10 12:02 note_to_angry_emu_hacker.txt
-rw-r--r-- 1 root root 486105 Mar 10 12:02 top_secret.png
emu001@e4b74d8b74d2:~$
```

step 4

Part 4 – Vulnerabilities

Feathered Forum – Part 1

Step 1

First we must click on the given hint within the question and identify what is needed to authenticate logins. Looking at the given code it is seen that the authentication relies on the “username cookie” and whether it matches with an existing username or not.

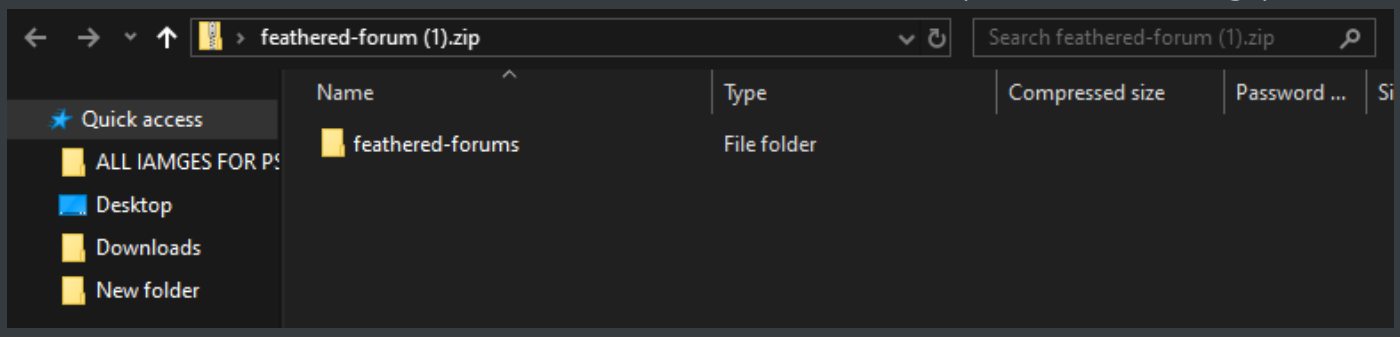
This can be identified here (see attached image):

```
    emu_usernames = [account["username"] for account in EMU_USERS_ACCOUNTS]
    # Just check that the "username" cookie matches an existing user
    if not request.cookies.get('username', None) in emu_usernames:
        return redirect(url_for('index'))
```

Step 2

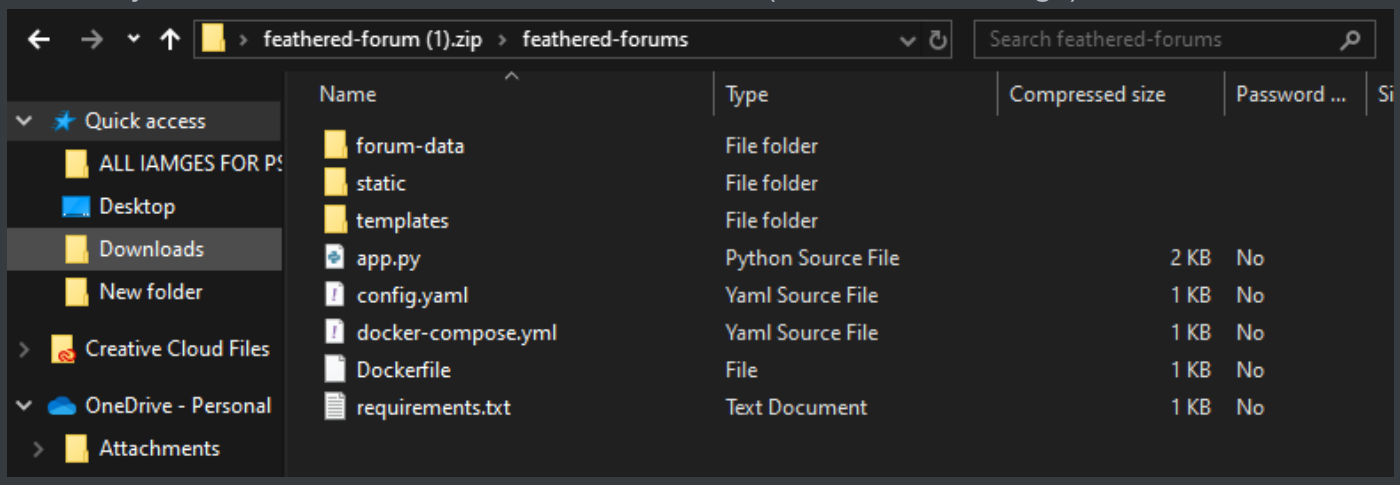
We will then need to figure out how we can access the already existing username as well as make a “username” cookie to match with the existing username.

To gain access to the already existing username we open the given “application code” that was given in the question. Opening it we are greeted with something like this, keep in mind this is a Windows device, but the file will look the same on IOS (see attached image):



Step 3

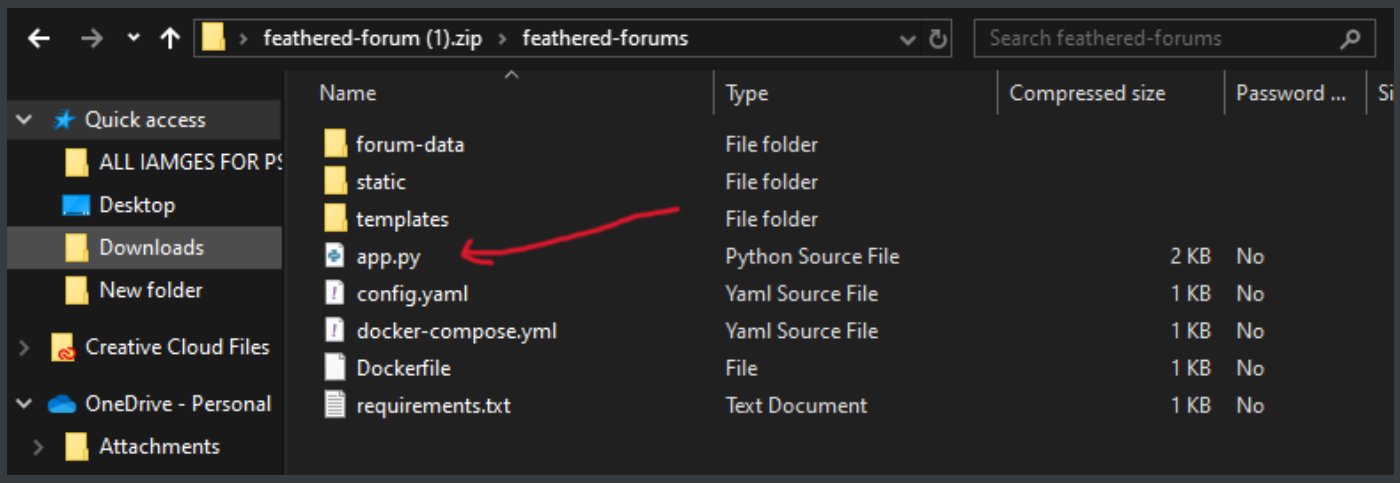
We will open the file and will see something like this, keep in mind this is a Windows device, but every file name will look the same on IOS also (see attached image):



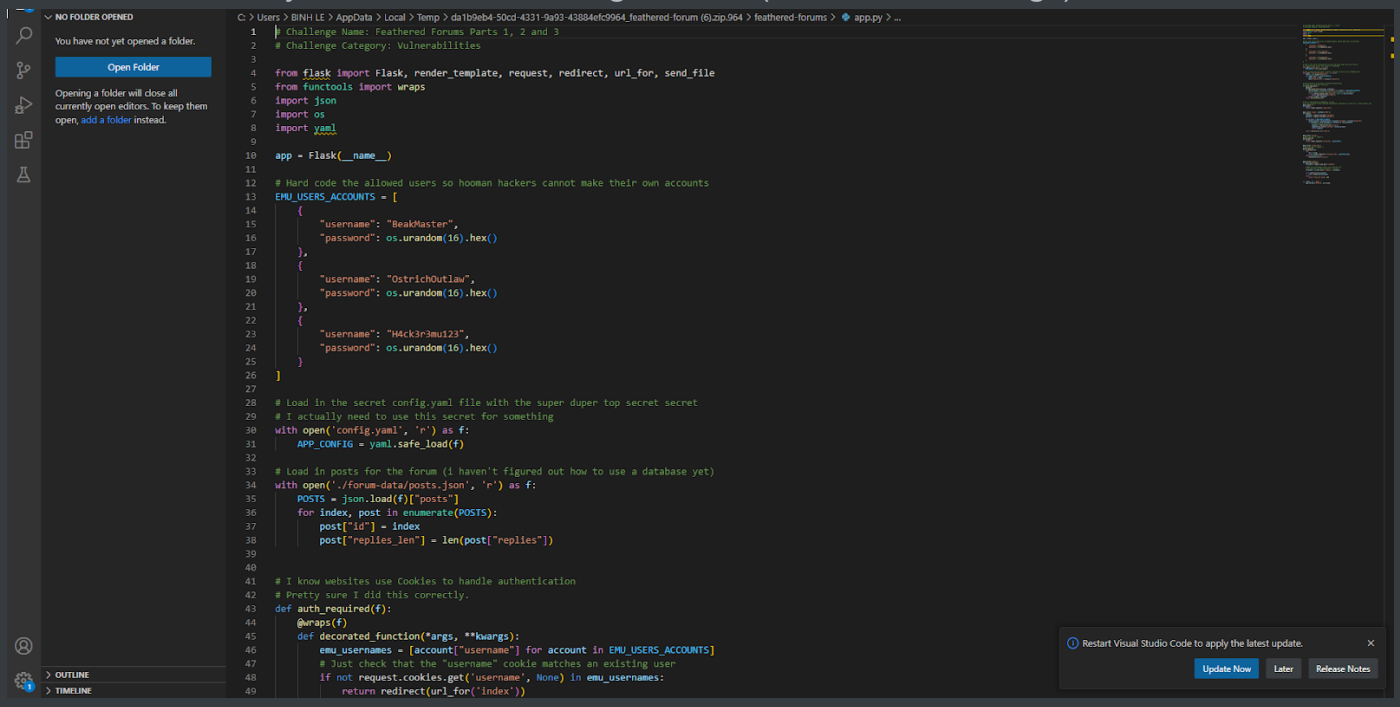
Step 4

We will then proceed to open the “app.py” file with any software that supports python. VScode would be recommended and will be the software I will personally use in the explanation.

(see attached image):

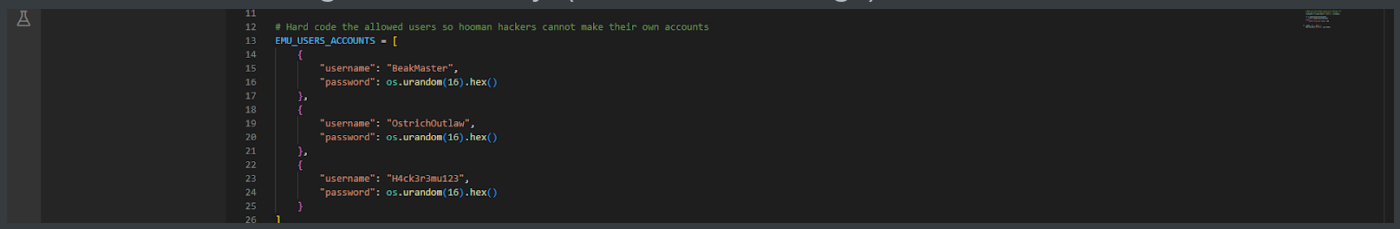


Once that is done you will see something like this (see attached image):



Step 5

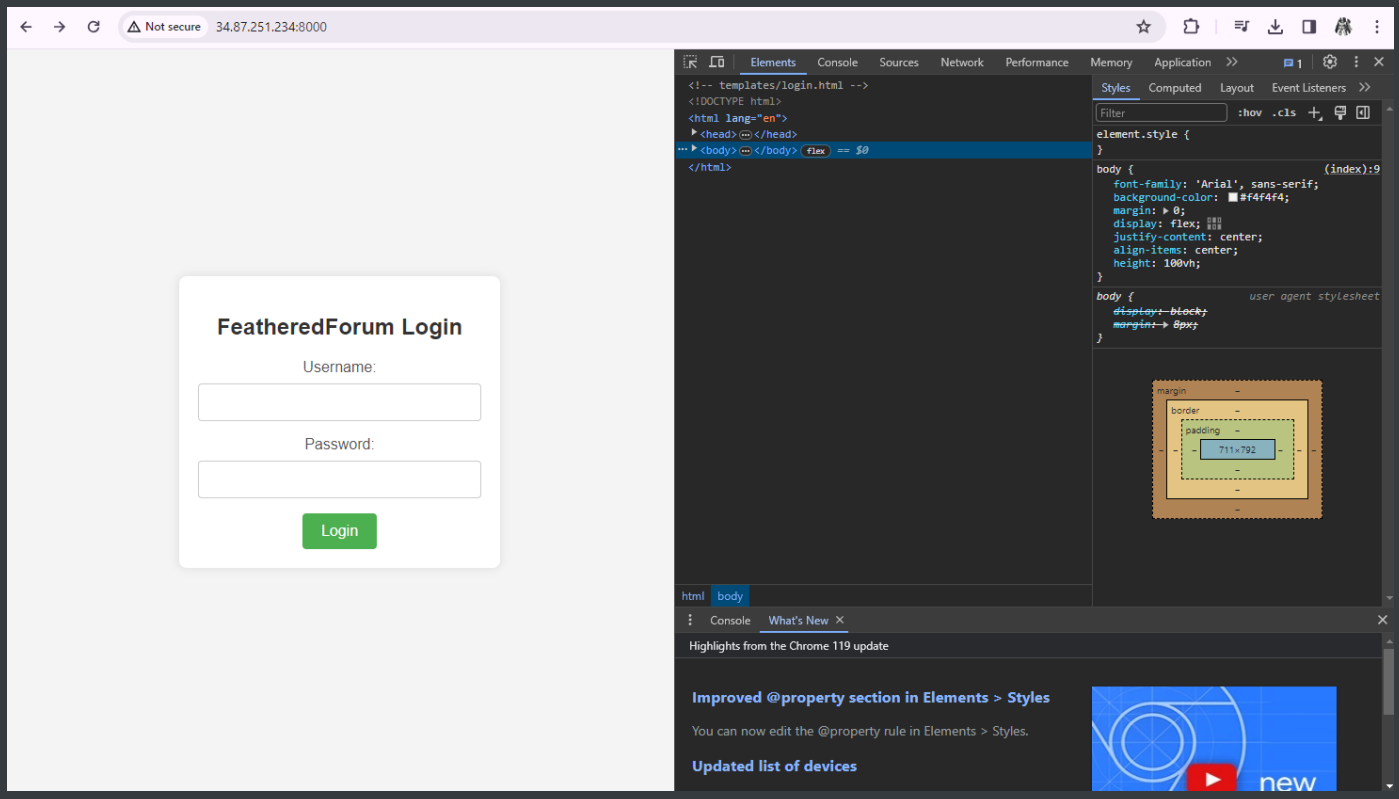
Without needing to scroll down we can already see that there is already a display of existing usernames of existing users already. (see attached image):



We will use this to help us to match it with the “username” cookie.

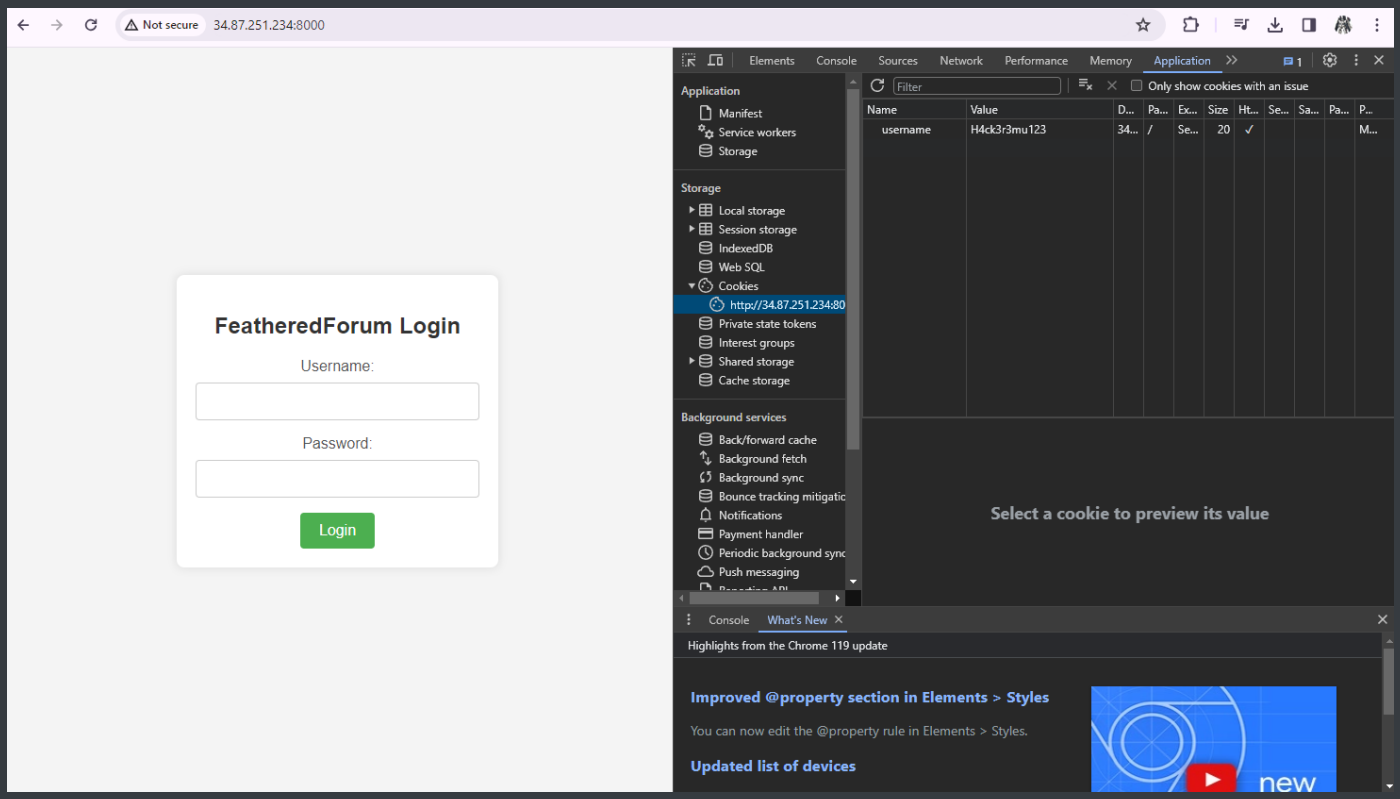
Step 6

We will then need to go to the website that the question gave “http://34.87.251.234:8000”. We will the need to right click on the website once in and select on the “inspect” button, it will look like this (see attachment):



Step 7

We will then proceed to click the “application” button in the top bar and click on the drop down bar under “cookies”. This will give us something like this (see attachment):

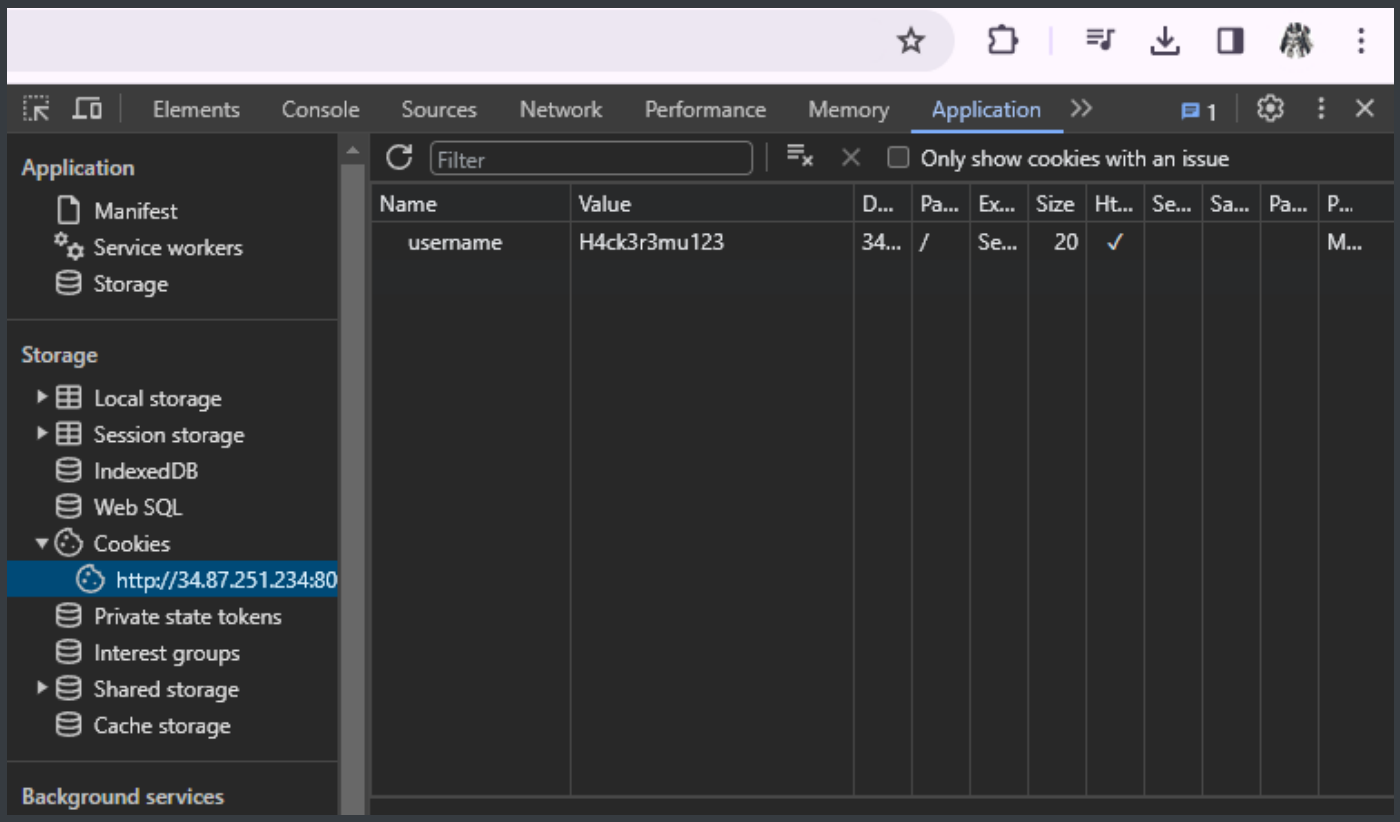


Step 8

We will then need to replace the value/word within the “name” bar with “username.

We will also need to replace the value within the “value” bar with any username of an existing user from the python code, i will use “H4ck3r3mu123”.

The final result should look like this:



Then proceed to click enter.

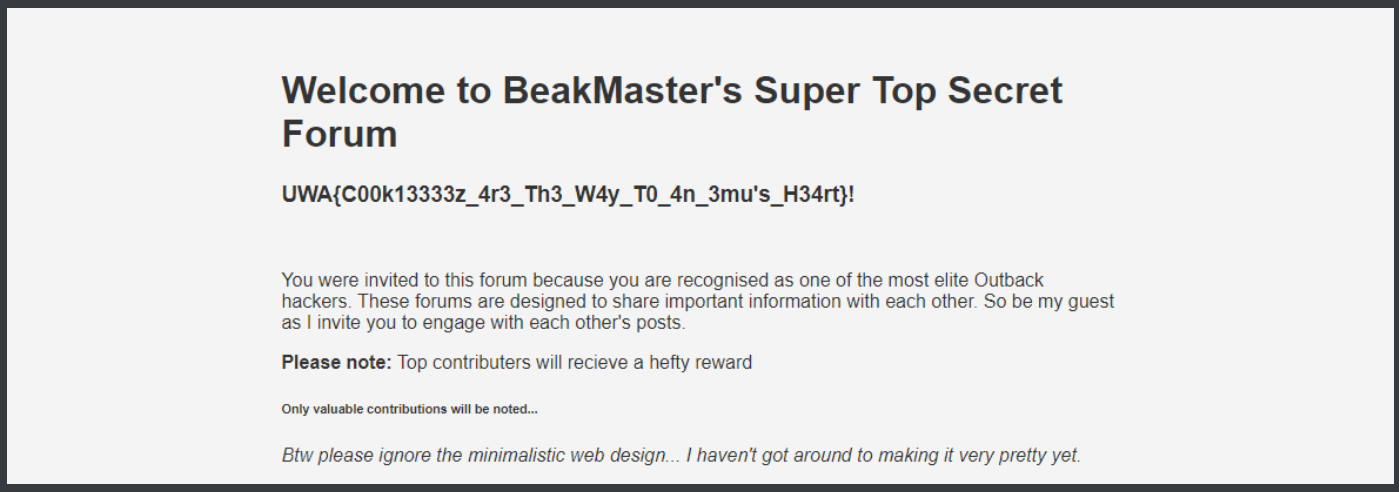
This is us creating a username cookie, by putting an existing username in the “Value” bar we are matching it with an existing one, which is that.

Step 9

Simply click on the search engines search bar and add “/forum” to the URL. This is mentioned in the original question.

The final URL on the search engine would look like this “http://34.87.251.234:8000/forum”

Once in you will see something like this:



We are now in the forum for all the juicy stuff.

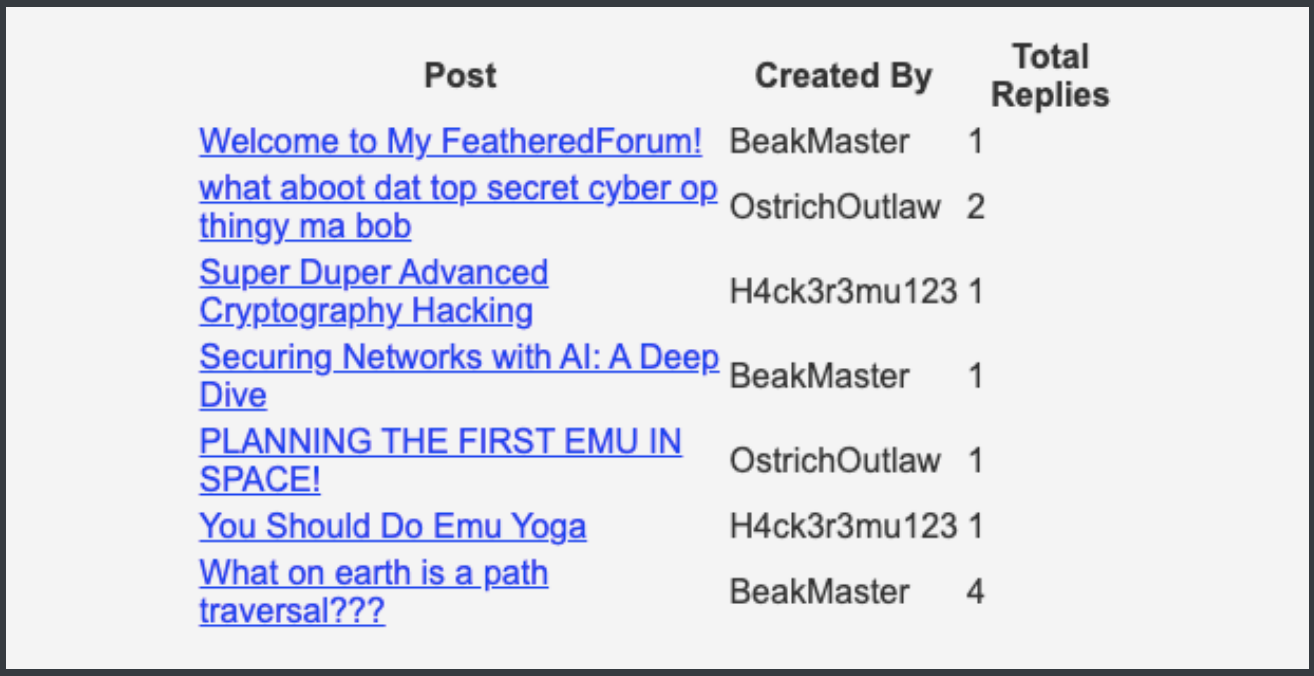
Flag:

```
UWA{C00k13333z_4r3_Th3_W4y_T0_4n_3mu's_H34rt}
```

Feathered Forum – Part 2

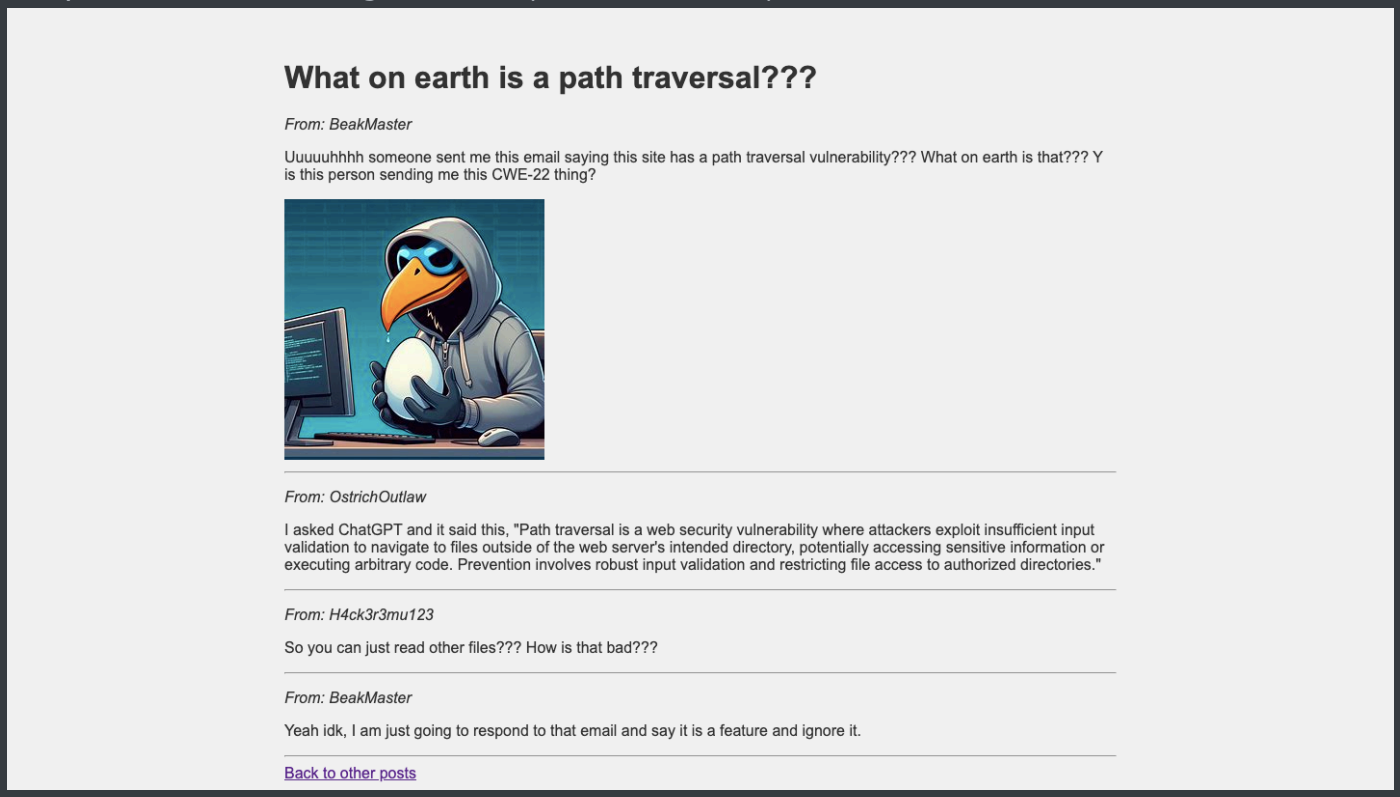
Step 1

we can identify that the once we are in the forum in part 1, that there are chats below. once we are given the flag we can scroll down to the different chats that are available. (see attachment):



Step 2

we will then proceed to click on the last post "What on earth is a path traaversal???" this will provide us with the given chat (see attachment):



What on earth is a path traversal???

From: BeakMaster

Uuuuuhhhh someone sent me this email saying this site has a path traversal vulnerability??? What on earth is that??? Y is this person sending me this CWE-22 thing?

From: OstrichOutlaw

I asked ChatGPT and it said this, "Path traversal is a web security vulnerability where attackers exploit insufficient input validation to navigate to files outside of the web server's intended directory, potentially accessing sensitive information or executing arbitrary code. Prevention involves robust input validation and restricting file access to authorized directories."

From: H4ck3r3mu123

So you can just read other files??? How is that bad???

From: BeakMaster

Yeah idk, I am just going to respond to that email and say it is a feature and ignore it.

Back to other posts

Step 3

from this, if we read we can easily identify that the CWE number is "22", therefore the flag would be UWA{CWE-22}

Flag Found

```
UWA{CWE-22}
```

Feathered Forum - Part 3

Step 1

A clear, and detailed description.

Step 2

Step X

Flag Found

```
UWA{xxxxxxxxxx}
```

Emu Apothecary

Step 1

A clear, and detailed description.

Step 2

Step X

Flag Found

```
UWA{xxxxxxxxxx}
```