IBOH 2025 Writeup

Team: Hokkien mee is Bred

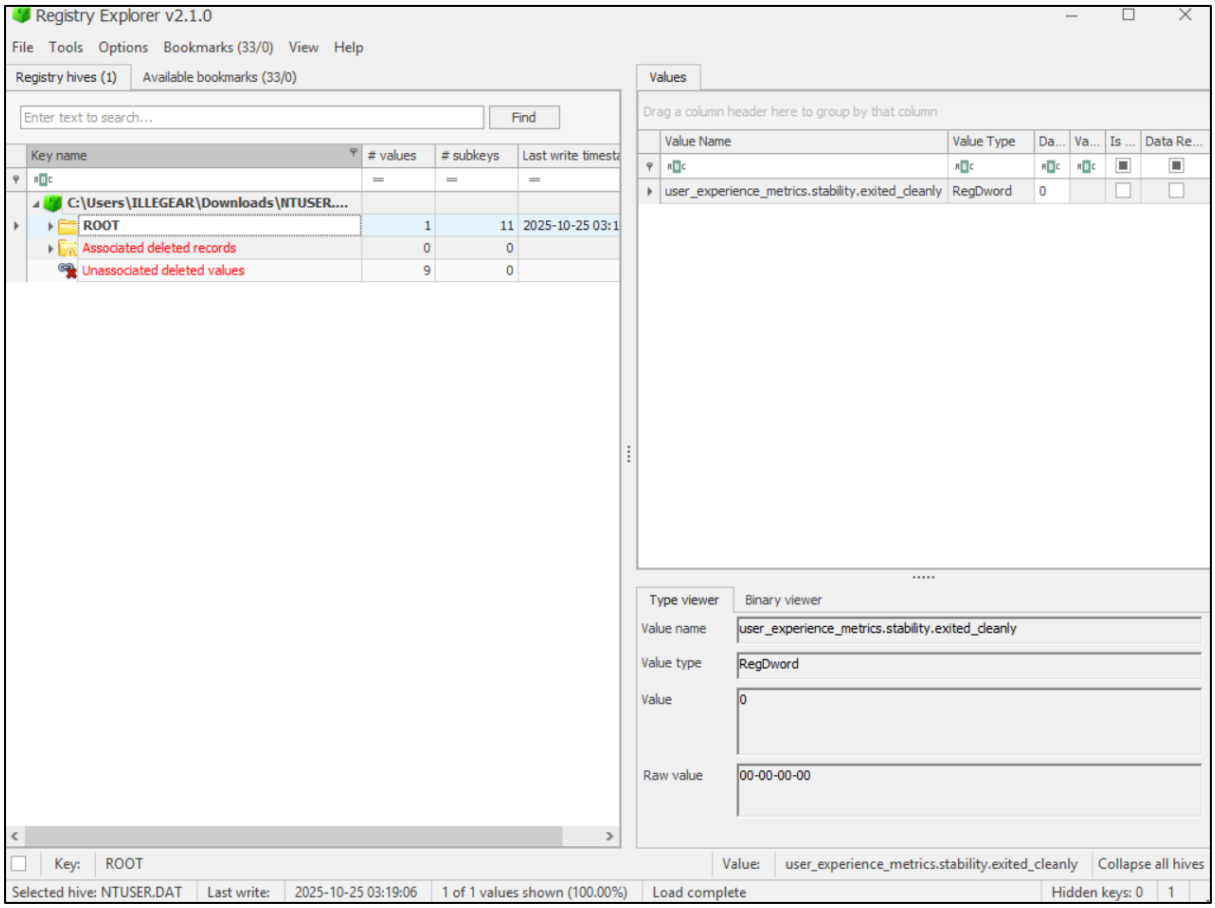
Written by: M4yb3

Table of Contents

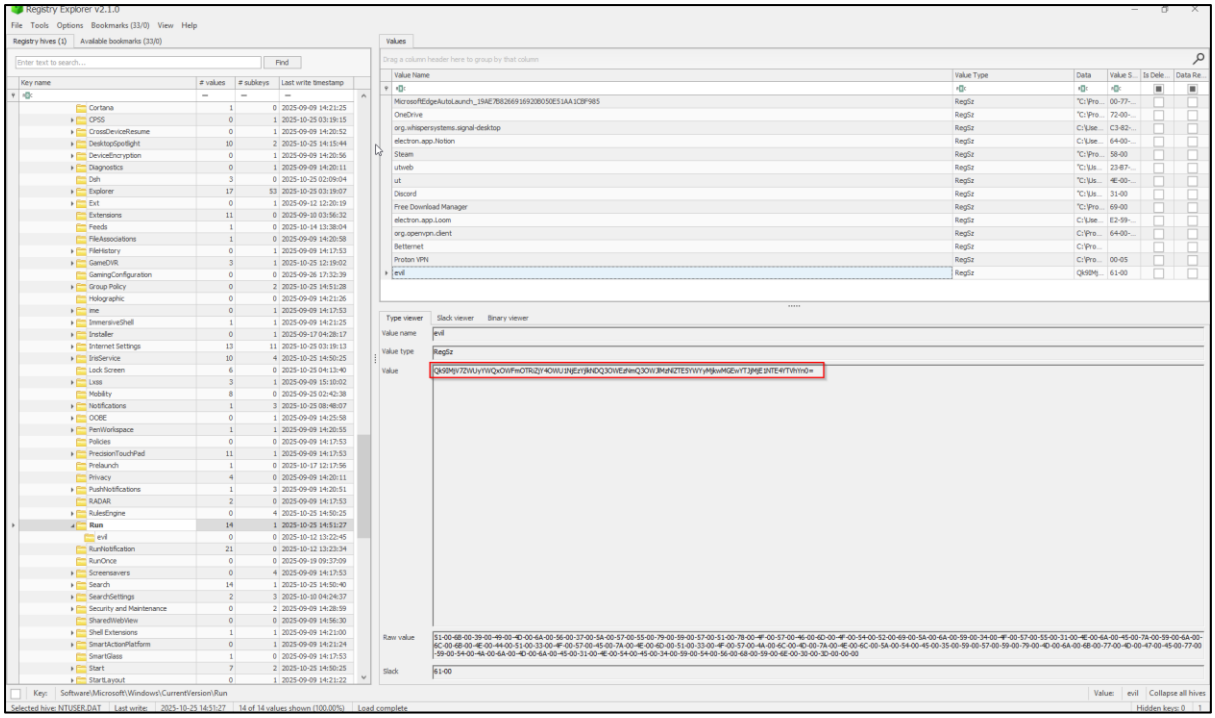
Forensic

Title: Hive

Description: Your manager bursts in, phone still buzzing: "The NSA hacked us, son. Find their artifact, get them, lad.". AV cleaned the obvious files and we rebooted, but the outbound beacons come right back. You've bagged a few critical system files for analysis, time to find the persistence and kick the intruder out.

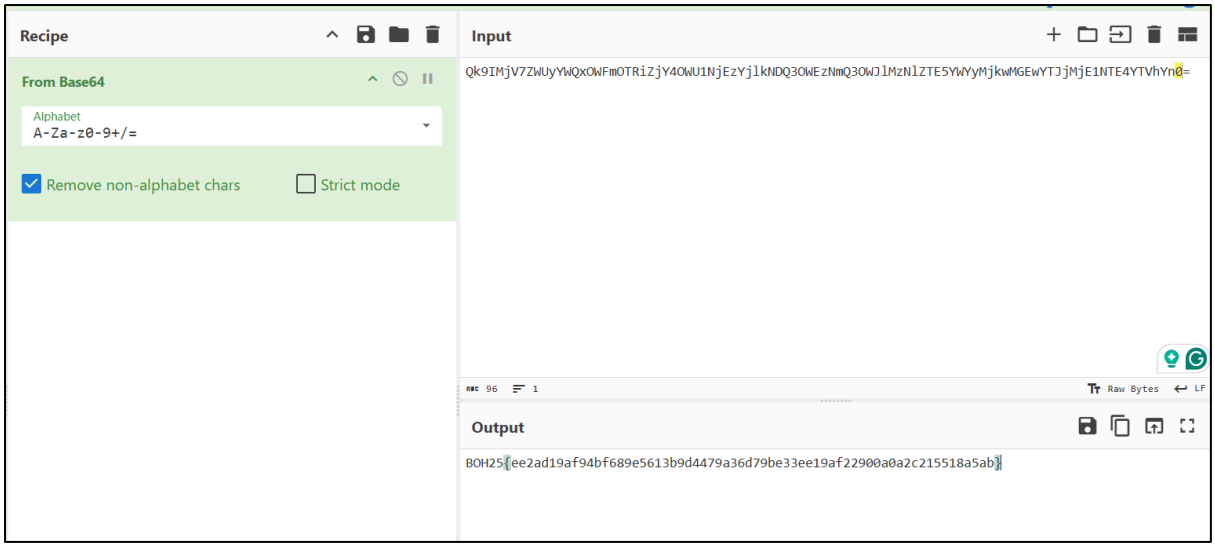


In this challenge, we were given a NTUSER.DAT file which can be simply open with Registry Explorer tool.



Based on the challenge described, we were asked to find the find the persistence done by the attackers. One of the common places to look for persistence done by the attackers in a NTUSER.DAT file is '*HKCU\Software\Microsoft\Windows\CurrentVersion\Run*', in this folder it will list down a list of programs that run at user logon. For more information can have a look here https://www.cybertriage.com/blog/ntuser-dat-forensics-analysis-2025/.

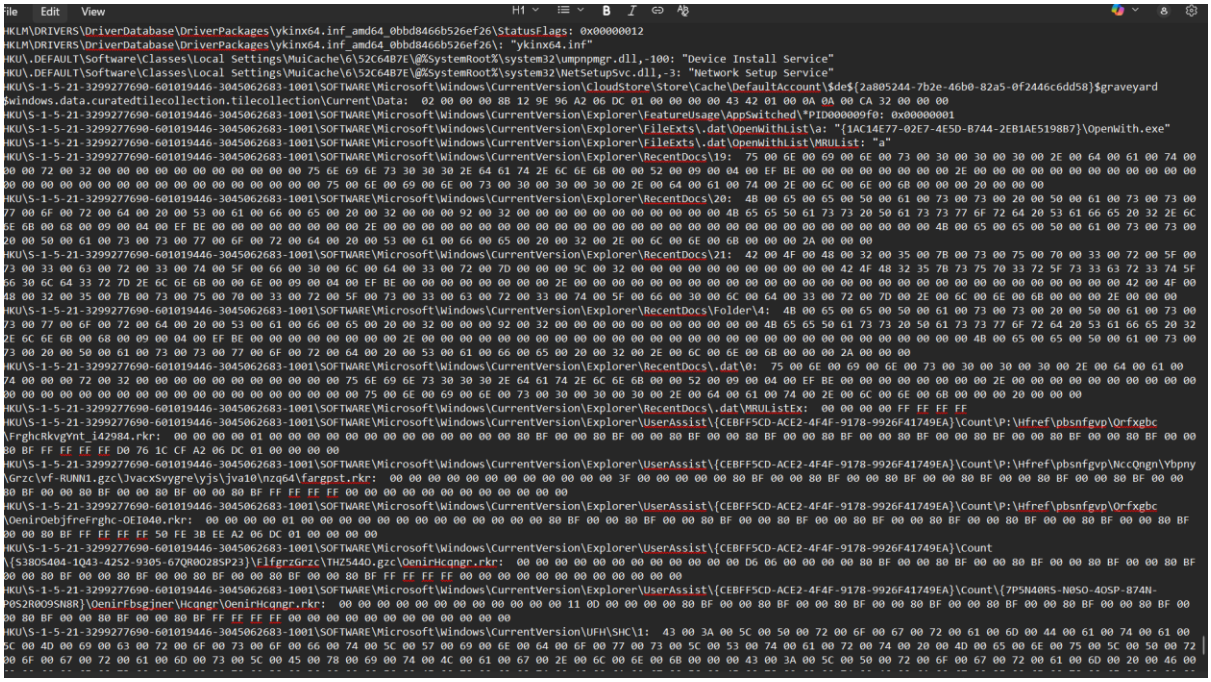
In the Run folder, there is a program named evil which the name itself is suspicious, check the program detail and we can see there is a base64 strings.



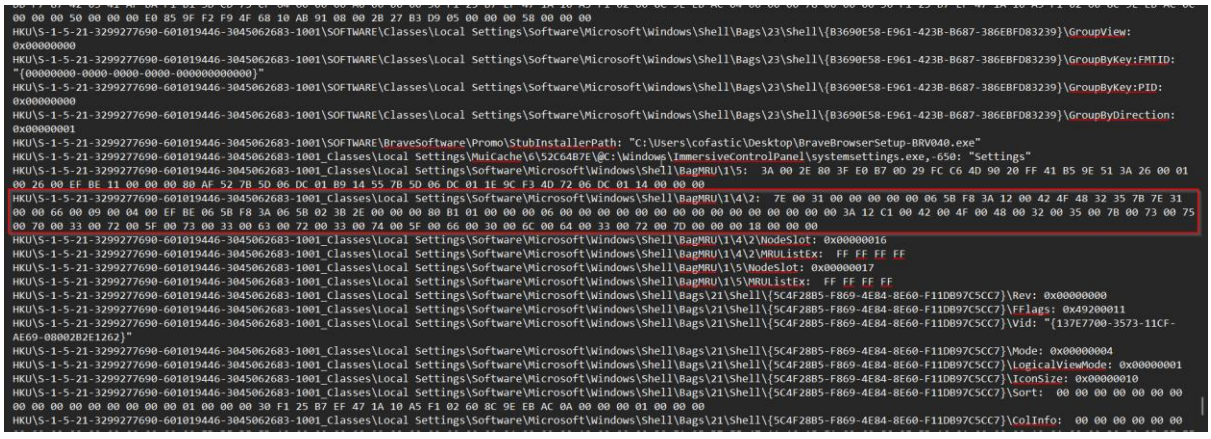
Decode the base64 strings and we got the flag!

Title: Snapshots

Description: In the year 2087, the Government monitors every digital footprint through a mandatory system surveillance. A rebellious employee workstation was seized after they were caught accessing the forbidden archives. Intelligence suggests they created a secret folder to store the critical intel that they had found, then changed its visibility to hidden to cover their tracks. The State's automated scanners missed it, but a before/after system snapshot was recovered that captured them in the act of creating and concealing the folder. Dig through the registry changes and extract the folder name. The resistance is counting on you.

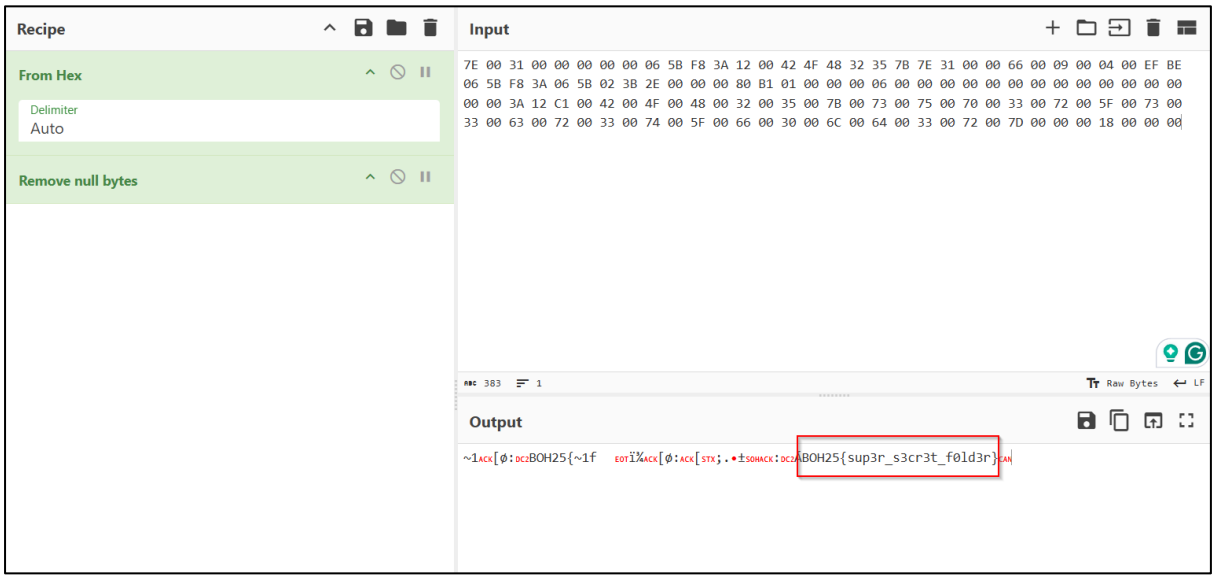


In this challenge, we were given a text file, and our mission is to find out the suspicious folder that has make changes.



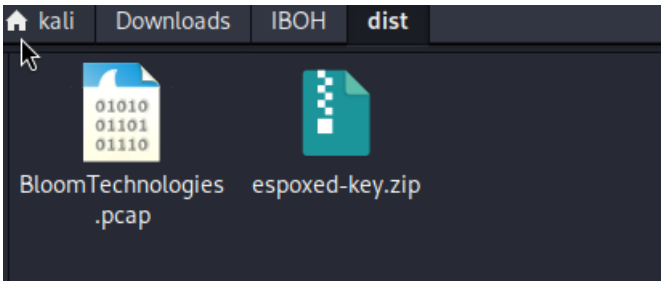
After some research, I found out that every folder that is recently browsed by the users will be recorded in the 'BagMRU' key inside the Shellbag.
https://www.cybertriage.com/blog/shellbags-forensic-analysis-2025/

Based on the given text file it appears to be extracted from USRCLASS.DAT. By searching for the keyword 'BagMRU' I noticed that there are few Hexes encoded strings. I try to decode the Hex, and I got the flag.

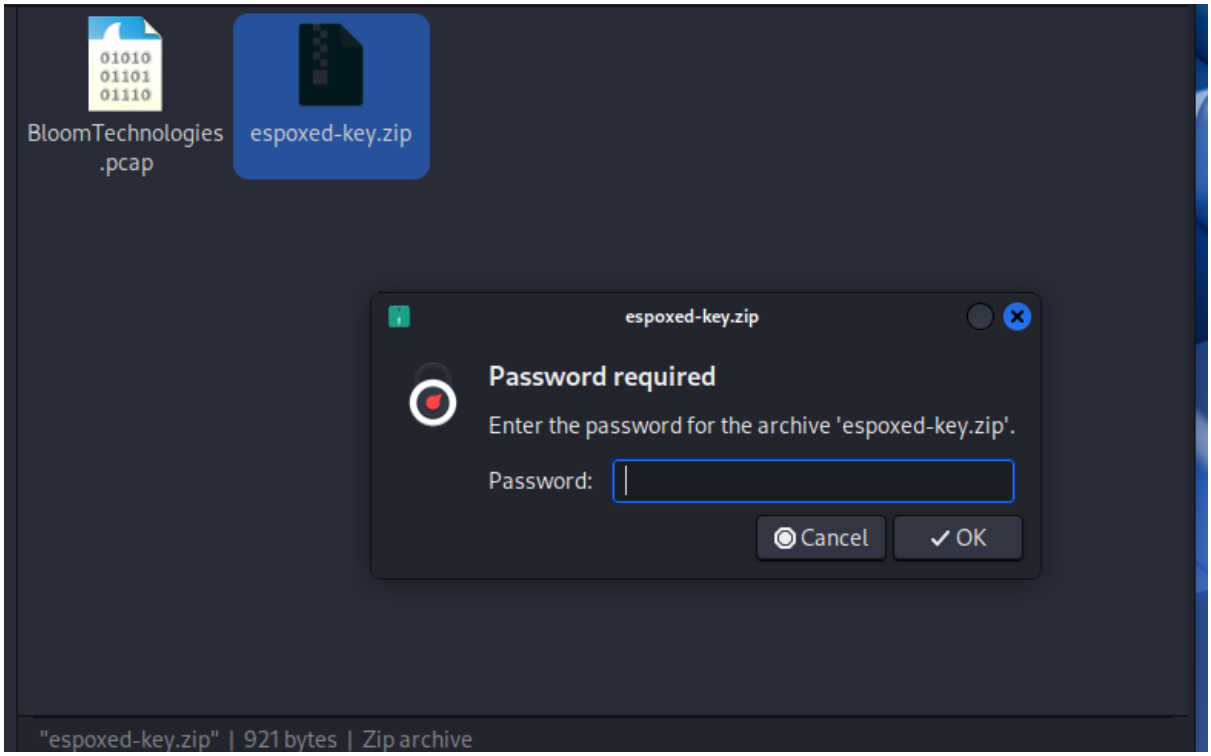


Title: Bloom (🩸 FIRST BLOOD)

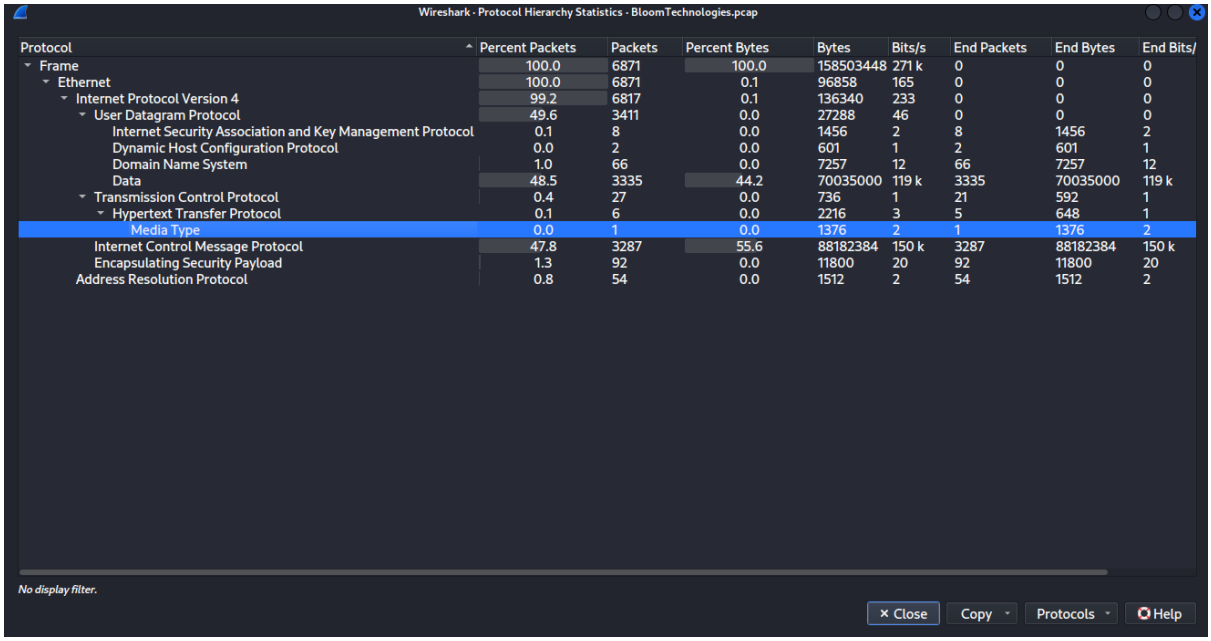
Description: Pskpskpsk, have u heard the chaos at Bloom Technologies? Their lead engineer Michael Chen didn't take rejection well and "backed up" a few things that weren't his. The SOC caught strange network chatter, especially during his final sync, and IR grabbed his laptop image before he could wipe it. Word is he even tracked his own traffic for fun. See what's hiding in there.



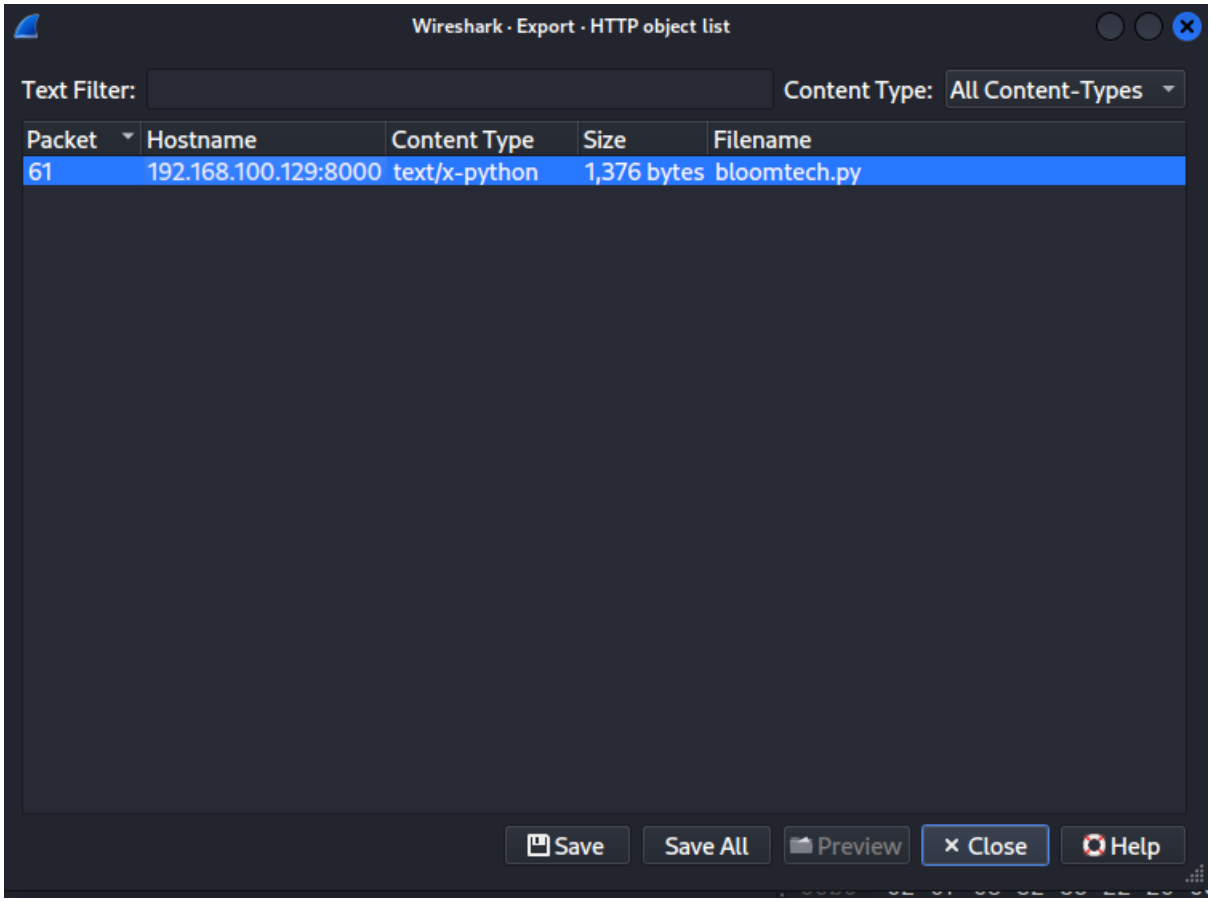
We were given a zip file and a pcap file for this challenge.



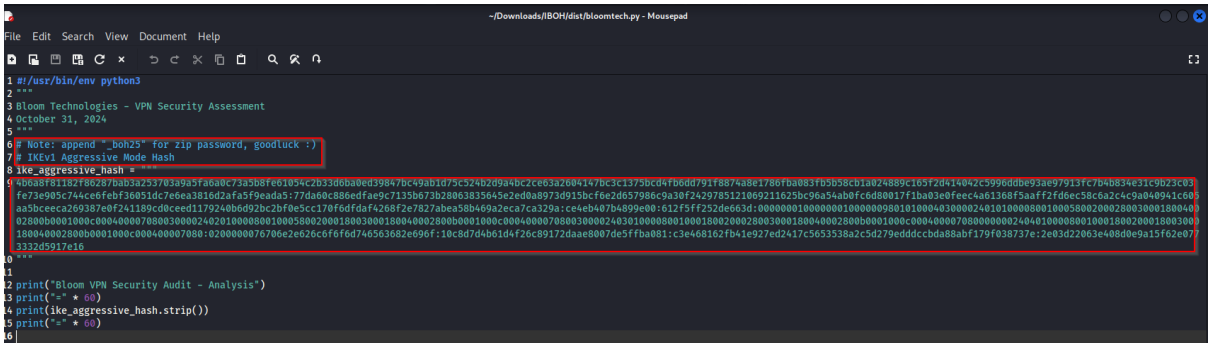
When I try to unzip the 'exposed-key.zip' it requires password to unlock the zip.



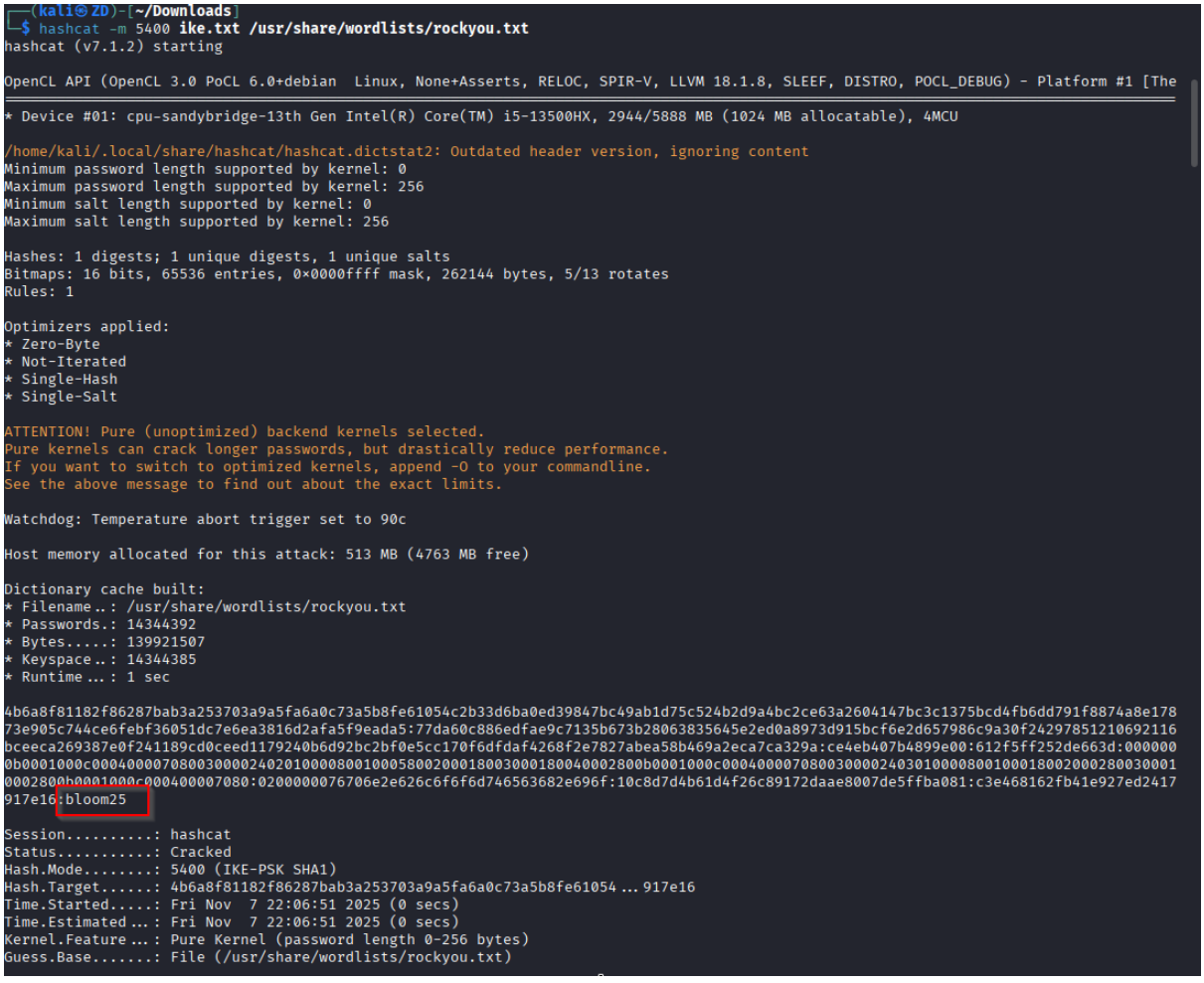
When I investigate the pcap with wireshark, I found out there are several protocols in this file and there is a media type file in the HTTP.



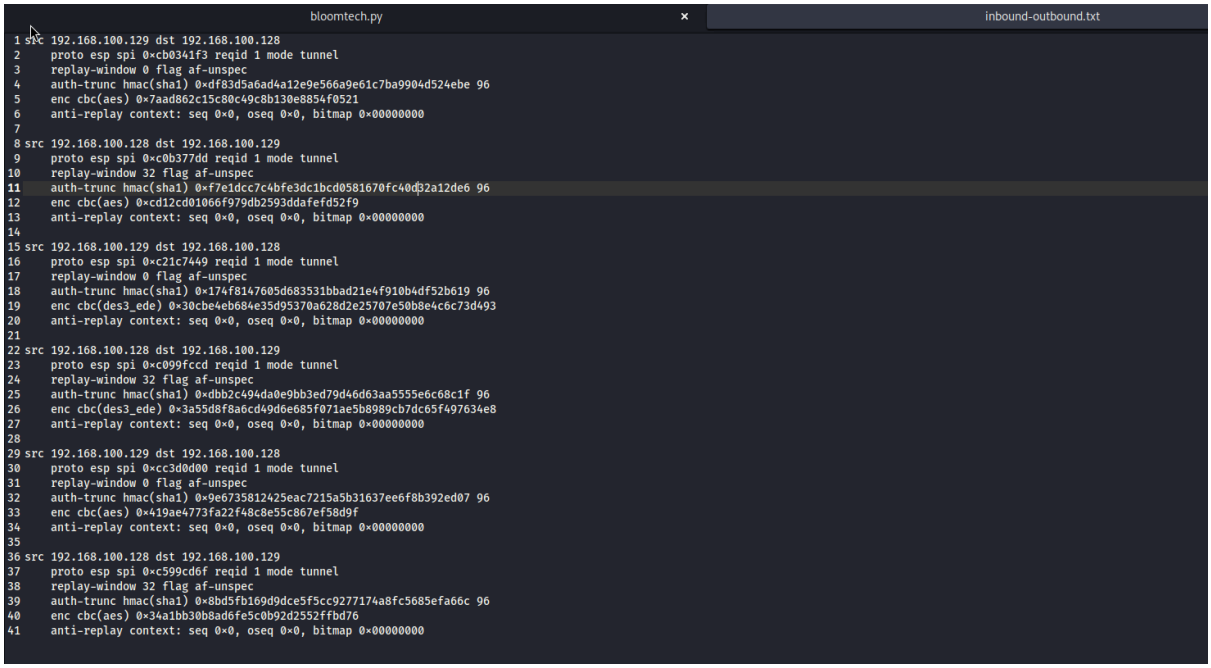
There is a file named ‘bloomtech.py’, save it locally for further investigation.



By checking the ‘bloomtech.py’ file, I saw there is an ikev1 hash and some hint regarding the zip file password. The first thing come into our mind is to try to crack this hash and append the ‘_boh25’ at the end of the password.



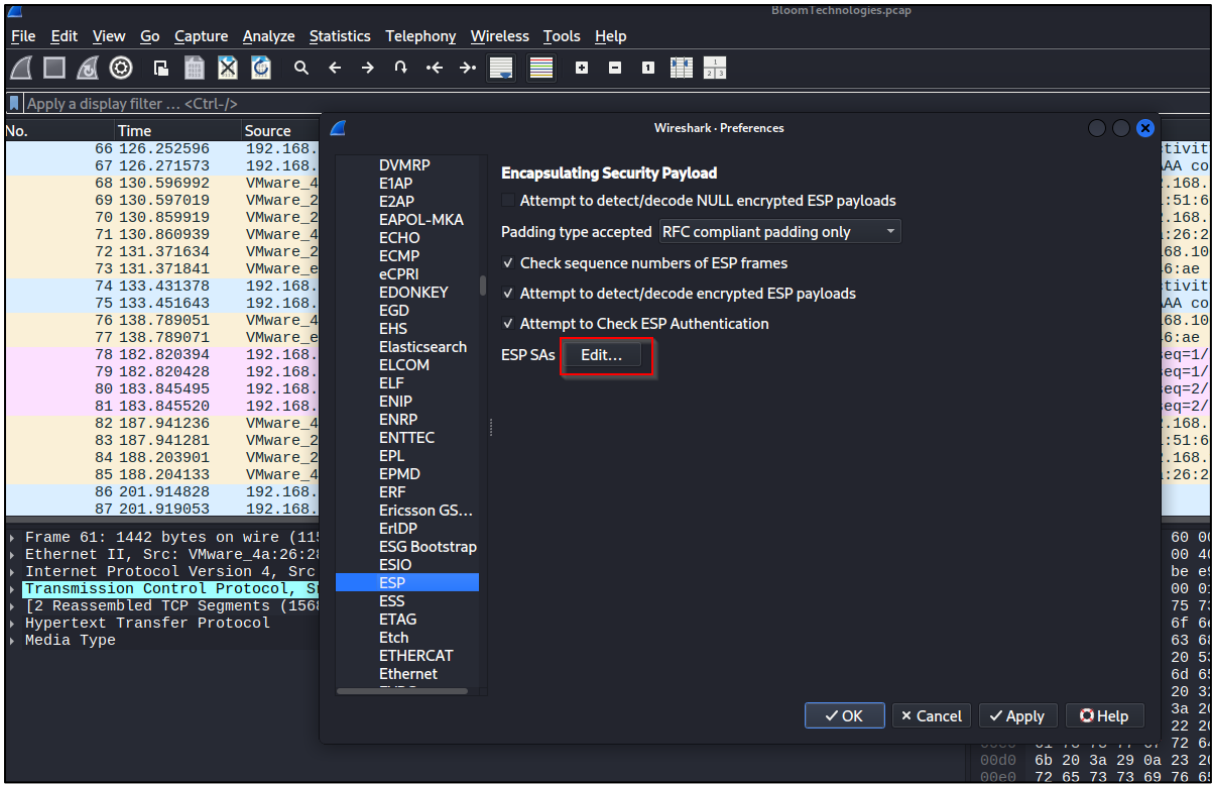
I managed to crack the ike hash and the password is ‘bloom25’. However, based on the python file just now the password the zip file should append ‘_boh25’ which mean the final password should be ‘bloom25_boh25’



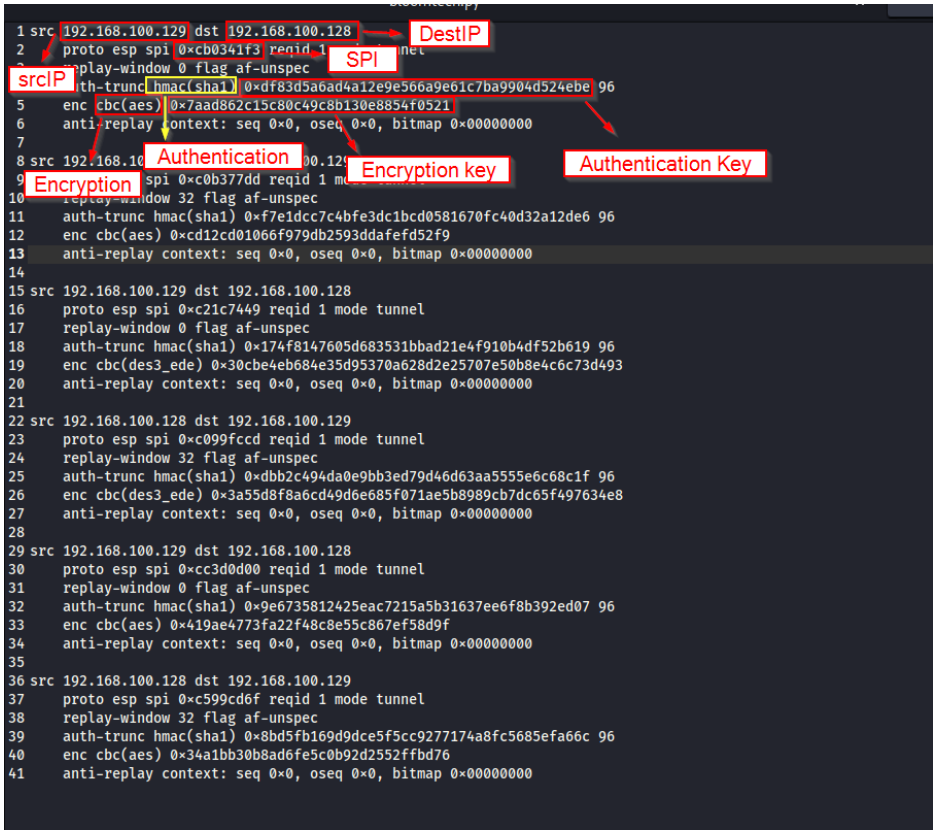
After unzipped the ‘exposed-key.zip’ with the password found above. We will see a text file name ‘inbound-outbound.txt’. From the information of the file, we can know that the information is related to the Encapsulating Security Payload (ESP) protocol.

After I did some research, I found out that we can use the information to decrypt the ESP traffics.

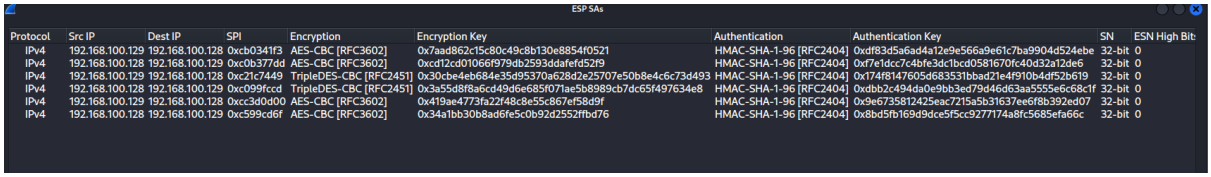
https://tech-academy.amarisoft.com/how_to_decrypt_wireshark_esp_packet_and_extract_sip_message.wiki



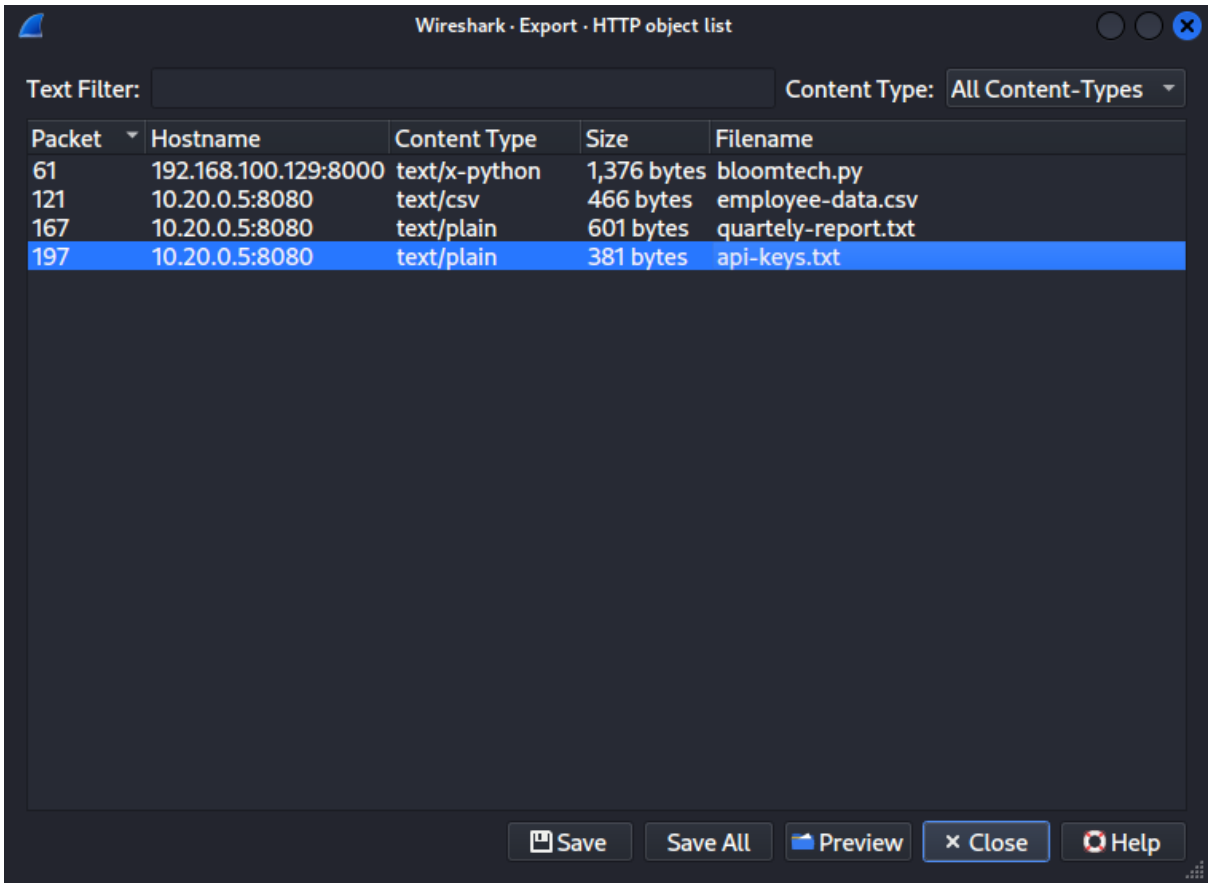
In wireshark, navigate to Edit > "preference", expand the "protocol" menu > ESP. From the you can see ESP SAs and click on edit.



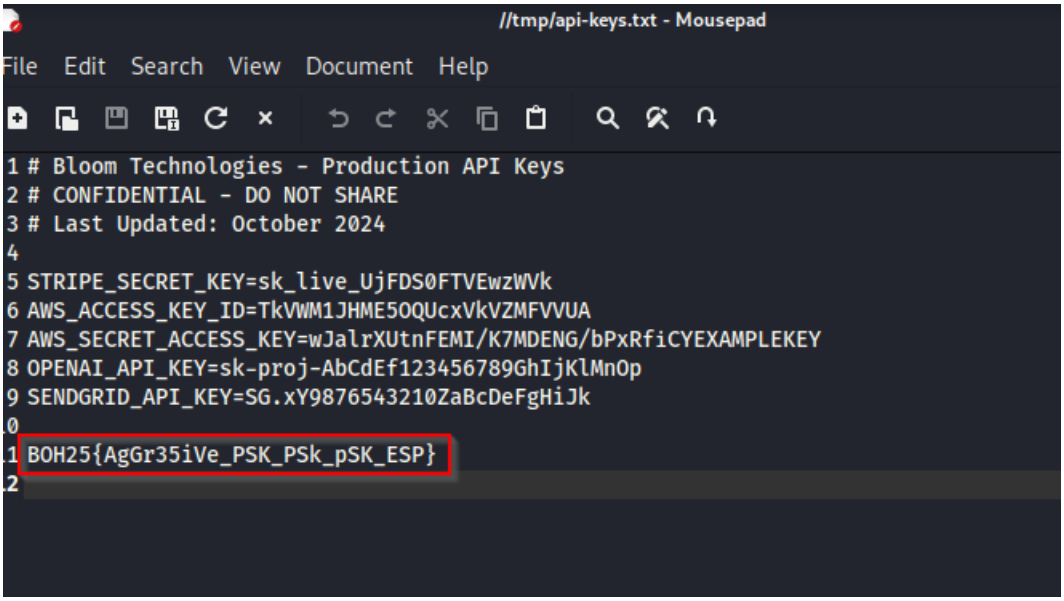
Then based on the information given from ‘inbound-outbound.txt’, insert everything to into the table as shown below.



Once completed click save and apply.



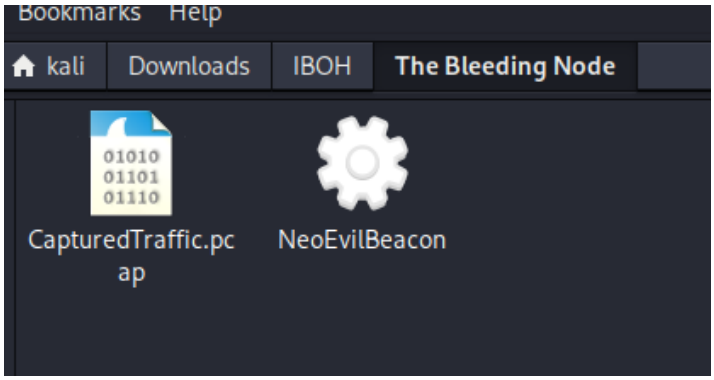
We will now notice the ESP is no longer in black colour and when I check the protocol hierarchy, I found out the HTTP have more media type files. Got the HTTP object list and check for every files.



In the ‘api-keys.txt’ I got the flag.

Title: The Bleeding Node

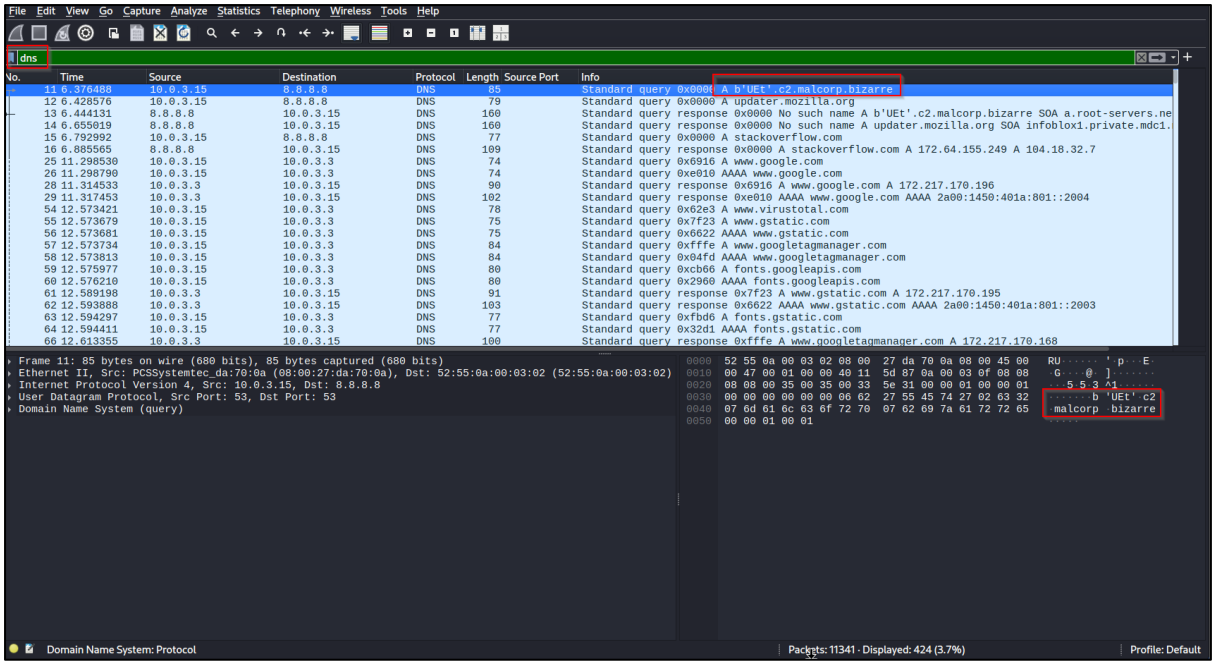
Description: In the continious war against The NeoCyberian Regime, FUN (Force Underground Network) was breached. Our Net Watchers found unusual outbound traffics from a crucial node. They managed to get an artifact from a memory dump believed to be a malware. They also captured the full traffic on the compromised host moments before it went dark. It is believed that the malware contains crucial intel on which data was being stolen.



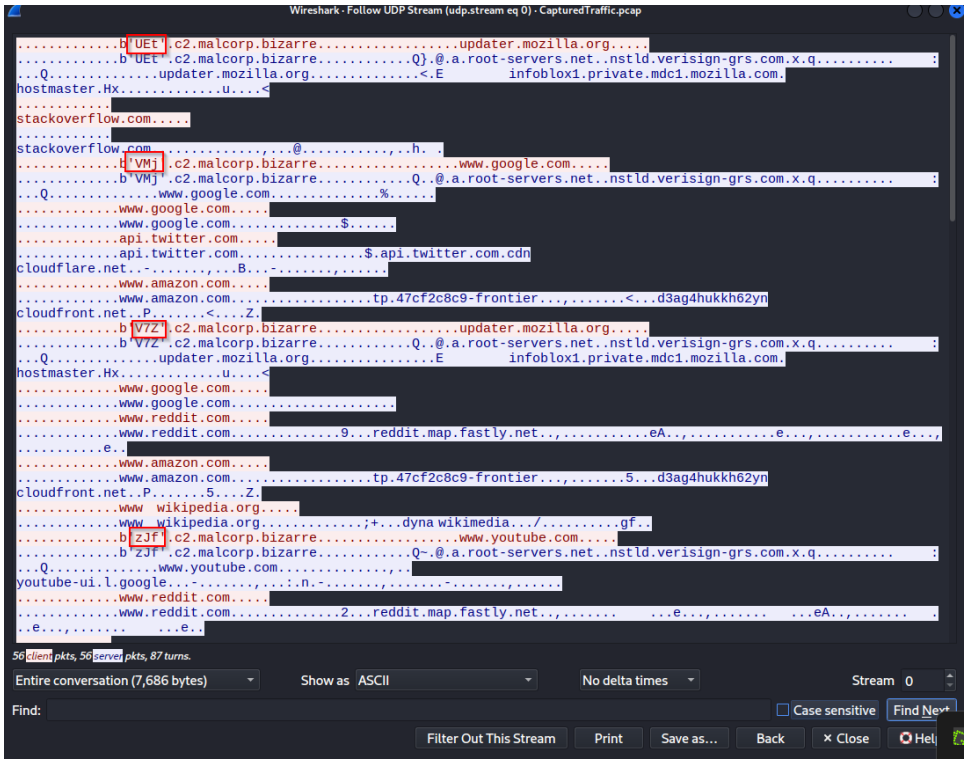
For this challenge, we were given 2 files, a pcap file and an executable file.

```
┌──(kali㉿ZD)-[~/Downloads/IBOH/The Bleeding Node]
└─$ ./NeoEvilBeacon
Initializing C2 connection...
Target Domain: c2.malcorp.bizarre
Exfil Protocol: DNS-QUERY
Data Encoding Algorithm: Vigenère Encode → Base64 Encode
Key: owned
Beaconing complete.
```

When I execute the 'NeoEvilBeacon', I got the information as shown in above figure. From the figure we can obtain some critical information. The program seems to connect to a domain named 'c2.malcorp.bizarre' and some data were exfiltrated via the DNS. The data undergoes the encoding process with Vigenère with the key 'owned' then Base64.

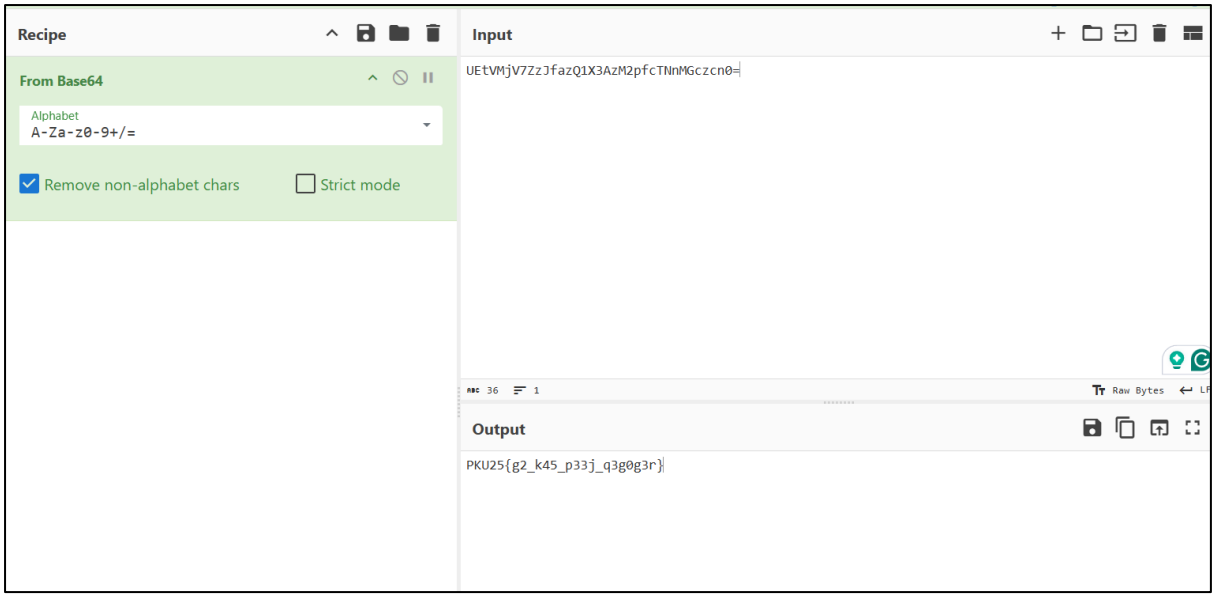


By looking at the DNS traffic from the pcap file, we can see that there is communication with the ‘c2.malcorp.bizarre’ domain.

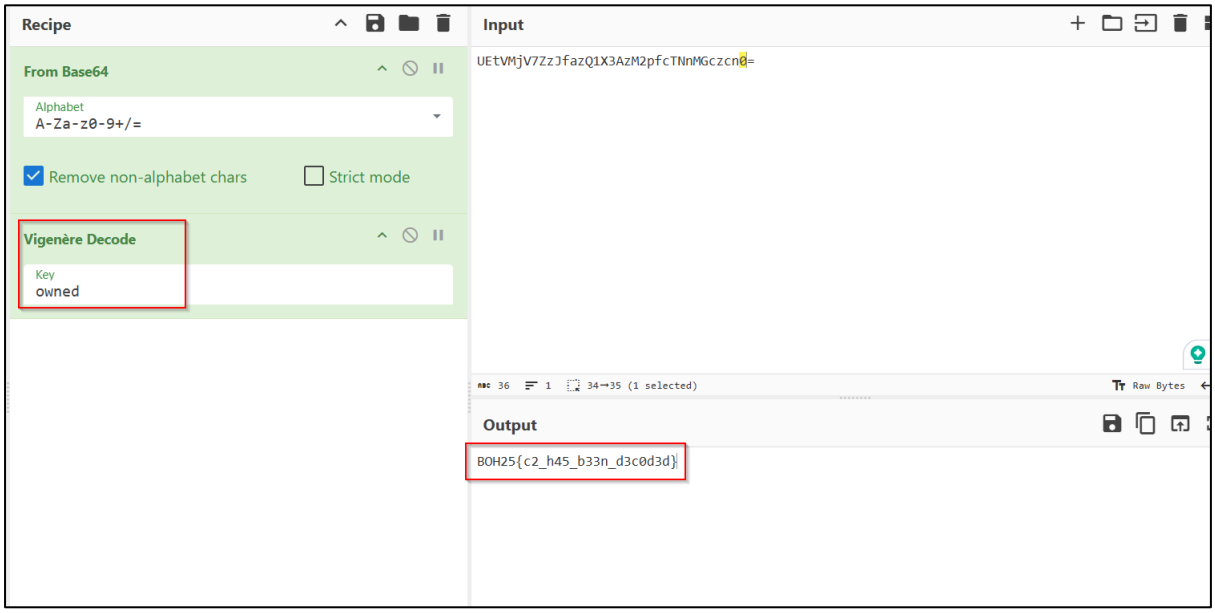


I tried to follow the UDP stream, and I found out that every communication with the DNS have some suspicious strings in the front. Try to reassemble them from the first packet to the last. We will get a complete base64 strings.

‘UEtVMjV7ZzJfazQ1X3AzM2pfcTNnMGczcn0=’



Now we got the based64 and it is time to decode the strings backward from base64 to Vigenère. After decoded the base64 I got 'PKU25{g2_k45_p33j_q3g0g3r}'

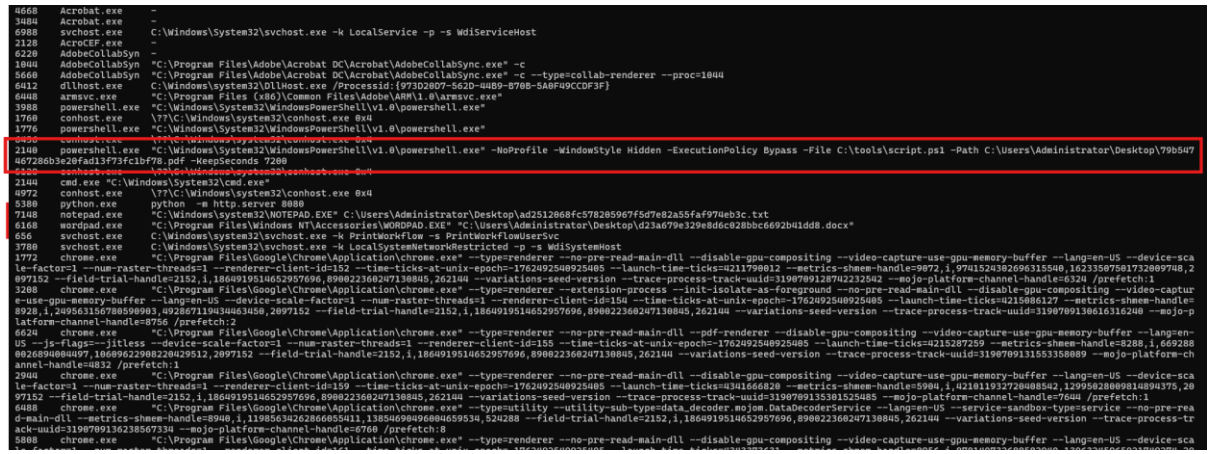


Then add the Vigenère decoder with the key 'owned'. We got the flag.

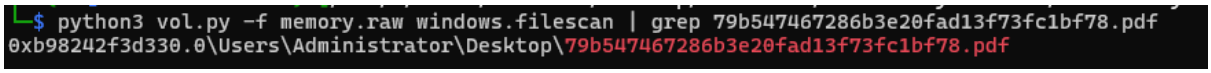
Title: Now I'm here, now I'm not

Description: I was viewing a document, then something happened, and my document closed! I can't find it anymore, can you?

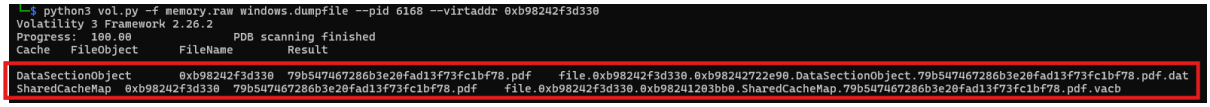
https://drive.google.com/file/d/12J9wqM2fOix1oe8CaG21COWIGVZXxkm6/view?usp=sharing



The file we downloaded was a memory raw file, which can be analyse with Volatility 3. By using the '*python3 vol.py -f memory.raw windows.cmdline*' command, it shows the list of executed command. From the output I found a suspicious command with the PID of 2140 and there is a path led to '467286b3e20fad13f73fc1bf78.pdf'.



I try to run the file scan to look for the suspicious pdf file, and located it offset - 0xb98242f3d330.



I was then use the offset of the pdf file to extract it, and it output 2 files with different extension for.

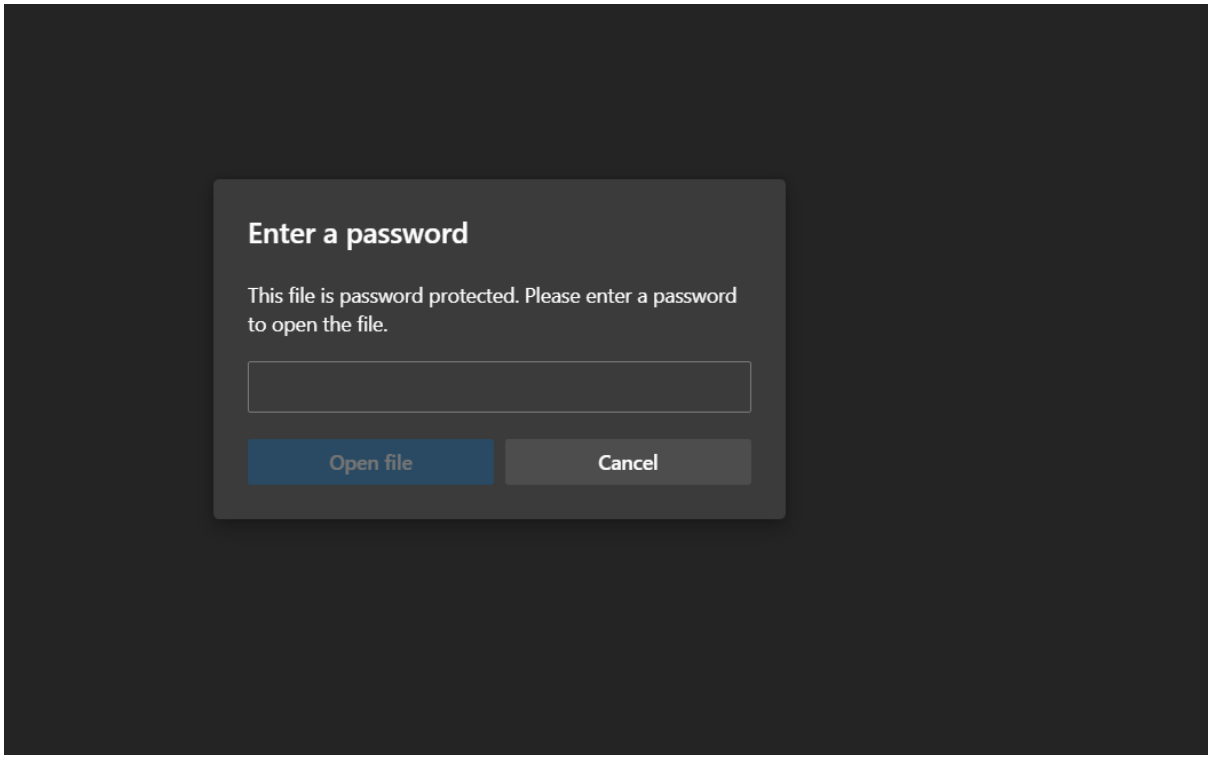
```
┌──(ZD㉿DESKTOP-3RKP009)-[/mnt/c/Users/ILLEGEAR/Desktop/CTF tools/volatility3-2.26.2/volatility3-2.26.2]
└─$ head file.0xb98242f3d330.0xb98242722e90.DataSectionObject.79b547467286b3e20fad13f73fc1bf78.pdf.dat
%PDF-1.4
%◆◆◆◆
1 0 obj
<<
/Filter /Standard
/V 2
/Length 128
/R 3
/O <8A28A76D2036237642B22159285A04E781B86D1C974F7A0D1FC9ED263089303A>
/U <D21B6122B62A9C56E21F212F0C49209928BF4E5E4E758A4164004E56FFFA0108>

┌──(ZD㉿DESKTOP-3RKP009)-[/mnt/c/Users/ILLEGEAR/Desktop/CTF tools/volatility3-2.26.2/volatility3-2.26.2]
└─$ head  file.0xb98242f3d330.0xb98241203bb0.SharedCacheMap.79b547467286b3e20fad13f73fc1bf78.pdf.vacb
%PDF-1.4
%◆◆◆◆
1 0 obj
<<
/Filter /Standard
/V 2
/Length 128
/R 3
/O <8A28A76D2036237642B22159285A04E781B86D1C974F7A0D1FC9ED263089303A>
/U <D21B6122B62A9C56E21F212F0C49209928BF4E5E4E758A4164004E56FFFA0108>
```

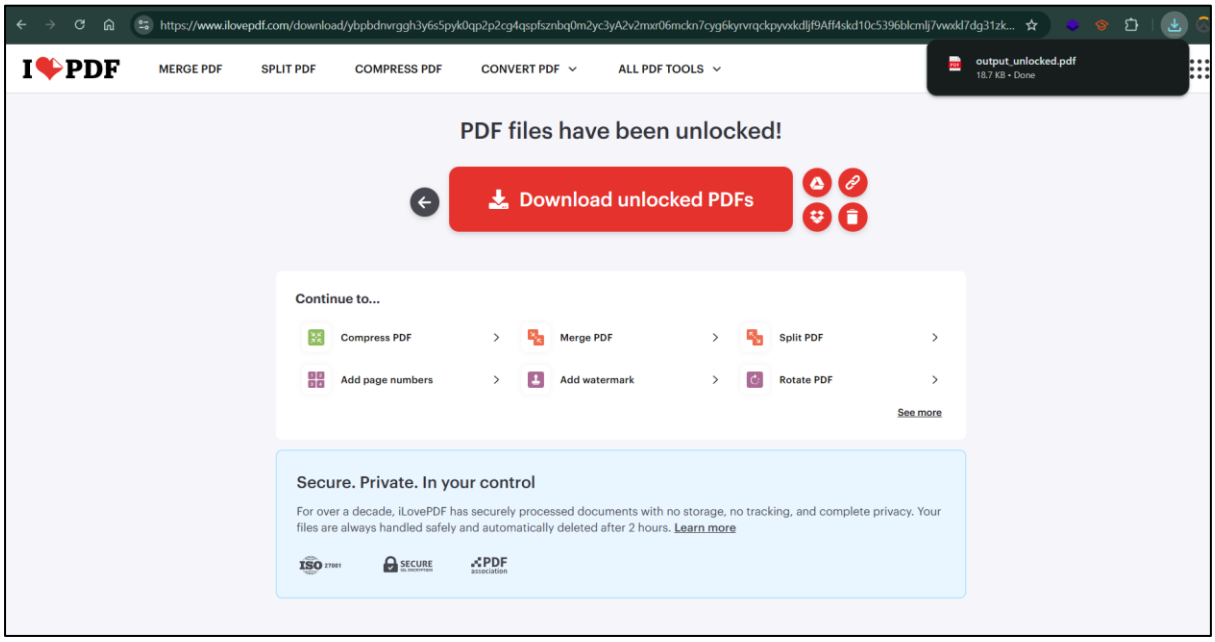
When I check for the file detail, I noticed that they are the same file just output into different type.

```
└─$ file output.pdf
output.pdf: PDF document, version 1.4, 1 page(s)
```

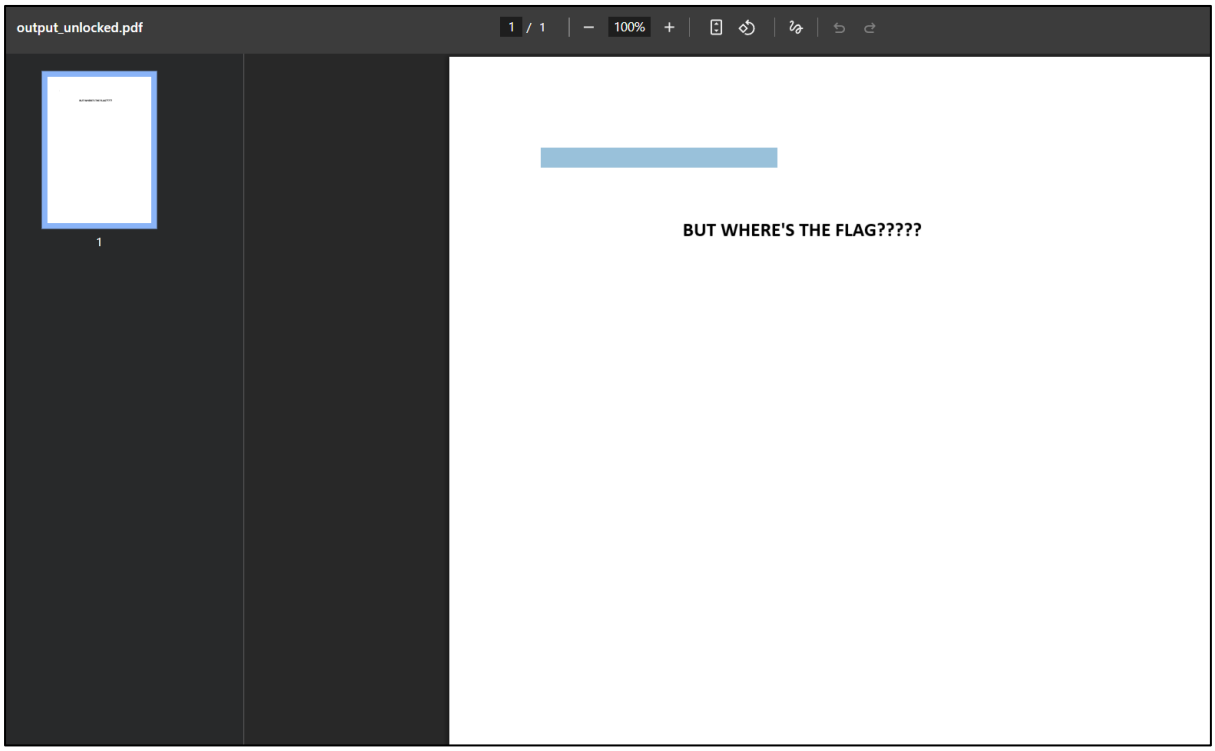
Since I know it is PDF file, I change the file and extension.



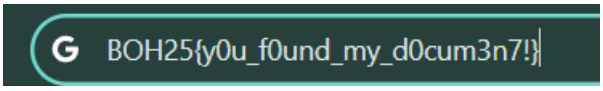
The PDF file require password to unlock.



There is an online tool that can help to remove the pdf password.
https://www.ilovepdf.com/



When I open the pdf file, the flag is in white font. Copy the highlighted part and paste it somewhere else to reveal the flag.

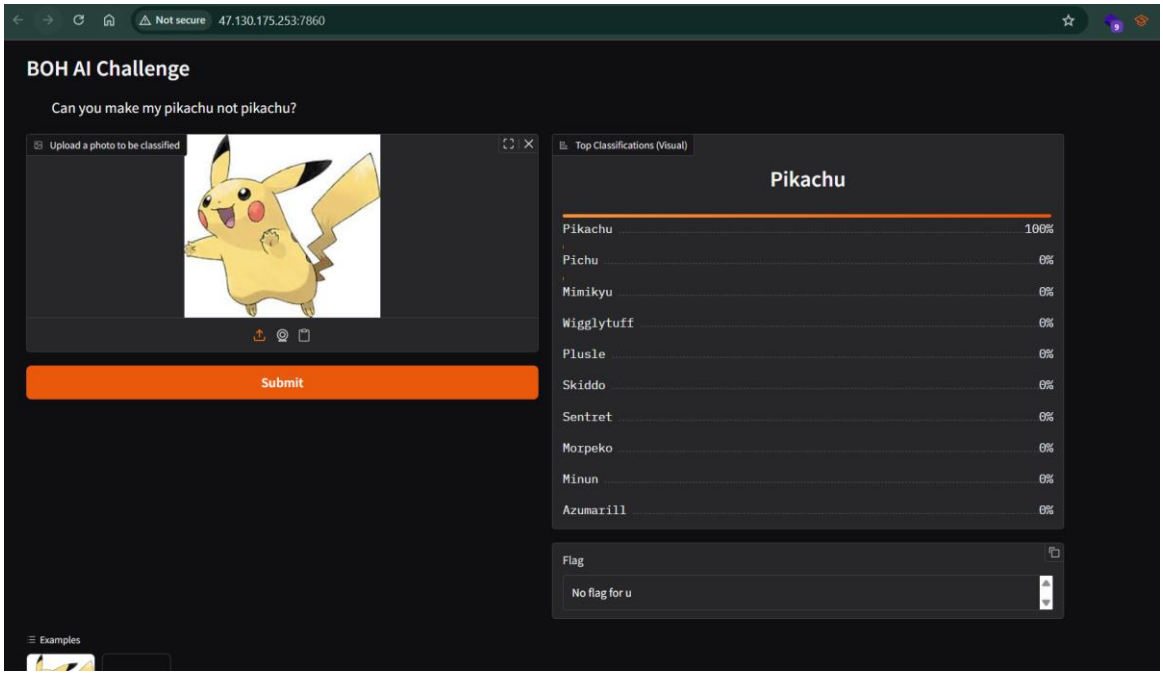


Got the flag.

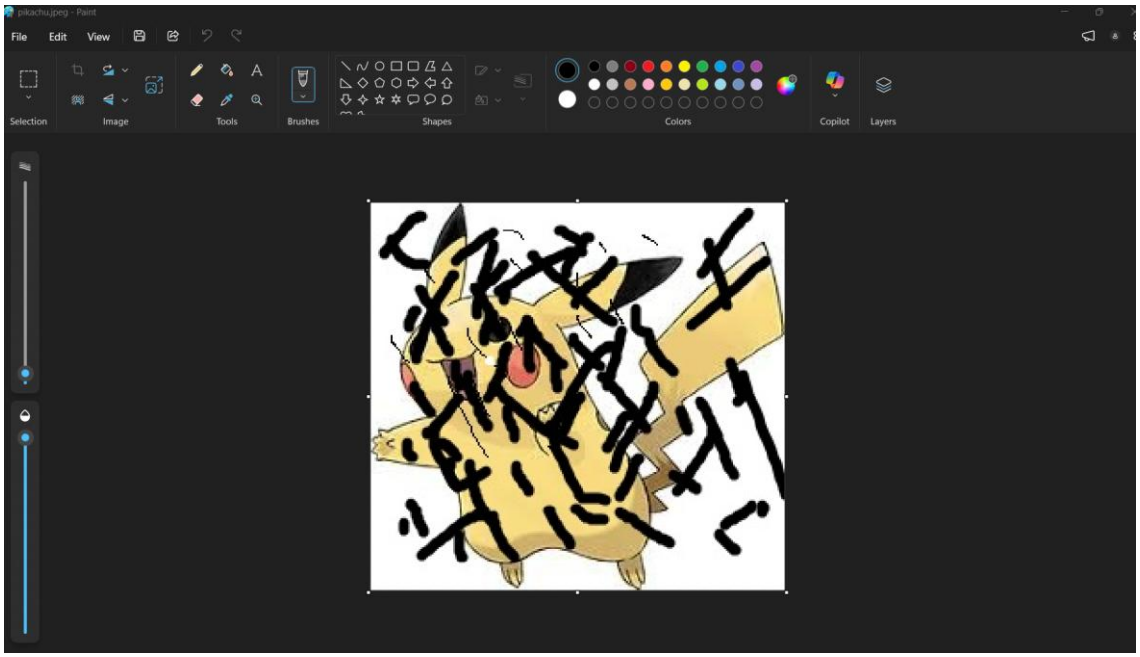
AI

Title: pika pika

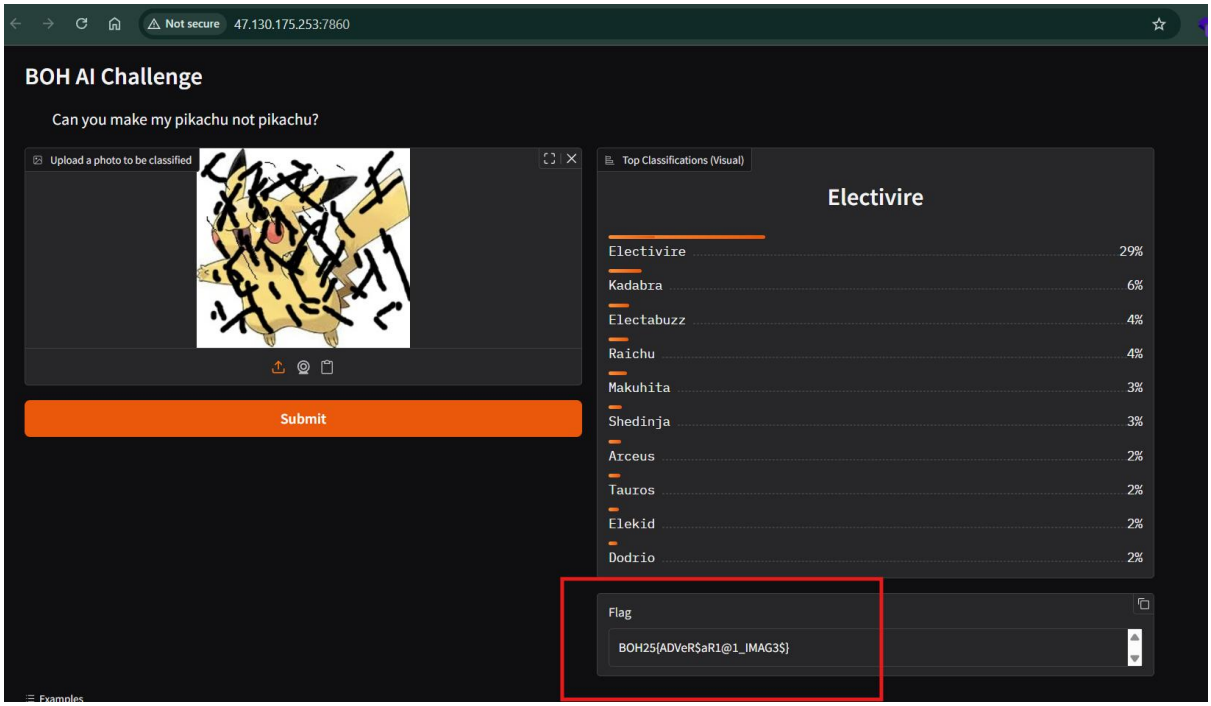
Description: Can you make my pikachu not pikachu?



This is a webpage that use AI to identify Pikachu. What we need to do here is to confuse the AI to make it cannot recognise Pikachu to get flag.



One of the simple way is to use Paint to draw random lines too the picture to confuse the AI.

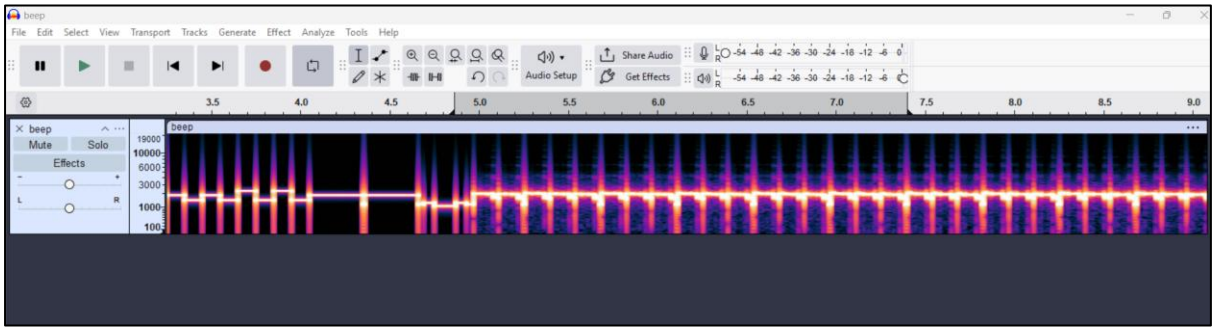


Upload the picture and we got the flag.

Misc

Title: Bleep Bloop

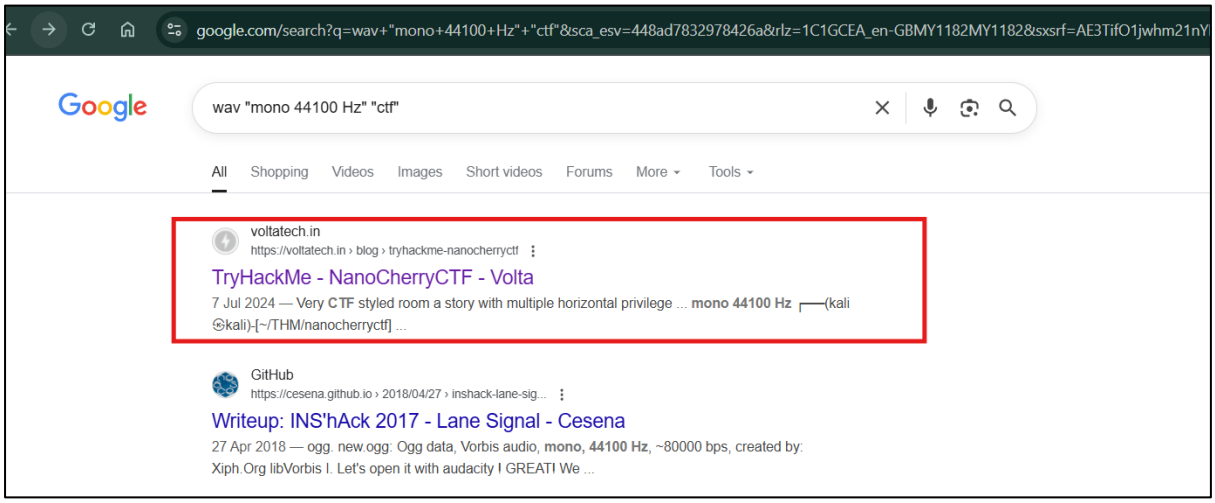
Description: An audio file transmitting weird bleeping sound is found.



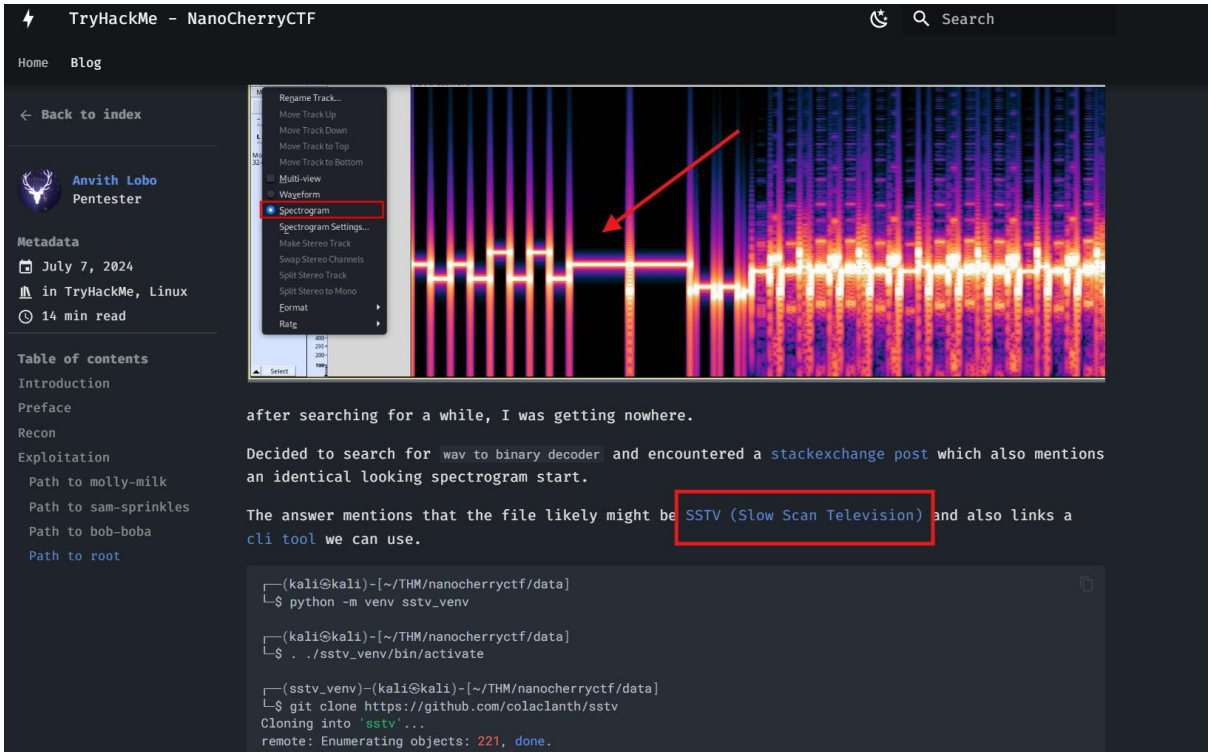
Use Audacity to check the audio pattern in spectrogram mode. Found some unique pattern at the start of the audio.

```
└─$ file beep.wav
beep.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, mono 44100 Hz
```

Get the detail of the wav file.



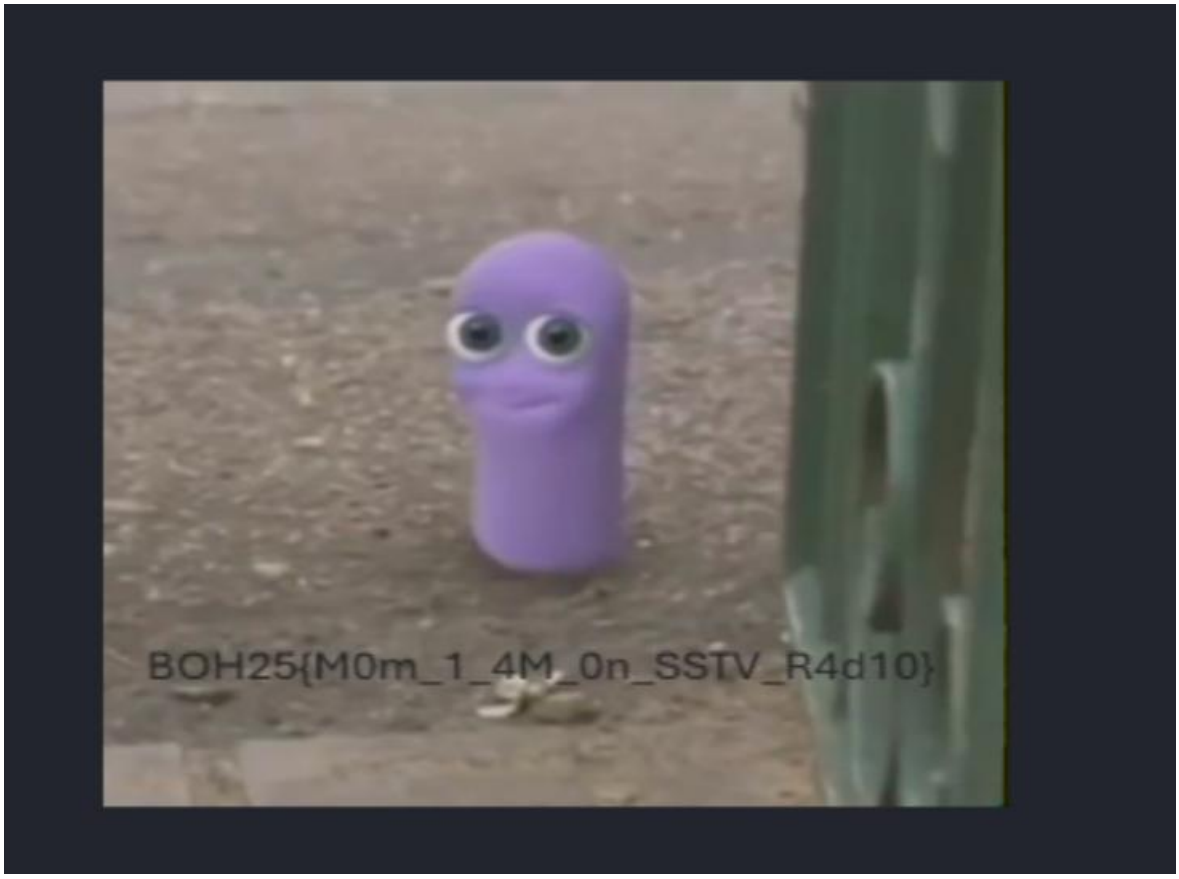
Did some 'Googling' and found an interesting writeup page.
https://voltatech.in/blog/2024/tryhackme-nanocherryctf/



By reading through the writeup, I found out that the audio pattern is similar to the challenge audio file and I get to know that the audio could be Slow Scan Television (SSTV). From the writeup I also got the tools to decode the audio.

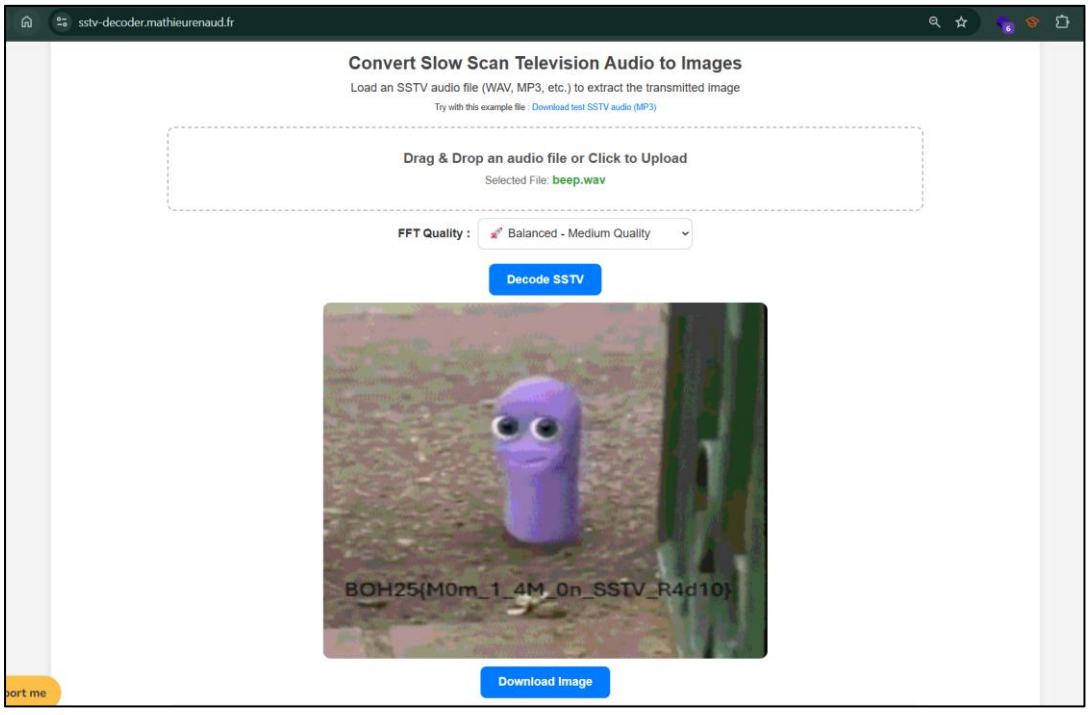
```
┌──(sstv_venv)-(kali㉿ZD)-[~/Downloads/sstv]
└─$ sstv -d ../beep.wav -o result.png
[sstv] Searching for calibration header... Found!
[sstv] Detected SSTV mode Scottie 1
[sstv] Decoding image...    [##############################################################################]  99%
[sstv] Reached end of audio whilst decoding.
[sstv] Drawing image data...
[sstv] ...Done!
```

Use the tool (https://github.com/colaclanth/sstv) to decode the audio wav file.



Got the flag.

There is an easier alternative way. Once I had identified it is SSTV I can use online decoder to get the flag. Website link (https://sstv-decoder.mathieurenaud.fr/).

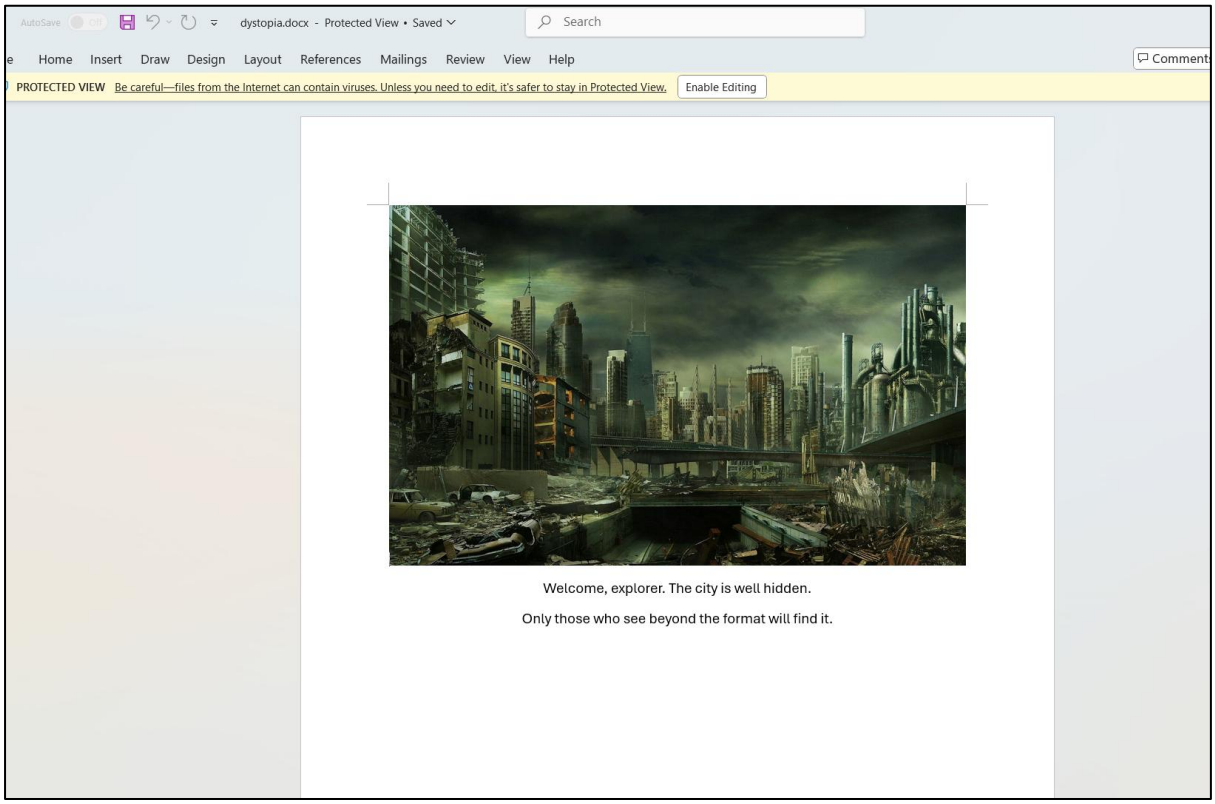


Title: City of Dystopia

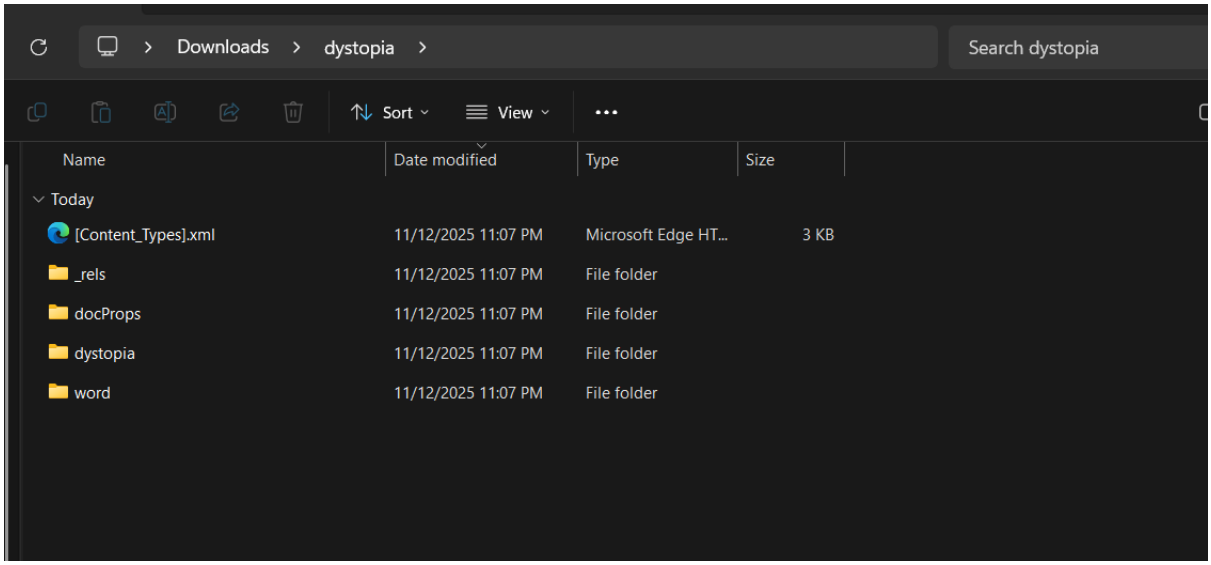
Description: Year 2085. The megacities are under total control. Memories are erased. History is rewritten. Hope is extinct.

```
└─$ file dystopia.docx
dystopia.docx: Zip archive data, at least v1.0 to extract, compression method=store
```

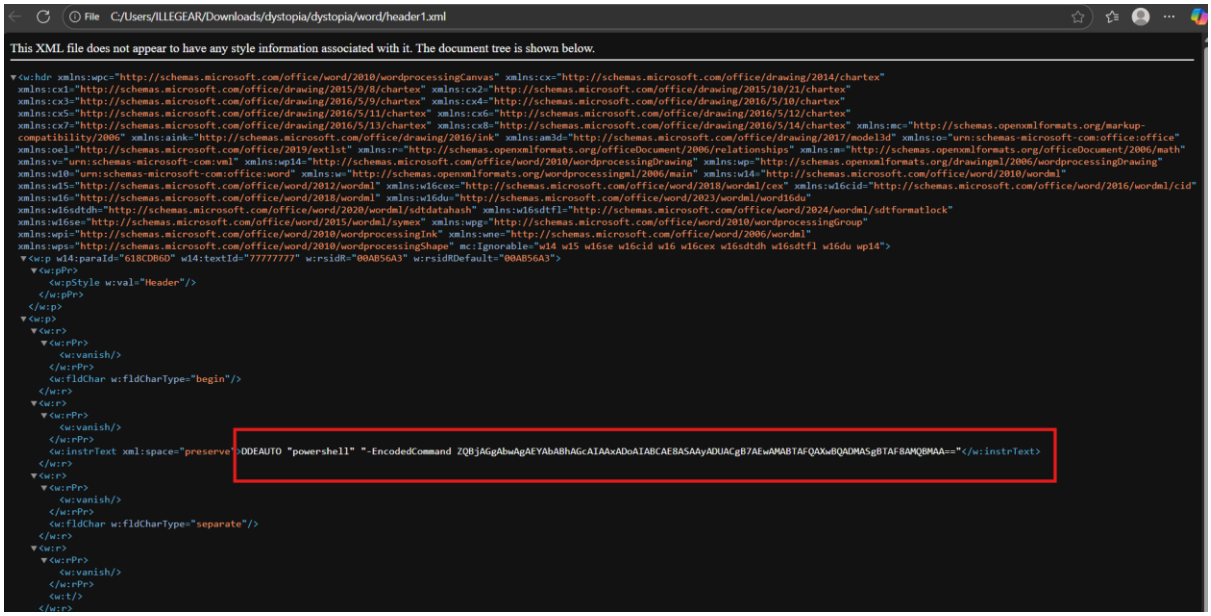
In this challenge we got a docx file.



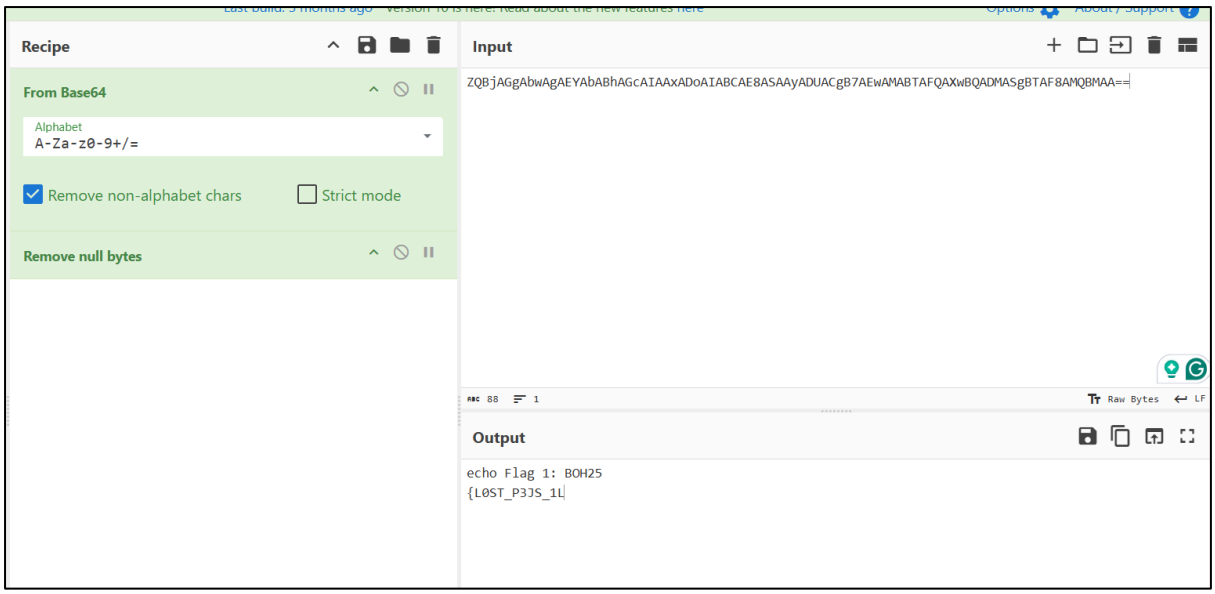
When open it up, there is nothing much to see. Since we know all docx file is Zip archive file. We can try to unzip it and look for the metadata.



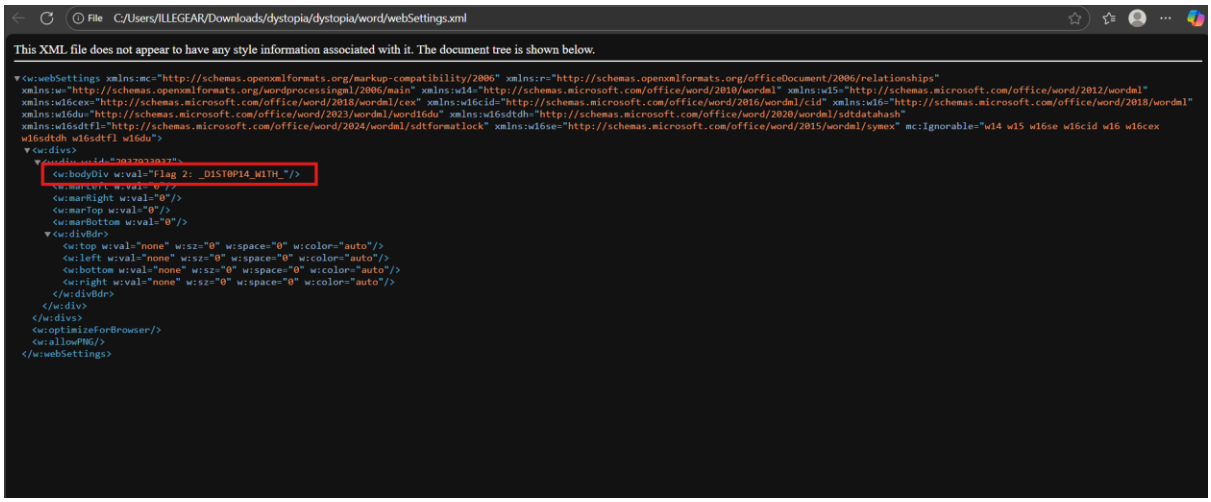
After unzipped the file, we get these folders. Then try go check every file to get the flag.



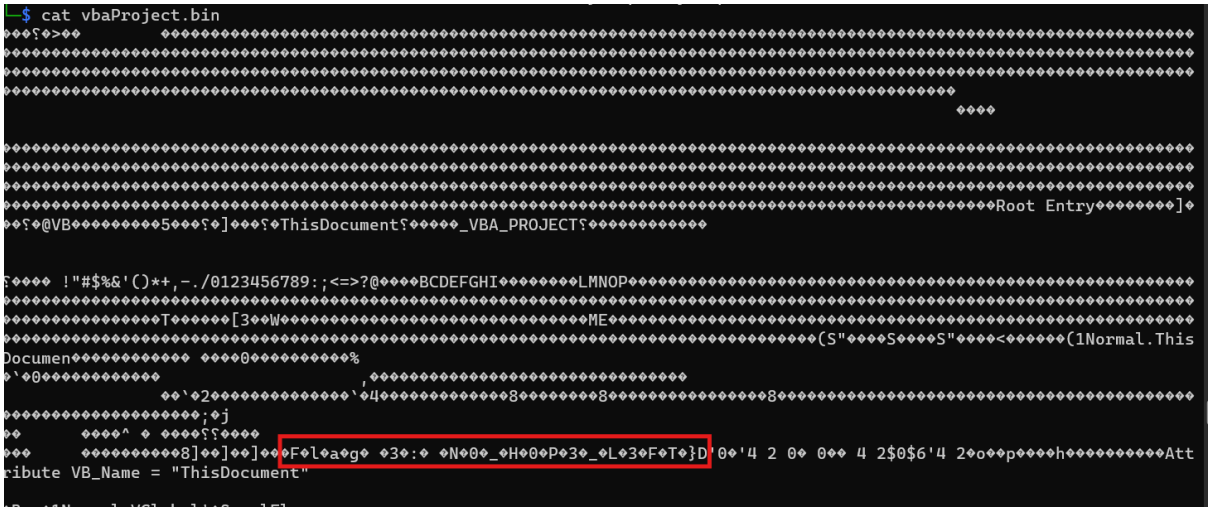
The is a base64 strings in the ‘\dystopia\word\header1.xml’.



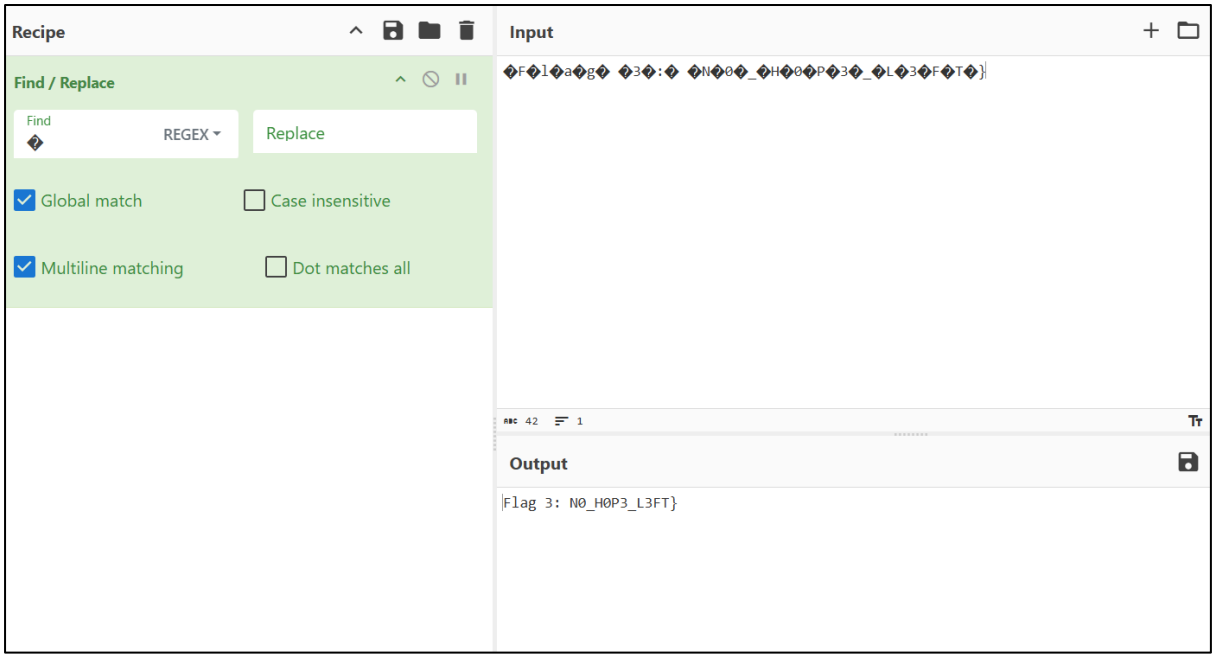
Decode the base64 and we for the first part of the flag.



Second part of the flag located in "\dystopia\word\webSettings.xml"



Third part of the flag can be found by checking the ‘vbaProject.bin’.



Third part of the flag obtained.

OSINT

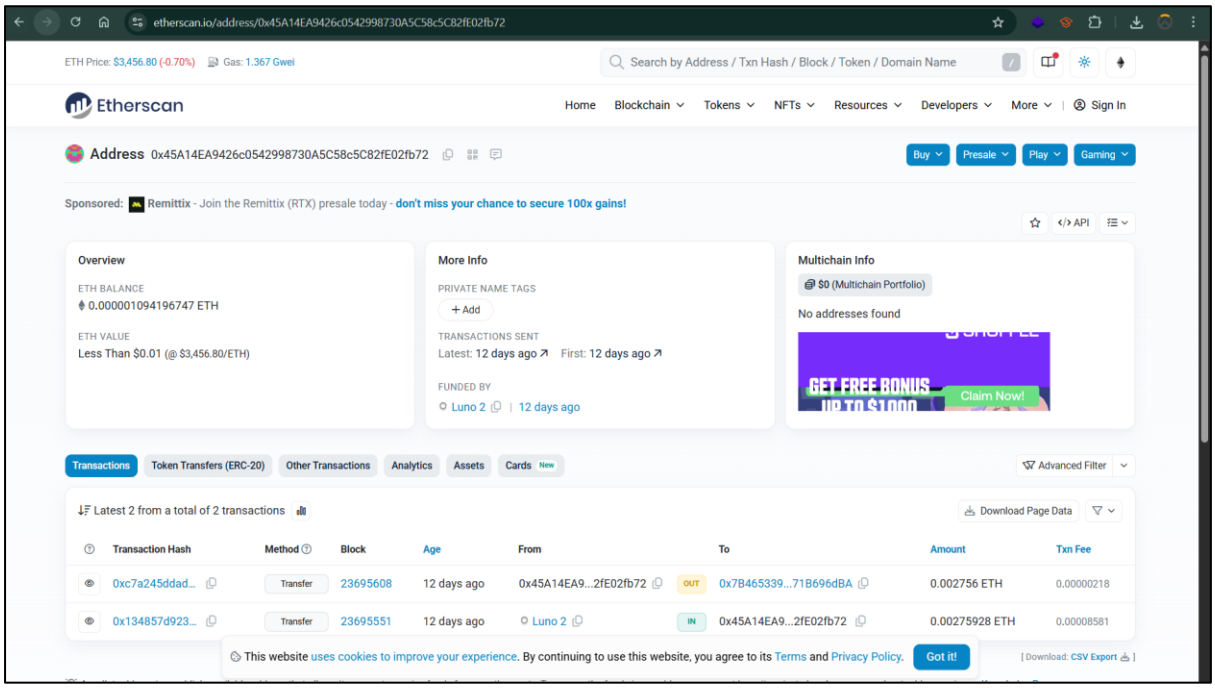
Title: The Laundromat

Description: CryptoEx, a cryptocurrency exchange under investigation for fraudulent activity, scammed a victim, StarDreamer out of their cryptocurrency holdings. The stolen funds were moved through multiple wallet addresses by CryptoEx to obscure the money trail.

Your task: Track the complete money flow from the starting address to the final destination. Count the number of addresses that CryptoEx used (excluding the starting address given below), then identify the final destination wallet.

Starting wallet address: 0x45A14EA9426c0542998730A5C58c5C82fE02fb72

Flag Format: BOH25{numberOfCryptoExAddresses_finalWalletAddress} (case sensitive)



One of the well-known webpage to keep track of every block chain transection is Etherscan.
https://etherscan.io/address/0x45A14EA9426c0542998730A5C58c5C82fE02fb72

For this challenge we just need to keep track by click the OUT address till the end then we will be able to get the final wallet address and the total time of transactions happens.

No.	Wallet Address	Link
1	0x45A14EA9426c0542998730A5C58c5C82fE02fb72	https://etherscan.io/address/0x45A14EA9426c0542998730A5C58c5C82fE02fb72
2	0x7B46533930119A1D824ac2FB836267F71B696dBA	https://etherscan.io/address/0x7b46533930119a1d824ac2fb836267f71b696dba
3	0x1c022C2C168dF73705823257702406530bD70E32	https://etherscan.io/address/0x1c022c2c168df737058232577024065 30bd70e32
4	0xA3c366a65cf4B4ae8F53EbDF27F25732682082e9	https://etherscan.io/address/0xa3c366a65cf4b4ae8f53ebdf27f25732682082e9
5	0x2221754606f6743537a805b097B40F8eBBDC9Db0	https://etherscan.io/address/0x2221754606f6743537a805b097b40f8ebbdc9db0

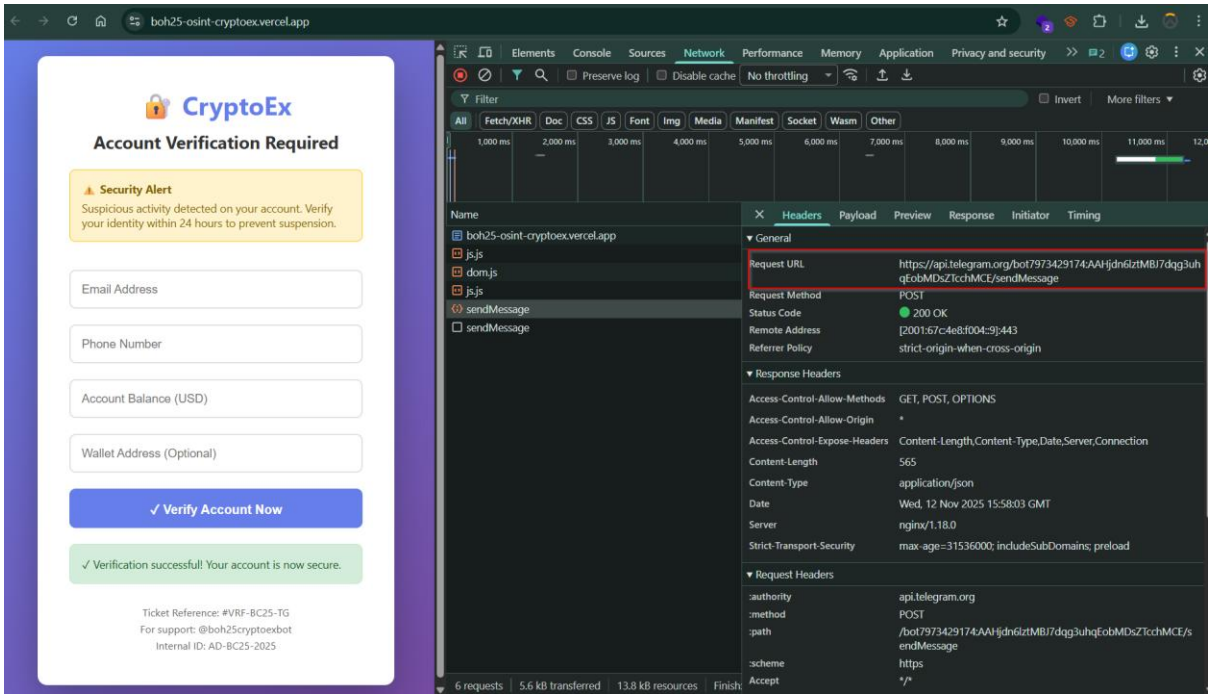
Flag: BOH25{5_0x2221754606f6743537a805b097B40F8eBBDC9Db0}

Title: Operation CryptoEx

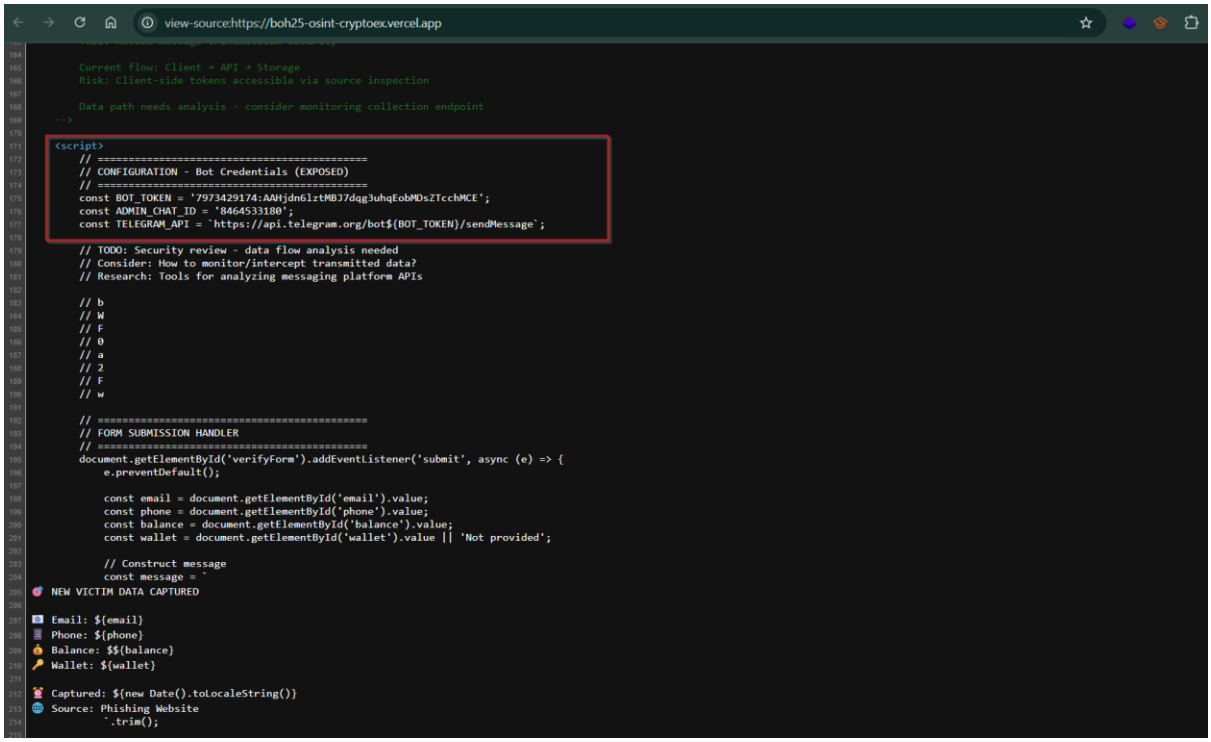
Description: A phishing operation is targeting cryptocurrency users. Investigate the infrastructure and identify the threat actor.

Objective: Find the flag hidden in their infrastructure and see what they're up to.

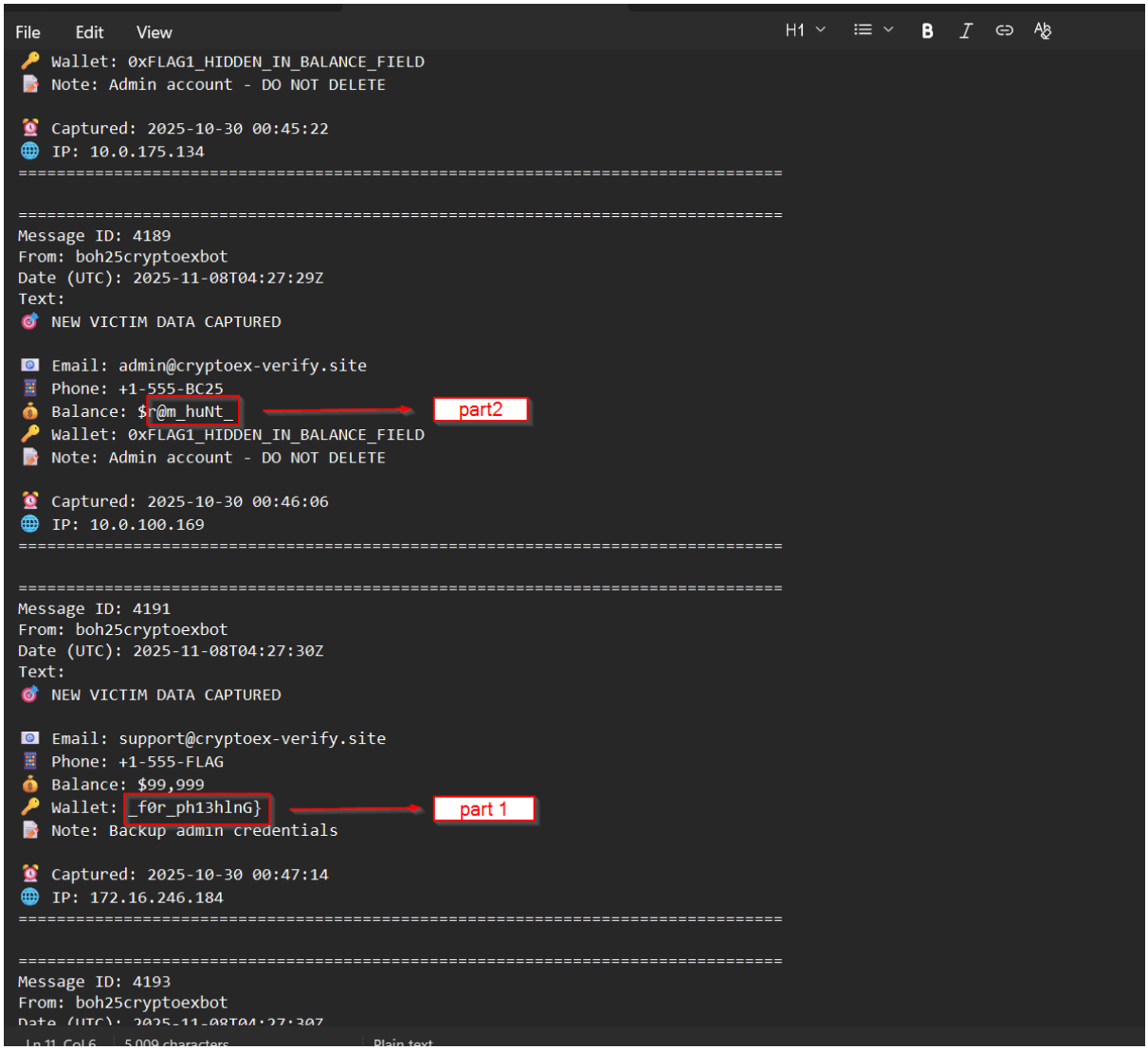
Phishing Website: https://boh25-osint-cryptoex.vercel.app



Once I submitted the form, I noticed there is a telegram API that use to interact with the bot.

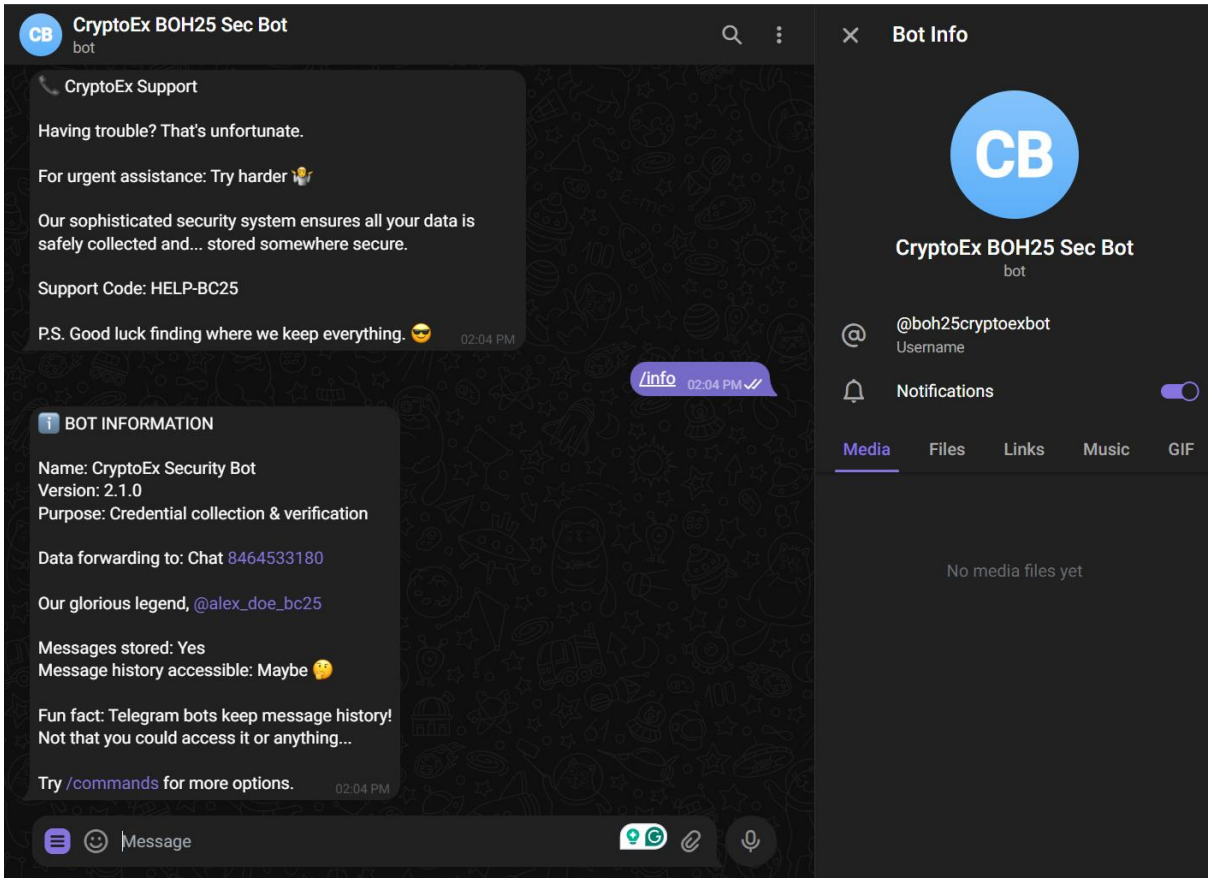


When I viewed the page source code I found the bot credential.

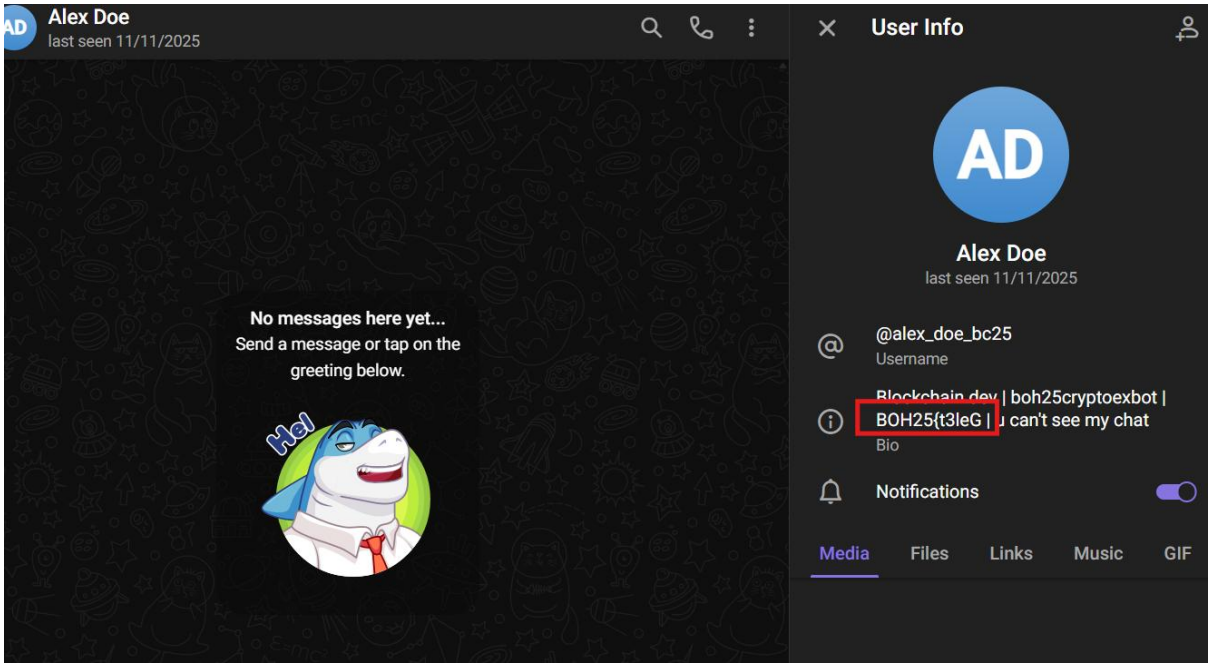


By using the script given by my teammate and modified by ChatGPT based on our requirement, I managed to dump all the information from the bots which include the second and third part of the flag. The script can be download here:

main (1).py



To get the first part of the flag we need to interact with the chatbot. '@boh25cryptoexbot' which the bot username can be found from the webpage in the first figure. When I asked for info, it will show another username '@alex_doe_bc25'



Check the user bio and we got the first part of the flag.