

TSN3251

CTF

Assignment

Writeup

Group Name: Sun_Microsystems

Members:

Student ID	Name	Role
1161100699	Muhamad Afif Bin Kamarul Imran	Leader
1191302540	Tan Shupei	Member
1191302473	Khairul Hanie Hazierah Binti Mohd Azmi	Member
1191302166	Cheok Yi Xuan	Member

Buffer Overflows

Stuff In Security !!!

Challenge 72 Solves ×

Stuff In Security !!!

30

Here's a simple program. What's the password?

ad82deb...

Flag Submit

Involved members: Cheok Yi Xuan

Thought process/methodologies:

- Debug the .exe file with Visual Studio Code
 - Input a random string when prompted to input a password
 - No result
 - Input a long string when prompted to input a password
 - Obtain the flag

- Flag Revealed: ***nexa{badcodefails}***

Cracking

Basic WiFi Cracking

Challenge 73 Solves X

Basic Wifi Cracking

50

Can you find the wifi password for me?
Flag Format: nexa{wifi_password}

View Hint

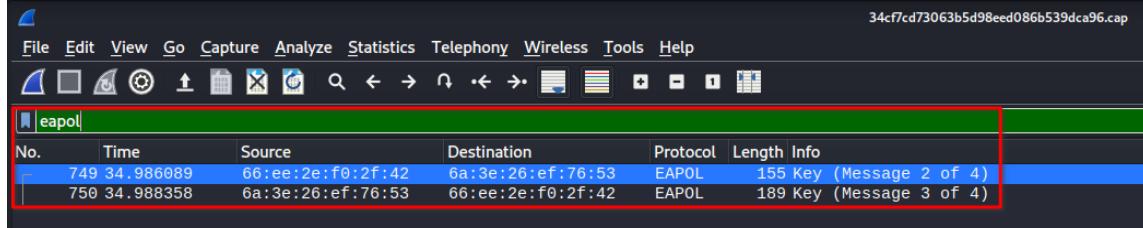
Download 34cf7cd7...

Flag Submit

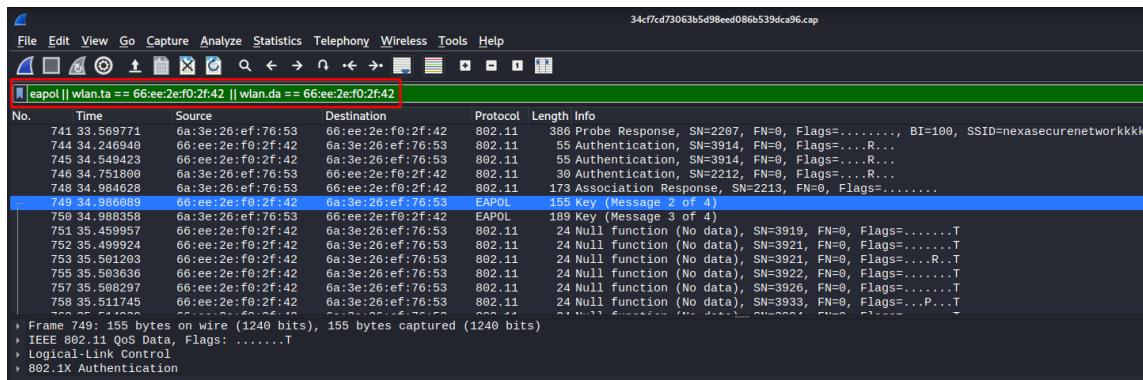
Members involved: Afif

Thought process/methodologies:

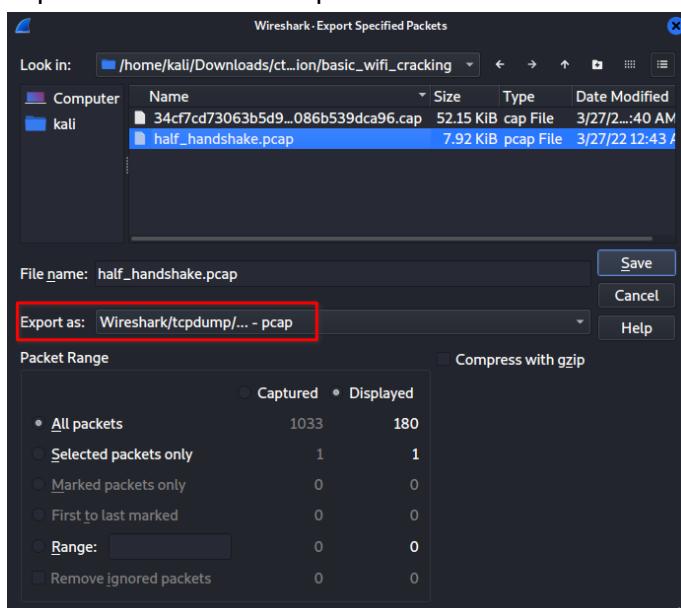
- Import packet file on *WireShark*
- Identify common Wi-Fi attack: *half-handshake* as the 4-way handshake is not complete (must be 4 out of 4)
- Search *eapol* flag to list its packets



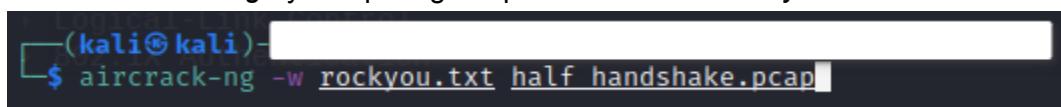
- Extract source address
- Append *wlan* source and destination addresses through piping with *eapol/*



- Export results as a new packet file



- Execute *aircrack-ng* by comparing the packet file with a *rockyou* breach wordlist



- Wi-Fi password found

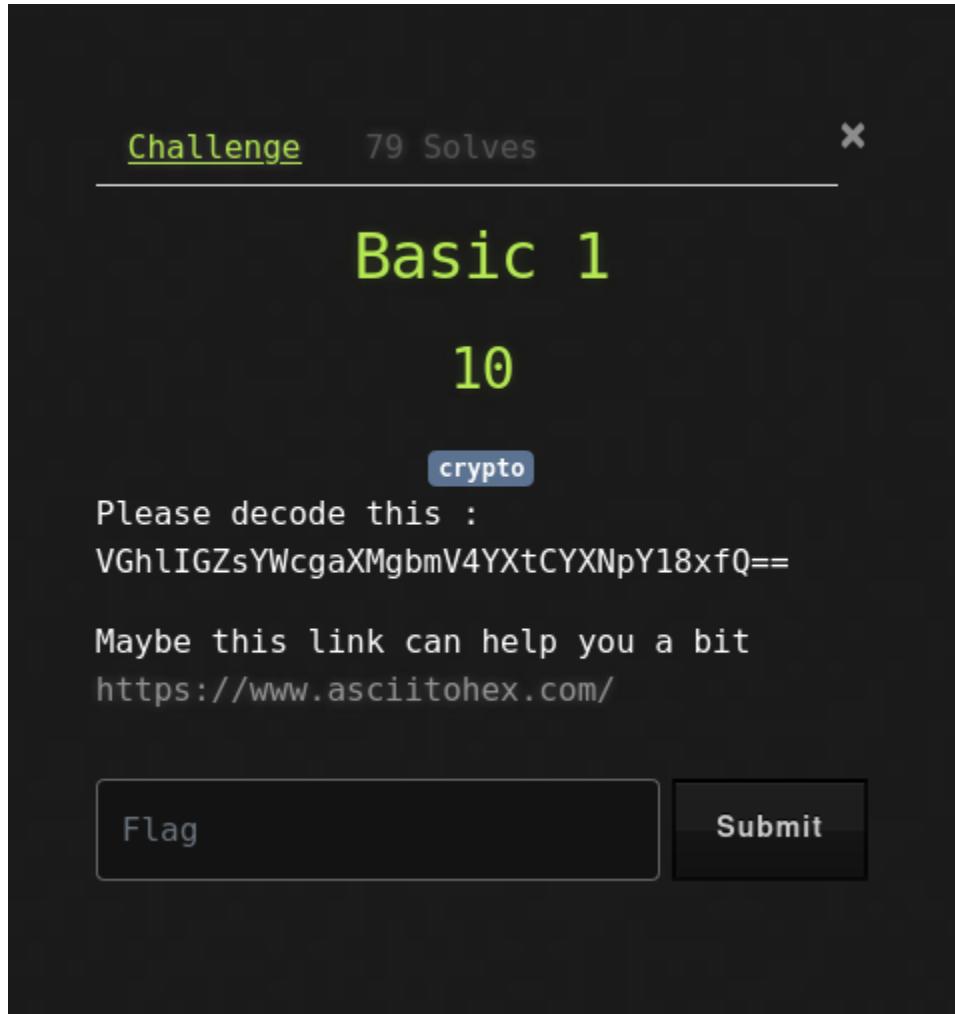
```
pol@wlan: ~ 66:ee:2e:f0:2f:42 Aircrack-ng 1.6
[00:15:29] 9642246/14344391 keys tested (10540.19 k/s)
Time left: 7 minutes, 26 seconds 67.22%
745 34.545423 66:ee:2e:f0:2f:42 6a:3e:20:ef:76:53
746 34.751800 66:ee:2e:f0:2f:42 6a:3e:20:ef:76:53
747 34.984628 66:ee:2e:f0:2f:42 6a:3e:20:ef:76:53
748 34.986089 66:ee:2e:f0:2f:42 6a:3e:20:ef:76:53
Master Key : D7 DD EC 07 8D 08 3E 0A CF A8 9F D2 1B 55 A5 EC
              1F 94 94 55 5C 5A 10 7F B9 C8 90 67 79 B5 A2 9E
Transient Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
EAPOL HMAC : B7 82 C5 21 6F A8 1C 76 D4 B1 96 BD 2A CD 24 A3
all 749: 155 bytes on wire (1240 bits), 155 bytes captured

```

- Flag revealed: ***nexa{betoerestmivid12}***

Cryptography

Basic 1



Challenge 79 Solves X

Basic 1

10

crypto

Please decode this :

VGh1IGZsYWcgaxMgbmV4YXtCYXNpY18xfQ==

Maybe this link can help you a bit

<https://www.asciiitohex.com/>

Members involved: Afif

Thought process/methodologies:

- Identify which string it belongs to with different conversion methods
- String given is identified as BASE64
- Input the string into BASE64 converter

ASCII to Hex

...and other free text conversion tools

Text (ASCII / ANSI)

```
I gave a cry of astonishment. I saw and thought
nothing of the other four Martian monsters; my
attention was riveted upon the nearer incident.
Simultaneously two other shells burst in the air near
the body as the hood twisted round in time to
receive, but not in time to dodge, the fourth shell.
```

Convert **Highlight Text**

Binary

```
01001001 00100000 01100111 01100001
01100110 00100101 00100000 01100001
00100000 01000111 01100100 01110001
00100000 01010111 01100110 00100000
01100001 01110011 0110100 01010111
01101110 01101001 01110011 0101000
01101101 01100101 01101110 01110010
00101110 00100000 01001001 00100000
```

Convert **Highlight Text**

Hexadecimal

```
49 20 67 61 76 65 20 61 20 63 72 79 20 6f 66 20 61
73 74 6f 6e 69 73 68 6d 65 6e 74 2e 20 49 20 73 61
77 20 61 6e 64 20 74 68 61 75 67 68 74 20 6e 6f 74
68 69 65 67 20 6f 66 20 74 68 65 20 6f 74 68 65 72
20 66 67 75 72 20 4d 61 72 74 69 61 6e 20 6d 6f 6e
73 74 65 72 73 3b 20 6d 79 20 61 74 74 65 6e 74 69
6f 6e 20 74 69 65 20 6e 65 61 72 65 72 20 69 6e 63
```

Convert **Highlight Text**

BASE64

```
GUgb3RoZKlgZm91ciBNYj0aWFuG1vbNn0ZxjzOy
~BteSBndIRlnRpzb24g2f2fHjdmV0ZWcgdXBvbIB
0aGUgbnVhcmVjYGluzV1zW50LbTaW1bIRhbm
VvdXNseSB0d2gb3R0ZXlg2hbgxzlGjcnN0J0gbm
HRoZ5BhaXjgbnVhcb0aGUgYm9ke5Bhcy0aAGUga
G9jZCB0d2ldgVhjyHjw5kIglUHRpWVJg0GBgm
VjZWI2Z5wgVnY0IG5vcPpb0B0W1lHrVGRzGdI
LCB0aGUgZm91cnRoHn0ZwxsLg==
```

Convert **Highlight Text**

Decimal

```
73 32 103 97 118 101 32 97 32 99 114 121 32 111
102 32 97 115 116 111 110 105 115 104 109 101
110 116 46 32 73 32 115 97 119 32 97 110 100 32
116 104 111 117 103 104 116 32 110 111 116 104
105 110 103 32 111 102 32 116 104 101 32 111 116
104 101 114 32 102 111 117 114 32 77 97 114 116
105 97 110 32 109 111 110 115 116 101 114 115 59
32 109 121 32 97 116 116 101 110 116 105 111 110
```

Convert **Highlight Text**

URL Encoded

- ROT13

ASCII to Hex

...and other free text conversion tools

Text (ASCII / ANSI)

```
The flag is nexa{Basic_1};
```

Convert **Highlight Text**

Binary

```
01010100 01101000 01100101 00100000 01100110
01101100 01100001 01100111 00100000 01100001
01110011 00100000 01101110 01100101 01110000
01100001 01110011 01000000 01100001 01100011
01101001 01100011 01011111 00110001 01111101
```

Convert **Highlight Text**

Hexadecimal

```
54 68 65 20 66 6c 61 67 20 69 73 20 6e 65 78 61 7b
42 61 73 69 63 5f 31 7d
```

Convert **Highlight Text**

BASE64

```
VGHlGZsYWcgxMgbmV4YXtCYXNpY18xfQ==
```

Convert **Highlight Text**

Decimal

```
84 104 101 32 102 108 97 103 32 105 115 32 110
101 120 97 123 66 97 115 105 99 95 49 125
```

Convert **Highlight Text**

URL Encoded

- ROT13
- Flag revealed after conversion: ***nexa{Basic_1}***

Basic 2

Challenge 79 Solves X

Basic 2

10

```
.... --- ... / ... . . . - . / - . . . .  
/ -- . . . . . - - - - - - - - / . . . .  
. . . . . . . . / - . . . . / . . . . .  
- - - . / - - - . . . . . - - - - - - -  
- - - - . . . . . . . . . . . . . . . . .  
- - - - .
```

You know?

Flag format: nexa{<flag_flag>}

Members involved: Afif

Thought process/methodologies:

- Given input string is identified as morse code
 - Convert the string into plaintext



- Flag revealed: *nexa{MORSE_CODE_IS_NICE}*

Intermediate 1

Challenge 78 Solves X

Intermediate 1

20

"Dear Decision maker ; Especially for you - this cutting-edge information . This is a one time mailing there is no need to request removal if you won't want any more . This mail is being sent in compliance with Senate bill 1626 ; Title 6 , Section 304 . This is a legitimate business proposal . Why work for somebody else when you can become rich within 86 WEEKS . Have you ever noticed the baby boomers are more demanding than their parents plus more people than ever are surfing the web ! Well, now is your chance to capitalize on this . We will help you use credit

I think there is hidden message in above mail, can you decode it for me?

Flag Submit

Members involved: Cheok Yi Xuan

Thought process/methodologies:

- Google the text provided
- Decode the text at <http://www.spammimic.com/decode.shtml>

The screenshot shows the 'Decode' page of spammimic.com. At the top, it says 'Decode'. Below that is a text area with a placeholder 'Paste in a spam-encoded message'. The message content is a long, encoded spam text. Below the message is a 'Decode' button. Underneath the message, it says 'Alternate decodings:' followed by a list of links: 'Decode spam with a password', 'Decode fake spreadsheet NEW', 'Decode fake PGP', 'Decode fake Russian', and 'Decode space'. At the bottom of the page, there is a copyright notice: 'Copyright © 2000-2020 spammimic.com. All rights reserved.'

The screenshot shows the 'Decoded' page of spammimic.com. At the top, it says 'Decoded'. Below that, it says 'Your spam message Dear Decision maker ; Especially for you... decodes to:'. There is a text input field containing 'nexa{intermediate_2}' and an 'Encode' button. Below the input field, it says 'Look wrong? try the [old version](#)'. At the bottom, there is a copyright notice: 'Copyright © 2000-2020 spammimic.com. All rights reserved.'

- Flag revealed: ***nexa{intermediate_2}***

Intermediate 2

Challenge 77 Solves X

Intermediate 2

20

1104101412049741234100410149941054109497
4108495410541154954974119410141154111410
941014125

Is there any meaning of this number?

FlagSubmit

Members involved: Afif

Thought process/methodologies:

- Converting decimal to plaintext
- String given is identified as a decimal or base-10
- Convert the given input into plaintext



- Flag revealed: ***nexa{decimal_is_awesome}***

Intermediate 3

Challenge 78 Solves ×

Intermediate 3

20

YXJrbntVbmVxX2xyZ19wbmFfb3JfZmJ5aXJxfQ==

Base64 is not enough to know the message...help me! !

Flag Submit

Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- The question states that base 64 is not enough.
- Firstly, the message given is decoded through the Base64 converter. It will decode the message, however, the message is not decoded thoroughly yet.

BASE64

```
YXJrbntVbmVxX2xyZ19wbmFfb3JfZmJ5aXJxfQ==
```

Convert **Highlight Text**

Text (ASCII / ANSI)

```
arkn{Uneq_lrg_pna_or_fbyirq}
```

Convert **Highlight Text**

ROT13

```
nex{Hard_yet_can_be_solved}
```

Convert **Highlight Text**

- Using ROT13, the flag is revealed to be ***nex{Hard_yet_can_be_solved}***

Hard 1

Challenge 5 Solves X

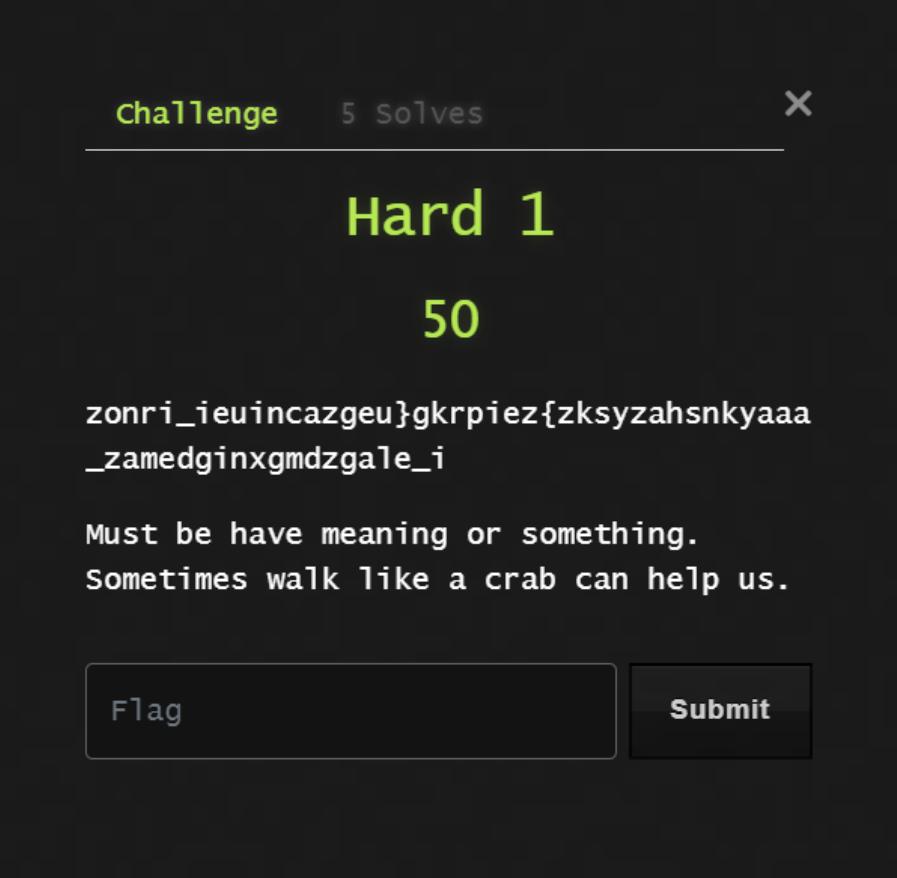
Hard 1

50

zonri_ieuincnazgeu}gkrpiez{zksyzahsnkyaaa
_zamedginxgmdzgale_i

Must be have meaning or something.
Sometimes walk like a crab can help us.

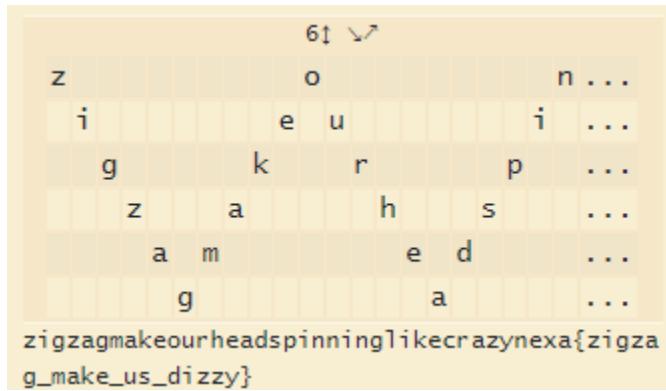
Flag Submit



Members involved: Tan Shupei, Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- Tan Shupei:
 - As it is mentioned in the question that “walking like a crab can help us”, two-square cipher (horizontal two-square) is used.
 - “Must be have meaning or something. “ and “Sometimes walk like a crab can help us. “ were used as the secret keywords in the two-square cipher decryption attempt.
 - However, the decryption did not take place successfully.
- Khairul Hanie Hazierah Binti Mohd Azmi:
 - The ciphertext is analyzed using the hint given, which is “Sometimes walk like a crab can help us”
 - The hint indicates that the ciphertext letters are arranged diagonally/zigzag, which hints at the rail fence cipher.



The image shows a 6x6 grid of letters and a decoded message. The grid is as follows:

z			o		n ...
i			e	u	i ...
g		k	r	p	...
z	a		h	s	...
a	m		e	d	...
g			a		...

Below the grid, the decoded message is shown:

zigzagmakeourheadspinninglikecrazynex{zigzag_make_us_dizzy}

- Using $k = 6$, the flag is revealed to be **nex{zigzag_make_us_dizzy}**

Hard 2

challenge 22 solves X

Hard 2

50

You managed unlock this level!

```
01100001 01100101 01100111 00110010
01100001 01100110 01111010 01100111
01101000 01100001 01101110 01101100
01111001 01101110 01111010 01100111
01100110 01110011 01100111 01101101
01110010 01100100 01101001 01101001
01111010 01100101 01100111 01011111
01101111 01100101 01101001 01100001
01110101 01110011 01101110 01101011
01100001 01111000 01101001 01110111
01011111 01110100 01111010 01101011
01101111 01110000 01101110 01100101
01110010 01100001 01111010 01101001
01101000 01111101 01100101 01101001
01100011 01111011 01110100
```

I manage to decode this binary, but cant think about next step. The only hint i get is "Perhaps the beat is off..try 1-8"

Flag Submit

Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

VIEW

Bytes ▾

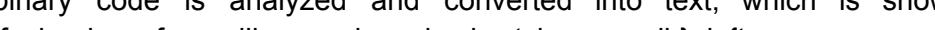
FORMAT	GROUP BY
Binary	Byte
01100001 01100101 01100111 00110010 01100001 01100110	
01111010 01100111 01101000 01100001 01101110 01101100	
01111001 01101110 01111010 01100111 01100110 01110011	
01100111 01101101 01110010 01100100 01101001 01101001	
01111010 01100101 01100111 01011111 01101111 01100101	
01101001 01100001 01110101 01110011 01101110 01101011	
01100001 01111000 01101001 01110111 01011111 01110100	
01111010 01101011 01101111 01110000 01101110 01100101	
01110010 01100001 01111010 01101001 01101000 01111101	
01100101 01101001 01100011 01111011 01110100	

VIEW

3

Text ▾

aeg2afzghanlynzgfgmrdiizeg_oeiausnkaxiw_tzkopnerazih}ei
cft

- The binary code is analyzed and converted into text, which is shown to be aeg2afzghanlynzgsgmrdiizeg_oeiausnkaxiw_tzkopnerazih}eic{t


fset]zigzagmakeourheadspinninglikecrazy2nexa
fziqzaq_with_of
 - Using $k = 6$, the plaintext is starting to make sense, however, the arrangement of the letters is off.

```
6↑ ↘ (+6)  
a e ...  
z g h ...  
g m r ...  
i a u ...  
z k o ...  
e ...  
zigzagmakeourheadspinninglikecrazy2nexa{zigz  
aq_with_offset}
```

- Using the hint given, “Perhaps the beat is off..try 1-8”, with the combination of offset = 6 and k = 6, the flag is revealed to be **nexa{zigzag_with_offset}**

Forensic

Basic 3



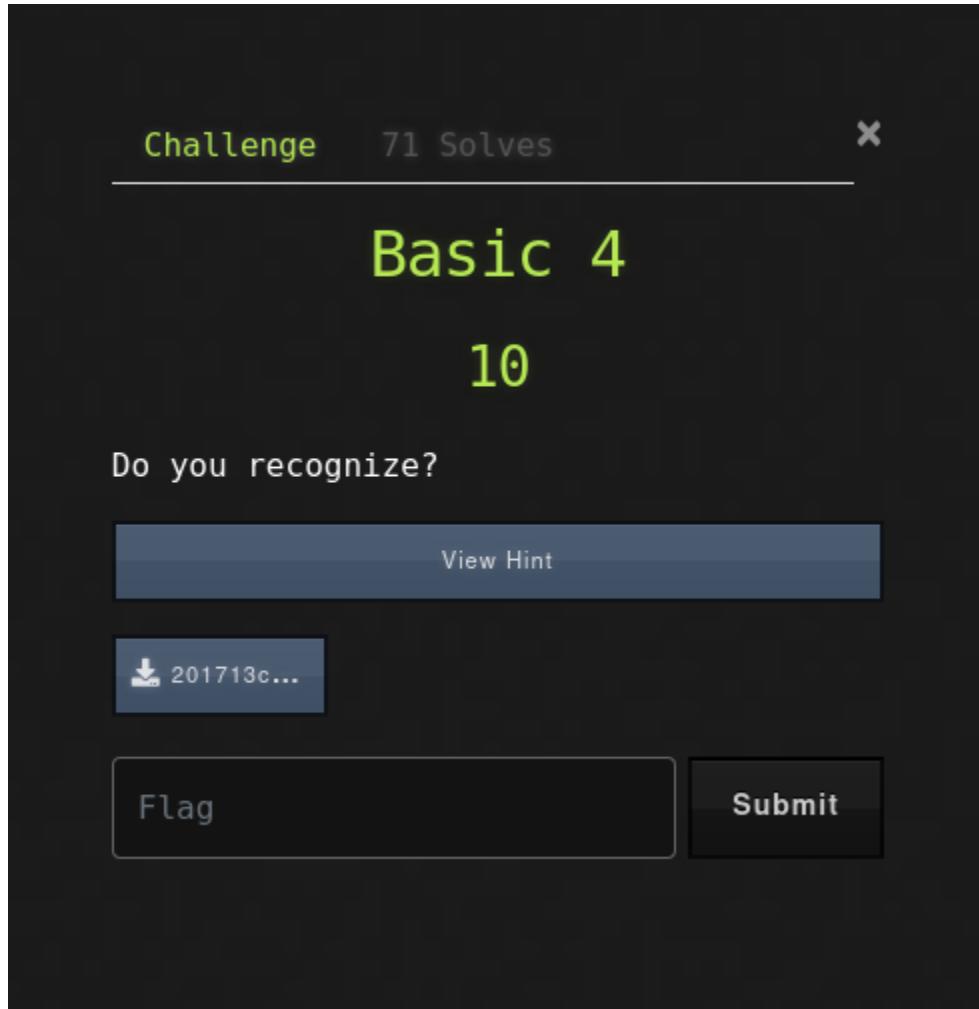
Members involved: Cheok Yi Xuan

Thought process/methodologies:

- Open the file as text file

- Flag revealed: *nexa/stego, basically*

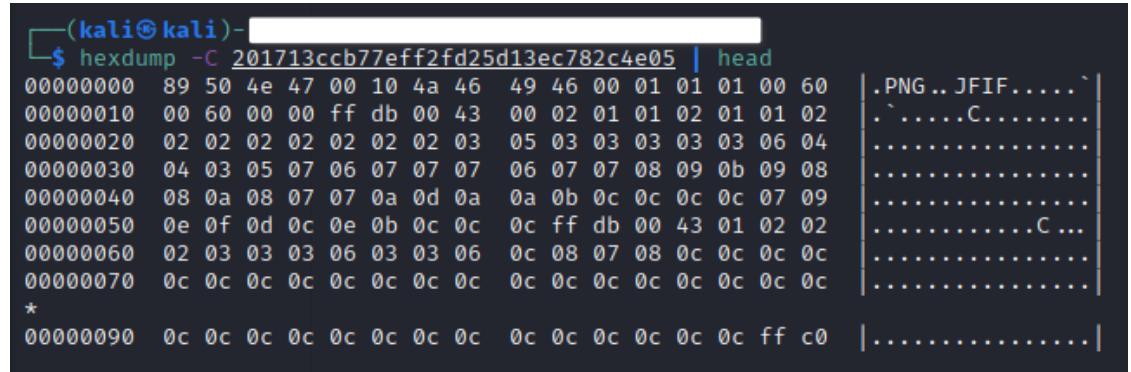
Basic 4



Members involved: Afif, Tan Shupei

Thought process/methodologies:

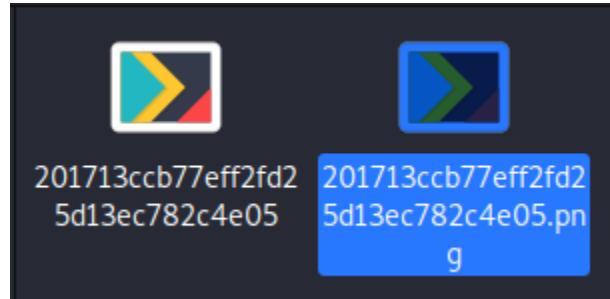
- Afif
 - Identify file type by dumping its hex content in hex+ASCII format only from the beginning of the file



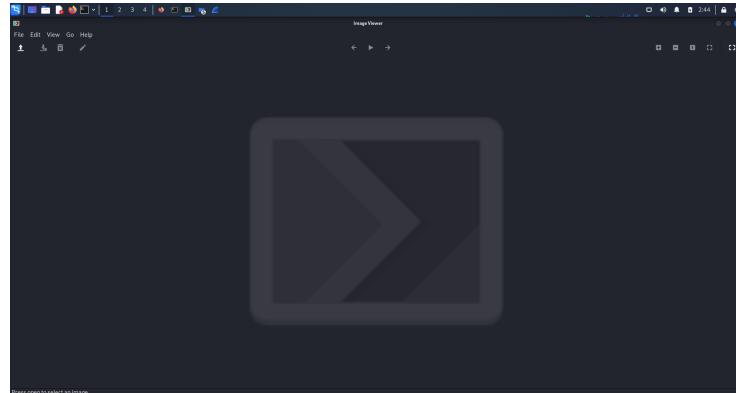
```
(kali㉿kali)-[~]
$ hexdump -C 201713ccb77eff2fd25d13ec782c4e05 | head
00000000  89 50 4e 47 00 10 4a 46  49 46 00 01 01 01 01 00 60  .PNG..JFIF....` 
00000010  00 60 00 00 ff db 00 43  00 02 01 01 02 01 01 02  .`.....C.....` 
00000020  02 02 02 02 02 02 03  05 03 03 03 03 03 03 06 04  .`.....`.....`.....` 
00000030  04 03 05 07 06 07 07 07  06 07 07 08 09 0b 09 08  .`.....`.....`.....` 
00000040  08 0a 08 07 07 0a 0d 0a  0a 0b 0c 0c 0c 0c 07 09  .`.....`.....`.....` 
00000050  0e 0f 0d 0c 0e 0b 0c 0c  0c ff db 00 43 01 02 02  .`.....`.....`.....` 
00000060  02 03 03 03 06 03 03 06  0c 08 07 08 0c 0c 0c 0c  .`.....`.....`.....` 
00000070  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c 0c  .`.....`.....`.....` 
* 
00000090  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c ff c0  .`.....`.....`.....` 


```

- File type appears to be in *PNG*
- Rename the file by adding the *PNG* extension behind it



- Upon opening the file, no image is shown and appears to be corrupted

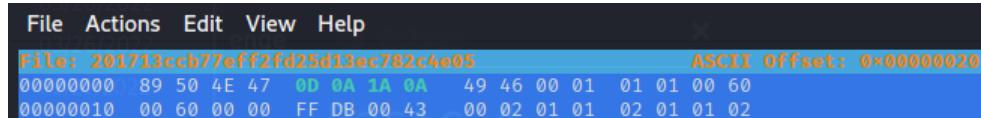


- Unable to proceed from here...

- Tan Shupei
 - Hexdump was used to check the hex content of the file first to see if there is any corruption in the file header.

```
(tan2540㉿kali)-[~/Downloads]
└─$ hexdump -C 201713ccb77eff2fd25d13ec782c4e05
00000000  89 50 4e 47 00 10 4a 46  49 46 00 01 01 01 00 60  |.PNG..JFIF....`|
00000010  00 60 00 00 ff db 00 43  00 02 01 01 02 01 01 02  |.~....C.....|
00000020  02 02 02 02 02 02 03  05 03 03 03 03 03 06 04  |.....|
00000030  04 03 05 07 06 07 07 07  06 07 07 08 09 0b 09 08  |.....|
00000040  08 0a 08 07 07 0a 0d 0a  0a 0b 0c 0c 0c 0c 07 09  |.....|
00000050  0e 0f 0d 0c 0e 0b 0c 0c  0c ff db 00 43 01 02 02  |.~....C...|
00000060  02 03 03 03 06 03 03 06  0c 08 07 08 0c 0c 0c 0c  |.....|
00000070  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c  |.....|
```

- By comparing the file signature of a PNG file and the one in the screenshot above, there might be some corruption in the file header as the actual file signature for PNG file should be **89 50 4E 47 0D 0A 1A 0A**, thus hexeditor was used to modify the header.



- However, the modification was not successful as no image was shown while attempting to open the modified file.

```
(tan2540㉿kali)-[~/Downloads]
└─$ hexdump -C 201713ccb77eff2fd25d13ec782c4e05png | head
00000000  89 50 4e 47 0d 0a 1a 0a  49 46 00 01 01 01 00 60  |.PNG....IF....`|
00000010  00 60 00 00 ff db 00 43  00 02 01 01 02 01 01 02  |.~....C.....|
00000020  02 02 02 02 02 02 03  05 03 03 03 03 03 06 04  |.....|
00000030  04 03 05 07 06 07 07 07  06 07 07 08 09 0b 09 08  |.....|
00000040  08 0a 08 07 07 0a 0d 0a  0a 0b 0c 0c 0c 0c 07 09  |.....|
00000050  0e 0f 0d 0c 0e 0b 0c 0c  0c ff db 00 43 01 02 02  |.~....C...|
00000060  02 03 03 03 06 03 03 06  0c 08 07 08 0c 0c 0c 0c  |.....|
00000070  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c  |.....|
*          00000090  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c ff c0  |.....|
```

After running through a pngcheck, it was shown that the modified file has an invalid chunk name.

```
(tan2540㉿kali)-[~/Downloads]
└─$ pngcheck -v 201713ccb77eff2fd25d13ec782c4e05png
File: 201713ccb77eff2fd25d13ec782c4e05png (21263 bytes)
  invalid chunk name "" (01 01 00 60)
  ERRORS DETECTED in 201713ccb77eff2fd25d13ec782c4e05png
```

- After comparing the modified file's header with a valid PNG file's header, another idea came in mind whereby the original file might be a JPEG file rather than a PNG file. Therefore another attempt was made, which is to modify the file header into a JPEG file header.

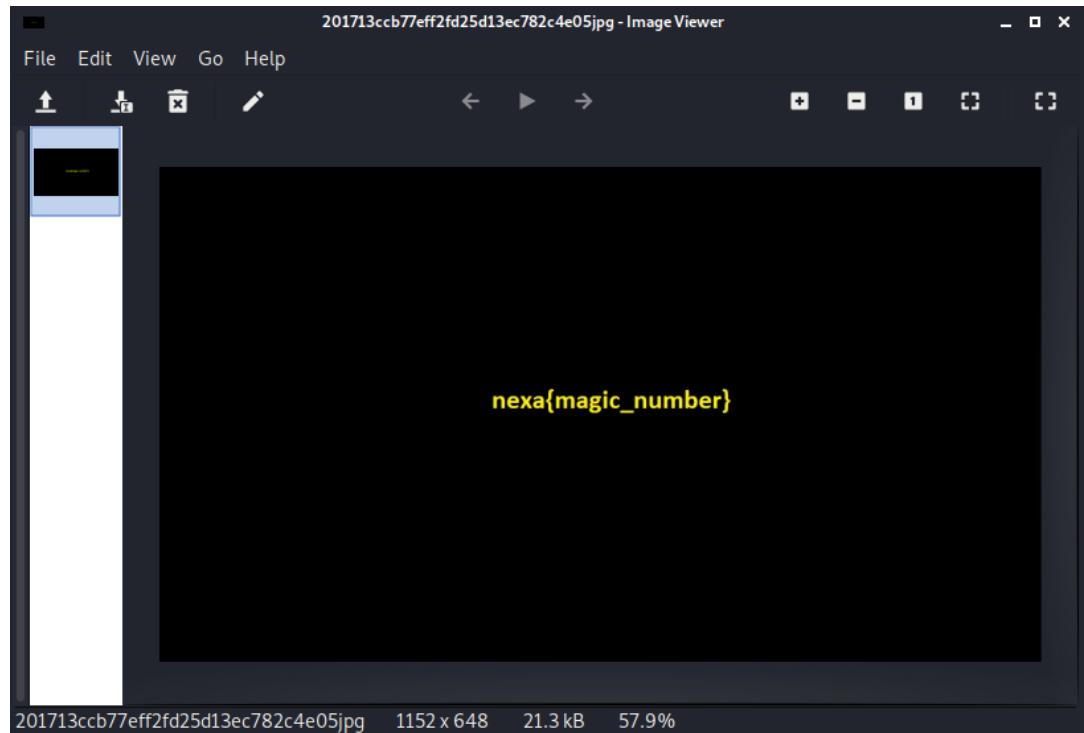
```
(tan2540㉿kali)-[~/Downloads]
└─$ hexdump -C image-1648235030658.png | head
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG.....IHDR|
```

- As the file signature of a JPEG file is **FF D8 FF E0 00 10 4A 46 49 46 00 01**, it matches more of the original file's hex values. Therefore, hexeditor was

used again to modify the header.

```
(tan2540㉿kali)-[~/Downloads]
$ hexdump -C 201713ccb77eff2fd25d13ec782c4e05.jpg | head
00000000  ff d8 ff e0 00 10 4a 46  49 46 00 01 01 01 01 00 60  | .....JFIF....`|
00000010  00 60 00 00 ff db 00 43  00 02 01 01 02 01 01 02 00 02  | `.....C.....|
00000020  02 02 02 02 02 02 03  05 03 03 03 03 03 03 06 04 00 00  | .....|
00000030  04 03 05 07 06 07 07 07  06 07 07 08 09 0b 09 08 00 00  | .....|
00000040  08 0a 08 07 07 0a 0d 0a  0a 0b 0c 0c 0c 0c 07 09 00 00  | .....|
00000050  0e 0f 0d 0c 0e 0b 0c 0c  0c ff db 00 43 01 02 02 00 00  | .....C...|
00000060  02 03 03 03 06 03 03 06  0c 08 07 08 0c 0c 0c 0c 00 00  | .....|
00000070  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c 0c 0c 00 00  | .....|
*
00000090  0c 0c 0c 0c 0c 0c 0c 0c  0c 0c 0c 0c 0c 0c ff c0 00 00  | .....|
```

- This time, the attempt was successful and the image was returned in a JPEG file as shown below.



- Flag revealed: ***nexa{magic_number}***
 - Unfortunately, the flag was found after the first CTF challenge while going through the questions again and reattempting them therefore we did not manage to get the marks for this question during CTF Challenge 1.

Normal PCAP: Part 1

Challenge 79 Solves ×

Normal PCAP: Part 1

20

Find the flag on the web service!

[View Hint](#)

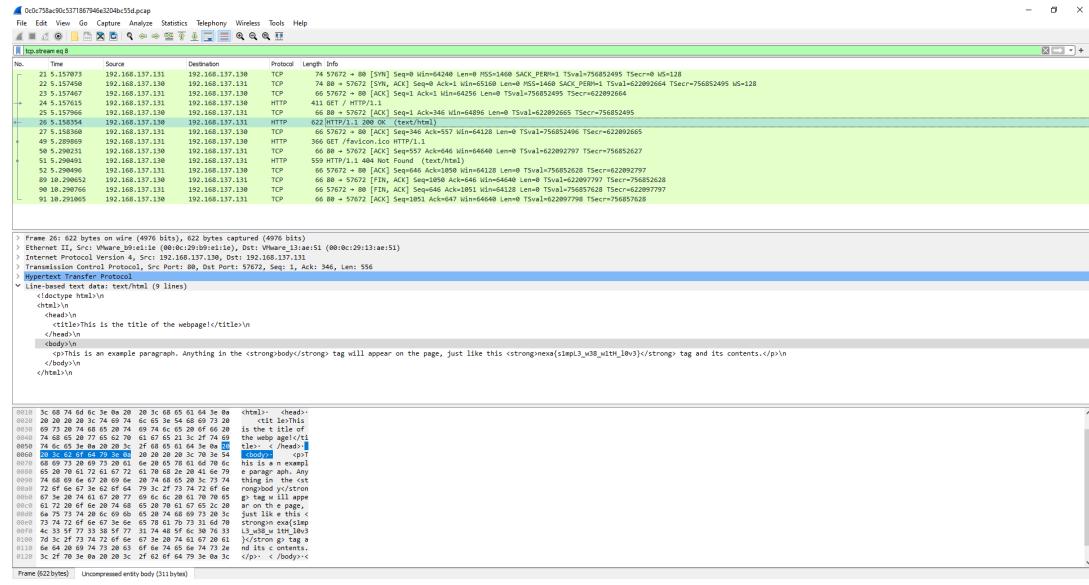
 0c0c758...

Flag Submit

Members involved: Cheok Yi Xuan

Thought process/methodologies:

- View the packet with Wireshark
- Find the packet with longer length



- Flag: **nexa{simpL3_w38_w1tH_I0v3}**

Normal PCAP: Part 2

Challenge 79 Solves ×

Normal PCAP: Part 2

20

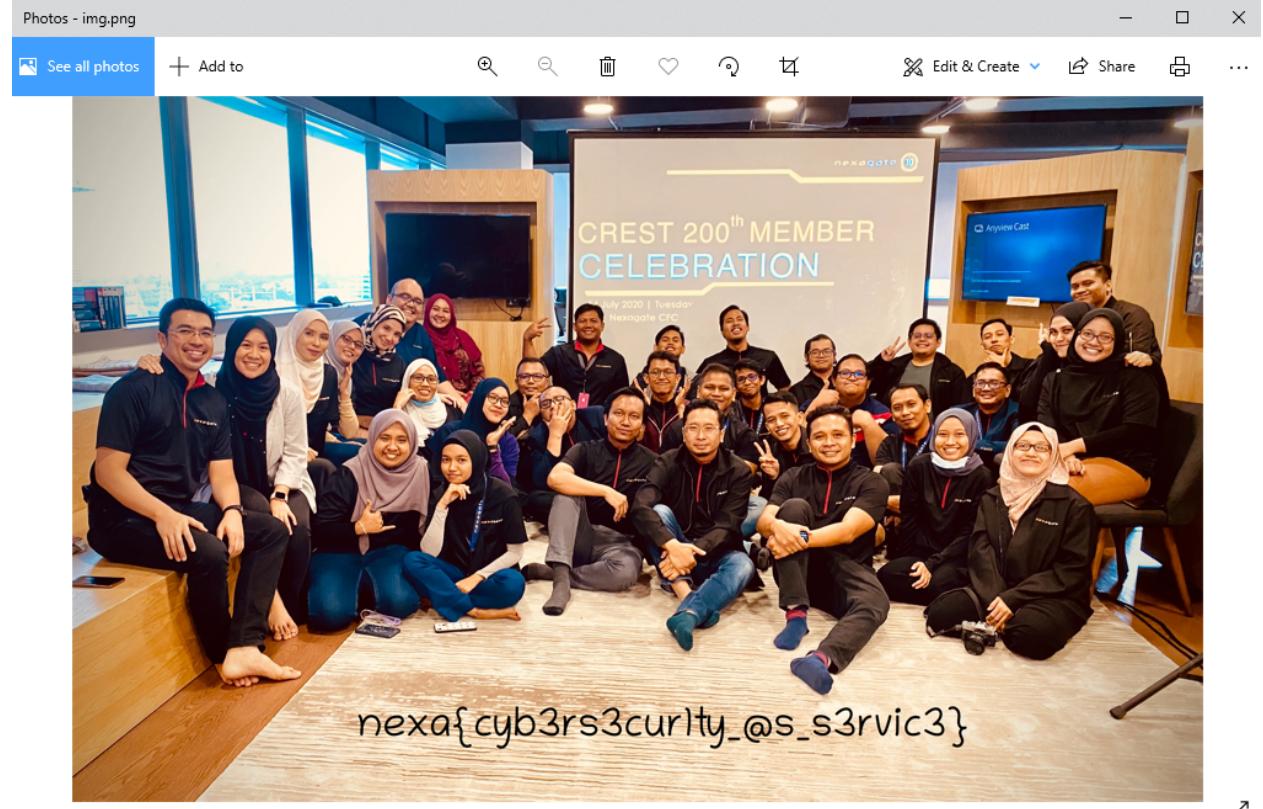
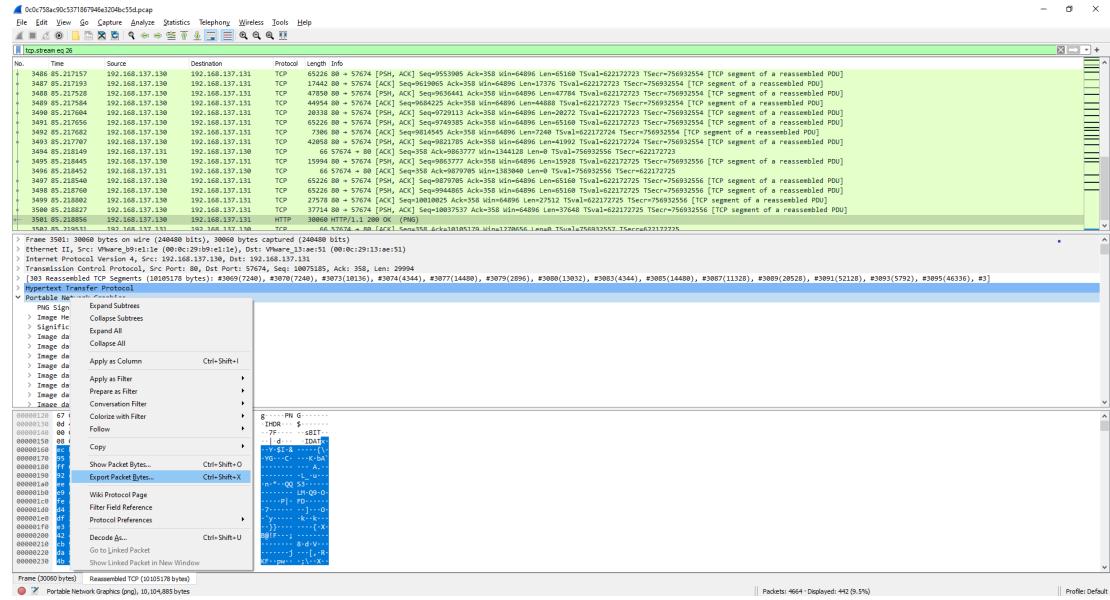
Find the flag on the web attachment!
(use same attachment from "Normal PCAP: Part 1")

Flag Submit

Members involved: Cheok Yi Xuan

Thought process/methodologies:

- View the file with Wireshark
- Find the PNG attachment
- Build the packet into a file



- Flag: **nexa{cyb3rs3cur1ty_@s_s3rvic3}**

Shift Your Focus



Challenge 70 Solves X

Shift your focus

20

Find the **flag** in this text document!!!!

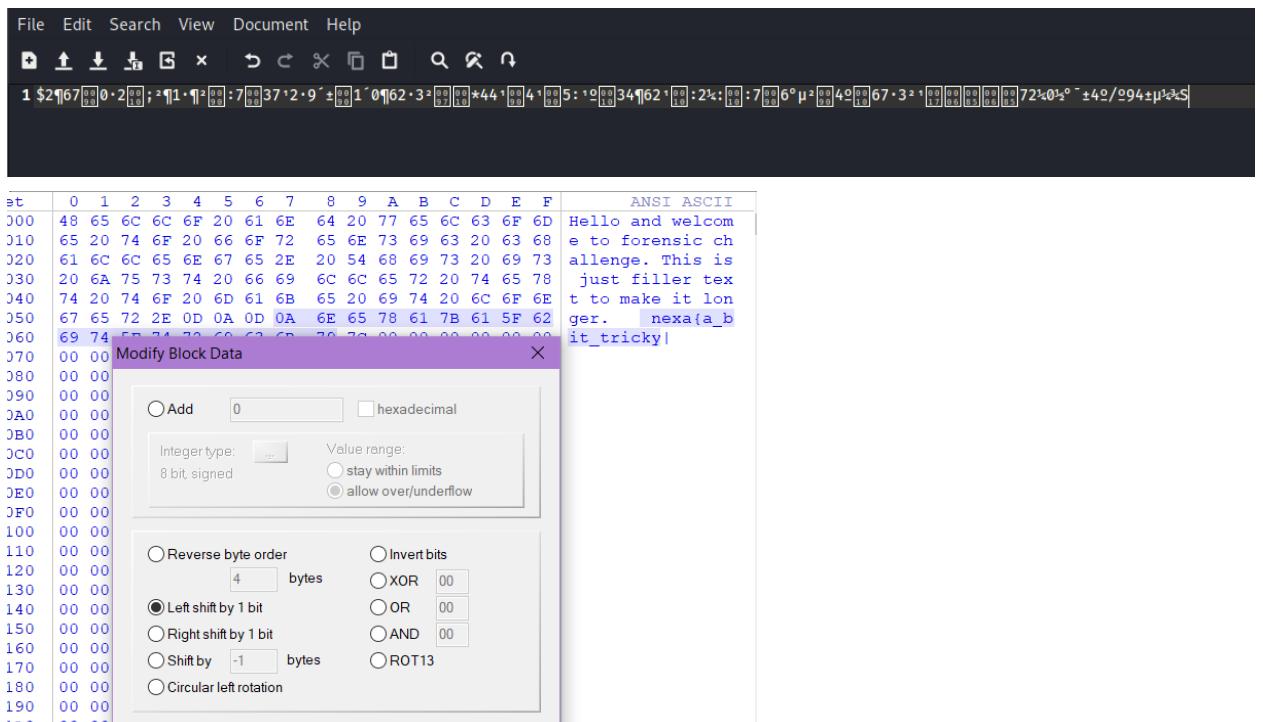
 43cddc4...

Flag Submit

Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- The question given is about finding the flag in the text document.
- The given text file is opened and analyzed.



- By using a hex editor and shifting the bits to the left once, the flag is revealed to be **nexa{a_bit_tricky}**

Normal PCAP: Part 3

Challenge 74 Solves X

Normal PCAP: Part 3

30

Find the flag on the netcat communication! (use same attachment from "Normal PCAP: Part 1")

Flag

Submit

Members involved:

Thought process/methodologies:

Normal PCAP: Part 4

Challenge 79 Solves X

Normal PCAP: Part 4

30

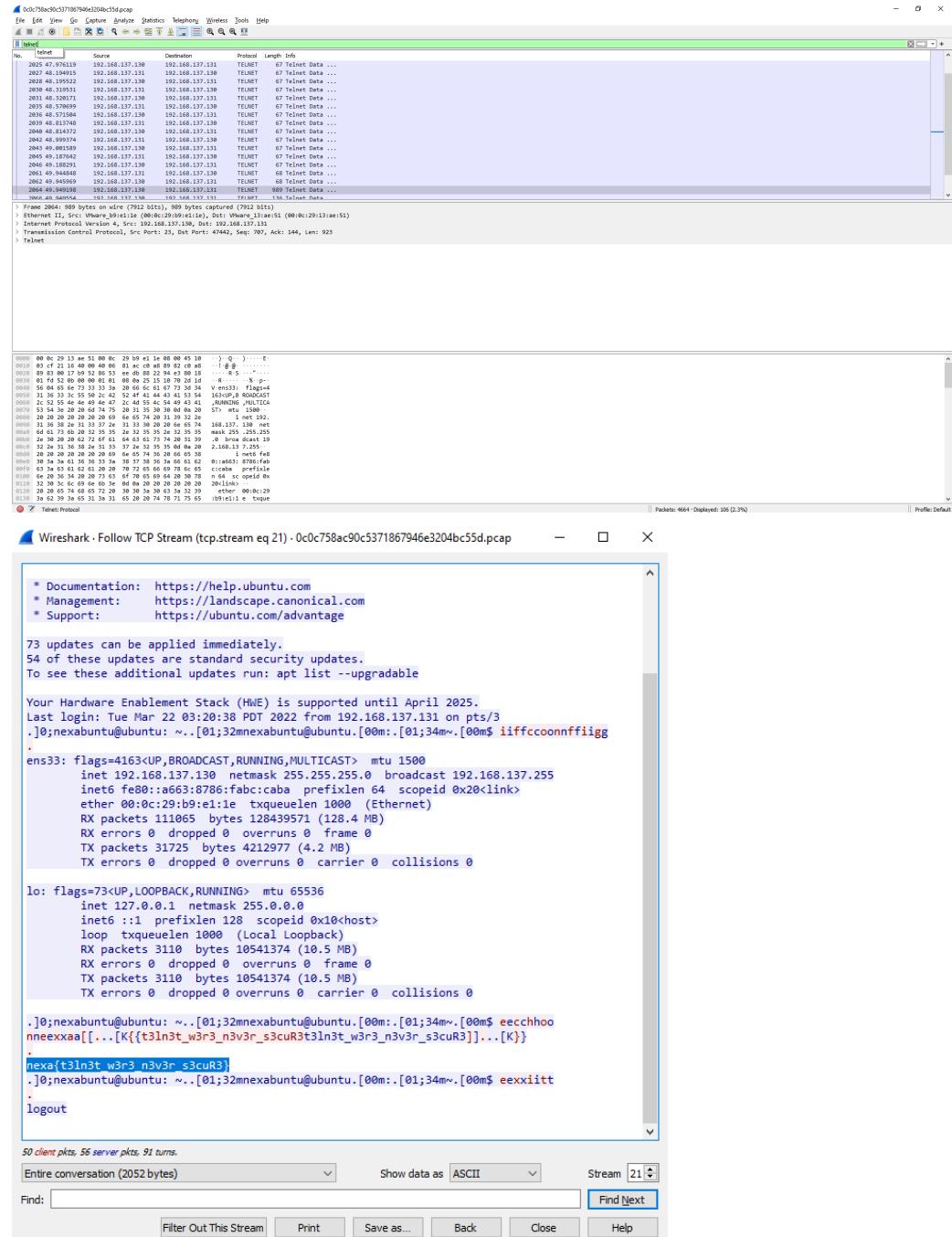
Find the flag on the telnet connection!
(use same attachment from "Normal PCAP:
Part 1")

Submit

Members involved: Cheok Yi Xuan

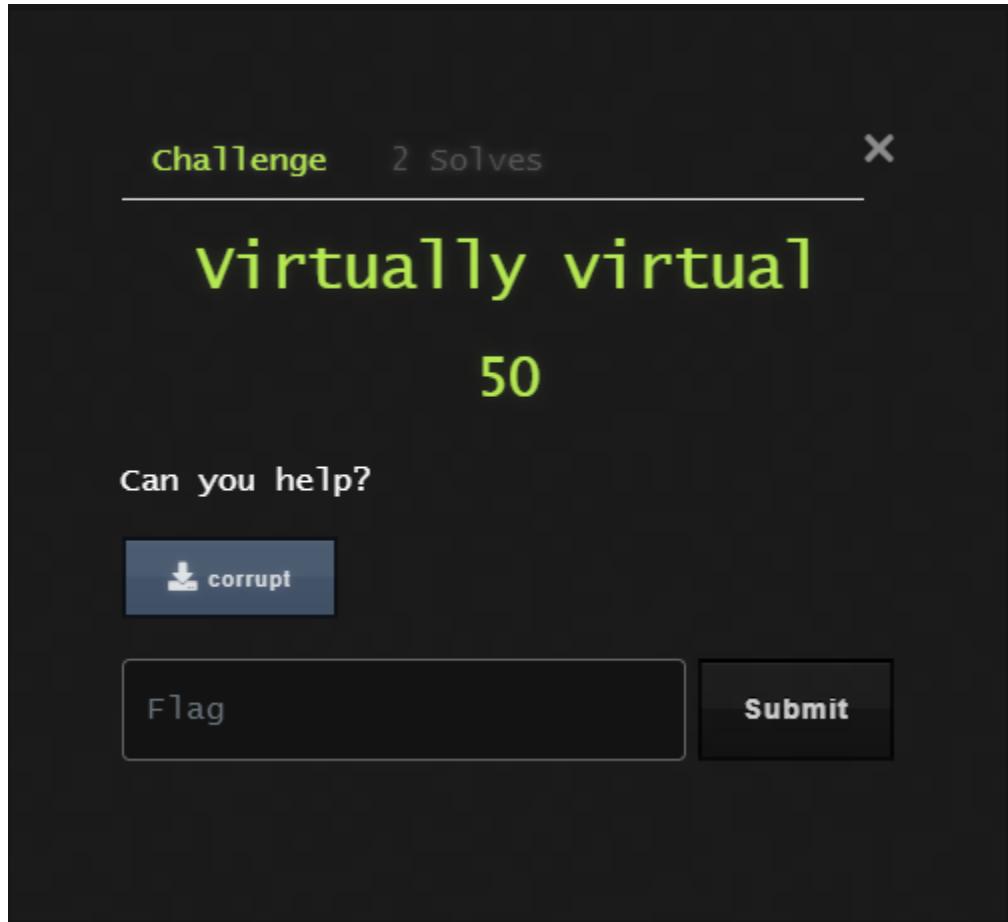
Thought process/methodologies:

- Open the file with Wireshark
- Sort by telnet



- Flag: **nexa{t3ln3t_w3r3_n3v3r_s3cuR3}**

Virtually Virtual



Members involved: Afif, Tan Shupei, Cheok Yi Xuan

Thought process/methodologies:

- Afif:

- Download a given file
 - Identify the file type with *file* on a terminal

```
[S] file corrupt
corrupt: DOS/MBR boot sector MS-MBR Windows 7 english at offset 0x163 "Invalid partition table" at offset 0x17b "Error loading operating system" at offset 0x19a "Missing operating system", disk signature 0x8f9e04e2; partition 1 : ID:0x0e, start-CHS (0x0,2,3), end-CHS (0x2,207,13), startsector 128, 45056 sectors
```

- File seems to be an *MBR* partition
 - Perform *MBR* partition analysis by using *hexeditor* included in Kali Linux

- 000001F0 00 00 00 00 00 00 00 00 00 00 00 00 55 AAU.

#	Flag	Type	Starting LBA address	Size
1	0x00	0x0e	0x80	0xB000
2	0x00	0x00	0x00	0x00
3	0x00	0x00	0x00	0x00
4	0x00	0x00	0x00	0x00

- partition #1:
 - 0x00: non bootable
 - 0x0e: FAT-16 LBA
 - 0x80: 128
 - 0xB000: 45,056
 - $45,056 * 512 = 366,957,580$ bytes
 - bytes to gigabytes: roughly 0.337GB
 -
 - Partition number one is a FAT-16 type, seems to contain roughly 337MB of data
 - Unable to proceed from here...

- Tan Shupei

- As the file type is unknown, hexdump was used to check the file header.

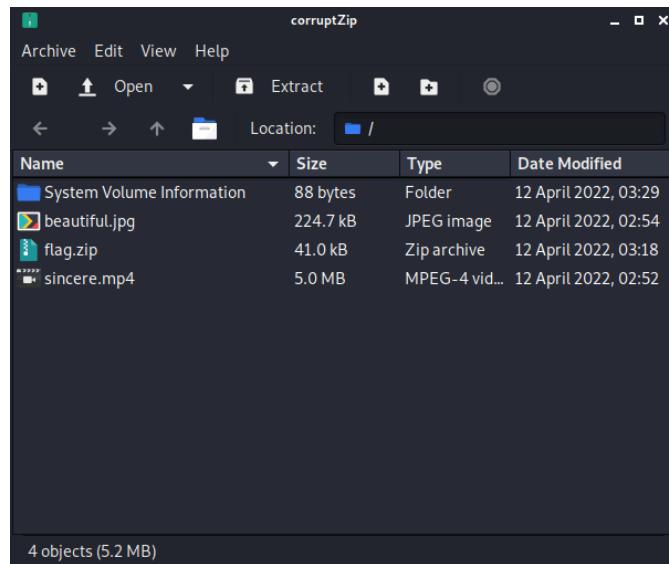
```
(tan2540㉿kali)-[~/Downloads]
$ hexdump -C corrupt | head
00000000 33 c0 8e d0 bc 7c 8e c0 8e d8 be 00 7c bf 00 |3....|.....|..|
00000010 06 b9 00 02 fc f3 a4 50 68 1c 06 cb fb b9 04 00 |.....Ph.....|
00000020 bd be 07 80 7e 00 00 7c 0b 0f 85 0e 01 83 c5 10 |.....~|.....|
00000030 e2 f1 cd 18 88 56 00 55 c6 46 11 05 c6 46 10 00 |.....,U.F..F..|
00000040 b4 41 bb aa 55 cd 13 5d 72 0f 81 fb 55 aa 75 09 |A.U...]r...U.u..|
00000050 c1 c1 00 01 74 03 fe 46 10 66 60 80 7e 10 00 74 |....t.F.f..~t..|
00000060 26 66 68 00 00 00 00 00 66 ff 76 08 68 00 00 68 00 |&fh...f.v.h..h..|
00000070 7c 68 01 00 68 10 00 b4 42 8a 56 00 8b f4 cd 13 |||h..h...B.V....|
00000080 f9 83 c4 10 9e eb 14 b8 01 02 bb 00 7c 8a 56 00 |.....,V.|
00000090 8a 76 01 8a 4e 02 8a 6e 03 cd 13 66 61 73 1c fe |.v..N..n...fas..|
```

However, after going through [the list of file signatures](#), nothing similar was found.

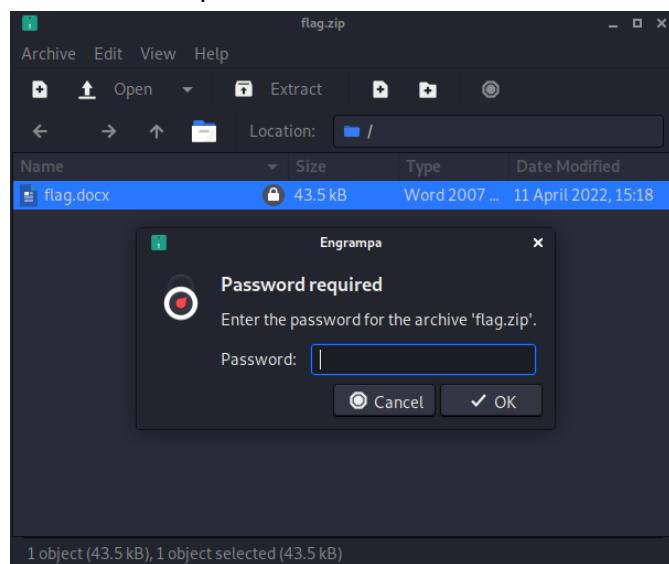
- From the hint given, the file could be a .zip file therefore an attempt was made to modify the file header to a .zip file's signature as shown below.

```
(tan2540㉿kali)-[~/Downloads]
$ hexdump -C corruptZip | head
00000000  50 4b 03 04 bc 00 7c 8e  c0 8e d8 be 00 7c bf 00  |PK.....|.....|..|
00000010  06 b9 00 02 fc f3 a4 50  68 1c 06 cb fb b9 04 00  |.....Ph.....|
00000020  bd be 07 80 7e 00 00 c5  0b 0f 85 0e 01 83 c5 10  |.....~...|.....|
00000030  e2 f1 cd 18 88 56 00 55  c6 46 11 05 c6 46 10 00  |.....V.U.F....F..|
00000040  b4 41 bb aa 55 cd 13 5d  72 0f 81 fb 55 aa 75 09  |.A..U..]r...U.u.|
00000050  f7 c1 01 00 74 03 fe 46  10 66 60 80 7e 10 00 74  |....t..F.f`..~..t|
00000060  26 66 68 00 00 00 00 66  ff 76 08 68 00 00 68 00  |6fh.....f.v.h..h.|
00000070  7c 68 01 00 68 10 00 b4  42 8a 56 00 8b f4 cd 13  ||h..h...B.V.....|
00000080  9f 83 c4 10 9e eb 14 b8  01 02 bb 00 7c 8a 56 00  |.....|..V.|
00000090  8a 76 01 8a 4e 02 8a 6e  03 cd 13 66 61 73 1c fe  |.v..N..n...fas..|
```

- Fortunately the approach was correct and 4 files were returned.



- However, the password to extract this file was nowhere to be found and could not be cracked successfully even using John The Ripper or by observing the strings of the files. Words like "beautiful", "sincere", "flag" were used as well but none of them were the password to extract the file.



- Therefore the extraction was unable to take place and the flag could not be revealed any further.
- An attempt to check the file headers of the two files named beautiful.jpg and sincere.mp4 was further made and something strange was spotted.

```
(tan2540㉿kali)-[~/Downloads/corruptUnzip]
└─$ hexdump -C beautiful.jpg | head
00000000  89 50 4e 47 0d 0a 0a 00 00 00 0d 49 48 44 52  .PNG.....IHDR
00000010  00 00 03 56 00 00 01 b5 08 02 00 00 00 75 71 f2  ...V.....uq.
00000020  39 00 00 00 01 73 52 47 42 00 ae ce 1c e9 00 00  9....SRGB.....
00000030  00 04 67 41 4d 41 00 00 b1 8f 0b fc 61 05 00 00  ..gAMA.....a ...
00000040  00 09 70 48 59 73 00 00 0e c3 00 00 0e c3 01 c7  ..pHYs.....
00000050  6f a8 64 00 00 ff a5 49 44 41 54 78 5e ec fd 87  o.d....IDATx^ ...
00000060  76 24 4b 9a df 09 9a 74 1d 0a 2a d5 15 55 d5 dd  v$K....t..*..U..
00000070  64 0f b9 bb 67 c9 90 fb 00 9c 57 59 79 96 b3 0f  d...g.....WYy ...
00000080  b1 dc 87 db 73 76 76 76 48 0e d9 dd 55 ec 12 b7  ....svvvH...U...
00000090  ee cd 9b 02 2a 84 6b 53 fb 7d e6 01 64 24 12 22  ....*.ks.}..d$."
```

The original file type was in JPEG however the file header shows the other, which is type PNG. However no differences were spotted while converting the file to a PNG file or modifying the file to a JPEG file.

- This time, John The Ripper was used along with the rockyou.txt wordlist and lavidaesbella seems to be the password.

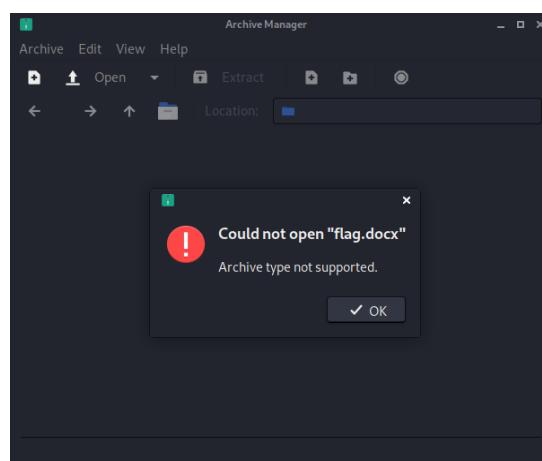
```
(tan2540㉿kali)-[~/Downloads/corruptUnzip]
└─$ zip2john flag.zip > password.txt
ver 2.0 efn 9901 flag.zip/flag.docx PKZIP Encr: cmplen=40831, decmplen=43499, crc=C6B64A22

(tan2540㉿kali)-[~/Downloads/corruptUnzip]
└─$ john --show password.txt
flag.zip/flag.docx:lavidaesbella:flag.docx:flag.zip:flag.zip

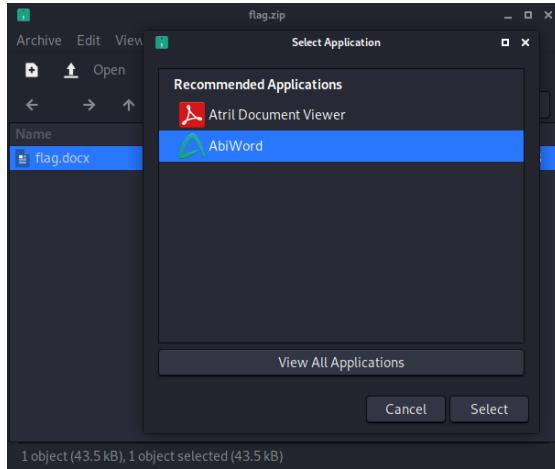
(tan2540㉿kali)-[~/Downloads/corruptUnzip]
└─$ john password.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (ZIP, WinZip [PBKDF2-SHA1 256/256 AVX2 8x])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
lavidaesbella  (flag.zip/flag.docx)
1g 0:00:00:00 DONE (2022-04-13 23:55) 2.380g/s 19504p/s 19504c/s 19504C/s newzealand..whitetiger
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

- However the extraction still did not take place successfully as the system says that they do not support the .docx file type.

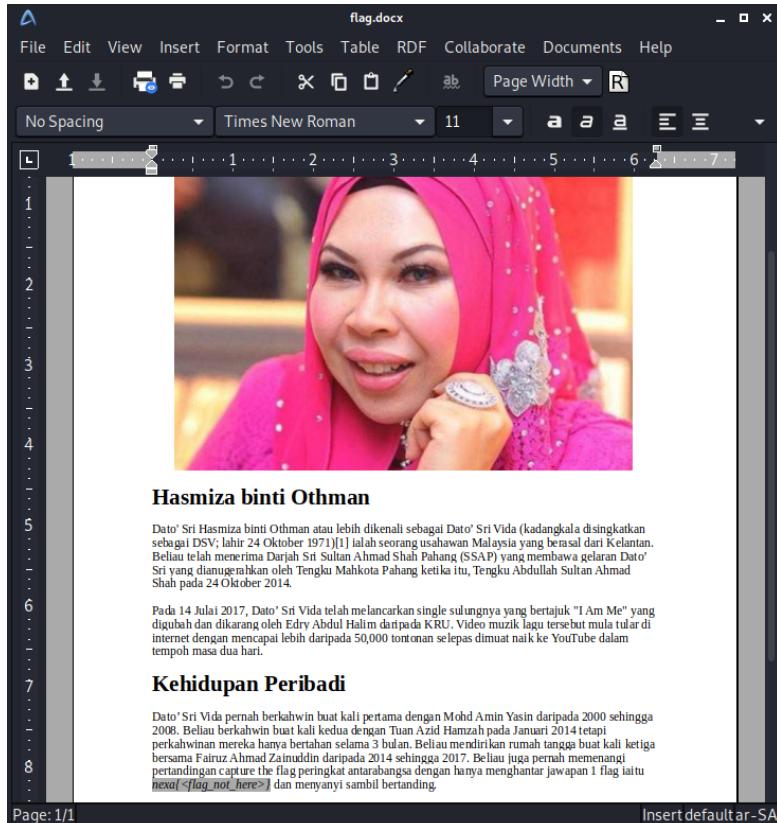
```
(tan2540㉿kali)-[~/Downloads/corruptUnzip]
└─$ unzip -P lavidaesbella /home/tan2540/Downloads/corruptUnzip/flag.zip -d /home/tan2540/Downloads/corruptUnzip
Archive: /home/tan2540/Downloads/corruptUnzip/flag.zip
      skipping: flag.docx                      unsupported compression method 99
```



- After multiple attempts in trying to extract the .docx file (e.g. uploading the .zip file to google drive and download then extracting it in Windows etc.), another approach was made to open the file with a free and open-source word processor called Abiword.



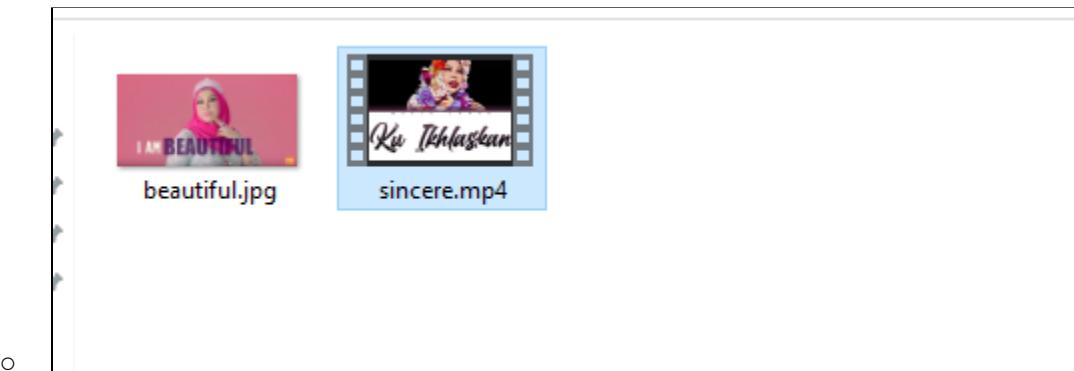
- Finally, the flag.docx file can be viewed and the flag is noticed in the second paragraph.



Flag revealed: ***nexa{<flag_not_here>}***

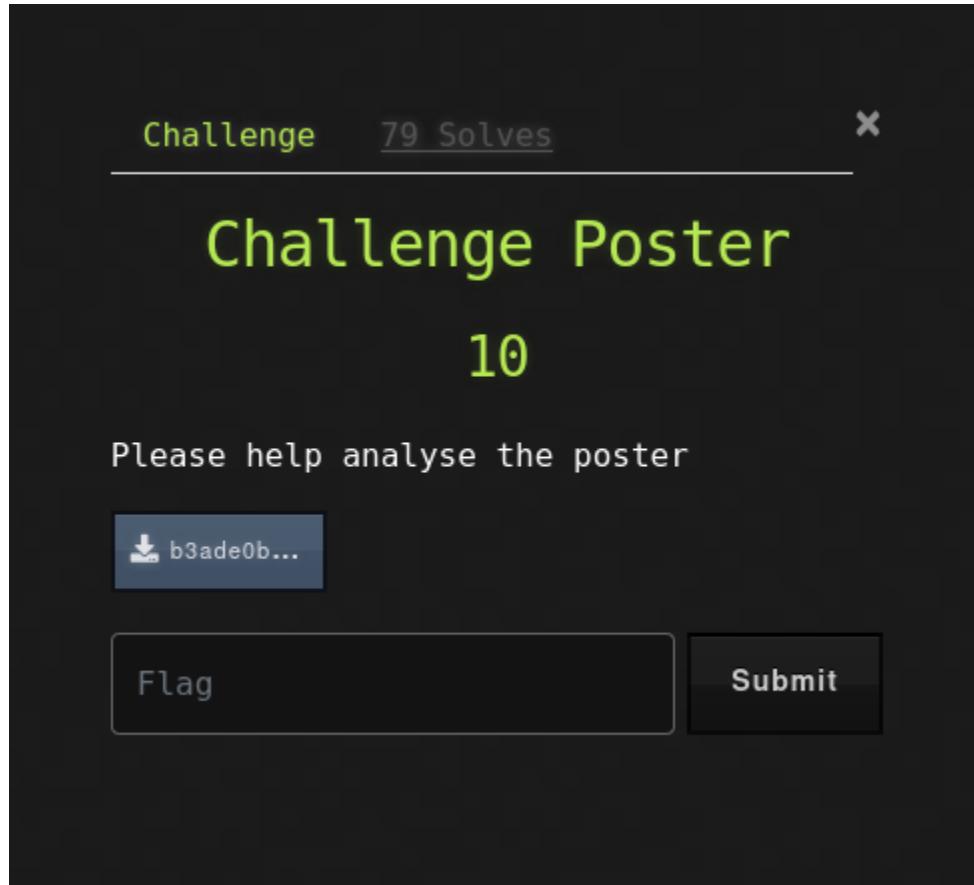
- Unfortunately, the flag was found after the second CTF challenge while taking more time to crack the password therefore we did not manage to get the marks for this question during CTF Challenge 2.

- Cheok Yi Xuan:
 - Change the file extension to .zip
 - Extract the zip file to reveal a .jpg and .mp4 file



Misc

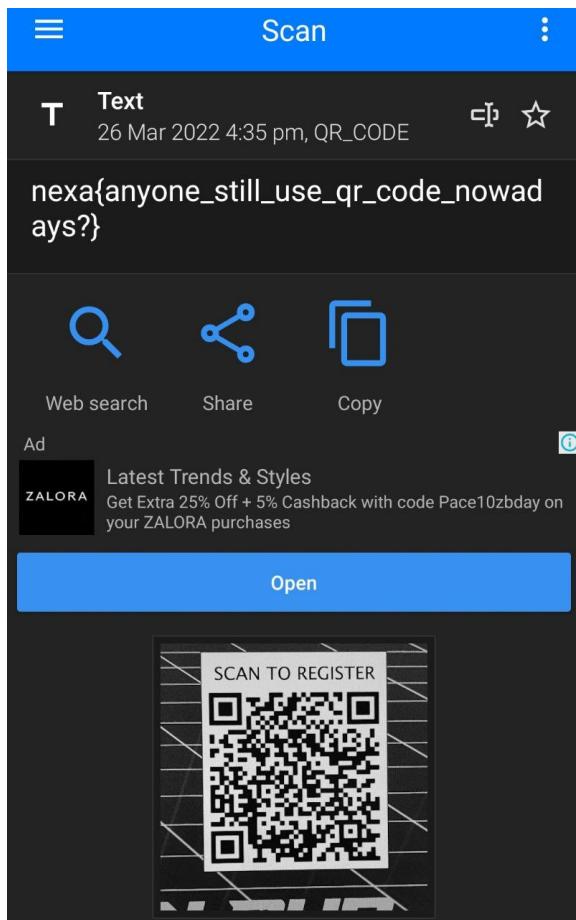
Challenge Poster



Members involved: Afif

Thought process/methodologies:

- Open image file
- Scan the QR code



- Flag: *nexa{anyone_still_use_qr_code_nowadays?}*

Click for Surprise!!!!



Members involved: Cheok Yi Xuan

Thought process/methodologies:

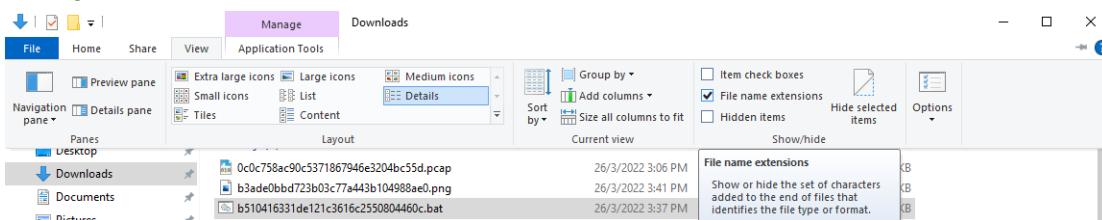
- Run the bat file



- Shuts down the computer



- Change the extension to .txt and open



- Rename

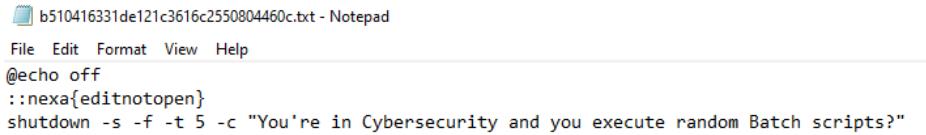


If you change a file name extension, the file might become unusable.

Are you sure you want to change it?

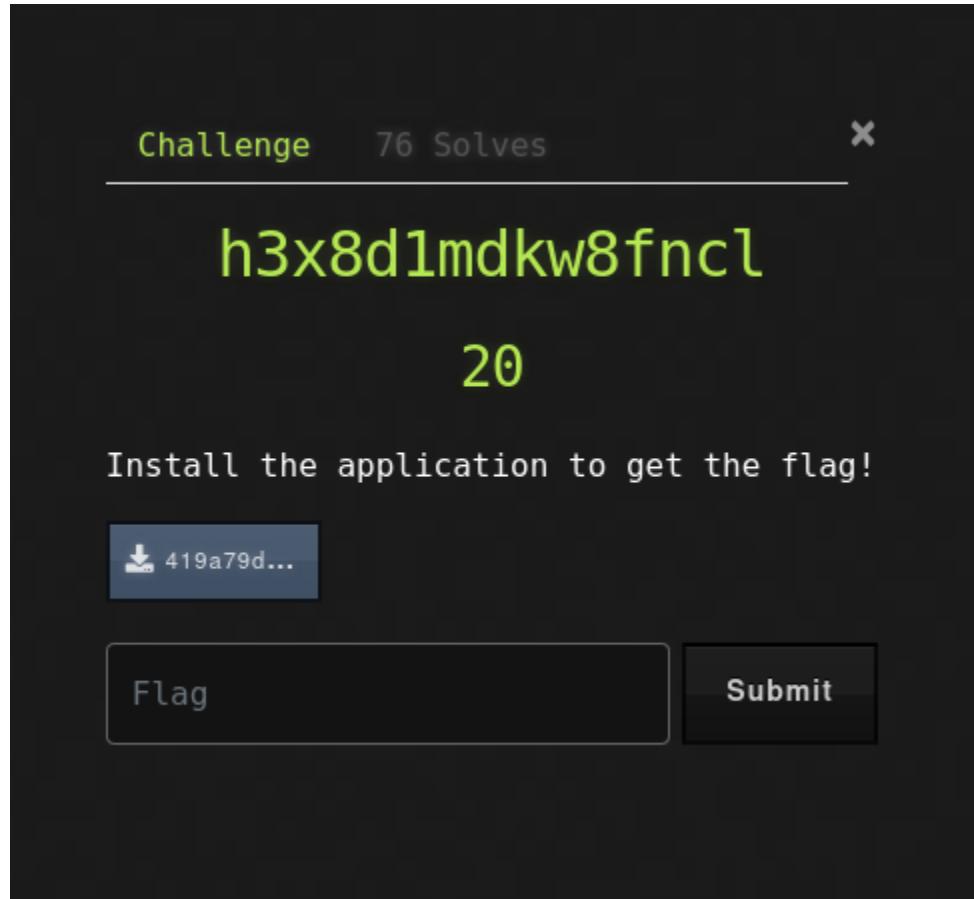


-



- Flag: **nexa{editnotopen}**

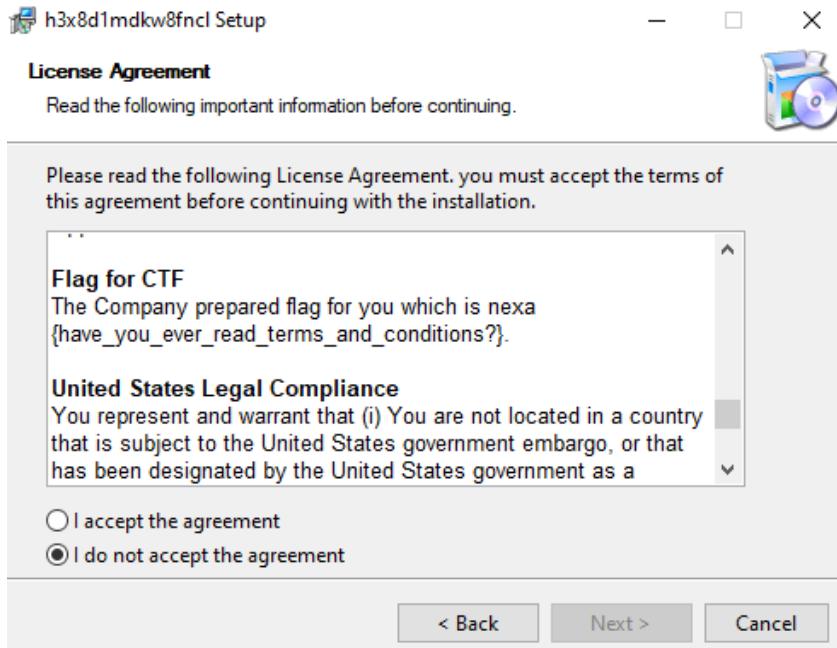
H3x8d1mdkw8fncl



Members involved: Cheok Yi Xuan

Thought process/methodologies:

- Install the application
- Find the link to the youtube video
- Reinstall the application
- Look around the license agreement



-
- Flag: `nexax{have_you_ever_read_terms_and_conditions?}`

Unknown File Type

Challenge 70 Solves X

Unknown file type

30

Can you help me to recover this broken file?

[View Hint](#)

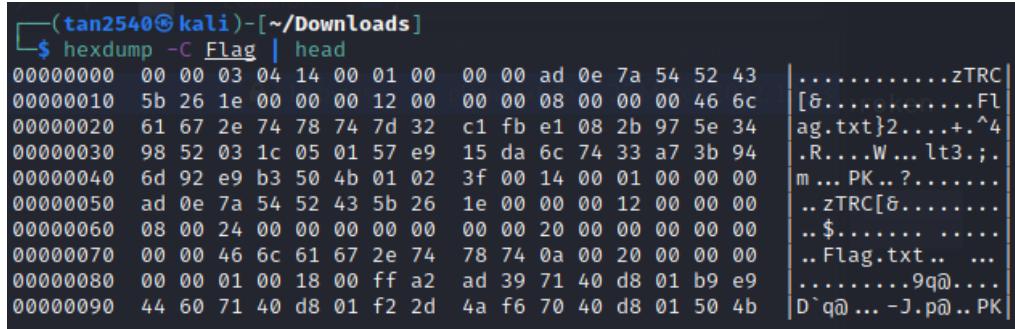
[!\[\]\(8bfac490c4bdf1ca8fe56891643838ca_img.jpg\) Flag](#)

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

Members involved: Afif, Tan Shupei

Thought process/methodologies:

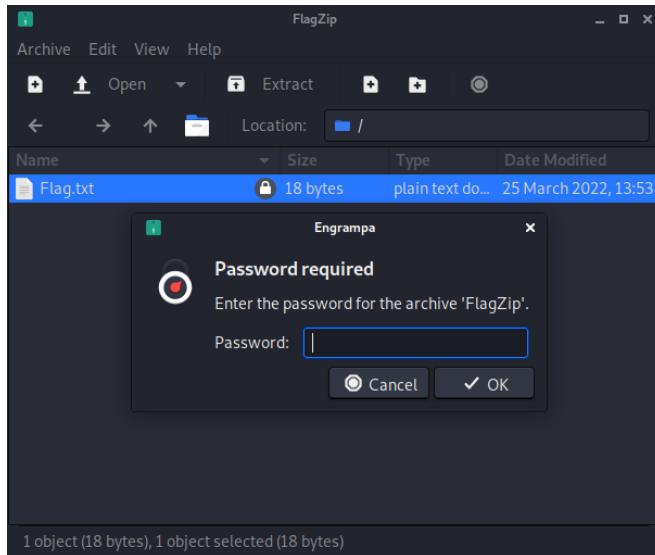
- Afif
 - Identify file extension type by running `file` on the downloaded *Flag* file
 -  \$ file --brief -l --extension Flag
application/zip; charset=binary
 - Add `.zip` extension behind the file name
 - File returns as an archived file; can be opened by an archiving software
 - Archived file contains a text file and seems to be protected by a password
 - Password not identified yet
 - Unable to solve the challenge...
- Tan Shupei
 - Hexdump was used to check the hex content of the file first to identify the file type and to check if there is any corruption at the file header.



```
(tan2540㉿kali)-[~/Downloads]
└─$ hexdump -C Flag | head
00000000  00 00 03 04 14 00 01 00  00 00 ad 0e 7a 54 52 43  .....zTRC
00000010  5b 26 1e 00 00 00 12 00  00 00 08 00 00 00 46 6c  [&.....Fl
00000020  61 67 2e 74 78 74 7d 32  c1 fb e1 08 2b 97 5e 34  ag.txt}2....+.^4
00000030  98 52 03 1c 05 01 57 e9  15 da 6c 74 33 a7 3b 94  .R....W...lt3.;.
00000040  6d 92 e9 b3 50 4b 01 02  3f 00 14 00 01 00 00 00  m...PK..?.....
00000050  ad 0e 7a 54 52 43 5b 26  1e 00 00 00 12 00 00 00  ..zTRC[&.....
00000060  08 00 24 00 00 00 00 00  00 00 20 00 00 00 00 00 00  ..$..... .....
00000070  00 00 46 6c 61 67 2e 74  78 74 0a 00 20 00 00 00 00  ..Flag.txt.. ...
00000080  00 00 01 00 18 00 ff a2  ad 39 71 40 d8 01 b9 e9  .....9q@.....
00000090  44 60 71 40 d8 01 f2 2d  4a f6 70 40 d8 01 50 4b  D'q@... -J.p@..PK
```

Flag.txt was noticed on the 8th line therefore the file type is suspected to be a `.zip` file.

- After using hexeditor to modify the file header, a `.zip` file was returned with a `Flag.txt` file as predicted earlier but password is required to extract the text file.

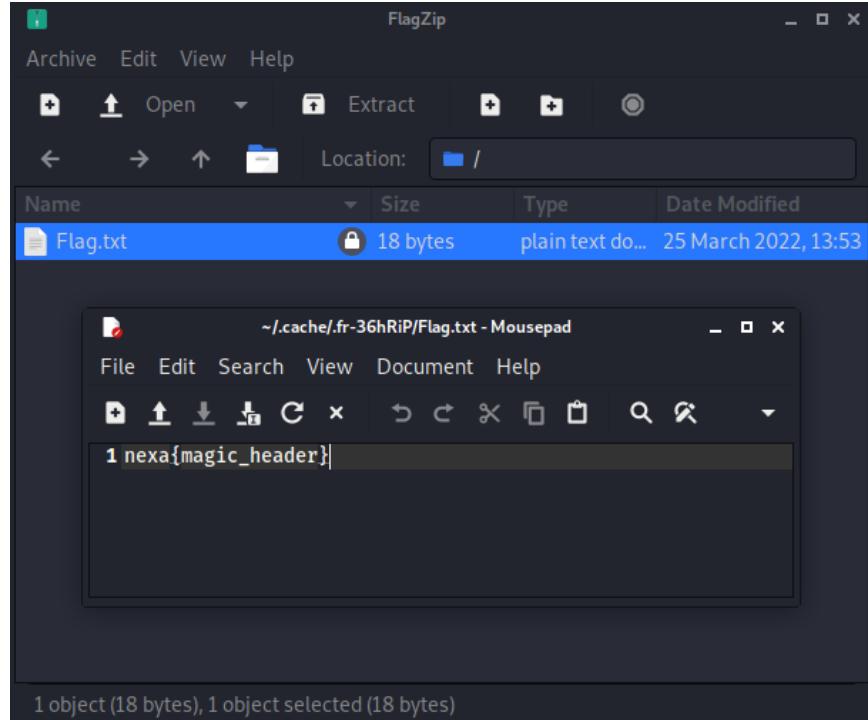


- John The Ripper was used to crack the password of the .zip file and **1234** seems to be the password to extract the .zip file.

```
(tan2540㉿kali)-[~/Downloads]
$ zip2john FlagZip > FlagZip.txt
ver 2.0 FlagZip/Flag.txt PKZIP Encr: cmplen=30, decmplen=18, crc=265B4352
                                          )roken

(tan2540㉿kali)-[~/Downloads]
$ john FlagZip.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 5 candidates buffered for the current salt, minimum 8 needed for performance.
Warning: Only 1 candidate buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 6 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234
(FlagZip/Flag.txt)
1g 0:00:00:00 DONE 2/3 (2022-04-13 11:35) 14.28g/s 379800p/s 379800c/s 379800C/s 123456 .. Peter
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

- **1234** was indeed the password and the .zip file extraction took place successfully.



- Flag revealed: **nexa{magic_header}**
- Unfortunately, same as "Basic 4", the flag was found after the first CTF challenge while going through the questions again and reattempting them therefore we did not manage to get the marks for this question during CTF Challenge 1.

Barcode I

Challenge 27 Solves X

Barcode I

10

You know what to do?

 [barcode_...](#)

Flag Submit

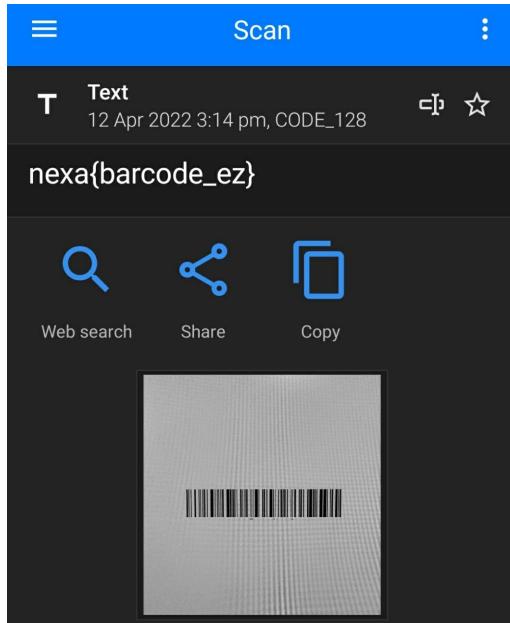
Members involved: Afif, Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- Afif:
 - The file given is an image file with a PNG extension
 - Open the file



-
- Use a phone application to scan the barcode



-
- Flag is shown as: ***nexa{barcode_ez}***

- Khairul Hanie Hazierah Binti Mohd Azmi:
 - The picture in PNG format was open and inspected.

Result

Format:

CODE-128

Content:

nexa{barcode_ez}

- Using a barcode scanner online, the flag is revealed to be ***nexa{barcode_ez}***

Document 1

Challenge 21 Solves X

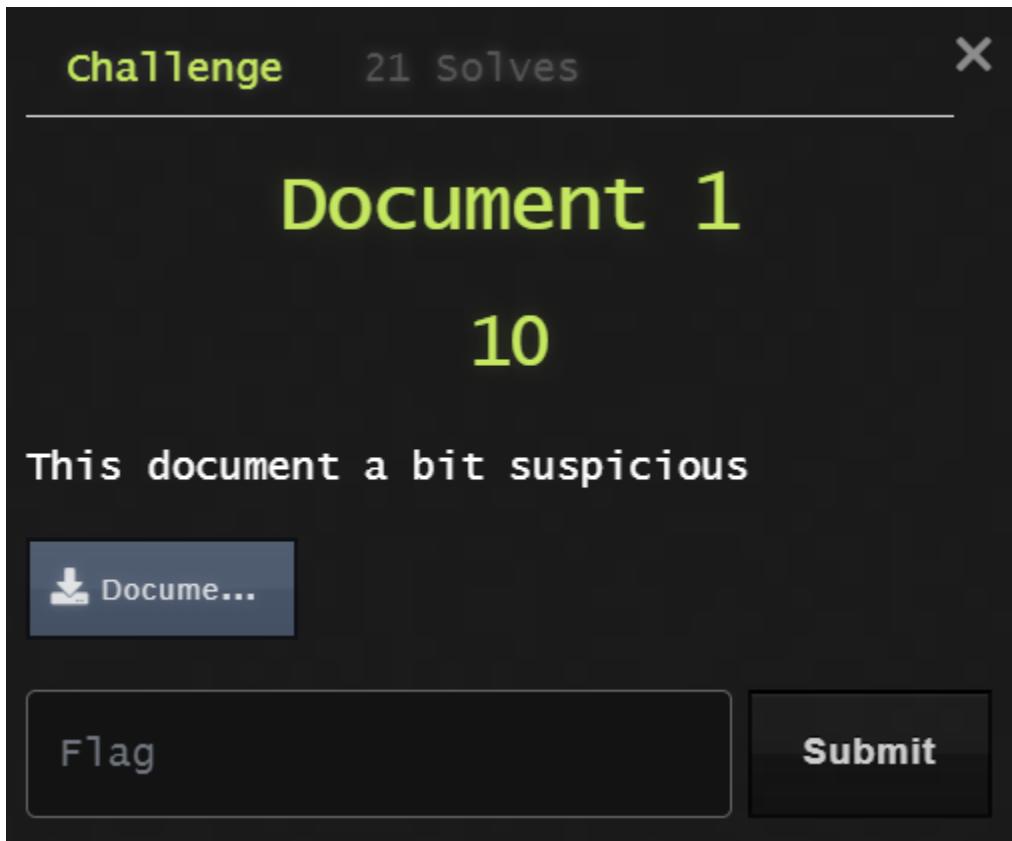
Document 1

10

This document a bit suspicious

 Docume...

Flag Submit



Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- The file given was open and inspected.

cover page, and text box designs that complement each other. Then add a matching cover page, header, and sidebar. Click on the 'Design' tab to see the changes. Finally, click on the 'File' tab to see the document coordinated. `nexa{always_check_all}`

When you click Design and choose a new Theme, the graphics change to match your new theme. When you click on the 'File' tab, the document is coordinated.

- After changing the font color to red, the flag is revealed to be `nexa{always_check_all}`

Document 2

Challenge 3 Solves X

Document 2

10

I got 1337 document. Is there any meaning of this?

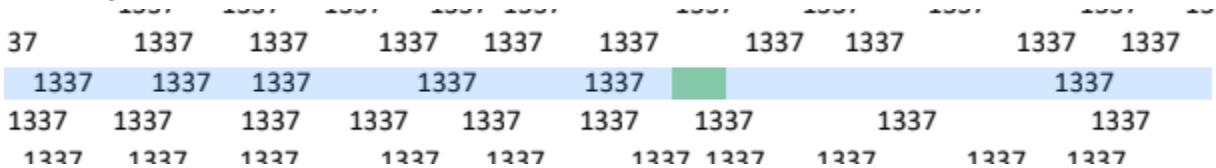
 Random...

Flag Submit

Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

- The file given was open and inspected.



```
37      1337  1337  1337  1337  1337  1337  1337  1337  1337  1337  1337
      1337  1337  1337      1337      1337  1337  1337  1337      1337  1337
1337  1337  1337  1337  1337  1337  1337  1337  1337  1337  1337
1227  1227  1227  1227  1227  1227  1227  1227  1227  1227  1227
```

- Using the control F command, the flag location was known but it cannot be seen.

nt1337from1337the1337different1337galleries.1337nexa{document_can_be_annoying}1337Themes

- The invisible flag then was copied and pasted on google docs and the flag is revealed to be *nexa{document_can_be_annoying}*

Call From Anonymous !!!



Members involved: Afif, Tan Shupei

Thought process/methodologies:

- Afif:

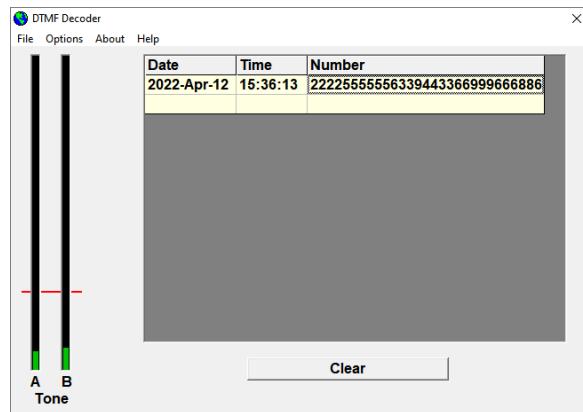
- File given is an MP3 audio file
 - Open file
 - File title gives a clue as to what the demonstration is about

 [DTMF detection demo](#)

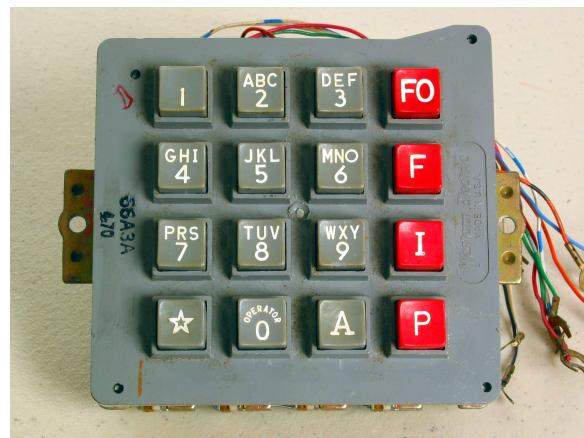
- The demonstration indicates DTMF
 - Download and install a DTMF decoder software. [Link here](#)



- Run software, play audio and begin analyzing the given numbers decoded



- Audio analysis can be inconsistent due to window time of tone being played each time. A good spot for gap time between each tone is from 100ms to 250ms
 - DTMF recorded: **2222555555633944336699966688663333633**
 - Find an image referencing a DTMF keypad



- Translate DTMF to letters

- DTMF translated: **ALMEWHENYOUNEME**
- Unable to decrypt the message...
- Tan Shupei
 - Based on the writeup provided by Afif, a different approach was done to translate the recorded DTMF which is to group 3 same numbers together if it occurs more than 3 times as shown below:
222-2-555-555-6-33-9-44-33-66-999-666-88-66-33-33-3-6-33
 - Therefore, the DTMF is translated as C-A-L-L-M-E-W-H-E-N-Y-O-U-N-E-E-D-M-E
 - Flag revealed: ***nexa{callmewhenyouneedme}***

MD5 Collisions

Challenge 1 Solves X

MD5 Collisions

20

You are required to find the executable file with the MD5 hash of **2a2992c5eff3645f92e66f96fd269c2d** that performs a malicious task. Good Luck!

 2a2992c...

Flag Submit

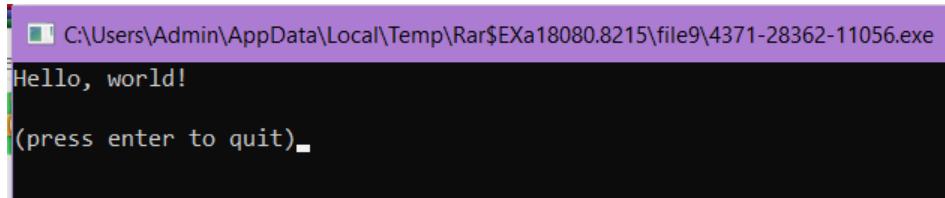


Members involved: Khairul Hanie Hazierah Binti Mohd Azmi

Thought process/methodologies:

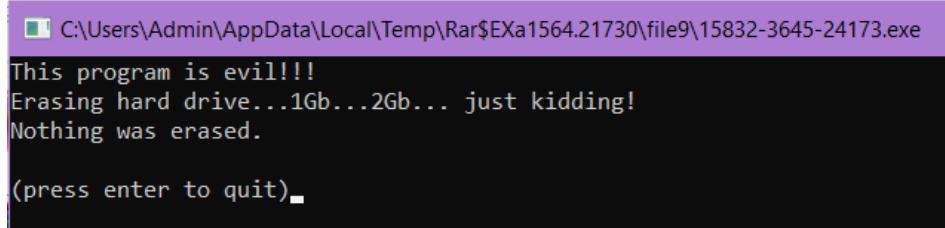
- The file given was open and inspected.

13046-26279-51...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2
13312-21101-19...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2
13824-27822-21...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2
15832-3645-241...	6,144	2,422	Application	30/5/2018 7:16 ...	A89052AE
15849-30384-34...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2
15999-17275-48...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2
17580-12781-24...	6,144	2,400	Application	30/5/2018 7:16 ...	745475B2



```
C:\Users\Admin\AppData\Local\Temp\Rar$EXa18080.8215\file9\4371-28362-11056.exe
Hello, world!
(press enter to quit)
```

- After running all the .exe files, most of the files contain the same data which is “Hello, world!”



```
C:\Users\Admin\AppData\Local\Temp\Rar$EXa1564.21730\file9\15832-3645-24173.exe
This program is evil!!!
Erasing hard drive...1Gb...2Gb... just kidding!
Nothing was erased.

(press enter to quit)
```

- File “15832-3645-24173.exe” contains different data that performs the malicious tasks.
- Therefore, the flag is revealed to be **nexa{15832-3645-24173.exe}**

Too Much?



Challenge 0 Solves X

Too Much?

40

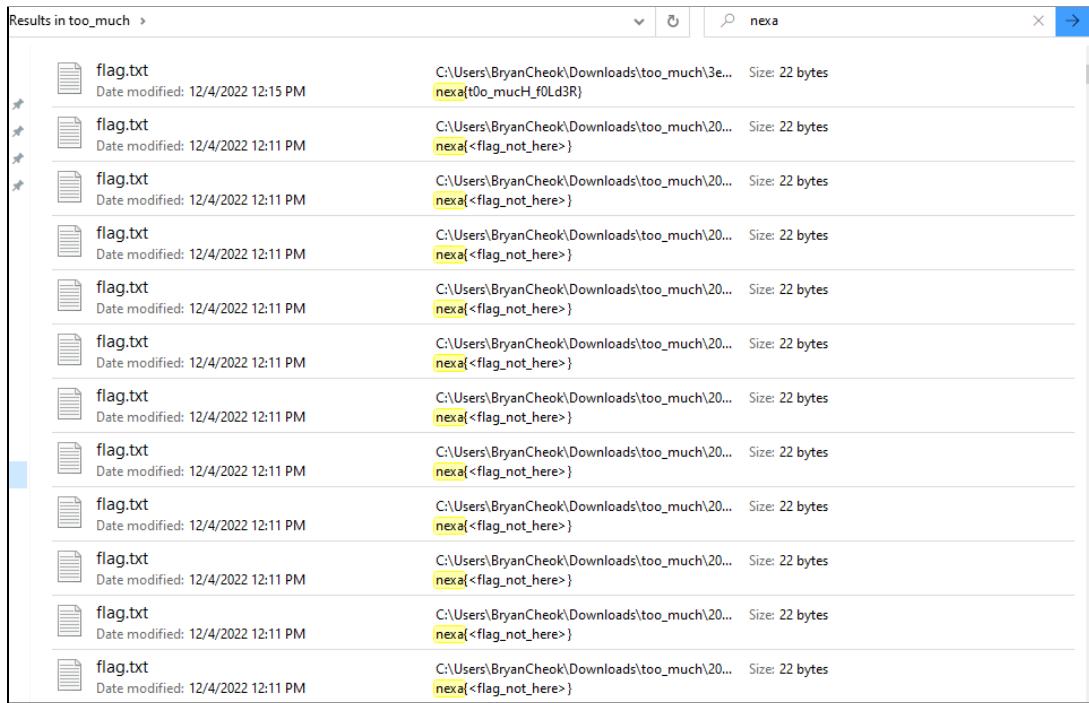
I heard the flag is hiding somewhere.
Help me find it!

Flag Submit

Members involved: Cheok Yi Xuan

Thought process/methodologies:

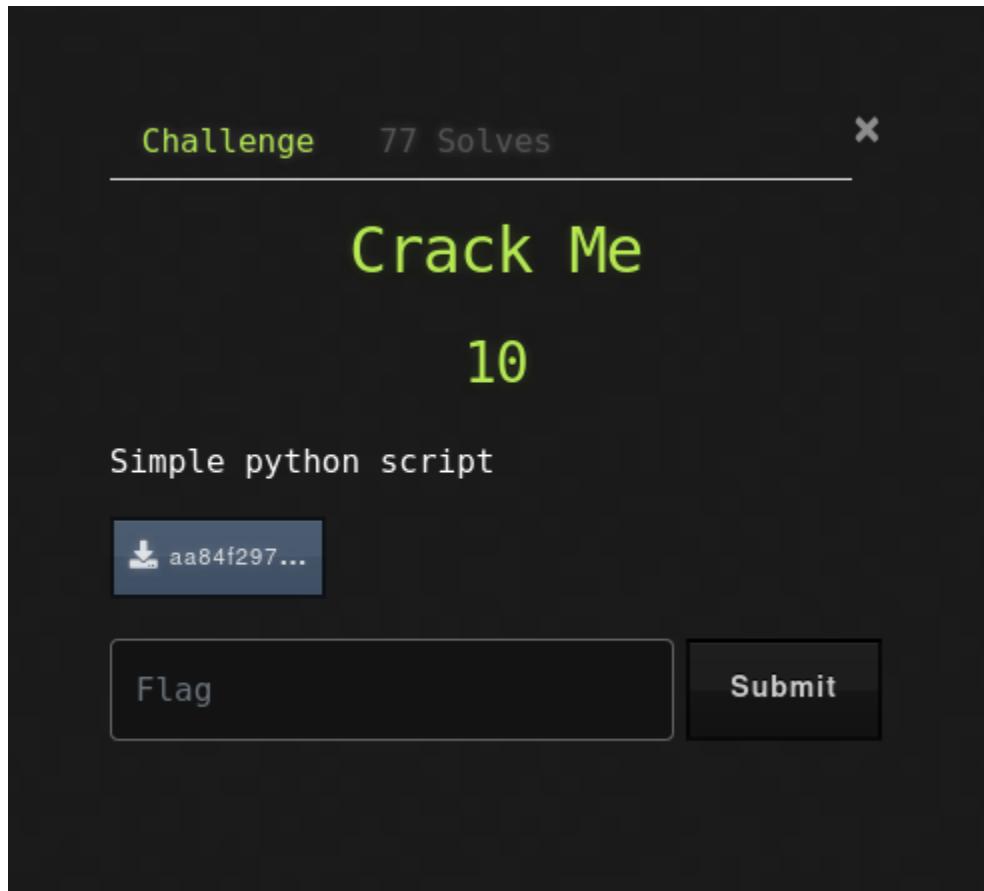
- Search for the word “nexa” in the directory



- Obtain the flag
 - Flag revealed: *nexa{t0o_mucH_f0Ld3R}*

Reverse Engineering

Crack Me



Members involved: Afif

Thought process/methodologies:

- Open Python file with a text editor
- Hint given is a variable which happens to be a secret key is called *bezos_cc_secret*

```
1  # Hiding this really important number in an obscure piece of code is brilliant!
2  # AND it's encrypted!
3  # We want our biggest client to know his information is safe with us.
4  bezos_cc_secret = "E<08R)&@6:8E8@t"
5
6  # Reference alphabet
7  ▼ alphabet = "IV#$&(')*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNPQRSTUVWXYZ"+ \
8      "[\\v]_`abcdefghijklmnopqrstuvwxyz{|}~"
9
10 ▼ def decode_secret(secret):
11     # Encryption key
12     rotate_const = 229
13
14     # Storage for decoded secret
15     decoded = ""
16
17     # decode loop
18     ▼ for c in secret:
19         index = alphabet.find(c)
20         original_index = (index + rotate_const) % len(alphabet)
21         decoded = decoded + alphabet[original_index]
22
23     print(decoded)
```

- Call function *decode_secret(secret)*, and pass the *bezos_cc_secret* variable by value

```
18 ▼ for c in secret:
19     index = alphabet.find(c)
20     original_index = (index + rotate_const) % len(alphabet)
21     decoded = decoded + alphabet[original_index]
22
23     print(decoded)
24
25 ▼ def choose_greatest():
26     """Echo the largest of the two numbers given by the user to the program
27
28     Warning: this function was written quickly and needs proper error handling
29
30
31     user_value_1 = input("What's your first number? ")
32     user_value_2 = input("What's your second number? ")
33     greatest_value = user_value_1 # need a value to return if 1 & 2 are equal
34
35     if user_value_1 > user_value_2:
36         greatest_value = user_value_1
37     elif user_value_1 < user_value_2:
38         greatest_value = user_value_2
39
40     print( "The number with largest positive magnitude is "
41           + str(greatest_value) )
42
43     #choose_greatest()
44
45     decode_secret(bezos_cc_secret)
46
```

- Run the Python file

```
● $ python aa84f29754b12220ea353db9c2867fa9.py
● nexa{ROTi_canai}
● Flag revealed: nexa{ROTi_canai}
```

Lets Play

Challenge 75 Solves X

Lets Play

30

Enough question! Let's play some game!

 MAZE.exe

FlagSubmit

Members involved: Cheok Yi Xuan

Thought process/methodologies:

- Run the MAZE.exe file
- Beat the 2 maze games by searching for the traversal path from the finish line
- Obtain the flag

X

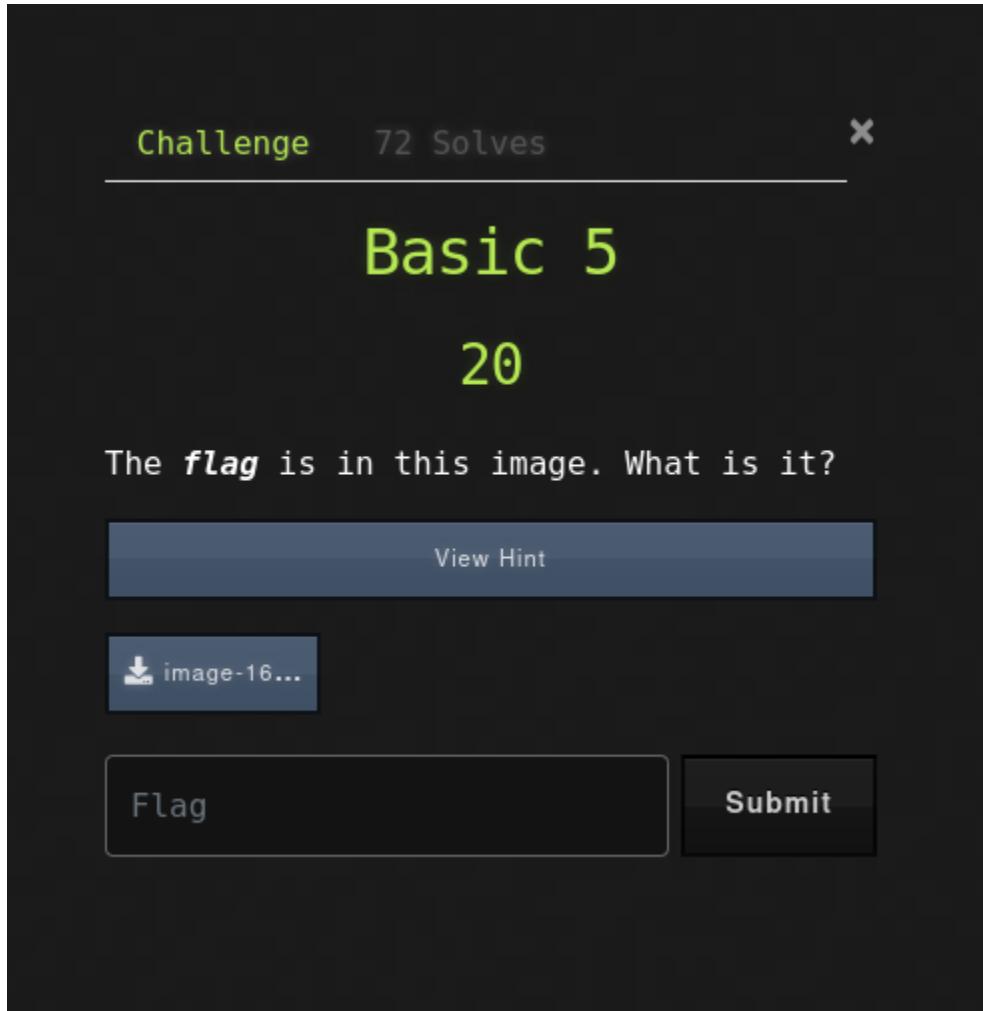
CONGRATULATION Here the flag `nexa{did_you_play_or_reverse_it?}`



Flag: `nexa{did_you_play_or_reverse_it?}`

Steganography

Basic 5



The screenshot shows a challenge interface with the following elements:

- Challenge** 72 Solves
- Basic 5**
- 20**
- The *flag* is in this image. What is it?**
- View Hint** button
- image-16...** download link
- Flag** input field
- Submit** button

Members involved: Tan Shupei

Thought process/methodologies:

- As the file type of this image file is PNG, steghide cannot be used to extract the flag.

```
[tan2540㉿kali)-[~/Downloads]
$ steghide info image-1648235030658.png
steghide: the file format of the file "image-1648235030658.png" is not supported.
```

- An attempt was made using hexdump to check if there is any hidden message at the IEND chunk but nothing strange was spotted.

```
└─$ hexdump -C image-1648235030658.png | tail
00094a20  1b 77 a1 1a 44 34 16 7a  b3 53 84 48 be f6 66 62  |.w..D4.z.S.H..fb|
00094a30  e3 13 aa d0 72 86 bb d1  15 ce 05 04 95 b7 6c 68  |....r.....l..h|
00094a40  8a 5d 9a 5a 9a 2a 7b 36  d3 13 67 53 fd bc 7e ee  |.].Z.*{6..gS..~..|
00094a50  f0 a2 2c a8 64 12 23 df  ad 6d 2e 1c 49 ba 4c 50  |..,..d.#..m..I..LP|
00094a60  0c 20 a0 7e ae a2 59 55  a5 21 f7 5d 7c 88 38 16  |..~..YU!.!.|.8..|
00094a70  01 5e d4 c2 99 6a a0 6b  38 80 2b 74 c2 59 01 e0  |.^...j.k8.+t.Y..|
00094a80  74 ef b9 83 03 90 36 ef  0a 55 66 d3 fe 1f 24 00  |t.....6..Uf ...$.|
00094a90  b5 36 98 57 e4 46 00 00  00 00 49 45 4e 44 ae 42  |.6.W.F....IEND.B|
00094aa0  60 82
00094aa2  |`..|
```

- Another attempt was made using StegOnline which is used to manipulate the color of the image but still the flag or anything strange was spotted from the results.
 - A browser extension named “PassLok Image Steganography” was also used as suggested in the hint given however a password is needed. Words like “flag”, “MMU”, “Basic 5”, “1234” were used but none of them seems to be the correct password.



Intermediate 4

Challenge 71 Solves ×

Intermediate 4

20

some **admin** give me the picture of digimon. but why....

 RookChe...

Flag Submit

Members involved: Tan Shupei

Thought process/methodologies:

- Since the file type is JPEG instead of PNG, steghide can be used.

```
(tan2540㉿kali)-[~]
$ steghide info /home/tan2540/Downloads/RookChessmon.jpeg
"RookChessmon.jpeg":
  format: jpeg
  capacity: 50.6 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "flag.txt":
    size: 19.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

admin was used as a passphrase as it is bolded in the question.

- Steghide was further used to reveal the flag extracted from the RookChessmon.jpeg file.

```
(tan2540㉿kali)-[~]
$ steghide extract -sf /home/tan2540/Downloads/RookChessmon.jpeg
Enter passphrase:
wrote extracted data to "flag.txt".
```

```
(tan2540㉿kali)-[~]
$ cat flag.txt
arkn{fgrtb_vf_sha}
```

- ROT13 is used to reveal the flag extracted.

Text (ASCII / ANSI)

```
arkn{fgrtb_vf_sha}
```

Convert

Highlight Text

ROT13

```
nex{stego_is_fun}
```

Convert

Highlight Text

- Flag revealed: ***nex{stego_is_fun}***

Who's That Pokémon?

Challenge 0 solves X

Who's That Pokémon?

10

The **flag** is hiding somewhere around the audio...Good Luck & Have Fun!

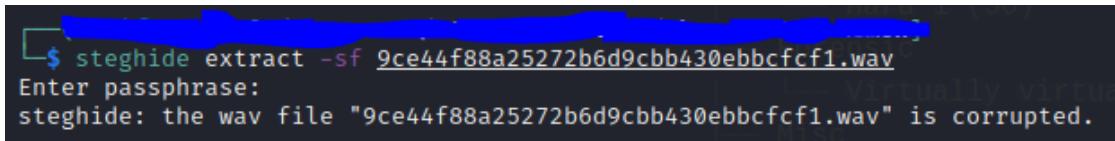
 9ce44f88...

FlagSubmit

Members involved: Tan Shupei, Afif

Thought process/methodologies:

- Afif:

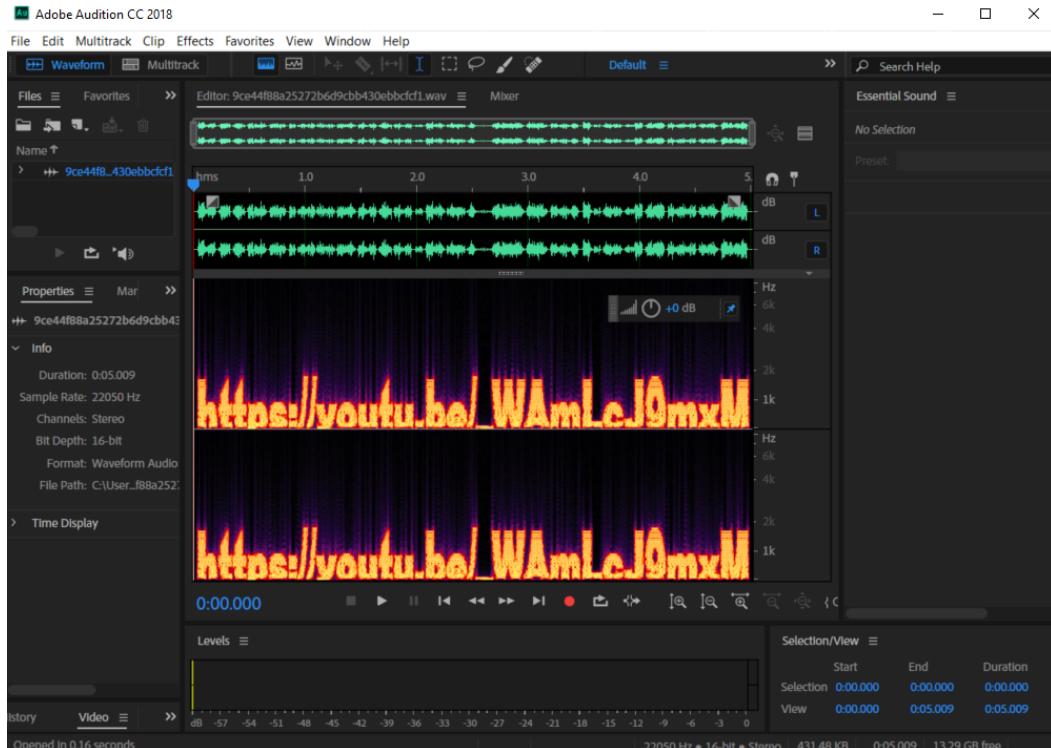


```
$ steghide extract -sf 9ce44f88a25272b6d9cbb430ebbcfcf1.wav
Enter passphrase:
steghide: the wav file "9ce44f88a25272b6d9cbb430ebbcfcf1.wav" is corrupted.
```

- Audio file is given
- Try to crack the stego file with *steghide*
- Requires passphrase, and it has not been found yet
- Unable to decode the audio file...

- Tan Shupei

- The same approach as Afif was done using the passphrase “flag” but it did not decode the audio file.
- Therefore another approach was done which is to use Adobe Audition to view the audio spectrum and a YouTube link was shown in the spectrogram.



- https://www.youtube.com/watch?v=_WAmLcJ9mxM
The flag “bmV4YXtoZWxsb190aGVyZV95b3VfZm91bmRfdGhIX3Bpa2FjaHV9”

was then found in the video description box but it is not the exact flag yet.

Pikachu MMU CTF Challenge 2022 !!!

Unlisted

112 views • Mar 27, 2022

4 DISLIKE

Oscar Lim
5 subscribers

Congratulations you have done your first step !!!
Here is your flag : bmV4YXtoZWxsb190aGVyZV95b3VfZm91bmRfdGhIX3Bpa2FjaHV9

- Therefore, a ASCII converter tool is used to convert the flag from Base64 to ASCII as shown below.

BASE64	Text (ASCII / ANSI)
bmV4YXtoZWxsb190aGVyZV95b3VfZm91bmRfdGhIX3Bpa2FjaHV9	nex{hello_there_you_found_the_pikachu}

Convert **Highlight Text**

Convert **Highlight Text**

- Flag revealed: ***nex{hello_there_you_found_the_pikachu}***

Web

Say the MAGIC WORD!



Members involved: Tan Shupei, Afif, Khairul Hanie Hazierah Binti Mohd Azmi, Cheok Yi Xuan

Thought process/methodologies:

- Tan Shupei
 - The inspect of the page was checked but nothing strange was spotted.
 - Since the question is asking for magic word, words like “please”, “thank you”, “abracadabra” were used as an input but none revealed the flag.
 - An attempt to view flagplease.php was also made however it is not doable as PHP is a server-side programming language therefore flagplease.php is not viewable at all.
 - An attempt was also made to change the form method to GET but the flag still could not be revealed successfully.
- Afif:
 - Click on link given by the question
 - Link leads to a page requesting user input as *Request Flag*

THE FLAG ORGANIZATION @NEXAGATE

Want the flag? Say the magic word

Magic Word:

- — Flag Organization Team 2022 [Nexagate x MMU 2022]
 - By viewing the page source of the HTML file, the variable assignment *magic* might give a clue
- ```
<!DOCTYPE html>
<html>
 <head>...</head>
 <body>
 <article>
 <h1>THE FLAG ORGANIZATION @NEXAGATE</h1>
 <div>
 <p>Want the flag? Say the magic word</p>
 <form action="/flagplease.php" target="_blank" method="POST">
 " Magic Word:"

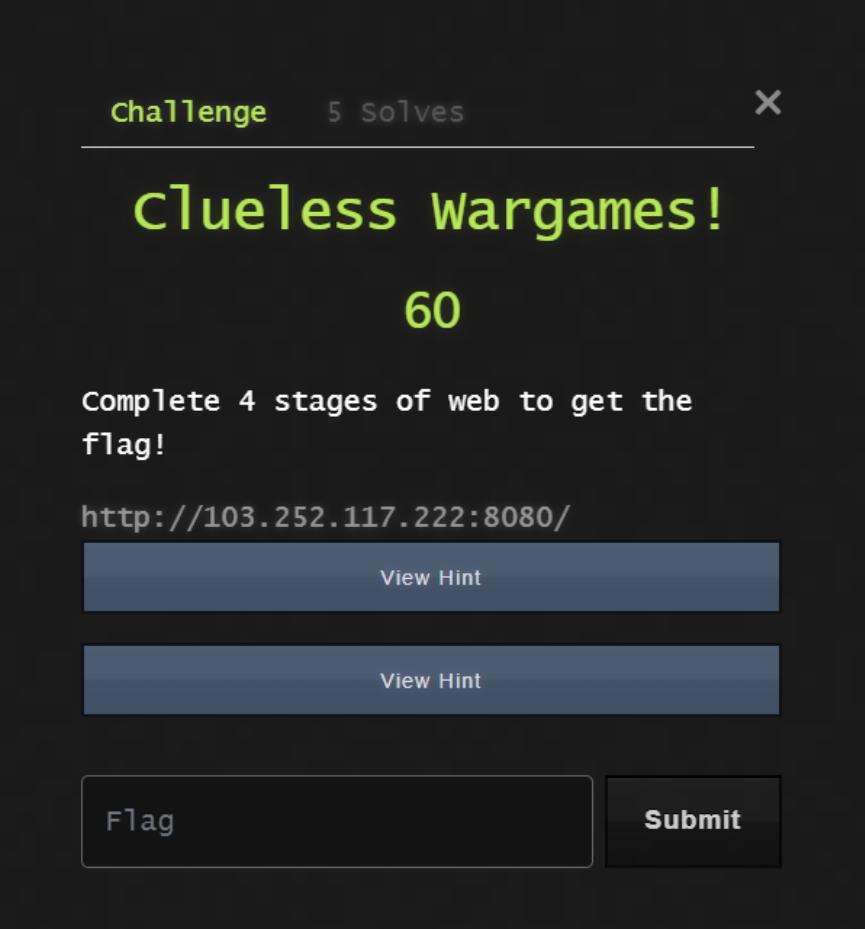
 <input type="text" name="magic" value="Tolong?"/> == $0
 <input type="submit" value="Request Flag">
 </form>
 <p>— Flag Organization Team 2022 [Nexagate x MMU 2022]</p>
 </div>
 </article>
 </body>
</html>
```
- No obvious hardcoded flag found anywhere
  - Unable to proceed from here...

- Khairul Hanie Hazierah Binti Mohd Azmi:

- The link given was open and the elements and the sources were analyzed.
- Based on the hint given, “wstg-inpv-03” indicates a HTTP verb tampering/parameter tampering.

- A lot of trial and error occurred but the flag is nowhere to be found.
- Cheok Yi Xuan
  - Open the link provided
  - Tried multiple inputs such as “magic”, “help”, “please”, etc
  - Changed the form method using inspect element and attempt to submit
  - Tried using command console “curl” command with multiple different method calls

## Clueless Wargames!



Challenge 5 Solves X

# Clueless Wargames!

60

Complete 4 stages of web to get the flag!

<http://103.252.117.222:8080/>

View Hint

View Hint

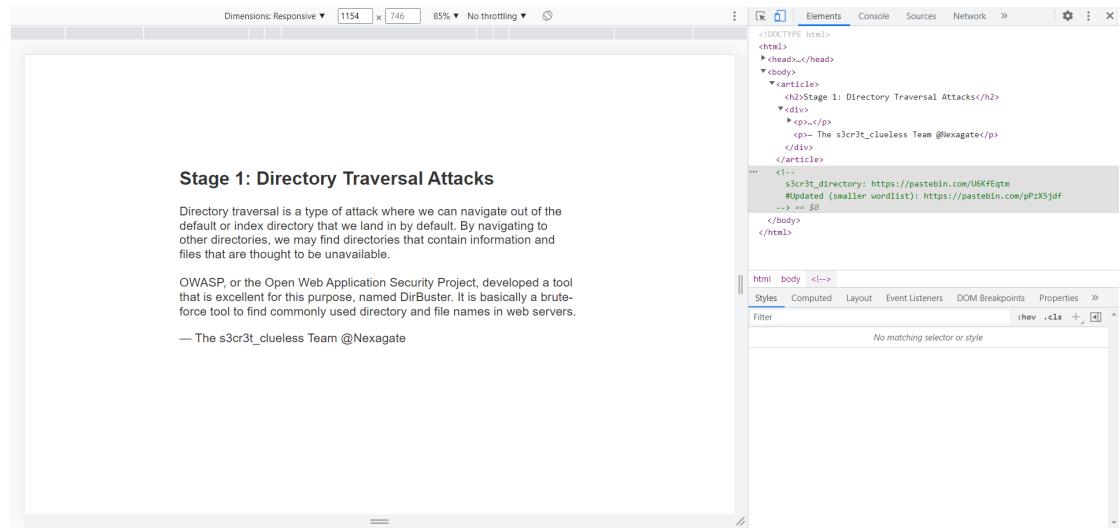
Flag

Submit

**Members involved:** Tan Shupei, Afif

## Thought process/methodologies:

- Tan Shupei
  - The inspect of the page was checked and something was spotted in the code of lines that were commented.



**s3cr3t\_directory: <https://pastebin.com/U6KfEqtm>**  
**#Updated (smaller wordlist): <https://pastebin.com/pPzX5jdf>**

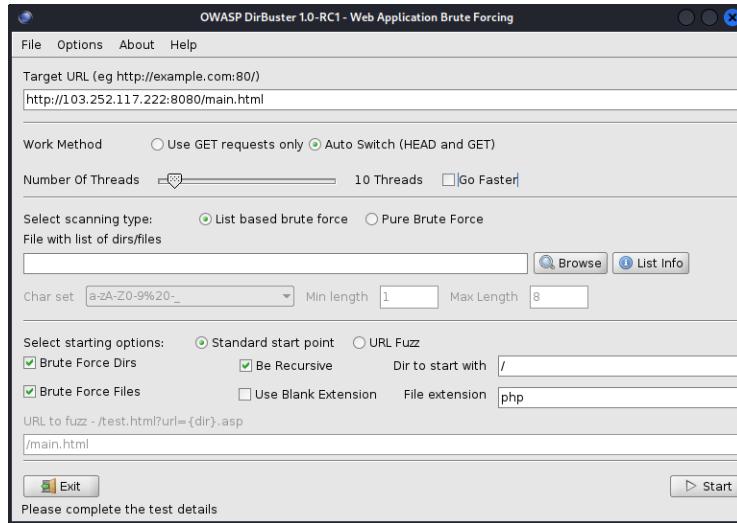
- Afif:

## Stage 1: Directory Traversal Attacks

Directory traversal is a type of attack where we can navigate out of the default or index directory that we land in by default. By navigating to other directories, we may find directories that contain information and files that are thought to be unavailable.

OWASP, or the Open Web Application Security Project, developed a tool that is excellent for this purpose, named DirBuster. It is basically a brute-force tool to find commonly used directory and file names in web servers.

— The s3cr3t\_clueless Team @Nexagate



- Question mentions about a software named *DirBuster* that checks for common directory exploits in websites
- Run *DirBuster*, paste in the page's URL, and start the test
- Unable to continue from here due to many program errors in the brute forcing process...