

**TEH3261/ FTEH3261**  
**ETHICAL HACKING AND SECURITY ASSESSMENT**  
**CTF Write Up**

**Lab Session : 1BV**

**Team Name : 3AM Hackers**

**Team Leader : 1191302861 Mayar Abdulmalik M Shenawi**

**Group Member 1 : 1181102166 Ahmed Aldughhaither**

**Group Member 2 : 1191302763 Al Ghamdi Omar Saeed O**

**Group Member 3 : 1191201179 Rian Tiew Ming Sheen**

**Group Member 4 : 1171103833 Loo Wei Jun**

```
|--- Cracking
|   |--- Basic Wifi Cracking (50) ✓
|--- Cryptography
|   |--- Basic 1 (10) ✓
|   |--- Basic 2 (10) ✓
|   |--- Intermediate 1 (20) ✓
|   |--- Intermediate 2 (20) ✓
|   |--- Intermediate 3 (20) ✓
|--- Forensic
|   |--- Basic 3 (10) ✓
|   |--- Basic 4 (10) ✓
|   |--- Normal PCAP: Part 1 (20) ✓
|   |--- Normal PCAP: Part 2 (20) ✓
|   |--- Shift your focus (20) ✓
|   |--- Normal PCAP: Part 3 (30) ✓
|   |--- Normal PCAP: Part 4 (30) ✓
|--- Misc
|   |--- Challenge Poster (10) ✓
|   |--- Click for Surprise!!!! (20) ✓
|   |--- h3x8d1mdkw8fncl (20) ✓
|   |--- Unknown file type (30) ✓
|--- Reverse Engineering
|   |--- Crack Me (10) ✓
|--- Steganography
|   |--- Basic 5 (20) ✓
|   |--- Intermediate 4 (20) ✓
```

```
|--- Buffer Overflows
|   |--- Stuff In Security !!! (30) ✓
|--- Cryptography
|   |--- Hard 1 (50) ✓
|   |--- Hard 2 (50)
|--- For Fun
|   |--- 3ware (0)
|   |--- Basic HTML (0)
|   |--- MasterMind (0)
|--- Forensic
|   |--- Virtually virtual (50)
|--- Misc
|   |--- Barcode I (10) ✓
|   |--- Document 1 (10) ✓
|   |--- Document 2 (10) ✓
|   |--- Call From Anonymous !!! (20)
|   |--- MD5 Collisions (20) ✓
|   |--- Too Much? (40) ✓
|--- Reverse Engineering
|   |--- Lets Play (30) ✓
|--- Steganography
|   |--- Who's That Pok  mon? (10) ✓
|--- Web
|   |--- Say the MAGIC WORD! (10)
|   |--- Clueless Wargames! (60)
```

## Questions

## Wifi crack

Select your cap file - accepts .cap, .pcap, .pcapng, .gz

Browse ...

## Flag nexa{betoerresmivilida12}

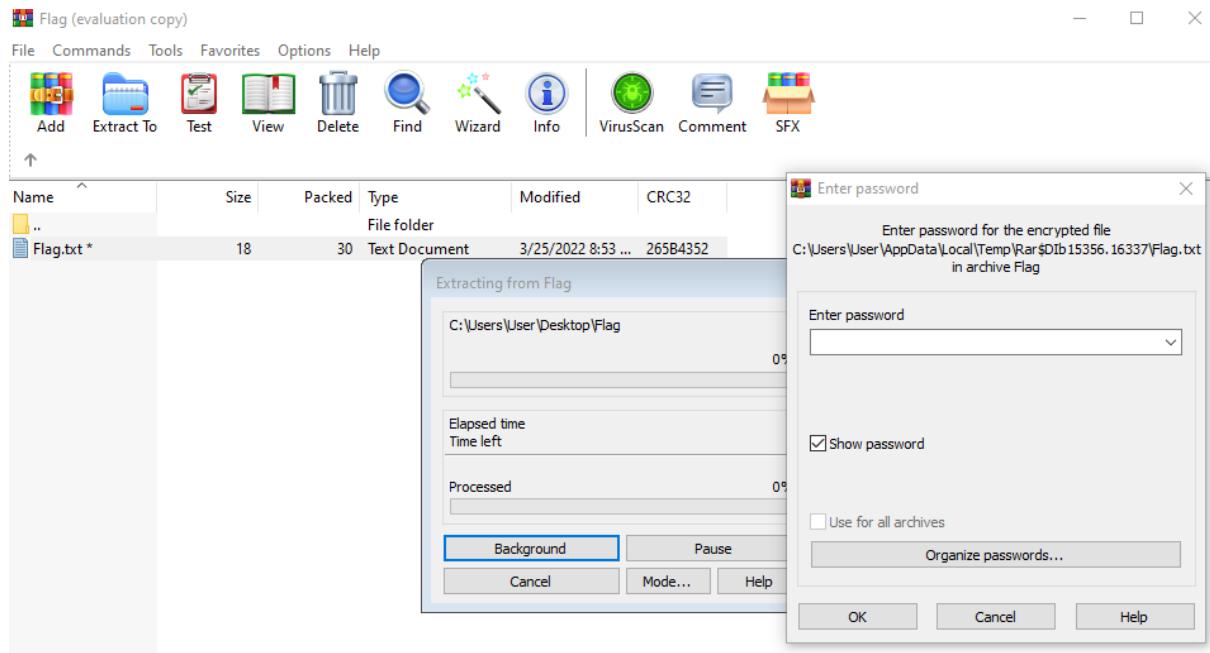
## Unknown file type

Flag	Flag (1)		
Offset	0 1 2 3 4 5 6 7 8 9 A B C D E F	ANSI	ASCII
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	[&	F1
00000020	61 67 2E 74 78 74 7D 32 C1 FB E1 08 2B 97 5E 34	ag.txt}2Áúá +-^4	
00000030	98 52 03 1C 05 01 57 E9 15 DA 6C 74 33 A7 3B 94	R Wé Últ\$;"	
00000040	6D 92 E9 B3 50 4B 01 02 3F 00 14 00 01 00 00 00	m' é*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	\$	
00000070	00 00 46 6C 61 67 2E 74 78 74 0A 00 20 00 00 00	Flag.txt	
00000080	00 00 01 00 18 00 FF A2 AD 39 71 40 D8 01 B9 E9	ÿ¢-9q@Ø `é	
00000090	44 60 71 40 D8 01 F2 2D 4A F6 70 40 D8 01 50 4B	D`q@Ø ð-Jöp@Ø PK	
000000A0	05 06 00 00 00 00 01 00 01 00 5A 00 00 00 44 00	Z D	
000000B0	00 00 00 00		

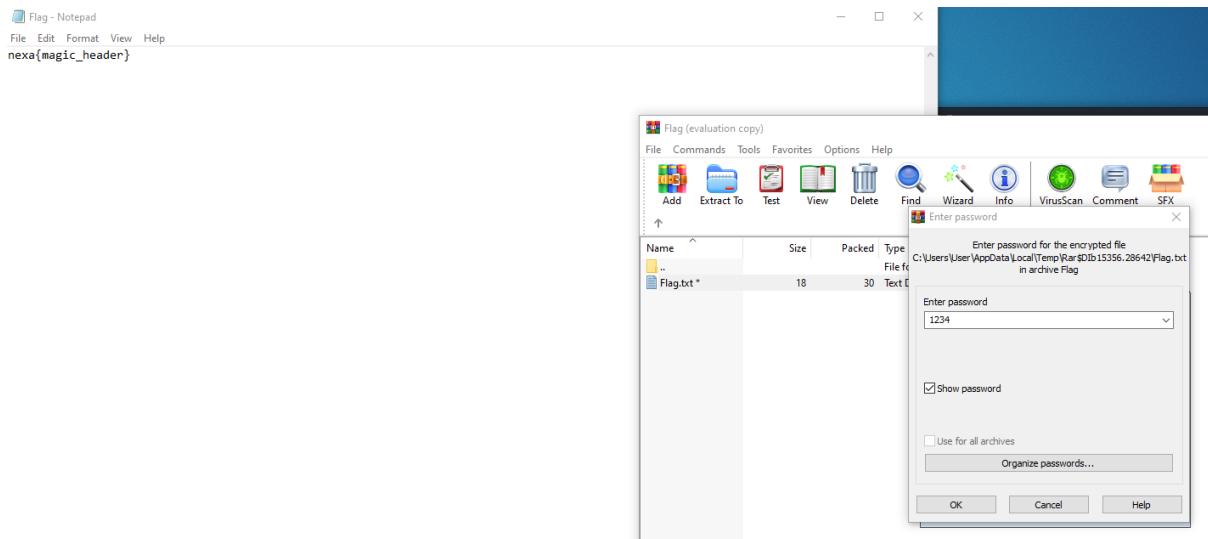
Open the flag with winhex

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	50	4B	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	[& F1
00000020	61	67	2E	74	78	74	7D	32	C1	FB	E1	08	2B	97	5E	34	ag.txt}2Áúá +—^4
00000030	98	52	03	1C	05	01	57	E9	15	DA	6C	74	33	A7	3B	94	R Wé Últ3\$;"
00000040	6D	92	E9	B3	50	4B	01	02	3F	00	14	00	01	00	00	00	m'éPK ?
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	\$
00000070	00	00	46	6C	61	67	2E	74	78	74	0A	00	20	00	00	00	Flag.txt
00000080	00	00	01	00	18	00	FF	A2	AD	39	71	40	D8	01	B9	E9	ÿc-9q@0 ^é
00000090	44	60	71	40	D8	01	F2	2D	4A	F6	70	40	D8	01	50	4B	D`q@0 ò-Jöp@0 PK
000000A0	05	06	00	00	00	00	01	00	01	00	5A	00	00	00	44	00	Z D
000000B0	00	00	00	00													

Change the first few hex to change to a zip file



Open the file with winrar and it will request a password



Using jack the ripper we found the password to be 1234  
And the flag is **nexax{magic\_header}**

## Basic 1

Do you have to deal with **Base64** format? Then this site is perfect for you! Use our super handy online tool to encode or **decode** your data.

**Decode from Base64 format**

Simply enter your data then push the decode button.

VGhlIGZsYVcgXMgbmV4YXtCYXNpY18xfQ

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

Source character set.

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

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

**< DECODE >** Decodes your data into the area below.

The flag is **nexax{Basic\_1}**

Basic base64 converter showed the answer is **nexax{Basic\_1}**

## Basic 2

## Translate morse code

Mode:  Morse code to text  Text to morse code

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

**Copy** **Paste**

### Translation

```
YOU FIND THE MESSAGE. HERE'S THE FLAG MORSE_CODE_IS_NICE
```

**Copy**

Morse code converter showed flag is `nexa{MORSE_CODE_IS_NICE}`

**Intermediate 1**

To decode the email message, we used spammimic.  
The encoded message is **nexax{intermediate\_2}**

be Classroom CAMSYS Netflix MMLS2 CS LEC TSI LEC EH LEC SIRS EH LAB

**spam mimic**

Own the Learn h

Encode

Decode

Explanation

Credits

FAQ & Feedback

Terms

Français

Your spam message **Dear Decision maker ; Especially for you...** decodes to:  
nexa{intermediate\_2} [Encode](#)

Look wrong?, try the [old version](#)

Copyright © 2000-2020 spammimic.com, All rights reserved

## Forensic PCAP part 1

```
+ 24 5.157615 192.168.137.131 192.168.137.130 HTTP 411 GET / HTTP/1.1
+ 26 5.158354 192.168.137.130 192.168.137.131 HTTP 622 HTTP/1.1 200 OK (text/html)
+ 49 5.289869 192.168.137.131 192.168.137.130 HTTP 366 GET /favicon.ico HTTP/1.1
+ 51 5.290491 192.168.137.130 192.168.137.131 HTTP 559 HTTP/1.1 404 Not Found (text/html)
+ 657 16.839946 192.168.137.130 connectivity-check.. HTTP 141 GET / HTTP/1.1
1290 17.010377 192.168.137.130 connectivity-check.. HTTP 260 HTTP/1.1 204 No Content
3667 85.818659 192.168.137.131 192.168.137.130 HTTP 423 GET /download.png HTTP/1.1
3981 85.218856 192.168.137.131 192.168.137.131 HTTP 38080 HTTP/1.1 200 OK (PHOTO)
3939 138.478281 192.168.137.131 connectivity-check.. HTTP 141 GET / HTTP/1.1
3954 138.682641 connectivity-check.. 192.168.137.131 HTTP 282 HTTP/1.1 204 No Content
129 11.689055 rr1.cn-uh-38alr.goo.. 192.168.137.131 QUIC 1389 Handshake, DCID=fe9726, SCID=17162c8656777554
131 11.689055 rr1.cn-uh-38alr.goo.. 192.168.137.131 QUIC 1389 Handshake, DCID=fe9726, SCID=17162c8656777554
[Time since request: 0.000739000 seconds]
[Request in frame: 24]
[Response in frame: 49]
[Text response in frame: 51]
[Request URI: http://192.168.137.130/]
Content-Encoded entity body (gzip): 219 bytes -> 311 bytes
File Data: 311 bytes
Line Data: text data: text/html (9 lines)
<!DOCTYPE html>
<html>
<head>
<title>This is the title of the webpage</title>
</head>
<body>
<p>This is an example paragraph. Anything in the <strong>body</strong> tag will appear on the page, just like this <strong>nexxa{simp3_v38_with_10v3}</strong> tag and its contents.</p>
</body>
</html>
```

Frame (622 bytes) | Uncompressed entity body (311 bytes)

Text item (text), 189 bytes

|| Packets: 4664 - Displayed: 4664 (100.0%) || Profile

The code was hidden in HTTP line-based data and the flag was **nexxa{simp3\_v38\_with\_10v3}**

## Intermediate 3

### Base64 Decode

Base64 online decode function

YXJrbntVbmVxX2xyZ19wbmFfb3JfZmJ5aXJxfQ==

Decode  Auto Update

arkn{Uneq\_lrg\_pna\_or\_fbyirq}

Search for a tool

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

★ BROWSE THE FULL DCODE TOOLS' LIST

**Results**

nex{Hard\_yet\_can\_be\_solved}

anywhere with  
Acrobat Pro DC.

Try free

Adobe

ROT-13 Cipher - [dCode](#)  
Tag(s) : Substitution Cipher

**ROT13 DECODER**

★ ROT13 CIPHERTEXT  
arkn{Uneq\_lrg\_pna\_or\_fbyirq}

★ APPLY ROT-5 ON NUMBERS

► DECRYPT ROT13

See also: ROT Cipher – Caesar Cipher – ROT-47 Cipher

**ROT13 ENCODER**

★ ROT13 PLAIN TEXT  
dCode Rot-13

★ APPLY ROT-5 ON NUMBERS

Sir

Support

To get the flag you need decode with base64 then decode it again with rot-13 cipher to get the flag which was **nex{Hard\_yet\_can\_be\_solved}**

## Forensic Normal PCAP Part 2

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Open Recent

Merge...

Import from Hex Dump...

Close

Save

Save As...

File Set

Export Specified Packets...

Export Packet Dissections...

Export Packet Bytes...

Export PDUs to File...

Export TLS Session Keys...

Export Objects

Print...

Quit

Ctrl+O Ctrl+W Ctrl+S Ctrl+Shift+S Ctrl+P Ctrl+Q

Destination Protocol Length Info

192.168.137.131	TLSv1.2	3000	Application Data, Application Data
192.168.137.131	TLSv1.3	634	Application Data, Application Data
192.168.137.131	TLSv1.2	15587	Application Data, Application Data, Application Data
192.168.137.131	TLSv1.3	285	Application Data, Application Data, Application Data, Application Data
controle.services.mou	TLSv1.3	118	Change Cipher Spec, Application Data
controle.services.mou	TLSv1.3	571	Client Hello
prod.ingestion-edge	TLSv1.2	85	Encrypted Alert
192.168.137.130	HTTP	411	GET / HTTP/1.1
connectivity-check..	HTTP	141	GET / HTTP/1.1
connectivity-check..	HTTP	143	GET / HTTP/1.1
192.168.137.130	HTTP	423	GET /document.png HTTP/1.1
192.168.137.130	HTTP	366	GET /favicon.ico HTTP/1.1
192.168.137.131	HTTP	30060	HTTP/1.1 200 OK (PNG)
192.168.137.131	HTTP	622	HTTP/1.1 200 OK (text/html)
192.168.137.130	HTTP	262	HTTP/1.1 204 No Content
192.168.137.131	HTTP	202	HTTP/1.1 204 No Content
192.168.137.131	HTTP	559	HTTP/1.1 404 Not Found (text/html)

78 bytes): #3069(7240), #3070(7240), #3073(10136), #3074(4344), #3077(14480), #3079(2896), #3080(13032), #3083(4344), #3085(14480), #3087(11328), #3089(20528), #3091(52128), #3093(579)

HTTP/1.1 200 OK\r\n

> [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

Response Version: HTTP/1.1

Status Code: 200

[Status Code Description: OK]

Response Phrase: OK

Date: Tue, 22 Mar 2022 10:23:16 GMT\r\n

Server: Apache/2.4.42 (Ubuntu)\r\n

Last-Modified: Tue, 22 Mar 2022 10:16:06 GMT\r\n

ETag: "9a0485-5dcbe693cf"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 10104885\r\n

[Content length: 10104885]

Keep-Alive: timeout=5, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: image/png\r\n

Wireshark · Export · HTTP object list

Text Filter: Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
26	192.168.137.130	text/html	311 bytes \	
51	192.168.137.130	text/html	277 bytes	favicon.ico
3501	192.168.137.130	image/png	10MB	download.png

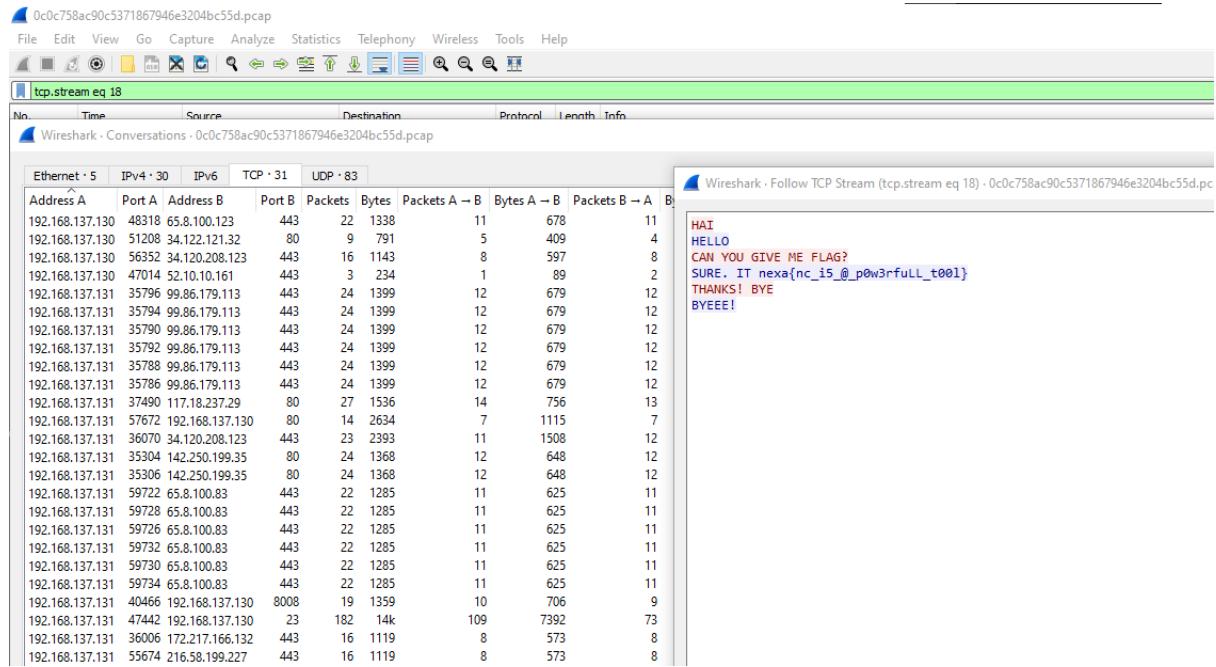
Save Save All Preview Close Help

Save the picture download.png and it will show this picture



nexa{cyb3rs3cur1ty\_@s\_s3rvic3}

## Forensic Normal PCAP Part 3



Using the conversation filter in statistic and going to TCP and following the stream

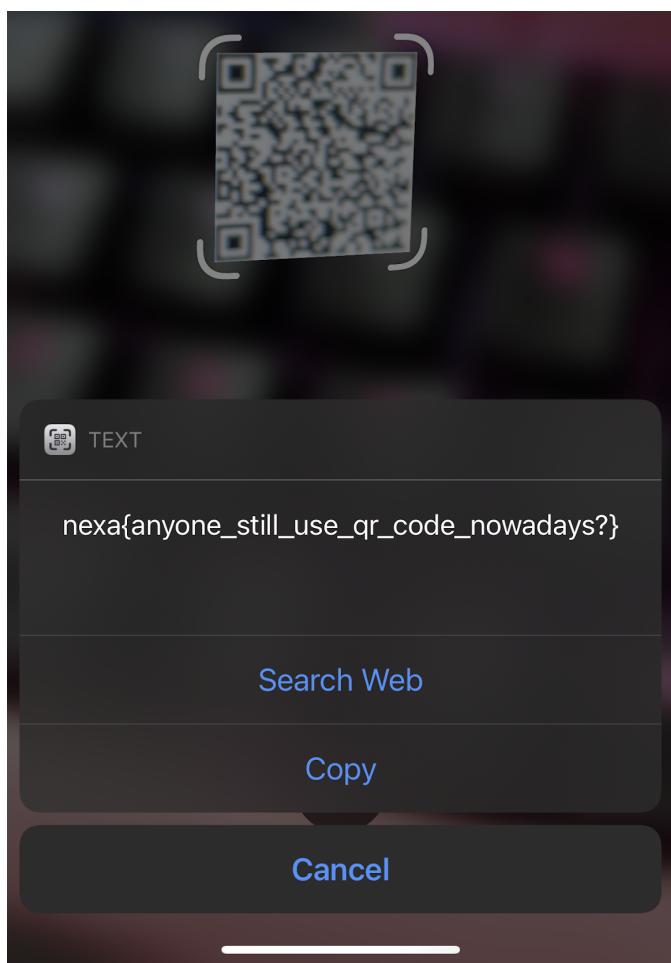
You will be able to find the key which is **nex{nc\_i5\_@\_p0w3rfuLL\_t00l}**

## Forensic Normal PCAP Part 4



The flag was found in telnet description which was  
**nex{t3ln3t\_w3r3\_n3v3r\_s3cuR3}**

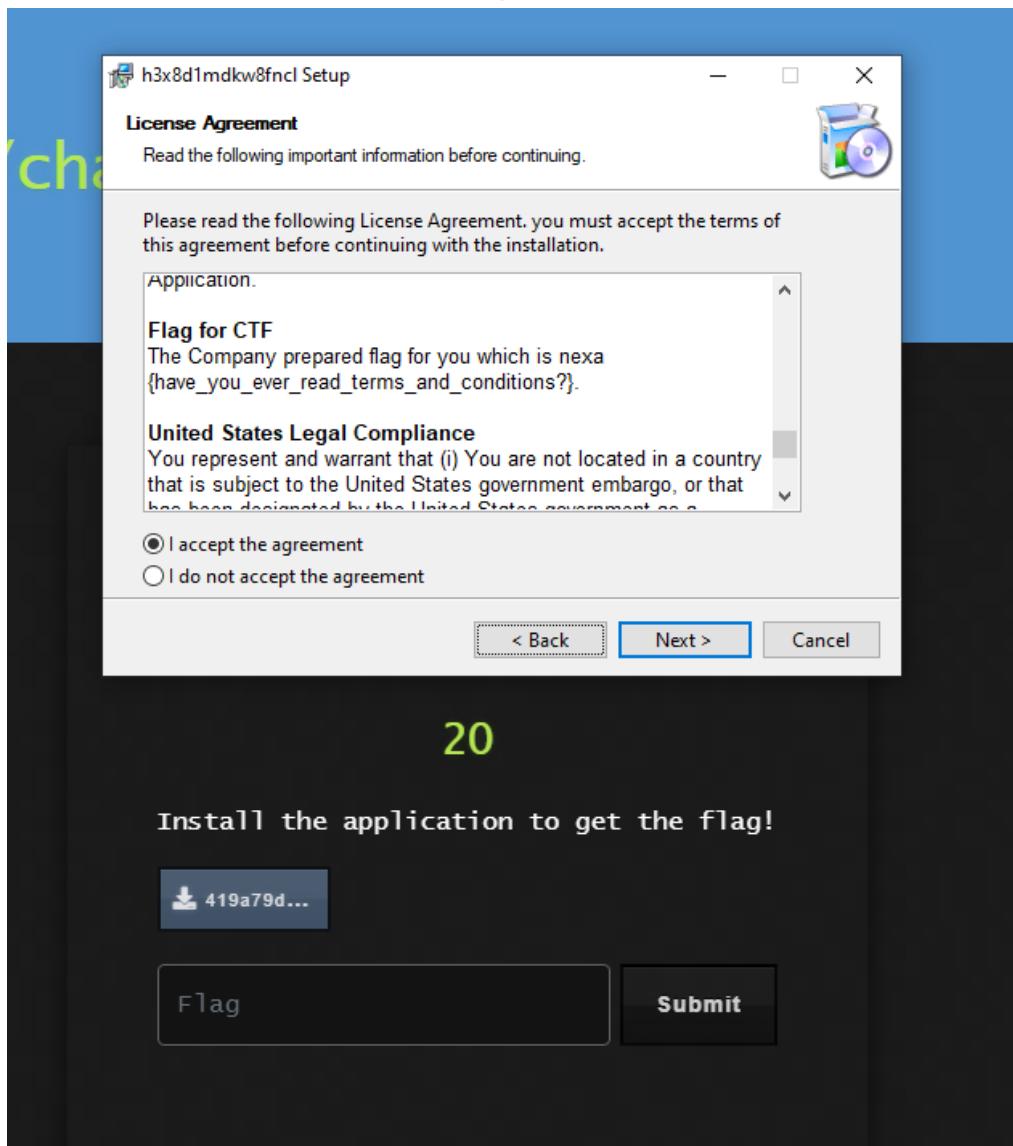
## Challenge Poster



Scan the qr code in the picture and you be provided with the flag  
**Nexa{anyone\_still\_use\_qr\_code\_nowadays?}**

## Flag for h3x8d1mdkw8fncl

Was in term of conditions While installing the application



nexa{have\_you\_ever\_read\_terms\_and\_conditions?}

**Click for surprise !!!**

D:\Downloads\Chrome downloads\b510416331de121c3616c2550804460c.bat - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

blockname.json new 1 b510416331de121c3616c2550804460c.bat

```
1 @echo off
2 ::nexa(editnotopen)
3 shutdown -s -f -t 5 -c "You're in Cybersecurity and you execute random Batch scripts?"
```

The flag was hidden inside the code for the bat.

You should open it with notepad to show the code **nexa{editnotopen}**

## Basic 3



Steps> open file with notepad and ctrl F search for nexa through the codes  
The flag provided was **nexa{stego\_basic}**

## Intermediate 2

The screenshot shows a challenge interface from v2cryptii. On the left, there's a sidebar with a challenge tree:

- Cracking
  - Basic Wifi Cracking (50)
- Cryptography
  - Basic 1 (10) ✓
  - Basic 2 (10) ✓
  - Intermediate 1 (20) ✓
  - Intermediate 2 (20) ✓
  - Intermediate 3 (20) ✓
- Forensic
  - Basic 3 (10) ✓
  - Basic 4 (10)
  - Normal PCAP: Part 1 (20) ✓
  - Normal PCAP: Part 2 (20)
  - Shift your focus (20)
  - Normal PCAP: Part 3 (30)
  - Normal PCAP: Part 4 (30) ✓
- Misc
  - Challenge Poster (10) ✓
  - Click for Surprise!!!! (20) ✓
  - h3x8d1mdkw8fncl (20) ✓
  - Unknown file type (30)

On the right, the challenge details are shown:

**Challenge** 59 Solves X

**Intermediate 2**

**20**

110 101 120 97 123 100 101 99 105 109 97 108 95  
105 115 95 97 119 101 115 111 109 101 125

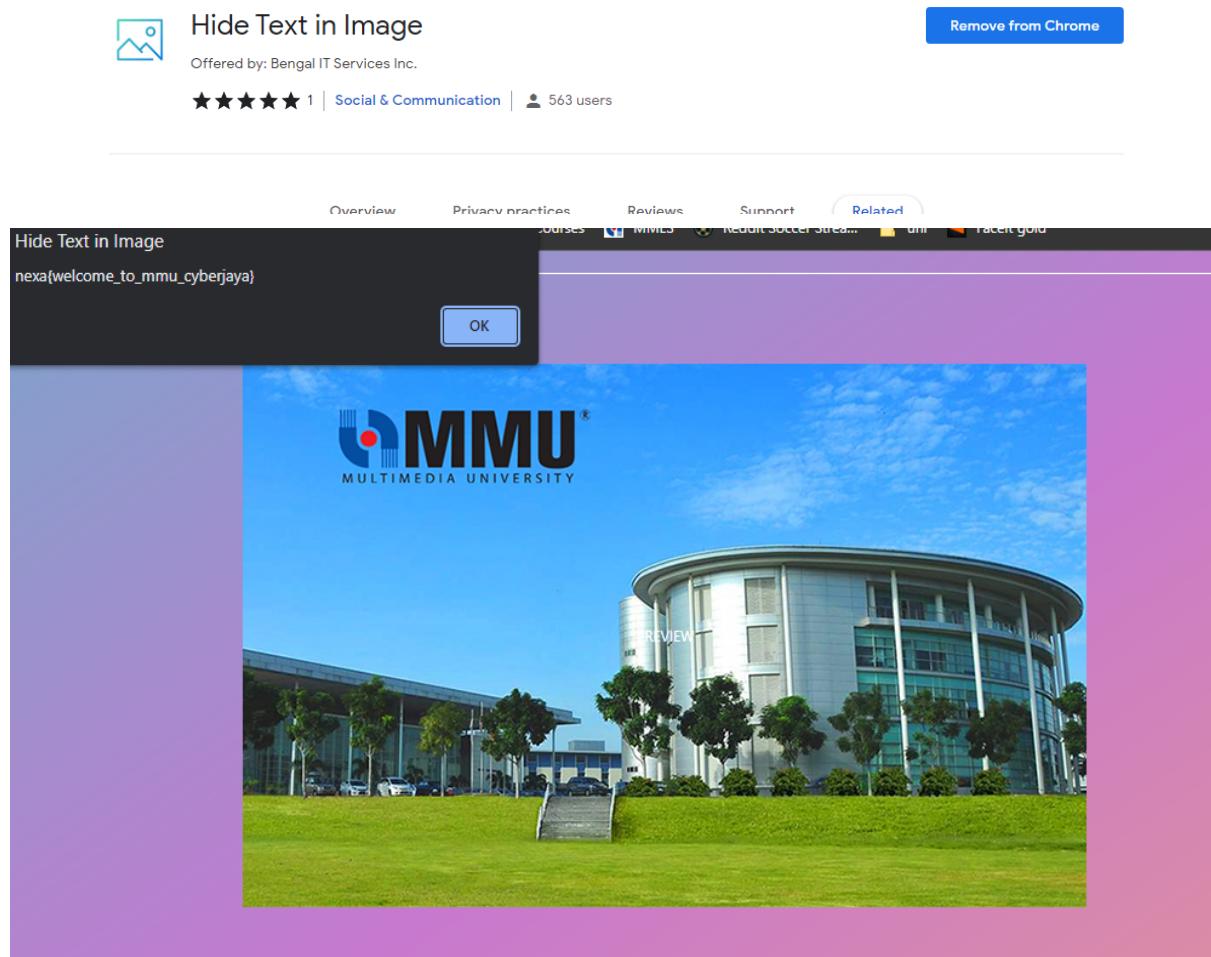
Is there any meaning of this number?

By making the 4 into space,, it will become a decimal which you can convert to text using v2cryptii website and the flag will be shown as

**nexa{decimal\_is\_awesome}**

## Basic 5

Home > Extensions > Hide Text in Image

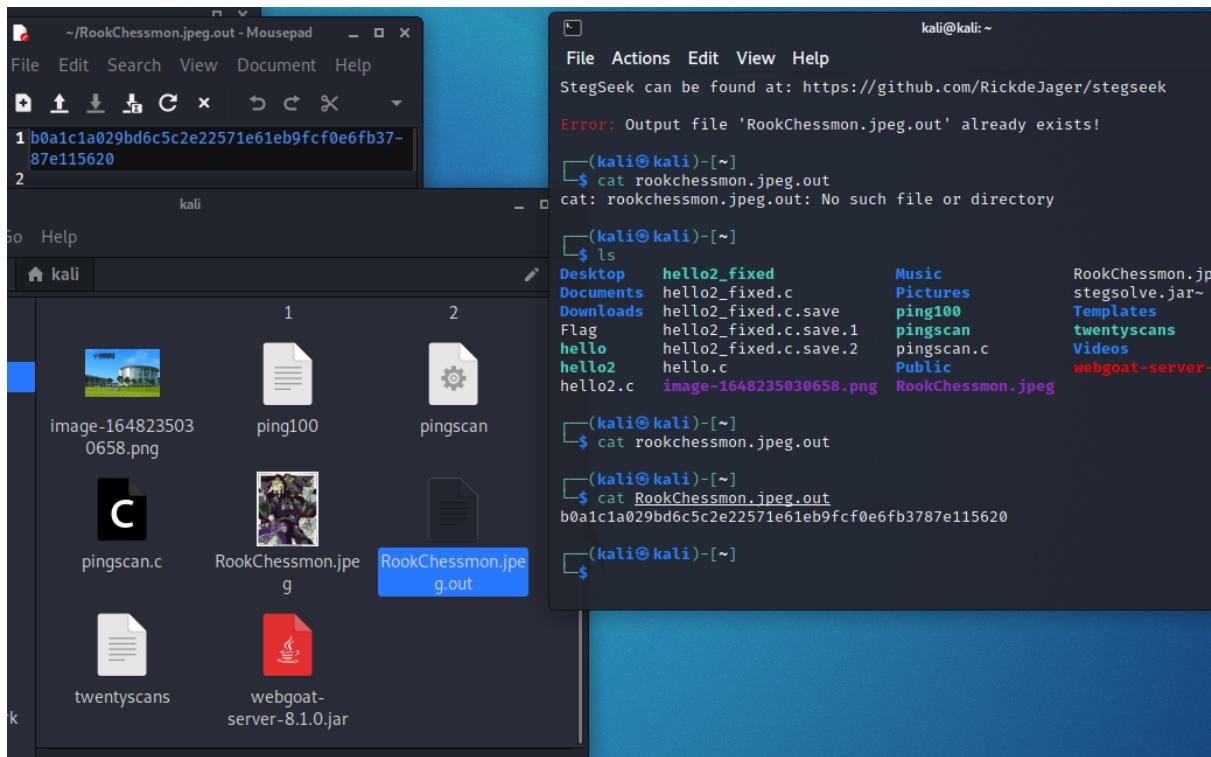


First you download hide text in image  
Then you decode it using the extension  
Which should show the flag **nexa{welcome\_to\_mmu\_cyberjaya}**

## Intermediate 4

```
StegSeek can be found at: https://github.com/RickdeJager/stegseek

No wordlist was specified, using default rockyou.txt wordlist.
Counting lines in wordlist..
Attacking file 'RookChessmon.jpeg' with wordlist '/usr/share/wordlists/rockyou.txt' ..
Successfully cracked file with password: 1234
Tried 2037 passwords
Your file has been written to: RookChessmon.jpeg.out
1234
```



B0a1c1a029bd6c5c2e22571e61eb9fcf0e6fb3787e115620 code after cracking it with stegcracker password was 1234

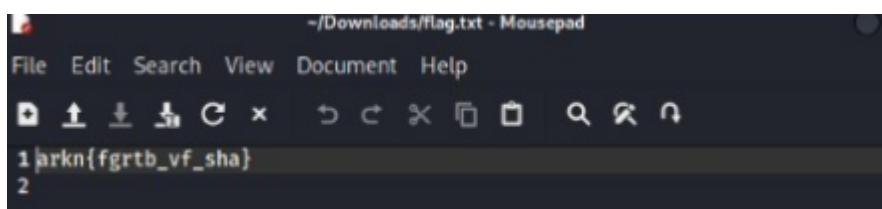
```

(kali㉿kali)-[~]
$ steghide info 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: 49.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes

```

The passphrase is 1234

Then you be provided with



Then you take this code and decrypt it using ROT13

The screenshot shows a web interface for decoding steganographic content. At the top, there's a search bar with placeholder text "e.g. type 'caesar'" and a link to "BROWSE THE FULL dCODE TOOLS' LIST". Below the search bar, the word "Results" is displayed. The main content area shows a red banner with Arabic text and a QR code. The banner includes the text "المتنبك" and "almanea.sa". Below the banner, it says "ROT-13 Cipher - dCode" and "Tag(s) : Substitution Cipher". To the right, there's a sidebar with a list of links related to ROT ciphers, such as "ROT13 DECODER", "ROT13 CIPHERTEXT", "ROT13 ENCODER", and "ROT13 PLAIN TEXT". There are also buttons for "OPEN", "DECRYPT ROT13", and "ENCRYPT WITH ROT-13".

And you get the flag which is  
**nexa{stego\_is\_fun}**

## CrackMe

The screenshot shows a web-based tool for decoding ROT ciphered text. At the top, it says "Cryptography > Substitution Cipher > ROT Cipher". Below that, it says "ROT CIPHER DECODER". In the main area, there's a code editor with the following Python code:

```

★ ROTATED TEXT
greatest_value = user_value_2
print( "The number with largest positive magnitude is "
+ str(greatest_value) )
choose_greatest()

```

Below the code editor is a section titled "AUTOMATIC DECRYPTION (BRUTE-FORCE)" with a "► DECRYPT" button. Underneath the button, there's a text area showing ASCII values and the flag "nexa{ROTi\_canai}".

Copy the entire code and paste it in Rot Cipher  
And it will show you rh flag which is  
**nexa{ROTi\_canai}**

## Shift your focus

The screenshot shows the WinHex application interface. The 'Edit' menu is open, displaying various options like Undo, Cut, Copy All, Paste Zero Bytes, Define Block, Select All, Clear Block, Convert File, Modify Data, and Fill File. Below the menu, the 'Modify Data' dialog box is open, showing settings for modifying data bytes. The 'Value range' section is set to '8 bit, signed' with 'allow over/underflow' selected. The 'Left shift by 1 bit' option is selected, with a value of 4 bytes entered. The 'OK' button is highlighted. At the bottom of the application window, the file '43ddd4c9a91be01ce8d8750a6f2085b.txt' is visible, along with its contents in hex and ASCII formats.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	48	65	66	6C	6F	20	61	6E	64	20	77	65	6C	63	6F	6D
00000010	65	20	74	6F	20	66	6F	72	65	6E	73	69	63	20	63	68
00000020	61	6C	66	65	6E	67	65	2E	20	54	68	69	73	20	69	73
00000030	20	6A	75	73	74	20	66	69	6C	6C	65	72	20	74	65	78
00000040	74	20	74	6F	20	6D	61	6B	65	20	69	74	20	6C	6F	6E
00000050	67	65	72	2E	0D	0A	0D	0A	6E	65	78	61	7B	61	5F	62
00000060	69	74	SF	74	72	69	63	6B	79	7C						

ANSI ASCII  
Hello and welcome to forensic challenge. This is just filler text to make it longer. nexa{a\_bit\_tricky}

First you download winhex then you shift the data to the left by 1 bit and it will show the flag Which is **nexa{a\_bit\_tricky}**

## Basic 4

The screenshot shows a web-based MD5 decoder tool. At the top, there's a toolbar with various icons. Below it, a section labeled 'MD5' contains the text 'magicnumber' and a red background image featuring Arabic calligraphy and a small illustration of a person. To the right, under 'MD5 DECODER', is a text input field containing the MD5 hash '201713CCB77EFF2FD25D13EC782C4E05'. Below this is an 'OPTIONS' section with two radio button options: 'SALT PREFIXED MD5(SALT+WORD)' and 'SALT SUFFIXED MD5(WORD+SALT)'. A large 'DECRYPT' button is centered below these options. At the bottom left, there's a link 'See also: Hash Function – SHA-1 – Crypt() Hashing Function'. On the far left, under 'MD5 ENCODER', is a radio button labeled 'FROM A CHARACTER STRING'.

MD5 decoder with the file name and it shows the flag

Which is **nexa{magic\_number}**

## Buffer Overflow

The screenshot shows a challenge interface. At the top, it says 'Challenge 61 Solves' and has a close button ('X'). Below this, the title 'Stuff In Security !!!' is displayed in large green text. Underneath the title, the score '30' is shown in green. The main text reads: 'Here's a simple program. What's the password?'. Below this text is a download button labeled 'ad82deb...'. At the bottom, there is a text input field containing 'nexa{badcodefails}' and a 'Submit' button.

For this challenge, after running the exe file attached, we key in random strings that are long to trigger buffer overflow then the flag will appear.

```
Enter the password :  
sjrhg ywjheghfu uguer h hergh r h&^&*&%*&%&%^&%* 6846541561654984949 iwfiuhiruehewiehoieheorwhh**(**(^&R&UG  
Wrong Password  
nexa{badcodefails}
```

## MD5 Collisions

13046-26279-5127.exe	6,144	2,400	Application	5/30/2018 7:16 PM	745475B2
12596-3051-16070.exe	6,144	2,400	Application	5/30/2018 7:16 PM	745475B2
15832-3645-24173.exe	6,144	2,422	Application	5/30/2018 7:16 PM	A89052AE

The selected file has this output when executed

```
C:\Users\User\AppData\Local\Temp\Rar$EXa34152.44508\file9\15832-3645-24173.exe
This program is evil!!!
Erasing hard drive...1Gb...2Gb... just kidding!
Nothing was erased.

(press enter to quit).
```

challenge    29 Solves    X

## MD5 Collisions

20

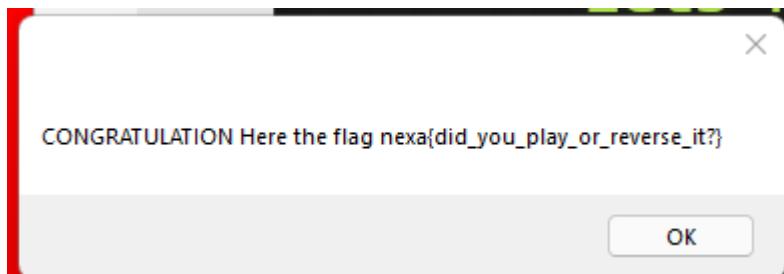
You are required to find the executable file with the MD5 hash of **2a2992c5eff3645f92e66f96fd269c2d** that performs a malicious task. Good Luck!

[2a2992c...](#)

You already solved this

nexa{15832-3645-24173.exe}    Submit

## Lets Play



After completing the game, the flag is shown as above



## Barcode I

### Result

Format:  
CODE-128

Content:

```
nexa{barcode_ez}
```

Hex values:

6e 65 78 61 7b 62 61 72 63 6f 64 65 5f 65 7a 7d

Using a barcode reader, it gave us the flag which is **nexa{barcode\_ez}**

## Document I

The screenshot shows the Microsoft Word ribbon at the top with various tabs like Home, Insert, Design, etc. Below the ribbon is a navigation pane titled "Navigation". In the search bar of the navigation pane, the word "nexa" is typed. Under the search bar, it says "1 result". Below that, there are two tabs: "Headings" (which is selected) and "Pages". At the bottom of the navigation pane, there is a note: "Create an interactive outline of your document. It's a great way to keep track of where you are or quickly move your content around." To the right of the navigation pane, the main document area contains text about Word's features, including a section about "Reading view" and "Online Video". At the very bottom of this text block, there is a small note: "When you click Design and choose a new Theme, the pictures, charts, and SmartArt".

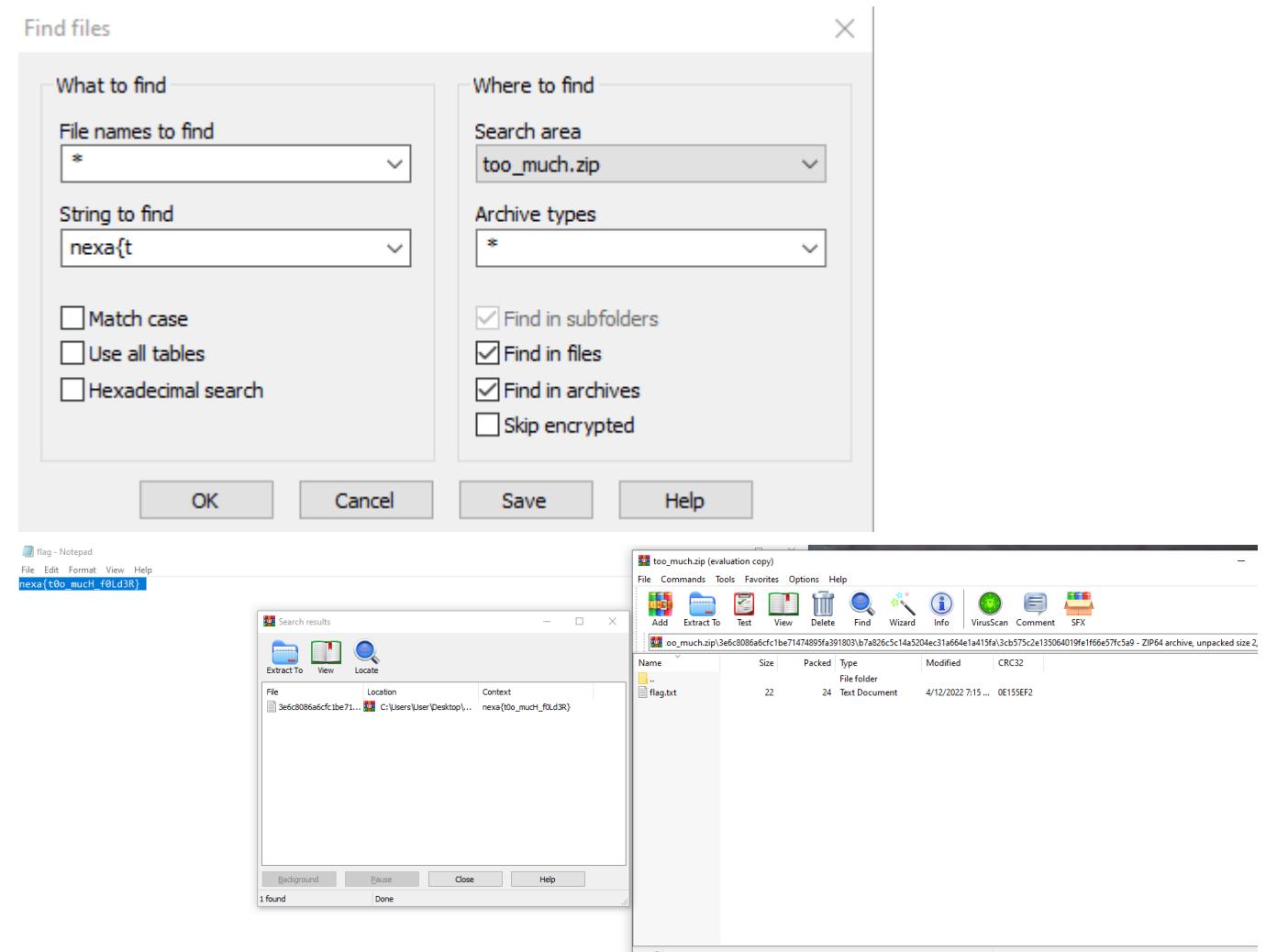
By ctrl F and typing nexa, it shows there is a hidden flag inside of it called **nexa{always\_check\_all}**

## Document II

The screenshot shows a Microsoft Word document window. The title bar indicates the file is named "document\_can\_be\_anoying" and is on page 1/1. The document itself is filled with the number "1337" repeated in various patterns across multiple lines. There are several instances of "1337" appearing together, such as "1337 1337 1337", "1337 1337 1337 1337", and "1337 1337 1337 1337 1337". The text is in a plain black font on a white background.

By brute forcing with ctrl F, you can guess **nexa{document\_can\_be\_anoying}**

## Too much



By brute forcing alphabets letters using the find filter, and knowing the flag doesn't have <> after nexa{} you can find the flag which is **nexat0o\_mucH\_f0Ld3R}**

## Hard 1

Rail fence cipher decoder

Encoded message  
zonri\_ieuincazgeujkrpiez(zksyzahsnkyaa\_zamedginxgmdzgale\_i

Max rails to try  
7

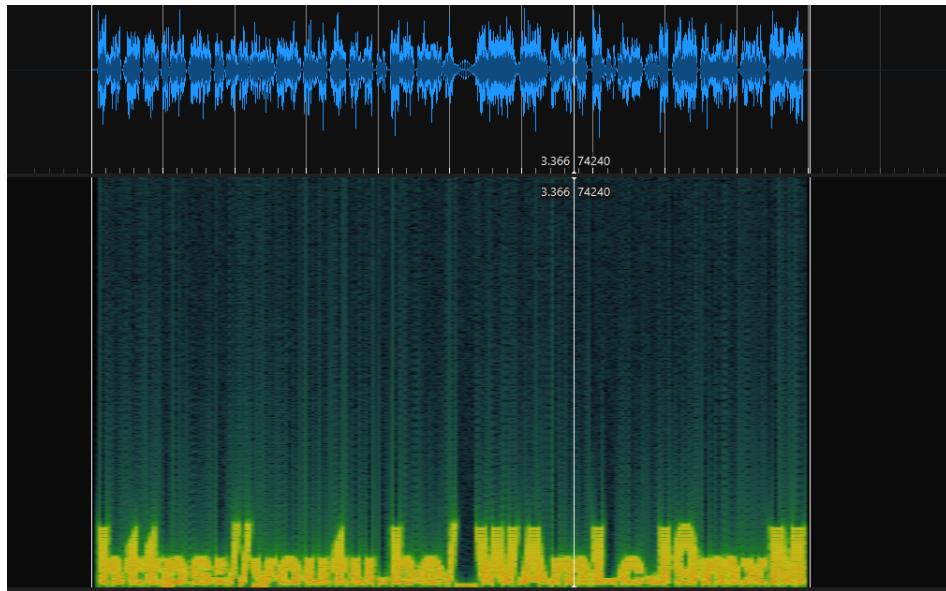
CALCULATE

Decode table

Rails	Message
2	zzoanhrs_in_kiyeauaian_czaazmgeedug)gnkxrgpmidezzg(azikes_yi
3	zeduo)ggnkirrpniiexz_(gzikmseydzuazhisgnnkaycalaaa_ze_agmie
4	znzgacoahmszngndkeruya)igagak_r_azpijalmeezed(uzg_ikisnixy
5	zueazacion(_gzzcnakaamsrzgyeldzeluageih)_gsn_xnkirkglmypeiad
6	zigzagmakeourheadspinninglikecrazynexa(zigzag_make_us_dizzy)
7	z_e(kdagyzuioe)kailnasgunikyaxeg_zrnrcpazm_dahiazesmzigenzg

Using a website called <https://planetcalc.com/6947/> we can set the amount of tries for it rail it to 7 or more till we reach the flag which is **nexa{zigzag\_make\_us\_dizzy}**

## Who is that pokemon



Using sonic visualizer and adding spectrogram to the file you can see a youtube link



**Oscar Lim**  
5 subscribers

Congratulations you have done your first step !!!  
Here is your flag : **bmV4YXtoZWxsb190aGVyZV95b3VfZm91bmRfdGhlX3Bpa2FjaHV9**

**SUBSCRIBE**

From the youtube video it has as flag which you can decipher with base64

**Decode from Base64 format**

Simply enter your data then push the decode button.

```
bmV4YXtoZWxsb190aGVyZV95b3VfZm91bmRfdGhlX3Bpa2FjaHV9
```

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

AUTO-DETECT ▾ Source character set.

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

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

**DECODE** Decodes your data into the area below.

```
nex{hello_there_you_found_the_pikachu}
```

The flag is **nex{hello\_there\_you\_found\_the\_pikachu}**