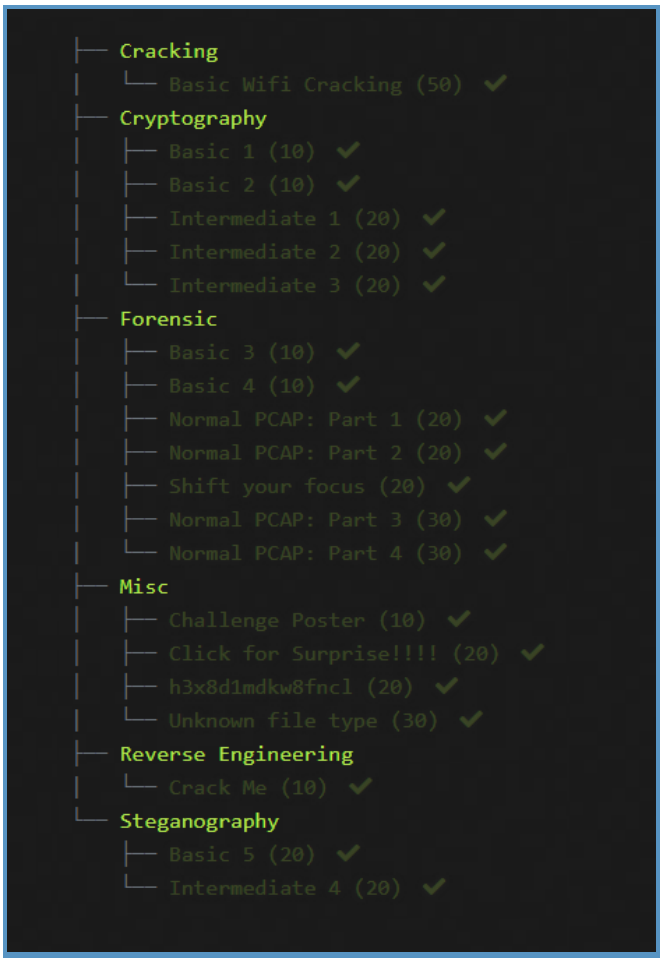
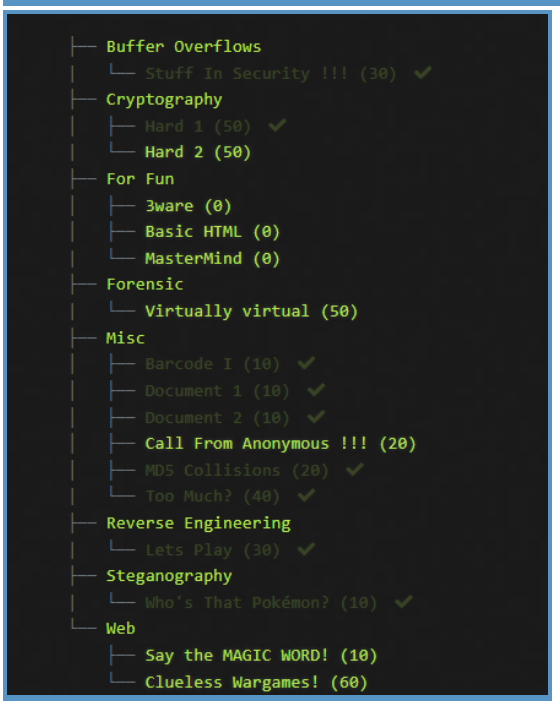
TEH3261/ FTEH3261
ETHICAL HACKING AND SECURITY ASSESSMENT
CTF Write Up

Lab Session : 1BV
Team Name : 3AM Hackers
Team Leader : 1191302861 Mayar Abdulmalik M Shenawi
Group Member 1 : 1181102166 Ahmed Aldughaither
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 Pokémon? (10)
Web
Say the MAGIC WORD! (10)
Clueless Wargames! (60)

Questions

Wifi crack

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

Select file... Browse...

Output - display best & valid hash(es)

```
WPA*02*b782c5216fa81c76d4b196bd2acd24a3*6a3e26ef7653*66ee2ef02f42*6e6578617365637
572656e6574776f726b6b6b6b*9f7ffde8f23b69646b624f05fabf4399cfad10b4ec1543f012dd82be
592dd7bf*0103007502010a00100000000000000000002a3d29d0cde0ede058efc8915648ba0b4d01
b6c1eaffeb96e086c469082d25a900000000000000000000000000000000000000000000000000000
```

Flag nexa{betoeresmivida12}

Unknown file type

Flag | Flag (1)

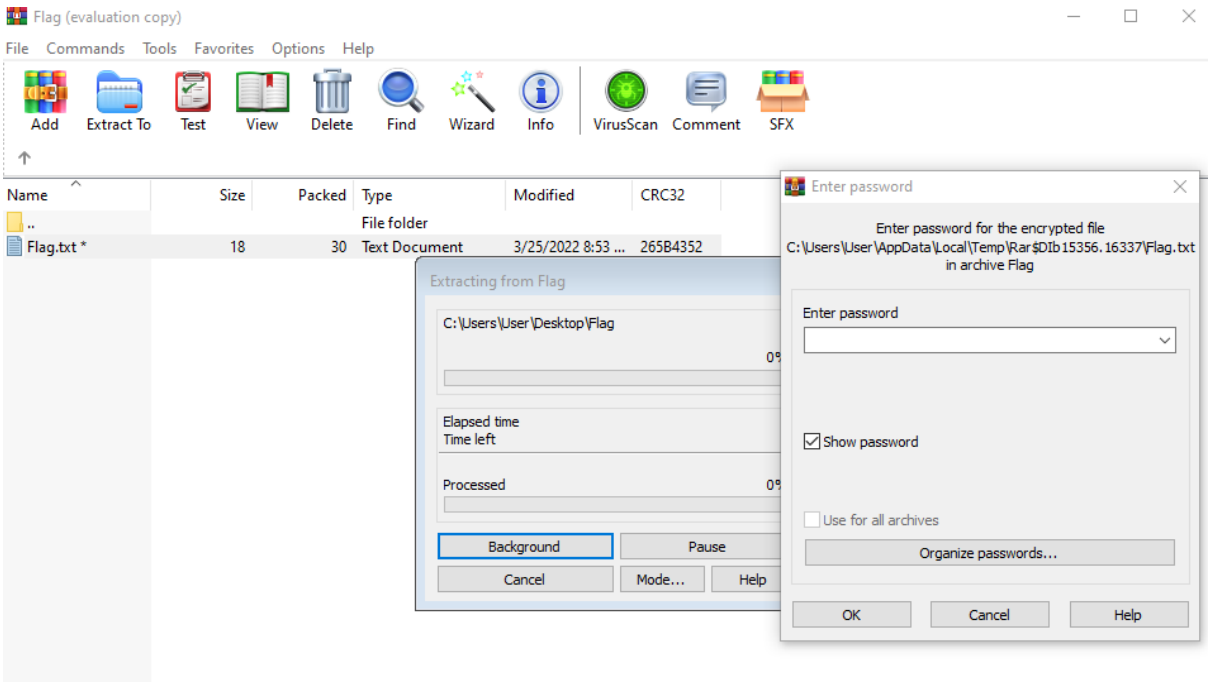
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	00	00	03	04	14	00	01	00	00	00	AD	0E	7A	54	52	43	- zTRC
00000010	5B	26	1E	00	00	00	12	00	00	00	08	00	00	00	46	6C	[& Fl
00000020	61	67	2E	74	78	74	7D	32	C1	FB	E1	08	2B	97	5E	34	ag.txt}2ÁûÁ +—^4
00000030	98	52	03	1C	05	01	57	E9	15	DA	6C	74	33	A7	3B	94	R Wé Últ3§;”
00000040	6D	92	E9	B3	50	4B	01	02	3F	00	14	00	01	00	00	00	m’é³PK ?
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 ¹é
00000090	44	60	71	40	D8	01	F2	2D	4A	F6	70	40	D8	01	50	4B	D`q@Ø ò-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

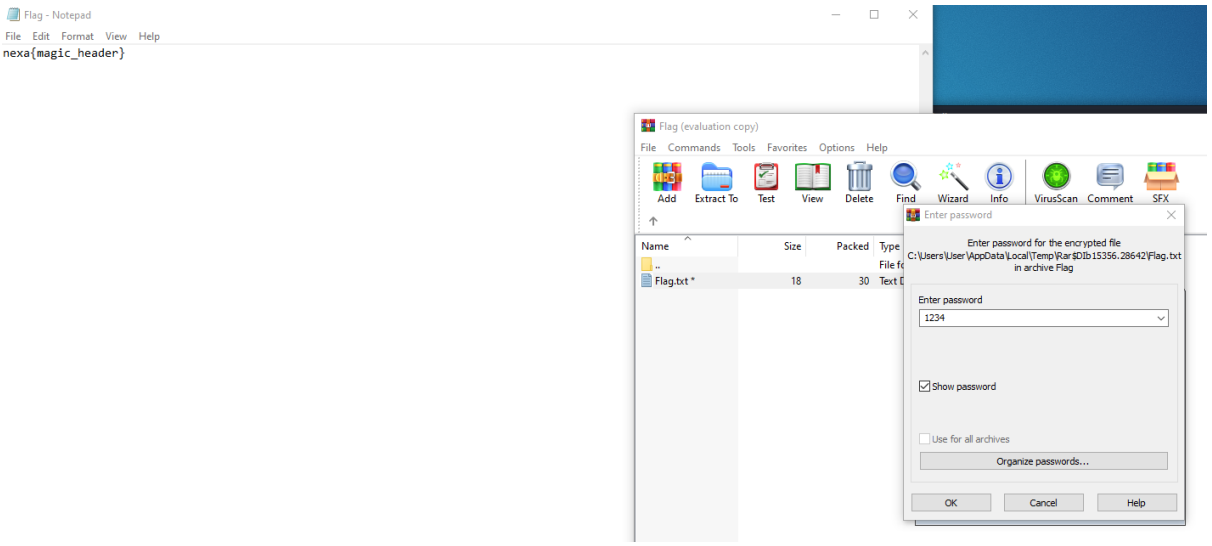
Flag

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	PK - 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}2ÁûÁ +—^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'é³PK ?
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 ¹é
00000090	44	60	71	40	D8	01	F2	2D	4A	F6	70	40	D8	01	50	4B	D`q@Ø ò-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													

Change the first few hex to change to a zip file

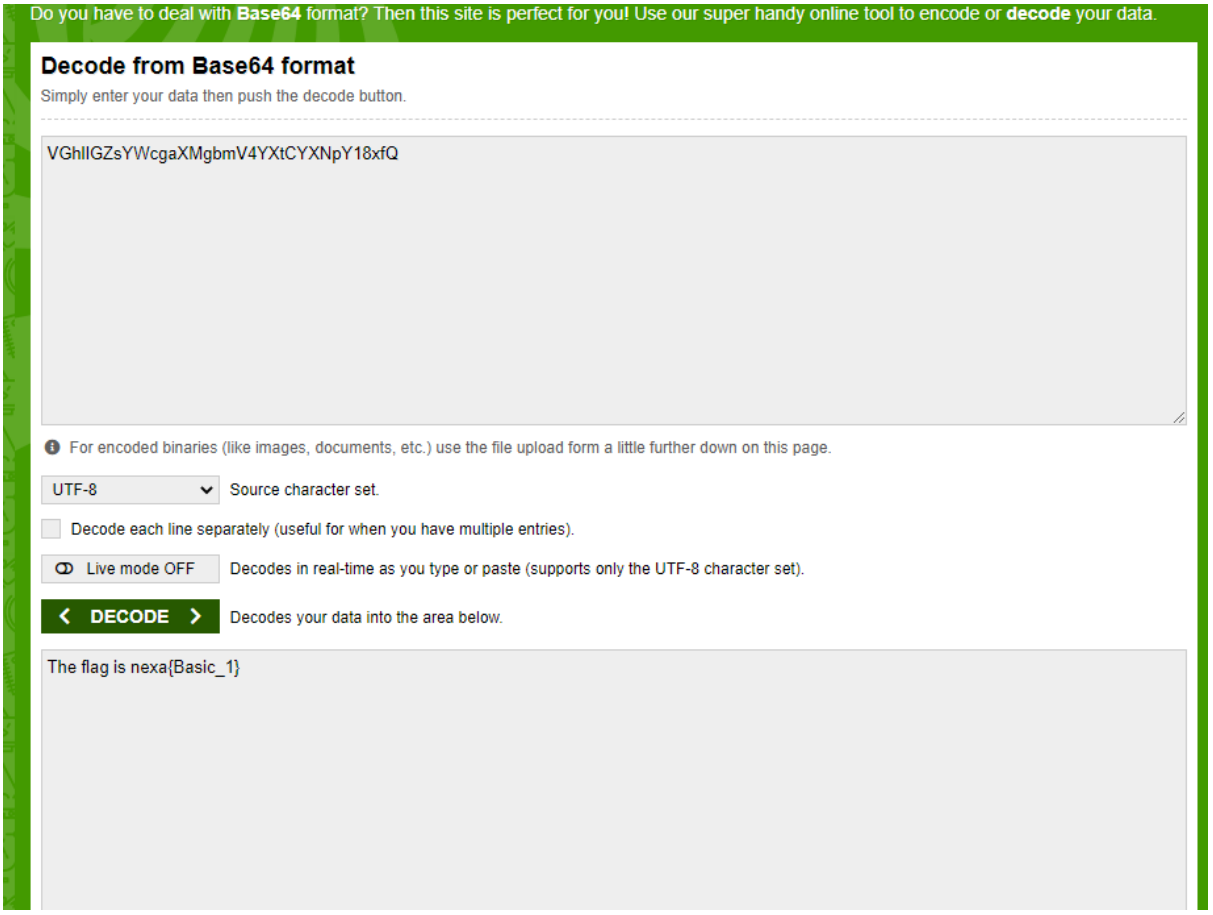


Open the file with winrar and it will request a password



Using jack the ripper we found the password to be 1234
And the flag is **nexa{magic_header}**

Basic 1



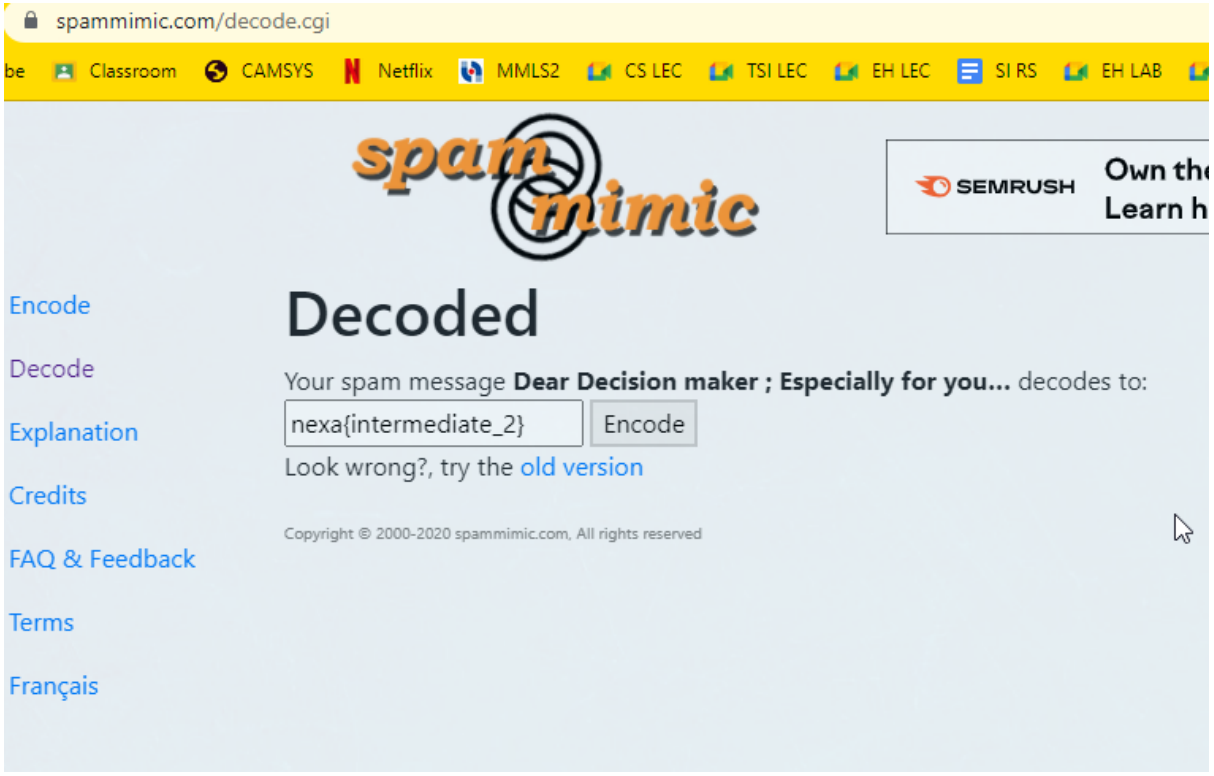
Basic base64 converter showed the answer is **nexa{Basic_1}**

Basic 2

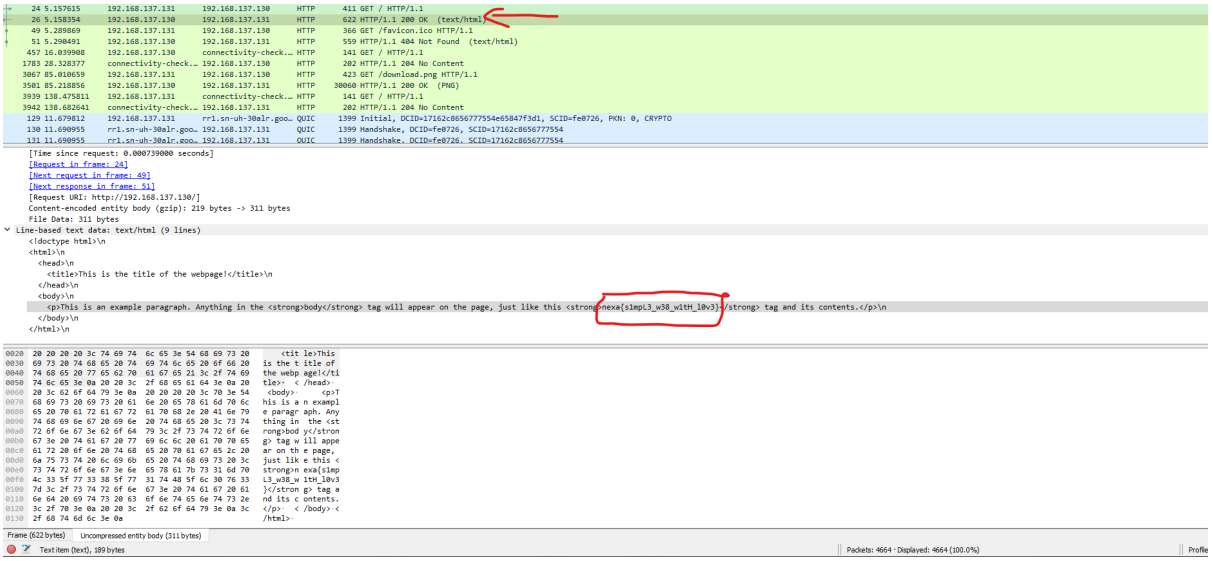
Morse code converter showed flag is **nexa{MORSE_CODE_IS_NICE}**

Intermediate 1

To decode the email message, we used spammimic.
The encoded message is **nexa{intermediate_2}**



Forensic PCAP part 1

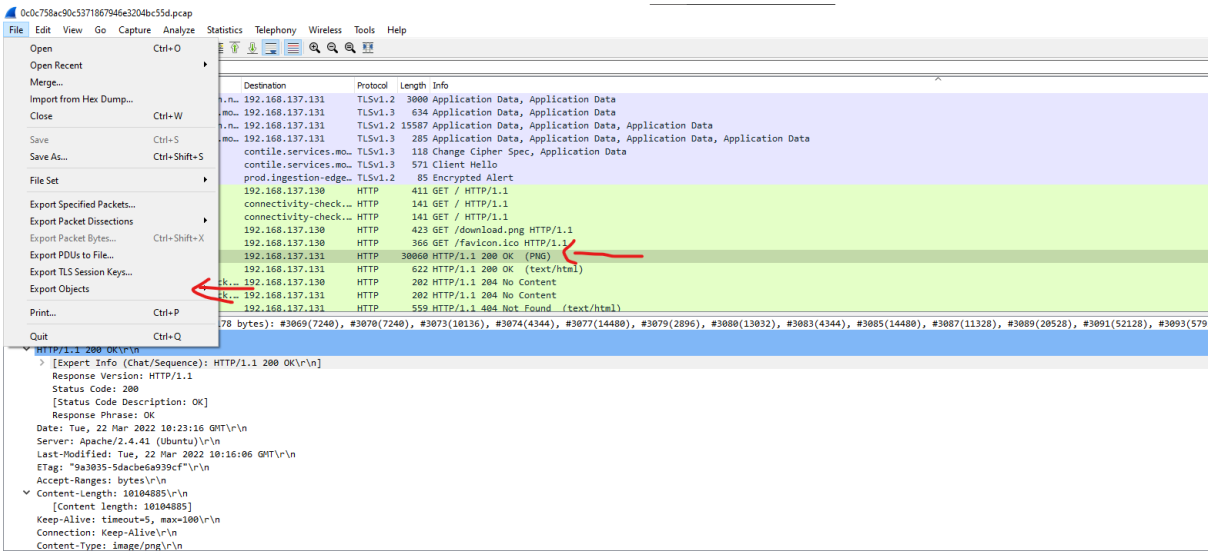


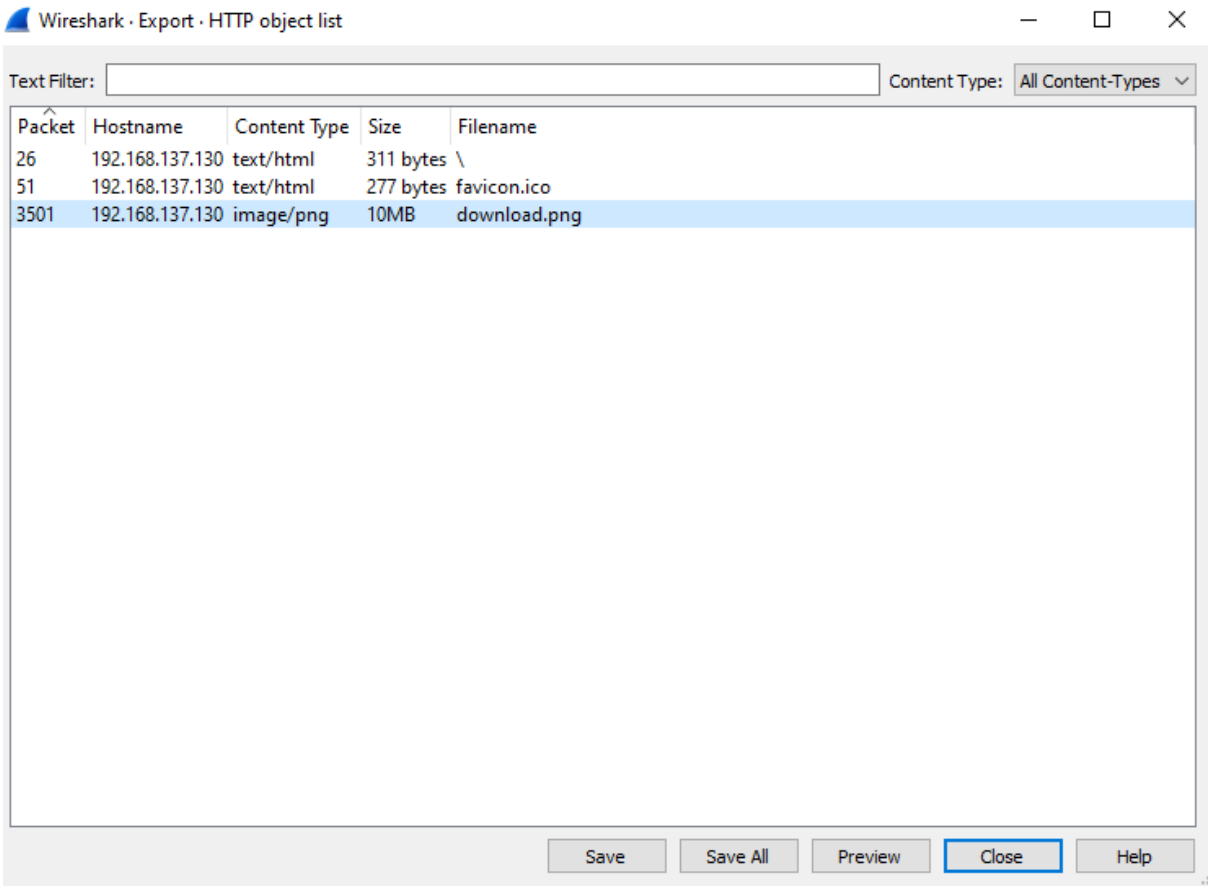
The code was hidden in HTTP line-based data
and the flag was **nexa{s1mpL3_w38_w1tH_l0v3}**

Intermediate 3

To get the flag you need decode with base64 then decode it again with rot-13 cipher to get the flag which was **nexa{Hard_yet_can_be_solved}**

Forensic Normal PCAP Part 2

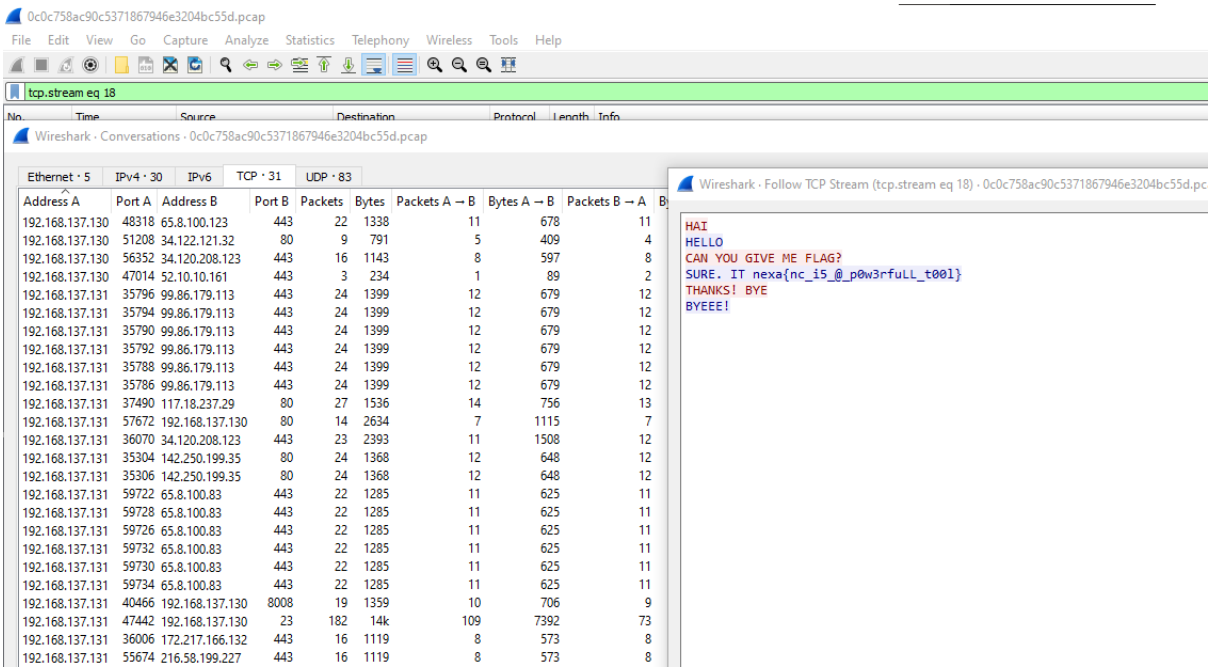




Save the picture download.png and it will show this picture

nexa{cyb3rs3cur1ty_@s_s3rvic3}

Forensic Normal PCAP Part 3

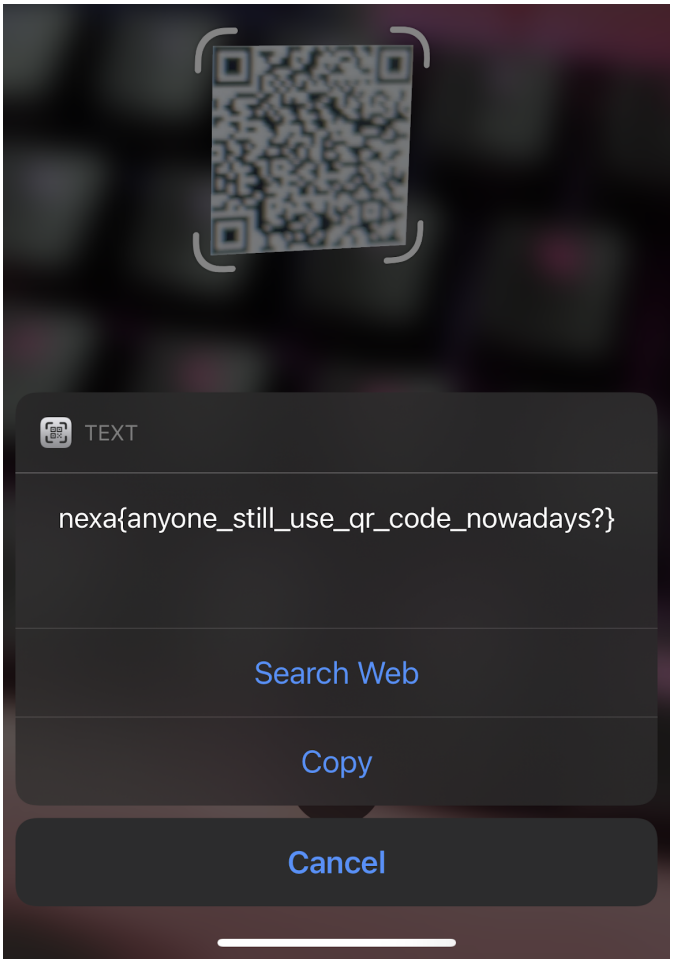


Using the conversation filter in statistic and going to TCP and following the stream
You will be able to find the key which is **nexa{nc_i5_@_p0w3rfuLL_t00l}**

Forensic Normal PCAP Part 4

The flag was found in telnet description which was
nexa{t3ln3t_w3r3_n3v3r_s3cuR3}

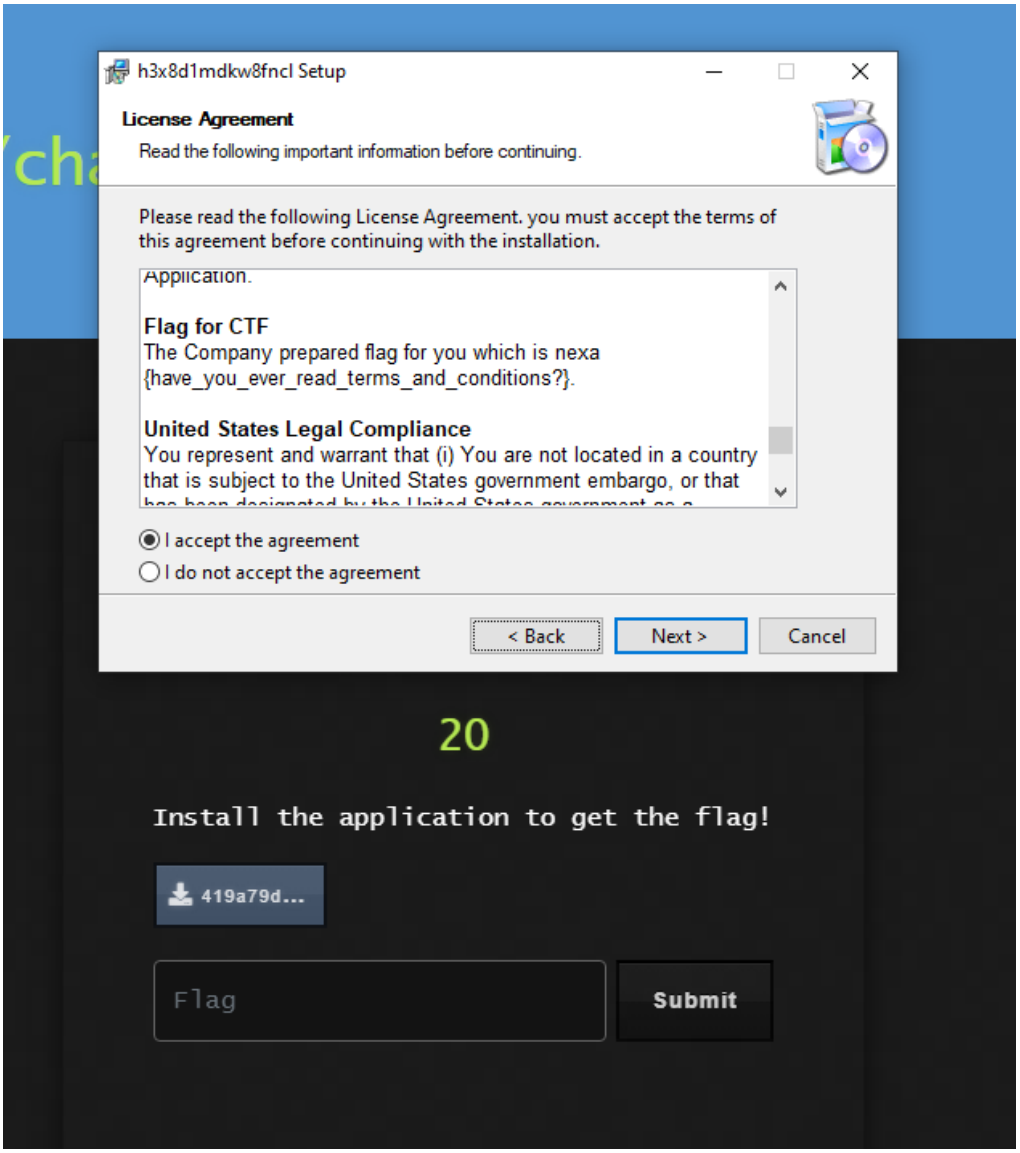
Challenge Poster



Scan the qr code in the picture and you be provided with the flag
Nexa{anyone_still_use_qr_code_nowadays?}

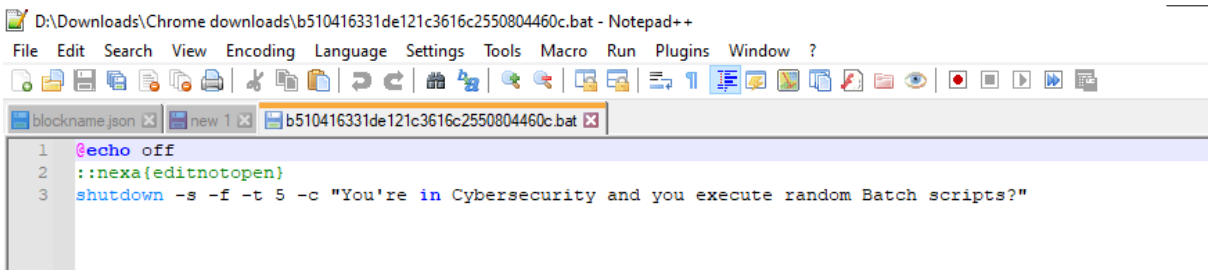
Flag for **h3x8d1mdkw8fncl**

Was in term of conditions While installing the application



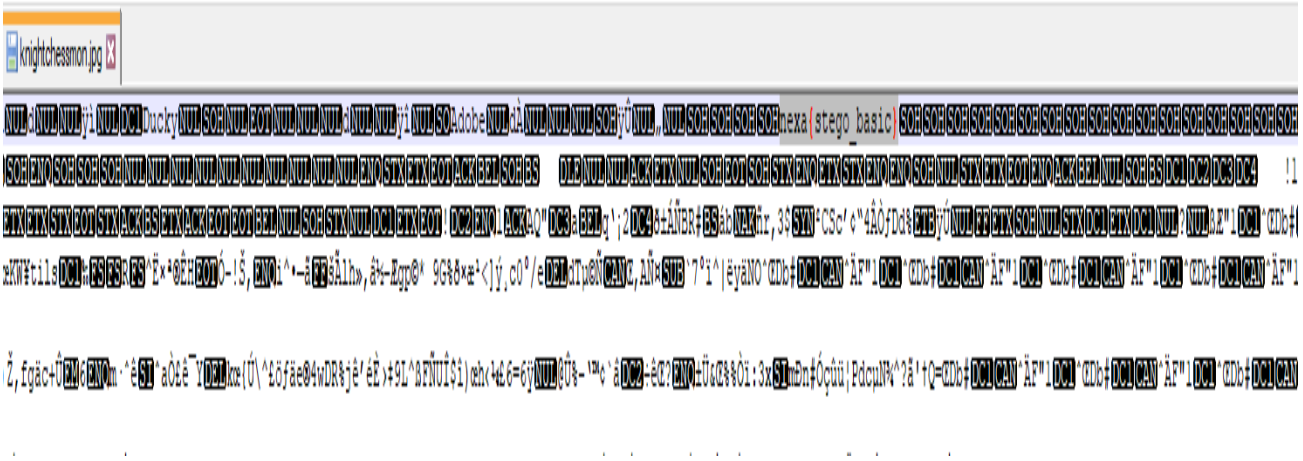
nexa{have_you_ever_read_terms_and_conditions?}

Click for surprise !!!



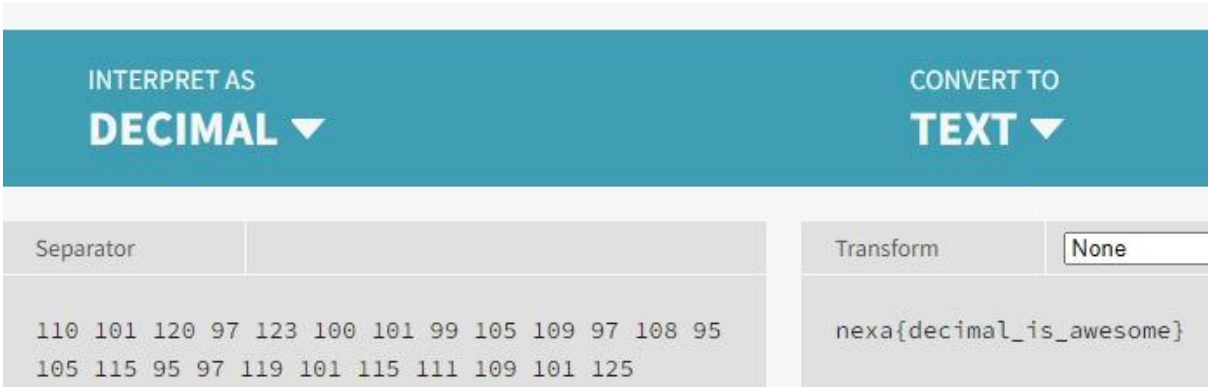
The flag was hidden inside the code for the bat.
You should open it with notepad to show the code **nexa{editnotopen}**

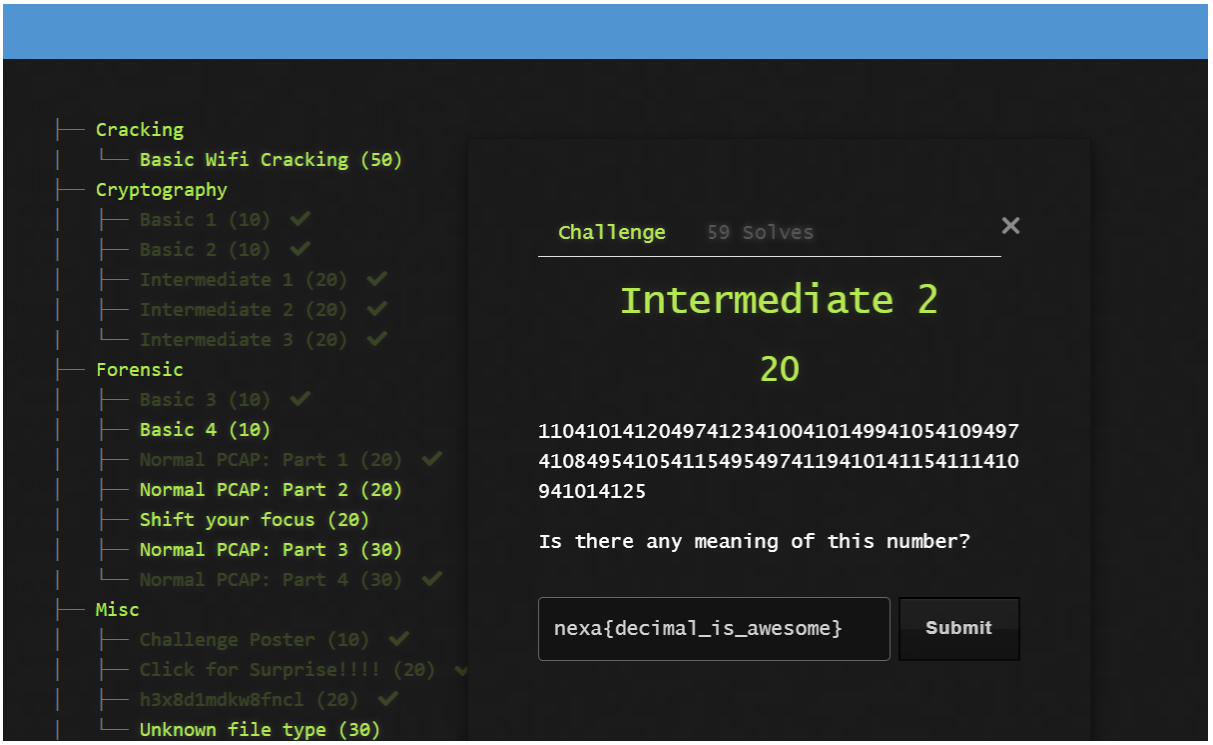
Basic 3



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

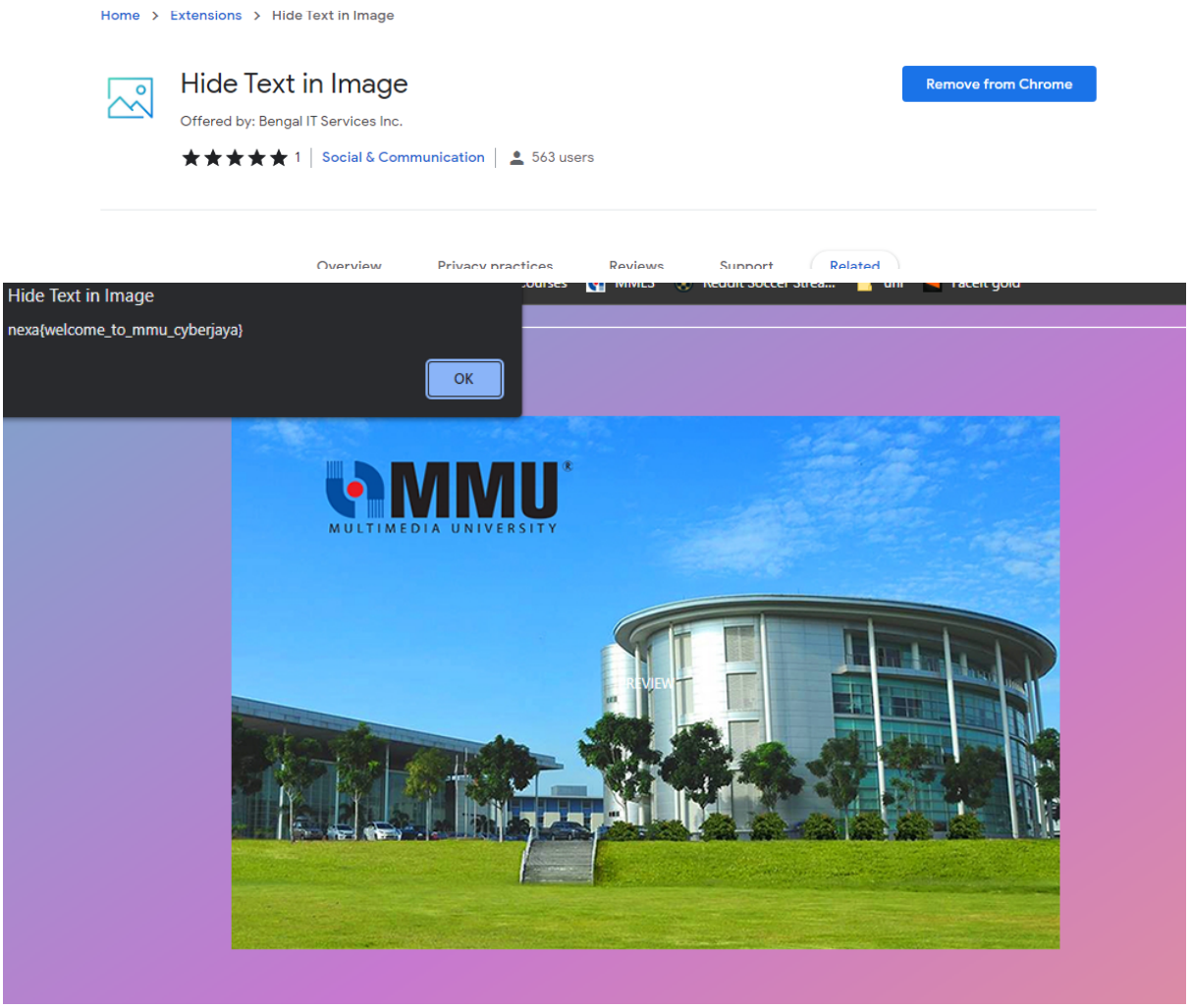
Intermediate 2





By making the 4 into space,, it will become a decimal which you can convert to text using v2 cryptii website and the flag will be shown as
nexa{decimal_is_awesome}

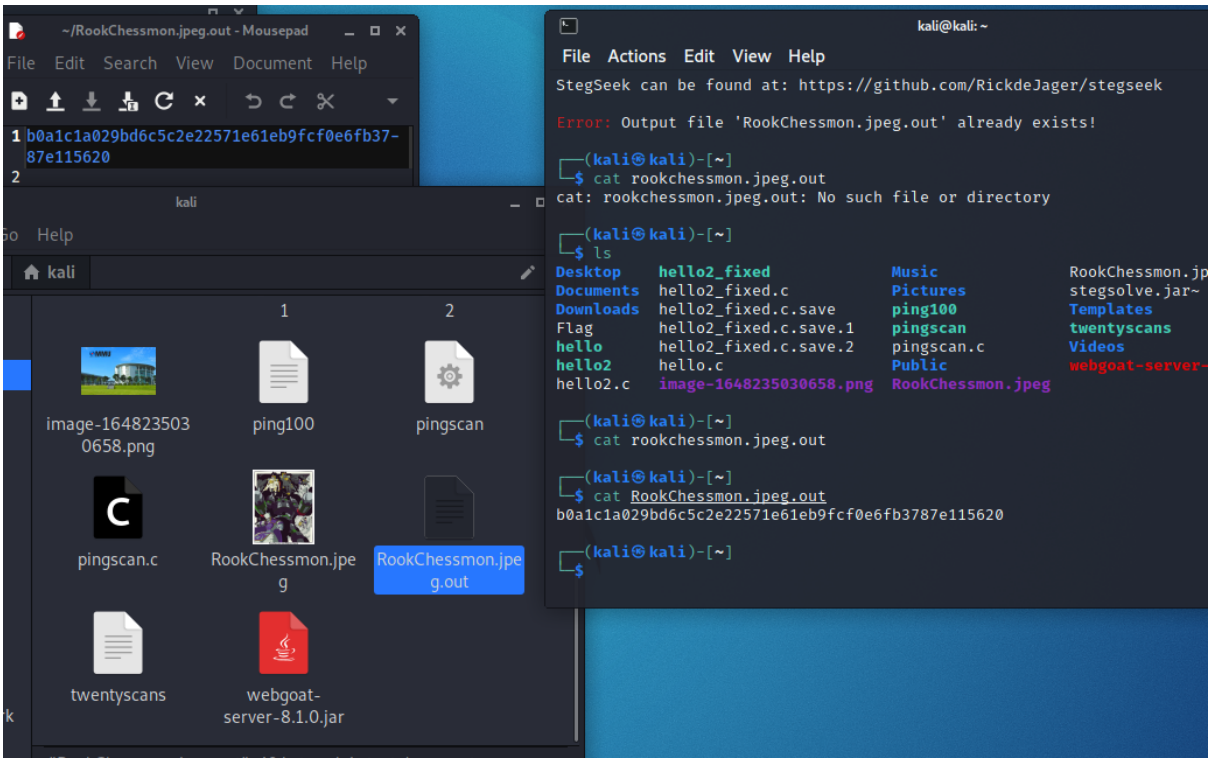
Basic 5



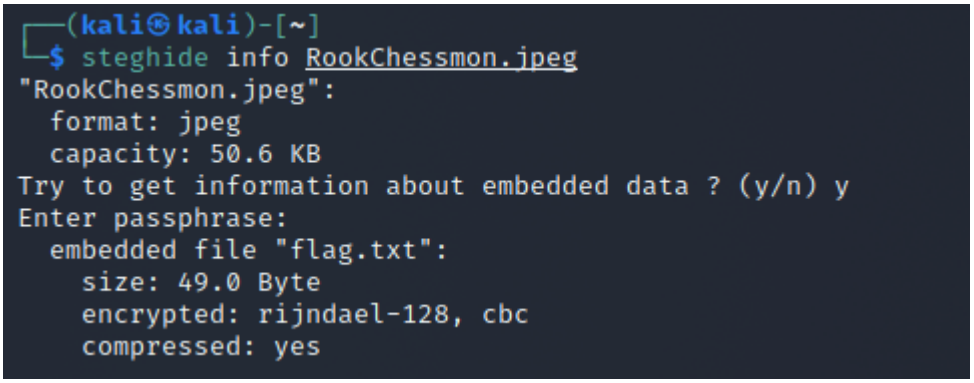
First you download hide text in image
Then you decode it using the extension
Which should show the flag **nexa{welcome_to_mmu_cyberjaya}**

Intermediate 4

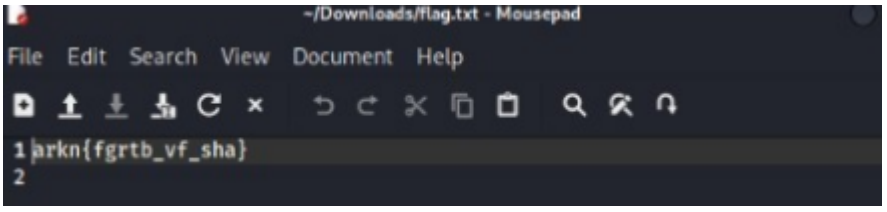
```
StegSeek can be found at: https://github.com/RickdeJager/stegseek

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
```



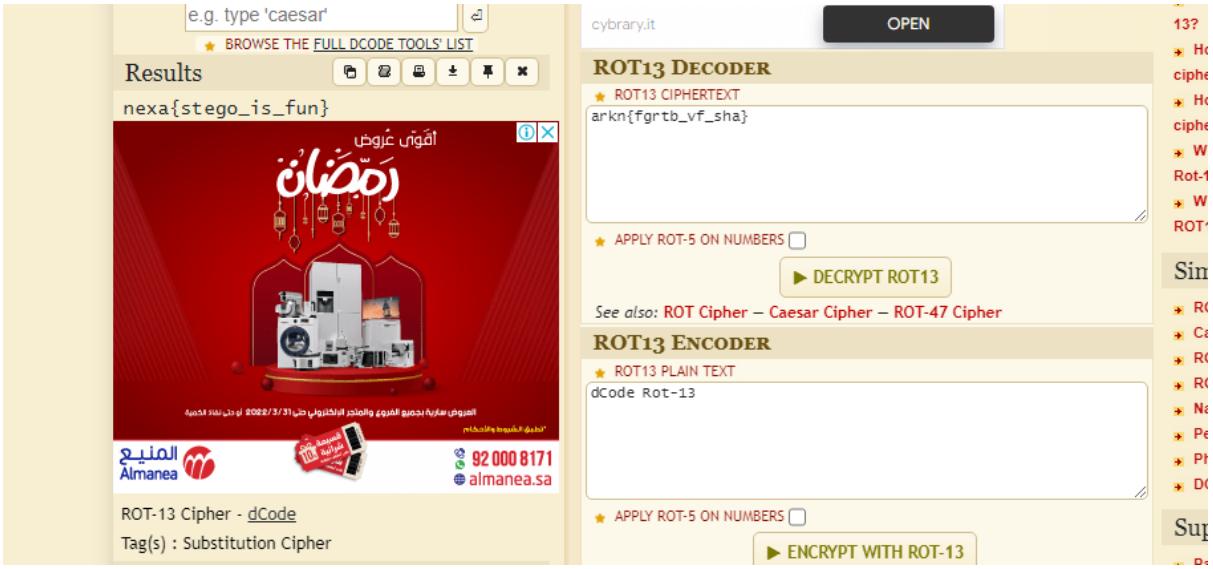
B0a1c1a029bd6c5c2e22571e61eb9fcf0e6fb3787e115620 code after cracking it with stegcracker password was 1234



The passphrase is 1234
Then you be provided with

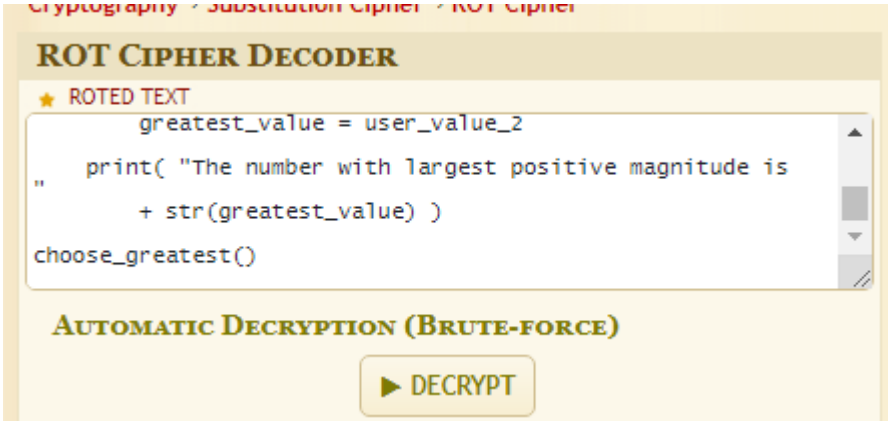


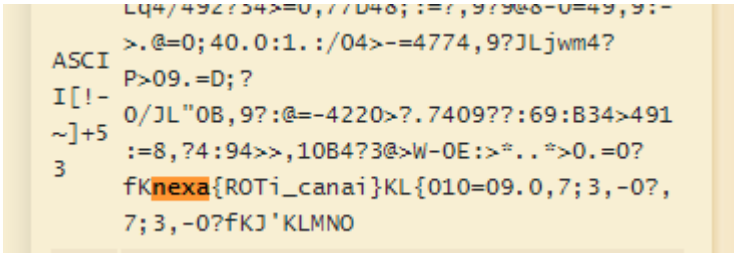
Then you take this code and decrypt it using ROT13



And you get the flag which is
nexa{stego_is_fun}

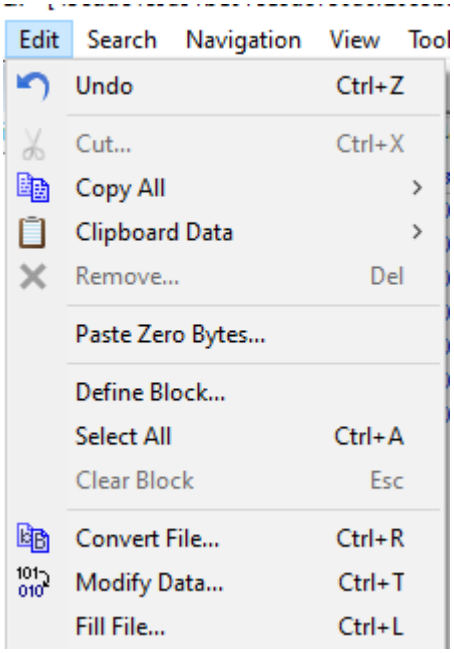
CrackMe

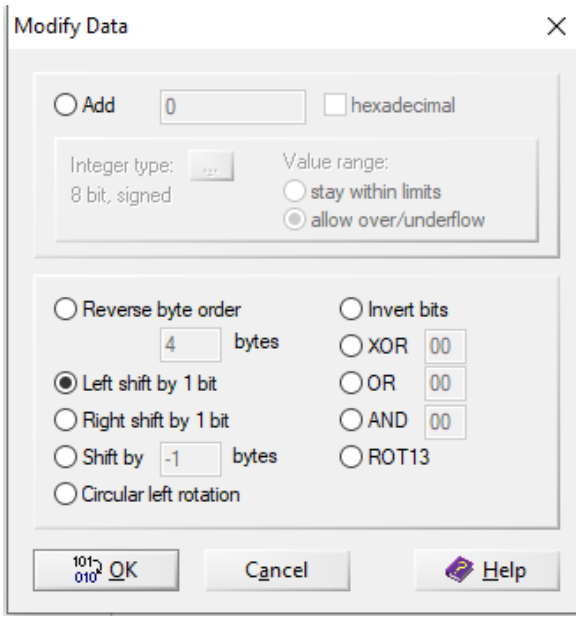


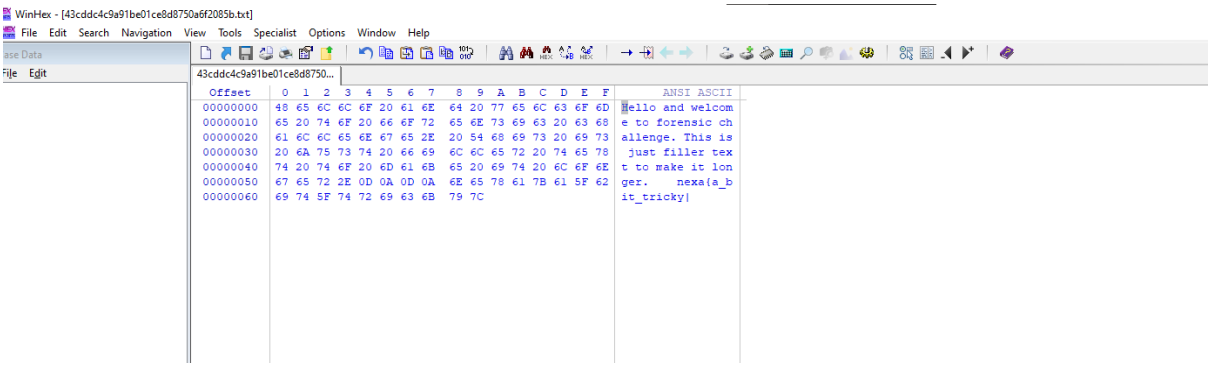


Copy the entire code and paste it in Rot Cipher
And it will show you rh flag which is
nexa{ROti_canai}

Shift your focus







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

MD5 decoder with the file name and it shows the flag
Which is **nexa{magic_number}**

Buffer Overflow

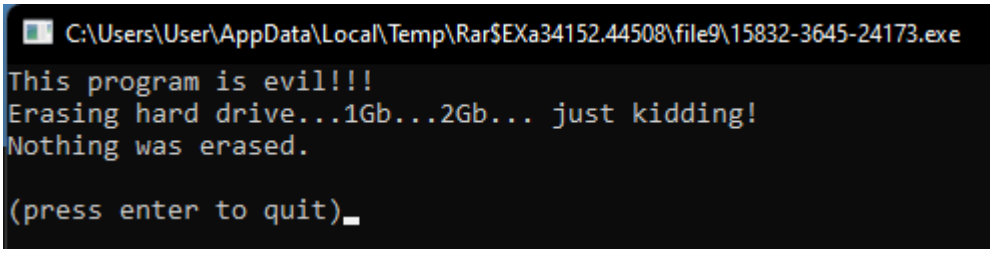
For this challenge, after running the exe file attached, we key in random strings that are long to trigger buffer overflow then the flag will appear.

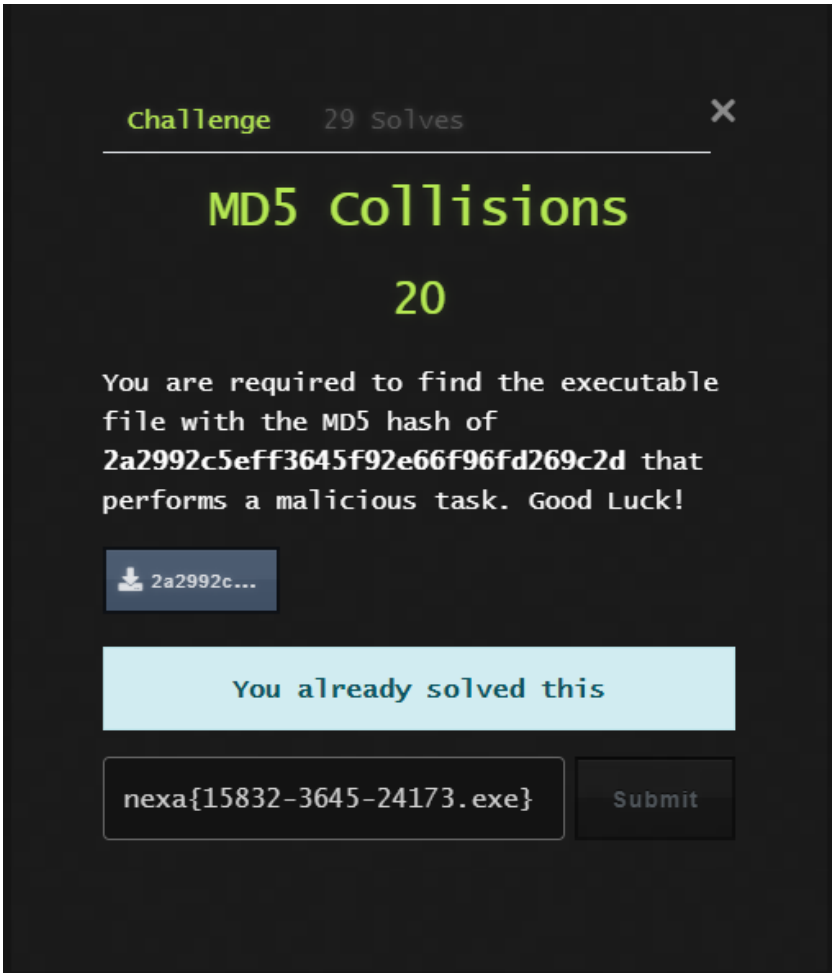
```
Enter the password :
sjrhg ywjhegfhu uguer h hergh r h&^&*^%*&%&^*%^&%%* 6846541561654984949 iwfiuhiruehewiehoieheorwhh*^(**(^&R&UG

Wrong Password

nexa{badcodefails}
```

MD5 Collisions

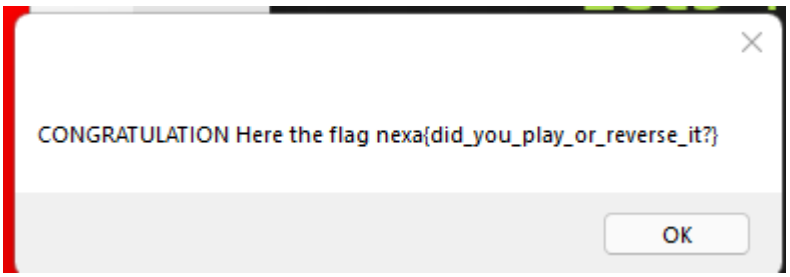
13046-26279-5127.exe	6,144	2,400	Application	5/30/2018 7:16 PM	745475B2
12596-3051-16070.exe	6,144	2,400	Application	5/30/2018 7:16 PM	745475B2
15832-3645-24173.exe	6,144	2,422	Application	5/30/2018 7:16 PM	A89052AE

The selected file has this output when executed



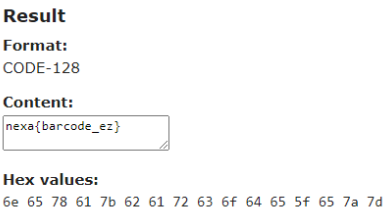


Lets Play



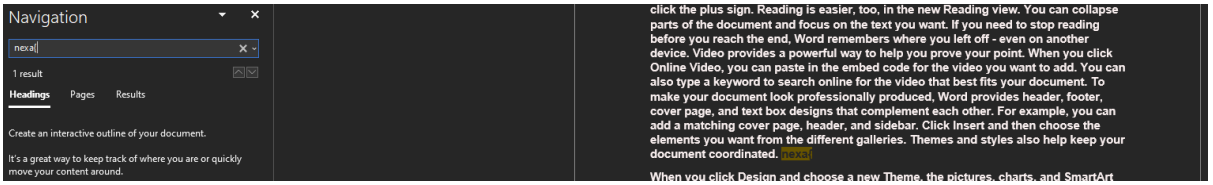
After completing the game, the flag is shown as above

Barcode I



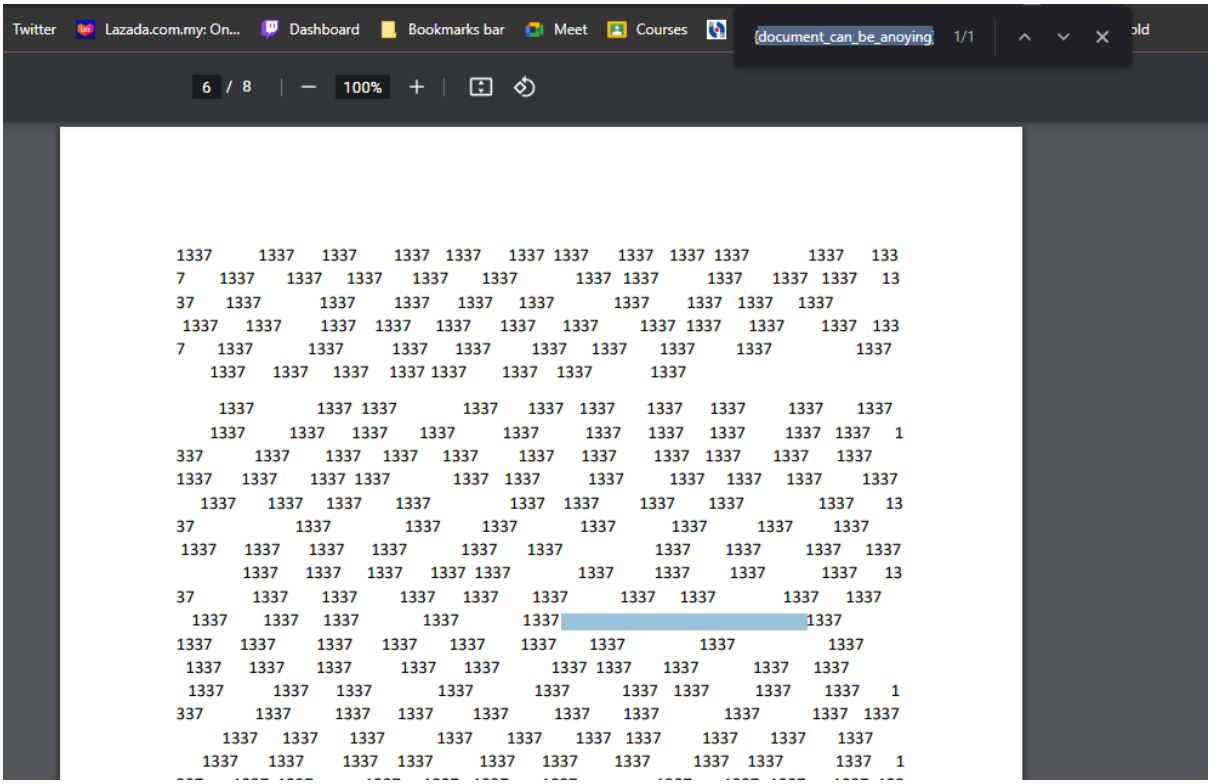
Using a barcode reader, it gave us the flag which is **nexa{barcode_ez}**

Document I



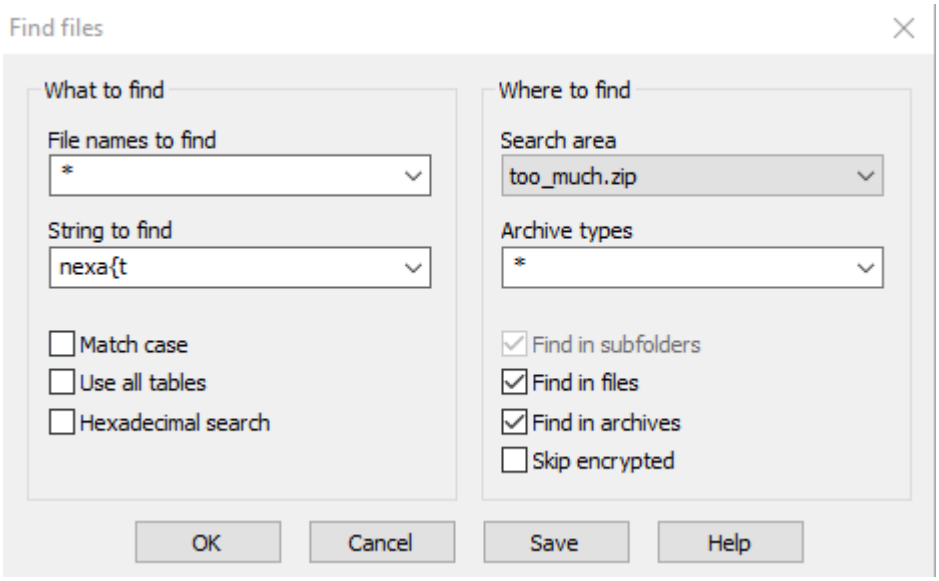
By ctrl F and typing nexa, it shows there is a hidden flag inside of it called **nexa{always_check_all}**

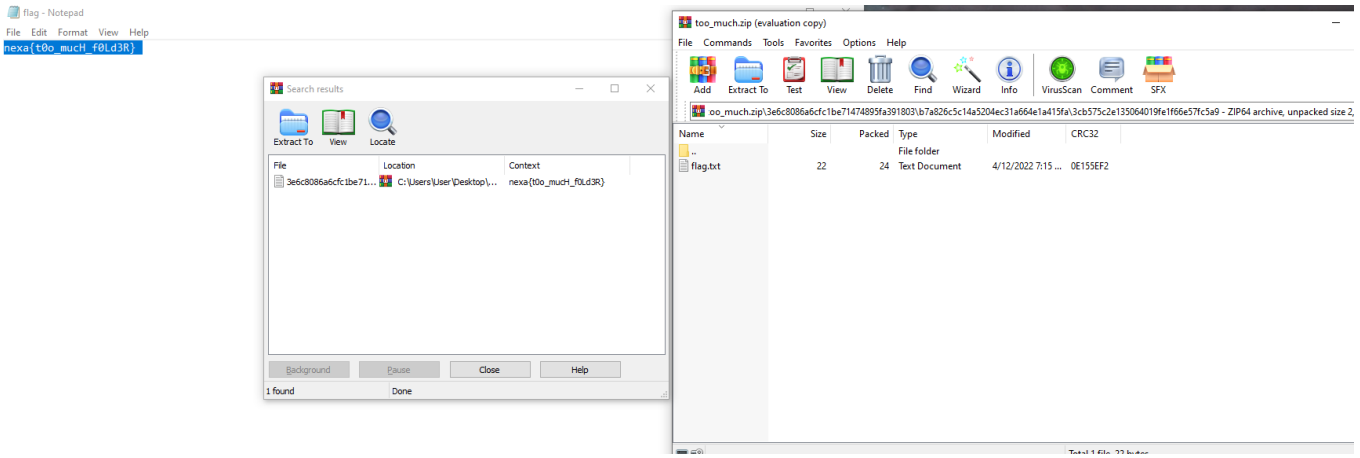
Document II



By brute forcing with ctrl F, you can guess **nexa{document_can_be_anoying}**

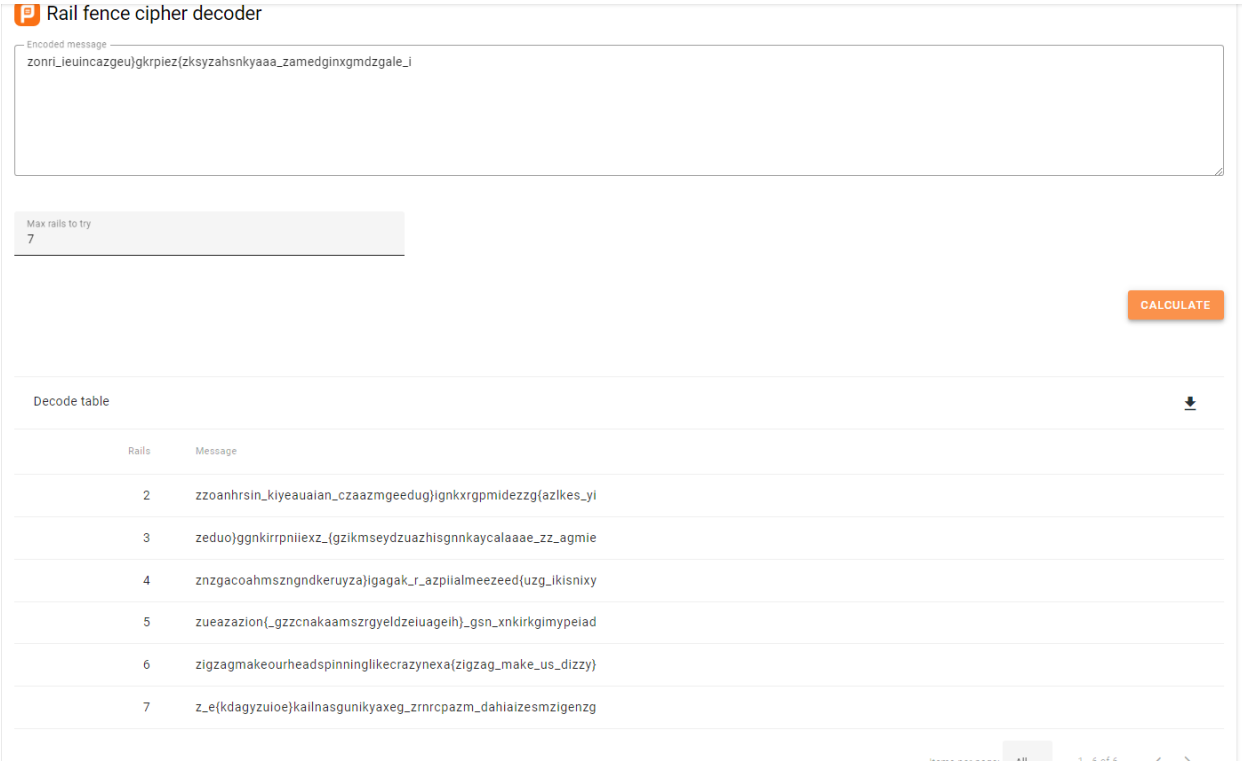
Too much





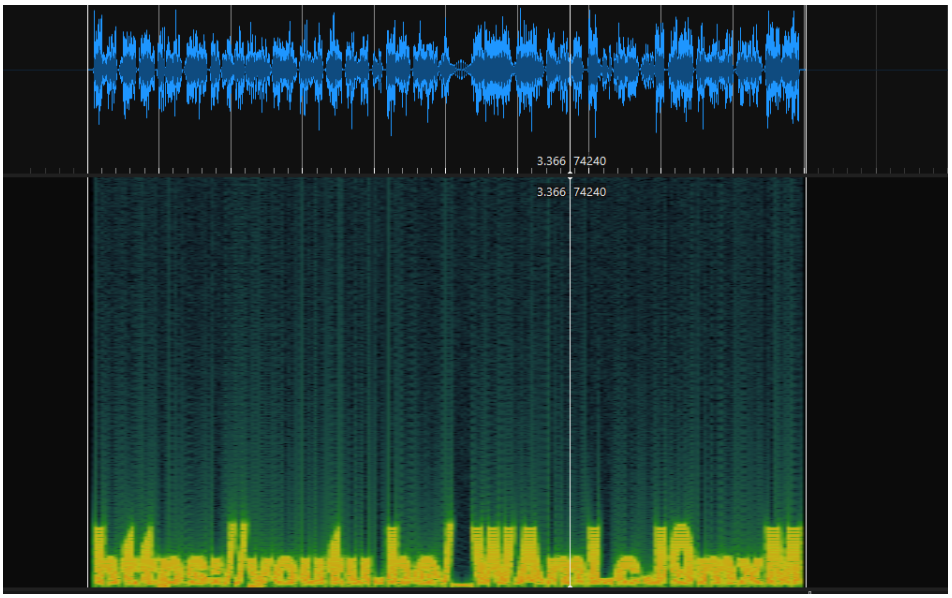
By brute forcing alphabets letters using the find filter, and knowing the flag doesn't have <> after nexa{} you can find the flag which is **nexa{t0o_mucH_f0Ld3R}**

Hard 1

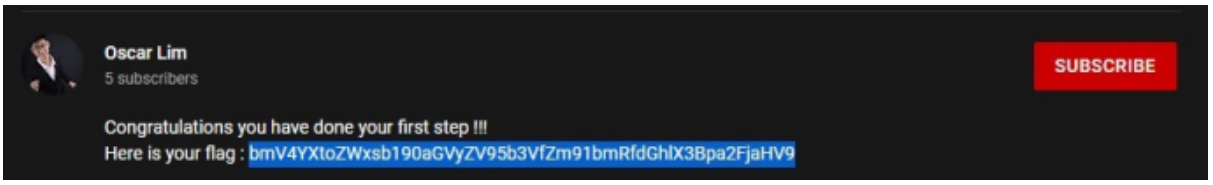


Using a website called https://planetcalc.com/6947/ we can set the amount of tries for it rail it to 7 or more till we reach the flag which is **nexa{zigzag_make_us_dizzy}**

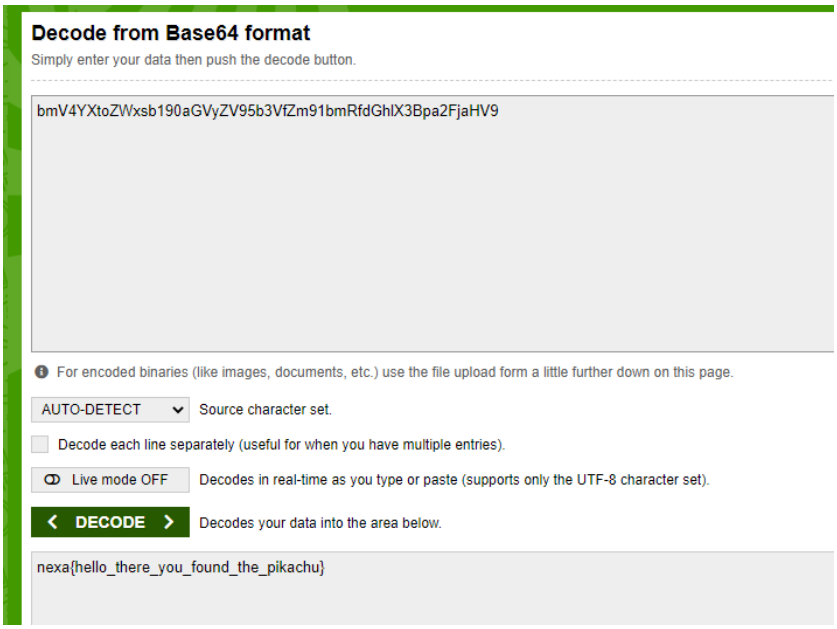
Who is that pokemon



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



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



The flag is **nexa{hello_there_you_found_the_pikachu}**