# Nexa x MMU CTF 2022 Part 1

## Author : yialexlee / w9u0l1.l2lvi

## Reverse Engineering

### Crack Me


Download the python file and open it. We found the encrypted flag

```
bezos_cc_secret = "E<O8R)&+@6:8E8@T"

# Reference alphabet
```

After look around the source code, we decode the encrypted flag becos_cc_secret.

```
        + str(greatest_value) )

decode_secret(bezos_cc_secret)
```

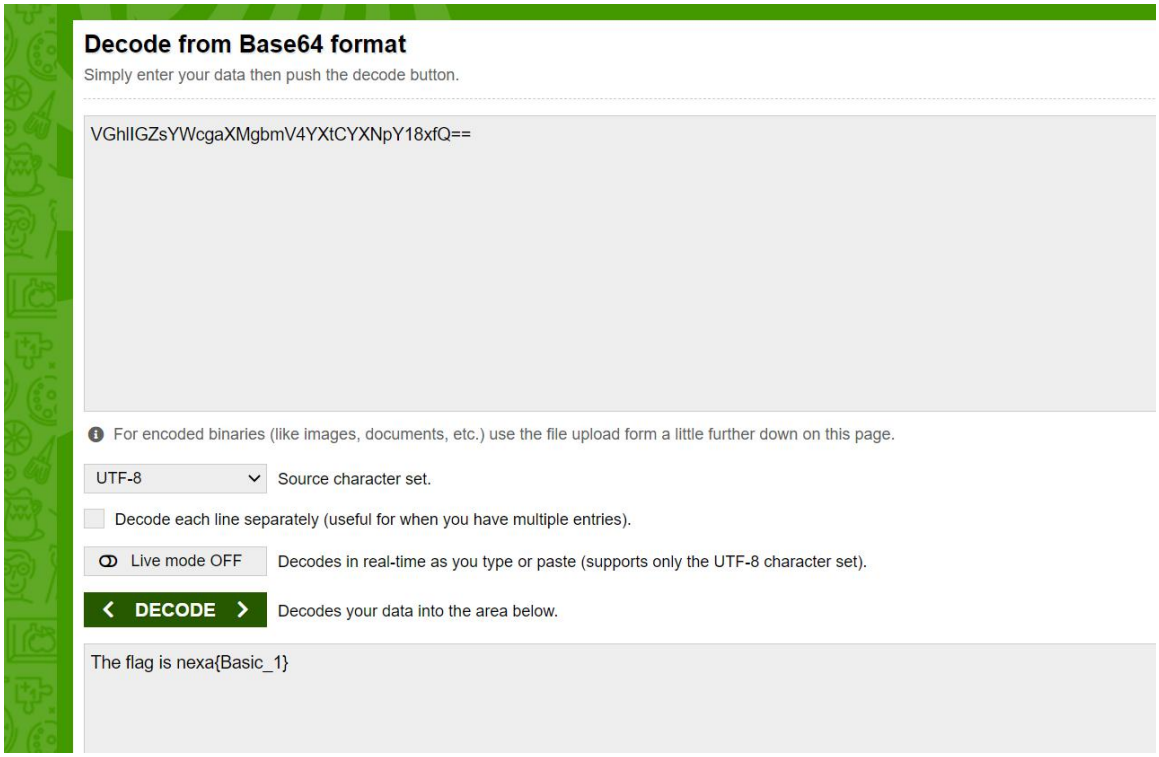
After run, we get the flag.

```
Install the latest PowerShell for new features and improveme

PS C:\Users\lee52\Downloads>  & 'C:\Users\lee52\AppData\Loca
-' 'c:\Users\lee52\Downloads\aa84f29754b12220ea353db9c2867fa
nexa{ROTi_canai}
PS C:\Users\lee52\Downloads>
```

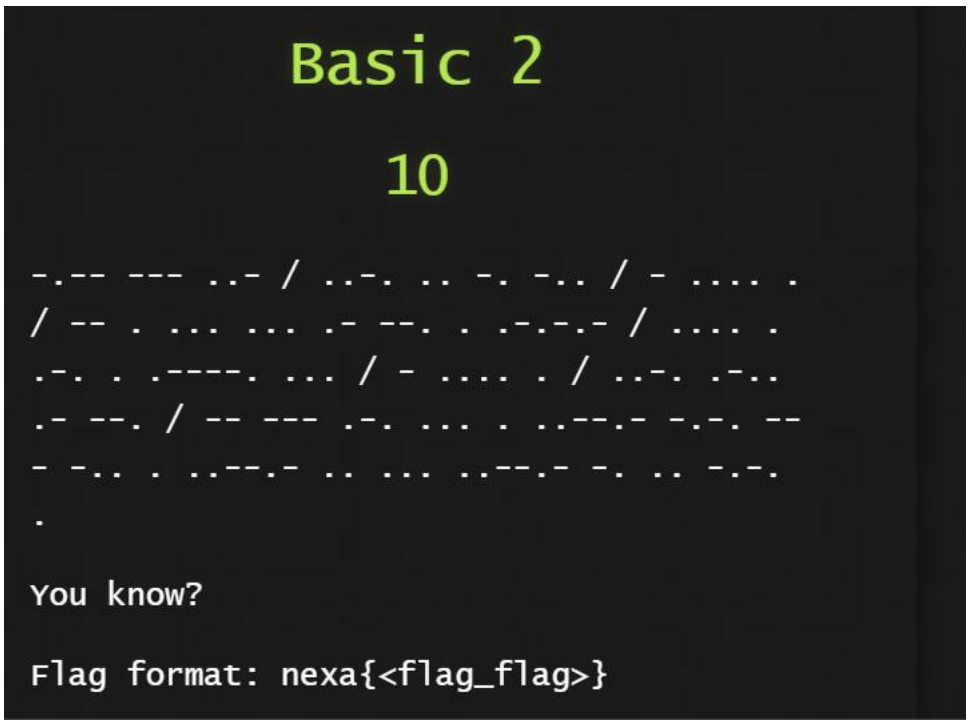
# Cryptography

## Basic 1


Its base64 encrypted, so just use any base64 decoder to decode



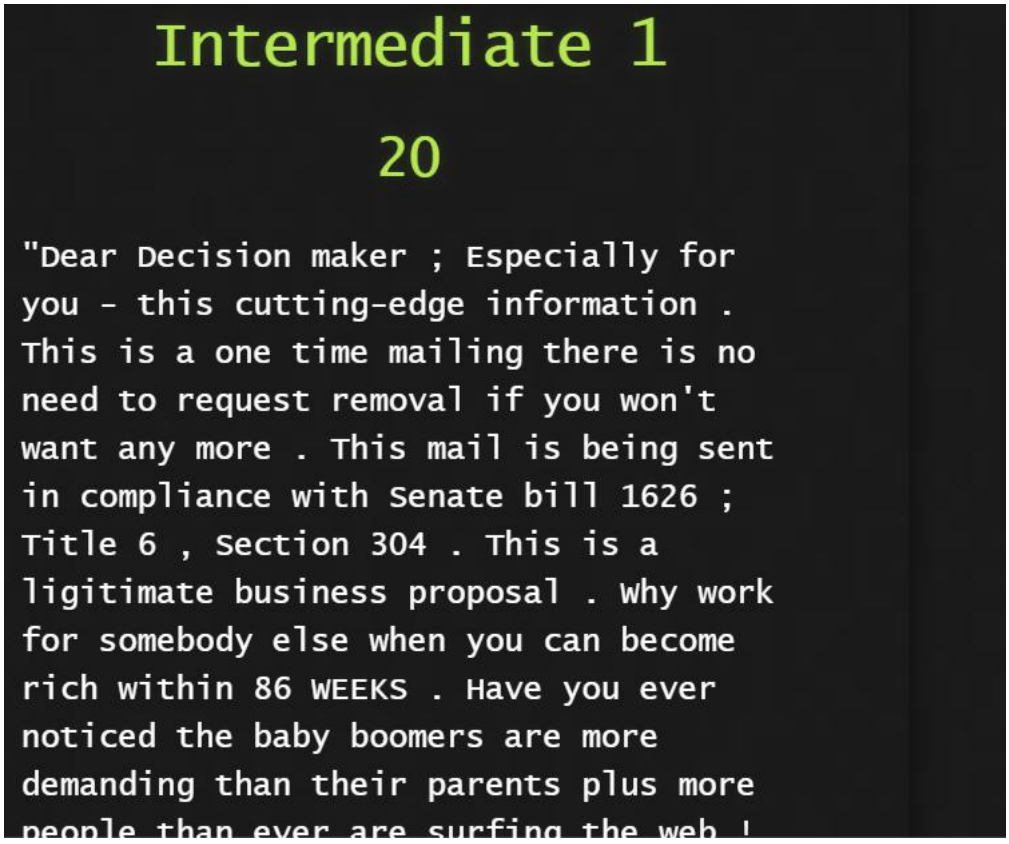
## Basic 2




It is a morse code, so we use morse code decoder to decode

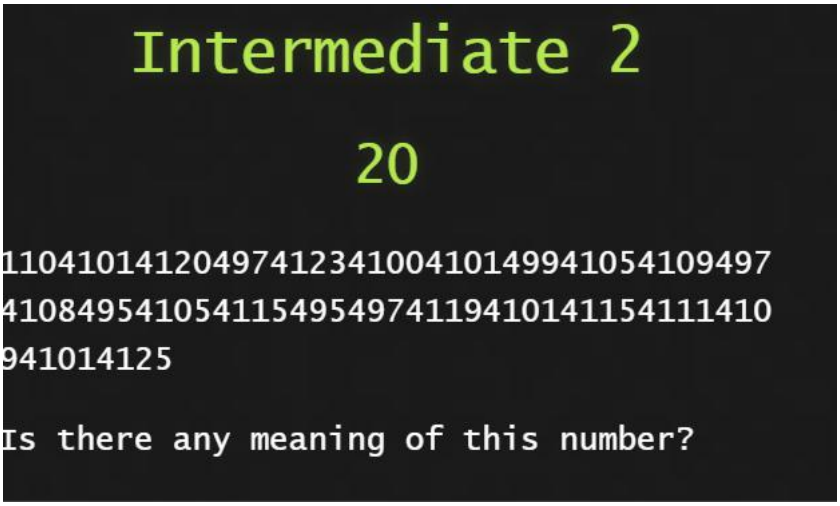

Nexa{MORSE_CODE_IS_NICE}

**Intermediate 1**



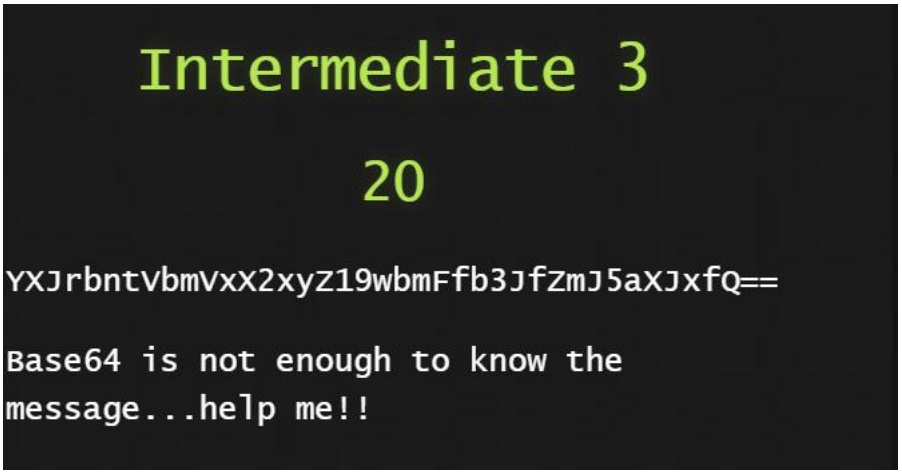

We use spamimic to decode. But remember to remove the “ ”

## Intermediate 2

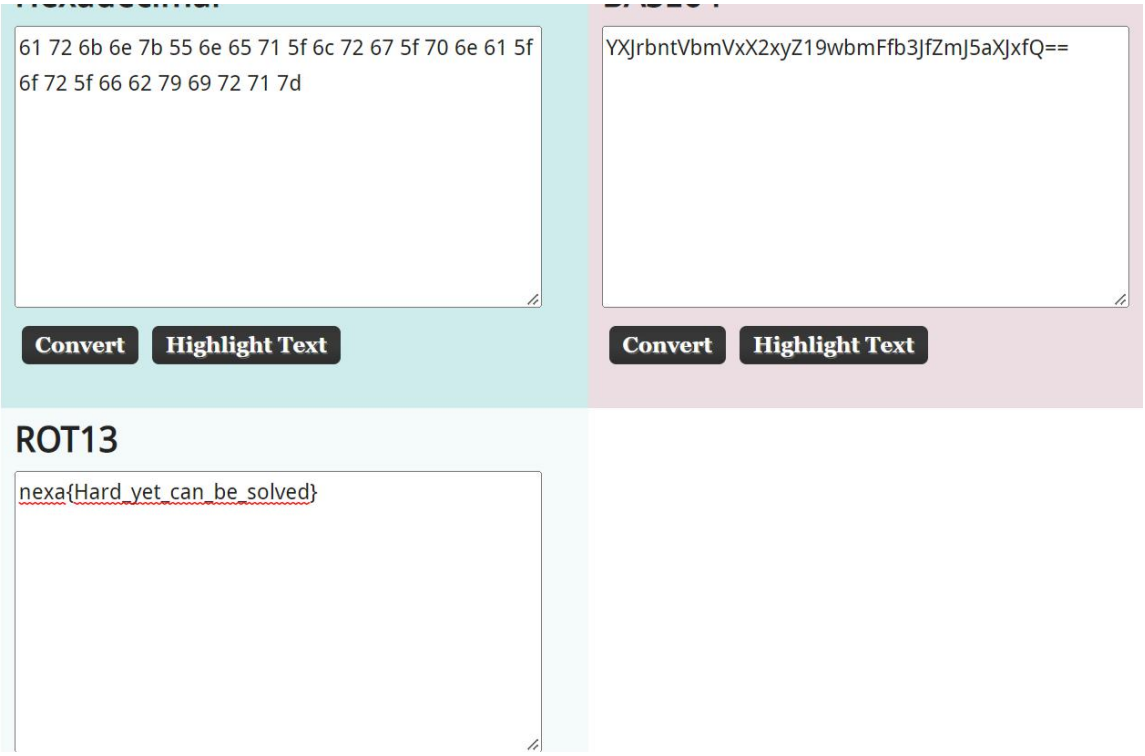



Remove the number 4 replace with space. Now it's a decimal, we convert decimal to text.

## Intermediate 3




We use convert tool to decode, and the flag is shown at ROT13



# Cracking

## Basic Wifi Cracking


Download the cap file and use aircrack-ng or hashcat to crack it. Here we use aircrack with wordlist (rockyou.txt) to crack the cap file

```
┌──(root💀kali)-[~]
└─# aircrack-ng /home/kali/Downloads/34cf7cd73063b5d98eed086b539dca96.cap -w /usr/share/wordlists/rockyou.txt
Reading packets, please wait...
Opening /home/kali/Downloads/34cf7cd73063b5d98eed086b539dca96.cap
Read 1033 packets.

   #  BSSID              ESSID                     Encryption

   1  6A:3E:26:EF:76:53  nexasecurenetworkkkk      WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening /home/kali/Downloads/34cf7cd73063b5d98eed086b539dca96.cap
Read 1033 packets.

1 potential targets
```

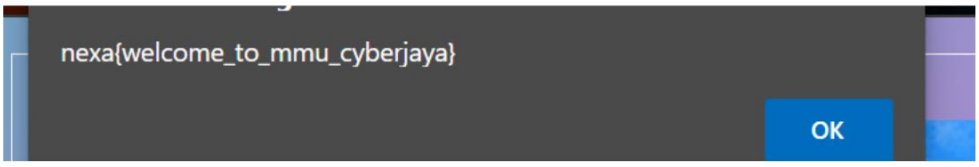
```
                      Aircrack-ng 1.6

[00:31:44] 9770559/14344392 keys tested (5052.16 k/s)

Time left: 15 minutes, 5 seconds                          68.11%

                KEY FOUND! [ betoeresmivida12 ]


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  : 70 86 5D 38 89 9E E9 12 93 2A 08 99 C5 BD 97 48
                 80 2F 64 21 1C 81 01 1D BE 69 85 AF B7 9F 25 E6
                 D9 49 04 C8 35 07 54 D5 16 27 88 87 52 E9 3F E2
                 96 95 9F 2C 38 C5 A2 49 77 06 9A 89 01 2D 89 00

EAPOL HMAC     : B7 82 C5 21 6F A8 1C 76 D4 B1 96 BD 2A CD 24 A3
```

And we get the password betoeresmivida12 so the flag is nexa{betoeresmivida12}

# Steganography

## Basic 5

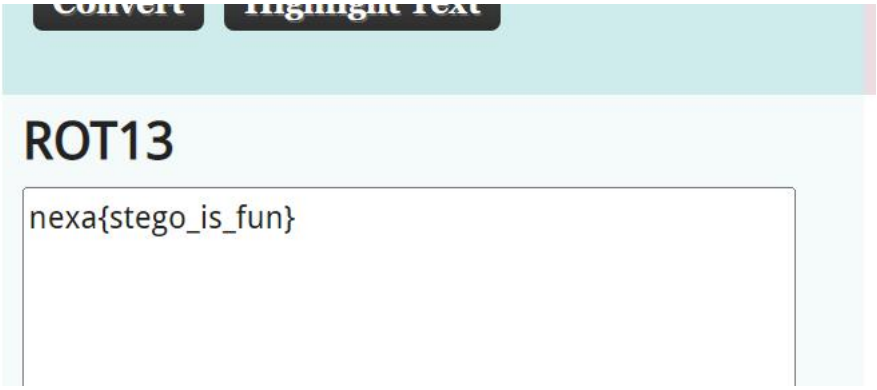

Download the image file and use hide text in image extension to solve it

## Intermediate 4


Download the image and use steghide to extract the  flag.txt file inside. Based on the hint given, we guess the pass is admin, and it really is.

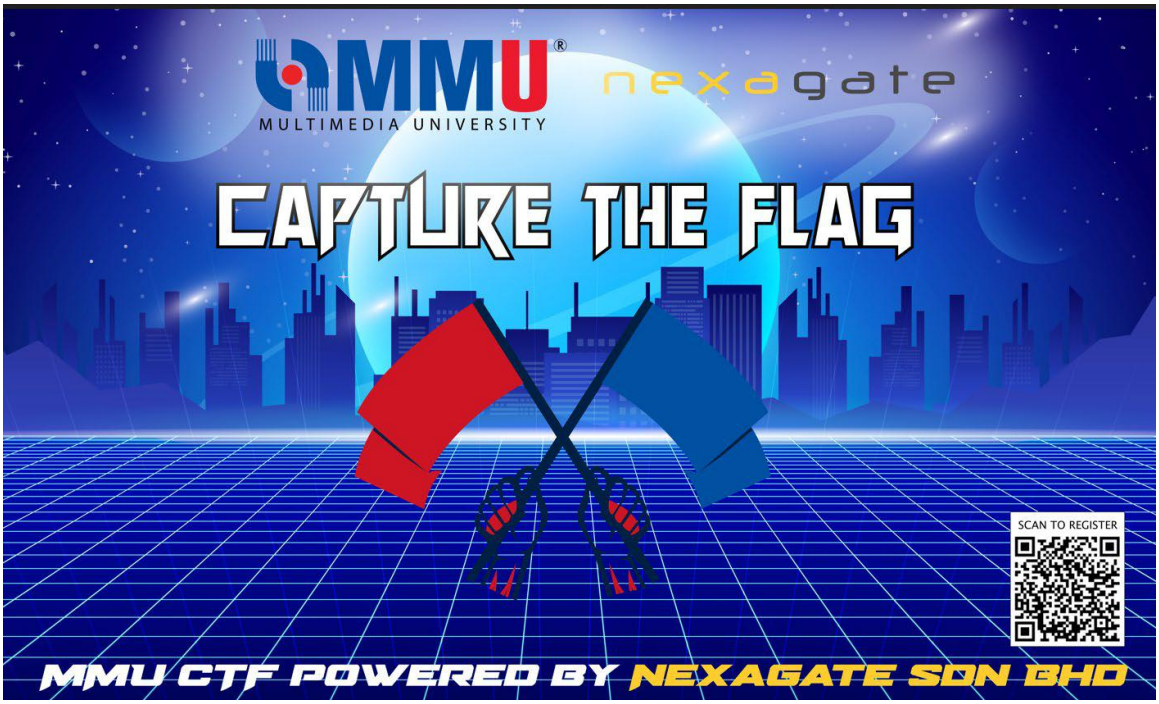

We get the fake flag. After some trying we found that this fake flag is encrypted version of real flag. We get the real flag after ROT13 decode



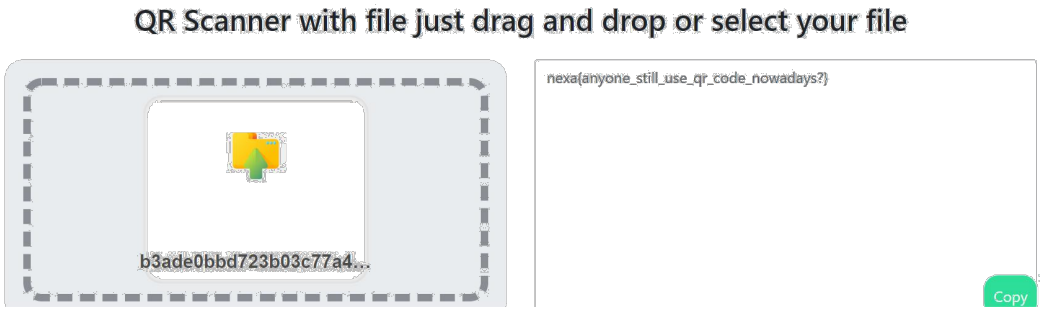
# MISC

## Challenge Poster


Download the poster

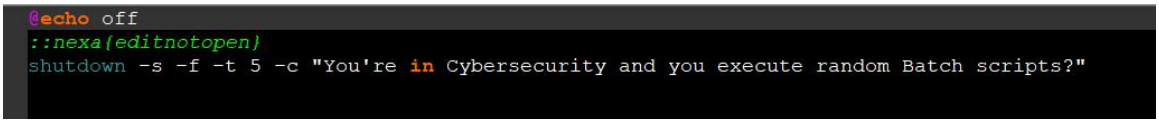



Scan the qrcode and get the flag



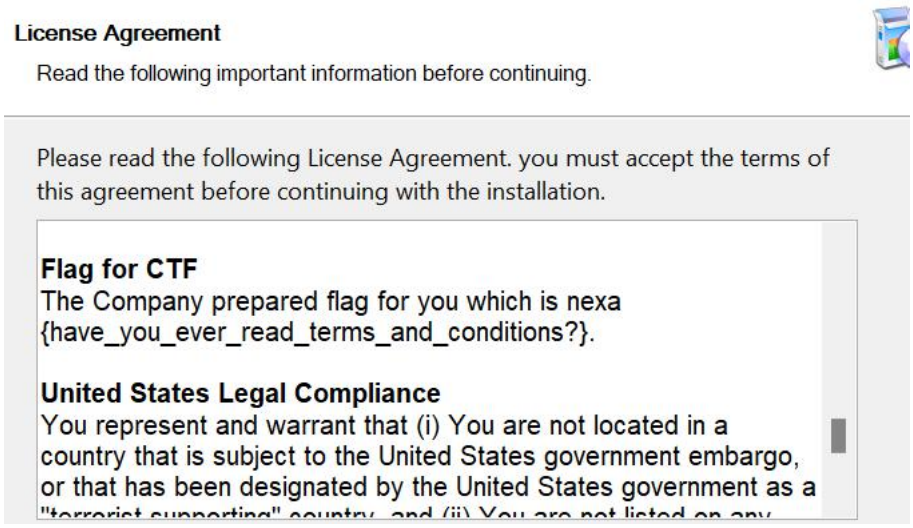

## Click for surprise


I really run the exe file at first…… Its shutdown my laptop. So I sure this is a bat that contains shutdown command. So this times I use notepad++ to open, and the flag is there.



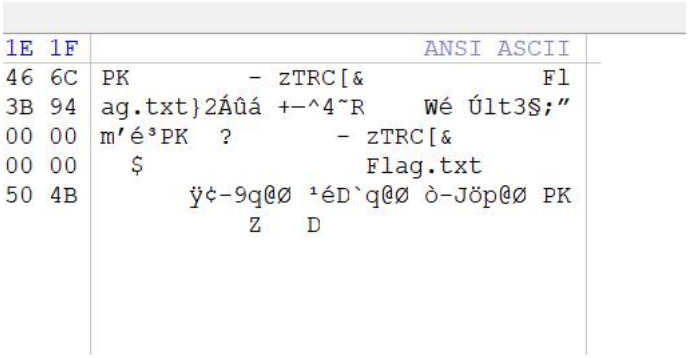
## h3x8d1mdkw8fncl


I forgot to screenshot, but I installed the application at first. After install there is a picture on my desktop, and nothing. So I looking for the directory of the application, there is a flag.html but it's a trap haha. After checking all the file in application directory and found nothing, I uninstall the application and reinstall again to make sure I did not miss something. And we found the flag in the process of installation, at the T&C part.  Ok, interesting. Next time I will read the T&C haha.

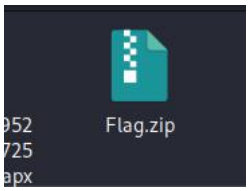


## Unknown file type


Use winhex to check the hex, we see PK and flag.txt inside. So we know this is a zip file and its contain a flag.txt inside. But the magic header is empty. So, we add the missing PK at the beginning.



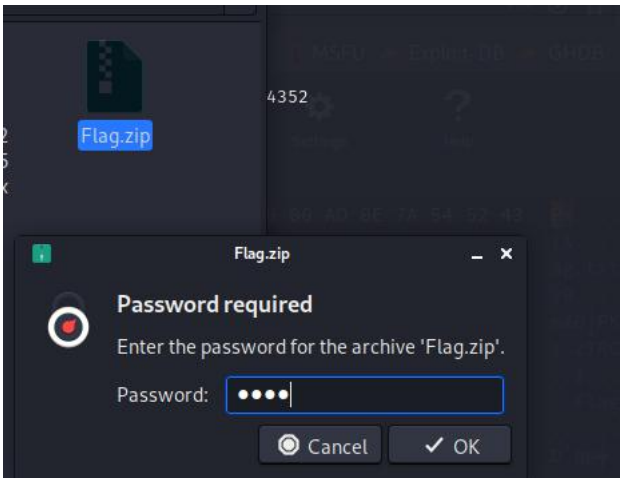

Now, the hex is correct now. So we change the file extension to .zip




We try to extract but it required password. So, we use john the ripper to crack it. The password is 1234

```
┌──(root💀kali)-[~]
└─# zip2john /home/kali/Downloads/Flag.zip > flag.txt
ver 2.0 Flag.zip/Flag.txt PKZIP Encr: cmplen=30, decmplen=18, crc=265B4352

┌──(root💀kali)-[~]
└─# john flag.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 7 candidates buffered for the current salt, minimum 8 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 4 candidates buffered for the current salt, minimum 8 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
1234             (Flag.zip/Flag.txt)
1g 0:00:00:00 DONE 2/3 (2022-03-26 10:19) 33.33g/s 1594Kp/s 1594Kc/s 1594KC/s 123456..ferrises
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

Extract the zip file with the password we cracked:1234



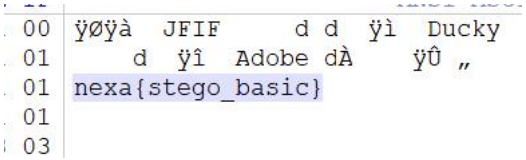

Open the flag.txt and we get the flag

```
nexa{magic_header}
```

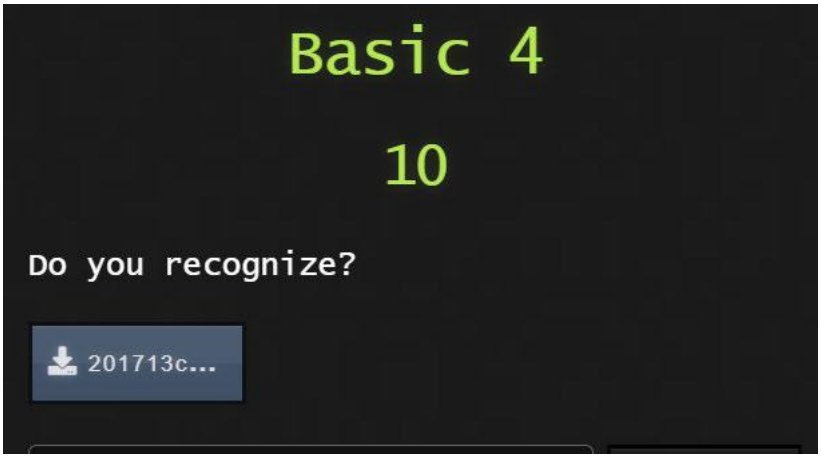
# Forensic

## Basic3

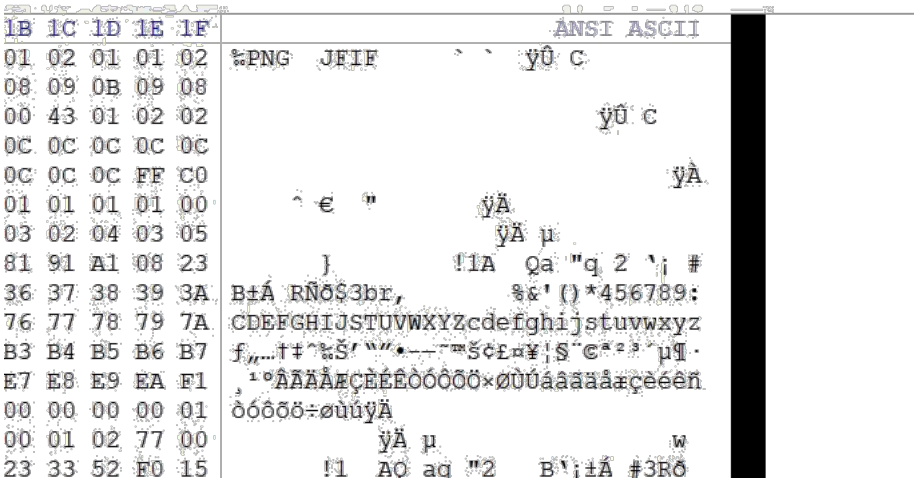

Use winhex to chack the hex and we can see the flag is already there




## Basic4

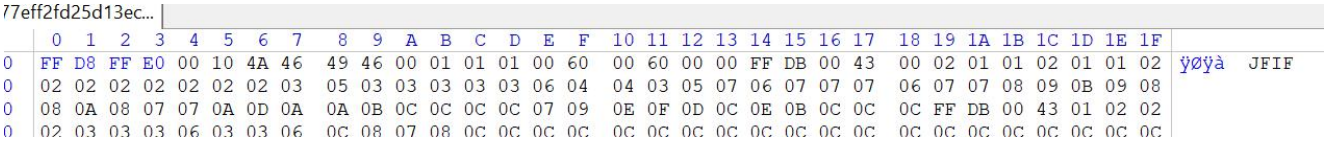



Same, we use winhex to check the hex first.



```
1B 1C 1D 1E 1F                                    ANSI ASCII
01 02 01 01 02  %PNG  JFIF      ` `   ÿÛ C
08 09 0B 09 08
00 43 01 02 02                          ÿÛ C
0C 0C 0C 0C 0C
0C 0C 0C FF C0                                  ÿÀ
01 01 01 01 00      ^ €  "        ÿÄ
03 02 04 03 05                    ÿÄ µ
81 91 A1 08 23        }          !1A  Qa "q 2 `¡ #
36 37 38 39 3A  B±Á RÑð$3br,          %&'()*456789:
76 77 78 79 7A  CDEFGHIJSTUVWXYZcdefghijstuvwxyz
B3 B4 B5 B6 B7  ƒ„…†‡^‰Š‹Œ••—˜™š¢£¤¥¦§¨©ª²³´µ¶·
E7 E8 E9 EA F1   ¹ºÂÃÄÅÆÇÈÉÊÒÓÔÕÖ×ØÙÚáâãäåæçèéêñ
00 00 00 00 01  òóôõö÷øùúÿÄ
00 01 02 77 00            ÿÄ µ                  w
23 33 52 F0 15          !1  AQ aq "2    B‘¡±Á #3Rð
```

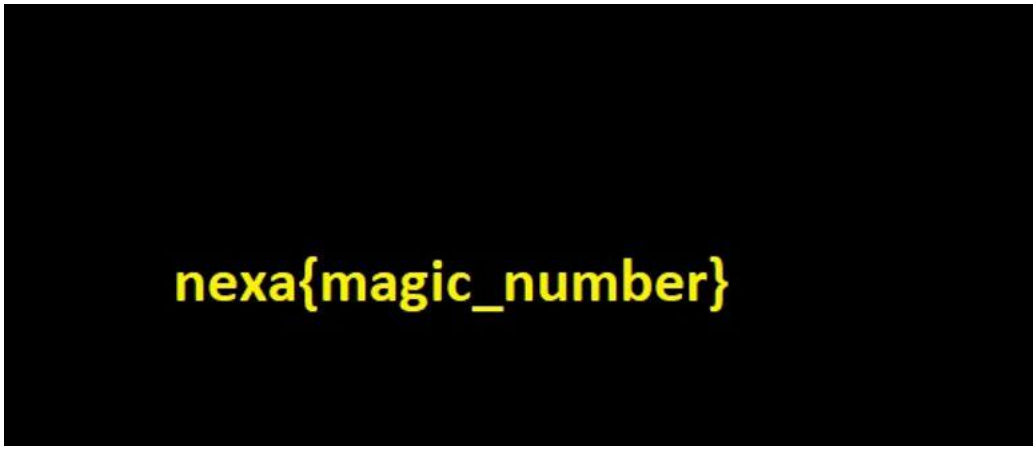
We can see the magic header is png , but there is a JFIF . So we guess that is a hint to tell us this file is not png, it's a jfif. So, we change the magic header to jfif magic header and save as jpg



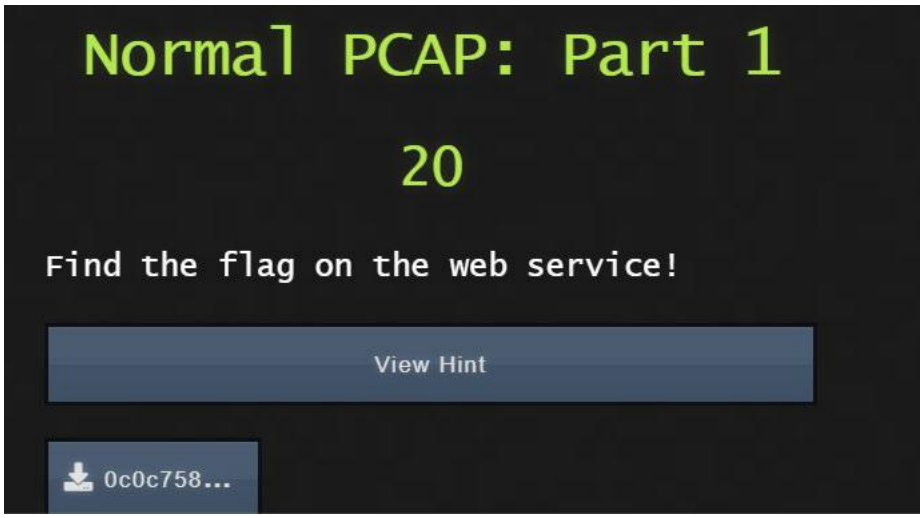
77eff2fd25d13ec...

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | FF | D8 | FF | E0 | 00 | 10 | 4A | 46 | 49 | 46 | 00 | 01 | 01 | 01 | 00 | 60 | 00 | 60 | 00 | 00 | FF | DB | 00 | 43 | 00 | 02 | 01 | 01 | 02 | 01 | 01 | 02 | ÿØÿà | JFIF |
| 0 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 03 | 05 | 03 | 03 | 03 | 03 | 03 | 06 | 04 | 04 | 03 | 05 | 07 | 06 | 07 | 07 | 07 | 06 | 07 | 07 | 08 | 09 | 0B | 09 | 08 | | |
| 0 | 08 | 0A | 08 | 07 | 07 | 0A | 0D | 0A | 0A | 0B | 0C | 0C | 0C | 0C | 07 | 09 | 0E | 0F | 0D | 0C | 0E | 0B | 0C | 0C | 0C | FF | DB | 00 | 43 | 01 | 02 | 02 | | |
| 0 | 02 | 03 | 03 | 03 | 06 | 03 | 03 | 06 | 0C | 08 | 07 | 08 | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | | |

Open the jpg and we get the flag



**Normal PCAP: Part 1**

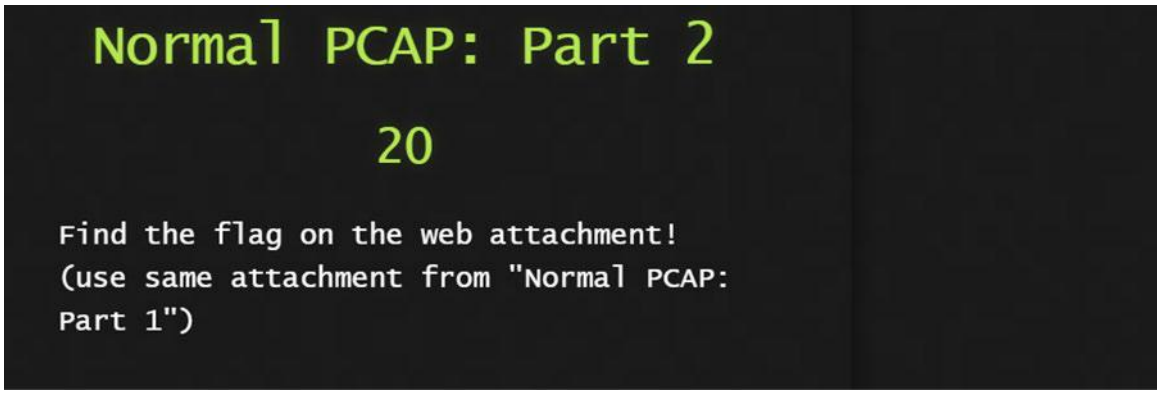



The hint mention it is a webserver, so we search http and https, and we found flag in one of the http packet

```
uery 0xa568 A incoming.telemetry.mozilla.org OPT
uery 0x3ab3 AAAA incoming.telemetry.mozilla.org OPT
n Data
n Data
```

```
on the page, just like this <strong>nexa{s1mpL3_w38_w1tH_l0v3}</strong> tag and its
```

## Normal PCAP: Part 2

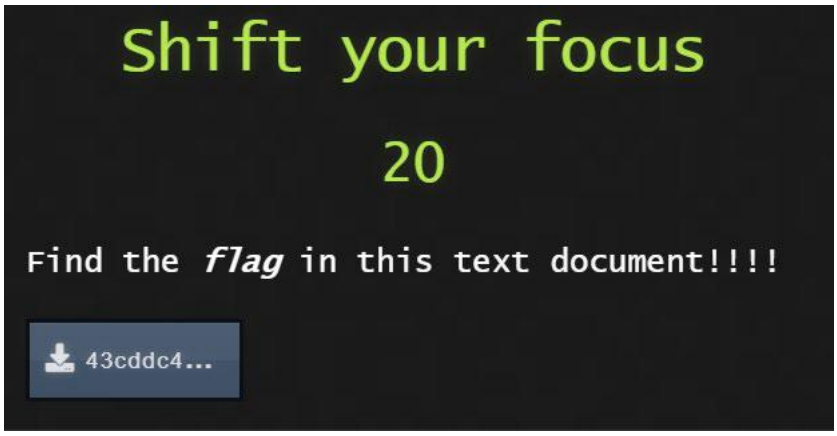



The hint mention it's a web attachment, so we find all the web attachment in this pcap file

| 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 | 10 MB | download.png |

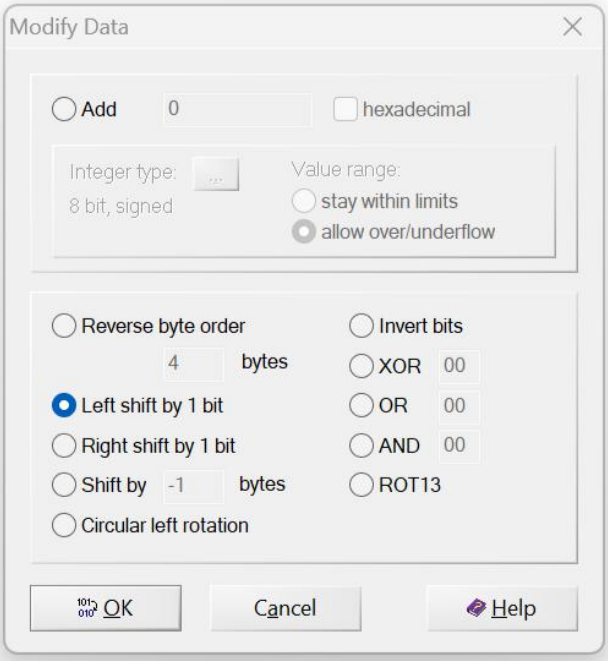
After looking for favicon.ico and download.png, we found flag in download.png

## Shift your focus




Use winhex to left shift by 1 bit

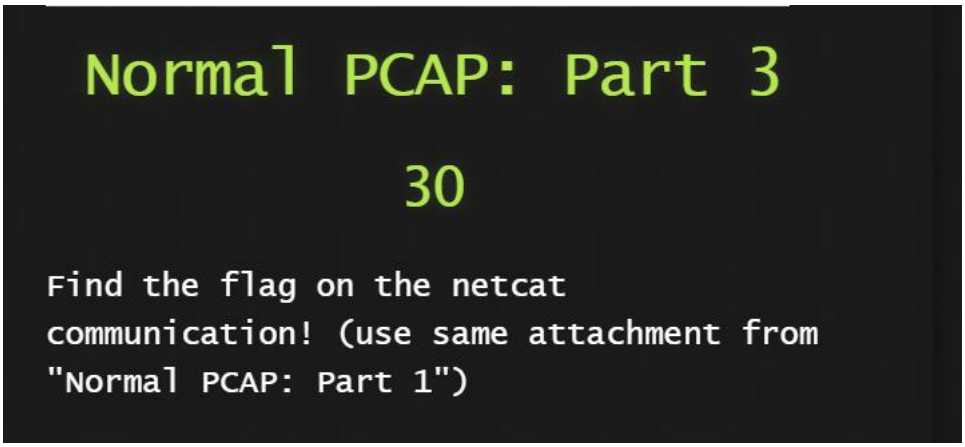
| C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F | AN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 31 | B7 | B6 | B2 | 90 | 3A | 37 | 90 | 33 | 37 | B9 | 32 | B7 | 39 | B4 | B1 | 90 | 31 | B4 | $2¶67 0·2 ;²¶1·¶² :7 37¹ |
| 9 | 90 | 34 | B9 | 90 | 35 | 3A | B9 | BA | 10 | 33 | 34 | B6 | 36 | 32 | B9 | 10 | 3A | 32 | BC | 0¶62·3²— *44¹ 4¹ 5:¹° 34 |
| 0 | 36 | 37 | B7 | 33 | B2 | B9 | 17 | 06 | 85 | 06 | 85 | 37 | 32 | BC | 30 | BD | B0 | AF | B1 | : :7 6°µ² 4° 67·3²¹ … … |
| | | | | | | | | | | | | | | | | | | | | 4°/°94±µ¼¾¼ |



And we can see the flag

```
Hello and welcome to forensic ch
allenge. This is just filler tex
t to make it longer.    nexa{a_b
it_tricky}
```

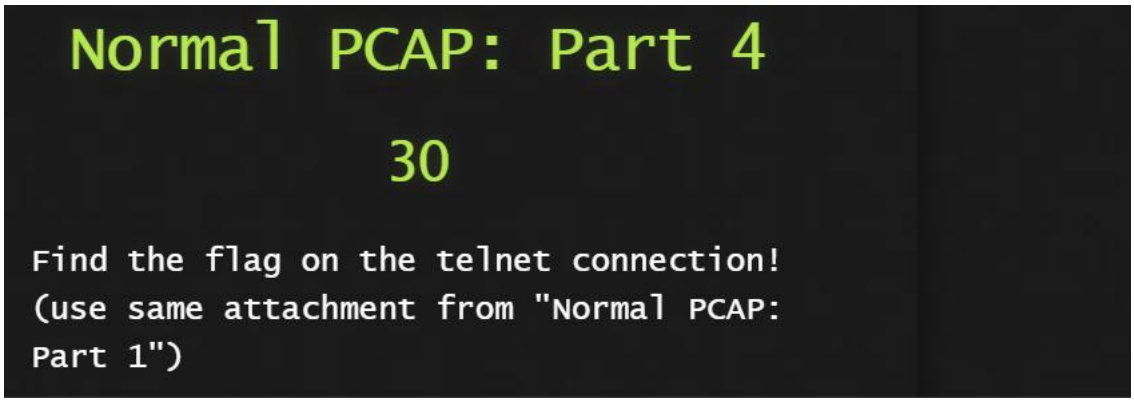
## Normal PCAP: Part 3




Hint mention the netcat, so we looking for icmp,tcp and udp. At the end we found flag in one of the tcp packet

```
..)..... ..%...-.
.·SURE.  IT nexa{
nc_i5_@_ p0w3rfuL
L_t00l}·
```

**Normal PCAP: Part 4**




Hint mentions the telnet, so we looking for telnet. And we found flag in one of the telnet packets

```
c0 a8   ··!)@·@· ········
80 18   ·····R·S ···"····
2d 1d   ··P=···· ··%·`&-··
77 33   ··nexa{t 3ln3t_w3
33 7d   r3_n3v3r _s3cuR3}
75 40   ···]0;ne xabuntu@
3b 33   ubuntu:  ~··[01;3
75 6e   2mnexabu ntu@ubun
34 6d   tu·[00m: ·[01;34m
```