

CTF c4ptur3th3fl4g

Kumpel7


December 2023


Those are the tasks i have to do, I am using the ubuntu machine with openvpn connection to tryhackme network.


1 Questions

[illegible]


Task 2








A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonographs, voiceprints, or voicegrams. When the data is represented in a 3D plot they may be called waterfalls.


 Download Task Files

Answer the questions below

Download the file

Answer format: *****

 Submit

 Hint

Task 3
Steganography
Download Task Files

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video.

Answer the questions below

Decode the image to reveal the answer.

Answer format: *****

Submit

Task 4
Security through obscurity
Download Task Files

Security through obscurity is the reliance in security engineering on the secrecy of the design or implementation as the main method of providing security for a system or component of a system.

Answer the questions below

Download and get 'inside' the file. What is the first filename & extension?

Answer format: *****

Submit Hint

Get inside the archive and inspect the file carefully. Find the hidden text.

Answer format: *****

Submit Hint

1st question is easy: Can you capture the flag? The second is in binary, so each octet gives one letter: "let's try some binary out!" is the answer. The 3rd one is in base32. That was a guess because of the equal signs – they make the whole sequence be divisible by 8 ($56/8=7$). The next one is base64 (Each Base64 digit represents exactly 6 bits of data. – this is the answer). The next one is hex – "hexadecimal or base16"? The next one is ceasar cipher with the key equal to 13, so the solution is " Rotate me 13 places!"

The next one was tough, as I had to write my own script to see it. This is like a ceasar cipher but in the whole ASCII region. Look at the answer:

```

dan@kumpel:~/temp$ python c.py
51
Dla i = 0
Dla i = 1
Dla i = 2
Dla i = 3
Dla i = 4
.DJ HE6C B2 G64SI GD3C9 767N G64SI GD3C9 SgJ I6B2HT
Dla i = 5
Dla i = 6
Dla i = 7
Dla i = 8
Dla i = 9
Dla i = 10
Dla i = 11
Dla i = 12
Dla i = 13
Dla i = 14
Dla i = 15
9OU SPIN ME RIGHT ROUND BABY RIGHT ROUND fru TIMESg
Dla i = 16
Dla i = 17
Dla i = 18
Dla i = 19
Dla i = 20
Dla i = 21
?U1 YVOT SK XOMNZ XU1TJ HGHS XOMNZ XU1TJ lxQ ZOSKYN
Dla i = 22
Dla i = 23
Dla i = 24
Dla i = 25
Dla i = 26
Dla i = 27
Dla i = 28
Dla i = 29
Dla i = 30
Dla i = 31
Dla i = 32
Dla i = 33
Dla i = 34
Dla i = 35
Dla i = 36
Dla i = 37
Dla i = 38
Dla i = 39
Dla i = 40
Dla i = 41
Dla i = 42
Dla i = 43
Dla i = 44
Dla i = 45
Dla i = 46
Dla i = 47
You spin me right round baby right round (47 times)
Dla i = 48
Dla i = 49

```

```

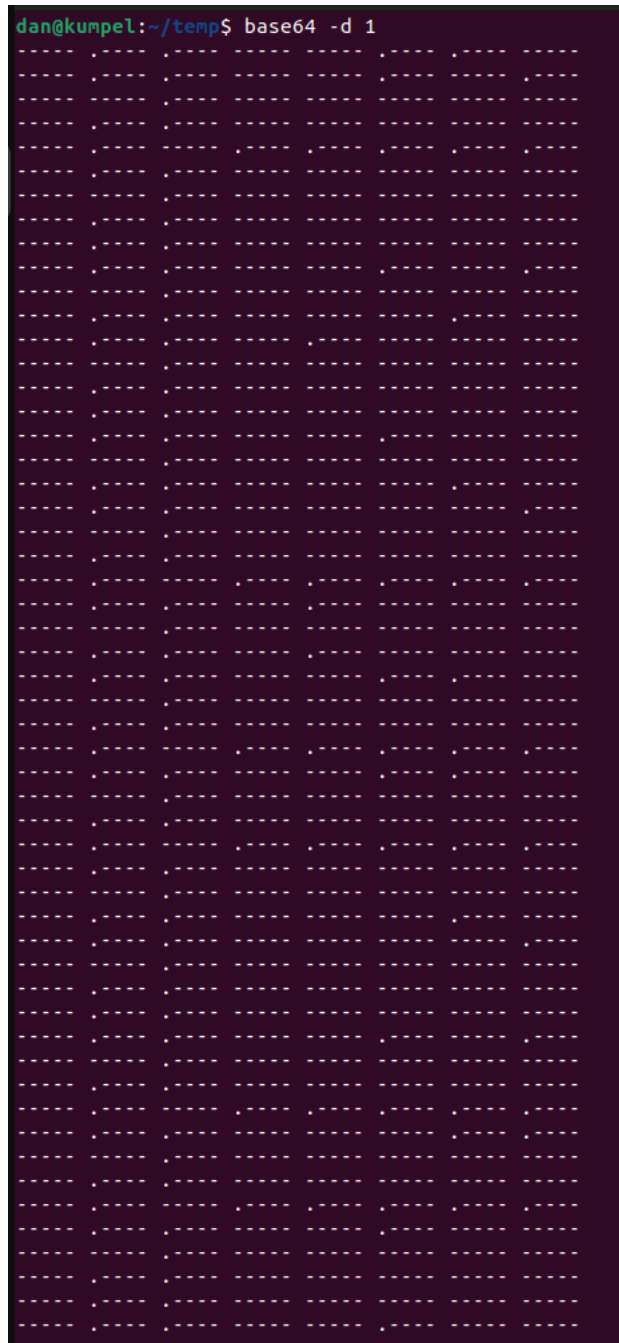
1 def checks(a):
2     a = str(a)
3     temp = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.!?1234567890()"
4     for s in temp:
5         if ord(s) == ord(a):
6             return True
7     return False
8
9 k = 0
10 t = '*@F DA:? >6 C:89E C@F?5 323J C:89E C@F?5 Wcf E:>6DX'
11 print(len(t))
12
13 for i in range(100):
14     j = 0
15     decoded_string = ""
16     print("Dla i = ", i)
17
18     for x in t:
19         if x != ' ':
20             decoded_char1 = chr((ord(x) + i) % 256)
21             decoded_char2 = chr((ord(x) - i) % 256)
22             if checks(decoded_char1):
23                 j += 1
24             decoded_string += decoded_char1
25             # print(x, ' -> ', decoded_char1)
26             elif checks(decoded_char2):
27                 j += 1
28             decoded_string += decoded_char2
29             # print(x, ' -> ', decoded_char2)
30         else:
31             decoded_string += ' '
32             j += 1
33
34     if j == len(t):
35         k = i
36         print(decoded_string)

```

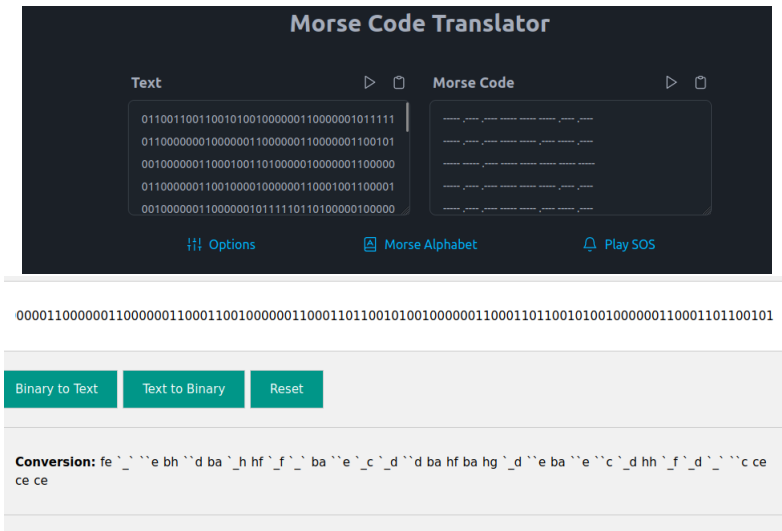
Screenshot 1: Using bruteforce i managed to solve this one

Now the next one is just morse code. TELECOMMUNICATION ENCODING is the solution. The next one is just ASCII code for "Unpack this BCD".

The last one looks tough, but is not. If we decode it in base64, we see that it is a morse code.



Screenshot 2: now just copy it and decode



Screenshot 3: Not the end yet :)

Now we have 2 or 3 symbols, that might suggest that we will encounter the numbers from 10 to 256, so I will modify my script c.py to seek for such combinations:



Screenshot 4: now just copy it and decode... again :). The answer: "Let's make this a bit trickier..."

2 Task 2

To solve it, one needs to use audacity and open the file in the spectrogram view.



3 Task 3

To solve it i used a tool online. This one I think is the best, as it just runs all the checks. Answer is in steghide.7z file. <https://www.aperisolve.com/735f86f28e424dc6a376b546be938633> (Spaghettesteg)

4 Task 4

To solve this task I will use binwalk to extract files encoded inside the .jpg file. Then, we will upload the hackerchat.png onto Aperi'Solve site. We will find our answer there.