Writeup for CGC Qualifiers 2022

By Jun Hao

Details

15-16 July 2022
Final standings: 8/17
Total points: 800

Overview

Here are the challenges that I solved

Challenge	Category	Value	Time
What's For Lunch?	OSINT	100	July 15th, 9:59:47 PM
ezTransposition	Crypto	100	July 15th, 11:30:16 PM
pwn-1	Pwn	100	July 15th, 11:34:08 PM
Homework	Forensics	100	July 16th, 1:58:59 PM
ezSubstitution	Crypto	100	July 16th, 2:19:07 PM
helloworld	RE	100	July 16th, 3:02:07 PM
Not Crypto	Crypto	100	July 16th, 3:24:40 PM
meow	Stego	100	July 16th, 4:36:30 PM

I feel like the time given for this CTF is quite tight and I didn't even have the chance to try every single challenge. Nonetheless, the challenges itself were mostly quite okay.

RE

helloworld

Value: 100

The first thing I did was to analyse it using radare and found that there was a function called printFlag

```
0x00001168      8b45fc          mov eax, dword [var_4h]
0x0000116b      4898            cdqe
0x0000116d      488d150c2f00.   lea rdx, qword obj.ct       ; 0x4080 ; "!|q#\x19\x11"
0x00001174      0fb63410        movzx esi, byte [rax + rdx]
0x00001178      8b4dfc          mov ecx, dword [var_4h]
0x0000117b      4863c1          movsxd rax, ecx
0x0000117e      4869c0abaaaa.   imul rax, rax, 0x2aaaaaab
0x00001185      48c1e820        shr rax, 0x20
0x00001189      d1f8            sar eax, 1
0x0000118b      89cf            mov edi, ecx
0x0000118d      c1ff1f          sar edi, 0x1f
0x00001190      29f8            sub eax, edi
0x00001192      89c2            mov edx, eax
0x00001194      89d0            mov eax, edx
0x00001196      01c0            add eax, eax
0x00001198      01d0            add eax, edx
0x0000119a      c1e002          shl eax, 2
0x0000119d      29c1            sub ecx, eax
0x0000119f      89ca            mov edx, ecx
0x000011a1      4863c2          movsxd rax, edx
0x000011a4      488d15b52e00.   lea rdx, qword obj.key      ; 0x4060 ; "g00dby3w0rld"
0x000011ab      0fb60410        movzx eax, byte [rax + rdx]
0x000011af      31c6            xor esi, eax
0x000011b1      89f1            mov ecx, esi
0x000011b3      8b45fc          mov eax, dword [var_4h]
0x000011b6      4863d0          movsxd rdx, eax
0x000011b9      488b45f0        mov rax, qword [var_10h]
0x000011bd      4801d0          add rax, rdx
0x000011c0      89ca            mov edx, ecx
0x000011c2      8810            mov byte [rax], dl
0x000011c4      8345fc01        add dword [var_4h], 1
; CODE XREF from sym.printFlag @ 0x1166
0x000011c8      8b45fc          mov eax, dword [var_4h]
0x000011cb      83f821          cmp eax, 0x21
0x000011ce      7698            jbe 0x1168
```

Seems like it's trying to XOR ct (which I assume is ciphertext) with key, so I tried to get ct in hex values as I realised some parts of ct is unprintable ascii values

```
0000 0000 0000 0067 3030 6462 7933 7730  .......g00dby3w0
726c 6400 0000 0000 0000 0000 0000 0000  rld.............
0000 0000 0000 0021 7c71 2319 1100 1b5c  .......!|q#....\
4233 1357 425c 005d 2659 2205 0633 0356  B3.WB\.]&Y"..3.V
5d5d 573d 1f5f 4357 0f00 0000 0000 00ff  ]]W=._CW........
```

With the values, I did the XOR and get the flag

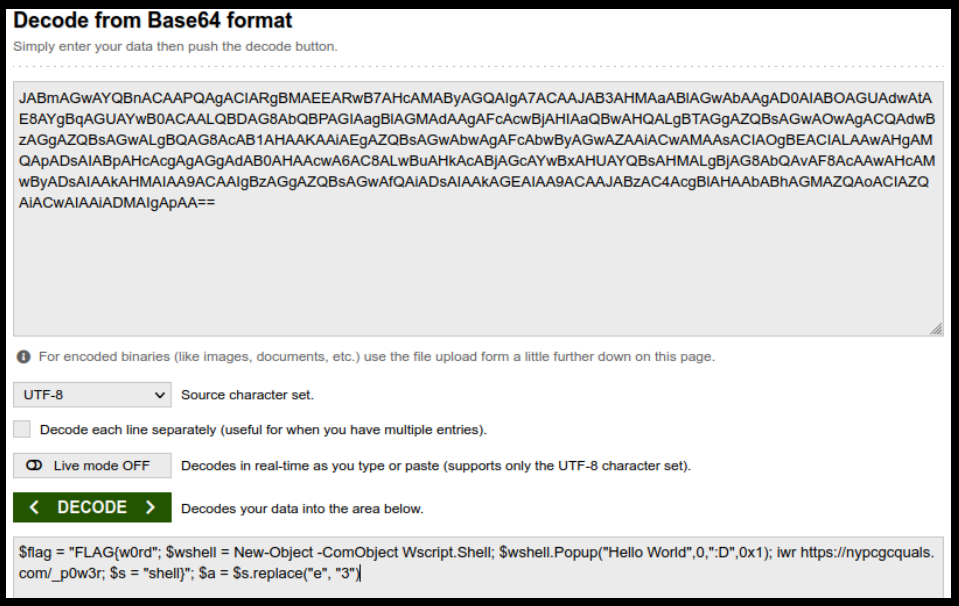
Forensics

Homework
Value: 100

With strings, you can see that there is a powershell script encoded in base64

```
powershell.exe -encoded JABmAGwAYQBnACAAPQAgACIARgBMAEEARwB7AHcAMAByAGQAIgA
IAagBlAGMAdAAgAFcAcwBjAHIAaQBwAHQALgBTAGgAZQBsAGwAOwAgACQAdwBzAGgAZQBsAGwAL
AHgAMQApADsAIABpAHcAcgAgAGgAdAB0AHAAcwA6AC8ALwBuAHkAcABjAGcAYwBxAHUAYQBsAHM
AkAGEAIAA9ACAAJABzAC4AcgBlAHAAbABhAGMAZQAoACIAZQAiACwAIAAiADMAIgApAA==
```

Upon decoding, you get the following script

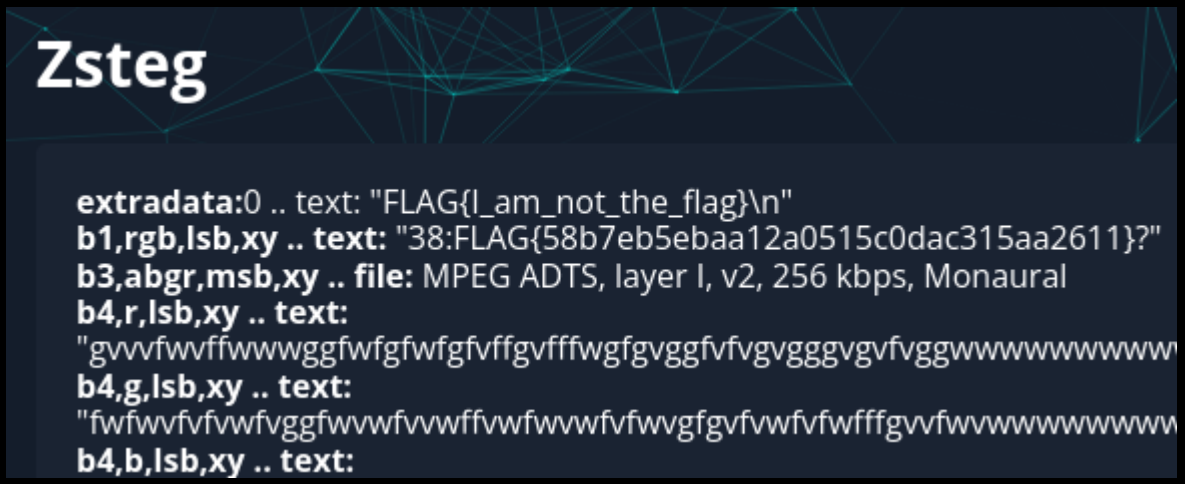


I reversed the powershell script and get the flag

Stego

meow
Value: 100

With the image given, I tried using various tools through aprisolve (Online tool)



When using Zsteg, the image returned with the flag

Crypto

Not Crypto
Value: 100

This challenge is quite straightforward, but the issue I had initially and I think others faced too is that this can only be solved with cyberchef.

```
RkxBR3s= ...-- -. -.-. ----- -.. `?80` 35 6e 74 5f 63 ⠒⠒⠒⠒ 137 142 162 60 175
FLAG{3NC0D1ng_15nt_cRYPT0_br0}
```

Decode each part of the string [base64][morse code][rot 47 - You can figure this out with dcode (Online tool)][hex][braille - Only with cyberchef][octal] to get the flag.