網路管理與系統管理 HW12

B13902022 賴昱錡

2025 年 5 月 28 日

2. 畫中有話

(a)

首先我們可以知道 pixels 是 secret_mygo.png 的像素資料，每一項都是一個長度為 3 的 tuple，分別代表紅、藍、綠的數值 (介於 0-255)，在讀入某個字串後 (data)，將每一個字元轉為 unicode，再轉為 8 位的 binary string。在處理 pixels 裡面的資料時，這份扣會把連續三格 pixel (pixels[3*i], pixels[3*i+1], pixels[3*i+2]) 變成一個長度為 9 的 list，之後再根據對應到 data 的第 i 個字元所轉成的字串 (方法如前述)，改變 list 中的數值 (一個 binary string 有 8 項，也對應到 list 的前八項)，最終再將 list 裡面的數值寫回圖片。

(b)

為了找到藏在圖片裡面的 string，我們可以用 3 個像素為單位來解析，很容易可以發現到，如果 (a) 所述的 binary string 的一項是"1"，則 colors 對應到的一項數值必為奇數，反之則為偶數，所以我們可以跑過足夠多的像素，然後找到很多 binary string 及其對應的 unicode and 字元，用其來拼湊出 flag。

在這裡我使用 python 來找出 flag，程式碼如下，最終得到的結果是：(btw 在那之前可能要安裝 PIL module)

`HW12{S4KiCh4n_sakiCHAN_S4k1ChaN}`

參考資料是我自己，好耶，這是少數完全不用仰賴 GPT 解出的題目。

```python
from PIL import Image

image_file = "secret_mygo.png"
img = Image.open(image_file)
pixels = list(img.getdata())
flag = ""

for i in range(0, 100, 3):
    colors = list(pixels[i])+list(pixels[i+1])+list(pixels[i+2])
```

```
    tmp = ""
    for j in range(8):
        if colors[j] % 2 == 0:
            tmp = tmp + "0"
        else:
            tmp = tmp + "1"
    flag = flag+chr(int(tmp, 2))

print(flag)
```

4. Introduction to gnireenignE esreveR

這題的 flag 是:

HW12{hW0_8UT_WiTH_r3V3Rse_eN91NE3rinG}

我拿到 flag 的方法是把 chal.exe 丟進 dogbolt.org，然後發現 Hex-Rays 這個 decompiler 竟然幫我分析好原本程式的架構了。基本上，原本的程式包含以下重要變數:

```
char key[9] = "nAs42O2S";
char pattern[40] =
  {'&','\x16','B','\x06','I','\'','e','c','1','y','&','`','m','\x18'
,'[','\a','&','\x1E','\x01','\a','d','|','`','
  ','\v','\x1E','\x16','z','\v','~','|',
'\x16',']','3','\x1A','Z','u','2','\0','\0'};
int flag_len = 38;
```

看到程式接受輸入的地方 (應該是 main)，他會用 key 對 pattern 做 xor encryption，然後 check 程式接受的參數是不是等於加密後的 pattern。

```
int __fastcall main(int argc,
  const char ** argv,
    const char ** envp) {
  int i; // [rsp+1Ch] [rbp-4h]

  if (argc == 2) {
    for (i = 0; i < flag_len; ++i)
      pattern[i] ^= key[i % key_len];
    if (!strcmp(argv[1], pattern))
      puts("Congratulations! You found the flag!");
    else
      puts("Haha! wrong >:)!!!!!!");
```

```
13      return 0;
14    } else {
15      puts("Usage: ./chal.exe <flag>");
16      return 1;
17    }
18  }
```

所以我們可以用前面提到的變數實際去 xor 一遍，就可以得到程式中的 pattern/flag。(將 flag 作為 chal.exe 的參數執行，可以得到代表正確的訊息)

```
1  for (int i = 0; i < flag_len; ++i ) pattern[i] ^= key[i % key_len];
2  printf("%s\n", pattern);
```