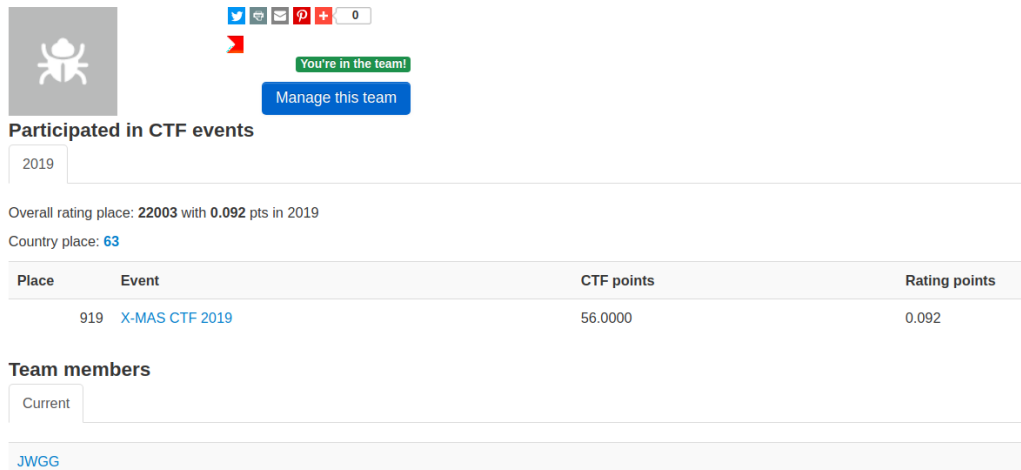


X-mas CTF write up

王俊翔 Nickname:JWGG ID:b06902006

Team : JWGGGGGGG

🇹🇼 JWGGGGGGG



The image shows a CTF profile for the team JWGGGGGGG. It includes a profile picture of a spider, social media icons, and a 'You're in the team!' message. Below this, it lists 'Participated in CTF events' for the year 2019, showing an overall rating place of 22003 and a country place of 63. A table lists the team's performance in the X-MAS CTF 2019, with a place of 919, 56,000 CTF points, and 0.092 rating points. The 'Team members' section lists the current member JWGG.

Place	Event	CTF points	Rating points
919	X-MAS CTF 2019	56.0000	0.092

1. Sequel Fun

FLAG : X-MAS{S0_1_c4n_b3_4dmin_w1th0ut_7h3_p4ssw0rd?}

這題是簡單的輸入帳密的 sql injection，嘗試了一些輸入，發現帳號打 admin' # 即可成功 exploit 得到 flag。

2. Forensics (忘記題目名字)

FLAG : X-MAS{W3lc0m3_t0_th3_N0rth_P0l3}

下載的是一個 png 檔

commend file png 檔 發現其實是 zip 檔

解壓縮後，打開裡面的檔案就有 flag 了

-----再來是看別人的 write up 寫的，因為當初沒什麼時間打-----

3. SNOWVERFLOW

這題看標題就知道要做 buffer overflow，而且沒有開 canary，reverse 後發現有個 function call 之後就會印出 flag，因此 overflow 到 return address，然後直接輸入 function 的位置(0x401156)，即可 leak 出 flag

FLAG : X-MAS{700_much_5n0000w}

4. Santa's crackme

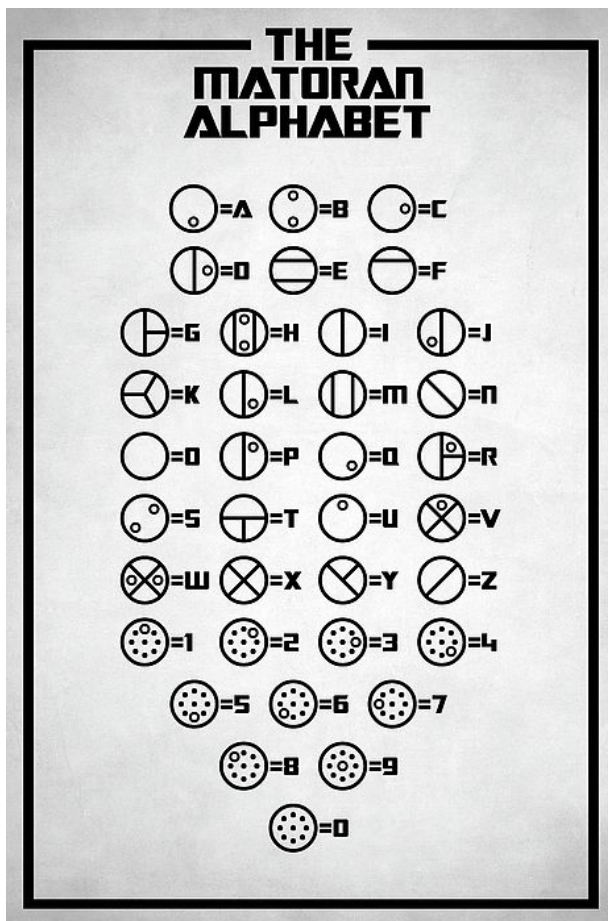
這題是 reverse 題，IDA 開下去查看裡面的 code，發現輸入的東西它每一個 char 會去 ^ 3 然後跟 flag_matrix + 1 * (目前位置，第一個是 0) 的位置中的 char 比，如果都一樣就過關，最後即可得出 flag。

FLAG : X-MAS{54n74_15_b4d_47_l1c3n53_ch3ck1n6}

5. Mata Nui's Cookies

這題是 crypto，當初想了很久，不知道是啥符號對應啥字母，結果發現有下面的圖，答案就出來了…

FLAG : X-MAS{MATANUIHASPREPAREDTHECOOKIES}



6. Bobi's wHack

這題超鳥，去看它 youtube 的 information 欄就找到 flag 了，我甚至花了很多時間找他的資料。

FLAG : X-MAS{subscribelikesharethanks-bobi:)}

7. Discount VMProtect

這題是我這次看的最難的一題，這題要你輸入 password 然後看對不對，然後是一個 reverse 題，以下是解法：

1. 用 gdb 把程式開起來，發現沒有 main 函式，進去 entry point 發現 main address 其實是在 400c88 (mov rdi, 0x400c88)。
2. main 函式主要是將 user input 存在 0x6026a0，然後如果 0x6025a0 的第一個 byte 是 \x01 就可以得到 flag，此外，它 call 了兩次 0x400857。
3. 0x400857：用 ida 之類的看會發現它的偽代碼是 return，很明顯其實不是，所以用 gdb 看了一下發現這個函式清了一個 0x30，然後在 rbp-0x28 的地方放 rdi，然後在 rbp-0x4 及 rbp-0x8 的地方放 0，最後 push 了一個 0x236c1a 再 xor 0x636465，然後這其實會是最後 return 會到的地方，也就是 0x40087f。
4. 0x40087f：這是最重要的地方，裡面再做的其實就是標題所說的 VM protected binary，VM protected binary 有自己的<instructions>，也就是有一個 function 會解讀這些 instructions，且有自己的 memory，所以這個 0x40087f 其實就是那個 function，然後它會做一系列的操作去 execute 這些 instructions，所以問題就變成：

(1) 這個 binary 從哪抓 instruction？

(2) 每個 opcode 在幹麻？

(3) FLAG 應該就藏在這吧？

4-1. 關鍵就在剛剛的那 0x30 個 memory 裡，可以發現在 opcode 中一直用到 rbp-0x8 等等這些值，所以必須回推 rdi 到底是啥，這部份用 gdb 看一下就知道了，然後就知道這個 function 中所使用的 instructions 到底是啥。

4-2. 這個用 IDA 之類的看就會懂了，可是非常的冗，所以這裡不綴述（我會附我看到的 reference），有幾個重要的，像是 0x30 是 break loop（不再讀 instructions），0x62 是檢查有沒有在 debug mode 還有檢查 0x602580 李的值是否等於 0x1b，如果有就 print NOOOOOOOO!，否則就將其加一等等....

4-3 綜合以上幾點其實就可以得出滿足條件的 flag 了。

reference：

<https://altelus1.github.io/writeups/xmasctf2019/discountvmprotect>

<https://resources.infosecinstitute.com/reverse-engineering-virtual-machine-protected-binaries/#gref>

FLAG : X-MAS{VMs_ar3_c00l_aNd_1nt3resting}