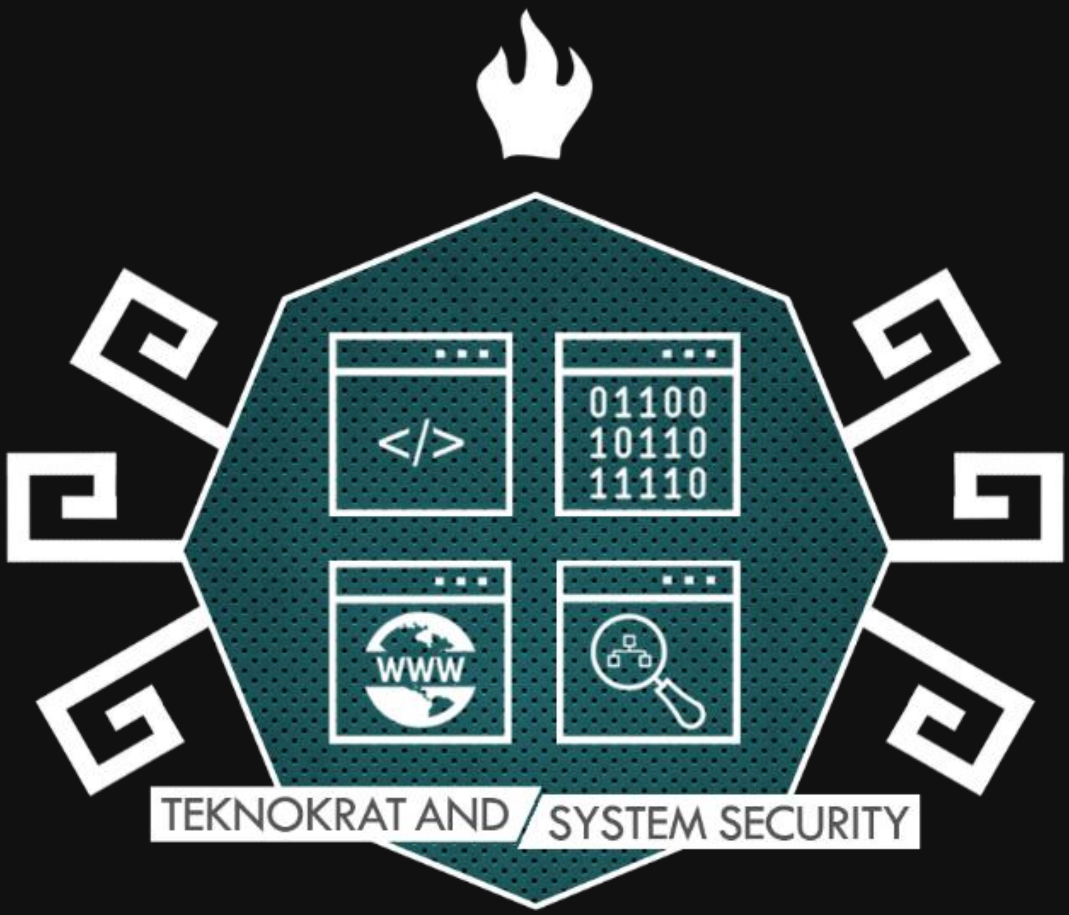
TENESYS


01100
10110
11110
WWW
TEKNOKRAT AND
SYSTEM SECURITY

CR1: Ultracoded [50 Points]

Soal

Fady didn't understand well the difference between encryption and encoding, so instead of encrypting some secret message to pass to his friend, he encoded it!
Hint: Fady's encoding doens't handly any special character
Link : https://ctf.oddcoder.com/files/512924e9369008c4d734fcccdc472367/zero_one

Solusi

- Soal berupa string ZERO ONE yang kita semua tau bahwa itu adalah binary, kita convert menggunakan script python berikut

```
#binary
import binascii

strings = '''
#insert strings here
'''

binary = ''
split = strings.split(" ")
for word in split:
     if "ONE" in word:
          binary += '1'
     elif "ZERO" in word:
          binary += '0'
print binascii.unhexlify("%x" % int(binary, 2))
```

- Setelah diconvert didapat string base64

```
Li0gLi0uLiAuIC0uLi0gLS4tLiAtIC4uLS4gLSAuLi4uIC4tLS0tIC4uLi4uIC0
tLSAuLS0tLSAuLi4gLS0tIC4uLi4uIC4uLSAuLS0uIC4uLi0tIC4tLiAtLS0gLi
4uLi4gLiAtLi0uIC4tLiAuLi4tLSAtIC0tLSAtIC0uLi0gLQ==
```

- Setelah didecode didapat sebuah sandi

```
.- .-.. . -..- -.-. - ..-. - .... .---- ..... --- .---- ... ---
..... ..- .--. ...-- .-. --- ..... . -.-. .-. ...-- - --- - -..-
-
```

- Sandi tersebut adalah sandi morse, mari kita convert kedalam bahasa manusia, Didapat string berikut

```
alexctfth15o1so5up3ro5ecr3totxt
```

Kami berasumsi bahwa o artinya _ (underscore), dan karena format flag ALEXCTF{[A-Za-z0-9_]*}, didapatlah flagnya **ALEXCTF{th15_1s_5up3r_5ecr3t_txt}**

TR4: Doesn't our logo look cool ? [40 Points]

Solusi

Logo yang dimaksud adalah logo di home https://ctf.oddcoder.com/ kami buat mirror berikut http://pastebin.com/PJyrTCGS
Terdapat huruf A, L, E, X, disini kita sudah dapat menebak bahwa didalam logo ini terdapat flag, mari hilangkan # ` @ + . ; dll, dalu didapat string berikut

```
ALEXCTF{0UR_L0G0_R0CKS}
```

Didapatlah flagnya **ALEXCTF{0UR_L0G0_R0CKS}**

RE1: Gifted [50 Points]

Soal

Soal berupa file ELF 32-bit LSB executable

Solusi

Cari string flag pada file soal dengan perintah ***strings gifted | grep ‘AlexCTF’***

Munculah string flag
AlexCTF{Y0u_h4v3_45t0n15h1ng_futur3_1n_r3v3r5ing}

RE2: C++ is awesome [100 Points]

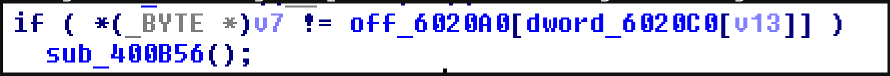
Soal

They say C++ is complex, prove them wrong!

Soal berupa file ELF 64-bit LSB executable

Solusi

- Buka file soal dengan IDA Pro 64 bit
- Pada fungsi **sub_400B89,** ada kondisi yang sepertinya membandingkan string inputan user dengan flag



```
if ( *(_BYTE *)v7 != off_6020A0[dword_6020C0[v13]] )
  sub_400B56();
```

- Dimana v7 adalah inputan user dan off_6020A0[dword_6020C0[v13]] adalah string flag

```
#!/usr/bin/python

data1=['24','5','36','65','7','27','26','2D','1','3','0D','56',
'1','3','65','3','2D','16','2','15','3','65','29','44','44','1'
,'44','2B']
data2="L3t_ME_T3ll_Y0u_S0m3th1ng_1mp0rtant_A_{FL4G}_W0nt_b3_3X4
ctly_th4t_345y_t0_c4ptur3_H0wev3r_1T_w1ll_b3_C00l_1F_Y0u_g0t_1t
"

flag = ""
for i in range(len(data1)):
     flag += data2[int(data1[i],16)]
print flag
```

```
AEXCTF{W3_0v3_C_W1th_C45535}
```

Flag yang didapat masih kurang huruf L

ALEXCTF{W3_L0v3_C_W1th_CL45535}

RE4: unVM me [250 Points]

Soal

If I tell you what version of python I used .. where is the fun in that?

Soal berupa file python 2.7 byte-compiled

Solusi

- Decompile file soal dengan Easy Python Decompiler

Hasil Decompiler file soal

```
import md5
md5s = [174282896860968005525213562254350376167L,
 137092044126081477479435678296496849608L,
 126300127609096051658061491018211963916L,
 314989972419727999226545215739316729360L,
 256525866025901597224592941642385934114L,
 115141138810151571209618282728408211053L,
 8705973470942652577929336993839061582L,
 256697681645515528548061291580728800189L,
 39818552652170274340851144295913091599L,
 65313561977812018046200997898904313350L,
 230909080238053318105407334248228870753L,
 196125799557195268866757688147870815374L,
 74874145132345503095307276614727915885L]
print 'Can you turn me back to python ? ...'
flag = raw_input('well as you wish.. what is the flag: ')
if len(flag) > 69:
    print 'nice try'
    exit()
if len(flag) % 5 != 0:
    print 'nice try'
    exit()
for i in range(0, len(flag), 5):
    s = flag[i:i + 5]
    if int('0x' + md5.new(s).hexdigest(), 16) != md5s[i / 5]:
        print 'nice try'
        exit()

print 'Congratz now you have the flag'
```

Dari source code diatas dapat diambis kesimpulan bahwa:

- Panjang flag tidak lebih dari 69 karakter.
- Panjang flag adalah kelipatan lima.
- Flag dipecah menjadi 5 karakter dan hasil encrypt md5nya diubah ke dalam bentuk integer kemudian disimpan didalam array.

Penyelesaian:
Mancari string md5 di database web decrypt md5, semua bagian berhasil ditemukan kecuali bagian ke 7. Untuk itu terpaksa menggunakan metode bruteforce.

```
#!/usr/bin/python

import md5

def md2int(a):
     return int(md5.new(a).hexdigest(),16)

md5s = [174282896860968005525213562254350376167L,
 137092044126081477479435678296496849608L,
 126300127609096051658061491018211963916L,
 314989972419727999226545215739316729360L,
 256525866025901597224592941642385934114L,
 115141138810151571209618282728408211053L,
 8705973470942652577929336993839061582L,
 256697681645515528548061291580728800189L,
 39818552652170274340851144295913091599L,
 65313561977812018046200997898904313350L,
 230909080238053318105407334248228870753L,
 196125799557195268866757688147870815374L,
 74874145132345503095307276614727915885L]

 huruf = “abcdefghijklmnopqrstuvwxyz1234567890”

#Bruteforce bagian index ke 6
for bru1 in huruf:
     print bru1
     for bru2 in huruf:
          for bru3 in huruf:
               for bru4 in huruf:
                    for bru5 in huruf:
                         part = bru1+bru2+bru3+bru4+bru5
                         if str(md5s[6]) in str(md2int(part)):
                              print part
```

Setelah menunggu beberapa menit, bagian ke 6 berhasil didapatkan kemudian semua bagian flag di gabung.

Flagnya adalah
ALEXCTF{dv5d4s2vj8nk43s8d8l6m1n5l67ds9v41n52nv37j481h3d28n4b6v3k}

CR2: Many time secrets [100 Points]

Soal

This time Fady learned from his old mistake and decided to use onetime pad as his encryption technique, but he never knew why people call it one time pad!

Soal berupa file teks

```
0529242a631234122d2b36697f13272c207f2021283a6b0c7908
2f28202a302029142c653f3c7f2a2636273e3f2d653e25217908
322921780c3a235b3c2c3f207f372e21733a3a2b37263b313012
2f6c363b2b312b1e64651b6537222e37377f2020242b6b2c2d5d
283f652c2b31661426292b653a292c372a2f20212a316b283c09
29232178373c270f682c216532263b2d3632353c2c3c2a293504
613c37373531285b3c2a72273a67212a277f373a243c20203d5d
243a202a633d205b3c2d3765342236653a2c7423202f3f652a18
2239373d6f740a1e3c651f207f2c212a247f3d2e65262430791c
263e203d63232f0f20653f207f332065262c3168313722367918
2f2f372133202f142665212637222220733e383f2426386b
```

Solusi

- Pesan di enskripsi menggunakan metode xor dengan key yang sama kemudian diencode ke dalam bentuk hex.
- Tiap baris pesan memiliki panjang 26 karakter.
- Key kemungkinan adalah flagnya

```
C = Ciphertext
M = Plaintext

C1 ^ C2 == M1 ^ M2
```

- Contoh flag adalah ALEXCTF{}
- Oleh karena itu kita sudah mendapatkan “ALEXCTF{“ delapan karakter

```
#!/usr/bin/python

def strxor(s1,s2):
    return ‘’.join(chr(ord(a) ^ ord(b)) for a,b in zip(s1,s2))

texts=[‘0529242a631234122d2b36697f13272c207f2021283a6b0c7908’,’
2f28202a302029142c653f3c7f2a2636273e3f2d653e25217908’,’32292178
0c3a235b3c2c3f207f372e21733a3a2b37263b313012’,’2f6c363b2b312b1e
64651b6537222e37377f2020242b6b2c2d5d’,’283f652c2b31661426292b65
3a292c372a2f20212a316b283c09’,’29232178373c270f682c216532263b2d
3632353c2c3c2a293504’,’613c37373531285b3c2a72273a67212a277f373a
243c20203d5d’,’243a202a633d205b3c2d3765342236653a2c7423202f3f65
2a18’,’2239373d6f740a1e3c651f207f2c212a247f3d2e65262430791c’,’2
63e203d63232f0f20653f207f332065262c3168313722367918’,’2f2f37213
3202f1426652126372222220733e383f2426386b’]

key = “ALEXCTF{“

for i in texts:
     print strxor(key,i.decode(‘hex’)[0:8])
```

Output:

```
Dear Fri
nderstoo
sed One
n scheme
is the o
hod that
 proven
ever if
cure, Le
gree wit
ncryptio
```

- Sepertinya langkah pertama benar.
- Selanjutnya pada baris pertama sepertinya terusan plaintext yang terpotong adalah Dear Friend
- Kita coba langkah selanjutnya.

```
Part = “Dear Friend”

print strxor(texts[0].decode(‘hex’)[0:len(part)],part)
```

Output:

```
ALEXCTF{HER
```

- Kita coba mendecrypt dengan key yang sudah bertambah

```
for i in texts:
     print strxor(key,i.decode('hex')[0:len(key)])
```

```
Output:
Dear Friend
nderstood m
sed One tim
n scheme, I
is the only
hod that is
 proven to
ever if the
cure, Let M
gree with m
ncryption s
```

- Lakukan langkah selanjutnya seperti sebelumnya sampai semua key didapatkan secara utuh

Flag adalah **ALEXCTF{HERE_GOES_THE_KEY}**

CR3: What is this encryption? [150 Points]

Soal

Fady assumed this time that you will be so n00b to tell what encryption he is using
he send the following note to his friend in plain sight :

p=0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0d1dc7bd97055e8275cd4b78ec63c5d592f567c66393a061324aa2e6a8d8fc2a910cbee1ed9

q=0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5a1218f5d91fd0102a4c8de11f28be5e4d0ae91ab319f4537e97ed74bc663e972a4a9119307

e=0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc393ab7852fbcb11abbebfd6aaae8032db1316dc22d3f7c3d631e24df13ef23d3b381a1c3e04abcc745d402ee3a031ac2718fae63b240837b4f657f29ca4702da9af22a3a019d68904a969ddb01bcf941df70af042f4fae5cbeb9c2151b324f387e525094c41

c=0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db76d119a3efe24cb04b9449f53becd43b0b46e269826a983f832abb53b7a7e24a43ad15378344ed5c20f51e268186d24c76050c1e73647523bd5f91d9b6ad3e86bbf9126588b1dee21e6997372e36c3e742847347488918296650 86e0dc523ed23c386bb520

He is underestimating our crypto skills!

Solusi

- Pesan dienkripsi dengan menggunakan metode RSA

```
#!/usr/bin/python

import gmpy2

def num_to_str(num):
     res = “”
     while num > 0:
          res = chr(num % 256) + res
          num = num / 256
     return res
```

```
p=int("0xa6055ec186de51800ddd6fcbf0192384ff42d707a55f57af4fcfb0
d1dc7bd97055e8275cd4b78ec63c5d592f567c66393a061324aa2e6a8d8fc2a
910cbee1ed9",16)

q=int("0xfa0f9463ea0a93b929c099320d31c277e0b0dbc65b189ed76124f5
a1218f5d91fd0102a4c8de11f28be5e4d0ae91ab319f4537e97ed74bc663e97
2a4a9119307",16)
t = (p-1)*(q-1)
e=int("0x6d1fdab4ce3217b3fc32c9ed480a31d067fd57d93a9ab52b472dc3
93ab7852fbcb11abbebfd6aaae8032db1316dc22d3f7c3d631e24df13ef23d3
b381a1c3e04abcc745d402ee3a031ac2718fae63b240837b4f657f29ca4702d
a9af22a3a019d68904a969ddb01bcf941df70af042f4fae5cbeb9c2151b324f
387e525094c41",16)
d=gmpy2.invert(e,t)
c=int("0x7fe1a4f743675d1987d25d38111fae0f78bbea6852cba5beda47db
76d119a3efe24cb04b9449f53becd43b0b46e269826a983f832abb53b7a7e24
a43ad15378344ed5c20f51e268186d24c76050c1e73647523bd5f91d9b6ad3e
86bbf9126588b1dee21e6997372e36c3e74284734748891829665086e0dc523
ed23c386bb520",16)
n=p*q
m=pow(c,d,n)

print num_to_str(m)
```

Flag adalah **ALEXCTF{RS4_I5_E55ENT1AL_T0_D0_BY_H4ND}**

CR4: Poor RSA [200 Points]

Soal

This time Fady decided to go for modern cryptography implementations, He is fascinated with choosing his own prime numbers, so he picked up RSA once more. Yet he was unlucky again!

Soal berupa file :

```
flag.b64: ASCII text
key.pub: ASCII text
```

Solusi

```
openssl rsa -noout -text -inform PEM -in key.pub -pubin
```

Output:

```
Public-Key: (399 bit)
Modulus:
    52:a9:9e:24:9e:e7:cf:3c:0c:bf:96:3a:00:96:61:
    77:2b:c9:cd:f6:e1:e3:fb:fc:6e:44:a0:7a:5e:0f:
    89:44:57:a9:f8:1c:3a:e1:32:ac:56:83:d3:5b:28:
    ba:5c:32:42:43
Exponent: 65537 (0x10001)
```

- Kita harus mendapatkan nilai p dan q, kita bisa menggunakan web factordb.com
- Sebelum kita harus mengubah nilai hex modulus menjadi integer kemudian kita factorkan di web factordb.com
- Setelah didapatkan nilai n, p, q dan e, maka kita bisa melakukan proses dekripsi.

```
#!/usr/bin/python

import gmpy2

def num_to_str(num):
     res = “”
     while num > 0:
          res = chr(num % 256) + res
          num = num / 256
     return res
```

```
n=833810193564967701912362955539789451139872863794534923259743419423089229206473091408403560311191545764221310666338878019
p=863653476616376575308866344984576466644942572246900013156919
q=965445304326998194798282228842484732438457170595999523426901
t=(p-1)*(q-1)
e=65537
d=gmpy2.invert(e,t)
c=546514681109897642377058885254740938209760605167931631034138331308564188002339494648530153228068817245276146038543125484
m=pow(c,d,n)

print num_to_str(m)
```

Flag adalah **ALEXCTF{SMALL_PRIMES_ARE_BAD}**

Fore1: Hit the core [50 Points]

Soal

Soal berupa fore1.core: ELF 64-bit LSB core file x86-64, version 1 (SYSV), SVR4-style, from ‘./code’

Solusi

- Kita coba extract menggunakan binwalk.
  Binwalk –e fore1.core
- Hasil:
  ```
  0.elf:     ELF 64-bit LSB core file x86-64, version 1
  (SYSV), SVR4-style, from ‘./code’
  53B14.elf: ERROR: ELF 64-bit LSB shared object, x86-64,
  version 1 (GNU/Linux), statically linked error reading
  (Invalid argument)
  55DBC.elf: ELF 64-bit LSB shared object, x86-64, version 1
  (SYSV), dynamically linked,
  BuildID[sha1]=e73446e22b4b8fb02da6444491213842935б1a86,
  stripped
  DBC.elf:   ELF 64-bit LSB executable, x86-64, version 1
  (SYSV), dynamically linked, interpreter /lib64/ld-linux-
  x86-64.so.2, stripped
  ```
- Kita coba jalankan file yang dibold, tetapi tidak bisa
- Selanjutnya kita coba buka dengan IDA Pro.
- Kemungkinan flag dihasilkan oleh fungsi sub_400616

Output IDA Pro

```
__int64 sub_400616()
{
  size_t v0; // rbx@3
  char v2; // [sp+0h] [bp-60h]@1
  char *s; // [sp+40h] [bp-20h]@1
  int i; // [sp+4Ch] [bp-14h]@1

  s =
“cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_
e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_n
kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}”;
  _isoc99_scanf(4196294LL, &v2);
  for ( i = 3; ; i += 5 )
  {
    v0 = i;
    if ( v0 >= strlen(s) )
```

```
      break;
    putchar(s[i]);
  }
  putchar(10);
  return 0LL;
}
```

```
#!/usr/bin/python

s=”cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePg
b_e_rwqr7fvbmHjklo3tews_hmkogooyf0vbnk0ii87Drfgh_n
kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}”
flag = “”
for i in range(3,len(s),5):
     flag += s[i]


print flag
```

Flag adalah **ALEXCTF{K33P_7H3_g00D_w0rk_up}**

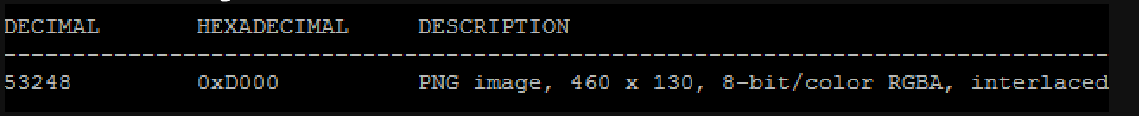
Fore3: USB probing [150 Points]

Soal

Soal berupa fore2.pcap: tcpdump capture file (little-endian) - version 2.4 (Memory-mapped Linux USB, capture length 262144)

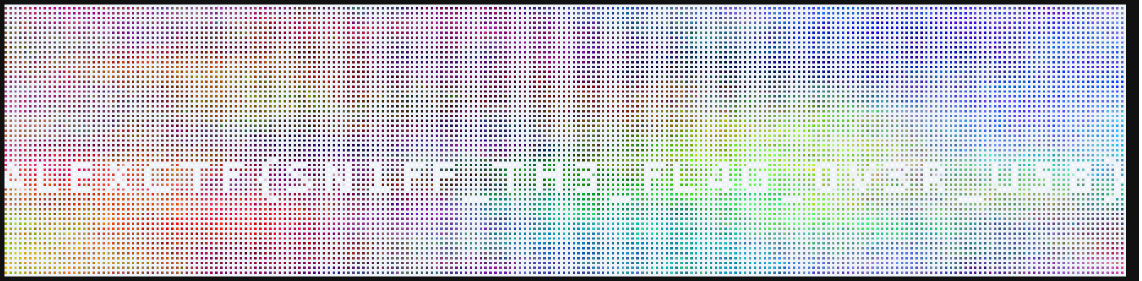
Solusi

- Buka file soal dengan Wireshark
- Kita urutkan berdasarkan length, kemudian kita extract data yang lengthnya lebih dari 95
- Beri nama file sesuai nomor.
- Kemudian gabung semua file yang telah di extract.
  cat 39.bin 49.bin 63.bin 69.bin 83.bin 89.bin 95.bin 101.bin 119.bin 125.bin >> flag
- Pindai dengan binwalk
  Binwalk flag



```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
53248         0xD000          PNG image, 460 x 130, 8-bit/color RGBA, interlaced
```

- Exract
  binwalk --dd='.*' flag



Flag adalah ALEXCTF{SN1FF_TH3_FL4G_0V3R_U58}

SC1: Math bot [100 Points]

Soal

It is well known that computers can do tedious math faster than human.

```
nc 195.154.53.62 1337
```

Solusi

```
#!/usr/bin/python

from pwn import *
r = remote('195.154.53.62',1337)
for soal in range(1,251):
        print r.recvuntil(":\n")
        prob = r.recvline()
        print prob
        prob = prob.strip().split(" ")
        bil1 = int(prob[0])
        bil2 = int(prob[2])
        op = prob[1]
        if op=='+':
                jawab = bil1+bil2
        elif op=='-':
                jawab = bil1-bil2
        elif op=='*':
                jawab = bil1*bil2
        elif op=='/':
                jawab = bil1/bil2
        elif op=='%':
                jawab = bil1%bil2
        r.sendline(str(jawab))
        print '['+str(soal)+"] "+str(bil1)+" "+op+" 
"+str(bil2)+" = "+str(jawab)

print r.recv()
print r.recv()
```

Flag adalah **ALEXCTF{1_4M_l33t_b0t}**