



## [Capture The Flag]

NAMA TIM : [Hmmm SIJA]

Institusi : SMKN 1 Cimahi

Selasa, 24 November 2020

### Ketua Tim

1. Abdul Rozaqi Wildan

### Member

1. Iftala Zahri Sukmana
2. Robi Setia Permadi

# Daftar Isi

Cryptography.....	1
Aha.....	1
rox.....	3
Kracken.....	6
Basic.....	8
Reversing & PWN.....	9
Apakah perlu patching?.....	9
Serial.....	10
Forensic.....	12
Hardwired.....	12
InspectUs.....	16
Audit 101.....	19
Web.....	21
HLA Basic.....	21
Rrrrrrrrrrrrrrrrr.....	22

## Cryptography

Aha

Challenge 7 Solves ×

aha  
464

[ahash.py](#)

Flag

Submit

### Abstraksi

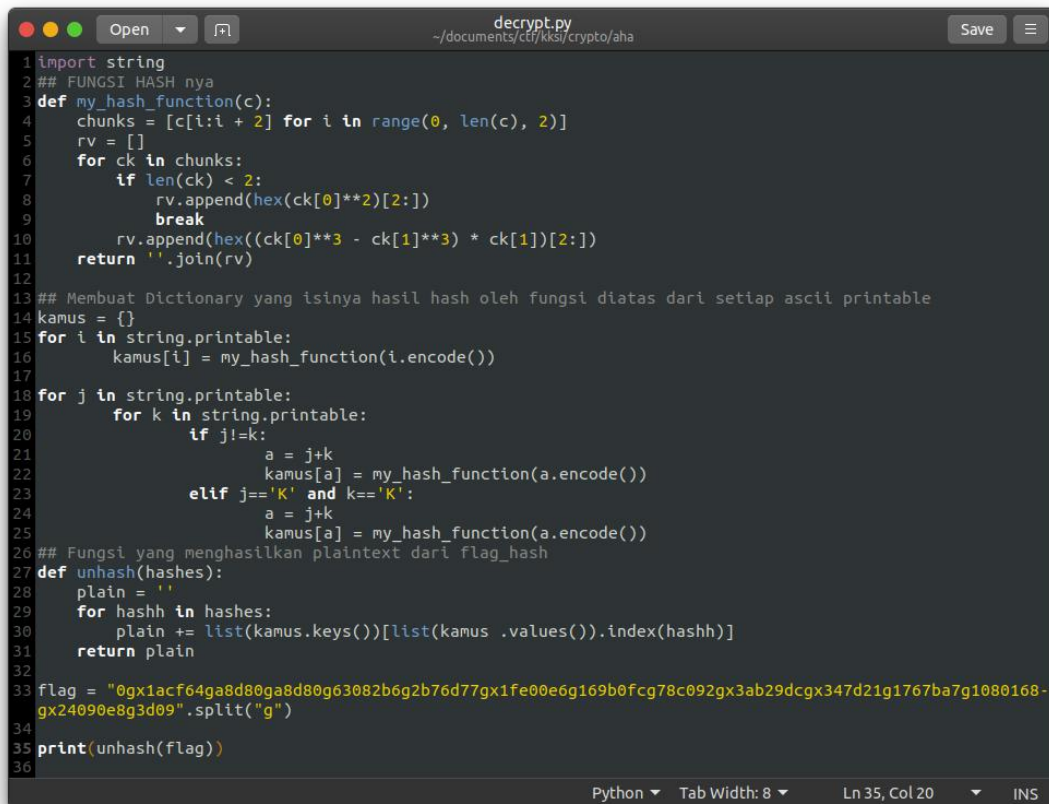
Pada soal ini dikasih sebuah file bernama ahash.py yang isinya merupakan script python dengan isi lengkap sebagai berikut :

```
1 """
2 just learned python and i can't
3 find builtin hashing functions -
4 so, i tried to make one myself.
5 not sure whether it is a hashing
6 or not because it produces diff-
7 erent length for different inpu-
8 ts. but, who cares?
9 """
10
11
12 def my_hash_function(c):
13     chunks = [c[i:i + 2] for i in range(0, len(c), 2)]
14     rv = [hex(len(c))[2:]]
15     for ck in chunks:
16         if len(ck) < 2:
17             rv.append(hex(ck[0]**2)[2:])
18             break
19         rv.append(hex((ck[0]**3 - ck[1]**3) * ck[1])[2:])
20     return 'g'.join(rv)
21
22
23 flag_hash = ('1dg0gx1acf64ga8d80ga8d80g63082b6g2b76d77gx1fe00e6g169b0fcg78c092gx3ab29dcgx347d21g1767ba-
24 7g1080168gx24090e8g3d09')
25
26 if __name__ == '__main__':
27     and my_hash_function(input('Validate your Flag = ').encode()) == flag_hash:
28         print('congrats!')
```

### Pembahasan

Setelah saya memahami script diatas, variabel flag\_hash merupakan flag yang sudah dihash oleh fungsi my\_hash\_function(). Oleh karena itu, targetnya adalah mereverse variabel flag\_hash menjadi flag yang belum dihash.

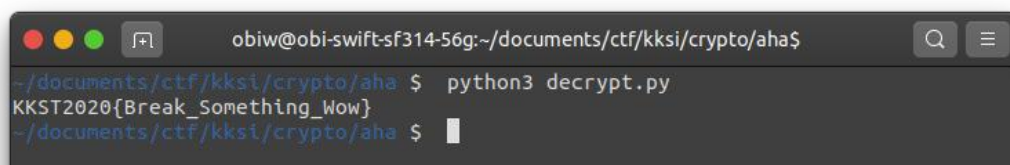
Kemudian kami membuat script untuk mereverse nilai dari variabel flag\_hash menjadi flag yang berisi sebagai berikut :



```
1 import string
2 ## FUNGSI HASH nya
3 def my_hash_function(c):
4     chunks = [c[i:i + 2] for i in range(0, len(c), 2)]
5     rv = []
6     for ck in chunks:
7         if len(ck) < 2:
8             rv.append(hex(ck[0]**2)[2:])
9             break
10        rv.append(hex((ck[0]**3 - ck[1]**3) * ck[1])[2:])
11    return ''.join(rv)
12
13 ## Membuat Dictionary yang isinya hasil hash oleh fungsi diatas dari setiap ascii printable
14 kamus = {}
15 for i in string.printable:
16     kamus[i] = my_hash_function(i.encode())
17
18 for j in string.printable:
19     for k in string.printable:
20         if j!=k:
21             a = j+k
22             kamus[a] = my_hash_function(a.encode())
23         elif j=='K' and k=='K':
24             a = j+k
25             kamus[a] = my_hash_function(a.encode())
26 ## Fungsi yang menghasilkan plaintext dari flag_hash
27 def unhash(hashhs):
28     plain = ''
29     for hashh in hashhs:
30         plain += list(kamus.keys())[list(kamus.values()).index(hashh)]
31     return plain
32
33 flag = "0gx1acf64ga8d80ga8d80g63082b6g2b76d77gx1fe00e6g169b0fcg78c092gx3ab29dcgx347d21g1767ba7g1080168-
34 gx24090e8g3d09".split("g")
35 print(unhash(flag))
36
```

Dari script diatas, pertama menulis kembali fungsi my\_hash\_function(). Selanjutnya membuat sebuah dictionary bernama kamus yang isinya merupakan hasil hash dari semua character yang dapat diprint. Kemudian membuat sebuah fungsi unhash() yang berfungsi untuk menampilkan plaintext dari variabel flag\_hash. Kemudian menjalankan fungsi unhash() dan diprint.

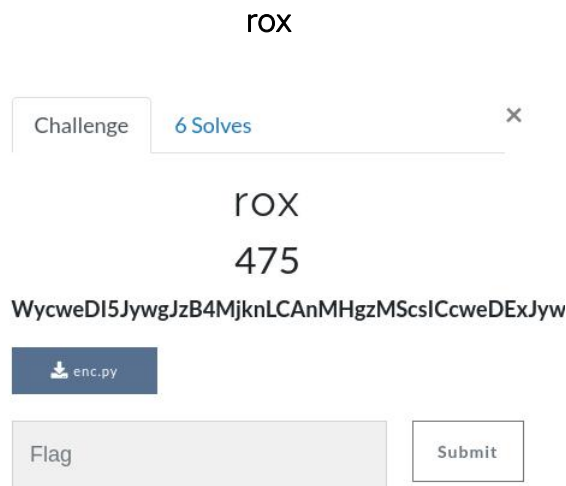
Script yang kami buat saat dijalankan akan langsung mengoutputkan flagnya seperti tangkapan layar dibawah.



```
obiw@obi-swift-sf314-56g:~/documents/ctf/kksi/crypto/aha$
~/documents/ctf/kksi/crypto/aha$ python3 decrypt.py
KKST2020{Break_Something_Wow}
~/documents/ctf/kksi/crypto/aha$
```

Setelah dijalankan scripturnya, didapatkan sebuah flag yaitu **KKST2020{Break\_Something\_Wow}**

**FLAG : KKST2020{Break\_Something\_Wow}**



## Abstraksi

Diberikan sebuah text yaitu "WycweDI5JywgJzB4MjknLCAnMHgzMScsICcweDExJywgJzB4NTAnLCAnMHg1YicsICcweDU5JywgJzB4NzUnLCAnMHgzZScsICcweDhJywgJzB4MjAnLCAnMHgyOCcsICcweDI2JywgJzB4MmUnLCAnMHgyZCcsICcweDFmJ10" dan sebuah file bernama enc.py yang berisi script seperti dibawah.

```
1 import string, random, base64
2
3 def gen_key():
4     k = ''.join([random.choice(string.ascii_letters) for x in range(0, 3)])
5     print(k)
6     return k
7
8 def _cipher(ky, pl):
9     r = random.randint(0, 10)
10    print(r)
11    random.seed(r)
12    cp = []
13    for p in pl:
14        cp.append(hex(ord(p) ^ ord(random.choice(ky))))
15    return cp
16
17 if __name__ == "__main__":
18     print(base64.b64encode(str(_cipher(gen_key(), input("Cipher : "))).encode("utf-8")).decode("utf-8"))
19 )
```

## Pembahasan

Berdasarkan soalnya, text "WycweDI5JywgJzB4MjknLCAnMHgzMScsICcweDExJywgJzB4NTAnLCAnMHg1YicsICcweDU5JywgJzB4NzUnLCAnMHgzZScsICcweDhJywgJzB4MjAnLCAnMHgyOCcsICcweDI2JywgJzB4MmUnLCAnMHgyZCcsICcweDFmJ10" merupakan hasil encrypt dari script enc.py pada soal.

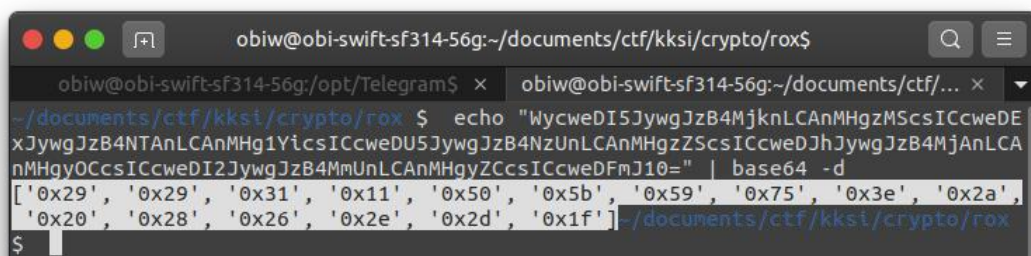
Seperti biasa, saya memahami terlebih dahulu dari script enkripsi ini. Setelah dipahami, ketahuan bahwa algoritma enkripsi ini adalah :

- 1) Menentukan plain text,
- 2) Merandom key yang terdiri dari 3 huruf( uppercase maupun lowercase),

- 3) Setelah key didapatkan, meng bitwise-xor tiap karakter dalam plain text dengan salah satu karakter pada key dimana salah satu karakter key ini didapat dengan cara dirandom,
- 4) Terakhir dioutput kan hasil bitwise xor dalam bentuk base64 encoding.

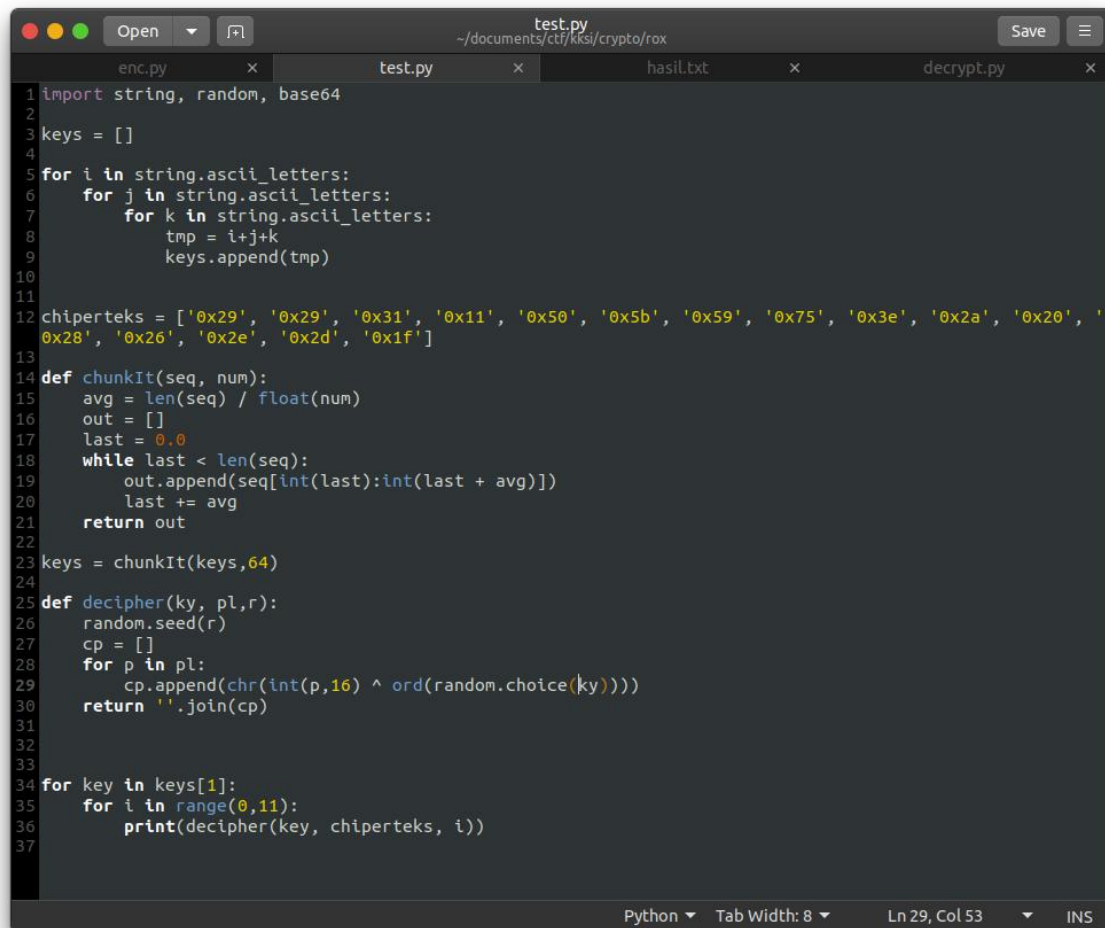
Berdasarkan algoritma itu, maka saya akan membuat sebuah script untuk membruteforce untuk mendapatkan flagnya.

Pertama saya men decoding teks "WycweDI5JywgJzB4MjknLCAnMHgzMScsICcweDE xJywgJzB4NTAnLCAnMHg1YicsICcweDU5JywgJzB4NzUnLCAnMHgzZScsICcweDJhJywgJzB4MjAnLCAnMHgyOCcsICcweDI2JywgJzB4MmUnLCAnMHgyZCcsICcweDFmJ10=" terlebih dahulu.



```
obiw@obi-swift-sf314-56g:~/documents/ctf/kksi/crypto/rox$
obiw@obi-swift-sf314-56g:/opt/Telegram$ x obiw@obi-swift-sf314-56g:~/documents/ctf/... x
~/documents/ctf/kksi/crypto/rox $ echo "WycweDI5JywgJzB4MjknLCAnMHgzMScsICcweDE
xJywgJzB4NTAnLCAnMHg1YicsICcweDU5JywgJzB4NzUnLCAnMHgzZScsICcweDJhJywgJzB4MjAnLCA
nMHgyOCcsICcweDI2JywgJzB4MmUnLCAnMHgyZCcsICcweDFmJ10=" | base64 -d
['\x29', '\x29', '\x31', '\x11', '\x50', '\x5b', '\x59', '\x75', '\x3e', '\x2a', '\x20', '\x28', '\x26', '\x2e', '\x2d', '\x1f']
~/documents/ctf/kksi/crypto/rox
$
```

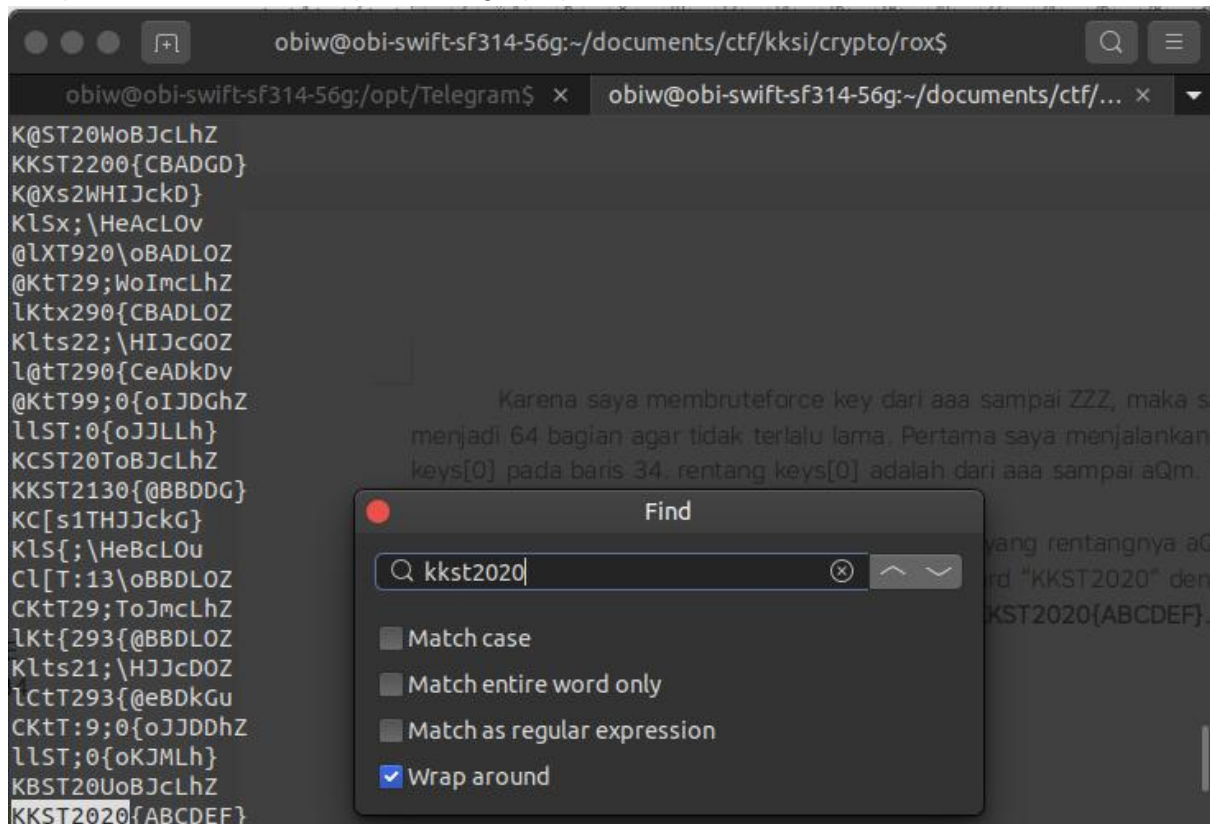
Setelah mendapat hasil decoding, saya kemudian membuat script seperti dibawah ini



```
test.py
~/documents/ctf/kksi/crypto/rox
enc.py x test.py x hasil.txt x decrypt.py x
1 import string, random, base64
2
3 keys = []
4
5 for i in string.ascii_letters:
6     for j in string.ascii_letters:
7         for k in string.ascii_letters:
8             tmp = i+j+k
9             keys.append(tmp)
10
11
12 chiperteks = ['\x29', '\x29', '\x31', '\x11', '\x50', '\x5b', '\x59', '\x75', '\x3e', '\x2a', '\x20', '\x28', '\x26', '\x2e', '\x2d', '\x1f']
13
14 def chunkIt(seq, num):
15     avg = len(seq) / float(num)
16     out = []
17     last = 0.0
18     while last < len(seq):
19         out.append(seq[int(last):int(last + avg)])
20         last += avg
21     return out
22
23 keys = chunkIt(keys,64)
24
25 def decipher(ky, pl,r):
26     random.seed(r)
27     cp = []
28     for p in pl:
29         cp.append(chr(int(p,16) ^ ord(random.choice(ky))))
30     return ''.join(cp)
31
32
33
34 for key in keys[1]:
35     for i in range(0,11):
36         print(decipher(key, chiperteks, i))
37
Python Tab Width: 8 Ln 29, Col 53 INS
```

Karena saya membruteforce key dari aaa sampai ZZZ, maka saya membagi tiap key itu menjadi 64 bagian agar tidak terlalu lama. Pertama saya menjalankan script tersebut dengan keys[0] pada baris 34. rentang keys[0] adalah dari aaa sampai aQm. Tetapi, pada rentang tersebut tidak menghasilkan flag.

Kemudian saya mencoba dengan keys[1] yang rentangnya aQn-bGz pada baris 34. Kemudian setelah dijalankan, saya mencari keyword "KKST2020" dengan fitur search yang ada pada terminal dan didapatkan flag nya yaitu **KKST2020{ABCDEF}**.

A terminal window with a dark background. The title bar shows the user 'obiw' and the path '~/documents/ctf/kksi/crypto/rox\$'. There are two tabs: 'obiw@obi-swift-sf314-56g:/opt/Telegram\$' and 'obiw@obi-swift-sf314-56g:~/documents/ctf/...'. The terminal displays a list of keys, with the last one, 'KKST2020{ABCDEF}', highlighted. A 'Find' dialog box is open in the foreground, with the search term 'kkst2020' entered. The dialog has checkboxes for 'Match case', 'Match entire word only', 'Match as regular expression', and 'Wrap around' (which is checked).

```
obiw@obi-swift-sf314-56g:~/documents/ctf/kksi/crypto/rox$  
obiw@obi-swift-sf314-56g:/opt/Telegram$ x obiw@obi-swift-sf314-56g:~/documents/ctf/... x  
K@ST20WoBJcLhZ  
KKST2200{CBADGD}  
K@Xs2WHIJckD}  
KLSx;\HeAcLOv  
@lXT920\oBADLOZ  
@KtT29;WoImcLhZ  
lKtx290{CBADLOZ  
Klts22;\HIJcGOZ  
l@tT290{CeAdkDv  
@KtT99;0{oIJDGhZ  
llST:0{oJJLLh}  
KCST20ToBJcLhZ  
KKST2130{@BBDDG}  
KC[s1THJJckG}  
KLS{;\HeBcLOu  
Cl[T:13\oBBDL0Z  
CKtT29;ToJmcLhZ  
lKt{293{@BBDL0Z  
Klts21;\HJJcD0Z  
lCtT293{@eBDkGu  
CKtT:9;0{oJJDDhZ  
llST;0{oKJMLh}  
KBST20UoBJcLhZ  
KKST2020{ABCDEF}
```

FLAG : KKST2020{ABCDEF}




## Kracken

Challenge 6 Solves X

# Kracken

## 475

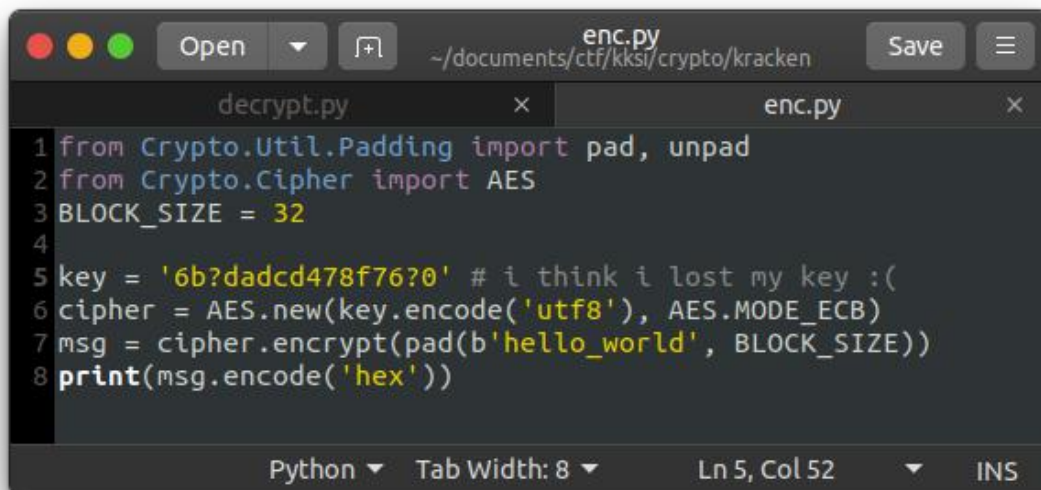
**\*\*5ada0e30fd3c562e3db448f17bbd2169a7ba768c84927798698c3acc8446f1486\*\***

 enc.py

Flag Submit

### Abstraksi

Didapatkan sebuah hexa yaitu **"5ada0e30fd3c562e3db448f17bbd2169a7ba768c8492798698c3acc8446f1486"** dan sebuah file enc.py yang berisi seperti dibawah ini



```
1 from Crypto.Util.Padding import pad, unpad
2 from Crypto.Cipher import AES
3 BLOCK_SIZE = 32
4
5 key = '6b?dadcd478f76?0' # i think i lost my key :(
6 cipher = AES.new(key.encode('utf8'), AES.MODE_ECB)
7 msg = cipher.encrypt(pad(b'hello_world', BLOCK_SIZE))
8 print(msg.encode('hex'))
```

Berdasarkan soalnya, text hexa tersebut merupakan hasil enkripsi menggunakan algoritma aes. Akan tetapi, berdasarkan clue nya key yang digunakan hilang 2 digit hexa. Oleh karena itu, saya akan mem bruteforce :)).



## Pembahasan

Saya membuat terlebih dahulu script untuk membuat list key. Habis bubur beureum bubur bodas, akhirnya didapatkan script yang mantap yaitu script dibawah ini

```
decrypt.py  x  enc.py  x
1 from Crypto.Util.Padding import pad, unpad
2 from Crypto.Cipher import AES
3 import string
4 BLOCK_SIZE = 32
5
6 flag = "Sada0e30fd3c562e3db448f17bbd2169a7ba768c8492798698c3acc8446f1486"
7 flag = bytes.fromhex(flag)
8 keys=['6b', '', 'dadcd478f76', '', '0']
9 key=[]
10 for a in string.hexdigits:
11     for b in string.hexdigits:
12         keys[1] = a.lower()
13         keys[3] = b.lower()
14         key.append(''.join(keys))
15 key = list(dict.fromkeys(key))
16 for k in key:
17     cipher = AES.new(k.encode('utf8'), AES.MODE_ECB)
18     ori = cipher.decrypt(pad(flag, BLOCK_SIZE))
19     print("key : {} \n {}".format(k,ori))
```

Setelah dijalankan script tersebut, maka didapatkan flagnya yaitu **KKST2020{Gigantic\_Sea\_Monster}** dengan menggunakan key 6b4dadcd478f76e0.

```
obiw@obi-swift-sf314-56g:~/documents/ctf/kksi/crypto/kracken$  
obiw@obi-swift-sf314-56g:/opt/Telegram$ x obiw@obi-swift-sf314-56g:~/documents/ctf/... $  
b'y9s\x9f\xa6\xdca\xfa\xae\x0c\xd6Lnh2\x07\x8c\xae\x19?\x13\xef\xf5b\xb5\x08\xfa  
\xde\x82\x1c\xbc\xf78\xc7\xfd"\x01X\x1bG\xb1\x1c\xf1\x91\x8f\xbb\xae\x84_\xc7\xf  
d"\x01X\x1bG\xb1\x1c\xf1\x91\x8f\xbb\xae\x84_  
key : 6b4dadcd478f76a0  
b'\x945-Y\xed\x8f;\xa053nE\b8\x15\x8bf\x1e\x9b\xa2\xa6\x15\x97\x80\x11\x1c\xe4  
\xd4\xa5\xabK]\xadS)Q\xc3\xe7F\xae\xab\x07@\xe3\xf9Ht$&\xeb)Q\xc3\xe7F\xae\xab\x  
07@\xe3\xf9Ht$&\xeb'  
key : 6b4dadcd478f76b0  
b"\xf8h17\cx4\x7f\x11\xd1h\xfd2zCI;\xa9\xdd,e\xc3\xb1<\xbe\xc5'\x10\xaaabx\xc6\  
xdd\xfo\xffI?3V\xaf*\xca=\x80\x8ei\xf9\xc9\x88\xbdA\xad?3V\xaf*\xca=\x80\x8ei\x1  
9\xc9\x88\xbdA\xad"  
key : 6b4dadcd478f76c0  
b'\xf0\x9f\xb9\x93\xeb\xc72t\x85\x8e\x0e\xeb\xe2q.T"uu\xc7>\xc0\x0e\xdd+N\x00\x  
a3\xdb3\x85\x10Ne\xbd\x86x\n\x01\xb0\xde\xf3qnrx{\rme\xbd\x86x\n\x01\xb0\xde\xf3q  
nxr{\rm'  
key : 6b4dadcd478f76d0  
b'"y\|xd8|\xfa|\xf3|x8d|x1a|x1b|\xfc|x7f|a0|x81f|xc1q|x14|x01|x8e|x0c~|x80|x90|xee  
|xcc|\xfc|\xebi|\xbc|\xcd|\xae|x03|\xbd|x93.\xb4|xce|\xee|\xb9]|x8f|x9e|x98\xff7|xda^T\  
|xbd|x93.\xb4|xce|\xee|\xb9]|x8f|x9e|x98\xff7|xda^T'  
key : 6b4dadcd478f76e0  
b'|KKS{T2020{Gigantic Sea Monster}\x02\x02\x8fq\x07\x8au\xdd\xa5\xf2\x06\xa9\x8b{  
-\xf2\x13\xa1\x8fq\x07\x8au\xdd\xa5\xf2\x06\xa9\x8b{-\xf2\x13\xa1'  
key : 6b4dadcd478f76f0  
b'\xe4m17|\xaf|\x9c|\xf2|\n|xdd|x12C}Y|xce|a244|\xef|\xe7|\xeeB|x7f|x06G0|xc4?|x19S|x
```

FLAG : KKST2020{Gigantic Sea Monster}

## Basic

Diberikan sebuah soal dengan hasil encode dari Sandi Morse

```
PPXY2020{-- .. .... -.-.- - ..... -.-.- --- .-.. -.-.- -- ..-..... --- --- ..-.-.-.-.- ..-.-.-
..}
```

Lalu saya Encode di web <https://morsedecoder.com/>

MISS\_THE\_OLD\_SCHOOL\_CTF?

Lalu hilangkan Underscore dan kalimatnya jadikan uppercase : **MISSTHEOLDSCHOOLCTF?**  
saya bingung ada huruf PPXY dan ternyata apabila di decode dengan rot21 menjadi KKST

PPKY2020



ROT21 ▾



KKST2020
----------

Flag : KKST2020{MISSTHEOLDSCHOOLCTF?}

## Reversing & PWN

Apakah perlu patching?

Challenge 12 Solves X

Apakah perlu patching?  
379

 cmad

Flag Submit

### Abstraksi

Diberikan sebuah elf file bernama cmad.

### Pembahasan

Saya mendecompile file tersebut menggunakan ida64. Setelah saya melihat-lihat kesana-kemari, pada fungsi validator terdapat beberapa karakter yang jika digabungkan menjadi **"PelajarSukaBelajar?"**. Kemudian saya mencoba memasukkannya sebagai flag dan ternyata memang benar flagnya adalah KKST2020{PelajarSukaBelajar?}.

```
mov     [rbp+var_70], 50h ; 'P'
mov     [rbp+var_6F], 65h ; 'e'
mov     [rbp+var_6E], 6Ch ; 'l'
mov     [rbp+var_6D], 61h ; 'a'
mov     [rbp+var_6C], 6Ah ; 'j'
mov     [rbp+var_6B], 61h ; 'a'
mov     [rbp+var_6A], 72h ; 'r'
mov     [rbp+var_69], 53h ; 'S'
mov     [rbp+var_68], 75h ; 'u'
mov     [rbp+var_67], 6Bh ; 'k'
mov     [rbp+var_66], 61h ; 'a'
mov     [rbp+var_65], 42h ; 'B'
mov     [rbp+var_64], 65h ; 'e'
mov     [rbp+var_63], 6Ch ; 'l'
mov     [rbp+var_62], 61h ; 'a'
mov     [rbp+var_61], 6Ah ; 'j'
mov     [rbp+var_60], 61h ; 'a'
mov     [rbp+var_5F], 72h ; 'r'
mov     [rbp+var_5E], 3Fh ; '?'
mov     [rbp+var_74], 0
jmp     short loc_7F1
```

FLAG : KKST2020{PelajarSukaBelajar?}

# Serial

## Abstraksi

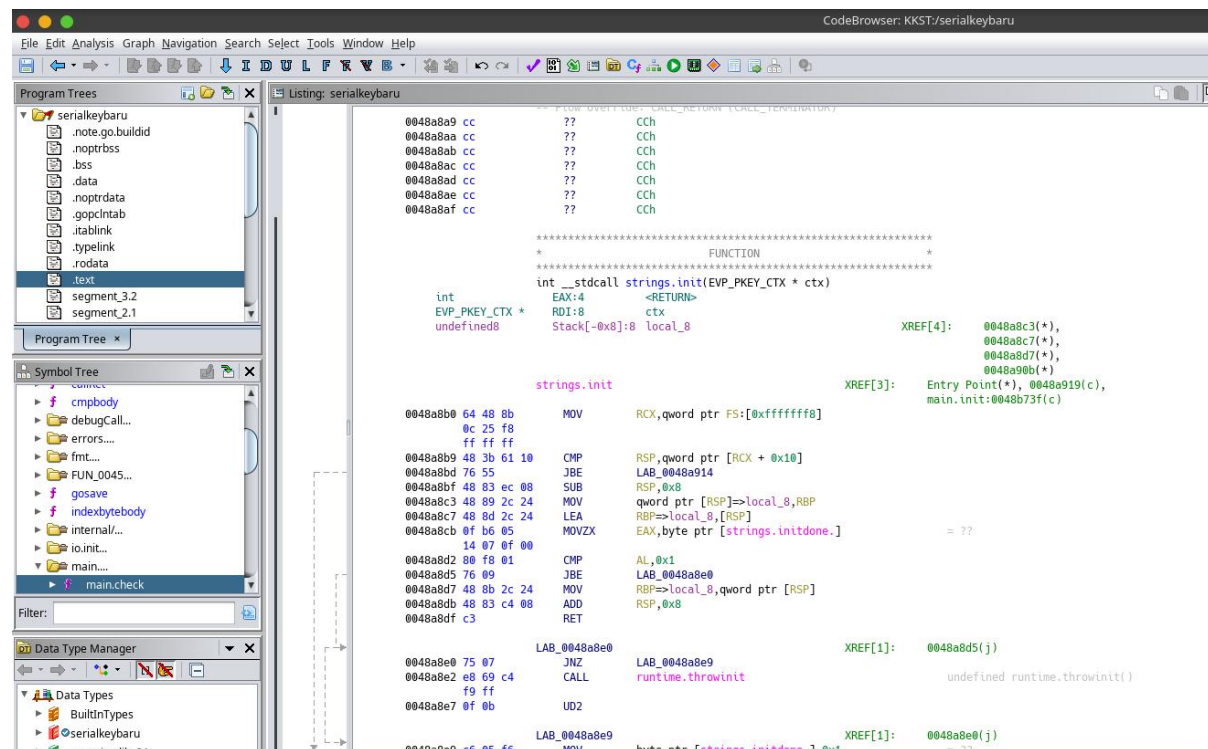
Diberikan sebuah file ELF, dimana kita diminta untuk memasukkan serial key untuk mendapatkan flag

## Pembahasan

Saya mencoba mengidentifikasi terlebih dahulu file tersebut

```
ifzahri@ifzahri-computer ~/Documents/CTF: file serialkey
serialkey: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, Go BuildID=ENdZ5bYWiwd4eBvrSHaX/z_zg4DDJSyxcNm2lOM-t/6Jha6p2lg1rqzBenkyJN/Bt6c0RXA7faVd0KwMfq8, not stripped
```

File ini ternyata file static, yang ketika dibuka akan menampilkan banyak sekali function, maka saya mencoba membuka ghidra untuk melakukan disassemble file tersebut



Pandangan saya tertuju pada function main.check yang telah didecompile. Bagian tersebut melakukan pengecekan perbandingan pada suatu stack namun stack yang dilakukan perbandingan pun merupakan stack yang sama namun yang diambil merupakan index yang berbeda. Ketika saya iseng melakukan concat manual pada string tersebut saya mendapati ilham bahwa inilah serial key yang dimaksud, yaitu XQWZ-OKLN-PWDT-TGBS

```

24 | undefined local_28 [10];
25 | undefined local_18 [16];
26 |
27 | puVar1 = (ulong *)((*long *) (in_FS_OFFSET + 0xffffffff8) + 0x10);
28 | if ((undefined *)*puVar1 <= local_d8 && local_d8 != (undefined *)*puVar1) {
29 |     if (in_stack_00000010 == 0x13) {
30 |         if ((((((((*in_stack_00000008 == 'X') && (in_stack_00000008[1] == 'Q')) &&
31 |             (in_stack_00000008[2] == 'W')) &&
32 |             ((in_stack_00000008[3] == 'Z' && (in_stack_00000008[4] == '-')))) &&
33 |             ((in_stack_00000008[5] == 'O' &&
34 |             ((in_stack_00000008[6] == 'K' && (in_stack_00000008[7] == 'L'))))) &&
35 |             (in_stack_00000008[8] == 'N')) &&
36 |             (((in_stack_00000008[9] == '-' && (in_stack_00000008[10] == 'P')) &&
37 |             (in_stack_00000008[0xb] == 'W')) &&
38 |             (((in_stack_00000008[0xc] == 'D' && (in_stack_00000008[0xd] == 'T')) &&
39 |             ((in_stack_00000008[0xe] == '-' &&
40 |             ((in_stack_00000008[0xf] == 'T' && (in_stack_00000008[0x10] == 'G')))))))) &&
41 |             ((in_stack_00000008[0x11] == 'B' && (in_stack_00000008[0x12] == 'S'))))) {
42 |             runtime.convTstring();
43 |             local_108 = CONCAT88(local_148,0x49c500);
44 |             fmt.Sprintf();
45 |             runtime.convTstring();
46 |             fmt.Fprintln();
47 |         }
48 |     else {

```

Maka saya jalankan program dan memasukkan serial tersebut, dan flag pun didapat.

```

ifzahri@ifzahri-computer ~/Documents/CTF ./serialkey
1 > Get Flag
2 > Exit
>>> 1
License Key : XQWZ-OKLN-PWDT-TGBS
KKST2020{XQWZ-OKLN-PWDT-TGBS}

```

KKST2020{XQWZ-OKLN-PWDT-TGBS}



Forensic

Hardwired

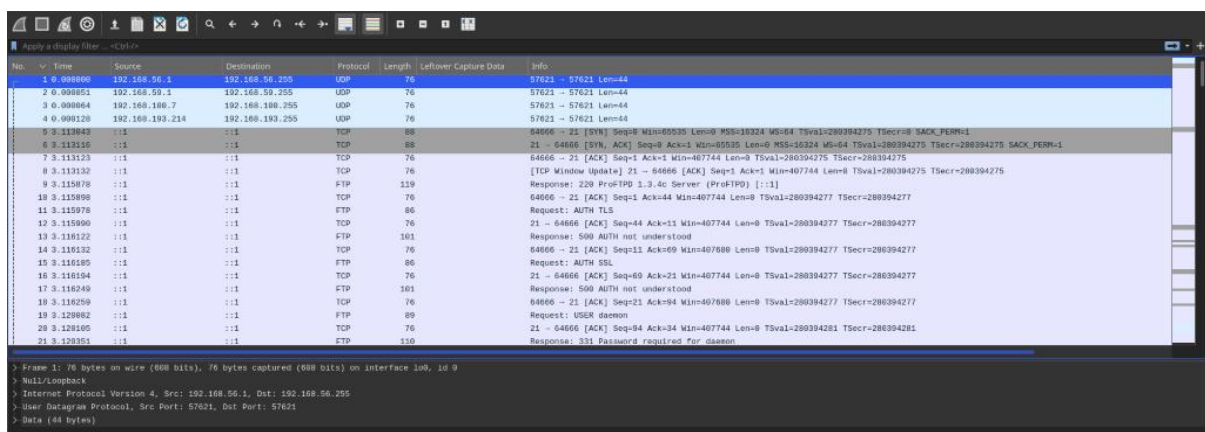
Abstraksi



Diberikan sebuah file berekstensi pcapng, yang ketika saya amati berisi aliran data pada FTP dan ada file yang terlibat untuk ditransaksikan di aliran data tersebut.

Pembahasan

Berikut tampilan dari filenya ketika dibuka menggunakan Wireshark

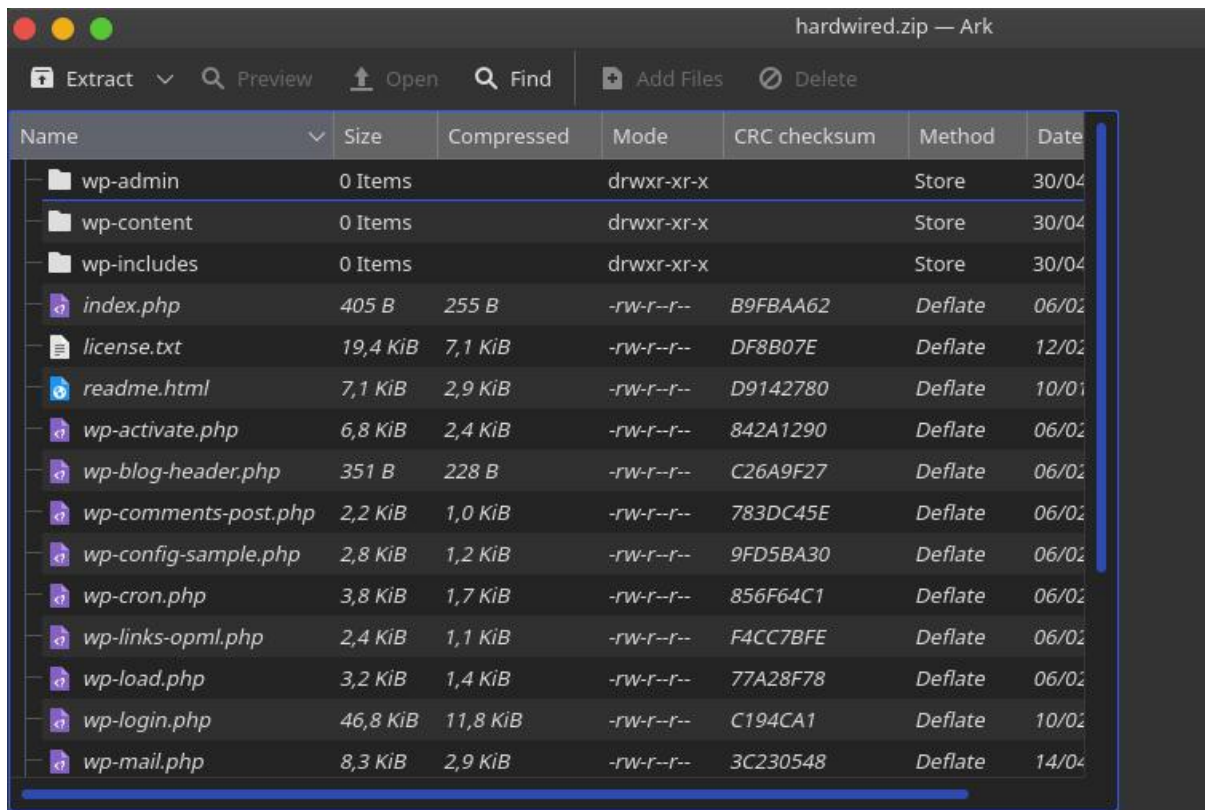


Kemudian saya cek pada protokol FTP-DATA, ada potongan file berekstensi ZIP didalamnya





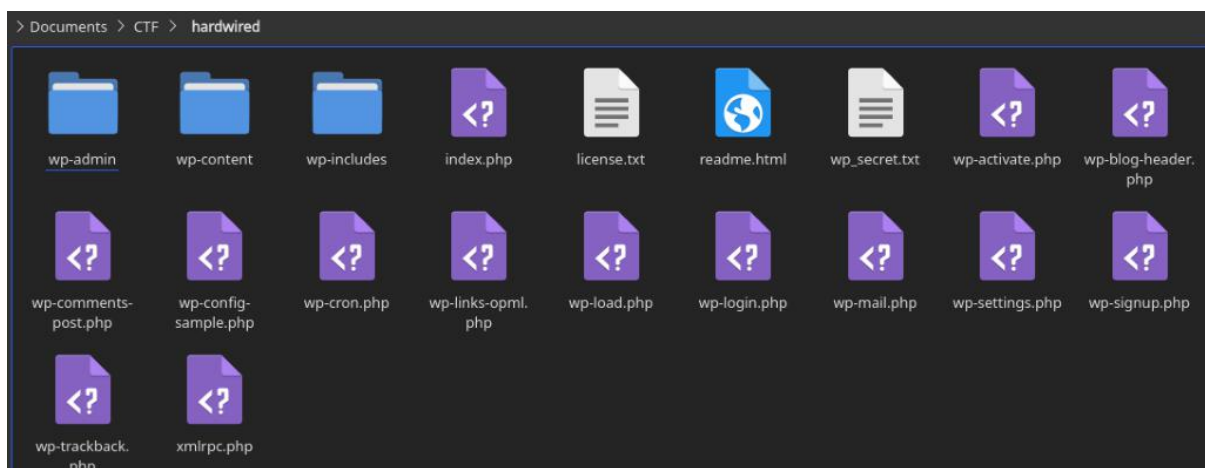
Saya ekspor string tersebut dan disimpan di local machine dengan ekstensi ZIP



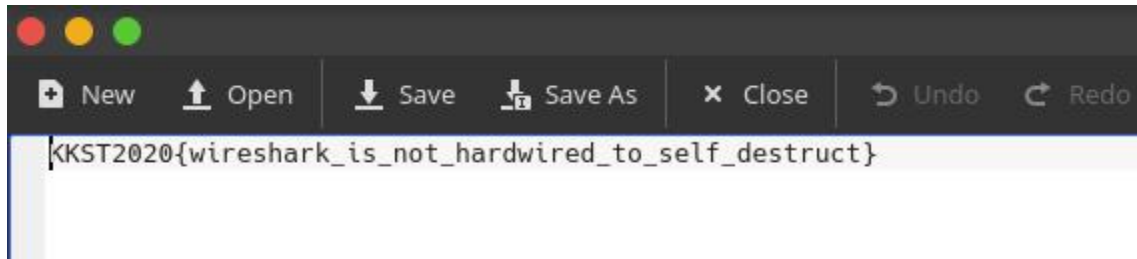
Ketika mencoba mengekstrak file ZIP tersebut, filenya terkunci dengan password. Saya tidak menemukan potensi kata kunci pada file dan membuka lagi Wireshark untuk melihat kemungkinan adanya string password yang terdeteksi

FTP	101	Response: 500 AUTH not understood
FTP	89	Request: USER daemon
FTP	110	Response: 331 Password required for daemon
FTP	105	Request: PASS hardwired_selfdestruct
FTP	103	Response: 230 User daemon logged in

Saya menemukan packet yang menarik yang diidentifikasi sebagai PASS, saya coba masukkan string tersebut dan filenya terekstrak



Semuanya merupakan file-file yang umum ditemui pada Wordpress, namun file bernama wp\_secret.txt menarik perhatian saya karena berpotensi menyimpan flag. Dan ketika dibuka, benar ada flag didalamnya.



KKST2020{wireshark\_is\_not\_hardwired\_to\_self\_destruct}



```

GET /inspectus/cooking.png HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:82.0) Gecko/20100101 Firefox/82.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://localhost/inspectus/
Cache-Control: max-age=0

HTTP/1.1 200 OK
Date: Sat, 14 Nov 2020 02:07:15 GMT
Server: Apache/2.4.37 (Unix) OpenSSL/1.0.2p PHP/7.2.12 mod_perl/2.0.8-dev Perl/v5.16.3
Last-Modified: Sat, 14 Nov 2020 02:02:09 GMT
ETag: "27bb-5b407885b2a40"
Accept-Ranges: bytes
Content-Length: 10171
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: image/png

.PNG
IHDR.....C.....PLTE.....ggg.....qqq000^^^DDDe6ennnLL==##aaz
W.....(((.....:.....IDATx.....N.....2.2(.....ow.0%$.....Z.....F.....T)5@.....?G.....$I.....TE.....(#.....3UDh*N~x!I..s..D.....ja.'~..?
I.....n.....cb.....l.v.....e.....g.Or..S7..%sPV/...?..+.30P...m.....@_@.....FZA.....A...u4M.....B...hR...a...m.....I...
49t.I.....a.Y.....aXI..S.?..&.....<G..?.degl.....
.k..J.....Z.....D.....NS.....0.....>D...A...7.....<0.L...?{.....s.L.....x.....".Ro.....wm.9..~{...Pd.r.f.6...>kM..2fV.03..L.....).y.....(.pW\..
5~..^.....sL...Z'h..>.....(.....00V.x%.....=>.3.....B..G.....c.a...b..Y'I..W..I..IQ=\..T]S...0H..6...y1.....<aqxb.f5..S).....h.....{...3.....6X..?x.r0..
5\..'.i...x.A..x%.....Z.3.sZ.k.#.....'vx.Ds..R.....Zt2.c<..C..].R.*.4.1.[.....!N1.O..].i.....'.u:r..7.C:
.F.i..M..@ ex<c.i
..+w
..].1..M1?~.h...wuZ09a...2e.....".....:q.....<...=.Q.Y:.....P0P9'.....;.....W+...!.....*k+.....
..4Y=...Yn..8...a.A.*^..3~..U..tfv.....x.L.B..1...{z]Z0B...3.$...qJ.....f.wX...TN...CI.....1...$.(.f..f..Kb...H'&j~...U.N.N
..a.c...0r.....G...../J.....%.....c.Qo...[T]2.....0..n].....-0.7)P..3.5.J.....[2(.....Ij.1.....t.s.E...@

```

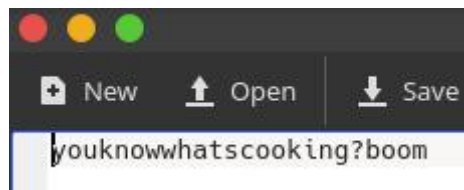
Saya mencoba mengeksplor file tersebut dan menyimpannya sebagai gambar berekstensi PNG. Ketika dibuka terlihat bahwa gambar telah di-render dengan baik. Berikut gambarnya



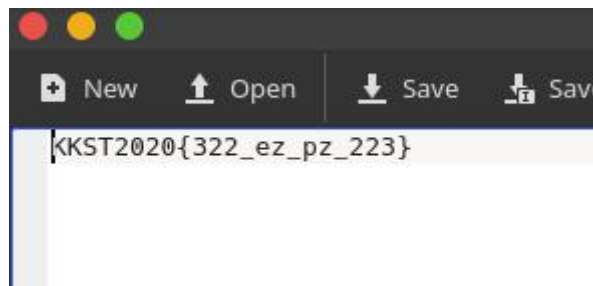
Sekilas tidak ada yang menarik di gambar tersebut yang bisa dijadikan hint step selanjutnya. Namun, ketika membedah metadata gambar tersebut, ada file zip yang terselip disitu. File tersebut terkunci ketika saya mengekstraknya

00000019.zip — Ark						
<div> <div>Extract</div> <div>Preview</div> <div>Open</div> <div>Find</div> <div>Add Files</div> <div>Delete</div> </div>						
Name	Size	Compressed	Mode	CRC checksum	Method	Date
recipe.txt	24 B	36 B	-rw-r--r--	5378982F	Store	14/11/20 09.00

Saya kembali menjelajahi wireshark dan mendapati file-file unik seperti rand.txt dan key.txt. Saya coba ekstrak file key.txt karena berpotensi mengandung password file recipe.zip. Berikut yang saya dapat



Saya coba inputkan string tersebut ke kolom password, dan zip berhasil terekstrak dan berisi file recipe.txt. Ketika dibuka file tersebut, ada flag yang telah menanti



KKST2020{322\_ez\_pz\_223}



# Audit 101

Disuguhkan dengan file yang didalamnya ada informasi akses log dari suatu web

```
127.0.0.1 - - [14/Nov/2020:21:50:07 +0700] "GET http://localhost/login/?username=admin&password=tes HTTP/1.1" 200 389
127.0.0.1 - - [14/Nov/2020:21:50:14 +0700] "GET http://localhost/login/?username=admin&password=admin HTTP/1.1" 200 364
127.0.0.1 - - [14/Nov/2020:21:50:27 +0700] "GET http://localhost/login/?username=admin'or%1x3d1--+&password=tes HTTP/1.1" 200 364
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22a%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22b%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22c%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22d%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22e%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22f%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22g%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22h%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22i%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22j%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22k%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22l%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22m%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22n%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22o%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22p%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22q%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22r%22)--"
:::1 - - [14/Nov/2020:21:50:41 +0700] "GET /login/index.php?username=admin'and+(select+true+from+dual+where+substr(database(),1,1)%3d%22s%22)--"
```

Lalu sempat terdiam sejenak, dan melihat keanehan di ujung ada angka 364 yang berbeda dengan yang lain, dan diawali huruf "K" maka saya seleksi dengan data yang ujungnya angka 364 dan berurutan 1.1 2.1 sampai 35.1, sehingga seperti ini :

[illegible]

Lalu decode URL di : <https://meyerweb.com/eric/tools/dencoder/> dan copas semua access log yang sudah di filter tadi

## URL Decoder/Encoder

[illegible]

Dan urutkan hurufnya sehingga membentuk flag seperti ini :

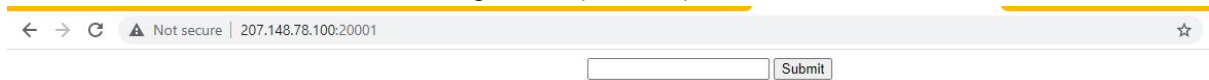
KKST2020{s1mple\_http\_l0g\_aud1t\_101}



## Web

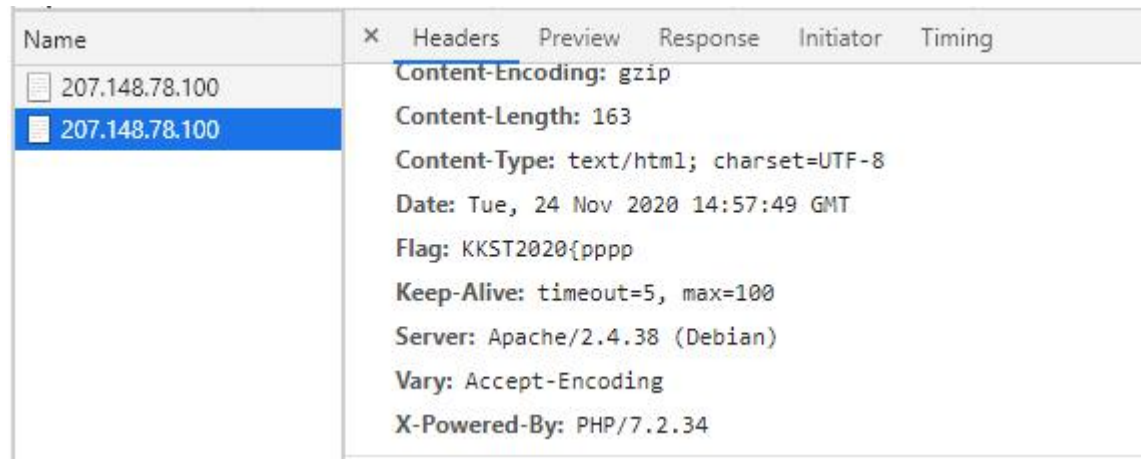
## HLA Basic

Diberikan sebuah Link website dengan tampilan seperti ini :



saya mengira ada vuln RCE, tetapi bukan.

Lalu saya menginputkan kata aaaaaaaa dan saya inspect dan lihat network nya



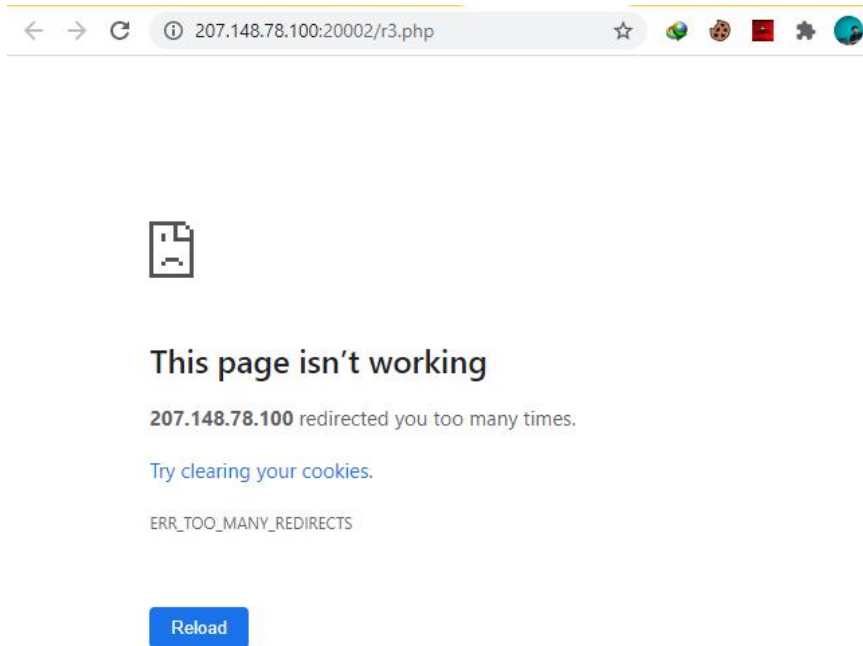
Disitu ada flag maka saya teruskan memasukan 104 karakter :

[illegible]

Dan didapatkan flagnya :

[illegible]

Diberikan sebuah web dengan tampilan seperti ini



Disini saya terheran, kenapa web nya down dan juga ada banyak request. Lalu saya membuka wireshark.

Dan memfilter nya dengan protokol HTTP, Disini saya membuka satu demi satu paket yang didapat dan didapati flagnya.

No.	Time	Source	Destination	Protocol	Length	Info
11	0.530005	207.168.43.212	207.148.78.100	HTTP	536	GET /index.php HTTP/1.1
13	0.649905	207.148.78.100	192.168.43.212	HTTP	304	HTTP/1.1 302 Found
14	0.668804	192.168.43.212	207.148.78.100	HTTP	533	GET /r1.php HTTP/1.1
16	0.731228	207.148.78.100	192.168.43.212	HTTP	303	HTTP/1.1 302 Found
18	0.889613	192.168.43.212	207.148.78.100	HTTP	533	GET /r2.php HTTP/1.1
19	0.945507	207.148.78.100	192.168.43.212	HTTP	303	HTTP/1.1 302 Found
22	0.981182	192.168.43.212	207.148.78.100	HTTP	533	GET /r3.php HTTP/1.1
24	1.161337	207.148.78.100	192.168.43.212	HTTP	330	HTTP/1.1 302 Found (text/html)
25	1.188480	192.168.43.212	207.148.78.100	HTTP	533	GET /r4.php HTTP/1.1
26	1.243679	207.148.78.100	192.168.43.212	HTTP	303	HTTP/1.1 302 Found
29	1.286064	192.168.43.212	207.148.78.100	HTTP	533	GET /r5.php HTTP/1.1

```

\r\n
[HTTP response 4/32]
[Time since request: 0.180155000 seconds]
[Prev request in frame: 18]
[Prev response in frame: 19]
[Request in frame: 22]
[Next request in frame: 25]
[Next response in frame: 26]
[Request URI: http://207.148.78.100:20002/index.php]
File Data: 26 bytes

```

▼ Line-based text data: text/html (1 lines)

```

KKST2020{TooMany Redir3ct}

0000 34 2e 33 38 20 28 44 65 62 69 61 6e 29 0d 0a 58 4:38 (De bian)~X
0000 2d 50 6f 77 65 72 65 64 2d 42 79 3a 20 50 58 40 -Powered -By: PHP
0000 2f 37 2e 32 2e 33 34 0d 0a 6c 6f 63 61 74 69 6f /7.2.34~location
0000 6e 3a 20 72 34 2e 70 68 70 0d 0a 43 6f 6e 74 65 n: r4.ph~Conte
0000 6e 74 2d 4c 65 6e 67 74 68 3a 20 32 36 0d 0a 4b nt~Length h: 26~K
0000 65 65 70 2d 41 6c 69 76 65 3a 20 74 69 6d 65 6f eep~Alive e: timeo
0000 75 74 3d 35 2c 20 6d 61 78 3d 39 37 0d 0a 43 6f ut=5, ma x=97~Co
0000 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 nnection : Keep-A
0000 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 54 79 live~Content~Ty
0010 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c 3b 20 63 pe: text /html; c
0020 68 61 72 73 65 74 3d 55 54 6d 2d 38 0d 0a 0d 0a harset=UTF-8~...
0030 4b 4b 53 54 32 30 32 30 7b 54 6f 6f 4d 61 6e 79 KKST2020 {TooMany
0040 5f 52 65 64 31 72 33 63 74 7d Redir3ct t}

```

Flag : KKST2020{TooMany\_Red1r3ct}