KMIPN VI

Inovasi Vokasi Untuk Tren Informatika Masa Depan

CTF Jeopardy – cyberitech.rar – POLIBAN

Muhammad Aldi

Reja Revaldy. F.

Ryan Rizky Pratama

Table of Content

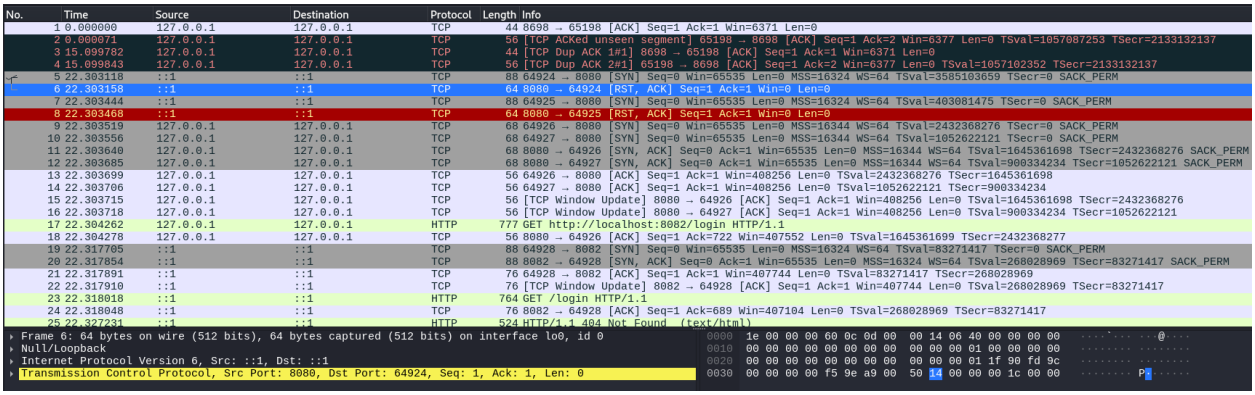
Forensic

1. campus record

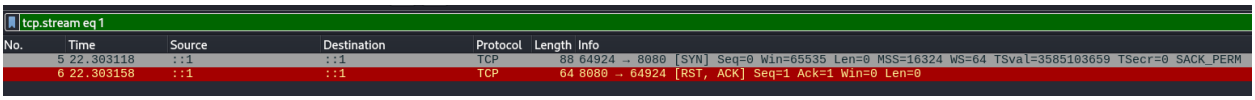
semalam network log dikampus ku lumayan padat, bisa bantu di check?

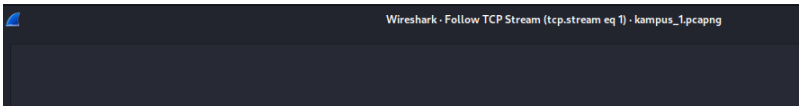
Point : 500

Solution : Diberikan sebuah file packet capture (pcap) bernama kampus_1.pcapng, lalu kami membukanya menggunakan Wireshark:

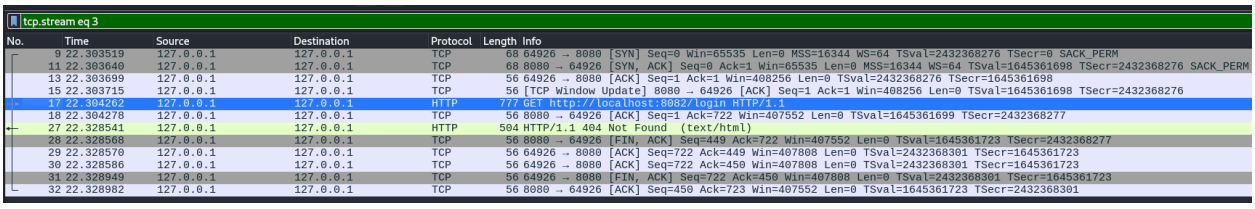


Hampir semua protocol adalah TCP. Kami lakukan analisa per-streamnya dengan follow -> TCP stream. Dimulai dari stream 1:





Hingga kami mulai menemukan clue di stream 3 dst. Mulai dari stream 3 ada ter-capture HTTP Request dan Response:



```
GET http://localhost:8082/login HTTP/1.1
Host: localhost:8082
Proxy-Connection: keep-alive
Cache-Control: max-age=0
sec-ch-ua: "Chromium";v="125", "Not.A/Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "macOS"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br, zstd
Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7

HTTP/1.1 404 Not Found
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Security-Policy: default-src 'none'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 144
Date: Mon, 24 Jun 2024 14:23:52 GMT
Connection: keep-alive
Keep-Alive: timeout=5

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /login</pre>
</body>
</html>
```

Mulai stream 9 dan 10 terlihat beberapa percobaan login yang gagal, nampaknya ada yang ingin mencoba SQL Injection:

tcp.stream eq 9

No.	Time	Source	Destination	Protocol	Length	Info
59	28.482689	::1	::1	TCP	88	64934 → 8082 [SYN] Seq=0 Win=65535 Len=0 MSS=16324 WS=64 TSval=176301004 TSecr=0 SACK_PERM
60	28.482717	::1	::1	TCP	88	8082 → 64934 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16324 WS=64 TSval=3455582535 TSecr=176301004 SACK_PERM
61	28.482720	::1	::1	TCP	76	64934 → 8082 [ACK] Seq=1 Ack=1 Win=407744 Len=0 TSval=176301004 TSecr=3455582535
62	28.482723	::1	::1	TCP	76	[TCP Window Update] 8082 → 64934 [ACK] Seq=1 Ack=1 Win=407744 Len=0 TSval=3455582535 TSecr=176301004
65	28.482900	::1	::1	HTTP/J…	411	POST /login HTTP/1.1 , JSON (application/json)
66	28.482910	::1	::1	TCP	76	8082 → 64934 [ACK] Seq=1 Ack=336 Win=407424 Len=0 TSval=3455582535 TSecr=176301004
69	28.486554	::1	::1	HTTP/J…	384	HTTP/1.1 200 OK , JSON (application/json)
70	28.486590	::1	::1	TCP	76	64934 → 8082 [ACK] Seq=336 Ack=309 Win=407488 Len=0 TSval=176301008 TSecr=3455582539

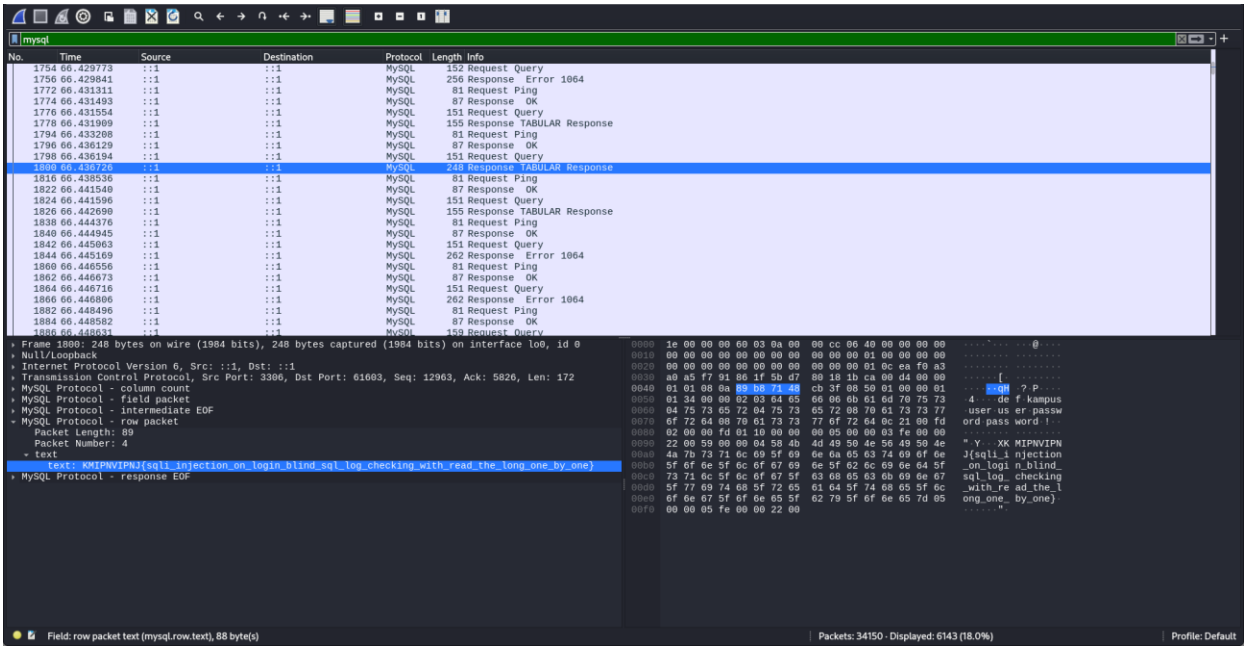
Wireshark · Follow TCP Stream (tcp.stream eq 9) · kampus_1.pcapng

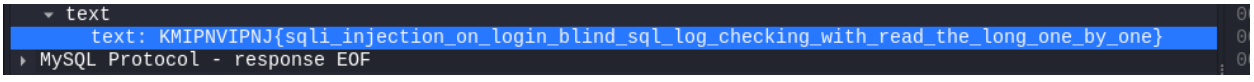
```
POST /login HTTP/1.1
Content-Type: application/json
User-Agent: PostmanRuntime/7.32.3
Accept: */*
Cache-Control: no-cache
Postman-Token: 25d55637-abd6-4e40-bccb-1205d9e7375f
Host: localhost:8082
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 51

{
    "username": "aaaa",
    "password": "aaaaa"
}HTTP/1.1 200 OK
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 41
ETag: W/"29-RZ2xxyMmmfQlLqkXiksyYt4ebt0"
Date: Mon, 24 Jun 2024 14:23:58 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{"erorr":true,"message":"wrong username"}POST /login HTTP/1.1
Content-Type: application/json
User-Agent: PostmanRuntime/7.32.3
Accept: */*
Cache-Control: no-cache
Postman-Token: 055f94be-c9c1-40b0-8679-c1ca40947349
Host: localhost:8082
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 52
```

tcp.stream eq 10

No.	Time	Source	Destination	Protocol	Length	Info
98	34.998621	::1	::1	MySQL	81	Request Ping
99	34.998881	::1	::1	TCP	76	3306 → 61603 [ACK] Seq=1 Ack=6 Win=7205 Len=0 TSval=2310534779 TSecr=3409874307
100	34.998931	::1	::1	MySQL	87	Response OK
101	34.998942	::1	::1	TCP	76	61603 → 3306 [ACK] Seq=6 Ack=12 Win=4597 Len=0 TSval=3409874307 TSecr=2310534779
102	34.999151	::1	::1	MySQL	130	Request Query
103	34.999179	::1	::1	TCP	76	3306 → 61603 [ACK] Seq=12 Ack=60 Win=7204 Len=0 TSval=2310534779 TSecr=3409874307
104	34.999378	::1	::1	MySQL	241	Response Error 1064

Kami coba filterisasi lebih spesifik untuk melihat semua hasil capture dari protocol MySQL. Dari banyaknya record, ada satu record dengan info Response TABULAR Response, setelah ditelusuri ternyata flag-nya ada di sini:





Flag:

KMIPNVIPNJ{sqli_injection_on_login_blind_sql_log_checking_with_read_the_long_one_by_one}

2. just simple image

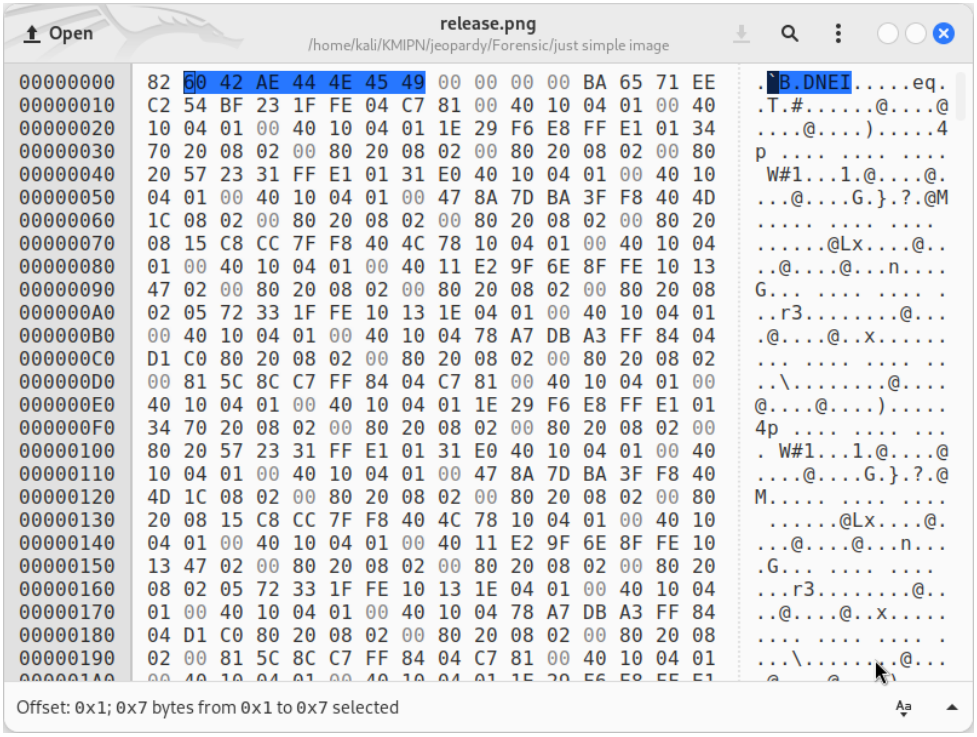
ini hanya gambar simple sesuai judul, cukup pahami saja

Point : 500

Solution: Diberikan sebuah file .png, lalu kami coba periksa menggunakan exiftool:

```
┌──(kali㉿kali)-[~/KMIPN/jeopardy/Forensic/just simple image]
└─$ exiftool release.png
ExifTool Version Number         : 12.76
File Name                       : release.png
Directory                       : .
File Size                       : 45 kB
File Modification Date/Time     : 2024:07:02 09:09:07-04:00
File Access Date/Time           : 2024:07:02 09:09:07-04:00
File Inode Change Date/Time     : 2024:07:02 09:09:07-04:00
File Permissions                : -rw-rw-r--
Error                           : File format error
```

karena kami tidak mendapatkan apa-apa, maka kami coba lihat file tersebut menggunakan hex editor:



Setelah memeriksa output, kami mencari referensi dari internet terkait soal ini:

On checking hexdump of the file, we get the output

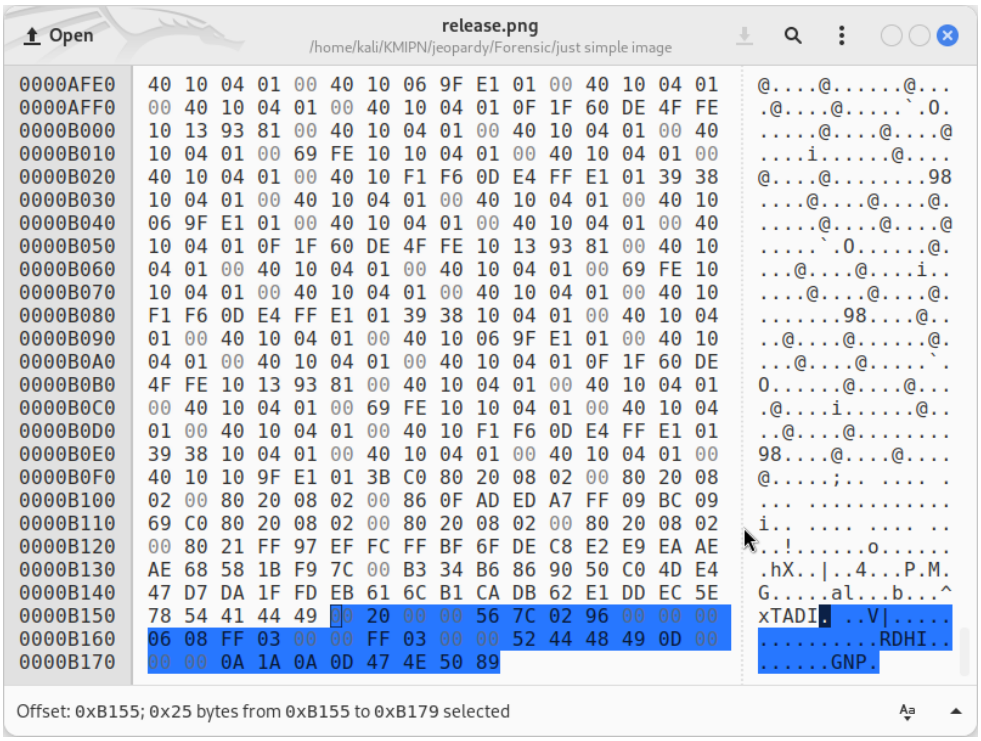
```
~/RiceTeaCatPanda/● hexdump -C motivation.txt| head
00000000  82 60 42 ae 44 4e 45 49  00 00 00 00 df db f8 e5  |.`B.DNEI........|
00000010  19 76 cb 05 03 ff ef fe  92 3f f8 11 ec 04 01 00  |.v.......?......|
00000020  40 10 04 01 00 40 10 04  01 00 40 10 04 51 d3 6e  |@....@....@..Q.n|
```

The chunks are reversed `IEND`. This hints at reversed `PNG` file. We check the tail to get the output

```
~/RiceTeaCatPanda● hexdump -C motivation.txt| tail
00040d40  c9 24 92 49 24 92 5f a3  8e 38 e3 df 38 e3 8e 38  |.$.I$._..8..8..8|
00040d50  fb fb ff ff e3 fd dd 44  0f dd ec 5e 78 54 41 44  |.......D...^xTAD|
00040d60  49 00 20 00 00 3b 4f 36  12 00 00 00 06 08 ad 03  |I. ..;O6........|
00040d70  00 00 e8 03 00 00 52 44  48 49 0d 00 00 00 0a 1a  |......RDHI......|
00040d80  0a 0d 47 4e 50 89                                 |..GNP.|
```

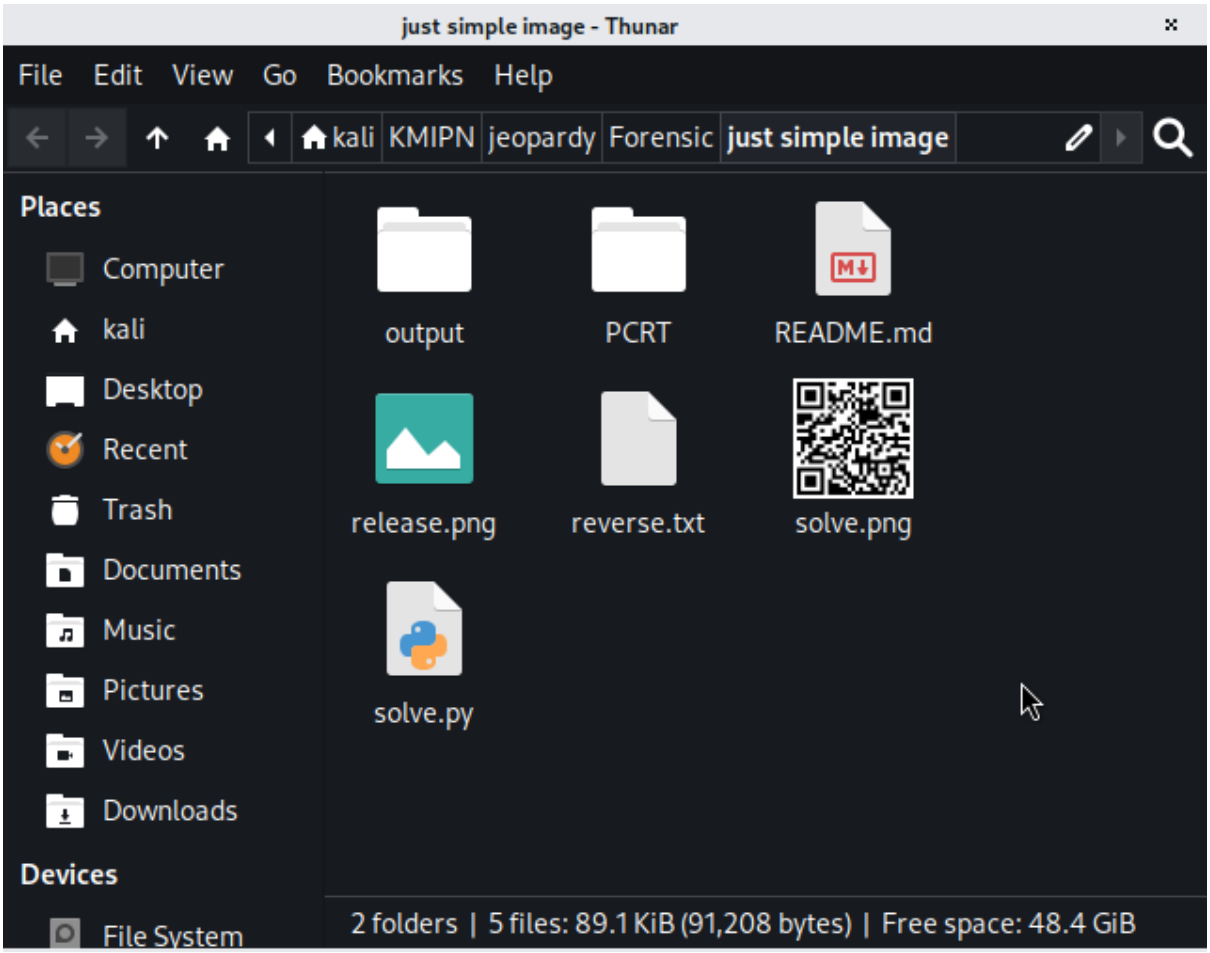
Berdasarkan referensi tersebut kami menyadari bahwa chunk tersebut adalah **IEND** yang reversed. Ini menunjukan bahwa ini adalah file PNG yang di reverse. Lalu kami memeriksa bagian akhir output dan menyadari bahwa file signature PNG tersebut tidak ada:

Maka kami coba menambahkan file signature PNG tersebut terlebih dahulu:

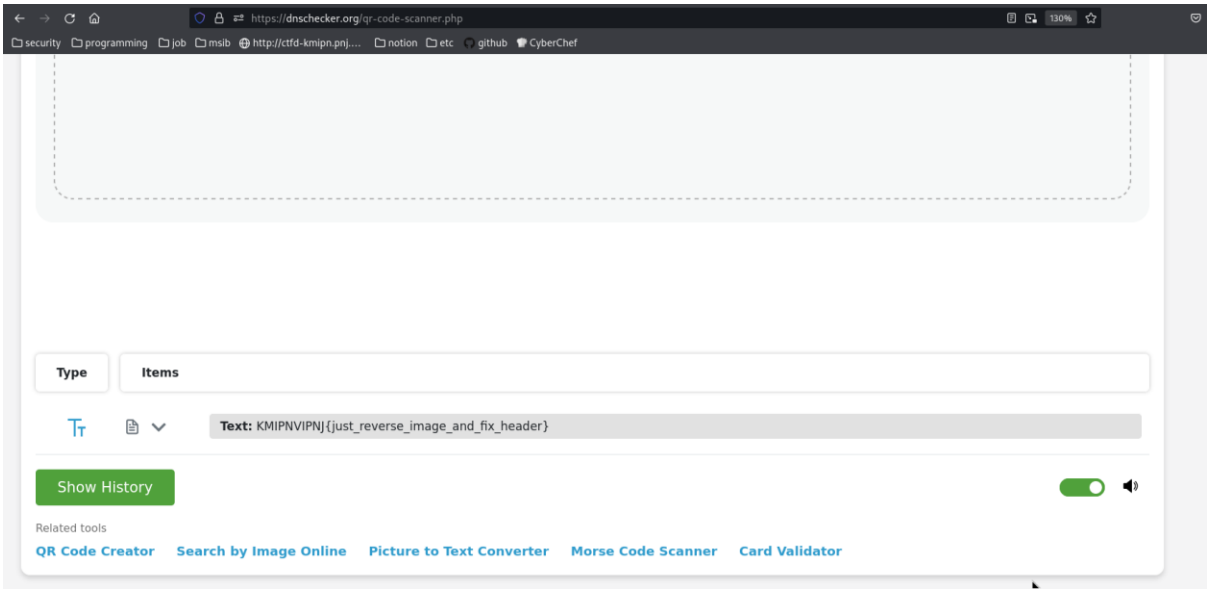


Setelah kami menyesuaikan signature-nya dengan referensi yang kami dapat, kami membuat script untuk mengembalikan file tersebut ke kondisi semula:

Setelah menjalankan script, kami mendapatkan gambar QR code:



Lalu kami coba scan menggunakan scanner yang tersedia online:



Flag : KMIPNVIPNJ{just_reverse_image_and_fix_header}

W
eb Exploitation

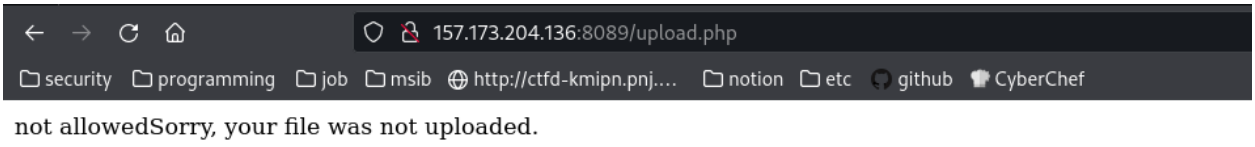
1. just simple upload

temen ku baru belajar bikin form upload, sepertinya tidak beres

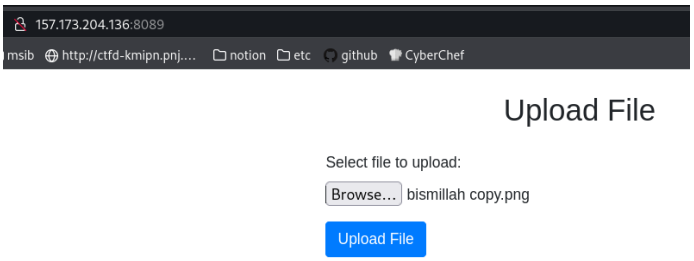
http://157.173.204.136:8089/

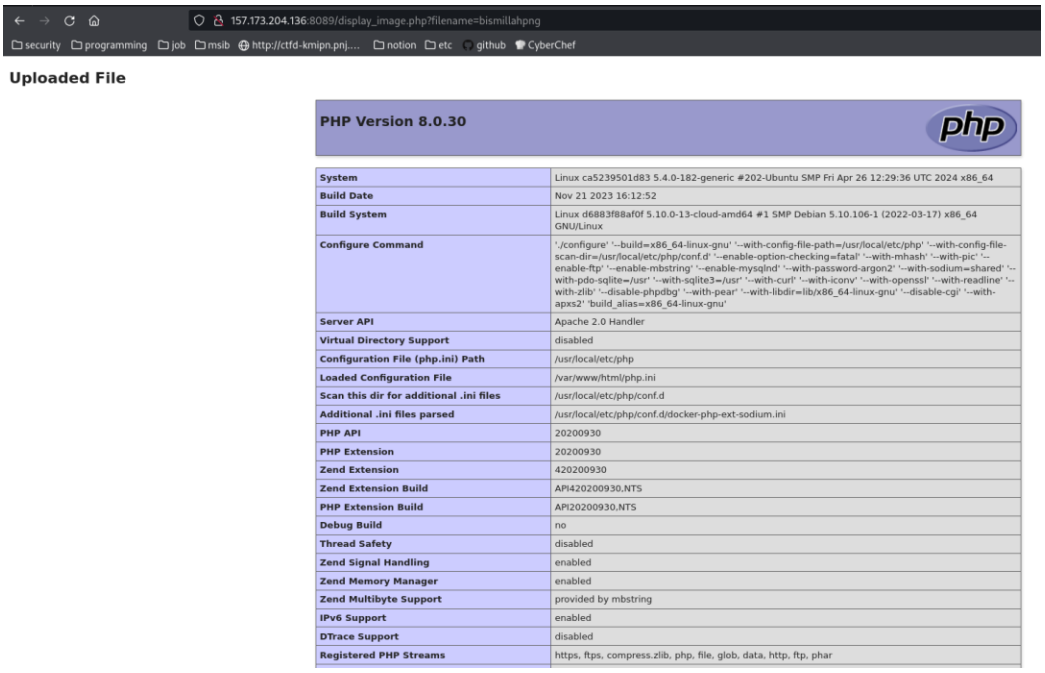
Point : 500

Solution: Disediakan sebuah website untuk upload file, lebih tepatnya upload gambar, karena kami familiar dengan kerentanan pada file upload maka saya coba masukkan file php guna untuk melakukan pengecekan apakah website nya memiliki kerentanan di sisi file upload:



ternyata tidak bisa dan kami pun mencoba hanya dengan mengubah extension dari file menjadi png:

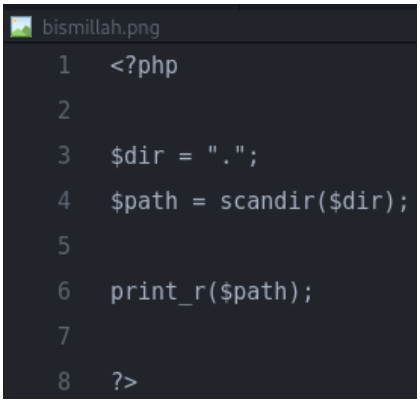


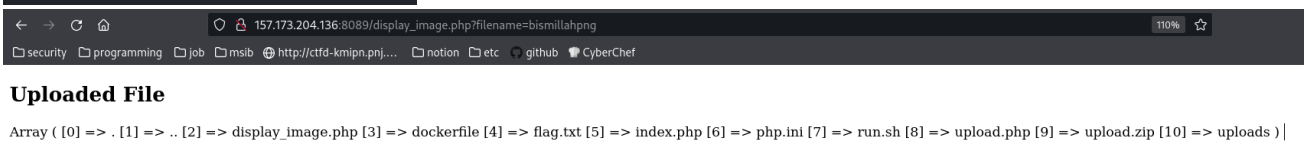


dan ternyata berhasil melakukan upload file php dan dapat menjalankan php_info disini, disini pun kami mencari sebanyak banyaknya informasi mengenai websitenya dan menemukan bahwa websitenya tidak dapat menjalankan beberapa perintah di linux seperti :

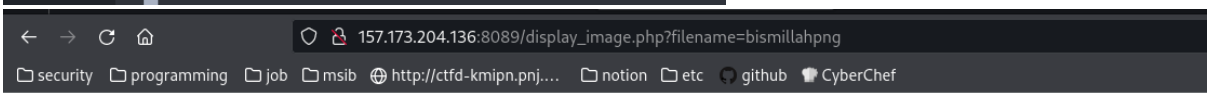
disable_functions	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,error_log,system,exec,shell_exec,proc_open,passthru,link,symlink,syslog,ld,mail	pcntl_alarm,pcntl_fork,pcntl_waitpid,pcntl_wait,pcntl_wifexited,pcntl_wifstopped,pcntl_wifsignaled,pcntl_wifcontinued,pcntl_wexitstatus,pcntl_wtermsig,pcntl_wstopsig,pcntl_signal,pcntl_signal_get_handler,pcntl_signal_dispatch,pcntl_get_last_error,pcntl_strerror,pcntl_sigprocmask,pcntl_sigwaitinfo,pcntl_sigtimedwait,pcntl_exec,pcntl_getpriority,pcntl_setpriority,pcntl_async_signals,error_log,system,exec,shell_exec,proc_open,passthru,link,symlink,syslog,ld,mail

Lalu setelah beberapa referensi kami menemukan bahwa scandir tidak di disable maka saya pun menggunakan function tersebut untuk mendapatkan lokasi dari flagnya:





Dan berhasil!!!, lokasi flag ada di lokasi yang sama dengan websitenya jadi kami mencari cara bagaimana untuk membaca flag tersebut, kami pun mendapatkan function file_get_contents dan tidak di disable di php website:



Uploaded File

KMIPNVIPNJ{disable_function_on_upload_file_exploit}

dan flag pun berhasil didapatkan

Flag : KMIPNVIPNJ{disable_function_on_upload_file_exploit}

2. quotes

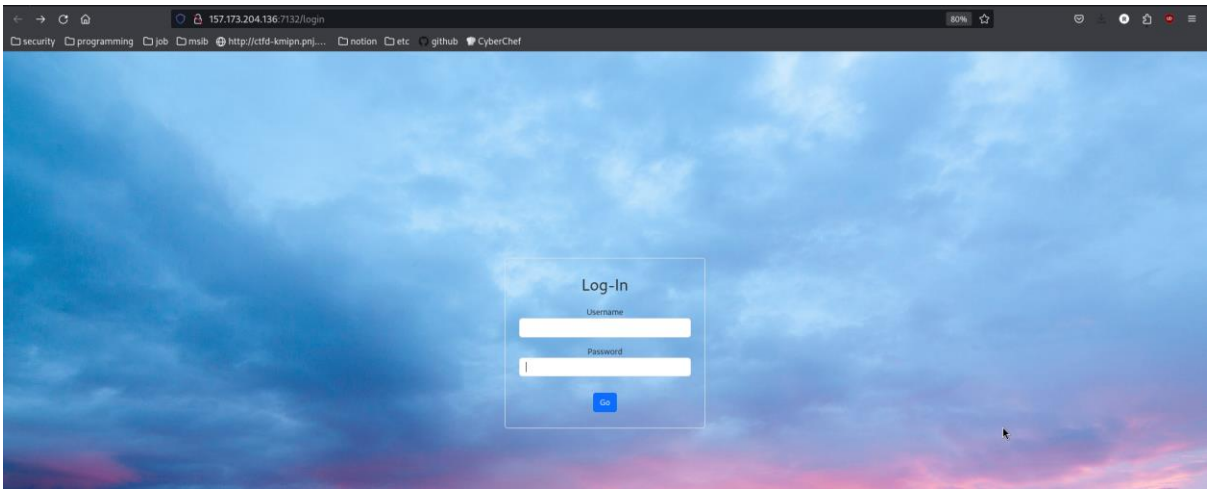
NO NO NO...

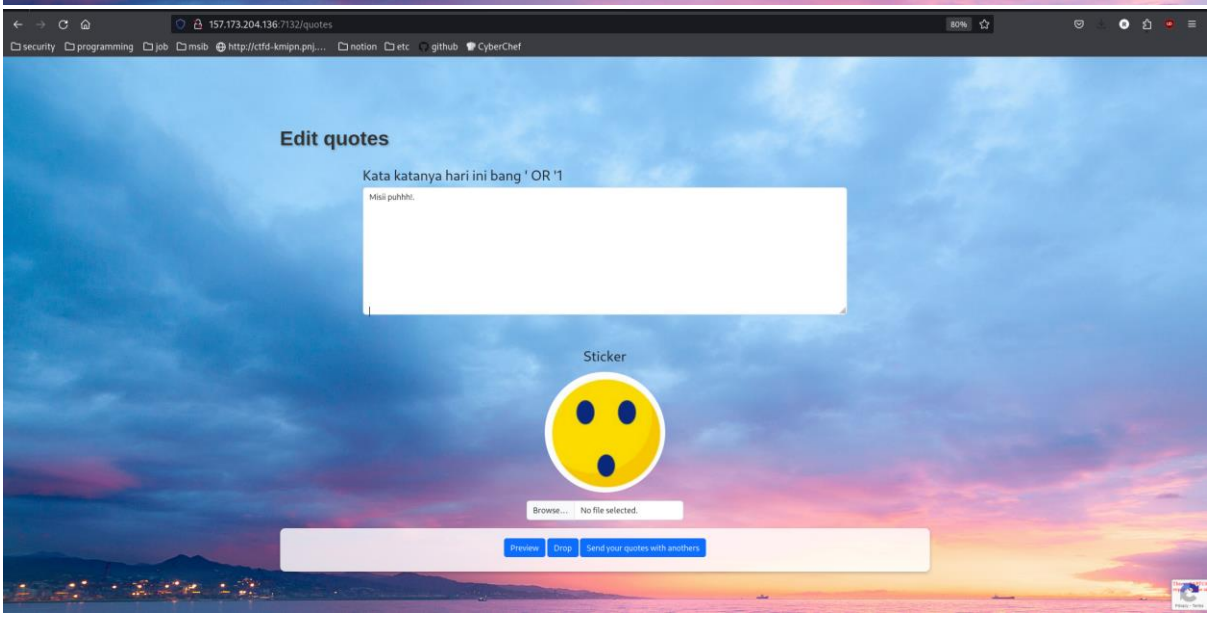
Tak selamanya "content secure sumber" yg terlihat aman benar² aman

http://157.173.204.136:7132

Point : 500

Solution: Diarahkan ke sebuah web dengan form login:





Kami belum berhasil mengeksploitasi kerentanan hingga flag-nya.

Flag : —

Cryptography

1. Reality Club

Alat enkripsi underrated yang TERBUKTI aman dari serangan hengker hengker jahat yang meiliki intensi merusak!! benarkah begitu? apakah ada kelemahan dalam enkripsi ini yang sudah banyak orang tahu?

nc 157.173.204.136 4423

Point : 500

Solution: Di soal ini merupakan soal encryption dengan rc4, kami pun mencoba mencari kerentanan yang ada di rc4 sesuai dengan soal yang diberikan dan mendapatkan beberapa referensi yang sesuai yaitu :

- https://ctftime.org/writeup/26072
- https://ctftime.org/writeup/26153
- https://github.com/dj311/rc4-key-recovery-attacks
- https://alvinferd.medium.com/writeup-cryptography-technofair-8-0-ctf-c8efb7abc5b0

Karena encryption mereka dilakukan secara random maka saya menggunakan pwntools untuk mempermudah prosesnya:

```python
from pwn import xor, remote

r = remote('157.173.204.136', 4423)

plaintext = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!\"#$%&'()*+,-./:;
<=>?@[\\]^_`{|}~"
r.sendlineafter(b'What you want to do?\n', b'1')
r.sendlineafter(b'Enter your message\n', plaintext.encode())
r.recvuntil(b'encrypted : ')
encrypted_message = r.recvline().strip().decode()

r.sendlineafter(b'What you want to do?\n', b'2')
r.recvuntil(b'encrypted : ')
encrypted_flag = r.recvline().strip().decode()

r.close()

ct1 = bytes.fromhex(encrypted_message)
enc_flag = bytes.fromhex(encrypted_flag)

keystream = xor(plaintext.encode(), ct1)

flag = xor(enc_flag, keystream)[:len(enc_flag)]
print(flag.decode())
```

Script Python di atas berguna untuk mengambil dan melakukan enkripsi dari teks yang tersedia di variabel plaintext. Menggunakan XOR untuk mendapatkan keystream dari plaintext dan ciphertext. Kemudian, keystream tersebut digunakan untuk mendekripsi flag terenkripsi:

```
┌──(kali㉿kali)-[~/…/jeopardy/Cryptography/Reality Club/solve]
└─$ python solve.py
[+] Opening connection to 157.173.204.136 on port 4423: Done
[*] Closed connection to 157.173.204.136 port 4423
KMIPNVIPNJ{4j4K_d4n_B4Wa_Aku_k3_Dun14Mu_y4n9_1nd4h_N4n_M394h_1tu_Fl0rAA4A!!!!!!>____<}
```

setelah dirunning maka flagpun berhasil ditemukan

Flag :

KMIPNVIPNJ{4j4K_d4n_B4w4_Aku_k3_Dun14Mu_y4n9_1nd4h_N4n_M394h_1tu_Fl0eAA4A!!!!!!

>____<}

2. Pal Signer

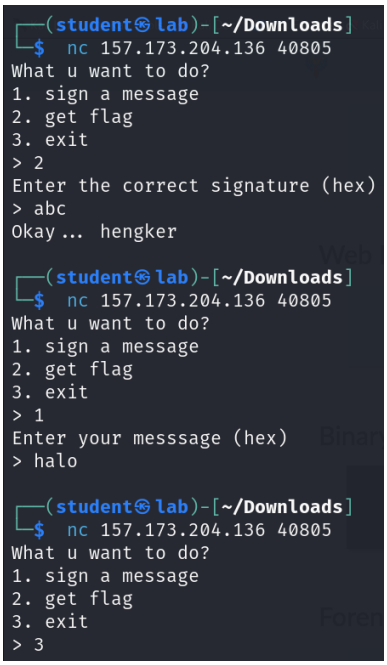
ayok ayok yang mau tanda tangan

bisakah kamu membuat tanda tangan palsu?

nc 157.173.204.136 40805

Point : 500

Solution: Ketika kita coba akses program melalui netcat terdapat 3 opsi yang bisa dilakukan:



Berikut isi dari program pailier.py. Program ini mengimplementasikan skema enkripsi Paillier, yang merupakan skema enkripsi kunci publik homomorfik. Ini adalah implementasi sederhana dari skema enkripsi Paillier, yang memungkinkan operasi aritmatika pada ciphertext:

```python
from Crypto.Util.number import *
from math import lcm
import random


class pailier:
    def __init__(self):
        while True:
            p, q = getPrime(512), getPrime(512)
            if GCD(p*q, (p-1)*(q-1)) == 1:
                break
        self.phi=lcm(p-1,q-1)
        self.n = p * q
        self.g = self.n+1
        self.miu=inverse(self.L(pow(self.g,self.phi,self.n**2)),self.n)

    def L(self,val):
        return (val-1)//self.n

    def pubkey(self):
        return (self.n,self.g)

    def encrypt(self,msg):
        r=random.randrange(0,self.n-1)
        gm=pow(self.g,msg,self.n**2)
        rn=pow(r,self.n,self.n**2)
        ct=(gm*rn)%(self.n**2)
        return ct

    def decrypt(self,ct):
        m=self.L(pow(ct,self.phi,self.n**2))%self.n
        m=(m*self.miu)%self.n
        return long_to_bytes(m)

```

Adapun isi dari program soal.py adalah sbb. Program ini adalah aplikasi interaktif yang menggunakan skema enkripsi Paillier untuk mengelola penandatanganan pesan dan pengungkapan "flag" rahasia:

```python
from Crypto.Util.number import *
from pailier import *
from secret import flag
import random




def encrypt_flag():
    flag_arr=[ord(i) for i in flag]
    for i in range(len(flag)):
        flag_arr[i]*=random.randrange(0,random.randrange(2**128))
    return flag_arr

cipher=pailier()
while True:
    print("What u want to do?")
    print("1. sign a message")
    print("2. get flag")
    print("3. exit")
    inp=int(input("> "))
    if inp==1:
        print("Enter your messsage (hex)")
        inp=input("> ")
        inp=int(inp,16)
        if (inp>0 and inp<cipher.n**2) and (long_to_bytes(inp)!=b"bwang flagnya dong"):
            print('your signature :','{0:x}'.format(cipher.encrypt(inp)))
        else:
            print("Okay... hengker")
            exit()
    elif inp==2:
        print("Enter the correct signature (hex)")
        inp=input("> ")
        inp=int(inp,16)
        if cipher.decrypt(inp)==b"bwang flagnya dong":
            print("Here's your flag xixixixi")
            print(encrypt_flag())
        else:
            print("Okay... hengker")
            exit()
    else:
        exit()
```

Kami telah berusaha untuk menyelesaikan challenge ini, namun belum berhasil menemukan flag-nya.

Flag : —

R
everse Engineering

1. Clown

An instance on another planet has just been cyber attacked, help them recover their imporant

data.

*Highly recommended to create an empty directory when debugging it

Point : 500

Solution: Diberikan sebuah file zip, ketika kami ekstrak hasilnya adalah sbb:

```
┌──(student㉿lab)-[~/Downloads]
└─$ unzip release.zip
Archive:  release.zip
   creating: release/
  inflating: release/free_vbucks.pyc
   creating: release/my_precious_folder/
  inflating: release/my_precious_folder/3cb6f905e5b4c904607bcba0282fd9f8.clown
  inflating: release/my_precious_folder/4fae74f5f173c6337764921545af64ee.clown
  inflating: release/my_precious_folder/79469bb82c755664e011a7c3ad1acb44.clown
  inflating: release/my_precious_folder/7e6e41e4ca7b3b7ccd04bcb1e0e2cebe.clown
  inflating: release/my_precious_folder/901c3e57149cdcdfe704cd75a6eb3e42.clown
  inflating: release/my_precious_folder/91b0a1a2b68abfabda1ca1bb24f24e21.clown
  inflating: release/my_precious_folder/b048da1d9367b05e7a07c39059700270.clown
  inflating: release/my_precious_folder/c5fca26a4d36d3818aacf16b32701770.clown
  inflating: release/my_precious_folder/d0fde56c56a9dbeb5a38777caa70e0ae.clown
  inflating: release/my_precious_folder/e6b968ccf26f89419b6c80b9028e064a.clown
```

Diketahui file free_vbucks.pyc adalah hasil compile program python dan semua file dengan

ekstensi .clown tidak diketahui informasi lebih lanjut:

```
┌──(student㉿lab)-[~/Downloads]
└─$ file release/free_vbucks.pyc
release/free_vbucks.pyc: Byte-compiled Python module for CPython 3.10, timestamp-based,
87 bytes

┌──(student㉿lab)-[~/Downloads]
└─$ file release/my_precious_folder/3cb6f905e5b4c904607bcba0282fd9f8.clown
release/my_precious_folder/3cb6f905e5b4c904607bcba0282fd9f8.clown: data
```

Kami telah berusaha memahami dan solving challenge ini, namun belum berhasil mendapatkan

flag.

Flag : —

2. Encryptinator

KUPERSEMBAHKAN SISTEM ENKRIPSI MUTAKHIR YANG DAPAT MEMPORAK-PORAKKAN

KETAHANAN KRIPTOGRAFI REPUBLIK ISEKAI

Point : 500

Solution: Diberikan sebuah file tanpa ekstensi bernama chall. Setelah dilihat berikut adalah

detailnya:

```
┌──(student㉿lab)-[~/Downloads]
└─$ file chall
chall: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked,
d4d8f5046b617c935974ee72154f92a152, for GNU/Linux 3.2.0, not stripped
```

File tersebut adalah sebuah executable yang sesuai dengan format ELF (Executable and Linkable

Format) 64-bit untuk arsitektur x86-64, yang digunakan pada sistem operasi GNU/Linux.

```
1  7f45 4c46 0201 0100 0000 0000 0000 0000
2  0300 3e00 0100 0000 0011 0000 0000 0000
3  4000 0000 0000 0000 603a 0000 0000 0000
4  0000 0000 4000 3800 0d00 4000 1f00 1e00
5  0600 0000 0400 0000 4000 0000 0000 0000
6  4000 0000 0000 0000 4000 0000 0000 0000
7  d802 0000 0000 0000 d802 0000 0000 0000
8  0800 0000 0000 0000 0300 0000 0400 0000
9  1803 0000 0000 0000 1803 0000 0000 0000
10 1803 0000 0000 0000 1c00 0000 0000 0000
11 1c00 0000 0000 0000 0100 0000 0000 0000
12 0100 0000 0400 0000 0000 0000 0000 0000
13 0000 0000 0000 0000 0000 0000 0000 0000
14 6007 0000 0000 0000 6007 0000 0000 0000
15 0010 0000 0000 0000 0100 0000 0500 0000
16 0010 0000 0000 0000 0010 0000 0000 0000
17 0010 0000 0000 0000 e506 0000 0000 0000
18 e506 0000 0000 0000 0010 0000 0000 0000
19 0100 0000 0400 0000 0020 0000 0000 0000
20 0020 0000 0000 0000 0020 0000 0000 0000
21 9801 0000 0000 0000 9801 0000 0000 0000
22 0010 0000 0000 0000 0100 0000 0600 0000
```

Kami telah berusaha menemukan flag di challange ini, namun belum berhasil.

Flag : —

Binary Exploitation

1. Bad Shell

Classic challenge.. this should be easy.. right? right?

nc 157.173.204.136 40802

Point : 500

Solution: Ketika program dijalankan, kita diminta memasukkan shellcode:

```
┌──(student㉿lab)-[~/Downloads]
└─$ nc 157.173.204.136 40802
Gimme your shellcode : 123

┌──(student㉿lab)-[~/Downloads]
└─$ nc 157.173.204.136 40802
Gimme your shellcode : abc
```

Setelah beberapa percobaan dan analisa, kami masih belum berhasil mendapatkan flag dari challenge ini.

Flag : —

2. Gets The Flag Out

There's a leak in our feedback BAAS(Binary as a Service). However the binary still working properly, even tho the file descriptor is closed.

nc 157.173.204.136 40801

Point : 500

Solution: Kita diminta untuk memasukkan dua input di program ini:

```
┌──(student㉿lab)-[~/Downloads]
└─$ nc 157.173.204.136 40801

                ____________________________________

                Damn Vulnerable Feedback Application

                ____________________________________

        [!] ALERT: Some data has been leaked [0×72d2a0]!
        [*] Restoring System
        [#] Hi There! We need your advice to improve our challenge
        [?] Advice      : pleasee
        [?] Team Name   : halo

Thank you. Good Luck Have Flag!
```

Setelah beberapa percobaan dan analisa, kam masihi belum berhasil menemukan leak maupun celah dari challenge ini.

Flag : —